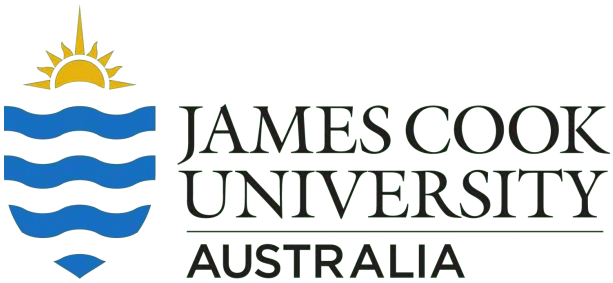


DOCTORAL THESIS

Light Field Reconstruction from Multi-View Images

Kang HAN

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

College of Science and Engineering

October 30, 2022

Acknowledgements

I would like to express my sincere gratitude to my supervisors Prof. Wei Xiang and Dr. Eric Wang. They brought me into the area of light field and I was thus able to start the journey of my four years of study and research. Prof. Wei Xiang gave me the great freedom to explore the topics that interested me. His visionary thoughts and broad knowledge have inspired me to think about the fundamental problems behind the research topics. He taught me what excellent research should look like and how to achieve it. I learned how to conceptualize research ideas and how to write clear high-quality papers. The principles, knowledge, and experience I gained from him have guided me all the time in my research and writing processes. He has also supported me in many aspects. Every time I had difficulties, he was always there to provide help. I also had his full support to visit La Trobe University in 2022, where I had a great time and met kind new friends.

My secondary supervisor Dr. Eric Wang has also taught me a lot about research in light field, as well as more generally image and video processing. I did enjoy every discussion with him and I learned much from him. His practical experience in light field has inspired me in many aspects. In the future, I will also focus on the application of light field reconstruction in real-world scenes from the practical point of view I learned from him.

I would also like to thank my friends Dr. Kevin Huang, Owen Matthew, Neethu Madhukumar, Dr. Yu Han, and Bing Wang. Thank them for many interesting discussions about research and life. And also thank them for making my PhD time colorful and enjoyable.

Lastly, I would like to thank my family for their support. My family takes care of things in life so that I can focus on my research to complete the degree. I dedicate this thesis to my family with gratitude.

Statement of the Contribution of Others

Nature of Assistance	Contribution	Names
Supervision	Primary supervision Secondary supervision	Prof. Wei Xiang Dr. Eric Wang
Intellectual support	Conceptualization Data analysis Paper/thesis revision Proofreading	Prof. Wei Xiang Dr. Eric Wang
Financial support	Tuition fee scholarship	James Cook University
Experiment	GPU resource Data storage	High Performance Computing in James Cook University
Infrastructure	Working office Computer	La Trobe University

Abstract

Light field is defined as the outgoing radiance at a point in a given direction. It is the result of the interaction between the incoming light and the surface with a specific material. Multi-view images captured by conventional cameras from multiple viewpoints are 2D projections of the light field. Reconstructing the underlying light field that produces the observed multi-view images is thus an inverse problem. Accurate light field reconstruction of a scene enables 3D understanding of the scene, which is important for many computer vision and machine intelligence problems. Thus, light field reconstruction is the core of many innovative technologies and applications, such as autonomous driving cars and metaverse. However, several challenges limit the practical application of light field reconstruction. Depth estimation is one of the crucial research problems in light field reconstruction, but existing work struggles to efficiently handle occlusions to preserve depth edges. In addition, effective light field representations capable of achieving photo-realistic novel view synthesis are desired to improve on the rendering quality in existing solutions. Novel algorithms dealing with these challenges will facilitate the applications of light field reconstruction in real-world scenarios.

This thesis addresses these challenges in light field reconstruction from geometric, local, and global levels. At the geometric level, the aim is to reconstruct light field geometry by depth estimation. We construct a novel cost from a new perspective that counts the number of refocused pixels whose deviations from the central-view pixel are less than a small threshold and utilizes that number to select the correct depth. We show that without the use of any explicit occlusion handling methods, the proposed method can inherently preserve edges and produces high-quality depth estimates.

Synthesizing intermediate novel views within existing views is the target of local light field reconstruction. This thesis presents an inference-reconstruction variational autoencoder to reconstruct a dense light field image out of four corner reference views in a light field image. The conditional latent variable in the inference network is regularized by the latent variable in the reconstruction network to facilitate information flow between the conditional latent variable and novel views. A viewpoint-dependent indirect view synthesis method is also introduced to synthesize novel views more efficiently by leveraging adaptive convolution.

Lastly, we reconstruct a global light field to enable photorealistic view rendering from any point and any view direction by using a novel neural radiance feature field.

We propose to use a multiscale tensor decomposition scheme to organize learnable features to represent scenes from coarse to fine scales. We demonstrate many benefits of the proposed multiscale representation, including more accurate scene shape and appearance reconstruction, and faster convergence compared with the single-scale representation. Instead of encoding view direction to model view-dependent effects, we further propose to encode the rendering equation in the feature space by employing an anisotropic spherical Gaussian mixture predicted from the proposed multiscale representation. Based on the proposed methods, we are able to reconstruct the accurate light field of a scene and achieve novel view rendering with high-fidelity view-dependent effects.

List of Publications

The following publications were produced during the period of candidature:

[1] K. Han, W. Xiang, E. Wang, and T. Huang, "A novel occlusion-aware vote cost for light field depth estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 8022-8035, Nov. 2022 (IF = 24.314). Related to Chapter 2.

[2] K. Han, and W. Xiang, "Inference-reconstruction variational autoencoder for light field image reconstruction," IEEE Transactions on Image Processing, vol. 31, pp. 5629-5644, Aug. 2022 (IF = 11.041). Related to Chapter 3.

[3] K. Han, and W. Xiang, "Neural radiance feature field for view rendering," Submitted to The IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023, under review. Related to Chapter 4.

[4] Y. Xu, K. Han, Y. Zhou, J. Wu, X. Xie, and W. Xiang, "Deep adaptive blending network for 3D magnetic resonance image denoising," IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, pp. 3321-3331, Sep. 2021 (IF=7.021).

Contents

Acknowledgements **iii**

Statement of the Contribution of Others **v**

Abstract **vii**

List of Publications **ix**

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**
1.1 Background . . . 1
1.2 Motivation . . . 2
1.2.1 Applications . . . 2
1.2.2 Challenges . . . 3
1.3 Research Questions . . . 5
1.4 Contributions . . . 5
1.5 Thesis Outline . . . 7

2 Occlusion-Aware Vote Cost for Light Field Depth Estimation **11**
2.1 Introduction . . . 11
2.2 Related Work . . . 14
2.3 Consistency in Refocused Angular Patch . . . 16
2.3.1 Consistency Analysis . . . 17
2.3.2 Pixel Deviation Histogram . . . 19
2.4 Occlusion-Aware Vote Cost . . . 21
2.4.1 Vote Cost . . . 22
2.4.2 Vote Threshold . . . 25
2.4.3 Refinement . . . 27
2.5 Experimental Results . . . 27
2.5.1 Objective Comparison . . . 29

2.5.2 Subjective Comparison . . . 33
2.5.3 Effect of the Vote Threshold . . . 33
2.5.4 Computation Time . . . 35
2.5.5 Limitations . . . 36
2.6 Conclusion . . . 37

3 Inference-Reconstruction Variational Autoencoder 39
3.1 Introduction . . . 39
3.2 Related work . . . 41
3.2.1 Light Field Angular Super-Resolution . . . 42
3.2.2 Variational Autoencoder . . . 43
3.2.3 Indirect View Synthesis . . . 44
3.3 Inference-Reconstruction Variational Autoencoder . . . 44
3.3.1 Problem Formulation . . . 44
3.3.2 Framework . . . 45
3.3.3 Mean Local Maximum Mean Discrepancy . . . 48
3.4 Encoder and Decoder Structures . . . 51
3.4.1 Viewpoint-dependent Indirect View Synthesis Method . . . 51
3.4.2 Encoder and Decoder Structures . . . 54
3.4.3 Network for Large Disparity . . . 56
3.4.4 Training and Testing Procedures . . . 57
3.5 Experimental Results . . . 58
3.5.1 Objective Results . . . 60
Results on Lytro Image Datasets . . . 60
Results on Microscope Dataset . . . 62
Results on Inria Synthetic Dataset . . . 62
3.5.2 Subjective Results . . . 63
3.5.3 Runtime and Memory Consumption . . . 63
3.5.4 Ablation Study . . . 65
3.5.5 Visualization of Viewpoint-dependent Adaptive Kernels . . . 66
3.5.6 Limitations . . . 67
3.6 Conclusion . . . 68

4 Neural Radiance Feature Field for View Rendering 69
4.1 Introduction . . . 69
4.2 Related Work . . . 71
4.2.1 Neural Representations . . . 71
4.2.2 Learnable Feature Representations . . . 72
4.3 Method . . . 72

4.3.1 Multiscale Tensor Decomposition 73
4.3.2 Rendering Equation Encoding 75
4.3.3 Volume Rendering 78
4.3.4 Training loss 79
4.4 Experiments 79
4.4.1 Objective Results 80
4.4.2 Subjective Results 81
4.4.3 Ablation Study 82
4.4.4 Limitations 83
4.5 Conclusion 85

5 Conclusion **87**

Bibliography **91**

List of Figures

1.1 Light field definition 2
1.2 Light field reconstruction levels 7

2.1 Occlusion handling and estimated depth by different methods 13
2.2 Refocused angular patches at correct and incorrect disparities 17
2.3 Consistency analysis 18
2.4 Pixel deviation histogram comparison 20
2.5 Effectiveness of the basic vote cost and the distinguishing cost 23
2.6 Comparison of different depth costs 24
2.7 Initial depth estimates using different costs 24
2.8 Effectiveness of the initial depth refined by the weighted median filter . 28
2.9 Visual comparison of estimated depth on synthetic datasets 34
2.10 Visual comparison of estimated depth on Stanford dataset 35

3.1 Overall framework of the proposed inference-reconstruction variational autoencoder (IR-VAE) 46
3.2 Process comparison between the conditional variational autoencoder (CVAE) and the proposed IR-VAE 47
3.3 Schematic of the indirect view synthesis method 51
3.4 Structure of encoder $p_{\boldsymbol{\psi}_1}(\mathbf{r}|\mathbf{x}_r)$ 52
3.5 Visual comparison of error maps of synthesized novel views 64
3.6 Subjective comparison of synthesized novel views for the Rock scene . . 65
3.7 Visualization of viewpoint-dependent adaptive kernels 68

4.1 Multiscale tensor decomposition representation 73
4.2 Visualization of plane feature maps of different resolutions 74
4.3 Illustration of the proposed rendering equation encoding 76
4.4 Subjective comparison of rendered views 81
4.5 Visualization of learned ASG functions and reconstructed light fields . . 82
4.6 Performance comparison over training steps for varying the number of scale levels 84

List of Tables

2.1 Comparison of the cumulative pixel probabilities 21
2.2 MSE results 30
2.3 BadPix results 31
2.4 Backgammon Fattening results 32
2.5 Q25 results 32
2.6 BadPix performance with varying vote thresholds 33
2.7 Average run time 36
2.8 Surface normal accuracy 37

3.1 Summary of properties of the encoders and decoders 54
3.2 Training and testing datasets 58
3.3 Objective quality comparison on Lytro LF image datasets 60
3.4 Objective quality comparison when only reconstructing the Y channel . 61
3.5 Objective quality comparison on the Microscope dataset 62
3.6 Objective quality comparison on Inria synthetic dataset 63
3.7 Runtime and memory consumption 66
3.8 Ablation study of the proposed method 67

4.1 Objective performance comparison of neural rendering results 80
4.2 Ablation study on the NeRF synthetic dataset 83

Chapter 1

Introduction

1.1 Background

Light field is defined as the radiance at a point in a given direction [1]. The spatial position of a point is determined by the 3D Cartesian coordinate system, while the direction is defined by the 2D spherical coordinate system. Thus, the light field is a 5D function $L(x, y, z, \theta, \phi)$ as shown in Fig. 1.1. The radiance here includes the light intensity and wavelength. The light intensity controls the brightness, while the wavelength determines the color. From the perspective of human vision, a light field evaluation for a 5D input yields a kind of color fused with brightness. Indeed, the light field is the resultant outgoing radiance of complex physical interactions of light sources at surface points with different materials. These interactions may include transmission, diffusion, reflection, etc., depending on the surface materials. Different types of ray/surface interactions produce various outgoing radiance for a considered light source in a given direction. Thus, a spatial point may show diverse colors when viewed from different directions.

Currently, there is no tool to directly capture the light field of a scene. The high dimensionality of the light field makes its acquisition by dense sampling impractical. Sampling here means recording the radiance at a point in a given direction. From this point of view, a 2D photograph captured by a common camera can be seen as a sample of the light field at the camera position with a bunch of radiance directions (each pixel corresponds to one radiance direction). As shown in Fig. 1.1, two cameras capture images of the scene from different positions and viewpoints. The resultant images are samples of the light field. Simply performing dense sampling by cameras is impractical for capturing the light field because the 5D space contains a large number of samples.

In lieu of dense sampling, recovering the properties of the scene from observed multi-view images is a more tractable approach to reconstructing the light field. As aforementioned, the light field is the resultant outgoing radiance after ray/surface interactions. One can estimate the scene geometry and appearance that explain the observed multi-view images, and then simulate the interactions to reproduce the light

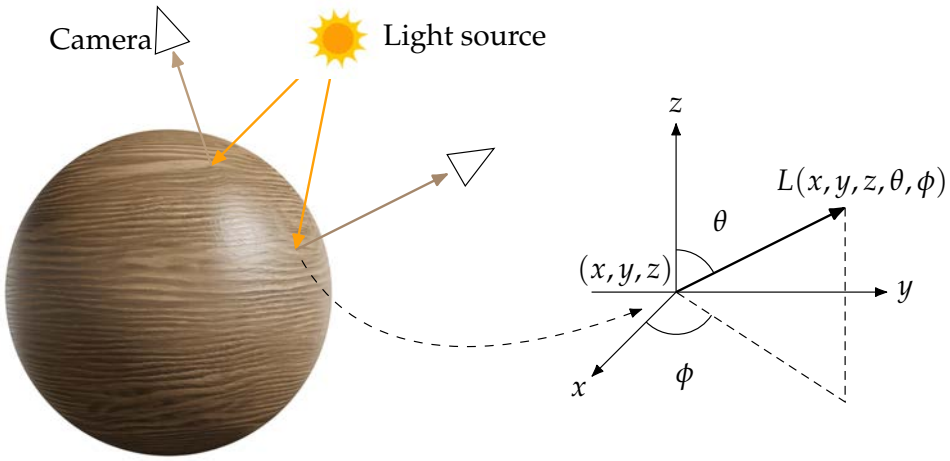


Figure 1.1: Light field is defined as the radiance at a point in a given direction. The light field of a scene is the resultant outgoing radiance of complex interactions of light source and surface. Images captured by conventional cameras are samples of the light field.

field. In this way, the task of light field reconstruction is simpler because a scene's geometry and appearance are much easier to reconstruct than the 5D light field. Numerous efforts have been made in this direction in both computer vision and computer graphics communities in the last few decades, as discussed in the review in [2].

Reconstructing the light field is still a challenging task even from the perspective of estimating its underlying geometry and appearance. Light field reconstruction from multi-view images based upon computer vision and computer graphics techniques is an inverse problem. An image captured by a camera is a 2D projection of the 5D light field. The projection is not invertible because points at different depths in a cast ray from the camera will be projected to the same pixel position. Multi-view images are used to resolve this ambiguity, but challenges including occlusions and effective representations of geometry and appearance limit the accuracy of light field reconstruction. Thus, effective approaches that can address these challenges are desired to facilitate wide-ranging light field applications.

1.2 Motivation

1.2.1 Applications

Accurate light field reconstruction could significantly advance research in computer vision and machine intelligence, powering many innovative applications. The light field contains more comprehensive geometry and appearance information of objects, which is very helpful for many fundamental computer vision and pattern recognition problems such as image recognition [3], object detection [4], and object segmentation

[5]. Also, the knowledge of objects' size and position in the light field is critical to understanding the 3D environment for autonomous driving. For example, estimating the depth from two or more views provides distance perception for autonomous driving cars. More recently, in 2022, the leading autonomous driving company Tesla introduced their latest full self-driving solution based on the occupancy network [6] trained by 3D data produced by the neural radiance field (NeRF) [7]. Similarly, the reconstructed light field by the NeRF has also been used to supervise robust robot vision systems to overcome the limitation that RGB-D sensors do not work well for reflective materials [8].

Another important application of the light field is Metaverse. Metaverse is expected to provide an immersive viewing experience of a virtual world by using virtual reality or augmented reality headsets. To be immersive, a rendered image from a user's position and viewing direction needs to be photo-realistic as viewing the real world [9]. Achieving this goal requires accurate light field reconstruction of a scene. One can render a photo-realistic view by evaluating the reconstructed light field with an arbitrary 5D input. Besides, augmented reality also requires accurate light field reconstruction to understand the real world to mix virtual and real contents. The new viewing and interaction methods provided by Metaverse have already enabled representative applications in many areas including entertainment, social media, education, and industry [10], [11].

1.2.2 Challenges

A key problem in light field reconstruction is how to represent the light field. Different application scenarios require different representations to facilitate the following processing, e.g., position and size perception and novel view rendering. Depending on whether a neural component is used to represent the light field, we follow the work in [2] that divides light field representation into conventional and neural representations. Conventional representations include point clouds, polygonal meshes, volume density, etc. These representations have an explicit geometry structure and can be easily integrated into the framework of computer graphics to achieve view rendering. For instance, each point in point clouds represents the spatial position of a point on an object, while meshes represent the surface of an object. Point clouds are also the output format of 3D data acquisition devices and algorithms. For example, an estimated depth map from multi-view images is related to point clouds. To reconstruct a real-world light field, depth estimation is usually the first step to obtain light field geometry, which helps reason the positions and sizes of objects for applications such as autonomous driving and robot vision systems. Thus, light field depth estimation [12] is important

for applications relying on accurate light field geometry reconstruction. In the field of depth estimation, handling occlusions to preserve depth edges is one of the main challenges. Many efforts have been made to deal with occlusions in the literature [13]–[15], but they involve complex processing steps, which are not effective and efficient.

The strong learning ability of deep learning enables light field representation by deep neural networks. One can treat a deep neural network as a function that represents the light field, where the inputs to the network are observed multi-view images and the goal is to predict the radiance of the light field for a given point position and view direction. An advantage of such a neural representation is that training is conducted in an end-to-end manner, meaning that the network is optimized to represent the geometry and appearance of the light field simultaneously.

We can further divide neural representations into two categories, namely feed-forward prediction [16]–[19] and per-scene optimization approaches [7], [20]–[23]. The feed-forward prediction approach follows an ordinary deep learning pipeline in which the networks are trained on a large amount of data to learn the relationship between input multi-view images and output novel views of the light field. The trained network can be generalized to other scenes. In this approach, an encoder is usually used to encode the input to a representation, and a decoder is followed to map the representation to the target output. For example, given some multi-view images, we want to synthesize novel views at novel viewpoints. Such view synthesis is usually conducted locally: predict the novel view from nearby existing views, involving reconstructing the light field in a small space. The limitation of existing solutions in this direction is that the network does not fully utilize the ground truth images to learn to produce a good light field representation, which is important for the subsequent light field view rendering quality.

Instead of learning the relationship that can be generalized to new scenes, a typical per-scene optimization approach optimizes a neural network, typically a multilayer perceptron (MLP), to represent the light field that fits the observed images. Such optimization needs to be done for each scene so that the full capability of the neural network is used to represent one scene. As a result, a global light field is reconstructed such that we can render novel views from any point in any view direction [7]. One can also understand this approach from the perspective of viewing the neural network as a universal function approximator to represent the light field function. The input to the MLP is the point position and view direction, and the MLP predicts its density and color. Such a neural representation is very compact: only weights in the MLPs are required. However, the computational cost of pure MLP-based representation is very high because hundreds of MLP evaluations are normally required to render a single pixel.

Recent research has shown that using extra learnable features can significantly accelerate the optimization and rendering processes [2]. Efficient data structures to organize these learnable features are desired to effectively represent the light field and keep the number of learnable features as small as possible. Besides, view direction encoding methods using basis functions have been widely used to facilitate the learning of view-dependent effects. These encoding methods improve the rendering performance but neglect the fact that the outgoing radiance is produced by complex ray/surface interaction that is usually modeled by the rendering equation [24]. Thus, the MLPs need to be large to model view-dependent effects [21], which increases the computational complexity and memory consumption.

In summary, light field reconstruction is key to many important applications but challenges in accurate light field reconstruction still exist. Novel solutions or algorithms that address these challenges can promote industrial applications and inspire researchers in many related fields. Thus, we are motivated to make progress in light field reconstruction by tackling these challenges.

1.3 Research Questions

According to the above discussions, this thesis aims to study the following research questions:

- How can we effectively deal with occlusions to preserve edges in light field depth estimation?
- How can we obtain a good local light field representation to render intermediate views from existing neighboring views?
- How can we compactly and effectively represent a global light field such that view-dependent effects can be well modeled?

1.4 Contributions

We have proposed novel approaches to address the aforementioned research questions. The advancement of knowledge and related publications are summarized as follows:

- We propose a novel occlusion-aware vote cost (OAVC) that is able to accurately preserve edges in the estimated depth map from a light field image. Instead of using photo-consistency as the indicator of the correct depth, we construct a

novel cost from a new perspective that counts the number of refocused pixels whose deviations from the central-view pixel are less than a small threshold and utilizes that number to select the correct depth. The pixels from occluders are thus excluded in determining the correct depth. Without use of any explicit occlusion handling methods, the proposed method can inherently preserve edges and produce high-quality depth estimates. Thanks to its simplicity, the proposed method is of low computational complexity and runs faster than existing methods on both CPU and GPU.

Related publication: K. Han, W. Xiang, E. Wang, and T. Huang, "A novel occlusion-aware vote cost for light field depth estimation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 8022-8035, Nov. 2022 (IF = 24.314).

- We propose an inference-reconstruction variational autoencoder (IR-VAE) that can facilitate information flow between latent variables and novel views for local light field reconstruction. We further propose a statistic distance measurement method dubbed the mean local maximum mean discrepancy (MLMMD) to measure the distance between two distributions with high-dimensional variables. Lastly, we propose a viewpoint-dependent indirect view synthesis method based on adaptive convolution.

  Related publication: K. Han, and W. Xiang, "Inference-reconstruction variational autoencoder for light field image reconstruction," IEEE Transactions on Image Processing, vol. 31, pp. 5629-5644, Aug. 2022 (IF = 11.041).

- We propose a neural radiance feature field (NRFF) to reconstruct a global light field that is capable of photo-realistic view rendering. We first propose a multi-scale tensor decomposition scheme to represent scenes from coarse to fine scales, enabling better rendering quality and fast convergence using fewer parameters than its single-scale counterpart. We then propose to encode the rendering equation using the anisotropic spherical Gaussian mixture in the feature space. Thus, the subsequent MLP is aware of the rendering equation so as to model complex view-dependent effects. Using the proposed NRFF, we significantly improve the rendering quality by over 1 dB in PSNR on the two widely used datasets.

  Related paper: K. Han, and W. Xiang, "Neural radiance feature field for view rendering," Submitted to The IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023, under review.

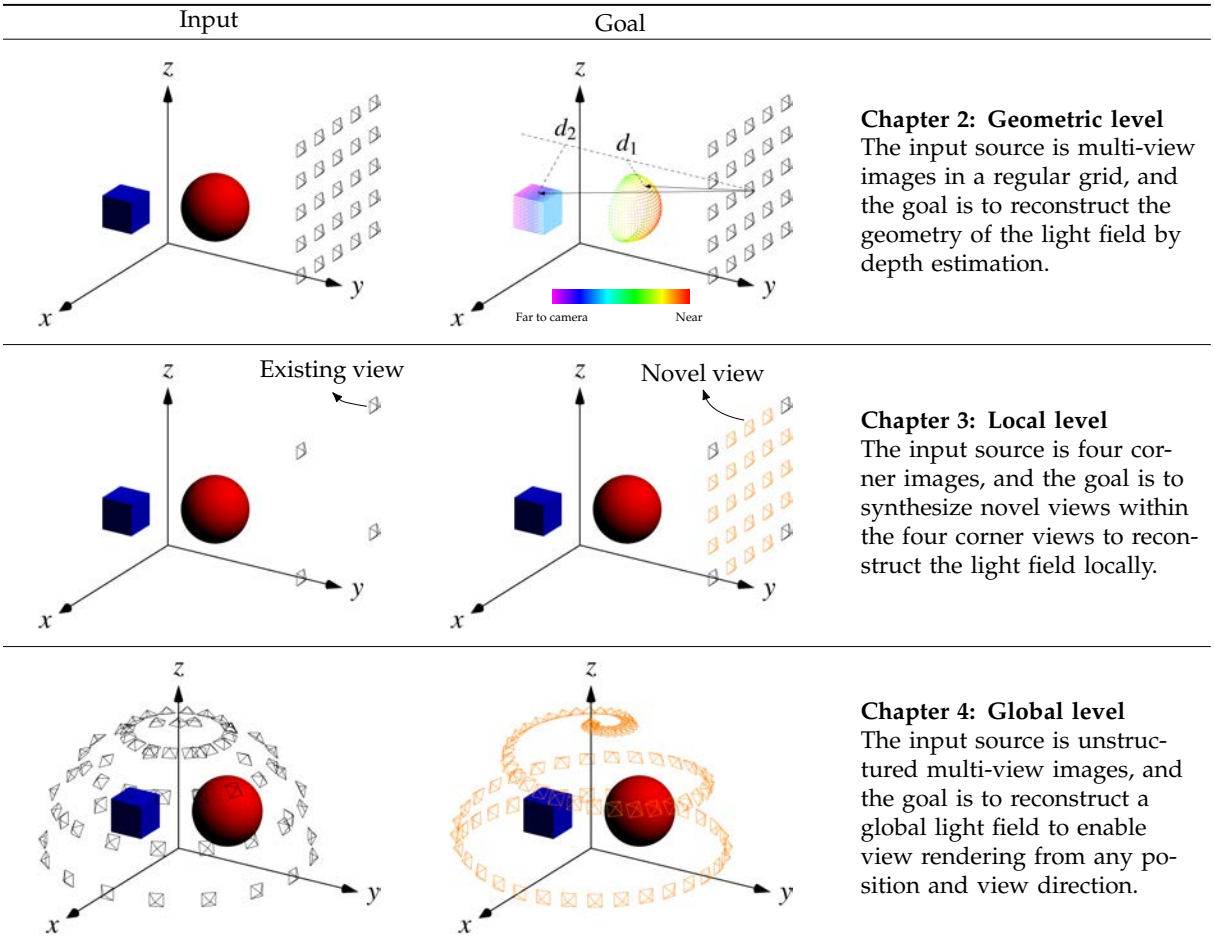


Figure 1.2: Light field reconstruction levels. At the geometric level, the input multi-view images captured in a regular grid are used to estimate the distances or depths between cameras and objects, providing geometric information such as objects' positions and sizes for vision systems to understand the 3D environment. At the local level, the input is four corner images captured in a small local area, so light field reconstruction by synthesizing novel views can be conducted locally bounded by the input images. At the global level, the input is unstructured multi-view images captured at many distinct positions and viewpoints that roughly cover a scene globally. The comprehensive information in the input images enables global light field reconstruction to render novel views at any position and view direction.

1.5 Thesis Outline

This thesis divides light field reconstruction into the geometric, local, and global levels based on the idea in [25]. As depicted in Fig. 1.2, each level has similar input multi-view images, while the goal changes from the light field geometry reconstruction to the local and global light field reconstructions at various levels. Each level of reconstruction has its corresponding applications. For instance, robot vision systems concerned more with scene geometry and immersive viewing experience requires global light field reconstruction to rendering photo-realistic images from any position and view direction. This thesis addresses the three-level reconstructions by elaborately designed

fast method, feed-forward prediction, and per-scene optimization approaches, respectively. Although global light field reconstruction is more accurate than its geometric or local-level counterparts, it needs more captured views and requires time-consuming per-scene optimization. Therefore, it is not practical for some applications at the time being.

Chapter 2 analyzes the consistency in refocused angular patches and describes a discovery that the consistency in unoccluded regions with correct refocusing is higher than that with incorrect refocusing. A quantitative analysis of the consistency by use of the pixel deviation histogram is given to show that refocused pixels with large pixel deviations (caused by occlusion or incorrect refocusing) have a negative effect on depth estimation. We then formulate the proposed vote cost by use of a threshold and deduce a simple form of vote cost when adding a distinguishing cost to deal with the scenario of an identical basic vote cost. This chapter also introduces an adaptive threshold method to adaptively determine the vote threshold based on local contextual information in the central spatial image. Experimental results are presented in this chapter to show that the proposed vote cost is able to achieve state-of-the-art performance in terms of depth estimation accuracy and computational speed.

Chapter 3 describes the novel inference-reconstruction variational autoencoder (IR-VAE) framework to synthesize novel views for the purpose of reconstructing local dense light field images. Starting with the standard variational autoencoder (VAE), this chapter discusses the problem of applying VAE in the content of light field reconstruction, and how the proposed IR-VAE solves that problem. To enable richer representations of reference views and viewpoints by high-resolution latent variables, we present a mean local maximum mean discrepancy (MLMMD) to measure the statistical distance of two distributions in the high-dimensional latent variable space. Finally, a viewpoint-dependent indirect view synthesis method capable of transforming the prediction of raw novel pixels into adaptive kernels and bias is introduced. An ablation study is conducted to show the effectiveness of the proposed modules. Experimental results are presented to demonstrate that the proposed model significantly outperforms existing state-of-the-art methods on both subjective and objective comparisons.

Chapter 4 presents the neural radiance feature field (NRFF) to represent scenes in the feature space. This chapter first introduces a multiscale tensor decomposition scheme to organize learnable features to represent scenes from coarse to fine scales. We demonstrate many benefits of the proposed multiscale representation, including more accurate scene shape and appearance reconstruction, and faster convergence compared with the single-scale representation. Then, this chapter describes how to encode the rendering equation in the feature space by employing anisotropic spherical Gaussian mixture predicted from the proposed multiscale representation. Lastly, experimental

results on both synthetic and real-world datasets are provided to demonstrate the efficacy of the proposed NRFF.

Chapter 5 concludes this thesis and discusses possible future work in light field reconstruction from different points of view.

Chapter 2

Occlusion-Aware Vote Cost for Light Field Depth Estimation

In this chapter, we reconstruct light field geometry by estimating a depth map from an light field image. Blurry edges in depth maps caused by occlusions are the key issue in light field depth estimation. We analyze the consistency properties in refocused angular patches, and reveal a new perspective to handle the occlusion problem. Based on this analysis, we propose the occlusion-aware vote cost that counts the number of refocused pixels whose deviations from the central-view pixel are less than a small threshold, and utilizes that number to select the correct depth. We demonstrate that the proposed OAVC is effective in dealing with occlusion, and computationally efficient due to its simplicity.

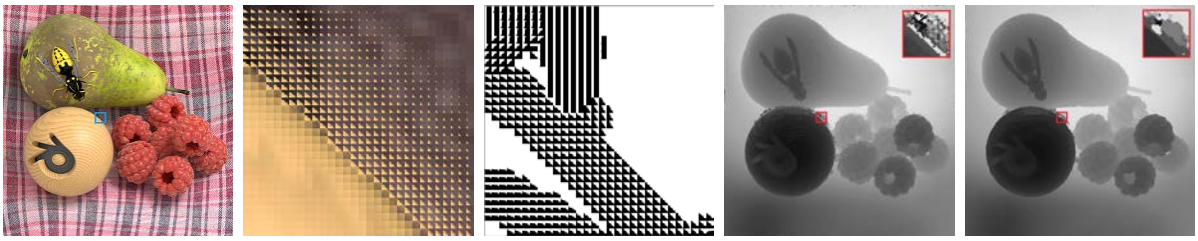
2.1 Introduction

Light field describes the distribution of light rays that are reflected from 3D points in the free space. Conventional photography records the intensities of light rays from multiple directions to a pixel by a camera with one main lens and forms a 2D image, which inevitably loses the information of light directions. In comparison, a typical light field imaging system captures not only the intensities of light rays in the 2D spatial domain, but their directions in the 2D angular domain, resulting in a common 4D representation of a light field image [26]. For a 4D light field image, fixing its angular coordinates means observing the scene from a fixed angle, which leads to a general 2D image (also called a view). On the other hand, fixing its spatial coordinates means gathering pixels at the same spatial positions from different angular views, which can form an angular patch. The angular patch is usually the raw structure of light field images that are captured by light field cameras based on a microlens array. An example of a 4D light field image in [12] provides a good visualization of the structure of light field images.

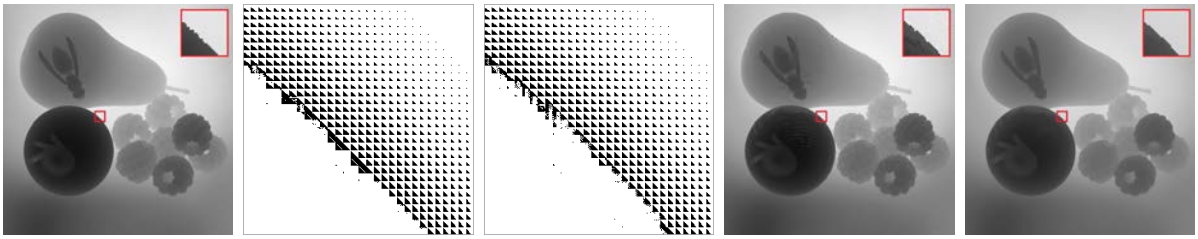
The extra light direction information in light field images enables new methods for recovering 3D geometry information from images. Light field depth estimation is one of the key research problems[12]. Existing work on light field depth estimation is based on a common assumption that pixels refocused from the correct depth in the angular patch are photo-consistent [12], [26]. Correspondence and defocus are two typical costs that are used to measure photo-consistency, and the highest consistency indicates the selection of the optimal depth among cost volumes. In refocused angular patches, the correspondence cost calculates standard deviation, while the defocus cost measures the mean absolute difference between the central-view pixel and other pixels. The correspondence and defocus costs work well in most regions and some researchers combined them to achieve better accuracy [13], [27]–[29].

However, photo-consistency is broken where occlusion occurs, which results in erroneous depth estimates. The correspondence and defocus costs generate very blurry depth estimation in occluded areas because pixels in the refocused angular patch may come from occluders [30]. To address this problem, researchers proposed occlusion models to exclude pixels from occluders to ensure photo-consistency [14], [30]–[32]. For example, Wang *et al*. [14], [30] showed that the edge separating the unoccluded and occluded pixels in the angular patch has the same orientation as the occlusion edge in the spatial domain. Based on this occlusion model, the authors separated the angular patch into two regions according to the edge orientation in the spatial domain, and only measured photo-consistency in the unoccluded region. Zhu *et al*. [31] extended Wang's work to the context of multi-occluder occlusion. The experimental results showed the effectiveness of integrating the occlusion models into light field depth estimation.

However, several drawbacks of existing occlusion handling methods limit their accuracy and computational performance. Explicit occlusion models [14], [30], [31] rely on edge detection in the spatial domain, which is, however, hard to ensure that occlusions are correctly detected. Also, these models tackle occlusion without consideration of the occlusion diffusion phenomenon. As can be seen from Fig. 2.1, the occlusion map (c) generated by occlusion-aware depth estimation (LF_OCC) [14] is far away from the real occlusion in (g). We note that occlusion not only happens along the edge between the foreground and background but also neighboring refocused angular patches along the normal line of the edge. As shown in Figs. 2.1 (b) and (g), occlusion in the refocused angular patch based on the true depth diffuses gradually along the normal line of the depth boundary. Therefore, these methods still need complex post-refinement algorithms like the Markov Random Field (MRF) to further enhance the estimated depth. Besides, existing occlusion-aware costs including the constrained angular entropy cost



(a) Central view (b) GT refocused (c) LF_OCC occ. (d) LF_OCC initial (e) LF_OCC final



(f) GT depth (g) GT occlusion (h) Our occlusion (i) Our initial (j) Our final

Figure 2.1: Occlusion handling and estimated depth by different methods in the *StillLife* scene from the HCI Blender [33] light field dataset. The table cloth is occluded by the wooden ball. The black pixels in (c)(g)(h) indicate there are occlusions. The occlusion map (c) generated by the LF_OCC [14] is far away from the real occlusion in (g). The proposed method produces better initial and final depth estimates, especially for preserving depth boundaries.

and the constrained adaptive defocus cost still need to implement a complex edge-preserving filter and the time-consuming graph cut algorithm to optimize the energy function [13].

As photo-consistency is only valid among unoccluded regions in the angular patch, pixels from occluders should markedly deviate from the central-view pixel. A straightforward idea to eliminate the negative effect of occlusion is to exclude occluder's pixels whose deviation from the central-view pixel is larger than a threshold. However, the photo-consistency after such exclusion is no longer selective for the correct depth, since non-consistent pixels are excluded and the intensities of the retained pixels are highly concentrated. In this chapter, we build a cost volume from a new perspective of counting the number of the retained pixels after the exclusion. We found that the number of the retained pixels whose deviation from the central-view pixel is within a threshold in the refocused angular patch can be utilized to effectively select correct depth. A disparity value achieving the largest number of retained pixels indicates the optimal selection of the correct depth. In the proposed method, every pixel in an angular patch votes to decide whether the current refocusing disparity is correct or not, and hence it is dubbed the occlusion-aware vote cost (OAVC). As shown in Fig. 2.1, without any explicit occlusion handling, the proposed OAVC is able to accurately estimate the correct depth in occlusion regions. Note that the proposed method does not directly generate Fig. 2.1 (h), which is obtained by refocusing according to the initial estimate of the

proposed method.

The threshold is usually a very small value, e.g., 0.005 for pixel intensity between 0 and 1, to ensure all irrelevant pixels are excluded. Here, the irrelevant pixels not only include pixels from occluders but also contain pixels refocused at incorrect depths. Therefore, the OAVCs are high at incorrectly refocused depths and thus the probability of erroneous depth estimates is low. The main artifact of the proposed method in the initially estimated depth is like salt-and-pepper noise, since there may be no pixels having small deviations from the central-view pixel and photo-consistency is not distinguishable in completely texture-less regions. This noise can be easily removed by a fast weighted median filter [34]. Our method does not need any further refinement like the MRF, so it also has the advantage of low computational complexity.

The rest of the chapter is organized as follows. We briefly introduce the related work in Section 2.2. In Section 2.3, a consistency analysis in the refocused angular patch is given to explain the theory behind the proposed OAVC. Section 2.4 presents the details of how to build the OAVC and discuss its properties. Section 2.5 presents and analyzes experimental results on both synthetic and real-world light field datasets to demonstrate the superiority of the proposed method in terms of depth accuracy and computational speed. We conclude this chapter in Section 2.6.

2.2 Related Work

Photo-consistency. Photo-consistency based on the Lambertian assumption [35] is that refocused pixels at the correct depth in an angular patch are consistent. Based on this assumption, two main consistency measurements are used in literature, namely correspondence and defocus. The correspondence measures the variances among pixels in an angular patch, while the defocus calculates the deviation between the central-view pixel and the other pixels [27], [31], [36], [37]. Research in [27], [28], [36] combined correspondence and defocus to obtain a more robust depth estimate. Methods based on the epipolar plane image (EPI) also utilize pixel consistency measurement to find the best slope [38], [39]. The methods based on photo-consistency work well in most regions and can usually generate reliable depth. However, photo-consistency is not valid where there is occlusion along the edges of the foreground and background. As a result, a variety of occlusion models and post-refinement methods have been proposed in an effort to improve accuracy.

Occlusion handling. Wang *et al*. [14], [30] demonstrated that the orientation of edges in the spatial domain can be used to separate an angular patch into unoccluded and occluded regions, and photo-consistency is only calculated among the pixels in the unoccluded region. The authors noted that there is an occlusion diffusion phenomenon

and they dilated the edges to tackle this problem. This occlusion model faces two problems. Firstly, accurate edge detection is not always possible, especially when there is complex background. Secondly, edge dilation cannot model occlusion diffusion very well. Zhu *et al*.[31] extended Wang's work to the scenario of multiple occuluders but their methods still face similar problems as Wang's. Chen *at al*. [32] detected partially occluded boundary regions (POBR) via superpixel-based regularization and process occlusion from the post-refinement perspective based on POBR. However, this model relies on superpixel and needs a series of refinements to generate the final depth. Handling occlusion from the perspective of data cost in [13] needs to implement the complex edge-preserving filter and the time-consuming graph cut algorithm to optimize the energy function.

Learning-based. The success of deep learning in computer vision inspires researchers to propose learning-based methods to estimate depth from light field images. Recent progress includes Epinet [40] which adopts an end-to-end fully convolutional neural network (CNN) to directly predict the depth from a stack of sub-aperture images in different directions. Heber *et al*. [41] extracted EPI patches as the input of a 5-layer convolutional network to regress the depth. Alperovich *et al*. [42] designed an encoder to learn a representation from horizontal and vertical EPIs. The representation can then be used to infer the depth by a decoder. Feng *et al*. also adopted a two-stream CNN that learns from horizontal and vertical EPIs [43]. Recently, Shi *et al*. utilized Flownet [44] in optical flow estimation to estimate the light field depth [45]. But their network needs to upsample images in the spatial domain to make them suitable for Flownet when using narrow baseline light field images. Existing learning-based methods can only estimate depth from EPI or part of sub-aperture images due to limited computation and memory resources, resulting in underuse of the full data of light field images. The lack of training data that are captured in real-world also limits the generalization ability of the networks for a variety of disparity ranges and camera parameters.

We note that Lee *et al*. [46] also utilized a voting strategy for light field depth estimation. However, the proposed OAVC is entirely different in three aspects of the theoretical hypothesis, the vote purpose, and the final depth acquisition. The theoretical hypothesis in [46] is that bundles of rays from the background are flipped on the conjugate plane [35], while our vote method assumes that unoccluded pixels should be of high consistency with the central-view pixel. The authors used disparity sign voting to separate the foreground and background in every refocused image at different depths, while we use highly consistent pixels to vote for the optimal depth estimate. Finally, Lee *et al*. [46] accumulated binary foreground and background maps to obtain the estimated depth, while we use the winner-takes-all approach to select the optimal depth estimate in the proposed OAVC.

2.3 Consistency in Refocused Angular Patch

In this section, we contribute an analysis of consistency in refocused angular patches, and find that the number of refocused pixels whose deviations from the central-view pixel are less than a small threshold is a metric that can be employed to tackle the issue of occlusion in light field depth estimation.

Denote by $L(x, y, u, v) \in \mathbb{R}^{X \times Y \times U \times V}$ a 4D light field image, where x, y are spatial coordinates and u, v are angular coordinates [26]. $X \times Y$ is the spatial resolution and $U \times V$ is the angular resolution. For instance, a light field image from the 4D Light Field Benchmark [47] has a resolution of $512 \times 512 \times 9 \times 9$. Such a light field image can also be interpreted as a grid of pinhole views, where there are 9×9 grid views and each has a spatial resolution of 512×512. The uv coordinates of the central view in any angular patch are $(0, 0)$. By setting $v = 0$ and fixing y, one can obtain a central EPI in the xu plane. As illustrated in Fig. 2.3, the slopes of the background and foreground in the EPI are $\tan\alpha$ and $\tan\beta$, respectively. Disparity d has a reciprocal relationship with slope $\tan\theta$

$$d = \frac{1}{\tan\theta}. \tag{2.1}$$

Refocusing to different possible disparities is the first step to build a cost volume for light field depth estimation. According to [26], for a 4D light field image $L(x, y, u, v)$, the light field image $L'_d(x, y, u, v)$ refocused to a candidate disparity d can be expressed as

$$L'_d(x, y, u, v) = L(x + ud, y + vd, u, v). \tag{2.2}$$

As shown in Fig. 2.2, for a fixed spatial position, angular patches are formed regarding uv coordinates after refocusing to different disparities. Pixels' color in the refocused angular patches are only consistent with the central-view pixel when refocusing at the correct disparity. Thus, photo-consistency can be used to determine the correct disparity. To measure the consistency in a refocused angular patch, we define pixel deviation $E_d(x, y, u, v)$ as the absolute difference between the refocused and the central-view pixel of the angular patch

$$E_d(x, y, u, v) = |L'_d(x, y, u, v) - L(x, y, 0, 0)|. \tag{2.3}$$

For color images, the pixel deviation is the average absolute difference over all the color channels. The mean of $E_d(x, y, u, v)$ over uv measures the consistency in an angular patch refocused to disparity d. A small value of the mean of $E_d(x, y, u, v)$ implies high consistency in the corresponding refocused angular patch.

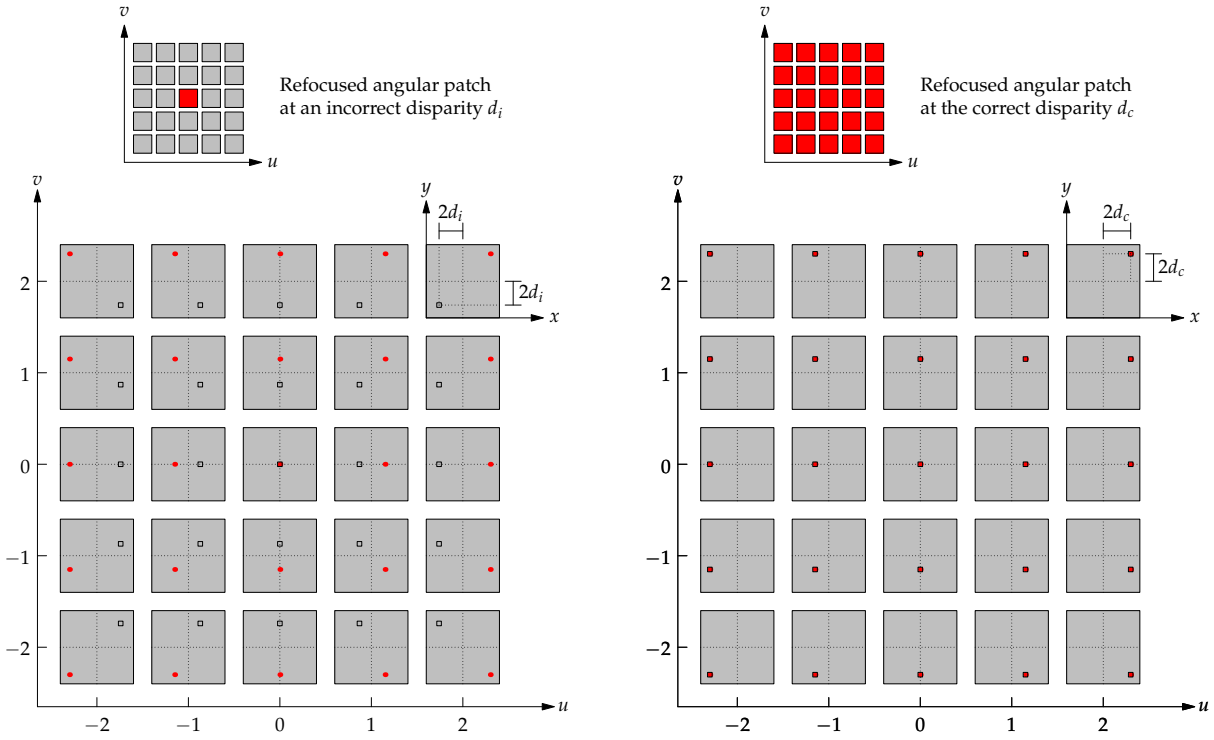


Figure 2.2: Refocused angular patches at correct and incorrect disparities. A red point with gray background in the 3D space is projected to different spatial xy coordinates in different views. In a view denoted by uv, the spatial coordinate changes are ud_c and vd_c. Refocusing at a disparity d means that we suppose the spatial coordinate changes are ud and vd. When refocusing at an incorrect disparity d_i, pixels in the refocused angular patch are from points in different depths and are inconsistent with the central-view pixel. In comparison, when refocusing at the correct disparity d_c, pixels in the refocused angular patch are from the same red point and are thus consistent in color.

2.3.1 Consistency Analysis

Photo-consistency in correctly refocused angular patches is a basic assumption for light field depth estimation. According to the Lambertian assumption, a 3D point in the space casts the same light onto microlenses or lenses in a camera array [35], resulting in consistency in a correctly refocused angular patch, which can be seen as a collector of multiple copies of the same light. Ideally, in the correctly refocused patch, all pixels will have an identical intensity if there is no occlusion. In the real world, even though consistency may not completely hold due to a weak Lambertian assumption or sub-pixel interpolation when refocusing, this consistency works very well for unoccluded regions in light field depth estimation.

Photo-consistency may also exist in angular patches refocused to incorrect disparities due to spatial consistency. Spatial consistency refers to the fact that neighboring pixels are consistent in the spatial domain. When refocusing to incorrect disparities, pixels in refocused angular patches come from neighboring pixels of correctly refocused pixels, resulting in consistency in incorrectly refocused angular patches. This

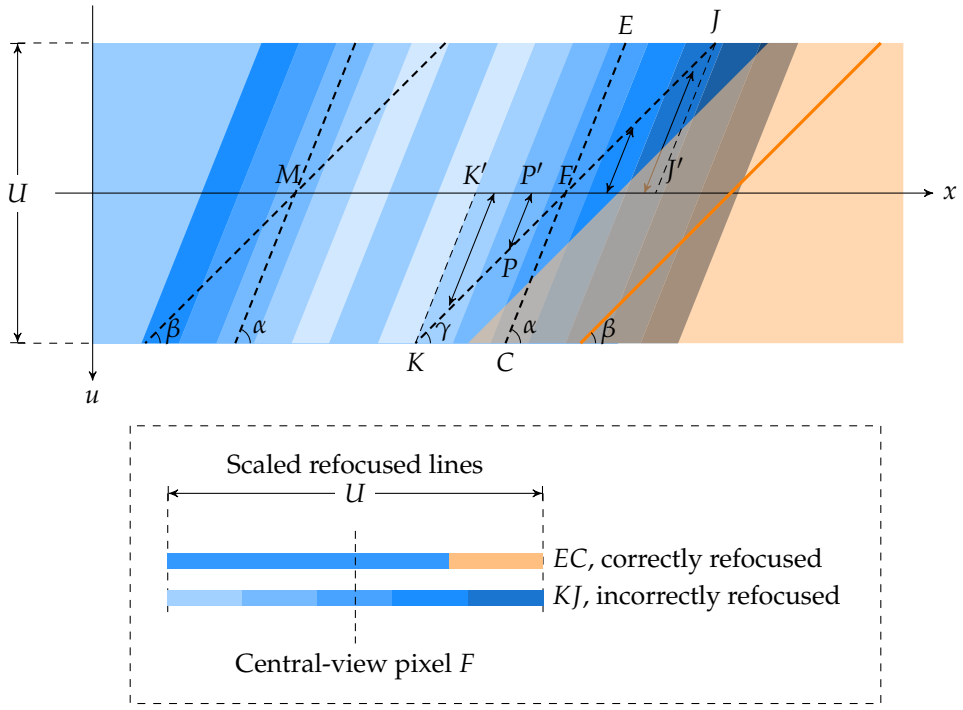


Figure 2.3: Consistency analysis. The orange part indicates the foreground which occludes the blue background. The transparency of the orange part is for better visualization. The photo-consistency in the unoccluded region (blue part in line EC) of a correctly refocused line (line EC) is stronger than that derived from the spatial consistency in an incorrectly refocused line (line KJ).

phenomenon can be clearly illustrated by the EPI shown in Fig. 2.3, where F is a central-view pixel (i.e., $u = v = 0$), and occlusion exists in the angular patch refocused to the correct disparity $1/\tan\alpha$. Pixel F shares the same disparity with its neighboring central-view pixels so that the correctly refocused lines across F and its neighboring central-view pixels share the same slope. The incorrectly refocused line KJ is actually formed by taking the pixels in the correctly refocused lines centered at F and its neighboring pixels in the central view. The pixels in line KJ are consistent with the neighboring pixels of F. For instance, pixel P in line KJ and P' in line $K'J'$ are the recorded light cast from the same 3D point. The pixels in incorrectly refocused line KJ are in turn consistent with F due to spatial consistency.

Consistency in correctly refocused patches is attributed to the Lambertian assumption, while consistency in incorrectly refocused patches stems from spatial consistency. The Lambertian consistency is stronger than spatial consistency, which enables precise depth estimation from light field images in the absence of occlusions. For instance, the consistency of the correctly refocused line cross M by angle α is higher than that of the incorrectly refocused line by angle β in Fig. 2.3. However, the consistency due to the Lambertian assumption in the whole angular patch fails, where occlusions are present. As can be observed from Fig. 2.3, line KJ is of higher consistency than line EC due

to occlusion. This is because the central-view pixel F has a consistent intensity with its neighboring pixels, and the occlusion in EC breaks the consistency derived from the Lambertian assumption. In such a situation, incorrect disparity estimation occurs, generating blurry estimates along edges.

Strong Lambertian consistency still holds in the unoccluded region of a correctly refocused angular patch. In Fig. 2.3, the consistency in the unoccluded part in line EC is higher than that of line KJ. Separating an angular patch into occluded and unoccluded regions, the Lambertian consistency in the unoccluded region will be more prominent than spatial consistency. However, as aforementioned in Section 2.1, the complex occlusion diffusion phenomenon indicates that existing occlusion models are far from being able to realistically model real occlusions.

The strong Lambertian consistency reveals a new perspective of handling the occlusion problem: separating Lambertian consistency from spatial consistency to distinguish between correct and incorrect refocusing. By inspecting high-consistency pixels in the scaled refocused lines in Fig. 2.3, it is found that the number of high-consistency pixels in a correctly refocused line is greater than that of its incorrectly refocused counterpart, even in the presence of occlusions. For instance, the length of the high-consistency (blue) segment of scaled line EC is longer than the high-consistency segment of scaled line KJ. This means that the number of pixels highly consistent with the central-view pixel in a refocused angular patch is effective in selecting the correct disparity when there is occlusion. The next problem is what level consistency can be regarded as high consistency so as to distinguish between two types of consistencies. As such, we intend to quantitatively analyze the consistency in refocused angular patches by use of the pixel deviation histogram.

2.3.2 Pixel Deviation Histogram

The high consistency among correctly refocused patches without occlusion can be demonstrated by comparing the consistency of correctly and incorrectly refocused angular patches. We exclude occlusions in correctly refocused angular patches from the comparison. We plot the pixel deviation histogram to demonstrate a threshold does exist that can well distinguish the consistency in correctly refocused angular patches from that attributable to the spatial consistency in incorrectly refocused angular patches.

The pixel deviation histogram describes the distribution of the consistency in refocused angular patches. In Fig. 2.4, correct refocusing means refocusing to the ground truth disparity. As a comparison, we calculate the second histogram over light field images refocused to incorrect disparities, with $\Delta \in \pm\{0.1, 0.2, 0.3\}$ indicating the deviation from the correct disparity. The histogram is averaged over a range of values of

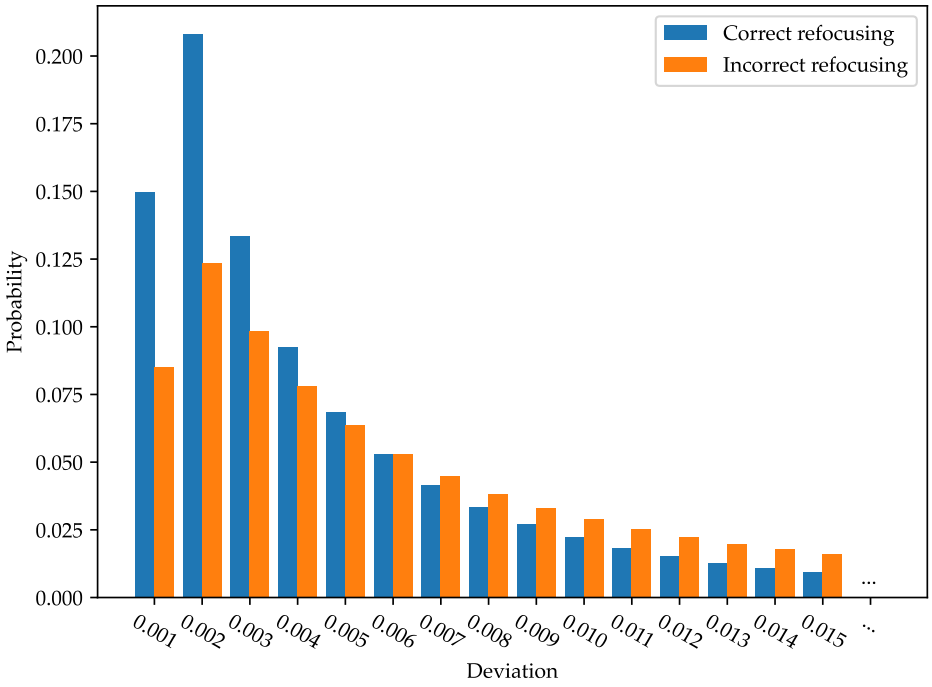


Figure 2.4: Pixel deviation histogram comparison between correctly and incorrectly refocused light field images. When the pixel deviation is less than 0.005, the pixel probability of correct refocusing is higher than that of incorrect refocusing.

Δ. The pixel intensity range is normalized to $[0, 1]$. The dataset employed is the *Additional* subset from the 4D Light Field Benchmark [47]. It contains 16 scenes with the ground truth disparity and all the scenes have the same spatial and angular resolution of $512 \times 512 \times 9 \times 9$. The histogram is calculated over the refocused light field images, and shown in Fig. 2.4, where the horizontal axis indicates the deviation of a pixel from its central-view peer in the angular patch and the vertical axis denotes the probability.

As can be observed from Fig. 2.4, an evident threshold for the pixel deviation exists, which separates the two distinct scenarios of correct and incorrect refocusing. In Fig. 2.4, the interval of the deviation bins is 0.001. Overall, the probability decreases with the pixel deviation for both scenarios. However, as can be seen from Fig. 2.4, the pixel probabilities of correct refocusing are larger than those of incorrect refocusing in deviation bins {0.001, 0.002, ..., 0.005}. In other words, this observation is valid when the pixel deviation is less than or equal to 0.005. Most pixels under correct refocusing concentrate in the low deviation bins, which means that the unoccluded regions in correctly refocused angular patches have greater consistency than when refocusing is incorrect as discussed in Section 2.3.1. The cumulative pixel probability of the deviation in refocused angular patches is shown in Table 2.1. In this table, the cumulative pixel probability of deviation 0.005 refers to the sum pixel probabilities of the deviations ranging from 0 to 0.005. A close inspection of the table reveals that over 65% of

Table 2.1: Comparison of the cumulative pixel probabilities of a variety of deviation values under correct and incorrect refocusing.

Deviation	0.001	0.002	0.003	0.004	0.005	0.006	0.007
Correct	0.150	0.358	0.491	0.584	0.653	0.705	0.747
Incorrect	0.085	0.209	0.307	0.385	0.449	0.502	0.547
Difference	0.065	0.149	0.184	0.199	0.204	0.203	0.200
Ratio	1.765	1.713	1.599	1.517	1.454	1.404	1.366

pixel deviations are less than 0.005 in the correctly refocused light field images, while this percentage is 44.9% for the case of incorrect refocusing. This means that a small threshold such as 0.005 is enough to retain most of the correctly refocused pixels in angular patches.

More importantly, the distribution statistics of the pixel deviation suggest that the refocused pixels whose deviations are larger than the threshold are the cause of incorrect depth estimation. It is observed that correct refocusing tends to lead to more concentrated cumulative pixel deviations than its incorrect counterpart. As can be seen from Table 2.1, the difference in the cumulative pixel deviation under correct and incorrect refocusing increases with the chosen threshold until reaching the maximum value of 0.204 at the deviation threshold of 0.005. After that, the difference will decrease. The reason why we use the difference to determine the threshold is that a turning point exists when using the difference, where the largest cumulative probability difference (or largest pixel number difference) can be achieved at the turning point. The largest cumulative probability difference means that the distance between the costs of correct and incorrect refocusing is maximized, which is beneficial for distinguishing these two cases. The ratio between the cumulative probabilities, however, as shown in Table 2.1, decreases monotonically and does not have such a turning point to determine the threshold. The above analysis and observation motivate us to propose the novel occlusion-aware vote cost in the next section.

2.4 Occlusion-Aware Vote Cost

In this section, we propose a novel occlusion-aware vote cost (OAVC) for light field depth estimation based on the consistency analysis given in the preceding section. Then we discuss the effect of the vote threshold and present an adaptive vote threshold method. Finally, a fast weighted median filter is used to refine the initially estimated depth.

2.4.1 Vote Cost

The pixels in a refocused angular patch are either retained or discarded, depending on their deviation relative to a preset threshold. The largest number of retained pixels indicates the optimal selection for the correct disparity. It works as though every pixel votes to decide whether this refocusing disparity is correct or incorrect. This is why it is dubbed the vote cost. The basic vote cost $C_B(d,t,x,y)$ for pixel (x,y) given the refocusing disparity d and the vote threshold t is defined as

$$C_B(d,t,x,y) = \sum_{u,v} H(E_d(x,y,u,v) - t) \tag{2.4}$$

where $H(s-t)$ is the shifted Heaviside step function with the deviation threshold t

$$H(s-t) = \begin{cases} 0, & \text{if} \quad s < t \\ 1, & \text{if} \quad s \geq t. \end{cases} \tag{2.5}$$

Here we set the vote value to 0, when the pixel deviation is smaller than the threshold t. Given a vote threshold T, the estimated disparity $D(T,x,y)$ is obtained by selecting the value of d that minimizes the basic vote cost $C_B(d,T,x,y)$ as follows

$$D(T,x,y) = \arg\min_d C_B(d,T,x,y). \tag{2.6}$$

The basic vote cost can inherently handle the occlusion problem and preserve depth edges very well, provided that the threshold is small. Fig. 2.5 (b) shows that the initially estimated depth through the use of the basic vote cost is sharp and clean along the depth boundaries. Even the depths of the small spire and flagpole on the towers are accurately estimated. This result demonstrates that the proposed basic vote cost is a very effective metric for estimating the depth in the presence of occlusion.

The problem of the basic vote cost is that it cannot distinguish the disparities that receive the same vote cost. We dub this a draw error, meaning that some possible disparities have the same number of 0 or 1 votes, as can be seen from Fig. 2.5 (b). To reduce draw errors, we add a distinguishing cost $\xi(d,t,x,y)$ to form the final vote cost $C(d,t,x,y)$

$$C(d,t,x,y) = C_B(d,t,x,y) + \xi(d,t,x,y). \tag{2.7}$$

The distinguishing cost $\xi(d,t,x,y)$ is the absolute difference between the central-view pixel and the pixels that have a smaller deviation than the threshold, which is defined as

$$\xi(d,t,x,y) = \frac{1}{Z} \sum_{u,v} E_d(x,y,u,v)(1 - H(E_d(x,y,u,v) - t)) \tag{2.8}$$

(a) Central view (b) basic vote cost (c) with distinguishing cost

Figure 2.5: Effectiveness of the basic vote cost and the distinguishing cost. The basic vote cost (b) can preserve sharp and clean edges. The addition of the distinguishing cost (c) further removes draw errors caused by an identical basic vote cost.

where $Z = UV + 1$ is a factor to ensure that the distinguishing cost is less than 1, so that it will only be used to distinguish the candidate disparities receiving the same basic vote cost, without affecting the other disparities that have different basic vote costs.

The final vote cost $C(d,t,x,y)$ can be simplified by dividing $E_d(x,y,u,v)$ into two segments according to the threshold t. Plugging (2.4) and (2.8) into (2.7) gives rise to

$$\begin{aligned} C(d,t,x,y) &= \sum_{u,v} \Big(H(E_d(x,y,u,v) - t) + \frac{1}{Z} E_d(x,y,u,v)(1 - H(E_d(x,y,u,v) - t)) \Big) \\ &= \sum_{u,v} F(E_d(x,y,u,v), t). \end{aligned} \tag{2.9}$$

When $E_d(x,y,u,v) < t$, the first term in $F(E_d(x,y,u,v),t)$ equals 0 and the second term is $1/Z E_d(x,y,u,v)$. When $E_d(x,y,u,v) \geq t$, the first term is 1 and the second term is 0. Therefore, $F(s,t)$ reduces to

$$F(s,t) = \begin{cases} \frac{1}{Z}s, & \text{if} \quad s < t \\ 1, & \text{if} \quad s \geq t. \end{cases} \tag{2.10}$$

Fig. 2.5 (c) shows the effectiveness of the distinguishing cost on reducing draw errors. A bilateral filter is then used to further refine the vote cost volume.

The proposed OAVC is very effective in yielding the correct depth estimate, especially along the depth boundaries. The small threshold excludes not only the pixels from occluders but also incorrectly refocused pixels. Most retained pixels come from the unoccluded regions in the correctly refocused angular patch. On the other hand,

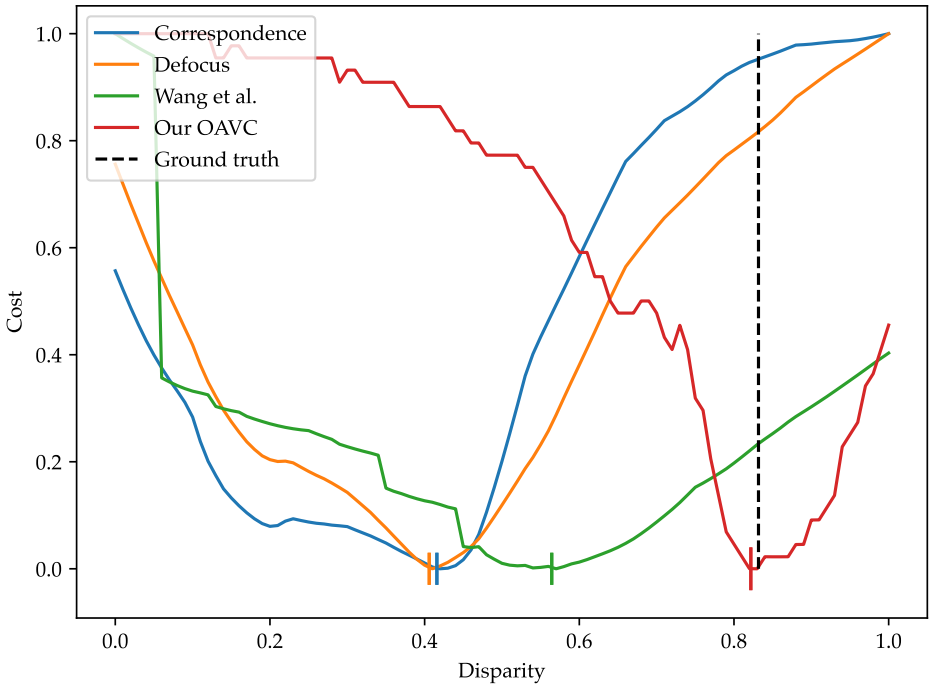
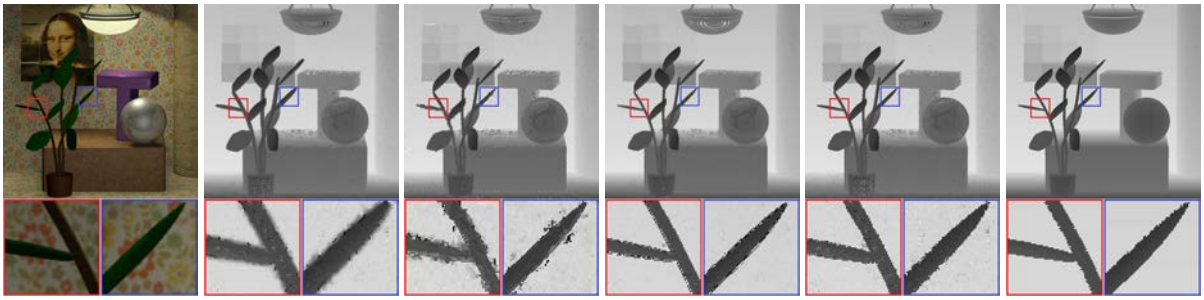


Figure 2.6: Comparison of different costs to estimate the depth of a pixel in a occlusion diffusion region. The proposed OAVC successfully produces the accurate depth estimate, while other costs fail the challenge due to ineffective occlusion handling.



(a) Central view (b) Defocus [27] (c) LF_OCC [14] (d) CAE [13] (e) OAVC (f) Ground truth

Figure 2.7: Initial depth estimates using different costs demonstrates the inherent occlusion-aware of the vote cost. The defocus cost (b) generates blurry depth estimates at the edges due to occlusion diffusion. The LF_OCC cost (c) has a lot of artifacts at the edges, while there is obvious noise along the edges in depth generated by the CAE (d). The proposed occlusion-aware vote cost (e) produces cleaner and sharper depth estimates. The reader is encouraged to zoom in for details.

in the incorrectly refocused angular patch, the small threshold is able to exclude most of the pixels, since they are neighboring pixels in the central spatial image (see Section 2.3.1). The neighboring pixels are consistent but the consistency is lower in comparison to the consistency due to highly concentrated pixels in the correctly refocused angular patch.

A comparison of different costs of a typical pixel in an occlusion diffusion region

will demonstrate the superiority of the proposed OAVC. In Fig. 2.6, the pixel is in the occlusion diffusion region near the leaf in the scene of *monasRoom* [33]. Both the cost and disparity are normalized to the range between 0 to 1. As shown in Fig. 2.6, the value of the OAVC is dominated by the basic vote cost. The distinguishing cost in the OAVC is small and only used to reduce the draw errors. The correspondence and defocus costs fail due to the occlusion. The cost with the occlusion model by Wang *et al*. [14] also fails because of the drawback of their occlusion model. In comparison, the proposed OAVC is able to successfully tackle the occlusion problem and produce a more accurate depth estimate.

The inherent occlusion-aware property of the proposed OAVC can help preserve edges in the estimated depth. We compare the initial depth estimates obtained by the proposed OAVC, the typical defocus cost in [27], the cost based upon the occlusion model in [14], and the occlusion-noise aware constrained angular entropy cost (CAE) in [13] to demonstrate the edge preservation of our method. Fig. 2.7 shows the initial depth estimates based upon these costs without any refinement. As can be observed from the figure, the proposed OAVC produces a better initial depth estimate than the other comparative costs. The defocus cost generates blurry edges caused by occlusion diffusion. The occlusion model proposed in [14] partly preserves the edges with obvious artifacts attributed to unreliable occlusion detection and approximation to occlusion diffusion. The CAE [13] yields noisy estimates along the edges. Without any explicit occlusion handling, the proposed OAVC produces cleaner and sharper depth estimates.

2.4.2 Vote Threshold

The pixel deviation histogram suggests that a vote threshold exists to separate the correctly and incorrectly refocused pixels. That threshold can be calculated providing that the ground truth depth is available. In fact, the threshold is more like a constant that is applicable to different light field images. We found a fixed threshold is appropriate for both synthetic and real-world light field images. However, adapting the vote threshold for each pixel in different light field images may improve the accuracy of depth estimation. Consequently, we propose an adaptive threshold method based on the measure of the spatial consistency.

The essential idea behind the adaptive threshold is that a small threshold should be employed to exclude incorrectly refocused pixels when they are highly consistent with the central-view pixel. In this way, we can separate the correctly and incorrectly refocused pixels to determine the optimal depth estimate. However, we cannot properly evaluate the consistency in incorrectly refocused angular patches since incorrect

disparities are unknown. As discussed in Section 2.3.1, the consistency in an incorrectly refocused angular patch is derived from the spatial consistency. Therefore, the spatial consistency defined as the average pixel deviation between a certain pixel and its neighbors in the central view provides a rough measure of the consistency in the incorrectly refocused angular patch. To exclude incorrectly refocused pixels, the pixels with deviations larger than the average pixel deviation should vote one. Formally, the proposed adaptive threshold t_a is defined as

$$t_a(x,y) = \frac{1}{UV-1}\sum_{u,v}|A_{\Delta\varepsilon}(x,y,u,v) - L(x,y,0,0)| \tag{2.11}$$

where $A_{\Delta\varepsilon}(x,y,u,v)$ represents the sampled neighboring pixels of the center-view pixel $L(x,y,0,0)$, which can be written as

$$A_{\Delta\varepsilon}(x,y,u,v) = L(x+u\Delta\varepsilon, y+v\Delta\varepsilon, 0, 0). \tag{2.12}$$

In (2.11), we set the number of sampled neighboring pixels for computing the spatial consistency to be UV, which is the same as the number of pixels in an angular patch. The sampling interval $\Delta\varepsilon$ determines the size of the local patch in the central view. Stronger consistency between a pixel and its neighboring pixels in the central spatial image leads to a smaller threshold. It should be noted that the computed spatial consistency cannot measure the consistency in incorrectly refocused angular patches, where occlusion is present. The occlusion along edges results in a large adaptive threshold which will cause difficulties in depth estimation in occlusion diffusion regions. Also, an overly small threshold may exclude most refocusing pixels including correctly refocused pixels and thus result in draw errors. To tackle these challenges, we apply maximum and minimum truncation to the above adaptive threshold

$$t_a^{\tau}(x,y) = \max(\min(t_a(x,y), \tau_{max}), \tau_{min}) \tag{2.13}$$

where $t_a^{\tau}(x,y)$ is the truncated adaptive threshold, and τ_{max} and τ_{min} are the maximum and minimum truncation values, respectively.

The truncated adaptive threshold can slightly improve depth accuracy with a small extra computational cost compared with a fixed threshold. Nevertheless, the pixel deviation histogram shown in Section 2.3.2 indicates that a vote threshold (except too small value since it will exclude most pixels) under a critical value should in general work well to distinguish correct and incorrect disparities. Adapting the threshold to each pixel provides more contextual information about the light field image in question and can bring about some performance improvement. This is an advantage of the proposed OAVC, since it does not need to carefully preset a threshold for different

light field images. A detailed comparison of the effect of the vote threshold on depth accuracy will be presented in Section 2.5.3.

2.4.3 Refinement

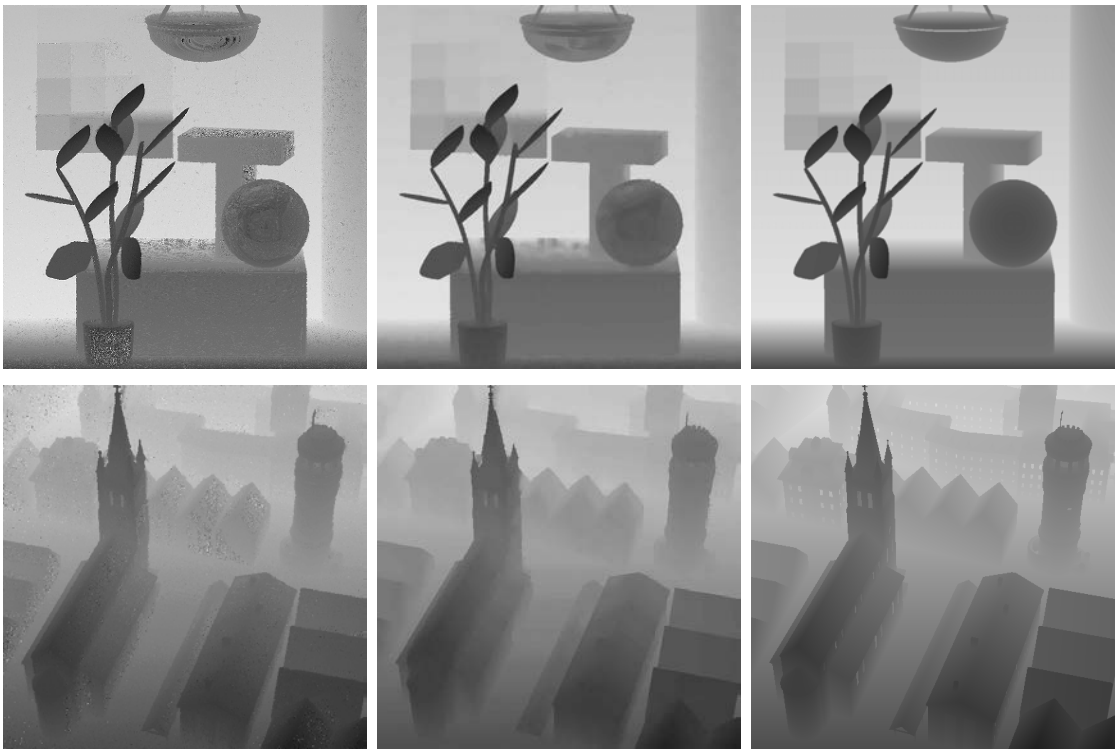
A refinement method can further smooth the estimated depth and reduce noise. The initial depth estimates using the proposed method are clean and sharp in most regions. This means that the proposed method does not need complex refinement methods like the MRF or the graph cut [12]. The main noise in the initial depth estimates is usually caused by draw errors. Most of the draw errors are removed by the distinguishing cost in the OAVC. But the distinguishing cost would not work, when pixels are in completely texture-less regions or when the chosen threshold excludes all refocused pixels.

The noise in the initial depth estimates generated by the OAVC can be easily removed by a weighted median filter (WMF). The WMF can run efficiently with a computational complexity that is linear to the kernel size [34]. Fig. 2.8 contrasts the initial depth estimates with their refined counterparts. As can be observed from the figure, the unreliable estimates are removed by the WMF, which yields a smoother and sharper final depth map. The draw errors in the flowerpot are completely filtered out, and the spire becomes more distinct.

2.5 Experimental Results

We conduct extensive experiments to show the superiority of the proposed method in terms of depth estimation accuracy and computational efficiency. Both objective and subjective comparative results will be presented. The hardware environment is Intel i7 2.4GHz CPU with 12 GB RAM. The implementation of the proposed OAVC computes one disparity cost of one sub-aperture image in every inner loop. This strategy can greatly reduce memory usage compared with the method of processing the whole light field image for every possible disparity. The datasets in our experiments include HCI Blender [33], 4D Light Field Benchmark [47], Inria Dense and Sparse [45] and real-world Stanford Lytro Light Field Archive [48].

We compare our method with recent proposed methods including occlusion-aware depth estimation (LF_OCC) [14], [30], constrained angular entropy cost (CAE) [13], spinning parallelogram operator (SPO) [49], robust pseudo random field (RPRF) [50] and partially occluded border region (POBR) [32]. Three deep learning-based methods, i.e., EPI network (Epinet) [40], flexible subset of dense and sparse (FSDS) [45] and light field attention network (LFattNet) [51], are also included in the comparisons. These



Initial Refined Ground truth

Figure 2.8: Effectiveness of the initial depth refined by the weighted median filter.

methods are chosen because they are published in reputable journals and conferences, and their source codes are available. Although there are some high-performing methods on the 4D Light Field Benchmark website, they lack implementation details and give no references. Besides, these methods seem to be learning-based and they may suffer from the generalization problem to some extent. They may not be as flexible as the proposed OAVC in dealing with large ranges of disparity values, as discussed in Section 2.5.1.

We use the mean squared error (MSE), BadPix [47], scene-specific Backgammon Fattening [47], maximum accuracy [52], and surface normal accuracy to evaluate objective depth accuracy. The performance results of the comparison algorithms except POBR and FSDS on the 4D Light Field Benchmark [47] are obtained from the benchmark website [53], where masks are applied to evaluate their performance. Since the estimated disparities of POBR and FSDS were not submitted to the 4D Light Field Benchmark [47], we use the evaluation tool provided by the benchmark to evaluate the performances of the two methods on the *Stratified (Backgammon, Pyramids, and Stripes)* and *Training (Boxes, Cotton, Dino, and Sideboard)* scenes, where the ground truth disparities of the *Stratified* and *Training* scenes are provided by the benchmark. The performances of POBR and FSDS on the *Test (Bedroom, Bicycle, Herbs, and Origami)* scenes are not available because the ground truth disparities are not made publicly available by the

benchmark. For LFattNet, the results on the *Stratified* and *Training* scenes from the 4D Light Field Benchmark [47], [53] are excluded since these scenes are used as validation data when training the LFattNet [51]. Lastly, the learning-based methods Epinet and LFattNet are not applicable to sparse light field images because the two methods are designed for dense light field images.

The angular resolution of the light field images for the experiments is 9×9, and the numbers of possible disparities are 101 for the dense light field and 201 for the sparse light field for methods based on disparity planes. We conduct experiments on both dense and sparse light field images. A dense light field means that the maximum disparity between adjacent views is usually less than 3.5 pixels, while this value is around 10 for a sparse light field. The depth accuracy is the results of an adaptive vote threshold, and the truncation values are $\tau_{max} = 0.005, \tau_{min} = 0.002$ for the dense light field and $\tau_{max} = 0.01, \tau_{min} = 0.002$ for the sparse light field. The spatial sampling interval $\Delta\varepsilon$ is set to 0.1 for both the dense and sparse light fields. $\Delta\varepsilon = 0.1$ seems a very small sampling interval to obtain ideal local consistency from the central view. However, this is enough to compute good local consistency for the adaptive threshold and we do not observe performance improvement when using larger values of $\Delta\varepsilon$ or other consistency computing methods with larger sampling areas. This is because a larger sampling area usually leads to greater local consistency that needs to be carefully mapped to a value lower than 0.005. Otherwise, the consistency will be truncated to 0.005, resulting in a fixed threshold for the pixel in question. Please see Section 2.5.3 for results using a fixed threshold.

2.5.1 Objective Comparison

As shown in Tables 2.2 and 2.3, the proposed OAVC achieves state-of-the-art depth accuracy in terms of the overall average MSE and BadPix scores on both the dense (excluding the *Dots* scene) and sparse light field datasets. Our method performs consistently well on different scenes except for the noisy *Dots* scene. The average MSE and BadPix (0.07) of the OAVC on the 4D Light Field Benchmark are not impressive but the two metrics excluding the *Dots* scene are comparably good. The limitation of the proposed OAVC in dealing with noisy scenes is discussed in Section 2.5.5. Deep learning methods perform well when the testing data shares a similar distribution with the training data, but suffer from generalization problems to some extent. For example, the most recent deep learning method LFattNet generalizes well on the Inria Dense dataset but performs badly on the HCI Blender dataset. The SPO based on the disparity planes achieves competitive results in terms of BadPix (0.07) but it runs very slow as shown in Table 2.7.

Table 2.2: MSE results of the comparison methods on the dense and sparse light field datasets (e. D. means excluding Dots, red = best, and blue = second best).

Dataset	Scene	LF_OC.	CAE	SPO	POBR	RPRF	Epinet	FSDS	LFatt.	OAVC
Dense datasets, MSE * 100										
HCI Blender	Buddha	0.91	0.64	0.54	0.52	0.28	0.36	0.40	0.33	0.36
	Buddha2	1.18	0.35	1.02	0.43	0.75	6.64	0.20	6.06	1.29
	Horse	1.36	0.79	1.37	0.46	0.50	7.35	0.74	6.32	0.53
	Medieval	1.15	0.97	0.91	0.59	0.79	2.28	0.61	0.50	0.88
	MonaRoom	0.73	0.50	0.55	0.27	0.47	1.33	0.33	0.79	0.44
	Papillon	1.00	0.63	0.66	0.59	0.66	6.12	0.46	4.98	0.84
	StillLife	4.29	1.24	1.51	3.72	1.09	2.43	0.99	14.1	1.07
	Average	1.52	0.73	0.94	0.94	0.65	3.79	0.53	4.72	0.77
4D Light Field Bench.	Backgammon	22.8	6.07	4.59	32.2	5.58	3.71	12.9	N/A	3.84
	Dots	3.19	5.08	5.24	6.65	21.2	1.48	28.4	N/A	16.6
	Pyramids	0.08	0.05	0.04	0.07	0.05	0.01	0.02	N/A	0.04
	Stripes	7.94	3.56	6.96	4.11	7.90	0.93	2.87	N/A	1.32
	Bedroom	0.53	0.23	0.21	N/A	0.26	0.20	N/A	0.37	0.21
	Bicycle	7.67	5.14	5.57	N/A	5.91	4.60	N/A	3.35	4.89
	Herbs	23.0	11.7	11.2	N/A	14.1	9.49	N/A	6.61	10.4
	Origami	2.22	1.78	2.03	N/A	1.94	1.48	N/A	1.73	1.48
	Boxes	9.59	8.42	9.11	10.9	8.55	5.97	9.53	N/A	6.99
	Cotton	1.07	1.51	1.31	4.06	0.81	0.20	0.67	N/A	0.60
	Dino	0.94	0.38	0.31	0.79	0.49	0.16	0.47	N/A	0.27
	Sideboard	2.07	0.88	1.02	1.61	1.34	0.80	1.20	N/A	1.05
	Average	6.76	3.73	3.97	7.55	5.68	2.42	7.00	3.01	3.97
	Average e. D.	7.08	3.61	3.85	7.68	4.27	2.50	3.95	3.01	2.82
Inria Dense	Flying dice	44.9	23.2	17.2	60.7	6.16	22.6	10.8	5.86	5.82
	Furniture	4.89	1.23	1.61	3.30	0.67	2.70	0.80	0.71	0.47
	Pinenuts blue	3.15	0.60	3.69	0.87	1.26	4.66	0.74	13.8	0.85
	Toy friends	1.31	0.77	2.72	0.70	0.61	1.02	0.56	4.25	1.53
	Average	13.6	6.45	6.31	16.4	2.18	7.74	3.21	6.15	2.17
Overall average		6.34	3.34	3.45	6.98	3.54	3.80	3.85	4.65	2.68
Overall average e. D.		6.49	3.26	3.37	7.08	2.74	3.90	2.48	4.65	2.05
Sparse dataset, MSE										
Inria Sparse	Electro devices	3.49	0.33	1.80	3.39	0.16	N/A	0.20	N/A	0.19
	Furniture	5.26	4.31	1.66	5.49	0.51	N/A	0.42	N/A	0.28
	Lion	3.93	0.45	0.23	2.44	0.16	N/A	0.09	N/A	0.09
	Toy bricks	5.54	0.36	1.93	5.46	0.33	N/A	0.57	N/A	0.23
	Average	4.56	1.36	1.41	4.20	0.29	N/A	0.32	N/A	0.20

Table 2.3: BadPix results of the comparison methods on the dense and sparse light field datasets (e. D. means excluding Dots, red = best, and blue = second best).

Dataset	Scene	LF_OC.	CAE	SPO	POBR	RPRF	Epinet	FSDS	LFatt.	OAVC
Dense datasets, BadPix (0.07)										
HCI Blender	Buddha	5.86	3.29	1.96	4.71	2.24	1.55	2.23	2.02	1.78
	Buddha2	14.4	7.08	10.2	10.0	8.67	34.8	3.17	34.2	11.7
	Horse	17.7	27.9	6.38	9.13	3.93	16.4	14.5	16.2	5.45
	Medieval	23.7	17.1	6.20	3.79	6.60	18.8	5.12	11.7	10.9
	MonaRoom	8.70	6.90	6.56	4.98	6.92	10.8	5.85	10.8	6.01
	Papillon	26.3	12.8	9.28	6.86	14.3	35.6	5.37	34.8	14.4
	StillLife	18.6	14.8	6.61	40.1	7.81	11.4	5.56	11.7	5.97
	Average	16.5	12.9	6.74	11.4	7.20	18.5	5.97	17.4	8.03
4D Light Field Bench.	Backgammon	13.5	3.92	3.78	19.8	3.74	3.50	20.8	N/A	3.12
	Dots	9.70	12.4	16.3	23.2	11.0	2.49	53.6	N/A	69.1
	Pyramids	1.45	1.68	0.86	0.84	0.88	0.16	0.71	N/A	0.83
	Stripes	18.3	7.87	15.0	23.0	17.2	2.46	40.3	N/A	2.90
	Bedroom	18.3	5.79	4.86	N/A	8.77	2.30	N/A	2.79	4.92
	Bicycle	19.0	11.2	10.9	N/A	12.2	9.61	N/A	9.51	12.2
	Herbs	17.7	9.55	8.26	N/A	8.67	11.0	N/A	5.22	8.73
	Origami	18.8	10.0	11.7	N/A	13.9	5.81	N/A	4.82	12.6
	Boxes	26.0	17.9	15.9	27.0	23.8	12.3	24.3	N/A	16.1
	Cotton	4.74	3.37	2.59	9.75	2.80	0.45	2.36	N/A	2.55
	Dino	15.4	4.97	2.18	6.65	5.38	1.21	4.70	N/A	3.94
	Sideboard	17.9	9.85	9.30	14.2	11.8	4.46	10.0	N/A	12.4
	Average	15.1	8.21	8.47	15.6	10.0	4.65	19.6	5.59	12.5
	Average e. D.	15.6	7.82	7.76	14.5	9.92	4.84	14.7	5.59	7.30
Inria Dense	Flying dice	41.9	37.9	9.23	54.6	24.0	31.5	20.1	7.95	7.98
	Furniture	30.3	20.8	7.07	17.6	8.29	5.05	8.61	3.70	3.53
	Pinenuts blue	7.23	24.6	11.7	6.27	3.50	19.0	2.81	2.02	2.13
	Toy friends	18.0	19.3	6.38	13.8	11.6	3.22	10.4	1.88	5.30
	Average	24.4	25.7	8.59	23.1	11.9	14.7	10.5	3.89	4.73
Overall average		17.1	12.7	7.96	15.6	9.49	10.6	12.7	10.6	9.76
Overall average e. D.		17.4	12.7	7.59	15.2	9.41	11.0	10.4	10.6	7.06
Sparse dataset, BadPix (0.3)										
Inria Sparse	Electro devices	32.3	11.3	8.57	42.2	5.91	N/A	13.4	N/A	4.87
	Furniture	46.6	22.5	10.9	66.7	3.59	N/A	14.2	N/A	3.22
	Lion	10.6	11.0	4.05	37.9	0.87	N/A	2.43	N/A	1.05
	Toy bricks	63.1	2.95	3.23	51.5	3.88	N/A	11.7	N/A	3.95
	Average	38.2	11.9	6.69	49.6	3.56	N/A	10.4	N/A	3.27

Table 2.4: Backgammon Fattening results of the comparison methods. (red = best, blue = second best).

LF_OCC	CAE	SPO	POBR	RPRF	Epinet	FSDS	OAVC
21.70	7.61	5.74	32.33	6.52	4.69	13.36	4.22

Table 2.5: Q25 results of the comparison methods (red = best and blue = second best).

Type	Dataset	LF_OCC	CAE	SPO	POBR	RPRF	Epinet	FSDS	LFattNet	OAVC
Dense	4D L. F. B.	1.86	0.66	0.89	1.70	1.09	0.31	0.80	0.14	0.90
	Inria Dense	2.43	2.33	1.23	2.62	1.10	0.57	0.56	0.16	0.90
	Overall average	2.00	1.07	0.97	2.01	1.09	0.38	0.72	0.15	0.90
Sparse	Inria Sparse	7.68	8.05	1.68	21.3	2.14	N/A	1.85	N/A	1.54

In the Inria Sparse dataset, adjacent sub-aperture images change significantly thanks to the large disparity. The problem of occlusion becomes more serious compared with the case of the dense light field, which brings about difficulty for depth estimation. As shown in Tables 2.2 and 2.3, the MSE and BadPix (0.3) of the sparse light field is much higher than that of the dense light field. The accuracy margin among different methods becomes greater compared with the case of the dense light field. Specifically, the OAVC, RPRF, and FSDS perform well on the sparse light field. The RPRF focuses more on post-refinement such that it can adapt to a variety of disparity ranges. The FSDS is based on a convolutional neural network, which is designed for optical flow estimation with large pixel motion [44]. The FSDS is naturally suited for the sparse light field, which is why it needs to upsample input images when dealing with the dense light field. The proposed OAVC, which focuses on data cost, is adaptive for various disparity ranges and achieves state-of-the-art overall depth accuracy on both the dense and sparse light field datasets.

The results of scene-specific Backgammon Fattening [47] demonstrate the superiority of the proposed OAVC in handling occlusion. The Backgammon Fattening metric measures the fraction of correctly estimated pixels in the background that are close to the foreground, which is designed to evaluate the performance of occlusion handling. It can be observed from Table 2.4 that the proposed OAVC performs better than the comparison non-learning and learning methods.

The proposed OAVC also achieves good performance in terms of maximum accuracy. The maximum accuracy is measured by Q25 (multiplied by 100) which is defined as the maximum absolute error of the best 25% disparity estimates of a scene [52], [53]. Three deep learning methods, Epinet, FSDS, and LFattNet achieve better results in terms of the overall average Q25 on the dense light field datasets as observed from Table 2.5. The overall average Q25 performance of the proposed OAVC is the best

among non-learning methods and comparable with learning-based methods on the dense light field datasets. Besides, our method performs better than all comparison methods on the sparse light field dataset.

2.5.2 Subjective Comparison

We conduct a subjective comparison of the comparative methods in an effort to demonstrate the superiority of the proposed OAVC. Fig. 2.9 compares the estimated depth on synthetic datasets. The CAE yields a good estimated depth map on the scene of *StillLife*, but it generates a lot of artifacts on the sparse scene of *Furniture*. The SPO faces a similar problem. The POBR always produces blurry edges on these scenes, especially on the sparse scene. The learning-based method Epinet produces the worst estimate on the scene *StillLife* with unexpected artifacts appearing on the pear. This indicates that learning-based methods are unreliable when there are not enough training data or the distributions of the training and testing data are different. Another learning-based method FSDS produces very smooth estimates on the sparse scene of *Furniture* but some details are lost. The RPRF generates very close results to the proposed OAVC. However, the local details of the estimated depth yielded by the proposed OAVC are better than those of the RPRF on both dense and sparse scenes.

In Fig. 2.10, we also compare the above methods on Stanford real-world light field images [48] captured by Lytro Illum. The RPRF generates comparably good results on synthetic datasets but loses too many local details on real-world light field images. The other methods face similar problems in their estimates on the synthetic datasets. Compared with existing methods, the proposed OAVC generates sharper and cleaner depth estimates on real-world light field images.

2.5.3 Effect of the Vote Threshold

Table 2.6: BadPix performance with varying vote thresholds on the dense (0.07) and sparse (0.3) light field datasets (red = best, blue = second best).

Dataset	Fixed thresholds							Adaptive
	0.001	0.002	0.003	0.004	0.005	0.006	0.007	t_a^τ
HCI Blender [33]	18.6	10.0	9.36	10.8	8.34	8.39	8.56	8.03
4D Light Field Benchmark [47]	15.2	7.86	6.70	8.06	6.41	6.74	7.28	5.99
Inria Dense [45]	15.8	5.97	4.90	5.78	5.58	6.18	6.46	4.73
Overall average	16.63	8.28	7.34	8.61	6.98	7.26	7.60	6.50
Inria Sparse [45]	6.03	4.62	4.22	4.34	3.88	3.93	4.07	3.27

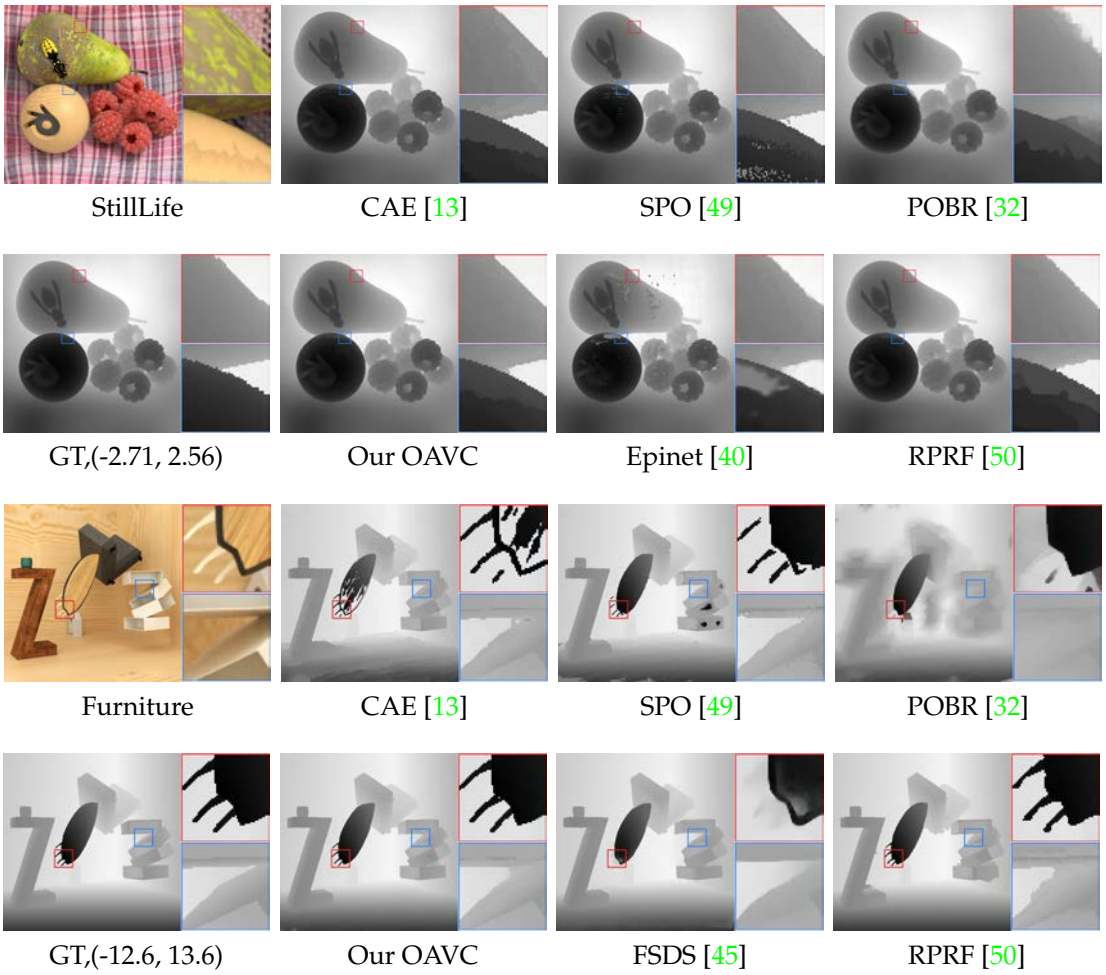


Figure 2.9: Visual comparison of estimated depth by different methods on dense and sparse synthetic light field datasets. The numbers after GT in the sub caption are the disparity ranges. The *StillLife* and *Furniture* are from HCI Blender [33] and Inria Sparse [45], respectively.

The proposed OAVC is insensitive to the vote threshold of small values. In this experiment, the truncation values and the spatial sampling interval for the adaptive threshold method are the same as introduced at the beginning of Section 2.5. Table 2.6 compares the BadPix scores of a range of fixed vote thresholds and the adaptive vote threshold on both dense and sparse light field datasets. The results on the 4D Light Field Benchmark do not include the *Test* scenes because their corresponding ground truth disparities are not publicly available, and do not include the *Dots* scene because the OAVC is not applicable to the scene with severe noise. As can be observed from the table, the fixed thresholds can generate good results. The best fixed vote threshold is 0.005 for both the dense and sparse light fields. The critical value of 0.005 we observed from the pixel deviation histogram on the *Additional* scenes from 4D Light Field Benchmark is also appropriate for other dense and sparse light field datasets. This is convenient for real-world applications, since one does not need to carefully select the

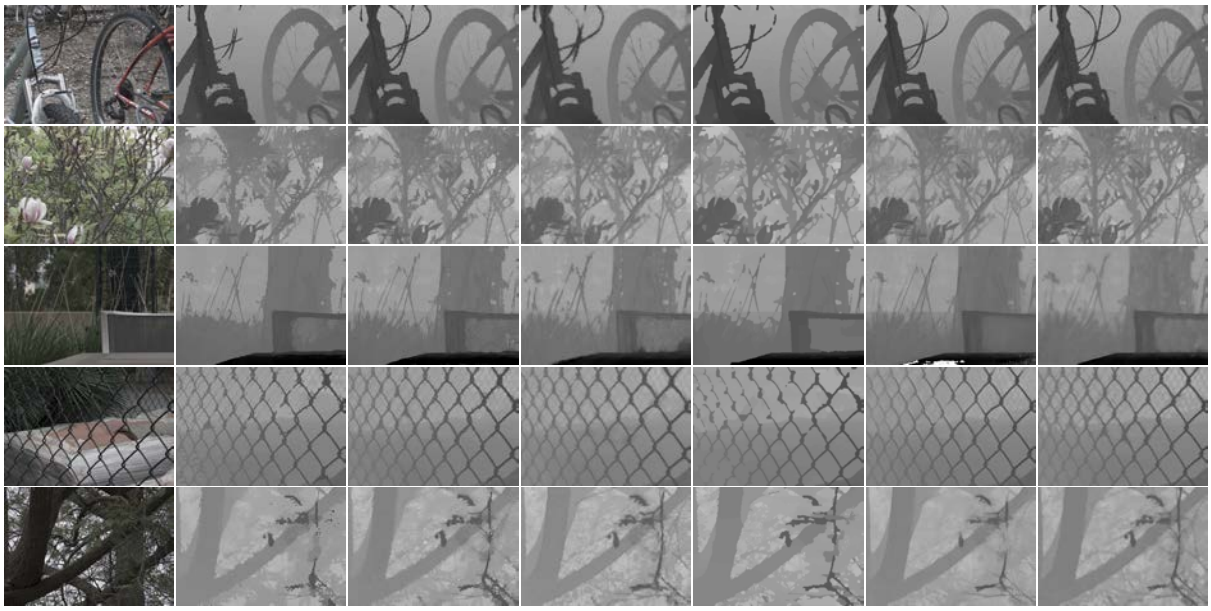


Figure 2.10: Visual comparison of estimated depth by different methods on Stanford real-world light field images [48] captured by Lytro Illum.

vote threshold.

The proposed adaptive vote threshold can improve the depth estimation accuracy for both the dense and sparse light fields. The BadPix scores decrease from 6.98 to 6.50 and 3.88 to 3.27 on the dense and sparse datasets, respectively. The perspectives of adjacent sub-aperture images vary greatly in the sparse light field. The consistency of some largely shifted pixels reduces so that the adaptive threshold with larger maximum truncation helps in distinguishing between the correctly and incorrectly refocused angular patches. In summary, an adaptive threshold is useful in improving the depth estimation accuracy with a negligible extra computational complexity.

2.5.4 Computation Time

The proposed method runs very fast thanks to the efficiency of the OAVC and the fast weighted median filter [34]. We report on the average run time per light field image to show the computational advantage of the proposed method. Since the execution time of the cost volume construction is linearly proportional to the number of possible disparities, we set this number to 101 for methods based on disparity planes. The experimental datasets include the 4D Light Field Benchmark and the Inria, where the input resolution of each light field image is $9 \times 9 \times 512 \times 512 \times 3$, and there are a total of 20 light field images. The CPU experiments are conducted using the same hardware specified at the beginning of Section 2.5. We also implement a GPU version of the

Table 2.7: Average run time in seconds per light field image (red = best and blue = second best).

	LF_OC.	CAE	SPO	POBR	RPRF	FSDS	LFatt.	Epin.	OAVC
CPU	306	780	1504	217	116	N/A	978	72	40
GPU	N/A	N/A	N/A	N/A	N/A	20	4	0.3	0.19

proposed method to compare with deep learning methods that are usually running on GPUs. The GPU implementation utilizes specialized texture memory to fetch pixels, leading to a significant speed improvement compared with its CPU counterpart. The GPU we use is the Nvidia Tesla V100 with 16 GB RAM. The reported GPU run time of our method includes the GPU run times of the initial depth estimation (0.18s) and the weighted median filter (0.01s). The GPU run time of the weighted median filter is based on the implementation in [54], which is around 0.01s for the same image size of 512×512 on the Nvidia GeForce GTX Titan X.

The average run time in seconds per light field is shown in Table 2.7. The proposed method is faster than the comparison methods on both the CPU and GPU. Since the initial estimate of the OAVC is much better than other costs like the LF_OCC or CAE, refining the initial estimate by a fast weighted median filter (less than 0.5s in the CPU in our experiment) can lead to more accurate depth estimation, as opposed to time-consuming refinement methods such as the MRF in LF_OCC and the graph cut in CAE. Thanks to hard-wired bilinear interpolation on GPU, our method is faster than three deep learning methods including FSDS, LFattNet, and Epinet on the GPU. LFattNet is much slower than the proposed method on the CPU because LFattNet entails 3D convolution. The CPU time of FSDS is not available since it relies on a GPU compiled module. Overall, the simplicity of the proposed OAVC underpins the computational performance on both the CPU and GPU platforms.

2.5.5 Limitations

One limitation of the proposed OAVC is that it does not perform well on scenes where the Lambertian assumption does not hold, as correctly refocused pixels are not strongly consistent with the central-view pixel in this situation. This may arise in scenes with severe noise or non-Lambertian reflectance. For instance, for the noisy *Dots* scene in Table 2.2, the performances of the OAVC with the adaptive threshold are 16.6 (MSE*100) and 69.1 (BadPix(0.07)), which are not impressive as opposed to the comparison algorithms. The OAVC with the fixed threshold of one degrades to the defocus cost and achieves better results (MSE 6.10 and BadPix 22.3) than those using the adaptive threshold.

Table 2.8: Surface normal accuracy of the comparison methods on the 4D Light Field Benchmark (red = best and blue = second best).

Metric	LF_OCC	CAE	SPO	POBR	RPRF	Epinet	FSDS	LFattNet	OAVC
MAE Con. Sur.	45.5	40.5	39.5	31.0	49.6	15.7	15.0	12.0	44.3
MAE Planes	33.6	34.0	30.4	19.7	40.5	18.4	5.75	8.08	37.3

The proposed OAVC is also not good at reconstructing continuous and planar surfaces. We report on the surface normal accuracy measured by the median angular error (MAE) in smooth non-planar (continuous surfaces) and planar regions. The results are directly retrieved from the 4D Light Field Benchmark website. As shown in Table 2.8, like the other traditional methods, the proposed OAVC does not perform well in terms of the MAE continuous surfaces and MAE planes on the 4D Light Field Benchmark, whereas the learning-based methods show advantages in these two measures. However, the MAEs are evaluated on continuous and planar surfaces disregarding occlusion boundaries, which is not the scenarios the proposed OAVC is designed for.

2.6 Conclusion

We proposed a novel occlusion-aware vote cost to accurately estimate depth from light field images. We analyzed the consistency in refocused angular patches and found that the consistency in unoccluded regions with correct refocusing is higher than that with incorrect refocusing. A quantitative analysis of the consistency by use of the pixel deviation histogram showed that refocused pixels with large pixel deviations (caused by occlusion or incorrect refocusing) have a negative effect on depth estimation. Based on these observations, we proposed the vote cost that separates refocused pixels by use of a threshold and utilizes the number of the separated pixels as an indicator of correct disparity estimation. A distinguishing cost was also proposed to deal with the scenario of an identical basic vote cost. Besides, we introduced an adaptive threshold method to adaptively determine the vote threshold based on local contextual information in the central spatial image. Without any explicit occlusion handling, the proposed vote cost can inherently preserve depth edges. Experimental results were presented to show that the proposed vote cost is capable of achieving the state-of-the-art performance in terms of depth estimation accuracy and computational speed.

Chapter 3

Inference-Reconstruction Variational Autoencoder for Light Field Image Reconstruction

In this chapter, we focus on local light field reconstruction aiming at synthesizing novel views within their neighboring existing views. This task requires not only light field geometry reconstruction, but also light field appearance reconstruction. An effective representation to represent the local light field is important for reconstruction quality. We obtain such representation by latent variable regularization using our proposed inference-reconstruction variational autoencoder. We also propose a viewpoint-dependent indirect view synthesis method based on adaptive convolution to effectively blend neighboring views to the target novel view.

3.1 Introduction

Light field (LF) imaging provides a novel approach to tackling traditional computer vision problems, such as depth estimation [30], [37], post-capture refocusing [12], [26], and image segmentation [12]. The wide applications of the LF have attracted a great deal of interest in both academia and industry. Much progress has been made and commercial products are now available in the market for a variety of real-world applications.

Current commercial LF cameras from Raytrix and Lytro are based on the micro-lens array. However, these micro-lens array-based cameras suffer from a trade-off between angular and spatial resolutions due to hardware limitations. Given the fixed resolution of the sensor plane, increasing the spatial resolution by adding more micro-lens will decrease the number of pixels that can be used to record the directions of light rays. Thus, the angular resolution will decrease simultaneously. To mitigate this problem, a plethora of algorithms have been proposed on super-resolution in both the spatial

and angular domains. Methods based on depth or disparity require accurate depth or disparity estimation, which is usually difficult to obtain in texture-less and occluded regions [12], [27], [55]. Learning-based methods like convolutional neural networks show better performance in spatial and angular super-resolution [55]–[59]. However, there is still great room for improvement for these methods in terms of both performance and flexibility.

The variational autoencoder (VAE) has shown to be able to synthesize photo-realistic novel views but it has not been studied in LF reconstruction. VAE-based network models usually encode reference views into a low-resolution latent distribution and then sample from the distribution to generate novel views by a decoder [60], [61]. However, the VAE tends to generate blurry images that cannot satisfy the performance requirement of LF reconstruction. Besides, VAE methods typically generate novel views by convolutional layers, which is a direct way. Research shows that indirect methods in video interpolation [62]–[65] and novel view synthesis [66] are easier to train and can obtain a better performance.

In this chapter, we propose an inference-reconstruction variational autoencoder (IR-VAE) for accurate and flexible synthesis of novel views from four corner reference views of an LF image with the goal of reconstructing a dense LF image. The flexibility here means the proposed method can synthesize novel views at arbitrary viewpoints within the four input corner views. As illustrated in Fig. 3.1, the proposed IR-VAE includes an inference network and a reconstruction network. The inference network encodes reference views and viewpoint conditions into a high-resolution latent variable and then decodes it into a target novel view. The reconstruction network encodes the reference views and the target novel view into another high-resolution latent variable and then decodes it back to the target novel view. As the input to the reconstruction network contains the target novel view, it is much easier for the reconstruction network to find the geometric information between the reference views and the target novel view. Thus, the reconstruction network has an almost ideal latent variable and can reconstruct an almost perfect novel view. In the training stage, the inference latent variable is regularized to the latent variable yielded by the reconstruction network instead of some prior distributions as in conventional VAEs. In this way, the reconstruction network can provide useful cues to the inference network in the training. Thus, the proposed IR-VAE is more effective in utilizing the geometric information between the reference views and the novel view than existing LF reconstruction networks in the training, where the novel view only serves as a training target to compute loss.

The proposed IR-VAE is similar to the well-known knowledge distillation [67] that has been widely used in the computer vision field. In knowledge distillation, a large

network (teacher) can transfer its knowledge to a small network (student) so as to improve the performance of the small network without changing its architecture. The essential difference between the proposed IR-VAE and the conventional knowledge distillation is that the reconstruction network (teacher) is much smaller than the inference network (student), because it is easy for the reconstruction network to recover the novel view when the input contains that novel view. In this chapter, one can regard the reconstruction network as the teacher network and the inference network as the student network. The reconstruction network transfers its knowledge to the inference network via latent variable regularization. The latent variable in the reconstruction network can be regarded roughly as an ideal latent variable to synthesize the target novel view as it contains the information of the target novel view. Thus, regularizing the inference latent variable to the reconstruction latent variable, instead of some prior distributions in previous work [60], can improve the quality of the inference latent variable, and in turn improve the quality of the synthesized target novel view.

We further propose a mean local maximum mean discrepancy (MLMMD) metric to measure the statistic distance between two distributions, which are with high-resolution variables to contain rich information for precise LF reconstruction. Finally, we propose a viewpoint-dependent indirect view synthesis method that predicts adaptive kernels and bias according to viewpoints and synthesizes novel views in an indirect way of adaptive convolution. The IR-VAE takes concatenated reference views as input and synthesizes arbitrary novel views between them with the condition of novel viewpoints. The major contributions of this chapter are three-fold:

The remainder of the chapter is organized as follows. We first briefly review the related work in Section 3.2. Then we describe the technical details of the proposed IR-VAE and MLMMD in Section 3.3. Section 3.4 introduces the proposed viewpoint-dependent indirect view synthesis method and network structures used to implement the IR-VAE. Section 3.5 presents experimental results and analysis. Finally, we conclude this chapter in Section 3.6.

3.2 Related work

The rich information captured by LF cameras provides more approaches to super-resolution in both the spatial and angular domains. We review the work of LF angular super-resolution, variational autoencoder, and indirect view synthesis methods which are related to this chapter.

3.2.1 Light Field Angular Super-Resolution

Depth image-based view synthesis first estimates the depth of views and then warps existing views to novel views according to the estimated depth information [12]. The view quality of depth image-based view synthesis depends highly on estimated depth maps. However, traditional matching or refocusing-based LF depth estimation methods require dense LF images to achieve a reliable performance [27], [30]. These methods do not work well when only sparse LF images are available for LF reconstruction tasks. Recent research shows that deep learning-based methods usually obtain more reliable depth maps from LF images but constructing pixel-level depth maps in real-world LF images remains an open problem.

The first deep learning-based LF view synthesis method was proposed by Kalantari *et al*. [55]. The authors extracted disparity features and they utilized a convolutional neural network (CNN) to predict disparity maps instead of selecting disparity level by a well-defined cost function [27], [30]. This method achieved better results compared to traditional depth-based methods. However, the performance of this partial learning method is implicitly limited by the extracted disparity features. It is well known that features learned from deep neural networks are generally better than hand-designed features [68], [69]. For example, Shi *et al*. utilized learning-based disparity estimation and achieved better LF reconstruction results [70]. Besides, Gul *et al*. employed a residual convolutional block attention module to refine warped images according to the estimated disparities using a CNN [71]. Jin *et al*. [72] also proposed a disparity-based method that is flexible with regards to the positions of the input views.

Mildenhall *et al.* proposed a local light field fusion (LLFF) method based on multiple plane images (MPIs) to synthesize novel views from images captured in an irregular grid pattern [16]. LLFF utilizes a 3D convolutional network to predict an MPI for each existing view. A target novel view can then be rendered by warping and blending neighboring MPIs according to the poses of the target and existing views. While LLFF achieves good view synthesis results for LF images with large disparities, it does not perform well when processing LF images with small disparities, as reported in [70]. Wang *et al*. proposed a ray transformer method that predicts colors and densities for volume rendering [17]. The method is flexible with regards to the number of input views and poses, as well as the poses of the output views. However, it runs slowly and requires a large amount of training data and computational resources to train the model.

Yoon *et al*. designed a deep CNN for realizing the spatial and angular super-resolution simultaneously [73]. They firstly up-sampled input images through a CNN and then fed the output to another CNN to synthesize novel views. However, the angular CNN reconstructs 9×9 LF images from 5×5 LF images, which means that it can

only synthesize an intermediate view between two existing views. Thus, it requires a nearly dense LF image for implementing the angular super-resolution. Some epipolar plane image (EPI)-based LF reconstruction algorithms also face a similar problem since they need enough views to form meaningful EPIs. For example, Wu *et al*. treated LF angular super-resolution as an EPI reconstruction problem and proposed a deep CNN to reconstruct a 7×7 LF image from a 3×3 LF image [74]. The model in [75] also needs 3×3 views to reconstruct the LF. In this chapter, we aim at reconstructing the LF from 2×2 sparse LF images.

Recently, 4D convolutions are popular in the area of LF image reconstruction [56], [57], [76]. Since the structure of LF images is also four-dimensional, 4D convolutions are naturally suited to extracting useful information from 4D LF images. The geometric information within the spatial and angular dimensions can be captured by 4D convolution simultaneously. The method from Wang *et al*. shows that such 4D inter-twined information can be disentangled for efficient LF image processing [77]. However, existing methods based on 4D convolutions are not flexible in synthesizing arbitrary novel views within existing reference views. In other words, the trained models can only reconstruct LF images with a fixed angular resolution. In this chapter, we show that the proposed IR-VAE based on 2D convolutions can achieve a better performance than existing 4D convolution-based methods and is also more flexible in synthesizing arbitrary novel views within reference views.

3.2.2 Variational Autoencoder

The variational autoencoder (VAE) was first proposed to learn the distribution of observed data and to generate novel views from the learned distribution [60]. Recent advances on the VAE that are related to the work in this chapter include the conditional VAE [78], [79] and InfoVAE with maximum mean discrepancy [80]. However, the VAE focuses more on generating photo-realistic novel views without the corresponding ground truth, while target novel views in LF reconstruction are deterministic. Existing viewpoint-dependent VAE methods can only synthesize novel views in low resolutions [61], [81], while LF reconstruction usually needs to perform in comparable high resolutions and more complex scenes. Generative adversarial network (GAN) [82] is also a popular model in generating images. The motivation of the proposed framework for LF image reconstruction is that properly regularizing the inference latent variable can lead to better view synthesis results. Achieving such regularization by minimizing the divergence between two distributions is straightforward in the framework of the VAE. As a comparison, implementing such regularization in the

GAN framework is not as straightforward as in the VAE. Therefore, we chose to base the proposed method for LF image reconstruction on the VAE in lieu of the GAN.

3.2.3 Indirect View Synthesis

Indirect view synthesis methods are popular in tackling the problems of video interpolation [64], [65], [83], [84] and novel view synthesis [66]. These methods share the same idea as LF reconstruction in the sense of synthesizing novel views from existing views. Indirect view synthesis uses some indirect variable, e.g., the optical flow or adaptive kernel, which contains geometric information to warp or blend existing views to novel views. For example, Jiang *et al*. used the optical flow to warp existing frames into novel frames between existing frames. Niklaus *et al*. utilized adaptive convolution to interpolate intermediate frames in a video sequence [64], [83]. This adaptive convolution shows a good performance in video interpolation, since it combines pixel warping and neighboring pixel sampling to generate more natural novel views. The challenge of directly using adaptive convolution to reconstruct the LF reconstruction lies in that it can only synthesize intermediate novel views from existing reference views. Thus Gao *et al*. leveraged adaptive convolution in a recursive manner to reconstruct LF [85], which is inefficient. In this chapter, we also use adaptive convolution to synthesize novel views indirectly, but the adaptive kernels are dependent on the novel viewpoints. Thus, existing views can be blended into arbitrary novel views to reconstruct LF images more efficiently.

3.3 Inference-Reconstruction Variational Autoencoder

In this section, we first formulate LF reconstruction as a novel view synthesis problem. Then we describe the proposed inference-reconstruction variational autoencoder in Section 3.3.2. To enable measurement of the statistic distance between two distributions in high-resolution latent space, we also propose a new statistic distance measure dubbed the mean local maximum mean discrepancy, as will be detailed in Section 3.3.3.

3.3.1 Problem Formulation

LF reconstruction can be described as a problem of synthesizing novel views according to existing reference views and viewpoints of novel views. We use the function $\mathbf{x}(x, y, u, v) \in \mathbb{R}^{X \times Y \times U \times V}$ to describe an LF image, where the coordinates x and y represent focal plane, while the coordinates u and v denote the camera plane [12]. $X \times Y$ is the spatial resolution and $U \times V$ is the angular resolution. We refer to a viewpoint

as a two-dimensional conditional vector as follows

$$\begin{aligned} \mathbf{c} &= \{(c_u, c_v) | c_u = \frac{u}{U-1}, c_v = \frac{v}{V-1}\} \\ u &= \{0, 1, ..., U-1\}, v = \{0, 1, ..., V-1\} \end{aligned} \tag{3.1}$$

where c_u and c_v are normalized angular coordinates. Given a reference sparse LF image $\mathbf{x}_r$ that includes existing reference views at four corners, we want to synthesize the novel view $\mathbf{x}_n$ for any $0 \leq c_u, c_v \leq 1$. This chapter mainly studies the problem of reconstructing a 8×8 LF image from a 2×2 LF image, which means that the reference views $\mathbf{x}_r(x, y, u_r, v_r) \in \mathbb{R}^{X \times Y \times 2 \times 2}$ are four corner views and the novel views $\mathbf{x}_n(x, y, u_n, v_n) \in \mathbb{R}^{X \times Y \times 8 \times 8}$ are located within the four corner views. LF reconstruction thus becomes a problem of inferring novel view $\mathbf{x}_n$ given the observation of $\mathbf{x}_r$ and viewpoint $\mathbf{c}$

$$(\mathbf{x}_r, \mathbf{c}) \rightarrow \mathbf{x}_n. \tag{3.2}$$

3.3.2 Framework

We model the conditional probability distribution of $\mathbf{x}_n$ as a marginal distribution of a latent variable $\mathbf{z}$

$$p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}_n | \mathbf{x}_r, \mathbf{c}) = \int p_{\boldsymbol{\theta}}(\mathbf{x}_n | \mathbf{z}) p_{\boldsymbol{\psi}}(\mathbf{z} | \mathbf{x}_r, \mathbf{c}) d\mathbf{z} \tag{3.3}$$

where $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$ are parameters of the model. Optimizing this model on training datasets through maximizing its log-likelihood is usually intractable because of the integration of the latent variable $\mathbf{z}$. Instead, we adopt the framework of the variational autoencoder with condition [78], [79], whose log-likelihood is

$$\begin{aligned} \log p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}_n | \mathbf{x}_r, \mathbf{c}) &= \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})} \left[\log \frac{p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}_n, \mathbf{z} | \mathbf{x}_r, \mathbf{c})}{q_{\boldsymbol{\phi}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c})} \right] + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})} \left[\frac{q_{\boldsymbol{\phi}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c})}{p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c})} \right] \\ &= \mathcal{L}_{\boldsymbol{\psi},\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c}) + D_{\mathrm{KL}} \left(q_{\boldsymbol{\phi}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c}) || p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c}) \right) \end{aligned} \tag{3.4}$$

where the second term is the Kullback-Leibler (KL) divergence between the proposal distribution $q_{\boldsymbol{\phi}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ and the true *posterior* distribution $p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{z} | \mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$. Since the KL divergence is non-negative, the empirical lower bound (ELBO) [60] $\mathcal{L}_{\boldsymbol{\psi},\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ defines a lower bound of $\log p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}_n | \mathbf{x}_r, \mathbf{c})$

$$\mathcal{L}_{\boldsymbol{\psi},\boldsymbol{\theta},\boldsymbol{\phi}}(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c}) \leq \log p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{x}_n | \mathbf{x}_r, \mathbf{c}). \tag{3.5}$$

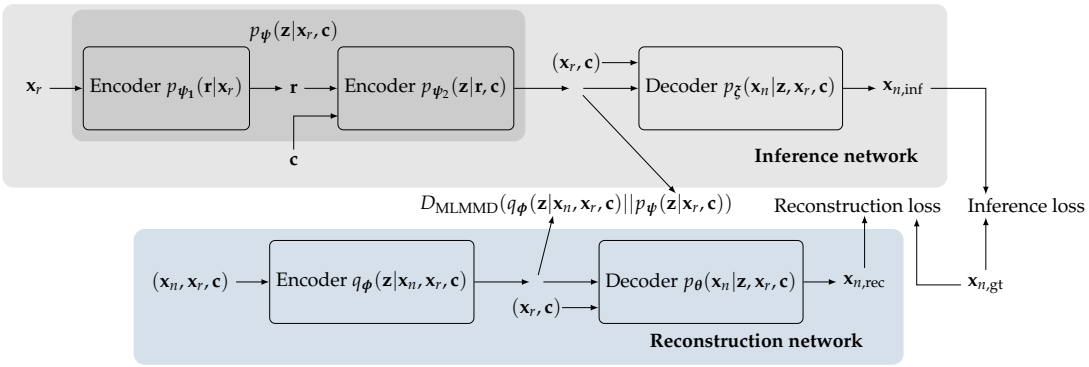


Figure 3.1: Overall framework of the proposed inference-reconstruction variational autoencoder (IR-VAE). The framework includes an inference and a reconstruction network, where the inference network infers novel view $\mathbf{x}_n$ from reference view $\mathbf{x}_r$ and viewpoint $\mathbf{c}$, while the reconstruction network reconstructs novel view $\mathbf{x}_n$ given $(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$. In the inference network, the encoder $p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ yields a latent variable that is then used to infer novel view $\mathbf{x}_{n,\text{inf}}$ by the decoder $p_{\xi}(\mathbf{z}, \mathbf{x}_r, \mathbf{c})$. In the reconstruction network, the latent variable is derived from the encoder $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$, and the decoder $p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$ decodes the latent variable into $\mathbf{x}_{n,\text{rec}}$. In the training stage, the statistic distance between $p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ and $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ is minimized by the proposed mean local maximum mean discrepancy (MLMMD) in an attempt to facilitate the information flow between $p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ and $\mathbf{x}_n$. Only the inference network is used to synthesize novel views for reconstructing LF images in the testing stage.

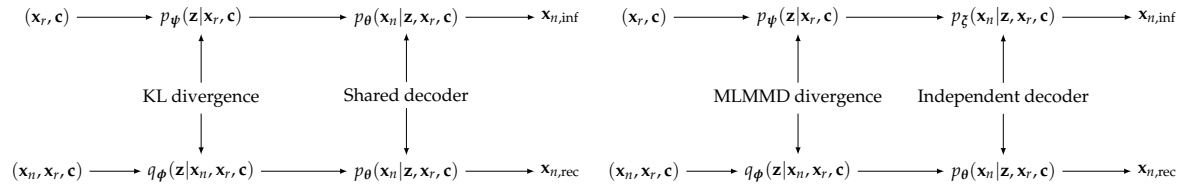
It is easier to optimize the ELBO $\mathcal{L}_{\psi,\theta,\phi}(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ than the log-likelihood $\log p_{\psi,\theta}(\mathbf{x}_n|\mathbf{x}_r, \mathbf{c})$ by rewritting the ELBO as

$$\mathcal{L}_{\psi,\theta,\phi}(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c}) = \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})}[\log p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})] - D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})||p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})). \tag{3.6}$$

Optimizing the parameters means maximizing the ELBO or minimizing the negative ELBO [78].

One problem of this framework is that the training and testing procedures are inconsistent. In the training stage, the ELBO objective function optimizes the reconstruction errors between the ground truth and the generated view by the decoder $p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$ from latent variable $\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$, while the same decoder is used to generate the novel view from latent variable $\mathbf{z} \sim p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ in the testing stage. Even though the statistic distance between $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ and $p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ is minimized by the KL divergence, the network trained by the lower bound in (3.6) is experimentally ineffective when performing inference in the testing stage. Therefore, Sohn *et al.* [79] proposed a hybrid training objective function that considers inference accuracy in the form of $\mathbb{E}_{p_{\psi}(\mathbf{z}|\mathbf{x}_r,\mathbf{c})}[\log p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})]$. By doing this, the network is able to perform consistent procedures in training and testing.

However, as shown in Fig. 3.2 (a), inferring and reconstructing novel views from



(a) Process in CVAE (b) Process in IR-VAE

Figure 3.2: Process comparison between the conditional variational autoencoder (CVAE) and the proposed IR-VAE in the context of LF reconstruction. The CVAE uses a shared decoder to infer and reconstruct the same novel view from two different latent variables. Even though the KL divergence between the two distributions is minimized, such a process in the CVAE causes interference in the decoder to output high-resolution LF views. By comparison, the proposed IR-VAE has two independent decoders that are responsible for inference and reconstruction, respectively. Consequently, the IR-VAE does not suffer from the aforementioned interference.

different latent variables using the same decoder cause interference in training. The procedure of inferring a novel view in the CVAE entails encoding $(\mathbf{x}_r, \mathbf{c})$ to a latent variable $\mathbf{z} \sim p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$, which is then decocded to the novel view by the decoder $p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$. The reconstruction follows the same procedure using the same decoder but a different encoder $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ with different input $(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$. Therefore, the latent variables used for inferring and reconstructing are different as they are generated by two distinct encoders with different inputs. The CVAE uses one decoder to decode these two different latent variables to the same novel view. This will cause interference for the network since the two latent variables are quite different. Even though the statistic distance between distributions $p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ and $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ is minimized in the training, the CVAE is shown to be ineffective for LF reconstruction in practice (see Section 3.5.4).

To eliminate the aforementioned interference, we propose to use another independent decoder $p_{\xi}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$ to generate the novel view from $\mathbf{z} \sim p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$, as shown in Fig. 3.2 (b), such that each decoder is only responsible for producing the novel view from their own input latent variables. The proposed framework is dubbed the inference-reconstruction variational autoencoder (IR-VAE), which is illustrated in Fig. 3.1. The objective function of the proposed method can be described as

$$\begin{aligned}\mathcal{L}_{\psi,\theta,\phi,\xi}(\mathbf{x}_r, \mathbf{x}_n, \mathbf{c}) =& \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})}[\log p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})] + \mathbb{E}_{p_{\psi}(\mathbf{z}|\mathbf{x}_r,\mathbf{c})}[\log p_{\xi}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})] \\ & - D_{\mathrm{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})||p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})).\end{aligned} \tag{3.7}$$

The proposed framework consists of two distinct networks, each of which includes its own encoder and decoder. Since the network composed of encoder $p_{\psi}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ and decoder $p_{\xi}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$ aims at inferring $\mathbf{x}_n$ from $(\mathbf{x}_r, \mathbf{c})$, we refer to it as the *inference network*. The network composed of encoder $q_{\phi}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ and decoder $p_{\theta}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$

is termed the *reconstruction network*, which first encodes $(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ to a latent variable $\mathbf{z}$, and then reconstructs $\mathbf{x}_n$ from that latent variable. The two networks are both used to train the network, while only the inference network is resorted to in the testing stage. Besides, the KL divergence $D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})||p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c}))$ is minimized to make $p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ get close to $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ because $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ contains the information of $\mathbf{x}_n$ which helps accurate reconstruction and inference of $\mathbf{x}_n$.

One intuitive approach to deriving (3.7) is to consider the KL divergence as a regularization term. Instead of regularizing the distribution $p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ to approach some prior distributions (typically Gaussian) in VAEs, the KL divergence in (3.7) regularizes $p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ to be close to $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ as much as possible. Given the strong learning ability of deep neural networks, the latent variable $\mathbf{z} \sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ can be regarded as an ideal representation to generate $\mathbf{x}_n$. Regularizing the conditional distribution $p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ to $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ is thus helpful for inferring $\mathbf{x}_n$ from $(\mathbf{x}_r, \mathbf{c})$ in the inference network.

3.3.3 Mean Local Maximum Mean Discrepancy

It has been found that the training ELBO objective in the VAE can cause inaccurate amortized inference distributions and the model may ignore latent variables to fit data distributions [80]. The first problem means that the inferred distribution $p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$ does not approximate the true *posterior* distribution $p_{\boldsymbol{\psi},\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$ very well, and the second problem implies that the model fails to learn meaningful latent representations. To solve the above problems, Zhao *et al*. [80] proposed a variant of the VAE dubbed InfoVAE which includes a mutual information maximization term in the objective. The authors then convert the new objective into a more tractable form and obtain a model called MMD-VAE using the maximum-mean discrepancy (MMD) [86], [87] as the divergence to measure the statistic distance between two distributions. We thus adopt the theory of MMD-VAE and introduce weights to each term. Our new objective is

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi},\boldsymbol{\psi},\boldsymbol{\xi}}(\mathbf{x}_n, \mathbf{x}_r, \mathbf{c}) =& \alpha \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})] \\ &+ \beta \mathbb{E}_{p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r,\mathbf{c})}[\log p_{\boldsymbol{\xi}}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})] \\ &- \gamma D_{\mathrm{MMD}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})||p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})).\end{aligned} \tag{3.8}$$

where α, β and γ are weights to adjust the importance of the corresponding terms. D_{MMD} is the MMD divergence between distributions $p(\mathbf{z})$ and $q(\mathbf{z})$

$$D_{\mathrm{MMD}}(p(\mathbf{z})||q(\mathbf{z})) = \mathbb{E}_{p(\mathbf{z}),p(\mathbf{z}')}[k(\mathbf{z}, \mathbf{z}')] - 2\mathbb{E}_{p(\mathbf{z}),q(\mathbf{z}')}[k(\mathbf{z}, \mathbf{z}')] + \mathbb{E}_{q(\mathbf{z}),q(\mathbf{z}')}[k(\mathbf{z}, \mathbf{z}')] \tag{3.9}$$

where $k(\cdot,\cdot)$ is a positive definite kernel function. D_{MMD} equals 0 if and only if $p(\mathbf{z}) = q(\mathbf{z})$.

The computational complexity of the MMD divergence is the quadratic time of the resolution of the latent variables. This complexity is not acceptable when the resolution of latent variables is large. For example, we may utilize the latent variable $\mathbf{z} \in \mathbb{R}^{H\times W\times C}$ where H, W, and C are the height, width, and channel. Maintaining a high-resolution representation is shown to be more accurate for position-sensitive tasks in [88]. In such a situation, the latent variables have the same H and W as those of input images. The computational complexity of the MMD on this latent variable is $O((HWC)^2)$, which is significant. For LF reconstruction, a predicted pixel in $\mathbf{x}_n$ locally depends on the corresponding part in $\mathbf{z}$. So, we propose the mean local MMD (MLMMD) method to measure the statistical distance between distributions with high-resolution latent variables to reduce the computational complexity. The calculation of the MLMMD is limited to local patches of latent variables

$$\begin{aligned} D_{\text{MLMMD}}(p(\mathbf{z})||q(\mathbf{z})) = \mathbb{E}_{\mathbf{z}_l\in\mathbf{z},\mathbf{z}'_l\in\mathbf{z}'}\Big[&\mathbb{E}_{p(\mathbf{z}_l),p(\mathbf{z}'_l)}[k(\mathbf{z}_l,\mathbf{z}'_l)] \\ &-2\mathbb{E}_{p(\mathbf{z}_l),q(\mathbf{z}'_l)}[k(\mathbf{z}_l,\mathbf{z}'_l)] + \mathbb{E}_{q(\mathbf{z}_l),q(\mathbf{z}'_l)}[k(\mathbf{z}_l,\mathbf{z}'_l)]\Big] \end{aligned} \tag{3.10}$$

where $\mathbf{z}_l, \mathbf{z}'_l \in \mathbb{R}^{D\times E\times C}$ are local patches of latent variables $\mathbf{z}, \mathbf{z}'$. The values of D, E are usually far less than those of H, W: $D \ll H, E \ll W$, i.e., $D = E = 8$. The computational complexity of the proposed MLMMD is $O(\frac{H}{D}\frac{W}{E}(DEC)^2)$, which allows high-resolution latent variables. More specifically, the MLMMD is computed in practice as follows

$$\begin{aligned} D_{\text{MLMMD}}(p(\mathbf{z})||q(\mathbf{z})) = &\frac{DE}{HW}\sum_{\mathbf{a}_l\in\mathbf{a},\mathbf{b}_l\in\mathbf{b}}\Big[\frac{1}{M^2}\sum_{i=1}^{M}\sum_{j=1}^{M}k(\mathbf{a}_l^i,\mathbf{a}_l^j) \\ &-\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}k(\mathbf{a}_l^i,\mathbf{b}_l^j) + \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}k(\mathbf{b}_l^i,\mathbf{b}_l^i)\Big] \end{aligned} \tag{3.11}$$

where $\{\mathbf{a}^i\}_{i=1}^M \sim p(\mathbf{z})$ and $\{\mathbf{b}^i\}_{i=1}^N \sim q(\mathbf{z})$ are two sets of samples, and $\mathbf{a}_l, \mathbf{b}_l$ are local patches of samples $\mathbf{a}, \mathbf{b}$. The kernel function $k(\cdot,\cdot)$ is chosen to be Gaussian in this work.

Based on the above descriptions, we provide the training loss of the proposed method. The expectation in (3.8) can be approximated by Monte Carlo (MC) sampling, which leads to a loss of mean square error (MSE). However, training the network by using the MSE tends to generate over-smoothed images, i.e. blurry synthesized views.

Instead, we replace the MSE with the mean absolute error (MAE), which is defined as

$$L_a(\mathbf{x}_n, \mathbf{x}_{n,\text{gt}}) = \frac{1}{M}\sum_{i=1}^{M}|\mathbf{x}_n^i - \mathbf{x}_{n,\text{gt}}^i| \tag{3.12}$$

where M is the batch size in training and $\mathbf{x}_{n,\text{gt}}$ is the ground truth view of the LF image. The MAE measures the pixel error between the synthesized view and the ground truth. Also, we introduce the following perceptual loss to reduce structure dissimilarity

$$L_p(\mathbf{x}_n, \mathbf{x}_{n,\text{gt}}) = \frac{1}{M}\sum_{i=1}^{M}|f(\mathbf{x}_n^i) - f(\mathbf{x}_{n,\text{gt}}^i)| \tag{3.13}$$

where function $f(\cdot)$ maps the input to *conv4_3* features of the ImageNet pre-trained VGG16 model [89]. The perceptual loss tends to preserve high-quality sharp details and encourages the network to produce visually pleasing results because features extracted from the pre-trained model contain information of real-world images [64], [90], [91]. According to (3.8), the overall loss is a weighted combination of the inference loss, reconstruction loss, and the MLMMD

$$\begin{aligned} L =& \alpha\big(L_a(\mathbf{x}_{n,\text{inf}}, \mathbf{x}_{n,\text{gt}}) + \eta L_p(\mathbf{x}_{n,\text{inf}}, \mathbf{x}_{n,\text{gt}})\big) \\ &+ \beta\big(L_a(\mathbf{x}_{n,\text{rec}}, \mathbf{x}_{n,\text{gt}}) + \frac{\eta}{\beta}L_p(\mathbf{x}_{n,\text{rec}}, \mathbf{x}_{n,\text{gt}})\big) \\ &+ \gamma D_{\text{MLMMD}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})||p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})) \end{aligned} \tag{3.14}$$

where η is a weight to balance between the MAE and the perceptual loss. In this chapter, pixel intensities are normalized to the range of 0 and 1, L_a is small relative to L_p so we set η to 0.01. Since we are mainly concerned with the performance of inferring novel views, $\alpha = 1$ and $\beta = 0.1$ are chosen to focus on optimizing the inference network. η/β is used to avoid too small final weight for $L_p(\mathbf{x}_{n,\text{rec}}, \mathbf{x}_{n,\text{gt}})$. Finally, we use $\gamma = 10$ to enlarge the MLMMD because it is a small value in the $\mathbf{z}$ space. These loss weights are general for various problem settings as their ratios do not change. For example, the reconstruction loss will always converge quickly to a small value as the input of the reconstruction network contains the target novel view. The MAE and perceptual losses simultaneously decrease during training. In this chapter, we use the same loss weights for experiments with both small and large disparities.

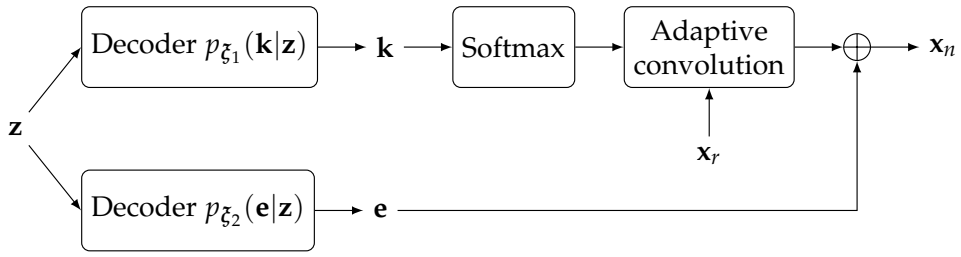


Figure 3.3: Schematic of the indirect view synthesis method. The decoder $p_\xi(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$ includes decoder $p_{\xi_1}(\mathbf{k}|\mathbf{z})$ and decoder $p_{\xi_2}(\mathbf{e}|\mathbf{z})$. The former produces the adaptive convolution kernel $\mathbf{k}$, while the latter yields the compensation bias $\mathbf{e}$, respectively. $\mathbf{k}$ is normalized by the softmax function. Adaptive convolution is then performed and added to compensation bias e to form the final synthesized novel view $\mathbf{x}_n$.

3.4 Encoder and Decoder Structures

In this section, we first present the proposed viewpoint-dependent indirect view synthesis method based on adaptive convolution. Then we describe the network structures of the encoders and decoders. The training and testing procedures are also given for a better understanding of the proposed IR-VAE.

3.4.1 Viewpoint-dependent Indirect View Synthesis Method

Novel view synthesis based on the VAE usually outputs the target view through the last convolutional layer [61], [81]. This way is not effective for LF reconstruction since the geometric relationship between the reference views and target novel views is easier to infer than photo-realistic novel views. But this relationship is not explicitly utilized in the framework when directly outputting novel views from latent variables via convolutional layers. Recent research on view synthesis [66] and video interpolation [62], [65] has shown that indirect novel view synthesis which explicitly utilizes the geometric relationship (in the form of the appearance flow [66] or optical flow [62], [65]) results in faster training and better performance. Therefore, we propose a viewpoint-dependent indirect view synthesis method cooperating with the IR-VAE framework to explicitly utilize the geometric relationship.

The indirect method $g(\cdot)$ takes reference views $\mathbf{x}_r$ and the indirect variable $\mathbf{w}$ containing the geometric relationship as inputs, and outputs a novel view as follows

$$\mathbf{x}_n = g(\mathbf{x}_r, \mathbf{w}). \tag{3.15}$$

where $\mathbf{w}$ is obtained by a decoder $p_\xi(\mathbf{w}|\mathbf{z})$. The function $g(\cdot)$ should be differentiable with respect to $\mathbf{w}$ such that gradients can be backpropagated to train the network. In

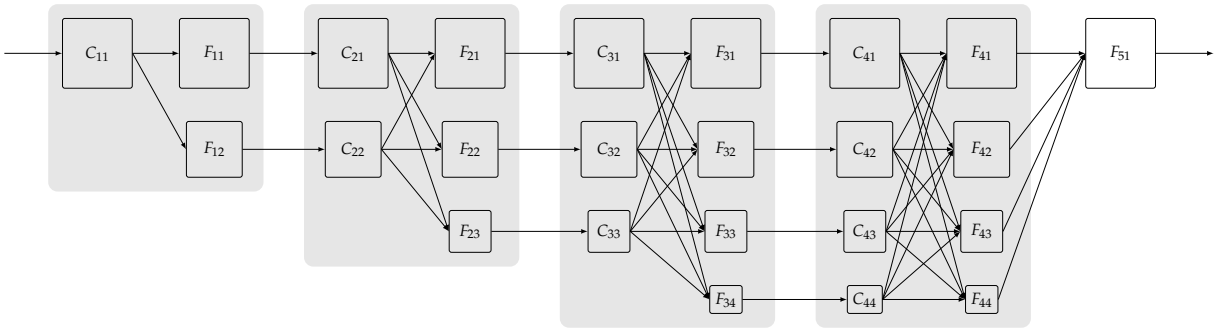


Figure 3.4: Structure of encoder $p_{\psi_1}(\mathbf{r}|\mathbf{x}_r)$. The structure is based on the high-resolution network (HRNet) [88] which maintains high-resolution representations through the whole process. C denote convolutional block, each of which contains several convolution + ReLU layers and F is the fusion layer which fuses representations from differents scales. C_{*1}, C_{*2}, C_{*3} and C_{*4} process their inputs at $1, 1/2, 1/4$ and $1/8$ scales, respectively.

other words, the following gradients should be tractably computed

$$\nabla_{\xi}\left[p_{\xi}(\mathbf{x}_n|\mathbf{z},\mathbf{x}_r,\mathbf{c})\right] = \frac{\partial L}{\partial g(\mathbf{x}_r,\mathbf{w})}\frac{\partial g(\mathbf{x}_r,\mathbf{w})}{\partial \mathbf{w}}\nabla_{\xi}\left[p_{\xi}(\mathbf{w}|\mathbf{z})\right]. \tag{3.16}$$

The decoder $p_{\xi}(\mathbf{w}|\mathbf{z})$ is usually implemented by a deep neural network and $\boldsymbol{\xi}$ are the parameters of the network. Thus, the gradients $\nabla_{\xi}\left[p_{\xi}(\mathbf{w}|\mathbf{z})\right]$ can be easily computed.

The remaining issue is how to choose the indirect function $g(\cdot)$. One choice is to treat $\mathbf{w}$ as the disparity and $g(\cdot)$ is such a warping function that warps $\mathbf{x}_r$ into $\mathbf{x}_n$ according to disparity $\mathbf{w}$. However, this warping method involves bilinear sampling that considers only four neighboring pixels. Also, it needs extra mask maps to handle occlusion to blend multiple warped views into the target novel view. Instead, we use adaptive convolution [64] as function $g(\cdot)$. The adaptive convolution combines the operations of warping pixels and sampling neighboring pixels into a single operation. An adaptive convolution layer is different from a conventional convolutional layer. In an adaptive convolutional layer, each element is assigned a unique kernel, while kernels are shared for all elements in a conventional convolutional layer. Adaptive convolution samples $M \times M$ pixels, where M is the kernel size. Therefore, adaptive convolution can utilize more information to produce more realistic views than bilinear interpolation. Besides, the proposed IR-VAE can learn to yield very small kernel weights for occluded pixels, enabling occlusion handling when using adaptive convolution.

Existing adaptive convolution methods for view synthesis take only two input views and synthesize intermediate views. Synthesizing dense LF views by use of adaptive convolution methods involves running the algorithm in a recursive manner [85]. Instead, in this chapter, the adaptive convolutional kernel $\mathbf{k}$ is dependent on the representation $\mathbf{r}$ and the novel viewpoints $\mathbf{c}$ as represented by $p_{\xi}(\mathbf{k}|\mathbf{z}), \mathbf{z} \sim p_{\psi}(\mathbf{z}|\mathbf{r},\mathbf{c})$. The

decoder $p_\xi(\mathbf{k}|\mathbf{z})$ is modelled by a CNN. The network can thus learn to generate necessary kernels to predict high-quality novel views according to $\mathbf{r}$ and $\mathbf{c}$. The adaptive kernel $\mathbf{k}$ is not only adaptive to different pixels, but also to different novel viewpoints. Thus, the proposed method is dubbed the viewpoint-dependent indirect view synthesis method.

The result of adaptive convolution may not be good enough due to inaccurate adaptive kernels and the limitation of indirect view synthesis. The limitation implies that target novel pixels are not always reproducible from reference views due to several factors, e.g., non-Lambertian reflection. We thus introduce a bias map to compensate for errors in the result of adaptive convolution as the way in [92]. The indirect variable $\mathbf{w}$ thus includes adaptive kernel $\mathbf{k}$ and bias $\mathbf{e}$. Fig. 3.3 describes the schematic of the viewpoint-dependent indirect view synthesis method. Decoders $p_{\xi_1}(\mathbf{k}|\mathbf{z})$ and $p_{\xi_2}(\mathbf{e}|\mathbf{z})$ produce adaptive kernel $\mathbf{k} \in \mathbb{R}^{X\times Y\times U_r\times V_r\times M\times M}$ and bias $\mathbf{e} \in \mathbb{R}^{X\times Y}$ from the conditional latent variable $\mathbf{z} \sim p_\psi(\mathbf{z}|\mathbf{r},\mathbf{c})$, where X, Y are the spatial height and width, while U_r, V_r are the angular height and width of the reference sparse LF image $\mathbf{x}_r$. As in [93], we first normalize adaptive kernel by softmax function

$$\hat{\mathbf{k}}(x,y,u_r,v_r,i,j) = \frac{\exp(\mathbf{k}(x,y,u_r,v_r,i,j))}{\sum_{u_r}^{U_r}\sum_{v_r}^{V_r}\sum_{i}^{M}\sum_{j}^{M}\exp(\mathbf{k}(x,y,u_r,v_r,i,j))}. \tag{3.17}$$

One example is reconstructing a 8×8 LF image from a 2×2 LF image. The kernel $\mathbf{k}$ is of resolution $\mathbb{R}^{X\times Y\times 2\times 2\times 15\times 15}$ with a kernel size of $M = 15$ for each view in the viewpoints within 8×8. The kernel size of M is chosen according to the maximum absolute disparity d_{max} (as both negative and positive disparities exist) between top-left and top-right reference views

$$M = 2d_{max} + 1. \tag{3.18}$$

A synthesized novel view at viewpoint (u_n, v_n) is obtained from the following adaptive convolution with bias

$$\begin{aligned} \mathbf{x}_n &= g(\mathbf{x}_r, \mathbf{k}, \mathbf{e}) \\ \mathbf{x}_n(x,y,u_n,v_n) &= \sum_{u_r}^{U_r}\sum_{v_r}^{V_r}\sum_{i}^{M}\sum_{j}^{M} a(\mathbf{x}_r,\hat{\mathbf{k}}) + \mathbf{e}(x,y) \\ a(\mathbf{x}_r,\hat{\mathbf{k}}) &= \mathbf{x}_r(x+i-\frac{M}{2}, y+j-\frac{M}{2}, u_r, v_r)\hat{\mathbf{k}}(x,y,u_r,v_r,i,j). \end{aligned} \tag{3.19}$$

For a color LF image, e.g., with RGB channels, an adaptive kernel for a certain pixel is shared in all channels. But a bias has the same number of channels as its corresponding pixel. As can be observed from (3.19), the adaptive convolution samples

Table 3.1: Summary of properties of the encoders and decoders.

Networks	Input		Output		Implementation
	Ch.	Description	Ch.	Description	
$p_{\psi_1}(\mathbf{r}\|\mathbf{x}_r)$	12	Concatenation of 4 corner reference images, each image with 3 channels (ch.)	64	Representation	HRNet [88]
$p_{\psi_2}(\mathbf{z}\|\mathbf{r},\mathbf{c})$	66	Concatenation of representation (64 ch.) and viewpoint condition (2 ch.)	64	Latent variable	RDB [94]
$p_{\xi_1}(\mathbf{k}\|\mathbf{z})$	64	Latent variable $\mathbf{z} \sim p_{\psi}(\mathbf{z}\|\mathbf{x}_r,\mathbf{c})$	15×15×4	Adaptive kernel	RDB [94]
$p_{\xi_2}(\mathbf{e}\|\mathbf{z})$	64	Latent variable $\mathbf{z} \sim p_{\psi}(\mathbf{z}\|\mathbf{x}_r,\mathbf{c})$	3	Adaptive bias	RDB [94]
None	17	Concatenation of 4 corner reference images, 1 target novel image, and viewpoint condition	64	Latent variable	RDB [94]
$p_{\theta_1}(\mathbf{k}\|\mathbf{z})$	64	Latent variable $\mathbf{z} \sim q_{\phi}(\mathbf{z}\|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})$	15×15×4	Adaptive kernel	RDB [94]
$p_{\theta_2}(\mathbf{e}\|\mathbf{z})$	64	Latent variable $\mathbf{z} \sim q_{\phi}(\mathbf{z}\|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})$	3	Adaptive bias	RDB [94]

many pixels from all reference views to produce a novel pixel. The maximum weight in an adaptive kernel indicates that its corresponding pixel is the most important pixel for yielding the novel pixel. Also, the position of the maximum weight relative to the adaptive convolution center implies the disparity information between the reference and novel pixels. The weights near the maximum weight are used to sample corresponding pixels to produce a reliable novel pixel. Adaptive kernels and bias are predicted from a CNN with the latent variable $\mathbf{z}$ as input, where $\mathbf{z}$ contains the information of reference views and the viewpoint of the novel view. Thus, the adaptive kernels and bias will adapt to the given viewpoint. If a reference pixel is occluded in the novel view, the IR-VAE will produce a very small weight for that pixel to alleviate the occlusion problem.

3.4.2 Encoder and Decoder Structures

In the proposed method, encoder $p_{\psi_1}(\mathbf{r}|\mathbf{x}_r)$ is the key network that is responsible for extracting a representation from reference views in LF images. This network is capable of capturing geometric information from reference views at multiple viewpoints. Coordinates of pixels recording light rays change in both the horizontal and vertical directions. It requires a powerful network to accurately estimate and represent the

complex geometric relationship within LF images. Typical networks for this task usually encode inputs into a low-resolution representation and recover a high-resolution representation from the low-resolution one by upsampling [65], [83], [95]. However, as pointed out in [88], such networks tend to lose details for position-sensitive tasks. Therefore, we utilize the high-resolution network (HRNet) [88] to capture geometric information and represent it as a high-resolution representation.

As shown in Fig. 3.4, the HRNet performs multi-scale convolutions in parallel and always maintains high-resolution representations throughout the full process. In this way, the HRNet is able to capture a richer and more precise high-resolution representation for LF reconstruction, where local details are important. The high-resolution representation $\mathbf{r}$ in conjunction with target novel viewpoint condition $\mathbf{c}$ is further encoded into the latent variable $\mathbf{z}$ by encoder $p_{\boldsymbol{\psi}_2}(\mathbf{z}|\mathbf{r},\mathbf{c})$. The latent variable is also a high-resolution representation. In the framework of the original MMD-VAE, the high-resolution latent variable is not applicable because of the difficulty of computing the MMD among high-resolution latent variables. The proposed MLMMD in Section 3.3.3 solves this problem by restricting the computation of the MMD locally, which enables the proposed IR-VAE framework to work with high-resolution latent variables and results in precise LF reconstruction.

The other encoders and decoders in both the inference and reconstruction networks are implemented by the residual dense block (RDB) [94], [96]. The RDB was first proposed in [94] for image denoising. The RDB fuses hierarchical features from all convolutional layers by dense connections and local feature fusion, which makes better use of features from all layers in the RDB. We apply a convolutional layer followed by the ReLU activation before the RDB to map the input into a feature with an acceptable channel. The same layer is also adopted after the RDB to map the output channel into the target channel, which results in the following network

$$\rightarrow Conv + ReLU \rightarrow RDB \rightarrow Conv + ReLU \rightarrow . \tag{3.20}$$

In this chapter, the number of input channels of the RDB is 64 and there are 6 convolutional layers in the RDB. This network is used to implement all encoders and decoders in Table 3.1 except encoder $p_{\boldsymbol{\psi}_1}(\mathbf{r}|\mathbf{x}_r)$.

To better understand the encoder and decoder structures, we summarize the properties of these encoders and decoders in Table 3.1, where 2×2 reference views are assumed. The input to encoder $p_{\boldsymbol{\psi}_1}(\mathbf{r}|\mathbf{x}_r)$ is the concatenation of 2×2 reference views, resulting in an input channel of 12. Multiple inputs are concatenated in the channel dimension to form one input to $p_{\boldsymbol{\psi}_2}(\mathbf{z}|\mathbf{r},\mathbf{c})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n,\mathbf{x}_r,\mathbf{c})$. The latent variable has the same height and width as the input images and its number of channels is set to

a moderate value of 64. The viewpoint condition $\mathbf{c} \in \mathbb{R}^2$ is expanded to $\mathbb{R}^{X \times Y \times 2}$ to match the size of $\mathbf{r}$ when performing concatenation. The adaptive kernel size is 15×15 and there are 2×2 reference views, so the number of the output channel of decoders $p_{\xi_1}(\mathbf{k}|\mathbf{z})$ and $p_{\theta_1}(\mathbf{k}|\mathbf{z})$ is $15 \times 15 \times 4$.

3.4.3 Network for Large Disparity

The aforementioned concatenation of reference images and the two-dimensional adaptive kernel are suitable for LF image reconstruction with small disparities but not large disparities. First, the encoder faces difficulty in handling large disparity perception by its receptive field. Second, large disparities require large adaptive kernels, which require too much memory. To tackle these challenges, we slightly modify the pre-processing of the input and the adaptive kernel to make our method more effective for reconstructing LF images with large disparities. Supposing that the disparity range between the top-left and top-right views is $[-d_{max}, d_{max}]$, we uniformly sample D disparities within the range to form a set of disparities $d = \{d_0, d_2, ..., d_{D-1}\}$. The input reference images are warped to the top-left view according to the disparity d_i as follows

$$\mathbf{x}_r^{d_i} = \mathbf{x}_r(x + (u_r - u_0)d_i, y + (v_r - v_0)d_i, u_r, v_r) \tag{3.21}$$

where $\mathbf{x}_r^{d_i}$ is the warped image, and (u_0, v_0) indicates the viewpoint of the top-left view. We set $D = 7$ and $d_{max} = 50$ for the experiment with large disparities, corresponding to 22 $(3 \times 7 + 1)$ warped images. The input is a concatenation of the warped images that have 66 (22×3) channels. This pre-processing enables the encoder to effectively deal with the large disparities in the input images.

Second, we use a one-dimensional adaptive kernel instead of a two-dimensional one to enlarge kernel size. Similar to (3.19), the one-dimensional adaptive convolution can be expressed as

$$\begin{aligned} \mathbf{x}_n(x, y, u_n, v_n) &= \sum_{u_r}^{U_r} \sum_{v_r}^{V_r} \sum_{i}^{M} a(\mathbf{x}_r, \hat{\mathbf{k}}) + \mathbf{e}(x, y) \\ a(\mathbf{x}_r, \hat{\mathbf{k}}) &= \mathbf{x}_r(x + d_x, y + d_y, u_r, v_r)\hat{\mathbf{k}}(x, y, u_r, v_r, i) \end{aligned} \tag{3.22}$$

where

$$\begin{aligned} d_x &= (i - \frac{M}{2})(u_n - u_r) \\ d_y &= (i - \frac{M}{2})(v_n - v_r). \end{aligned} \tag{3.23}$$

Algorithm 1: Training procedure

Data: 100 LF images with size $X \times Y \times U \times V$ from [55]
Input: Initial network parameters $\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\phi}, \boldsymbol{\theta}$, parameters $\alpha, \beta, \eta, \gamma$, reference view index u_r, v_r, training epoch E, and batch size B
Result: Trained parameters $\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\phi}, \boldsymbol{\theta}$

for $i = 0$ **to** $E - 1$ **do**
  $\mathbf{x}_{n,\text{gt}}, \mathbf{x}_r, \mathbf{c} \leftarrow$ Sample batch LF images with randomly selected novel viewpoints
  begin Inference network
    $\mathbf{z} \leftarrow$ Encoder $p_{\boldsymbol{\psi}}(\mathbf{z}|\mathbf{x}_r, \mathbf{c})$
    $\mathbf{x}_{n,\text{inf}} \leftarrow$ Decoder $p_{\boldsymbol{\xi}}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$
  end
  begin Reconstruction network
    $\mathbf{z} \leftarrow$ Encoder $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}_n, \mathbf{x}_r, \mathbf{c})$
    $\mathbf{x}_{n,\text{rec}} \leftarrow$ Decoder $p_{\boldsymbol{\theta}}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$
  end
  Computing loss L
  Backpropagating and updating parameters
end

This one-dimensional adaptive convolution weights pixels sampled from the reference views using M possible disparities.

3.4.4 Training and Testing Procedures

The proposed LF reconstruction network contains CNNs and adaptive convolution, both of which are differentiable so that the network can be trained end-to-end. During training, synthesizing all novel views for a large batch of LF images requires a great deal of memory, which means it is only feasible for a small batch size. However, this is not necessary since different views in LF are similar, and calculating the errors between all synthesized views and the ground truth is not informative to train the network. Instead of synthesizing all views, in each training iteration, we randomly choose a viewpoint (u, v) and its corresponding ground truth views to train the network. In this way, the network can process more LF images in each training iteration. The training procedure of the proposed method is detailed in Algorithm 1.

We describe the testing procedure of the proposed IR-VAE in Algorithm 2. Note that there is only one forward run for the encoder network $p_{\boldsymbol{\psi}_1}(\mathbf{r}|\mathbf{x}_r)$ to generate the representation for a sparse LF image. Each novel view can then be synthesized by encoder $p_{\boldsymbol{\psi}_2}(\mathbf{z}|\mathbf{r}, \mathbf{c})$ and decoder $p_{\boldsymbol{\xi}}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$ with the inputs of the reference views $\mathbf{x}_r$, the representation $\mathbf{r}$ and the viewpoint condition $\mathbf{c}$.

Algorithm 2: Testing procedure

Input: Sparse LF image with size $X \times Y \times U_r \times V_r$, trained parameters $\boldsymbol{\psi}, \boldsymbol{\xi}, \boldsymbol{\phi}, \boldsymbol{\theta}$
Result: Reconstructed LF image with size $X \times Y \times U \times V$

$\mathbf{x}_r \leftarrow$ input sparse LF image
$\mathbf{r} \leftarrow$ Encoder $p_{\boldsymbol{\psi}_1}(\mathbf{r}|\mathbf{x}_r)$
for $u = 0$ **to** $U - 1$ **do**
  for $v = 0$ **to** $V - 1$ **do**
    $\mathbf{c} \leftarrow (u/(U-1), v/(V-1))$
    $\mathbf{z} \leftarrow$ Encoder $p_{\boldsymbol{\psi}_2}(\mathbf{z}|\mathbf{r}, \mathbf{c})$
    $\mathbf{x}_{n,\text{inf}}(*, *, u, v) \leftarrow$ Decoder $p_{\boldsymbol{\xi}}(\mathbf{x}_n|\mathbf{z}, \mathbf{x}_r, \mathbf{c})$
  end
end

Table 3.2: Training and testing datasets.

Type	Dataset	No. of scenes
Training	100 scenes [55]	100
Testing	30 scenes [55]	30
	EPFL [97]	118
	Reflective [48]	32
	Occlusions [48]	51
	Inria [98]	36
	Microscope [99]	2

3.5 Experimental Results

We evaluate the proposed network on a wide range of real-world and synthetic LF datasets. For reconstructing LF images with small disparities, we use the 100 scenes from [55] to train the proposed network and test its performance on multiple LF datasets including: the 30 testing scenes [55], EPFL [97], Reflective [48], Occlusions [48], Inria [98], and Microscope [99]. This wide range of testing datasets can comprehensively demonstrate the performance and robustness of the LF image reconstruction methods. The number of scenes in each dataset is shown in Table 3.2. LF images captured by Lytro Illum cameras have an angular resolution of 14×14 and a spatial resolution of 376×541. Due to the shape of round micro-lens, edge views are usually black and thus we extract central 8×8 views for training and testing. The training LF images are cropped to $64 \times 64 \times 8 \times 8$ patches with a stride of 16.

For reconstructing LF images with large disparities, we conduct evaluation on the Inria synthetic dataset [45]. This dataset contains 53 scenes, each with a spatial resolution of 512×512 and an angular resolution of 9×9. The disparity range for these scenes is [-20, 20] between adjacent views. We use the central 5×5 views to do the

experiment of reconstructing 5×5 views from the 2×2 corner views. We resize the images to the spatial resolution of 256×256 so that the maximum absolute disparity d_{max} between top-left and top-right views is 50 pixels. The scenes of *Electro devices, Furniture, Lion, and Toy bricks* are used as the testing scenes, and the rest 49 scenes serve as the training scenes. The training LF images are cropped to $192 \times 192 \times 5 \times 5$ patches with a stride of 16.

We adopt the Adam [100] algorithm as our optimization method. The training batch size is 32 and the epoch number is 100. The learning rate is $1e-4$ with a decay of 0.95 each epoch. The network is implemented on the deep learning framework Pytorch [101] and we use a GPU module of adaptive convolution based on CUDA to accelerate the algorithm. We train the network on two Nvidia Tesla V100 GPUs and it takes around 20 hours to train the network.

The objective quality of synthesized novel views is measured by the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [102] in the RGB color space. Only synthesized novel views are involved in the objective evaluation. The value of the SSIM is between 0 and 1 and a higher value indicates higher perceptual quality with respect to the ground truth. As PSNR and SSIM are pixel-level evaluation metrics, small variations in some pixels over multiple runs of the reconstruction models lead to negligible performance deviations. To be consistent with the compared methods that also neglect deviations, we report the results of single run without deviations. Six latest learning-based LF reconstruction methods are used for performance comparison, i.e., Wang *et al*. [17], Meng *et al*. [57], Kalantari *et al*. [55], Yeung *et al*. [76], Meng *et al*. [59], and Shi *et al*. [70]. Since the source code of Meng *et al*. [59] is not available, the results on Reflective and Occlusions are directly taken from their paper. It is noted that the other results on 30 scenes, EPFL, and Inria are not reported in [59].

Two additional methods by Jin *et al*. [72] and Wang *et al*. [77] experimented on the Y channel of the YCbCr color space for the task of $2 \times 2 \rightarrow 7 \times 7$. To make our results comparable with those reported in [72], [77], we also conduct experiments on the Y channel for the task of $2 \times 2 \rightarrow 7 \times 7$ using the same training and testing datasets as in [72], [77]. To demonstrate that the proposed method can synthesize arbitrary novel views within the four input corner views, we also conduct experiments for the task of $2 \times 2 \rightarrow 4 \times 4$ using the same models trained on the task of $2 \times 2 \rightarrow 7 \times 7$. The method from Jin *et al*. [72] that is flexible with regards to the input views and output angular resolution is employed for comparison for this task.

Table 3.3: Objective quality comparison on Lytro LF image datasets for the task of $2 \times 2 \rightarrow 8 \times 8$.

	30 scenes	EPFL	Reflective	Occlusions	Inria	**Average**
PSNR						
Wang *et al.* [17]	34.76	36.09	33.15	30.16	34.99	34.31
Kalantari *et al.* [55]	38.21	38.70	35.84	31.81	36.03	36.64
Meng *et al.* [57]	37.77	38.55	36.41	33.21	36.94	36.98
Yeung *et al.* [76]	39.22	39.57	36.47	32.68	37.22	37.54
Shi *et al.* [70]	39.45	39.76	36.28	33.98	37.13	37.86
Meng *et al.* [59]	N/A	N/A	37.01	33.10	N/A	N/A
Ours	**40.48**	**40.56**	**37.24**	**34.98**	**38.41**	**38.81**
SSIM						
Wang *et al.* [17]	0.9490	0.9411	0.9266	0.8845	0.9340	0.9285
Kalantari *et al.* [55]	0.9736	0.9574	0.9416	0.8945	0.9395	0.9430
Meng *et al.* [57]	0.9636	0.9520	0.9454	0.9140	0.9426	0.9441
Yeung *et al.* [76]	0.9773	0.9637	0.9472	0.9061	0.9524	0.9510
Shi *et al.* [70]	0.9820	**0.9703**	0.9505	0.9291	0.9563	0.9596
Meng *et al.* [59]	N/A	N/A	0.9500	0.9120	N/A	N/A
Ours	**0.9834**	0.9701	**0.9550**	**0.9303**	**0.9579**	**0.9607**

3.5.1 Objective Results

Results on Lytro Image Datasets

For the task of $2 \times 2 \rightarrow 8 \times 8$, objective results of different methods show that the proposed network significantly improves the quality of LF image reconstruction compared with the comparative methods. Table 3.3 gives a detailed objective quality comparison of these methods. As can be seen from the table, our proposed method achieves the best objective quality on all the testing datasets by large margins except the SSIM on the EPFL dataset. The method proposed by Shi *et al.* [70] performs slightly better than the proposed IR-VAE in terms of SSIM on the EPFL dataset. The PSNR improvements on these datasets are over 1 dB or close to 1 dB compared to the existing state-of-the-art methods. For instance, the proposed method improves the PSNR from 39.45 dB to 40.48 dB (1.03 dB improvement) on the 30 scenes from [55] and from 37.22 dB to 38.41 dB (1.19 dB improvement) on the Inria dataset [48]. The average PSNR and SSIM among all testing LF images largely outperform the compared methods and there is an average PSNR improvement of 0.95 dB compared with the existing best method [70]. Moreover, the proposed method obtains the best quality on all the five testing datasets in terms of PSNR, which demonstrates the robustness of the proposed method over a variety of LF scenes.

We compare our method with two additional state-of-the-art (SOTA) methods [103], [104], where some of the same benchmark datasets are employed. The method by Yang

Table 3.4: Objective quality comparison when only reconstructing the Y channel in the YCBCr color space.

	HCI new	30 scenes	Occlusions	Reflective	**Average**
PSNR, $2 \times 2 \rightarrow 7 \times 7$					
Yeung *et al*. [76]	32.30	42.77	38.88	38.33	39.99
Jin *et al*. [72]	**37.14**	42.75	38.51	38.35	40.12
Wang *et al*. [77]	34.70	43.67	39.46	39.11	40.84
Ours	36.30	**43.70**	**40.45**	**39.81**	**41.41**
SSIM, $2 \times 2 \rightarrow 7 \times 7$					
Yeung *et al*. [76]	0.900	0.986	0.980	0.960	0.978
Jin *et al*. [72]	0.966	0.986	0.979	0.957	0.977
Wang *et al*. [77]	0.974	**0.995**	0.991	0.978	0.990
Ours	**0.979**	**0.995**	**0.992**	**0.982**	**0.991**
PSNR, $2 \times 2 \rightarrow 4 \times 4$					
Jin *et al*. [72]	39.54	43.60	40.50	39.84	41.57
Ours	**39.85**	**44.76**	**42.42**	**42.61**	**43.27**
SSIM, $2 \times 2 \rightarrow 4 \times 4$					
Jin *et al*. [72]	0.973	0.988	0.985	0.961	0.981
Ours	**0.987**	**0.996**	**0.995**	**0.991**	**0.994**

et al. [103] employs 4D convolution and deconvolution layers to exploit the structure and scene information in LF images. The PSNRs in the RGB space for the task of $2 \times 2 \rightarrow 8 \times 8$ are 38.61 dB, 32.90 dB, and 35.15 dB on the 30 scenes, Occlusions, and Reflective datasets, respectively. These results are all worse than those of the proposed method (PSNRs: 40.48 dB, 34.98 dB, and 37.24 dB). Wu *et al*. proposed a spatial-angular attention network for LF reconstruction [104]. Its results measured by PSNR are 39.98 dB, 33.77 dB, and 37.77 dB on the same three datasets. It should be noted that all the compared methods are different from the proposed one. We contribute to LF image reconstruction from the perspectives of effective latent variable regularization and viewpoint-dependent indirect view synthesis.

The proposed IR-VAE achieves the best average PSNR and SSIM results compared with the SOTA methods [72], [76], [77] on the Y channel of the YCbCr color space for the task of $2 \times 2 \rightarrow 7 \times 7$. As can be observed from Table 3.4, the proposed method achieves nearly 1 dB improvement (from 39.46 dB to 40.45 dB) on the Occlusions dataset, and 0.7 dB improvement (from 39.11 dB to 39.81 dB) on the Reflective dataset compared with the existing SOTA method [77]. On the HCI new dataset [47] with large disparities, the proposed method is worse than the method in [72] but better than the method in [77] in terms of PSNR. However, the proposed method achieves the best SSIM compared with the SOTA methods [72], [76], [77] on the HCI new dataset. Overall, the proposed method improves the existing SOTA average PSNR by 0.57 dB on the Y channel of the

Table 3.5: Objective quality comparison on the Microscope dataset [99] for the task of $2 \times 2 \rightarrow 8 \times 8$.

Method	Golgi20x		Golgi40x		**Average**	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Wang *et al.* [17]	20.33	0.7604	16.41	0.6605	18.37	0.7104
Kalantari *et al.* [55]	17.03	0.7005	11.53	0.5036	14.28	0.6021
Meng *et al.* [57]	19.22	0.8076	13.81	0.5536	16.51	0.6806
Yeung *et al.* [76]	20.37	0.8252	16.11	0.6679	18.24	0.7466
Shi *et al.* [70]	20.13	0.8124	15.84	0.6453	17.99	0.7288
Ours	**21.08**	**0.8512**	**16.74**	**0.6800**	**18.91**	**0.7656**

YCbCr color space.

The results on the Y channel of the YCBCr color space for the task of $2 \times 2 \rightarrow 4 \times 4$ demonstrate that the proposed method generalizes well in synthesizing arbitrary novel views within the four input corner views. As can be observed from Table 3.4, the proposed IR-VAE significantly outperforms the method from Jin *et al.* [72] in terms of the average PSNR and SSIM for the task of $2 \times 2 \rightarrow 4 \times 4$.

Results on Microscope Dataset

The proposed IR-VAE and the other compared methods do not perform well enough on the Microscope dataset [99]. As can be observed from Table 3.5, significant performance degradation is seen for all the compared methods when processing microscopic LF images. The most recent method published in Wang *et al.* [77] fails to produce meaningful results and it thus does not appear in Table 3.5. There are two possible reasons for the poor performances. First, the image quality of this dataset is not very good. Second, these models are trained on natural images, but the data distributions of natural images and microscopic images are quite different. The proposed IR-VAE achieves the best average PSNR and SSIM compared with all the other methods, suggesting a better generalization ability of the proposed method than those compared methods.

Results on Inria Synthetic Dataset

The objective quality comparison in Table 3.6 shows that the proposed method achieves good results when reconstructing LF images with large disparities. Only the method from Wang *et al.* is used for this comparison as the other compared methods did not report results on this task. As shown in Table 3.6, the proposed method outperforms

Table 3.6: Objective quality comparison on Inria synthetic dataset [45] for LF image reconstruction ($2 \times 2 \rightarrow 5 \times 5$) with large disparities.

Scene	Wang *et al.* [17]		Ours	
	PSNR	SSIM	PSNR	SSIM
Electro devices	29.10	0.8942	**30.07**	**0.9117**
Furniture	29.98	**0.8753**	**30.14**	0.8629
Lion	31.63	**0.8964**	**32.06**	0.8830
Toy bricks	29.46	0.8782	**31.36**	**0.8881**
Average	30.04	0.8860	**30.91**	**0.8864**

the method by Wang *et al.* [17] in terms of the average PSNR and SSIM on the four testing scenes. Besides, the proposed method also runs 36 times faster than the compared method [17] on the same Nvidia Tesla V100 GPU.

3.5.2 Subjective Results

Fig. 3.5 shows a visual comparison of the error maps of synthesized novel views on the 5 scenes from the dataset in [55]. The error maps of the synthesized novel views shown in Fig. 3.5 demonstrate the superiority of the proposed method in synthesizing high-quality novel views. The error maps of the proposed method have less significant erroneous pixels compared with the other methods. A subjective comparison of the synthesized novel views on the Rock scene from the dataset in [55] is given in Fig. 3.6. As can be observed from the figure, the compared methods fail in synthesizing photo-realistic local details, while the proposed method successfully produces a visually appealing result. Specifically, the ghost phenomenon appears in the local detail of Kalantari *et al.* [55]. The cars are distorted in the results from Meng *et al.* [57] and Yeung *et al.* [76]. The method from Shi *et al.* [70] produces a comparable result, but the details of the leaf are lost. By contrast, our result is sharper and more precise than these competing methods. This subjective comparison illustrates that our proposed method can well handle occlusive pixels and small objects.

3.5.3 Runtime and Memory Consumption

We provide detailed comparisons of runtime and memory consumption in Table 3.7. All reported runtimes are measured on the same Nvidia Tesla V100 GPU when reconstructing a full angular LF image of spatial resolution 376×541. As some methods divide an LF image into patches in reconstruction, memory consumptions are measured on the same hardware when reconstructing a full angular LF image of spatial resolution 128×128 without patching. As shown in Table 3.7, our method requires

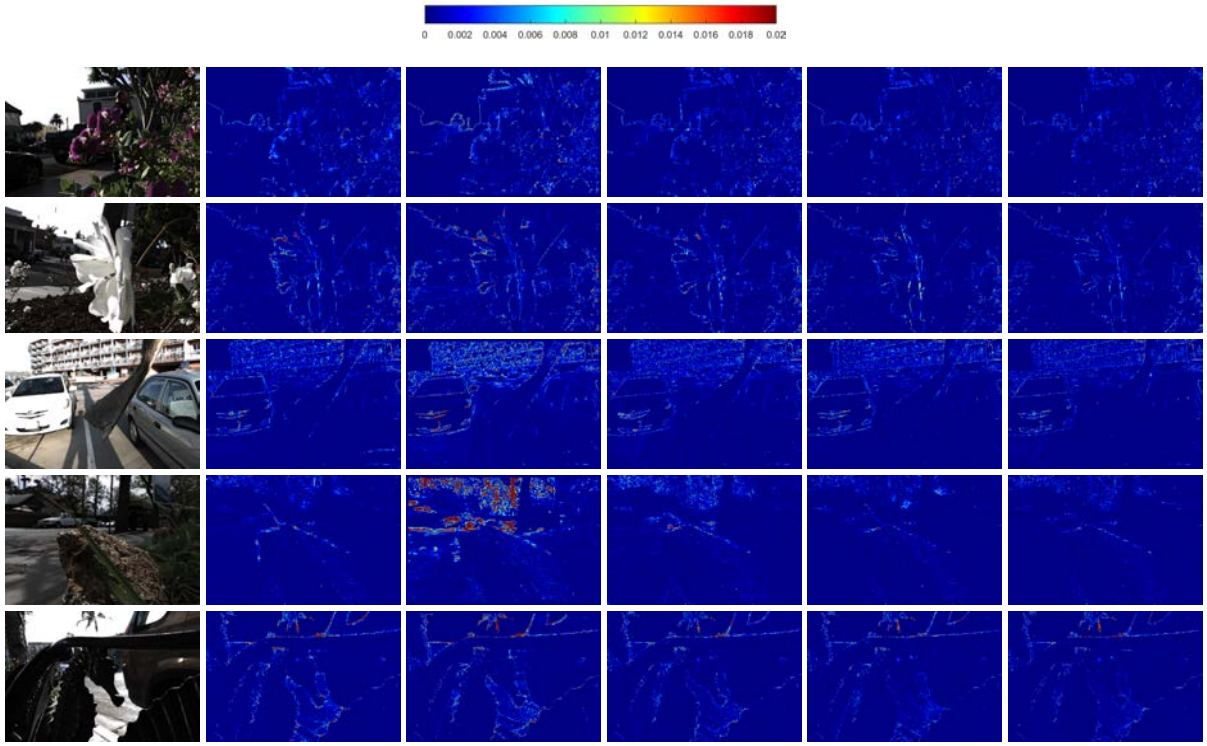
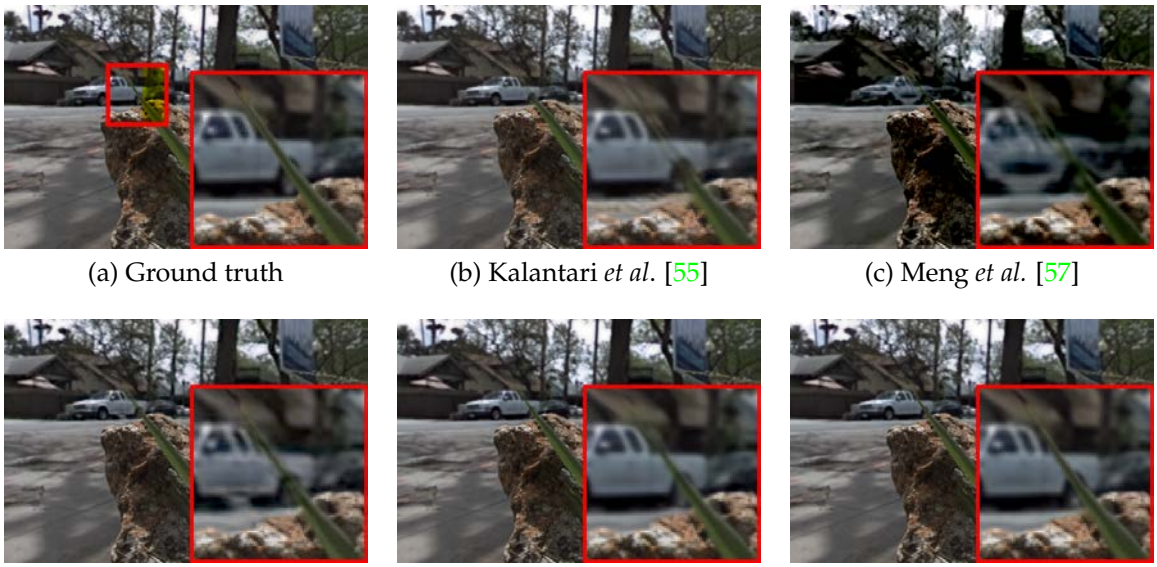


Figure 3.5: Visual comparison of error maps of synthesized novel views. The shown results are from (a) ground truth, (b) Kalantari *et al*. [55], (c) Meng *et al*. [57], (d) Yeung *et al*. [76], (e) Shi *et al*. [70]. The novel views are central views of the corresponding LF images and are also the most challenging ones in LF reconstruction. These scenes are named Flower1, Flower2, Cars, Rock, and Seahorse, respectively. The error maps are based upon the square error averaged over the RGB channels, where pixel intensities are between 0 and 1.

7.5 seconds to reconstruct 8×8 LF views from 2×2 views, which is slower than the methods by Yeung *et al*. [76] (1.3s) and Meng *et al*. [57] (5.5s), but faster than the other compared methods. When considering achieving the SOTA reconstruction quality in Tables 3.3 and 3.4, the proposed method has the minimum runtime. For example, the proposed method achieves a significantly better average PSNR than the existing best method in [77], and runs much faster than it. Besides, the proposed method has the minimum memory consumption among all the compared methods.

The computational performance of the proposed network stems from its architectural advantage. In the proposed network, the encoder $p_{\psi_1}(\mathbf{r}|\mathbf{x}_r)$ is a comparable large network but it only needs to be run once at the resolution of $X \times Y$ (spatial resolution of the input LF image). Each view in the LF image can then be synthesized by the small encoder $p_{\psi_2}(\mathbf{z}|\mathbf{r},\mathbf{c})$ and the small decoder $p_\xi(\mathbf{x}_n|\mathbf{z},\mathbf{x}_r,\mathbf{c})$ independently based on the representation from the encoder $p_{\psi_1}(\mathbf{r}|\mathbf{x}_r)$. This architecture design greatly reduces the inference time and memory consumption. As a comparison, for example, the network by Jin *et al*. [72] has a smaller model size than ours, but it needs to run the whole



(a) Ground truth (b) Kalantari *et al.* [55] (c) Meng *et al.* [57]

(d) Yeung *et al.* [76] (e) Shi *et al.* [70] (f) Ours

Figure 3.6: Subjective comparison of synthesized novel views for the Rock scene.

synthesis network for each view and refine the full angular LF image by 4D convolutions at the resolution of $X \times Y \times U \times V$, resulting in significantly longer runtime and larger memory consumption than the proposed network, as shown in Table 3.7.

3.5.4 Ablation Study

We study the impact of constituent modules of the proposed IR-VAE on LF reconstruction quality. Different network architectures use the same encoder and decoder structures as described in Section 3.4.2. The indirect output method means the proposed viewpoint-dependent indirect view synthesis (VIVS) method, while the direct output method means directly outputting novel pixels according to viewpoints by the last convolutional layer. As shown in Table 3.8, the inference-only network without the proposed VIVS does not perform well in terms of the average PSNR. As a comparison, the inference-only network in conjunction with the VIVS achieves a significantly better reconstruction quality (2.6 dB improvement in terms of the average PSNR) than the one without the VIVS. This comparison shows that carefully designing the architecture of the inference network alone cannot lead to good LF reconstruction quality. Also, the IR-VAE with the direct output method performs poorly. These results demonstrate the effectiveness of the proposed VIVS method.

The conditional variational autoencoder (CVAE) uses the proposed MMDVAE as the statistic measure between two distributions as the high-resolution representation is not applicable to standard MMD. The CVAE with VIVS performs even worse than the network containing only the inference network. The results of the CVAE validate our

Table 3.7: Runtime and memory consumption. All runtimes are measured on the same Nvidia Tesla V100 GPU when reconstructing a full angular LF image of spatial resolution 376×541. As some methods divide an LF image into patches in reconstruction, memory consumptions are measured when reconstructing a full angular LF image of spatial resolution 128×128 without patching.

Task	Method	Runtime in seconds	Memory in GiB
$2 \times 2 \rightarrow 8 \times 8$	Wang *et al.* [17]	130	3.51
	Kalantari *et al.* [55]	387	1.11
	Meng *et al.* [57]	5.5	3.21
	Yeung *et al.* [76]	**1.3**	1.04
	Shi *et al.* [70]	660	10.4
	Ours	7.5	**0.25**
$2 \times 2 \rightarrow 7 \times 7$	Jin *et al.* [72]	27	1.42
	Wang *et al.* [77]	14	0.43
	Ours	**4.9**	**0.24**
$2 \times 2 \rightarrow 4 \times 4$	Jin *et al.* [72]	11	1.22
	Ours	**1.6**	**0.24**

analysis in Section 3.3.2 that inferring and reconstructing novel views from different latent variables by the same decoder causes interference for the network. The proposed IR-VAE (with VIVS) avoids such interference through the newly designed inference-reconstruction framework and achieves significantly better reconstruction results than both the inference network alone and the CVAE. It should be noted that the proposed IR-VAE only evokes the inference network in testing, whereas the reconstruction network is only executed in training. This means the proposed IR-VAE shares the same network structure and the forward process with the inference-only network. The fact that the proposed IR-VAE outstrips the inference-only network demonstrates that a good latent variable (or a hidden representation) is crucial for LF reconstruction.

3.5.5 Visualization of Viewpoint-dependent Adaptive Kernels

We visualize the predicted adaptive kernels for various novel views to reveal how the adaptive kernels adapt to the viewpoints of novel views. As shown in Fig. 3.7, we select a pixel in a top-right reference view and visualize the generated adaptive kernels used to warp the reference pixel and its neighbors to different novel pixels. The adaptive kernel at viewpoint $(0, 7)$ is for the reference view itself so that the kernel has large weights around the central pixel. The values of the weights decrease with the distance between the reference and novel views, which means the contribution of this reference view to the novel views gradually diminishes. It is also noted that the positions of significant weights gradually move away from the central pixel when

Table 3.8: Ablation study of the proposed method.

Network	Output	30 scenes	EPFL	Reflective	Occlusions	Inria	**Average**
PSNR							
Only Inference	Direct	36.35	37.43	34.83	32.83	35.68	35.88
Only Inference	Indirect	40.12	40.23	36.90	34.64	38.14	38.48
CVAE	Indirect	39.87	40.17	36.91	34.49	38.11	38.40
IR-VAE	Direct	36.65	37.63	34.89	33.02	35.90	36.09
IR-VAE	Indirect	**40.48**	**40.56**	**37.24**	**34.98**	**38.41**	**38.81**
SSIM							
Only Inference	Direct	0.9706	0.9569	0.9411	0.9120	0.9429	0.9461
Only Inference	Indirect	0.9821	0.9683	0.9531	0.9244	0.9557	0.9581
CVAE	Indirect	0.9810	0.9675	0.9521	**0.9306**	0.9542	0.9584
IR-VAE	Direct	0.9622	0.9452	0.9286	0.9015	0.9291	0.9347
IR-VAE	Indirect	**0.9834**	**0.9701**	**0.9550**	0.9303	**0.9579**	**0.9607**

the distance between the reference view and the novel views grows. This position shift of the significant weights demonstrates that our model is capable of successfully capturing the geometric information between the reference and novel views without requiring the ground truth information.

3.5.6 Limitations

A limitation of the proposed method is that the computational complexity of the proposed IR-VAE is higher than existing fast methods in [57], [76]. As can be observed from Table 3.3, the proposed IR-VAE runs slower than those by Meng *et al.* [57] and Yeung *et al.* [76]. but faster than other compared methods. However, the reconstruction quality of the proposed IR-VAE is better than these existing methods. Therefore, the proposed IR-VAE is useful in the scenario where reconstruction quality is the priority concern. The proposed method that accepts the four corner views as input is not as flexible as the method from Jin *et al.* [72] that can handle the input of arbitrarily sampled LF views. Combining the method in [72] that explicitly utilizes the viewpoint relationship between the reference and novel views with our proposed method may circumvent this limitation, which we will investigate in our future work.

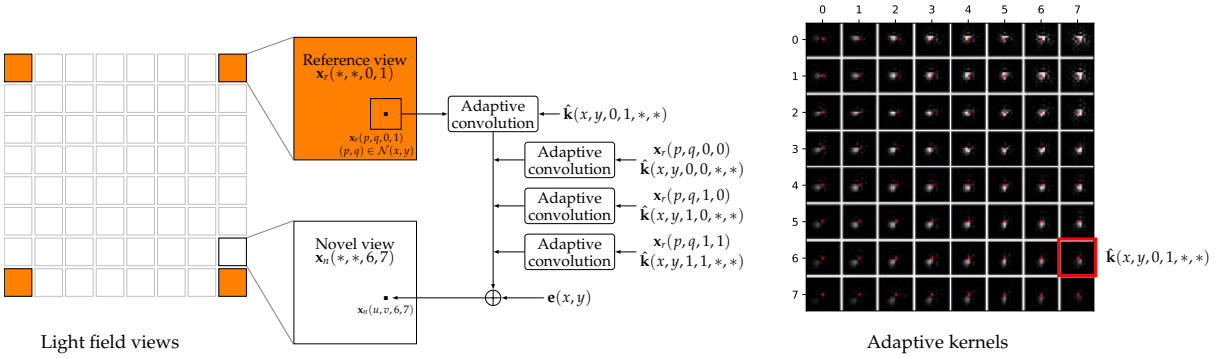


Figure 3.7: Visualization of viewpoint-dependent adaptive kernels. The left is the process of synthesizing the novel pixel $\mathbf{x}_n(x,y,6,7)$ by employing adaptive convolution on four reference views. Pixels at $(p,q) \in \mathcal{N}(x,y)$ are neighboring pixels of the central pixel at (x,y) which are applied with adaptive convolution. Adaptive kernel $\hat{\mathbf{k}}$ and bias $\mathbf{e}$ depend on the novel viewpoint. It should be noted that the reference views $\mathbf{x}_r$ are with the resolution of $X \times Y \times 2 \times 2$. So, the maximum angular coordinates for $\mathbf{x}_r$ are $(1,1)$. The right shows the adaptive kernels used to warp the reference pixels $\mathbf{x}_r(p,q,0,1)$ to different novel pixels, where the adaptive kernel $\hat{\mathbf{k}}(x,y,0,1,*,*)$ at $(6,7)$ in the red rectangle is used to produce novel pixel $\mathbf{x}_n(x,y,6,7)$. We highlight the kernel center in red. The axes give the position of viewpoints. A light intensity in adaptive kernels means a high adaptive weight. The weights decrease with the distance between the reference view $\mathbf{x}_r(*,*,0,1)$ and the novel view, indicating that the importance of $\mathbf{x}_r(*,*,0,1)$ decreases when the novel view moves away from it. The positions of significant weights also shift according to viewpoints. This means the proposed IR-VAE is capable of ascertaining the geometric relationship between the reference and novel views.

3.6 Conclusion

We proposed the novel inference-reconstruction variational autoencoder (IR-VAE) to synthesize novel views for the purpose of reconstructing dense LF images. The proposed IR-VAE framework utilizes the constituent inference and reconstruction networks to facilitate information flow between the latent variables and novel views and eliminate the interference caused by yielding the same novel view from two different latent variables by the same decoder. Then we proposed the mean local maximum mean discrepancy (MLMMD) to measure the statistical distance of two distributions in the latent variable space. This enables richer representations of reference views and viewpoints by high-resolution latent variables. Finally, we proposed the viewpoint-dependent indirect view synthesis method that transforms the prediction of raw novel pixels into adaptive kernels and bias. An ablation study was conducted to show the effectiveness of our proposed modules. Experimental results were presented to demonstrate that our model significantly outperforms existing state-of-the-art methods on both subjective and objective comparisons.

Chapter 4

Neural Radiance Feature Field for View Rendering

We can apply the local light field reconstruction method presented in the previous chapter to render novel view from any position when we have multi-view images that globally cover the scene. But such rendering only utilizes local captured images. A global light field reconstruction that leverages all multi-view images can lead to better reconstruction accuracy and rendering quality. This chapter investigates global light field reconstruction using neural rendering technique. Again, an effective scene representation is the key for reconstruction quality. In this chapter, we present a multiscale tensor decomposition representation and a rendering equation encoding method to represent scenes in the feature space, resulting in significantly better view rendering quality compared with existing state-of-the-art methods.

4.1 Introduction

View rendering aims at synthesizing unrecorded views from multiple captured views using computer vision techniques. A great deal of effort has been made to solve this problem in the past few decades [2]. The recently proposed neural radiance field (NeRF) [7] made a breakthrough in this area by modeling a scene via a multilayer perceptron (MLP). The NeRF achieves an impressive photo-realistic view synthesis quality with 6 degrees of freedom for the first time. The NeRF also represents a scene in a very compact form. That is, only a small number of parameters in the MLP, whose size is even smaller than the captured images. However, this advantage in model size comes at the expense of extensive computations. Numerous evaluations of the MLP are required to render a single pixel, incurring a significant challenge for both training and testing.

Representing a scene via learnable features is shown to be an effective alternative

approach for photo-realistic view rendering [22], [23], [105], [106]. Several data structures are employed to efficiently organize learnable features to achieve compact representations. Multiresolution hash encoding (MHE) [22] and tensor decomposition in TensoRF [23] are two typical works in this direction. MHE organizes learnable features in multiresolution hash tables. As each hash table corresponds to a distinct grid resolution, a point is thus indexed into different positions of the hash tables to mitigate the negative effects of hash collisions. However, this structure breaks the local coherence in nature scenes, even though the spatial hash function in MHE preserves the coherence to some extent. By comparison, TensoRF decomposes a 3D tensor into 2D plane and 1D line tensors, where the local coherence is largely preserved. However, TensoRF's decomposition is performed only in a single scale, whereas multiscale methods are much more desirable for wide-ranging computer vision tasks [107]–[109]. We thus propose a multiscale tensor decomposition (MTD) method to represent scenes from coarse to fine scales. We show that the proposed MTD method is able to reconstruct more accurate scene shapes and appearances, and also converges faster than the single-scale TensorRF. As a result, the proposed MTD method achieves better view rendering quality than TensoRF, even with fewer learnable features.

View direction encoding is the key to the success of neural rendering in modeling complex view-dependent effects. Frequency (or position encoding) [7] and spherical harmonics [21] are the two mostly used view direction encoding methods. The encoded feature vector of a view direction is then fed to an MLP to predict a view-dependent color. This approach models the 5D light field function (3D spatial position with 2D view direction) [1]. In computer graphics, the light field is usually modeled by the rendering equation [24], where the outgoing radiance is the interaction result of the incoming light at a point with a specific material. An accurate solution to the rendering equation involves Monte Carlo sampling and integration, which is computationally expensive, especially for the scenario of inverse rendering [110]. In this chapter, we propose to encode the rendering equation in the feature space in lieu of the color space using the predicted anisotropic spherical Gaussian mixture. In this way, the following MLP becomes aware of the rendering equation so as to better model complex view-dependent effects. As we use both neural and learnable feature representations as well as the rendering equation encoding in the feature space, we dub the proposed method the neural radiance feature field (NRFF). In summary, we make the following contributions:

- We propose a novel multiscale tensor decomposition scheme to represent scenes from coarse to fine scales, enabling better rendering quality and faster convergence with fewer learnable features;

- In lieu of direct encoding of view directions, we propose to encode the rendering equation in the feature space to facilitate the modeling of view-dependent effects.

4.2 Related Work

We divide view rendering methods into neural and learnable feature representations depending on whether extra learnable parameters are used to represent a scene in addition to weights and biases in neural networks.

4.2.1 Neural Representations

Neural representations mean representing a scene by neural networks, typically MLPs [7] or transformers [111]. Mildenhall *et al*. [7] first proposed this idea for view synthesis in the NeRF and achieved photo-realistic view rendering results. The MLP in the NeRF is optimized to predict the volume density and the view-dependent appearance of a 3D spatial point observed from a given 2D view direction. Each component in this 5D input is encoded by a set of functions, *e.g*., sine and cosine, with varying periods before being fed to the MLP. Such position or frequency encoding is one of the key factors to NeRF's success. The input encoding has been further explored in [112] by a neural tangent kernel and extended in mip-NeRF [20] to achieve anti-aliasing view rendering. Neural representations have the advantage of representing a scene in a very compact form. MLPs are also used to predict the light source visibility of a point to enable relighting [113], [114]. However, these methods are computationally expensive because numerous evaluations of the networks are needed to render a single pixel.

Encoding view directions is important for neural representations to achieve photo-realistic view rendering. Except for the aforementioned position encoding, spherical harmonics are also used to encode view directions with various frequency components [21]. This approach composed of view direction encoding and the following MLP modeling is the dominant solution in the current neural rendering approaches. Such view direction encoding methods provide view direction information in various frequencies but neglect the rich information contained in the well-known rendering equation [24]. In this chapter, instead of encoding view directions, we propose to encode the rendering equation to facilitate the learning of complex view-dependent effects for the following MLP.

4.2.2 Learnable Feature Representations

Learnable features are parameters that are also optimized by gradient descent in addition to weights and biases in neural networks. Learnable features are usually organized by the data structures of grids, sparse grids, trees, and hash tables. For a given input, interpolation is performed to obtain the corresponding features. The interpolated features can be directly interpreted as some properties, e.g., densities or colors, or optionally fed into neural networks to predict the designed outputs. Compared with pure neural representations, learnable feature representations are computationally efficient at the expense of memory footprint. As the features are also optimized for the considered scene, the task of inferring scene properties for the subsequent MLP is much easier in comparison with predicting from input coordinate encoding. As a result, with learnable feature representations, small MLPs are able to achieve a competitive rendering quality similar to pure neural representations.

Efficient data structures to arrange learnable features are crucial in terms of both computational cost and memory consumption. The 3D dense grid is a significant waste of memory because most of the voxels are empty. Its number of parameters increases by $\mathcal{O}(N^3)$. Thus, the 3D dense grid is only practical at low resolution, e.g., $N = 160$ in [106], limiting its rendering quality. The Octree [115] and sparse 3D grid [105] are also employed but data structures need to be updated progressively. Because scene geometry only emerges during training. The recently proposed MHE [22] is a very compact learnable feature representation but hash collision and the break of spatial coherence limit its rendering quality. Concurrent tensor decomposition in TensoRF [23] preserves spatial coherence but is only performed at a single scale. The benefits of multiscale schemes [107]–[109] studied in the literature inspire us to propose the MTD scheme to represent scenes at varying scales.

4.3 Method

The proposed NRFF obtains the view-dependent color of a point through two main steps. For a point $\mathbf{x} = (x, y, z)$ sampled from a cast ray $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, where $\mathbf{o}$ and $\mathbf{d}$ are the camera center and view direction, respectively, we first compute its feature vector from the proposed multiscale representation. The feature vector is fed into a spatial MLP to predict light parameters used to encode the rendering equation. Next, we apply the proposed rendering equation encoding and then use a directional MLP to predict the final color.

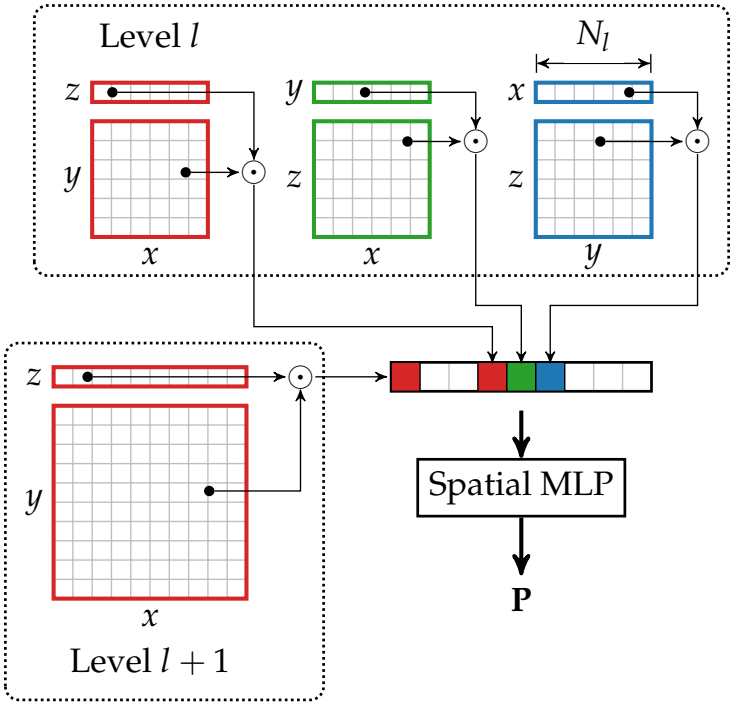


Figure 4.1: Multiscale tensor decomposition representation. At each level, a 3D tensor representation is decomposed to three sets of plane feature maps and line feature vectors. The resolution of decomposed tensors increases with the level, enabling scene representation at different scales. The concatenated feature vectors from all levels are used to predict parameters **P** by a spatial MLP.

4.3.1 Multiscale Tensor Decomposition

We propose a multiscale tensor decomposition (MTD) scheme to represent a scene from coarse to fine scales. Similar ideas have been widely used in other computer vision works in the literature [107]–[109]. We start with a base resolution of N_{min} and progressively increase the level resolution to the maximum resolution of N_{max} by a factor b, in line with the strategy in MHE [22]:

$$N_l = \lfloor N_{\text{min}} b^l \rfloor \tag{4.1}$$

$$b = \exp\left(\frac{\ln N_{\text{max}} - \ln N_{\text{min}}}{L-1}\right) \tag{4.2}$$

where N_l is the resolution at level l and L is the number of multiscale levels. Feature vectors of point $\mathbf{x}$ are obtained from the proposed MTD independently at different levels. As shown in Fig. 4.1, we use the tensor decomposition mechanism [23] that decomposes a 3D tensor representation into three plane feature maps and three line feature vectors. We apply linear interpolation (bilinear interpolation for 2D) to the plane feature map $\mathbf{F}^l_{xy}$ and the feature vector $\mathbf{F}^l_z$ using the corresponding decomposed

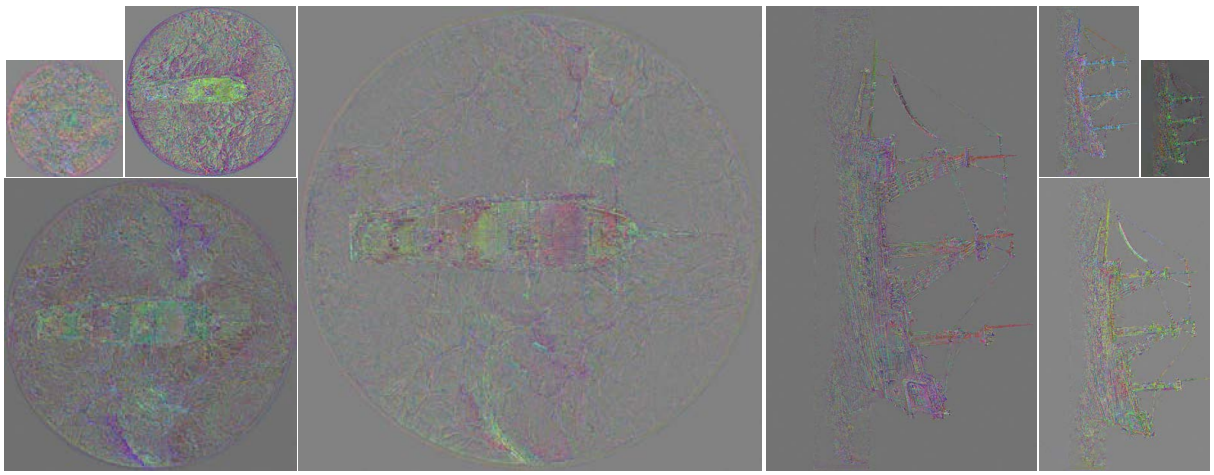


Figure 4.2: Visualization of plane feature maps of different resolutions on the *ship* scene from the NeRF synthetic dataset [7]. Coarse scene information is represented at low resolutions, while fine details are of high-resolution representations. Readers are encouraged to zoom in for a detailed inspection.

coordinates $\mathbf{x}_{xy}, \mathbf{x}_z$ to obtain the following two feature vectors:

$$\begin{aligned} \mathbf{f}_{xy}^l &= \text{Interp2D}(\mathbf{F}_{xy}^l, \mathbf{x}_{xy}) \\ \mathbf{f}_z^l &= \text{Interp1D}(\mathbf{F}_z^l, \mathbf{x}_z). \end{aligned} \tag{4.3}$$

The output feature vector at level l is obtained as follows:

$$\mathbf{f}_{xy,z}^l = \mathbf{f}_{xy}^l \odot \mathbf{f}_z^l \tag{4.4}$$

where $\odot$ denotes the element-wise multiplication. Feature vectors from other levels are obtained similarly. The output feature vectors $\left[..., \mathbf{f}_{xy,z}^l, \mathbf{f}_{xz,y}^l, \mathbf{f}_{yz,x}^l, \mathbf{f}_{xy,z}^{l+1}, ...\right]$ from all levels are concatenated and then fed into a spatial MLP to predict parameters $\mathbf{P}$, which will be detailed in Section 4.3.2.

The proposed multiscale scheme brings about three main benefits compared with the single-scale tensor decomposition in TensoRF [23]. First, it enables better exploration of the local smoothness of nature scenes at varying scales. Coarse-scale representations are inherently smooth, while fine-scale representations provide rich local details. Fig. 4.2 illustrates different levels of information represented in the multiscale feature maps. It should be noted that the goal of the multiscale scheme here is different from that of MHE [22]. MHE uses multiresolution mainly for mitigating the negative effects of hash collisions as points are indexed to different positions in the hash tables at varying resolutions. Second, the number of feature channels at each scale could be significantly smaller than that in the single-scale representation, enabling high-resolution representations to explore richer details. For example, a multiscale representation with

16 levels, a maximum resolution of 512, and 4 feature channels has 8.5M parameters, which are fewer than 13M parameters in a single-scale TensoRF with a resolution of 300 and 48 feature channels. In Section 4.4.3, we show that even with fewer parameters, the proposed MTD method outperforms the single-scale TensoRF in terms of rendering quality. Third, scene geometry appears fast in coarse-scale representations, leading to faster convergence than the single-scale representation.

4.3.2 Rendering Equation Encoding

A light field can be defined as the radiance at a point in a given direction [1]. It is thus represented by a 5D function $L(\mathbf{x}, \boldsymbol{\omega}_o)$, where $\mathbf{x} \in \mathbb{R}^3$ is the spatial position and $\boldsymbol{\omega}_o \in \mathbb{R}^2$ (spherical coordinate) is the outgoing radiance direction. This 5D light field is the result of the interaction of the scene shape, material, and lighting, which is usually modeled by the rendering equation [24] consisting of the diffuse and specular components:

$$\begin{aligned} L(\boldsymbol{\omega}_o; \mathbf{x}) &= \mathbf{c}_d + \mathbf{s} \int_\Omega L_i(\boldsymbol{\omega}_i; \mathbf{x}) \rho_s(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathbf{x}) (\mathbf{n} \cdot \boldsymbol{\omega}_i) \, d\boldsymbol{\omega}_i \\ &= \mathbf{c}_d + \mathbf{s} \int_\Omega f(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathbf{x}, \mathbf{n}) \, d\boldsymbol{\omega}_i \end{aligned} \tag{4.5}$$

where $\mathbf{c}_d$ indicates the diffuse color and $\mathbf{s}$ is the weight of the specular color. Symbol $\cdot$ indicates the dot product in the Cartesian coordinate system. $L_i(\boldsymbol{\omega}_i; \mathbf{x})$ is the incoming radiance from direction $\boldsymbol{\omega}_i$, and $\rho_s(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o; \mathbf{x})$ represents the specular component of the spatially-varying bidirectional reflectance distribution function (BRDF). For ease of exposition, we define f as a function describing the outgoing radiance after the ray interaction. The integral is solved over the hemisphere Ω defined by the normal vector $\mathbf{n}$ at point $\mathbf{x}$. In computer graphics, $L_i, \rho_s, \mathbf{n}$ are usually known functions or parameters that describe scene lighting, material, and shape. An accurate solution to the rendering equation is achieved by computationally intensive Monte Carlo estimation in the color space, *e.g.*, computing the discrete summation by evaluating L_i, ρ_s at sampled incoming radiance direction $\boldsymbol{\omega}_i$ for a given outgoing radiance direction $\boldsymbol{\omega}_o$.

In the inverse rendering problem, $L_i, \rho_s, \mathbf{n}$ are unknown functions or parameters. The most popular method in the inverse rendering to solve the equation is to treat it as a function of $\boldsymbol{\omega}_o$, and then employ an MLP to directly predict the integral result from the encoded $\boldsymbol{\omega}_o$. However, this simplification neglects the rich information described in the rendering equation and gives the MLP a complicated function to learn. Recent studies have also attempted to estimate the unknown properties to achieve relightable view rendering [114], [116]–[118]. But their rendering quality is inferior to methods [20]–[23] that focus only on view rendering with fixed lighting conditions. We instead propose to encode the rendering equation in the feature space and let the MLP predict

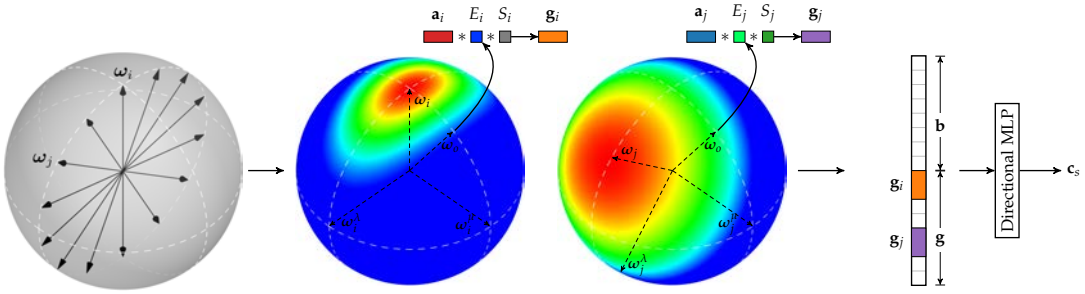


Figure 4.3: Illustration of the proposed rendering equation encoding. The rendering equation is encoded in the feature space by the learned ASG mixture with predefined orthonormal axes. The axes are defined by a set of radiance directions uniformly sampled on a unit sphere. Here only sampled $\boldsymbol{\omega}$ on a plane are shown for better visualization. For a sampled $\boldsymbol{\omega}_i$, an ASG function of the reparameterized view direction $\boldsymbol{\omega}_o$ is employed to determine the feature response $\mathbf{g}_i$. E_i and S_i are exponential and smooth terms, respectively, and $*$ denotes multiplication. Each ASG function is controlled by learned bandwidths λ and μ. The encoded feature vector $\mathbf{g}$ along with a bottleneck feature vector $\mathbf{b}$ depending only on the spatial position, are fed into a directional MLP to predict the specular color $\mathbf{c}_s$.

the integrated color from the resultant encoding. By doing this, the following MLP becomes aware of the rendering equation, making the learning task much easier for the MLP.

While encoding the rendering equation in the color space has a clear physical meaning, difficulties in three aspects limit its performance. First, the MLP yields the color parameters in the rendering equation by its final layer. Before the final layer, the MLP does not even know the outgoing radiance direction. This means the MLP does not benefit from the rendering equation as its input does not include information relevant to the rendering equation. Instead, the MLP only learns a spatial function of the position of the input point. Second, using the Monte Carlo integration technique to solve the rendering equation requires many samples to achieve a satisfactory quality in the color space, while extensive sampling is expensive in the inverse rendering problem [110], [114]. In the feature space, a feature vector consisting of a small number of sampled features could be a comprehensive representation. We show that 128 samples in the feature space are sufficient to render high-quality views. Last, in the color space, approximating the rendering equation by some basis functions (typically spherical Gaussians [119], [120] or spherical harmonics [121]) leads to a closed-form solution so that sampling over $\boldsymbol{\omega}_i$ can be avoided. However, for the inverse rendering problem, the parameters of the basis functions are unknown and predicted from the MLP. Deriving the final color using the computation (*e.g.*, the product of spherical harmonic coefficients [121]) of predicted parameters does not provide much additional useful information for the MLP.

We thus encode the rendering equation in the feature space by viewing f as a function of $\boldsymbol{\omega}_o$ for a sampled $\boldsymbol{\omega}_i$. In this perspective, we can apply a feature function to each sampled $\boldsymbol{\omega}_i$. We use the anisotropic spherical Gaussian (ASG) [119] as the feature function to encode the rendering equation:

$$\begin{aligned}\mathbf{c}'_s(\boldsymbol{\omega}_o;\mathbf{x}) &= \sum_{i=0}^{N-1} G_i(\boldsymbol{\omega}_o;\mathbf{x},[\boldsymbol{\omega}_i,\boldsymbol{\omega}_i^{\lambda},\boldsymbol{\omega}_i^{\mu}],[\lambda_i,\mu_i],\mathbf{a}_i) \\ &= \sum_{i=0}^{N-1} \mathbf{a}_i S(\boldsymbol{\omega}_o;\boldsymbol{\omega}_i)\exp\left(-\lambda_i(\boldsymbol{\omega}_o\cdot\boldsymbol{\omega}_i^{\lambda})^2-\mu_i(\boldsymbol{\omega}_o\cdot\boldsymbol{\omega}_i^{\mu})^2\right)\end{aligned} \tag{4.6}$$

where $\mathbf{c}'_s(\boldsymbol{\omega}_o;\mathbf{x})$ is a feature representation of the specular integral in (4.5); $\mathbf{a}_i$ is a feature vector; $[\boldsymbol{\omega}_i,\boldsymbol{\omega}_i^{\lambda},\boldsymbol{\omega}_i^{\mu}]$ (lobe, tangent and bi-tangent) are predefined orthonormal axes satisfying $\boldsymbol{\omega}_i\cdot\boldsymbol{\omega}_i^{\lambda}=\boldsymbol{\omega}_i\cdot\boldsymbol{\omega}_i^{\mu}=\boldsymbol{\omega}_i^{\lambda}\cdot\boldsymbol{\omega}_i^{\mu}=0$; $\lambda_i,\mu_i>0$ are the bandwidths for $\boldsymbol{\omega}_i^{\lambda},\boldsymbol{\omega}_i^{\mu}$ axes, controlling the shape of the ASG function; $S(\boldsymbol{\omega}_o;\boldsymbol{\omega}_i)=\max(\boldsymbol{\omega}_o\cdot\boldsymbol{\omega}_i,0)$ is a smooth term. G_i is thus viewed as a function of $\boldsymbol{\omega}_o$ defined at the sampled $\boldsymbol{\omega}_i$.

A problem of using $\boldsymbol{\omega}_o=-\mathbf{d}$ to encode the rendering equation is that the ASG does not match the behavior of physical specular reflection. According to the law of reflection, the most significant energy from an incoming radiance in direction $\boldsymbol{\omega}_i$ is in the area centered at the reflective direction defined to have the same angle to the surface normal as the incoming radiance, but on the opposite side [122]. However, the energy centers of the ASG functions are in the sampled incoming radiance directions $\boldsymbol{\omega}_i$. We tackle this problem by reparameterizing the view direction to the opposite reflective direction, and treat the reparameterized direction as the outgoing radiance direction $\boldsymbol{\omega}_o$:

$$\boldsymbol{\omega}_o = 2(\mathbf{d}\cdot\mathbf{n})\mathbf{n}-\mathbf{d}. \tag{4.7}$$

After reparameterization, the rendering equation encoding matches the physical specular reflection behavior as $\boldsymbol{\omega}_o$ aligns with $\boldsymbol{\omega}_i$. This reparameterization has also been shown to be able to simplify view interpolation, as studied in [21], [123].

As depicted in Fig. 4.3, we sample $N=8\times16$ lobes on a unit sphere and determine tangent and bi-tangent axes according to their orthonormal constraint. For a sampled $\boldsymbol{\omega}_i=(\theta,\phi)$ in the spherical coordinate system, we define $\boldsymbol{\omega}_i^{\lambda}=(\theta+\pi/2)$ and rotate $\boldsymbol{\omega}_i^{\lambda}$ around $\boldsymbol{\omega}_i$ by $\pi/2$ using the quaternion operation to obtain $\boldsymbol{\omega}_i^{\mu}$. Two ASG examples in Fig. 4.3 show that such ASGs have a strong representation ability to model the rendering equation in the feature space. One can simply solve (4.6) by computing the sum of encoded feature vectors. However, this sum reduction significantly reduces the channels of the feature representation, limiting its representative ability. Instead, we

form a comprehensive feature vector $\mathbf{g}$ by concatenating the encoded feature vectors:

$$\mathbf{g} = [\mathbf{g}_0, \mathbf{g}_1, ..., \mathbf{g}_{N-1}]. \tag{4.8}$$

Together with a spatial bottleneck feature vector $\mathbf{b}$, we apply a directional MLP to predict the specular color $\mathbf{c}_s$. The required parameters for the ASG encoding are from $\mathbf{P}$, which are predicted by the spatial MLP from the spatial feature vector obtained using (4.4) as aforementioned in Section 4.3.1. In summary, $\mathbf{P}$ include the following parameters: $\{\mathbf{c}_d, \mathbf{s}, \mathbf{n}, \mathbf{b}, \mathbf{a}_i, \lambda_i, \mu_i\}$. Finally, we apply the sigmoid activation function to the combined color to obtain the view-dependent color:

$$\mathbf{c} = \text{Sigmoid}(\mathbf{c}_d + \mathbf{s} \odot \mathbf{c}_s). \tag{4.9}$$

We can also interpret the proposed rendering equation encoding as a more advanced view direction encoding method. Our rendering equation encoding has twofold benefits compared with popular frequency encoding [7] and sphere harmonics [21]. First, every point now has its own independent encoding functions controlled by the predicted bandwidths in the ASGs, while the encoding functions are fixed for all points in existing works. Second, a diverse of ASG functions can be produced to achieve much richer encoding compared with a few fixed basis encoding functions in existing methods [7].

4.3.3 Volume Rendering

We use the differentiable volume rendering technique [7] to render a ray according to predicted densities and view-dependent colors. The scene density and appearance fields are modeled separately by two MTD representations. For a point $\mathbf{x}_i$ sampled at depth t_i, its density σ_i is the result of the softplus activation of the sum of the feature vectors at all levels. The color of the considered point is obtained by the method described in Section 4.3.2. We compute the color composition weights based on densities as follows:

$$w_i = \exp\left(-\sum_{j=0}^{i-1} \sigma_j \Delta_j\right)(1 - \exp(-\sigma_i \Delta_i)) \tag{4.10}$$

where Δ is the sampling interval. We follow the method in TensoRF [23] that only computes the colors of sampled points whose weights are larger than a predefined threshold. This strategy is effective in reducing the computational cost and makes the appearance representation focus on meaningful points. The rendered pixel color $\hat{\mathbf{c}}$ is a

weighted sum of the predicted colors:

$$\hat{\mathbf{c}} = \sum_{i=0}^{N-1} w_i \mathbf{c}_i. \tag{4.11}$$

4.3.4 Training loss

The training loss of the proposed method consists of the mean squared error of the rendered pixel value, a regularization term about the predicted surface normals [21], and a regularization term regarding the density features [23]. Mathematically, the training loss is written as:

$$\mathcal{L} = (\hat{\mathbf{c}} - \mathbf{c}_{gt}) + \alpha \frac{1}{N} \sum_{i=0}^{N-1} w_i \max(0, \mathbf{d} \cdot \mathbf{n}_i)^2 + \beta \frac{1}{M} \sum_{i=0}^{M-1} |\mathbf{F}_\sigma^i| \tag{4.12}$$

where $\mathbf{c}_{gt}$ is the ground truth color, N represents the number of samples in the cast ray, and M is the number of features in the density field representation. The normal regularization term, i.e., the second term in the above equation, penalizes the densities which decrease along the ray. In other words, it encourages concentrated modeling of the scene surface. The third term is density regularization defined as the mean absolute value of all features, which encourages a sparse density field. α, β are loss weights to balance the impact of the two regularization terms, and we empirically use $\alpha = 0.3$ and $\beta = 0.0004$ for all experiments as in [21], [23].

4.4 Experiments

We implement the proposed method using PyTorch [124]. There are a total of 16 levels starting with a base resolution of 16 and growing to a maximum resolution of 512. The number of feature channels is 4 for the appearance field and 2 for the density field. The sizes of the bottleneck $\mathbf{b}$ and feature vector $\mathbf{a}_i$ are 128 and 2, respectively. The spatial MLP has 3 layers, while the directional one has 6 layers. All layers contain 256 hidden units and ReLU activation. We optimize the proposed model using the Adam algorithm [100] with a learning rate of 2e-3 for the MTDs, and 1e-3 for two MLPs. The learning rates degrade log-linearly to 0.1 times their initial values.

We compare our method with methods based on both neural representations and learnable feature representations. The compared methods based on neural representations include NeRF [7], Mip-NeRF [20], and Ref-NeRF [21], while NSVF [115], DVGO [106], MHE [22], and TensoRF [23] belong to learnable feature representations. We evaluate the rendering quality of these methods using the PSNR, SSIM [102], and LPIPS [91]. Two synthetic datasets, namely the NeRF synthetic [7] and NSVF synthetic [115]

Table 4.1: Objective performance comparison. # denotes the number of learnable parameters. The LPIPS are evaluated using the VGG network, while * means results from the Alex network. Our LPIPS results with 60K training steps evaluated by the Alex network on the three datasets are 0.016, 0.007, and 0.092, respectively.

					NeRF Synthetic [7]		
	Steps	#Features	#MLP	Batch size	PSNR↑	SSIM↑	LPIPS↓
NeRF [7]	300K	N/A	1,191K	4096	31.01	0.947	0.081
Mip-NeRF [20]	1M	N/A	612K	4096	33.09	0.961	0.043
Ref-NeRF [21]	250K	N/A	902K	16384	33.99	0.966	0.038
NSVF [106]	150K	0.32∼3.2M	500K	8192	31.75	0.953	0.047*
DVGO [106]	30K	49M	22K	8192	31.95	0.957	0.053
MHE [22]	30K	12.6M	10K	4096	33.18	-	-
TensoRF [23]	30K	18.6M	36K	4096	33.14	0.963	0.047
Ours	30K	12.8M	549K	4096	34.65	0.975	0.034
Ours	60K	12.8M	549K	4096	**35.02**	**0.977**	**0.031**

		NSVF Synthetic [106]			Tanks & Temples [125]		
	Steps	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NeRF [7]	300K	30.81	0.952	0.043*	25.78	0.864	0.198*
NSVF [106]	150K	35.18	0.979	0.015*	28.48	0.901	0.155*
DVGO [106]	30K	35.08	0.975	0.033	28.41	0.911	0.155
TensoRF [23]	30K	36.52	0.982	0.026	28.56	0.920	0.140
Ours	30K	37.76	0.986	0.019	28.87	0.927	0.127
Ours	60K	**38.25**	**0.988**	**0.017**	**29.05**	**0.931**	**0.119**

datasets, and one real-world Tanks & Temples dataset [125] are used for evaluation. Model details including the number of parameters of learnable features and MLPs, batch size, and training steps are also presented for comparison.

4.4.1 Objective Results

The proposed method significantly outperforms existing state-of-the-art view synthesis approaches as shown in Table 4.1. Over 1 dB improvement in PSNR has been observed on both the NeRF and NSVF synthetic datasets. Pure MLP-based methods are compact in representing a scene but are computationally expensive. Besides, they also require a large number of training steps to converge. For example, Ref-NeRF [21] takes 250K steps to converge when using a large batch size of 16384. Thanks to the proposed MTD and encoding the rendering equation in the feature space, we are able to use 12.8M learnable features, which is similar to that in MHE [22] and fewer than those in DVGO [106] and TensoRF [23], to achieve significantly better rendering quality than those compared methods. The proposed NRFF also outperforms the compared

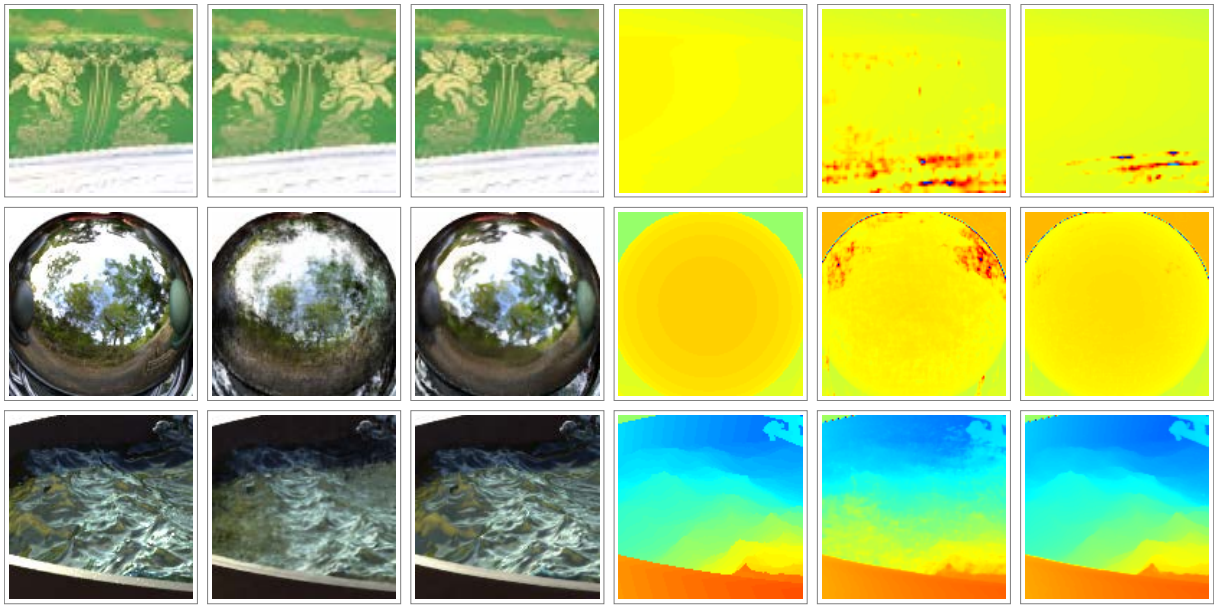


Figure 4.4: Subjective comparison of synthesized views. The left three columns show synthesized novel views and the right three columns are their corresponding depth maps. Our method recovers more accurate texture, specular surface, and geometry than TensoRF [23]. The scenes from top to bottom are *chair*, *materials*, and *ship* from the NeRF synthetic dataset [7].

methods on the Tanks & Temples dataset, demonstrating the efficacy of our method in representing real-world scenes.

4.4.2 Subjective Results

Subjective comparisons are presented in Fig. 4.4 to show that our method is able to recover accurate texture, specular surface, and geometry. For fair comparison, we use the results from TensoRF with decreased features (as detailed in the ablation study in Section 4.4.3) such that the model has a similar number of parameters in the learnable features and the MLP as ours. The comparison on scene *chair* in Fig. 4.4 shows that our method synthesizes sharper texture than TensoRF. This advantage stems from the high-resolution representation in our method, which provides rich local details for view rendering. The rendered balls in the scene *materials* demonstrate the superiority of our rendering equation encoding method in modeling the specular surface compared with the position encoding of view directions employed in TensoRF [23]. Finally, our multiscale representation enables more accurate geometry reconstruction as shown in the depth map of the scene *ship*, resulting in more realistic view synthesis of the water surface. Besides, it is observed from Fig. 4.5 that our model yields a diverse of ASG functions to encode the rendering equation and reconstructs accurate light fields of scenes.

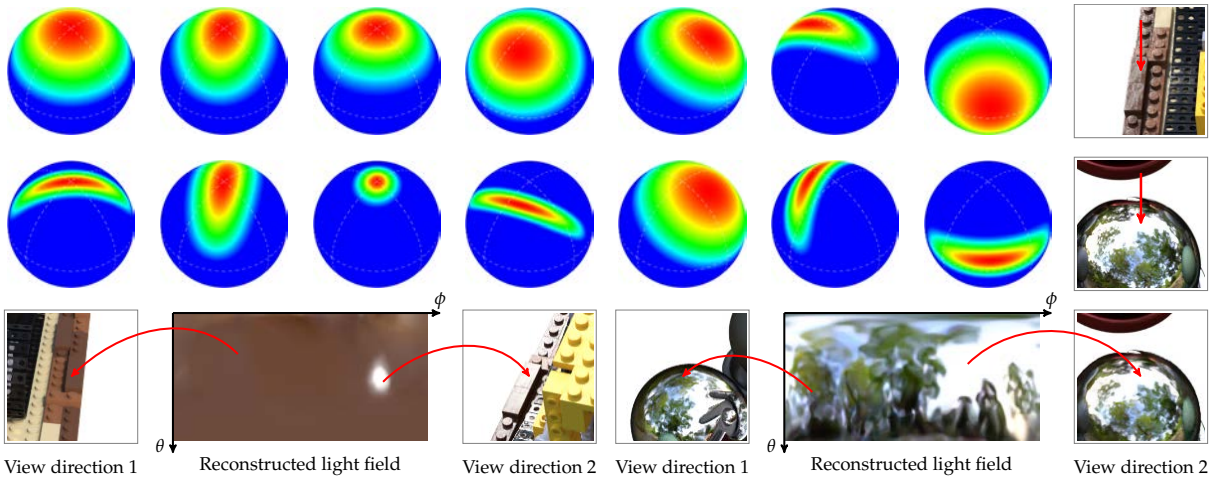


Figure 4.5: Visualization of the learned ASG functions in the feature space and reconstructed light fields in the color space on scenes *lego* and *materials*. The first and second rows show the learned ASG functions used to encode the rendering equation at two points. The two points are on the rays cast from the pixels' position indicated by the red arrows in the rightmost image patches in the first two rows. The two points have their independent and diverse ASG functions. Our model seems to produce more complex ASG functions on the specular surface (second row) to model the complex reflections. The reconstructed light fields and rendered images at different view directions imply successful modeling of complex view-dependent effects.

4.4.3 Ablation Study

We investigate the effectiveness of the proposed modules in Table 4.2. We start with the single-scale TensoRF [23] trained by 30K steps. Other reported results in this table are from models trained by 60K steps. Simply increasing the MLP's size in TensoRF to 10 layers greatly improves the rendering quality. This phenomenon highlights that both the learnable features and MLP are important for improving the rendering quality. When we decrease the number of learnable features in TensoRF to the same level as in our model, there is a small performance degradation (around 0.1 dB). As our encoding method produces a comparable larger encoding vector, for fair comparison, our models use MLPs with 9 layers to make the MLPs' parameters roughly consistent or fewer than that in TensorRF using 10 layers. Our multiscale representation using the position encoding (PE) to encode view directions achieves better rendering quality than the single-scale TensoRF, even with fewer learnable features and a smaller MLP.

Further quality improvement is observed when using our multiscale representation in conjunction with the proposed rendering equation encoding method (i.e., ours, full). As can be observed from Table 4.2 that our full model improves the PSNR from 34.58 dB (ours, multiscale, PE) to 35.02 dB, yielding the state-of-the-art rendering quality. We also experiment on the integrated directional encoding (IDE) [21] using our multiscale

Table 4.2: Ablation study on the NeRF synthetic dataset [7]. Layers indicate the number of MLP layers.

	#Features	#MLP	Layers	PSNR↑	SSIM↑	LPIPS↓
TensoRF [23]	18.6M	36K	4	33.14	0.963	0.047
TensoRF, large MLP	18.6M	568K	10	34.10	0.970	0.038
TensoRF, decrease features	13.4M	557K	10	33.99	0.969	0.039
Ours, multiscale, PE	12.8M	515K	9	34.58	0.975	0.034
Ours, multiscale, IDE	12.8M	532K	9	34.61	0.974	0.034
Ours, multiscale, color	12.8M	545K	9	33.53	0.965	0.043
Ours, full	12.8M	549K	9	**35.02**	**0.977**	**0.031**

representation as the input coordinate encoding instead of integrated positional encoding in mip-NeRF [20]. We do not observe a significant performance improvement when using the IDE method. The model (ours, multiscale, color) using the same form of the rendering equation encoding but in the color space performs poorly. This verifies the drawbacks of encoding the rendering equation in the color space, as discussed in Section 4.3.2.

A detailed performance evaluation over training steps for varying the number of scale levels in Fig. 4.6 demonstrates the benefits of the proposed MTD scheme. For each setup, we adjust the number of feature channels and the maximum resolution to keep roughly the same number of parameters (12.8M) in the learnable features. As shown in Fig. 4.6, the model with two levels trained by 30K steps already surpasses that with one level trained by 60K steps, suggesting faster convergence speed of the multiscale representation than its single-scale counterpart. 1 dB improvement of the final PSNR is observed (from 31.3 dB with $L = 1$ to 32.3 dB with $L = 2$) when we have the two-level representation. Nearly 1 dB additional performance gain (from 32.3 dB with $L = 2$ to 33.2 dB with $L = 16$) becomes observable when increasing the number of levels to 16.

4.4.4 Limitations

Our method use a comparable large MLP than popular methods [22], [23] with learnable features. Representations with more scale levels introduce extra computations for interpolation weights compared with single-scale representation. On the NeRF synthetic dataset, training takes 3∼4 hours for each scene on one Nvidia Tesla V100 with 32 GB memory, and rendering an image of resolution 800×800 requires 3∼4 seconds. The speed of the proposed method is slower than fast methods [22], [106], but faster than pure MLP methods [7], [20], [21]. We believe thorough optimization could overcome this limitation to some extent, considering that the hash encoding in [22] is fast

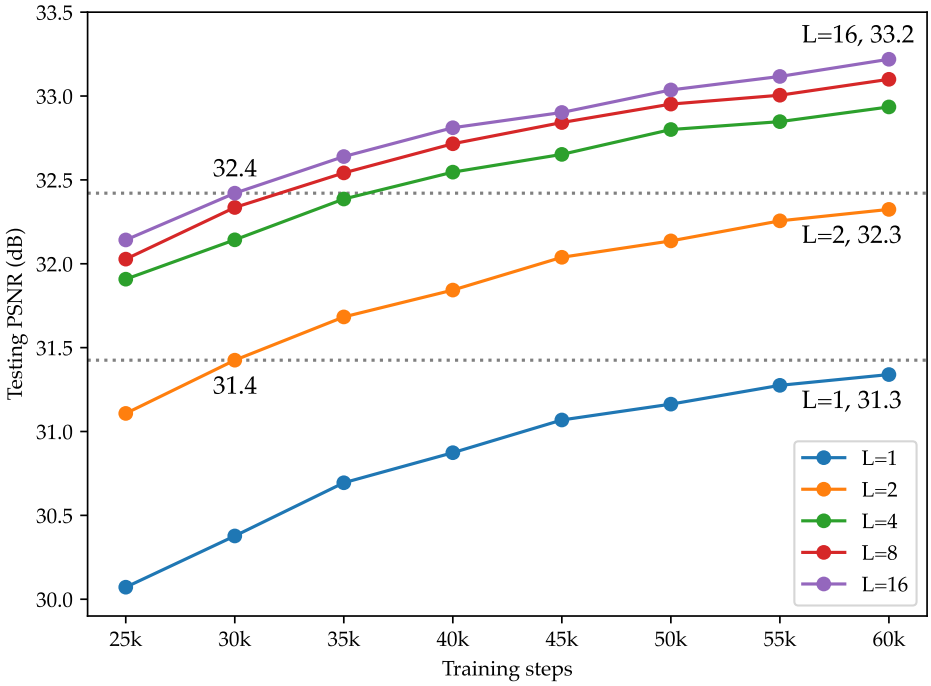


Figure 4.6: Performance comparison over training steps for varying the number of scale levels on scene *ship* from [7]. L indicates the number of levels. All models with different levels have roughly the same number of learnable features. Models with more levels not only converge faster but yield better final PSNSs.

thanks to the highly efficient implementation even with trilinear interpolation. Besides, in the testing stage, the plane feature maps could also be loaded to GPU texture memory to leverage hardware accelerated bilinear interpolation to fetch features more efficiently.

In addition to implementation engineering, reducing the number of samples per ray is a straightforward approach to improve the training and rendering speeds. This reduction can be achieved by importance sampling with the aid of a rough density distribution along the ray inferred by a fast guided network. Ideally, one sample per ray is possible during rendering using extracted geometry, e.g., depths or pointclouds, when we have accurate surface reconstruction. Currently, most neural rendering methods render each pixel independently. Exploring the spatial redundancy in rendered images to reduce the number of pixels needed to be rendered, for example, by spatial super-resolution with the help of the information provided in neural rendering, would also be a promising direction towards real-time rendering.

4.5 Conclusion

We proposed the novel neural radiance feature field (NRFF) to achieve photo-realistic view synthesis. The proposed multiscale tensor decomposition scheme represents scenes from coarse to fine scales, leading to faster convergence and a better rendering quality than the single-scale tensor decomposition. Our proposed rendering equation encoding in the feature space provides more knowledge about the outgoing radiance to the MLP and overcomes the limitations of encoding the rendering equation in the color space. Extensive experimental results were presented to demonstrate the efficacy of the proposed NRFF on both the synthetic and real-world datasets.

Chapter 5

Conclusion

This thesis presented elaborately designed and learning-based algorithms to address light field reconstructions at different levels. For a computer vision problem such as the studied light field depth estimation, one may choose to use deep learning without more consideration. We showed that the simple yet effective vote cost based on careful analysis of the light field image is able to achieve better or comparable performance compared with learning-based methods. Most computer vision problems, however, cannot be solved by a deduced non-learning method. But it is no doubt that priors and knowledge in the studied areas are the design guideline for learning-based algorithms. The proposed learning-based algorithms including both feed-forward prediction and per-scene optimization approaches followed this rule. The idea that the information between the input existing views and the target novel views could be better utilized led to the proposed IR-VAE, as presented in Chapter 3. In Chapter 4, we saw the success of encoding the rendering equation in the feature space, where the rendering equation is the widely used knowledge to rendering photo-realistic image [24]. Significant progress has been made in neural rendering from different angles by leveraging the priors and knowledge in computer graphics [20], [21], [110], [114], [116], [118], [126]. I believe this trend will continue and more innovative algorithms that merge the knowledge in computer graphics and neural rendering will appear to reconstruct real-world light fields.

The multiscale scheme was demonstrated to be effective in light field reconstruction. In the IR-VAE in Chapter 3, the multiscale encoder enables large context and depth perception in the stacked input views, which are important to produce a good underlying representation for the studied position-aware (in terms of pixels) reconstruction problem. Compared with single-scale representation, many benefits of multiscale representations have also been observed in the proposed neural radiance feature field in Chapter 4, including faster convergence, more accurate scene geometry, and texture reconstructions, even with fewer parameters. More broadly, the multiscale scheme has also shown to be effective in many computer vision problems [88], [95], [107], [109]. This thesis indicates that the multiscale scheme is an important backbone

architecture when dealing with light field reconstruction.

Scene representation is the core consideration for global light field reconstruction using per-scene optimization. Pure MLP-based representation is compact but requires large computational resources [7], [20], [21]. Representations using only learnable features are able to achieve similar rendering quality compared with pure MLP-based ones but run significantly faster [105]. Many works combine learnable features and small MLPs to achieve better rendering quality than representations containing only learnable features and keep the speed advantage simultaneously [22], [23], [106]. The proposed neural radiance feature field in Chapter 4 employs learnable features and comparable large MLPs to achieve significantly better rendering quality than existing solutions, highlighting the importance of both learnable features and MLPs. In future research, novel representations or data structures to organize learnable features are desired to tackle challenges in unbounded and dynamic scenes. Besides, explicit representations, e.g., meshes, that are compatible with the rendering pipeline in computer graphics to enable editing and relighting will boost practical applications. At the current stage, directly integrating mesh representation into the optimization pipeline faces challenges as complex topology optimization is not efficient. Extracting meshes from the optimized representation, e.g., density field, shows a performance degradation in rendering quality. Novel representations dealing with this problem will connect the domain knowledge in neural rendering and computer graphics to facilitate the creation of immersive content of real-world scenes.

Future works following this thesis lay emphasis on further improvements of light field reconstruction quality and speed, and extensions to other related fields. Integrating the proposed vote cost into a deep learning framework could utilize advantages of both approaches to obtain better depth estimates. It will be interesting to extend the idea of the vote cost to other similar problems. For example, we can vote to determine the best surface normal estimates with the input of many varying distant illuminations in photometric stereo [127]. We can extend the proposed IR-VAE using more geometric knowledge, e.g., volume rendering technique, and a global representation to enable global view renderings. For the studied NRFF, reducing the number of samples per ray, ideally one sample each ray, will improve both training and rendering speeds. This could be accomplished by importance sampling with the aid of a guided density field or back-projecting extracted geometric proxies, e.g., depths and pointclouds. The proposed rendering equation encoding models outgoing radiance. This approach could be further decomposed to model lighting and materials independently such that more complex effects like global illuminations can be better modeled. Lastly, when dealing with light field video reconstruction, developing a global flow field to warp the current frame representation to the next frame representation will accelerate video

reconstruction speed and facilitate the following video compression.

Although we still face many challenges in creating immersive viewing experience from real-world light fields, I am excited to see that considerable progress has been made in the recent two years. I believe extraordinary applications will appear when the tool of reconstructing light fields becomes more mature and accessible.

Bibliography

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, 1996, pp. 31–42.

[2] A. Tewari, J. Thies, B. Mildenhall, P. Srinivasan, E. Tretschk, W Yifan, C. Lassner, V. Sitzmann, R. Martin-Brualla, S. Lombardi, *et al.*, "Advances in neural rendering," *Computer Graphics Forum*, vol. 41, no. 2, pp. 703–735, 2022.

[3] Y. Tian, W. Song, L. Chen, S. Fong, Y. Sung, and J. Kwak, "A 3D object recognition method from LiDAR point cloud based on usae-bls," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 267–15 277, 2022.

[4] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 784–11 793.

[5] L. Ma, Y. Li, J. Li, W. Tan, Y. Yu, and M. A. Chapman, "Multi-scale point-wise convolutional neural networks for 3D object segmentation from lidar point clouds in large-scale environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 821–836, 2019.

[6] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4460–4470.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021.

[8] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-supervision: Learning dense object descriptors from neural radiance fields," *arXiv preprint arXiv:2203.01913*, 2022.

[9] J. M. Boyce, R. Doré, A. Dziembowski, J. Fleureau, J. Jung, B. Kroon, B. Salahieh, V. K. M. Vadakital, and L. Yu, "MPEG immersive video coding standard," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1521–1536, 2021.

[10] J. Anderson and L. Rainie, "The Metaverse in 2040," *Pew Research Center*, 2022.

[11] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on Metaverse: Fundamentals, security, and privacy," *IEEE Communications Surveys & Tutorials*, 2022.

[12] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 926–954, 2017.

[13] I. K. Park and K. M. Lee, "Robust light field depth estimation using occlusion-noise aware data costs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 10, pp. 2484–2497, 2018.

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

[15] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. S. Kweon, "Depth from a light field image with learning-based matching costs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 297–310, 2019.

[16] B. Mildenhall, P. P. Srinivasan, R. Ortiz-Cayon, N. K. Kalantari, R. Ramamoorthi, R. Ng, and A. Kar, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–14, 2019.

[17] Q. Wang, Z. Wang, K. Genova, P. P. Srinivasan, H. Zhou, J. T. Barron, R. Martin-Brualla, N. Snavely, and T. Funkhouser, "IBRNet: Learning multi-view image-based rendering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4690–4699.

[18] M. Suhail, C. Esteves, L. Sigal, and A. Makadia, "Generalizable patch-based neural rendering," in *European Conference on Computer Vision*, Springer, 2022, pp. 156–174.

[19] A. Chen, Z. Xu, F. Zhao, X. Zhang, F. Xiang, J. Yu, and H. Su, "MVSNeRF: Fast generalizable radiance field reconstruction from multi-view stereo," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 14 124–14 133.

[20] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, and P. P. Srinivasan, "Mip-NeRF: A multiscale representation for anti-aliasing neural radiance fields," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 5855–5864.

[21] D. Verbin, P. Hedman, B. Mildenhall, T. Zickler, J. T. Barron, and P. P. Srinivasan, "Ref-NeRF: Structured view-dependent appearance for neural radiance fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5491–5500.

[22] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, 102:1–102:15, Jul. 2022.

[23] A. Chen, Z. Xu, A. Geiger, J. Yu, and H. Su, "Tensorf: Tensorial radiance fields," *Proceedings of the European Conference on Computer Vision*, 2022.

[24] J. T. Kajiya, "The rendering equation," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, 1986, pp. 143–150.

[25] B. J. Mildenhall, *Neural Scene Representations for View Synthesis*. University of California, Berkeley, 2020.

[26] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, P. Hanrahan, *et al.*, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report*, vol. 2, no. 11, pp. 1–11, 2005.

[27] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi, "Depth from combining defocus and correspondence using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 673–680.

[28] M. W. Tao, P. P. Srinivasan, S. Hadap, S. Rusinkiewicz, J. Malik, and R. Ramamoorthi, "Shape estimation from shading, defocus, and correspondence using light-field angular coherence," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 3, pp. 546–560, 2017.

[29] W Williem and I. Kyu Park, "Robust light field depth estimation for noisy scene with occlusion," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4396–4404.

[30] T.-C. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3487–3495.

[31] H. Zhu, Q. Wang, and J. Yu, "Occlusion-model guided antiocclusion depth estimation in light field," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 965–978, 2017.

[32] J. Chen, J. Hou, Y. Ni, and L.-P. Chau, "Accurate light field depth estimation with superpixel regularization over partially occluded regions," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4889–4900, 2018.

[33] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4d light fields.," in *Conference on Vision, Modeling and Visualization*, 2013, pp. 225–226.

[34] Q. Zhang, L. Xu, and J. Jia, "100+ times faster weighted median filter (wmf)," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2830–2837.

[35] T. E. Bishop and P. Favaro, "The light field camera: Extended depth of field, aliasing, and superresolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 5, pp. 972–986, 2011.

[36] M. W. Tao, P. P. Srinivasan, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi, "Depth from shading, defocus, and correspondence using light-field angular coherence," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1940–1948.

[37] H.-G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y.-W. Tai, and I. So Kweon, "Accurate depth map estimation from a lenslet light field camera," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1547–1555.

[38] Y. Zhang, H. Lv, Y. Liu, H. Wang, X. Wang, Q. Huang, X. Xiang, and Q. Dai, "Light-field depth estimation via epipolar plane image analysis and locally linear embedding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 739–747, 2017.

[39] C. Kim, H. Zimmer, Y. Pritch, A. Sorkine-Hornung, and M. Gross, "Scene reconstruction from high spatio-angular resolution light fields," *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 1–12, 2013.

[40] C. Shin, H.-G. Jeon, Y. Yoon, I. So Kweon, and S. Joo Kim, "EPINet: A fully-convolutional neural network using epipolar geometry for depth from light field images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4748–4757.

[41] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3746–3754.

[42] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke, "Light field intrinsics with a deep encoder-decoder network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9145–9154.

[43] M. Feng, Y. Wang, J. Liu, L. Zhang, H. F. Zaki, and A. Mian, "Benchmark data set and method for depth estimation from light field images," *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3586–3598, 2018.

[44] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2462–2470.

[45] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 5867–5880, 2019.

[46] J. Y. Lee and R.-H. Park, "Depth estimation from light field by accumulating binary maps based on foreground–background separation," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 955–964, 2017.

[47] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4d light fields," in *Asian Conference on Computer Vision*, 2016, pp. 19–34.

[48] *Stanford light field archives*, 2016. [Online]. Available: http://lightfields.stanford.edu/.

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

[50] C.-T. Huang, "Empirical bayesian light-field stereo matching by robust pseudo random field modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 3, pp. 552–565, 2019.

[51] Y.-J. Tsai, Y.-L. Liu, M. Ouhyoung, and Y.-Y. Chuang, "Attention-based view selection networks for light-field disparity estimation.," in *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020, pp. 12 095–12 103.

[52] O. Johannsen, K. Honauer, B. Goldluecke, A. Alperovich, F. Battisti, Y. Bok, M. Brizzi, M. Carli, G. Choe, M. Diebold, *et al.*, "A taxonomy and evaluation of dense light field depth estimation algorithms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 82–99.

[53] HCI, *HCI 4d light field benchmark*, 2016. [Online]. Available: https://lightfield-analysis.uni-konstanz.de.

[54] H. Zhao, D. Gao, M. Wang, and Z. Pan, "Real-time edge-aware weighted median filtering on the gpu," *Computers Graphics*, vol. 61, pp. 11–18, 2016.

[55] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Transactions on Graphics*, vol. 35, no. 6, p. 193, 2016.

[56] N. Meng, X. Wu, J. Liu, and E. Lam, "High-order residual network for light field super-resolution," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 11 757–11 764.

[57] N. Meng, H. K.-H. So, X. Sun, and E. Lam, "High-dimensional dense residual convolutional neural network for light field reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 873–886, 2019.

[58] G. Wu, Y. Liu, Q. Dai, and T. Chai, "Learning sheared epi structure for light field reconstruction," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3261–3273, 2019.

[59] N. Meng, K. Li, J. Liu, and E. Y. Lam, "Light field view synthesis via aperture disparity and warping confidence map," *IEEE Transactions on Image Processing*, vol. 30, pp. 3908–3921, 2021.

[60] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[61] S. A. Eslami, D. J. Rezende, F. Besse, F. Viola, A. S. Morcos, M. Garnelo, A. Ruderman, A. A. Rusu, I. Danihelka, K. Gregor, *et al.*, "Neural scene representation and rendering," *Science*, vol. 360, no. 6394, pp. 1204–1210, 2018.

[62] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "Memc-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 3, pp. 933–948, 2021.

[63] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471.

[64] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive convolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 670–679.

[65] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, and J. Kautz, "Super slomo: High quality estimation of multiple intermediate frames for video interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9000–9008.

[66] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 286–301.

[67] G. Hinton, O. Vinyals, J. Dean, *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

[68] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[69] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[70] J. Shi, X. Jiang, and C. Guillemot, "Learning fused pixel and feature-based view reconstructions for light fields," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2555–2564.

[71] M. S. K. Gul, M. U. Mukati, M. Bätz, S. Forchhammer, and J. Keinert, "Light-field view synthesis using a convolutional block attention module," in *IEEE International Conference on Image Processing (ICIP)*, IEEE, 2021, pp. 3398–3402.

[72] J. Jin, J. Hou, J. Chen, H. Zeng, S. Kwong, and J. Yu, "Deep coarse-to-fine dense light field reconstruction with flexible sampling and geometry-aware fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 4, pp. 1819–1836, 2022.

[73] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. So Kweon, "Learning a deep convolutional network for light-field image super-resolution," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2015, pp. 24–32.

[74] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field reconstruction using deep convolutional network on epi," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6319–6327.

[75] Y. Wang, F. Liu, Z. Wang, G. Hou, Z. Sun, and T. Tan, "End-to-end view synthesis for light field imaging with pseudo 4DCNN," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 333–348.

[76] H. Wing Fung Yeung, J. Hou, J. Chen, Y. Ying Chung, and X. Chen, "Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues," in *Proceedings of the European Conference on Computer Vision*, 2018, pp. 137–152.

[77] Y. Wang, L. Wang, G. Wu, J. Yang, W. An, J. Yu, and Y. Guo, "Disentangling light fields for super-resolution and disparity estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2022.

[78] O. Ivanov, M. Figurnov, and D. Vetrov, "Variational autoencoder with arbitrary conditioning," in *International Conference on Learning Representations*, 2018, pp. 1–12.

[79] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, 2015, pp. 3483–3491.

[80] S. Zhao, J. Song, and S. Ermon, "Infovae: Balancing learning and inference in variational autoencoders," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 5885–5892.

[81] X. Liu, T. Che, Y. Lu, C. Yang, S. Li, and J. You, "AUTO3D: Novel view synthesis through unsupervisely learned variational viewpoint and global 3D representation," *arXiv preprint arXiv:2007.06620*, 2020.

[82] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[83] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.

[84] S. Niklaus and F. Liu, "Context-aware synthesis for video frame interpolation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1701–1710.

[85] Y. Gao and R. Koch, "Parallax view generation for static scenes using parallax-interpolation adaptive separable convolution," in *IEEE International Conference on Multimedia & Expo Workshops*, 2018, pp. 1–4.

[86] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Advances in Neural Information Processing Systems*, 2006, pp. 513–520.

[87] Y. Li, K. Swersky, and R. Zemel, "Generative moment matching networks," in *International Conference on Machine Learning*, 2015, pp. 1718–1727.

[88] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–16, 2020.

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

[90] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proceedings of the European Conference on Computer Vision*, 2016, pp. 694–711.

[91] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[92] Y. Xu, K. Han, Y. Zhou, J. Wu, X. Xie, and W. Xiang, "Deep adaptive blending network for 3D magnetic resonance image denoising," *IEEE Journal of Biomedical and Health Informatics*, pp. 1–11, 2021.

[93] T. Vogels, F. Rousselle, B. McWilliams, G. Röthlin, A. Harvill, D. Adler, M. Meyer, and J. Novák, "Denoising with kernel prediction and asymmetric loss functions," *ACM Transactions on Graphics*, vol. 37, no. 4, pp. 1–15, 2018.

[94] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481.

[95] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.

[96] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 7, pp. 2480–2495, 2021.

[97] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *8th International Conference on Quality of Multimedia Experience*, 2016.

[98] M. Le Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Transactions on Image Processing*, vol. 27, no. 4, pp. 1981–1993, 2018.

[99] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz, "Light field microscopy," *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 924–934, 2006.

[100] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[101] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *Annual Conference on Neural Information Processing Systems*, 2017, pp. 1–4.

[102] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[103] J. Yang, L. Wang, L. Ren, Y. Cao, and Y. Cao, "Light field angular super-resolution based on structure and scene information," *Applied Intelligence*, pp. 1–17, 2022.

[104] G. Wu, Y. Wang, Y. Liu, L. Fang, and T. Chai, "Spatial-angular attention network for light field reconstruction," *IEEE Transactions on Image Processing*, vol. 30, pp. 8999–9013, 2021.

[105] S. Fridovich-Keil, A. Yu, M. Tancik, Q. Chen, B. Recht, and A. Kanazawa, "Plenoxels: Radiance fields without neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5501–5510.

[106] C. Sun, M. Sun, and H.-T. Chen, "Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5459–5469.

[107] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.

[108] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5693–5703.

[109] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[110] J. Hasselgren, N. Hofmann, and J. Munkberg, "Shape, light & material decomposition from images using monte carlo rendering and denoising," *arXiv preprint arXiv:2206.03380*, 2022.

[111] M. Suhail, C. Esteves, L. Sigal, and A. Makadia, "Light field neural rendering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8269–8279.

[112] M. Tancik, P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J. Barron, and R. Ng, "Fourier features let networks learn high frequency functions in low dimensional domains," in *Advances in Neural Information Processing Systems*, 2020, pp. 7537–7547.

[113] P. P. Srinivasan, B. Deng, X. Zhang, M. Tancik, B. Mildenhall, and J. T. Barron, "Nerv: Neural reflectance and visibility fields for relighting and view synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7495–7504.

[114] X. Zhang, P. P. Srinivasan, B. Deng, P. Debevec, W. T. Freeman, and J. T. Barron, "NeRFactor: Neural factorization of shape and reflectance under an unknown illumination," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 6, pp. 1–18, 2021.

[115] L. Liu, J. Gu, K. Zaw Lin, T.-S. Chua, and C. Theobalt, "Neural sparse voxel fields," *Advances in Neural Information Processing Systems*, vol. 33, pp. 15 651–15 663, 2020.

[116] M. Boss, V. Jampani, R. Braun, C. Liu, J. Barron, and H. Lensch, "Neural-PIL: Neural pre-integrated lighting for reflectance decomposition," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 691–10 704, 2021.

[117] M. Boss, R. Braun, V. Jampani, J. T. Barron, C. Liu, and H. Lensch, "NeRD: Neural reflectance decomposition from image collections," in *Proceedings of the IEEE International Conference on Computer Vision*, 2021, pp. 12 684–12 694.

[118] L. Lyu, A. Tewari, T. Leimkühler, M. Habermann, and C. Theobalt, "Neural radiance transfer fields for relightable novel-view synthesis with global illumination," in *European Conference on Computer Vision*, Springer, 2022, pp. 153–169.

[119] K. Xu, W.-L. Sun, Z. Dong, D.-Y. Zhao, R.-D. Wu, and S.-M. Hu, "Anisotropic spherical gaussians," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, pp. 1–11, 2013.

[120] J. Wang, P. Ren, M. Gong, J. Snyder, and B. Guo, "All-frequency rendering of dynamic, spatially-varying reflectance," in *ACM SIGGRAPH Asia*, 2009, pp. 1–10.

[121] R. Ramamoorthi, "Modeling illumination variation with spherical harmonics," *Face Processing: Advanced Modeling Methods*, pp. 385–424, 2006.

[122] E. Haines, "Reflection and refraction formulas," in *Ray Tracing Gems II*, Springer, 2021, pp. 105–108.

[123] D. N. Wood, D. I. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. H. Salesin, and W. Stuetzle, "Surface light fields for 3D photography," in *Proceedings of the 27th Nnnual Conference on Computer Graphics and Interactive Techniques*, 2000, pp. 287–296.

[124] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[125] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples: Benchmarking large-scale scene reconstruction," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–13, 2017.

[126] K. Zhang, F. Luan, Q. Wang, K. Bala, and N. Snavely, "Physg: Inverse rendering with spherical gaussians for physics-based material editing and relighting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5453–5462.

[127] K. Enomoto, M. Waechter, F. Okura, K. N. Kutulakos, and Y. Matsushita, "Discrete search photometric stereo for fast and accurate shape estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.