

**An Integrated Learning Approach for Foveated
Rendering & Novel View Synthesis from Sparse RGB-D
Light Fields**

A thesis

submitted by

TNV VINEET

*in partial fulfilment of the requirements
for the award of the degree of*

BACHELOR OF TECHNOLOGY



**DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.**

MAY 2020

THESIS CERTIFICATE

This is to certify that the thesis entitled “**An Integrated Learning Approach for Foveated Rendering & Novel View Synthesis from Sparse RGB-D Light Fields**”, submitted by **TNV Vineet (EE16b064)**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Mansi Sharma

Project Guide

INSPIRE FACULTY

Dept. of Electrical Engineering

IIT Madras, 600 036

Place: Chennai

Date: 5th June 2020

ACKNOWLEDGEMENTS

Thanks to **Dr. Mansi Sharma** who has guided me through all stages of this project and provided me with computational resources required to complete this work.

ABSTRACT

KEYWORDS: Virtual Reality; Light Fields; Foveated Rendering, View Synthesis; CNN.

Near-eye light field displays provide a solution to visual discomfort when using head mounted displays by presenting accurate depth and focal cues. However, light field HMDs require rendering the scene from a large number of viewpoints. This computational challenge of rendering sharp imagery of the foveal region and reproduce retinal defocus blur that correctly drive accommodation is tackled in this thesis. We designed a novel end-to-end convolutional neural network that leverages human vision to perform both foveated reconstruction and view synthesis using only 1.2% of the total light field data. The proposed architecture comprises of log-polar sampling scheme followed by an interpolation stage and a convolutional neural network. To the best of our knowledge, this is the first attempt that synthesizes the entire light field from sparse RGB-D inputs and simultaneously addresses foveation rendering for computational displays. Our algorithm achieves fidelity in the fovea without any perceptible artifacts in the peripheral regions. The performance in the fovea is comparable to state-of-the-art view synthesis methods, despite using around $10\times$ less light field data.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	i
ABSTRACT	ii
LIST OF TABLES	v
LIST OF FIGURES	vi
ABBREVIATIONS	vii
NOTATION	viii
1 INTRODUCTION	1
2 RELATED WORK	3
2.1 Light Fields & View Synthesis	3
2.2 Foveated Rendering	4
3 PROPOSED SCHEME	5
3.1 Log Polar Sampling	5
3.2 Interpolation	6
3.3 Neural Network Architecture	7
3.3.1 Interleaving layers	9
3.3.2 Residual Network	9
3.3.3 U-Net	9
3.3.4 Losses	10
4 IMPLEMENTATION	11
5 EXPERIMENTS	12
6 COMPARATIVE ANALYSIS	15

LIST OF TABLES

5.1	Results at Near Focus	12
5.2	Results at Far Focus	12
6.1	Comparison of CNN Runtime (in ms)	15
6.2	Results for Additional Scenes	16

LIST OF FIGURES

3.1	Example binary sampling masks. White pixel locations are sampled. (a) Log-buffer scale 2, Gaze location is top left, (b) Log-buffer scale 4, Gaze location is the centre.	6
3.2	Example interpolation. (a) Original image, (b) Sampled image using a binary mask (log-buffer scale 2, gaze location at the centre), (c) Interpolated image.	7
3.3	Our proposed <i>ResUNet</i> network architecture.	8
5.1	Different radial regions at near (a) and far (b) focus. <i>Blue</i> : Boundary of Fovea, <i>Green</i> : Boundary of P1, <i>Red</i> : Boundary of P2.	13
5.2	Qualitative results at near focus. Red dot in the original image shows the gaze location (centre of the head).	13
5.3	Qualitative results at far focus. Red dot in the original image shows the gaze location (on the chair).	14
6.1	Qualitative results for additional scenes. Red dots in the original images point to the gaze locations.	16

ABBREVIATIONS

VR	Virtual Reality
HMD	Head Mounted Display
CNN	Convolutional Neural Network
VAC	Vergence-Accommodation Conflict
RGB-D	Red Blue Green Depth
SSIM	Structural Similarity Index

NOTATION

S	log-polar buffer scale
$P1$	periphery 1 of visual region
$P2$	periphery 2 of visual region

CHAPTER 1

INTRODUCTION

Emerging Virtual Reality (VR) and Augmented Reality (AR) head-mounted displays (HMDs) are revolutionizing many different fields such as gaming, entertainment, medicine, education, etc. Two main features are critical in HMDs to display stunningly vivid virtual objects right in front of the user: 1) reproduce an extended depth-of-focus (eDoF) with sharp imagery over the user's full accommodation range, 2) drive correct accommodation by depicting perceptually accurate retinal defocus blur. However, existing HMDs display virtual objects at a fixed optical focus and do not accurately reproduce retinal blur all over the extended scene. This leads to vergence-accommodation conflict (VAC). Sustained VAC in long hour wearing of HMDs has been associated with potential health concerns caused by biased depth perception and visual fatigue. Computational displays are evolving which integrate optics and rendering algorithms to enhance the capabilities of conventional HMDs by producing correct visual depth and focus cues that makes the entire experience feel natural.

There are three main contenders of computational VR/AR display technologies which receive particular attention: Varifocal, Multifocal, and Light field displays. The Varifocal and Multifocal displays create eDoF. Varifocal HMDs continuously adjust the virtual image distance, whereas Multifocal HMDs generate the multiple focal planes across the viewing zone. These advanced HMDs partially depend on synthetically rendered blur. However, do not account blur created optically due to the natural accommodative response of the user. These displays may produce incorrect focus cues without rendered blur. Recent findings establish that rendered blur is critical to effectively drive accommodation. While promising, synthesizing retinal defocus blur with perceptual accurateness is computationally expensive. In addition, accurate eye tracking is required in HMDs based on synthetically rendered blurring.

Near-eye light field displays provide a solution to these issues while maintaining a thin form factor and optimal field-of-view. Light field based HMDs approximate retinal blur by displaying an optical superposition of many novel viewpoints. However, still

require to render the scene from tens or even hundreds of viewpoints. This is yet another formidable computational challenge.

In this work, we present a novel integrated learning framework for efficient rendering of sparse light fields to reduce the computational cost associated with light field based HMDs. Our proposed *ResUNet* end-to-end convolutional neural network architecture effectively solves the computational tasks associated with sharp foveated reconstruction, reproduce retinal defocus blur to drive natural accommodation and high quality multi-view rendering using modest RGB-D light field inputs. The main contributions of our proposed model are:

- We propose a new method that renders virtual content for near-eye light field displays, contingent on the user gaze, using only a fraction of the total light field data, maintaining acceptable rendering quality.
- The proposed *ResUNet* network architecture is flexible in terms of sampling the patterns and performs foveated reconstruction of the full light field with high foveal fidelity from sparse color image(s) and depth map(s) inputs.

CHAPTER 2

RELATED WORK

In this section, we briefly review light field view synthesis and foveated rendering algorithms designed for VR/AR head mounted displays.

2.1 Light Fields & View Synthesis

Conventional 2D displays fail to provide accurate visual cues for comfortable 3D perception. In human visual perception, Vergence & Accommodation (VA) are neurally coupled which helps maintain a fused sharp image. In the real world, VA is matched to the distance of the object of fixation. As the eye accommodates to the object, natural depth-of-field blur is generated on retina for the rest of the scene. In HMDs, no blur is generated on the retina. The virtual scene appears at different distance, but light comes from the fixed distance of the display. Thus, VA is generally not matched because the eye accommodates to the fixed distance of the display screen, while converging to the varying distance of the virtual object. This VA conflict often causes visual discomfort and nausea. Koulieris et al. (1) results demonstrate that only focus-adjustable-lens combined with gaze-contingent depth-of-field blur successfully drives accommodation to the simulated distance of virtual object. Other conditions alone, such as, depth-of-field rendering and monovision are not very effective.

An alternate solution, 4D light field based displays provide natural viewing experience by presenting virtual objects at the correct focus to match their distance. It allows users to accommodate at different depths while approximating retinal defocus blur. However, rendering content for such a setup requires a large computation cost and typically has high latency. View Synthesis is an approach to reduce the number of rendered views. Wu et al. (2) reconstructs a 9×9 light field using 9 input views, Kalantari et al. (3) use only the four 4 corner views. Srinivasan et al. (4) render the 4D light field by sampling the input central view image. Xiao et al. (5) proposes a network architecture

for generating all the views using 9 or 5 sampled views which produces state-of-the-art results with low latency for real-time applications.

2.2 Foveated Rendering

Humans have a 135° vertical and 160° horizontal field of view. However, the perception of details is not uniform. The highest resolution vision occurs in a 5° central circle (6), where highest concentration of color receptors occur. Acuity quickly falls off radially as the sampling period decreasing roughly linearly with eccentricity, *i.e.*, the angular distance from the centre.

Araujo and Dias (7) estimate the cortical activation using a log-polar mapping, which has been used for GPU rendering of 2D images (8) and 3D images (9). Sun et al (10) proposed a scheme for foveated rendering of light fields using 16% to 30% of light rays while maintaining perceptual quality. *DeepFovea* (11) performed foveated reconstruction of images corrupted by a random binary mask using learned statistics from natural images. All of these approaches require fast gaze-tracking to drive the displays without degradation in visual quality. With recent advances in eye tracking technology, it is now possible to track user gaze in real-time often leveraging learning based techniques.

CHAPTER 3

PROPOSED SCHEME

There are three broad stages in our proposed foveated rendering scheme: sampling, interpolation and a convolutional neural network. At the first stage, the virtual scene is sampled for RGB intensity and depth data according to the log-polar scheme (7). Next second stage, interpolation of the color intensity image is performed for unsampled pixels. Finally, at the last third stage, a novel end-to-end convolutional neural network is employed to predict the full light field data for each color channel separately. Each stage is described in detail in the coming sections:

3.1 Log Polar Sampling

We use log-polar mapping to generate a binary mask for sampling RGB-D pixels (7). A log-buffer of size $R \times \Theta$ is defined first. Then, for each pixel in the buffer, the corresponding pixel in the image is selected to be part of the mask. The x, y pixels in the image corresponding to r, θ in log-buffer are related by

$$x = \exp\left(\frac{r \cdot \log(L)}{R}\right) \cos\left(\frac{2\pi\theta}{\Theta}\right), \quad (3.1)$$

and

$$y = \exp\left(\frac{r \cdot \log(L)}{R}\right) \sin\left(\frac{2\pi\theta}{\Theta}\right) \quad (3.2)$$

where, L denotes the maximum distance from the centre of fovea to the corners of the image. Choosing only integral values of x, y leads to far fewer than $R * \Theta$ sampled pixels in the generated mask. We control the sampling rate using a parameter S dubbed as log-buffer scale. Further, we choose $R = \frac{W}{S}$ and $\Theta = \frac{W}{S}$, where W is defined as the width of the image. Example sampling masks are shown in Fig. 3.1.

Incorporating polynomial kernel functions help in achieving an efficient control over the sampling density (9). In our proposed approach, the actual pixels sampled can be varied while using the same density by selecting offsets r' and θ' . These parameters

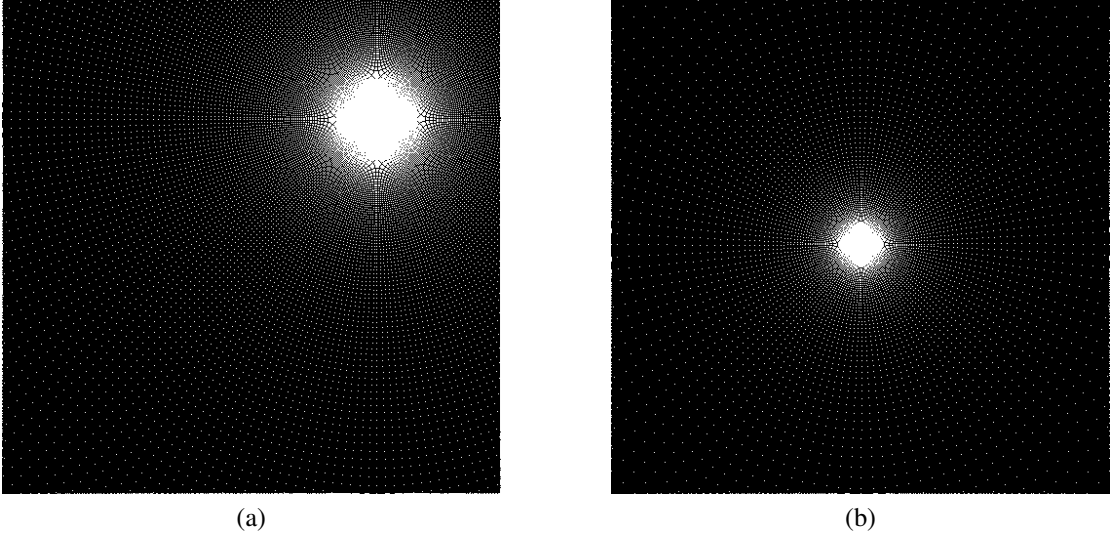


Figure 3.1: Example binary sampling masks. White pixel locations are sampled. (a) Log-buffer scale 2, Gaze location is top left, (b) Log-buffer scale 4, Gaze location is the centre.

are used to obtain different masks for each of the nine input views. The proposed new formulation is defined as

$$x = r' \times \exp\left(\frac{r^{\frac{1}{K}} \cdot \log(L)}{R}\right) \cos\left(\frac{2\pi\theta}{\Theta} + \theta'\right) \quad (3.3)$$

and

$$y = r' \times \exp\left(\frac{r^{\frac{1}{K}} \cdot \log(L)}{R}\right) \sin\left(\frac{2\pi\theta}{\Theta} + \theta'\right) \quad (3.4)$$

where, K is the kernel parameter. The inverse transformation from x, y to r, θ can be determined by

$$r = \left(\frac{\log(\sqrt{x^2 + y^2} \times \frac{1}{r'}) \cdot R}{\log(L)}\right)^K \quad (3.5)$$

$$\theta = \frac{\arctan(\frac{y}{x}) \cdot \Theta}{2\pi} - \theta' \quad (3.6)$$

The interpolation stage is described in the next section.

3.2 Interpolation

Kaplanyan et al. (11) employ Generative Adversarial Networks (GAN) and rely on in-hallucinating the video content based on the learned statistics to achieve foveated compression. However, the GAN model used by *DeepFovea* is not very efficient as it is

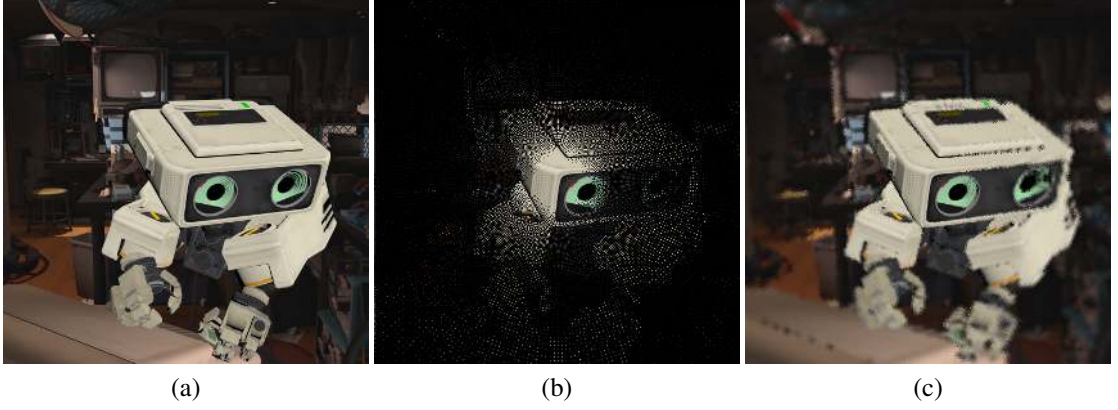


Figure 3.2: Example interpolation. (a) Original image, (b) Sampled image using a binary mask (log-buffer scale 2, gaze location at the centre), (c) Interpolated image.

difficult to train because of issues related to non-convergence and modal collapse.

We adopted a different strategy to avoid perceptual artifacts in the periphery. In our sampling strategy, peripheral regions have very sparse input data to correctly predict the intensity. Due to this nature, we propose an interpolation step for the color channels and simplify processing by formulating the problem as an image enhancement. Inverse distance interpolation using the nearest four pixels is adopted in our scheme. The formula used in the computation is given by

$$p = \frac{\sum_{i=1}^4 \frac{p_i}{d_i}}{\sum_{i=1}^4 \frac{1}{d_i}} \quad (3.7)$$

where, p is the target pixel intensity, p_i is the neighbourhood pixel intensity, and d_i is distance from the target to the neighbourhood pixel. Interpolation strategy works very well, since near the foveal region, there is adequate information to correctly predict the image pixels, whereas in the peripheral region only low frequency content is perceived by our eyes. Our method adequately captures this information. An example of our strategy is shown in Fig. 3.2.

3.3 Neural Network Architecture

In this section, we explain our proposed convolutional network architecture, dubbed as *ResUNet*. A block diagram is illustrated in Fig. 3.3. The objective of proposed *ResUNet* is to strike a balance between quality and runtime complexity, while rendering foveated

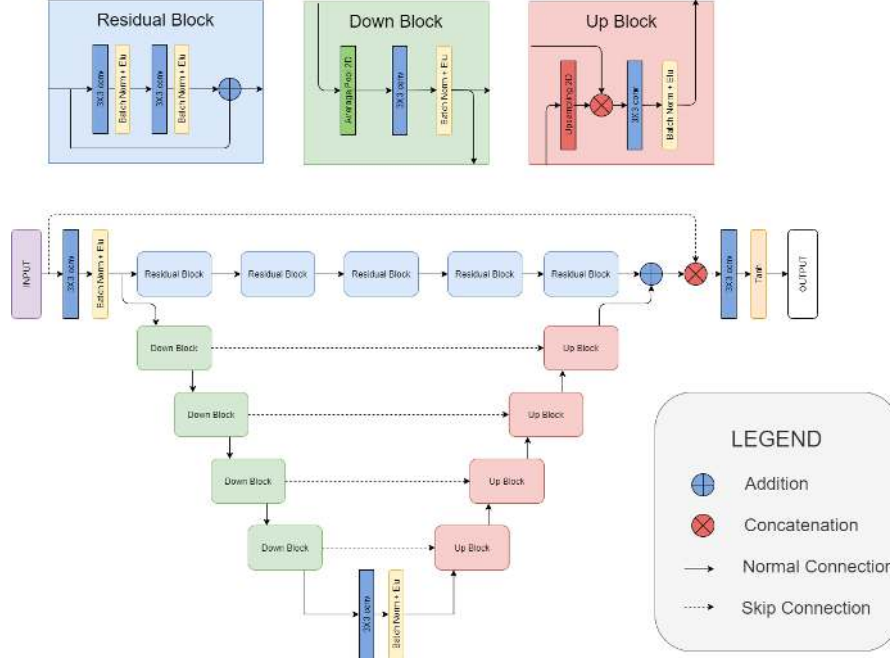


Figure 3.3: Our proposed *ResUNet* network architecture.

contents from input sparse RGB-D light fields. The *ResUNet* is built up of two components: 1) a fully efficient convolutional neural network made up of residual blocks, 2) an encoder-decoder *U-Net* that induces smoothness in the periphery of rendered light field images.

As shown in Fig. 3.3, each convolution layer in proposed *ResUNet* is followed by a batch normalization layer (13) along with an “exponential linear unit” activation function (14). The inputs to the network are sampling mask, sampled depth and interpolated intensity map of a single color image concatenated along the depth axis. Note that each color channel is processed separately with the sampling mask and sampled depth duplicated for red, blue and green channels. Further, the proposed *ResUNet* CNN architecture efficiently learns viewpoint translations. Applying the CNN presented by Xiao et al. (5) separately is not effective since their architecture has a small receptive field which makes it difficult for the network to correctly handle distorted content in the periphery. To overcome this issue, we introduced the *U-Net* in parallel with *Residual-Net* that helps to smooth out the blocky structures from the interpolation step.

3.3.1 Interleaving layers

We include an interleaving layer to reduce the spatial dimensions. This step is motivated by (5). In general, CNNs have a roughly linear relationship between run-time and spatial dimensions. Interleaving layers reshape an input with dimensions (C, H, W) to $(C \times r^2, \frac{H}{r}, \frac{W}{r})$, where r is the interleaving rate. This preserves the volume of the input while increasing the depth. We find using empirical analysis that $r = 2$ increases the performance without much degradation in quality. Besides, a de-interleaving layer in the network restores the outputs to original spatial dimensions of the input. The individual components of our network are described below:

3.3.2 Residual Network

A series of the residual blocks (16) with skip connections are employed in *ResUNet* without downsizing. To preserve the high frequency information of the input image, a long range skip connection from the inputs to the next-to-last layer is present. The residual network allows us to train deeper networks. It is beneficial in learning close to identity mappings with changes such as small translations. Thus, residual network in present view synthesis formulation proves to be much useful. Next, we introduce a *U-Net* architecture (12) to correct the blocky structure introduced in the interpolation step caused by a small receptive field.

3.3.3 U-Net

We proposed a *U-Net* style network that can effectively handle a larger receptive field. The proposed *U-Net* works parallel to the residual block chain. Average pooling and upsampling layers are used to downsize and upsize features respectively. Skip connections from down blocks to up blocks help in better estimating the gradient flow. The input features after the first convolutional block are downsized four times. The output of the proposed *U-Net* is added to the output of the series of residual blocks. This helps in producing the smoothness in the periphery of rendered light field images and reduce the undesired blocky effects.

3.3.4 Losses

The final prediction is done using a tanh activation function. The function is normalized to $[0, 1]$. The loss function used in the proposed model is defined as

$$Loss = \sum_N (-10 \log \|\hat{y} - y\|_2^2 - 10 \log \|\nabla \hat{y} - \nabla y\|_2^2) \quad (3.8)$$

where, \hat{y} is the predicted output and y is the target. The loss is computed by measuring the pixel-wise PSNR and a pixel-wise gradient PSNR components using the predicted output \hat{y} and the target y .

CHAPTER 4

IMPLEMENTATION

The dataset provided by (5) is used for the experiments and training our network. A procedural scene generator Houdini (15) is used to create the dataset. The dataset consists of 85 rendered scenes consisting of light field data with 81 (9×9) intensity and depth images. The RGB-D images used for the experiments is of resolution 512×512 pixels.

A sampling mask as described in section 3.1 is selected for a scene by randomly choosing a location for gaze from a set of 9 predetermined locations. The inputs for the network are generated using the mask by sampling pixels from the RGB-D images. Unsampling pixels are assigned the value 0 by default. The color intensity maps are interpolated by our proposed scheme explained in section 3.2. The robustness of proposed model is improved by training with multiple sampling rates at once. Sampling rate is varied by changing log-buffer scale S as defined in section 3.1. We choose $S = 4$ ($\sim 3.6\%$ pixels) for 40% of scenes, $S = 2$ ($\sim 10.6\%$ pixels) for 40% of scenes and $S = 0$ (100% pixels) for the remaining. The depth and sampling mask is replicated for each color channel. We extract overlapping patches of size 128×128 pixels from each scene to produce 147 data points ($49 \text{ patches} \times 3 \text{ color channels}$). This resulted in a training dataset of size 4,165. The training data size is not affecting our model performance, since the receptive field is smaller than the individual patches. We used 75 scenes for training and the rest for evaluation. Testing is performed on the additional scene provided in the dataset. We trained our model using TensorFlow open-source software library. The weights in our proposed *ResUNet* model are initialized following He et al. (16). We used *Adam* gradient-based stochastic optimization algorithm with the recommended hyperparameters for around 300 epochs choosing a batch size of 16.

CHAPTER 5

EXPERIMENTS

We simulate the retinal image by refocusing the light field image to the gaze position using shift and add algorithm (21). Each of the light fields used for experiments has a spatial resolution of 512×512 . A maximum disparity of 2.5 pixels between adjacent views is considered.

The quality assessment is performed by measuring standard Peak Signal to Noise ratio (PSNR) and Structural Similarity Index (SSIM) metrics on the synthesized foveated images. We estimated PSNR and SSIM measures at three radial regions: Fovea, Periphery 1 (P1) and Periphery 2 (P2). The generated images could be perceived through a near-eye light field display with a 35° field of view. We computed the diameter of fovea as $0.1L$ considering 5° central foveal region (6), where L is the diagonal length of the display. The diameters of P1 and P2 are considered to be $0.4L$ and $1L$. An illustration is provided in Fig. 5.1.

Table 5.1: Results at Near Focus

Model	% of Total Light Field Sampled	SSIM / PSNR (dB)			
		Fovea	P1	P2	Full
DeepFocus	11.11	0.9963 / 45.37	0.9936 / 44.04	0.9961 / 49.49	0.9955 / 47.47
ResUNet (S=2, K=1)	1.18	0.9892 / 41.81	0.9381 / 30.64	0.9585 / 32.98	0.9548 / 32.47
ResUNet (S=1.5, K=1)	1.82	0.9859 / 40.64	0.9407 / 30.83	0.9611 / 33.71	0.9577 / 33.04
ResUNet (S=2, K=3)	2.01	0.9854 / 40.45	0.9382 / 30.27	0.9569 / 33.16	0.9536 / 32.43
ResUNet (S=4, K=1)	0.38	0.9519 / 33.64	0.8368 / 25.08	0.9068 / 25.08	0.8914 / 26.73

Table 5.2: Results at Far Focus

Model	% of Total Light Field Sampled	SSIM / PSNR (dB)			
		Fovea	P1	P2	Full
DeepFocus	11.11	0.9964 / 50.64	0.9969 / 51.02	0.9967 / 50.76	0.9968 / 50.92
ResUNet (S=2, K=1)	1.18	0.9883 / 43.27	0.9269 / 32.33	0.9145 / 28.88	0.9021 / 28.38
ResUNet (S=1.5, K=1)	1.82	0.9917 / 44.88	0.9408 / 33.27	0.9367 / 32.15	0.9274 / 31.49
ResUNet (S=2, K=3)	2.01	0.9891 / 43.59	0.9353 / 32.89	0.9321 / 31.41	0.9206 / 30.56
ResUNet (S=4, K=1)	0.38	0.9376 / 34.11	0.8932 / 30.30	0.8084 / 21.14	0.8081 / 21.35



Figure 5.1: Different radial regions at near (a) and far (b) focus. *Blue*: Boundary of Fovea, *Green*: Boundary of P1, *Red*: Boundary of P2.

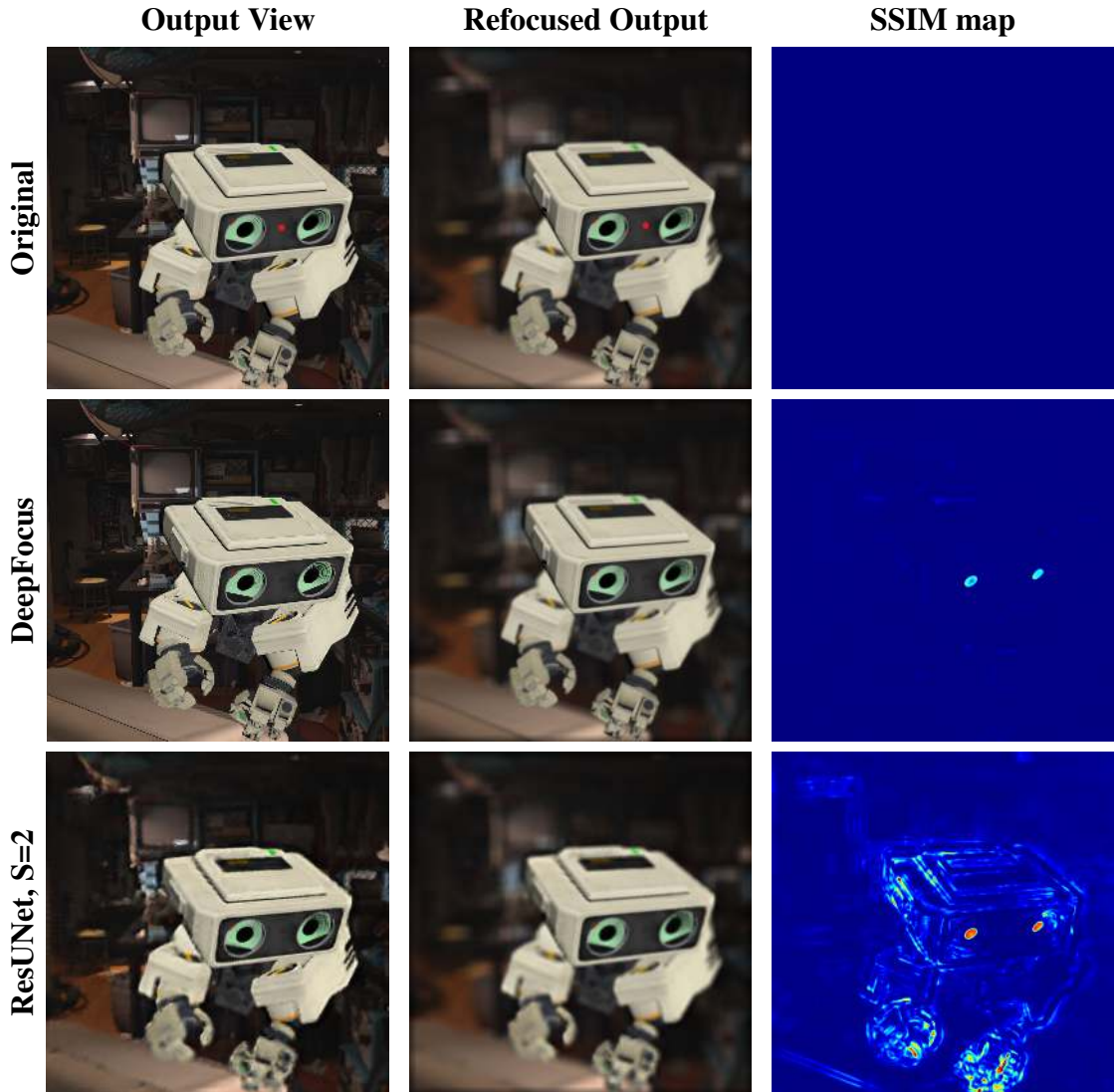


Figure 5.2: Qualitative results at near focus. Red dot in the original image shows the gaze location (centre of the head).

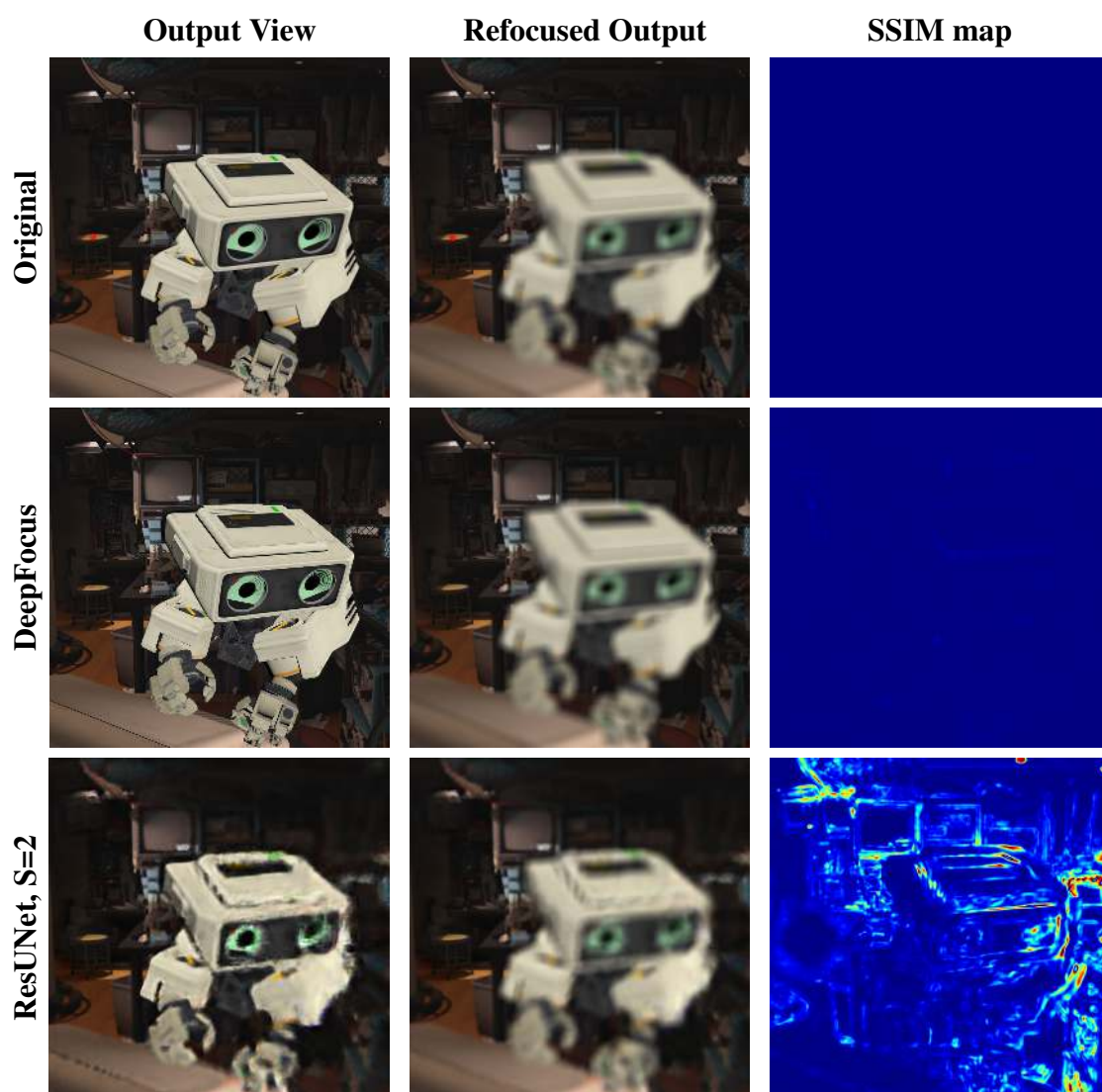


Figure 5.3: Qualitative results at far focus. Red dot in the original image shows the gaze location (on the chair).

CHAPTER 6

COMPARATIVE ANALYSIS

The performance of our proposed scheme is compared with latest work *DeepFocus* (5). Visual results are depicted in Fig. 5.2 and Fig. 5.3. The quantitative results are shown in Table 5.1 and Table 5.2. The results reported here are obtained from a single trained model. The network differs in sampling patterns for the input. At both near and far focus, our model has comparative foveal SSIM score, when per view sampling is greater than 10.6%. The SSIM map in Fig. 5.2 and Fig. 5.3 shows error near the eyes since it is most sensitive at 0 pixel intensity. Most of the errors are visible around the edges. Our model is flexible in terms of log-polar sampling patterns. Quite encouraging results are shown in Fig. 5.2-5.3 at both focal distances with $S = 2$. The reconstruction in P1 and P2 regions is of lower quality compared to the fovea. This is acceptable because of lower visual acuity in these regions.

In Table 6.1, the runtimes of our *ResUNet* and *DeepFocus* (5) CNN are reported. The models performance is tested on Nvidia GTX 1080 processor without TensorRT optimization. Our *ResUNet* runs slightly slower than (5). However, the number of light field rays to be sampled are considerably lower. We would further optimize our network architecture runtime performance by tuning hyperparameters.

Some more results are shown in Fig. 6.1 and reported in Table 6.2 of our model using $S = 2$. These results are computed on additional scenes provided by (19) which consists of color(s) and depth map(s) obtained using Blender’s (20) renderer.

Table 6.1: Comparison of CNN Runtime (in ms)

Resolution	DeepFocus	Ours
512 ²	733.3	1018.6
128 ²	52.5	65.5

Table 6.2: Results for Additional Scenes

Scene	SSIM			
	Fovea	P1	P2	Full
Town	0.9842	0.9298	0.7855	0.8184
Tomb	0.9724	0.9594	0.9591	0.9602
Kitchen	0.9607	0.8465	0.8472	0.8527

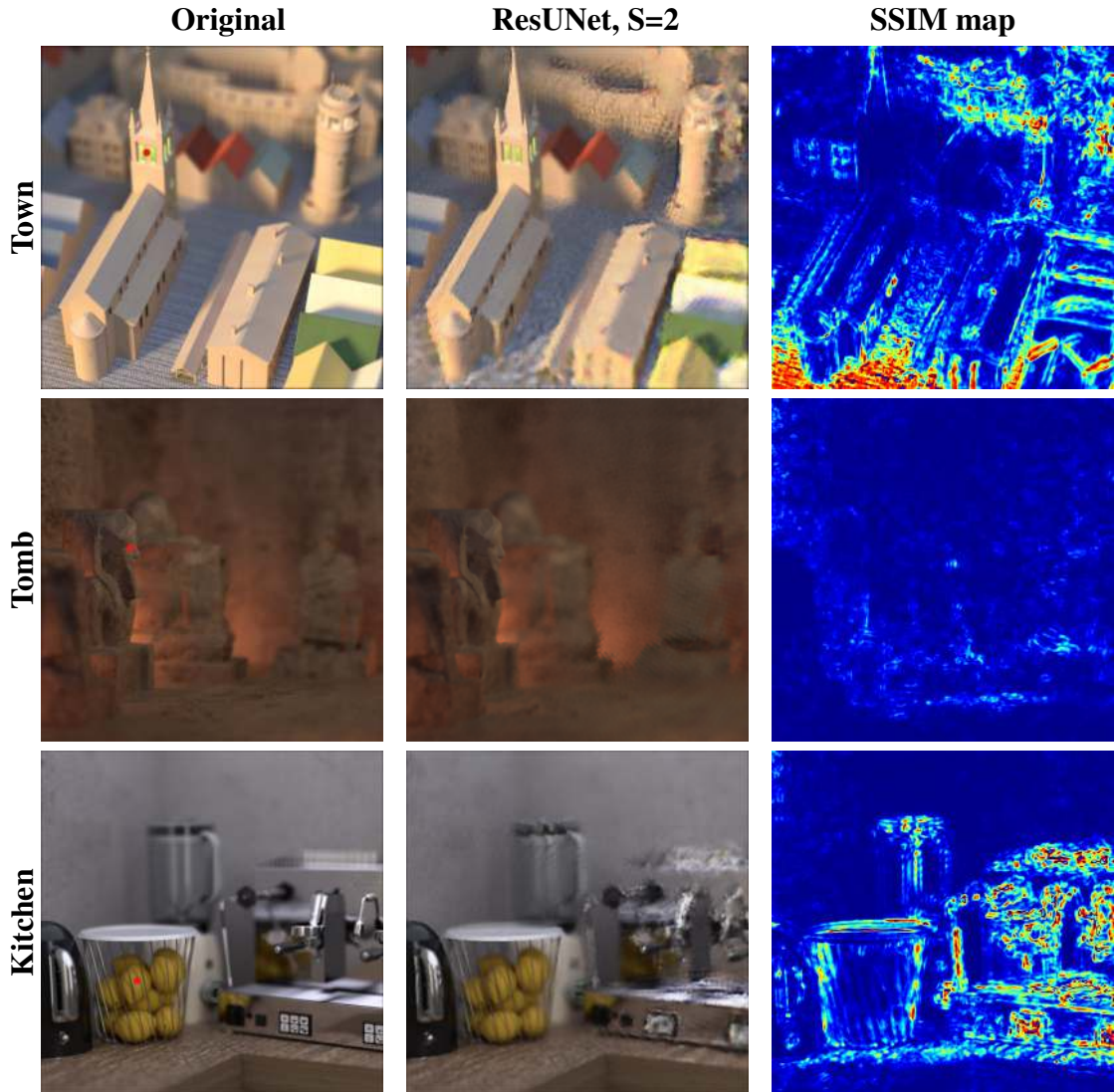


Figure 6.1: Qualitative results for additional scenes. Red dots in the original images point to the gaze locations.

CHAPTER 7

CONCLUSION

In this paper, we have proposed a novel flexible computational scheme that synthesizes realistic light field contents and foveated reconstruction using only sparse RGB-D light fields. The rendered light field images are contingent on the gaze location that leverages human vision. The potential advantage of our model is that it greatly reduces the number of pixels rendered while producing imagery for near-eye light field accommodation-supporting HMDs. We would only need to render approximately 1.2% of the total light fields. Thus, substantially reducing computational burden compared to the state-of-the-art algorithms without compromising with the reconstruction quality. Further, our model is also flexible in terms of sampling patterns. The results demonstrate that reconstruction in the foveal region is of high quality. Simultaneously, it avoids perceptible artifacts in the peripheral regions.

REFERENCES

- [1] George-Alex Koulieris, Bee Bui, Martin S Banks, and George Drettakis, *Accommodation and comfort in head-mounted displays*, ACM Trans. Graph., Vol. 36, No. 4, Article 87, 2017.
- [2] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai and Y. Liu, *Light Field Reconstruction Using Deep Convolutional Network on EPI*, IEEE CVPR, Honolulu, HI, 2017, pp. 1638-1646.
- [3] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi, *Learning-based view synthesis for light field cameras*, ACM Trans. Graph. 35, 6, Article 193, 2016, 10 pages.
- [4] Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng, *Learning to Synthesize a 4D RGBD Light Field from a Single Image*, In Proc. ICCV, 2017, 2262–2270.
- [5] Lei Xiao, Anton Kaplanyan, Alexander Fix, Matthew Chapman, and Douglas Lanman, *DeepFocus: learned image synthesis for computational displays*, ACM Trans. Graph., 37, 6, Article 200, 2018, 13 pages.
- [6] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder, *Foveated 3D graphics*, ACM Trans. Graph. 31, 6, Article 164, 2012, 10 pages.
- [7] H. Araujo and J. M. Dias, *An introduction to the log-polar mapping [image sampling]*, Proceedings II Workshop on Cybernetic Vision, Sao Carlos, Brazil, 1996, pp. 139-144.
- [8] M. Antonelli, F. D. Igual, Ramos, F. Ramos, V. Javier Traver, *Speeding up the log-polar transform with inexpensive parallel hardware: graphics units and multi-core architectures*, J Real-Time Image Proc 10, 533-550, 2015.
- [9] Xiaoxu Meng, Ruofei Du, Matthias Zwicker, and Amitabh Varshney, *Kernel Foveated Rendering*, Proc. ACM Comput. Graph. Interact. Tech. 1, 1, Article 5, 2018, 20 pages.
- [10] Qi Sun, Fu-Chung Huang, Joohwan Kim, Li-Yi Wei, David Luebke, and Arie Kaufman, *Perceptually-guided foveation for light field displays*, ACM Trans. Graph. 36, 6, Article 192, 2017, 13 pages.
- [11] Anton S. Kaplanyan, Anton Sochenov, Thomas Leimkühler, Mikhail Okunev, Todd Goodall, and Gizem Rufo, *DeepFovea: neural reconstruction for foveated rendering and video compression using learned statistics of natural videos*, ACM Trans. Graph. 38, 6, Article 212, 2019, 13 pages.
- [12] O. Ronneberger, P. Fischer, T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Lecture Notes in Computer Science, vol 9351. Springer.

- [13] Sergey Ioffe, Christian Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167, 2015.
- [14] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, *Fast and Accurate Deep Network Learning by Exponential Linear Units*, Conference on Learning Representations, ICLR 2016.
- [15] Houdini Procedural Generator. Side Effects. 1996–2018. <https://www.sidefx.com/>
- [16] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, CVPR, Las Vegas, NV, 2016, pp. 770-778.
- [17] Kingma, Diederik and Ba, Jimmy, *Adam: A Method for Stochastic Optimization*, International Conference on Learning Representations, 2014.
- [18] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli, *Image quality assessment: from error visibility to structural similarity*, IEEE Transactions on Image Processing, 13, 4, 2004, 600-612.
- [19] Katrin Honauer¹, Ole Johannsen, Daniel Kondermann , Bastian Goldluecke, *A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields*, Asian Conference on Computer Vision, 2016.
- [20] Community, B.O., Blender - a 3D modelling and rendering package, Stichting Blender Foundation, Amsterdam, 2018, Available at: <http://www.blender.org>.
- [21] Ren Ng, Digital light field photography, PhD Dissertation, Stanford University, July 2006.

LIST OF PAPERS BASED ON THESIS

1. Vineet Thummuluri, Mansi Sharma, *An Integrated Learning Approach for Foveated Rendering & Novel View Synthesis from Sparse RGB-D Light Fields*, Submitted to the 25th International Conference on Pattern Recognition, 2020.