A Unified Deep Learning Approach for Foveated Rendering & Novel View Synthesis from Sparse RGB-D Light Fields

Conference Paper · December 2020

DOI: 10.1109/IC3D51119.2020.9376340

CITATIONS
2

READS
89

2 authors:

Vineet Thumuluri
Indian Institute of Technology Madras
1 PUBLICATION **2** CITATIONS
SEE PROFILE

Mansi Sharma
Indian Institute of Technology Madras
45 PUBLICATIONS **62** CITATIONS
SEE PROFILE

A UNIFIED DEEP LEARNING APPROACH FOR FOVEATED RENDERING & NOVEL VIEW SYNTHESIS FROM SPARSE RGB-D LIGHT FIELDS

Vineet Thumuluri

Department of Electrical Engineering,
Indian Institute of Technology Madras, India
Email: vineetthumuluri@gmail.com

Mansi Sharma

Department of Electrical Engineering,
Indian Institute of Technology Madras, India
Email: mansisharmaiitm@gmail.com

ABSTRACT

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 drives accommodation is tackled in this paper. 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 fovea is comparable to the state-of-the-art view synthesis methods, despite using around $10\times$ less light field data.

***Index Terms*—** Light field, foveated rendering, view synthesis, RGB-D, 3D/VR/AR/MR, head-mounted display, convolutional neural network.

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 viewer: 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) [1]. 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 head mounted displays [2, 3, 4, 5, 6]. Douglas Lanman and David Luebke [3] designed a light field based thin, lightweight HMD that enables correct convergence, accommodation, binocular disparity, and retinal defocus depth cues. Itoh et al. [2] configure a completely different category of attenuation display with polarized optics to enable both see-through and color interference capabilities. They realize a spatially programmable color filter in the optics that perform color intensity image correction by spatially subtracting background environmental light pixel-wise in the user's see-through view. They aim to display multispectral images in a see-through system. Kim et al. [28] presented a near-eye augmented reality display with resolution and dynamically-foveated focal depth driven by gaze tracking. The display is made up of a traveling microdisplay, which relayed off a concave half-mirror magnifier to produce the high-resolution foveal region. It supports a wide field-of-view peripheral using a projector-based Maxwellian-view display. The viewer's pupil is followed by translating the nodal point of Maxwellian-view display during eye movements by employing a traveling holographic optical element. Their display supports accommodation cues by means of optics, varying the focal depth of the microdisplay in the foveal region. Simultaneously, they render simulated defocus on a scanning laser projector for peripheral region display. In general, their design drives the foveal and peripheral display location using the optics.

To facilitate accommodation and make the entire experience feel natural, there are three main contenders of computational display technologies which receive particular attention in the VR/AR community: Varifocal, Multifocal, and Light field displays [10]. The Varifocal and Multifocal displays create eDoF. Varifocal HMDs continuously adjust the virtual image distance, whereas, Multifocal displays generate multiple focal planes across the viewing zone. These advanced HMDs partially depend on synthetically rendered blur. But, they 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 the 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 requires to render the scene from tens or even hundreds of viewpoints. This is yet another formidable computational challenge.

In this paper, we present a novel unified learning framework for efficient rendering of sparse light fields to reduce the computational cost associated with light field based HMDs. Our proposed end-to-end convolutional neural network architecture *FVS-ResUNet* (Foveated View Synthesis - ResUNet) effectively solves the computational tasks associated with sharp foveated reconstruction, reproduce retinal defocus blur to drive the 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.
- A novel *FVS-ResUNet* network architecture which is flexible in terms of sampling the patterns, handling varying receptive field sizes, and performs foveated reconstruction of the full light field with high foveal fidelity from sparse color image(s) and depth map(s) inputs.

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

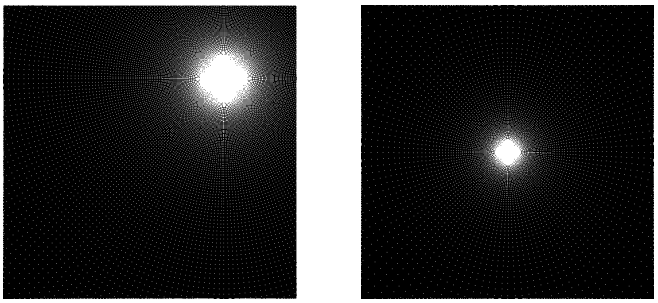
Conventional 2D displays fail to provide accurate visual cues for comfortable 3D perception. In human visual perception, Vergence and Accomodation (VA) are neurally coupled which helps maintain a fused sharp image. In the real world, VA is matched to the distance of the object of fixation. As the eye accommodate to the object, natural depth-of-field blur is generated on the retina for 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. [7] reconstructs a 9×9 light field using nine input views, Kalantari et al. [8] use only four corner views. Srinivasan et al. [9] render the 4D light field by sampling the input central view image. Xiao et al. [10] proposes a network architecture for generating all the views using nine or five sampled views, which produce 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 [11], where the highest concentration of color receptors occurs. Acuity quickly falls off radially as the sampling period decreases roughly linearly with eccentricity, *i.e.*, the angular distance from the centre.

Araujo and Dias [12] estimate the cortical activation using a log-polar mapping, which has been used for GPU rendering of 2D images [13] and 3D images [14]. Sun et al [15] proposed a scheme for foveated rendering of light fields using 16% to 30% of light rays while maintaining perceptual quality. DeepFovea [16] performs foveated reconstruction of images corrupted by a random binary mask using learned statistics from natural images. 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.



(a) (b)

Fig. 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.

(a) (b) (c)

Fig. 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.

3. PROBLEM FORMULATION

Consider N RGB-D light field views $\{I_i, D_i\}, i = 1, \ldots, N$, where $I_1, \ldots, I_N$ and $D_1, \ldots, D_N$ denotes the color images and their corresponding depth maps, respectively. Using a fixed set of sparse input views, each output view is predicted using a view interpolation function. The view synthesis problem can be formulated as

$$\hat{I_o}(p) = F([I_i], [D_i], P) \quad \forall\, o \,\epsilon\, \{1 \ldots N\} \tag{1}$$

where, the subscripts i and o denote input and output views, respectively. $[I_i]$, $[D_i]$ are the concatenation of the fixed input views that represents view sampling, F is the required transformation function modelled using a convolutional neural network with input patch P, whose size depends on the receptive field of the CNN and the output pixel location p.

Xiao et al. [10] shows a *ResNet* [21] type architecture that models view interpolation function F compared to other architectures such as an encoder-decoder *U-Net* [17] in terms of speed and number of parameters. However, the size of patch P in their proposed

CNN is small since it contains only convolutional, but not subsampling layers. This poses a problem when the sparsity is higher in regions of the image away from the gaze location. The function cannot correctly predict the output when sparse input patch does not contain enough information. Our proposed FVS-ResUNet solve this issue by learning to choose between the receptive field sizes based on sampling density and maintain high view synthesis quality. Our approach considers sparsity in the patch P and simultaneously retain the benefits of the *ResNet* [21]. This is the motivation behind our choice of the CNN. Mathematically, we formulate the problem for foveated view synthesis as

$$\hat{I}_o(p) = F_1([I_i], [D_i], S, P_1) + F_2([I_i], [D_i], S, P_2) \quad (2)$$

where, F_1 is a residual block chain with a small sized patch P_1 and F_2 is a set of encoder-decoder blocks with a large sized patch P_2, S is the binary sampling mask required for foveation rendering. Thus, the network can learn to choose between two patch sizes based on the sampling density.

4. PROPOSED FOVEATED RENDERING SCHEME FOR SPARSE LIGHT FIELDS

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 [12]. Next, in the 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 the coming sections:

4.1. Log Polar Sampling

We use log-polar mapping to generate a binary mask for sampling RGB-D pixels [12]. 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

$$x = exp(\frac{r \cdot \log(L)}{R}) \cos(\frac{2\pi\theta}{\Theta}), \quad (3)$$

and

$$y = exp(\frac{r \cdot \log(L)}{R}) \sin(\frac{2\pi\theta}{\Theta}) \quad (4)$$

where, L denotes the maximum distance from the centre of fovea to the corners of the image. Choosing only integral values of x, y lead 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. 1.

Incorporating polynomial kernel functions help in achieving an efficient control over the sampling density [14]. In our proposed approach, the actual pixels sampled can be varied while using the same density by selecting offsets $r^{'}$ and $\theta^{'}$. These parameters are used to obtain different masks for each of the nine input views. The proposed new formulation is defined as

$$x = r^{'} \times exp(\frac{r^{\frac{1}{K}} \cdot \log(L)}{R}) \cos(\frac{2\pi\theta}{\Theta} + \theta^{'}) \quad (5)$$

and

$$y = r^{'} \times exp(\frac{r^{\frac{1}{K}} \cdot \log(L)}{R}) \sin(\frac{2\pi\theta}{\Theta} + \theta^{'}) \quad (6)$$

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

$$r = (\frac{\log(\sqrt{x^2 + y^2} \times \frac{1}{r^{'}}) \cdot R}{log(L)})^K \quad (7)$$

$$\theta = \frac{arctan(\frac{y}{x}) \cdot \Theta}{2\pi} - \theta^{'} \quad (8)$$

The interpolation stage is described in the next section.

4.2. Interpolation

Kaplanyan et al. [16] 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 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

$$p = \frac{\sum_{i=1}^{4} \frac{p_i}{d_i}}{\sum_{i=1}^{4} \frac{1}{d_i}} \quad (9)$$

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. 2.

4.3. FVS-ResUNet: Proposed Neural Network Architecture

In this section, we explain our proposed convolutional network architecture, dubbed as *FVS-ResUNet*. A block diagram is illustrated in Fig. 3. The objective of proposed *FVS-ResUNet* is to strike a balance between quality and runtime complexity, while rendering foveated contents from input sparse RGB-D light fields. The *FVS-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, each convolution layer in proposed *FVS-ResUNet* is followed by a batch normalization layer [18] along with an "exponential linear unit" activation function [19]. 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 *FVS-ResUNet* CNN architecture efficiently learns the viewpoint translations. Applying the CNN presented by Xiao et al. [10] 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.

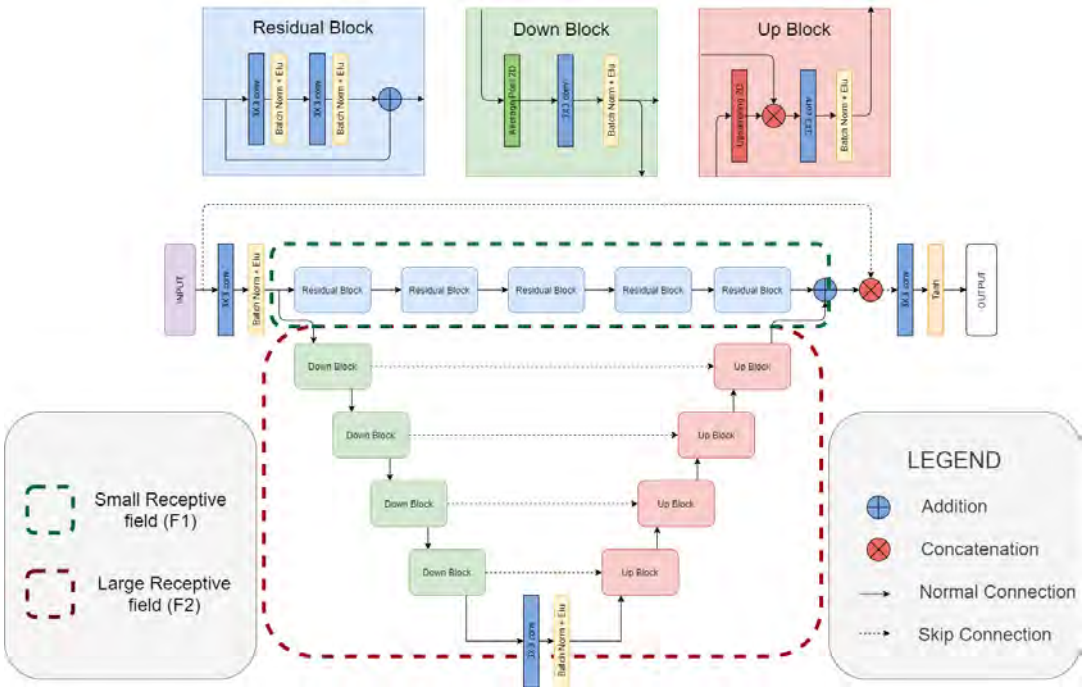


Fig. 3: Our proposed *FVS-ResUNet* network architecture.

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.

4.3.1. Interleaving layers

We include an interleaving layer to reduce the spatial dimensions. In general, CNNs have a roughly linear relationship between run-time and spatial dimensions [10]. 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 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:

4.3.2. Residual Network

A series of the residual blocks [21] with skip connections are employed in *FVS-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 presented. 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 to correct the blocky structure introduced in the interpolation step caused by a small receptive field.

4.3.3. 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. The 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.

It is critical to note that our proposed FVS-ResUNet architecture is different from Diakogiannis et al. [27] ResUNet-a network. Diakogiannis et al. [27] presents a deep learning modeling framework for semantic segmentation of high resolution aerial images. Their ResUNet-a architecture consists of a U-Net backbone with modified residual blocks of convolutional layers used in the place of encoder-decoder blocks. Besides, the multiple parallel atrous convolutions are employed within each residual building block in their ResUNet-a architecture. However, our proposed FVS-ResUNet architecture has a unique configuration made up of residual blocks and a U-Net in parallel. It is specifically designed to address foveated rendering and novel view synthesis using modest RGB-D light fields.

4.3.4. Losses

The final prediction in our FVS-ResUNet 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

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

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

(a) (b)

Fig. 4: Different radial regions at near (a) and far (b) focus. *Blue*: Boundary of Fovea, *Green*: Boundary of P1, *Red*: Boundary of P2.

of ∇y can be found by subtracting the image shifted by one pixel from the original image in the horizontal and vertical axes. Mathematically, we computed ∇y as

$$\nabla y(i,j) = (y(i,j) - y(i-1,j))\hat{i} + (y(i,j) - y(i,j-1))\hat{j}$$

where, (i,j) denotes the $(i,j)^{th}$ pixel location. The $\hat{i}$ and $\hat{j}$ denotes the horizontal and vertical axes, respectively.

5. IMPLEMENTATION

The dataset provided by [10] is used for the experiments and training our network. A procedural scene generator Houdini [20] 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 are of resolution 512×512 pixels.

A sampling mask as described in section 4.1 is selected by randomly choosing a location for gaze from a set of nine predetermined locations. The inputs to the network are generated using the mask by sampling pixels from the RGB-D images. Unsampled pixels are assigned the value 0 by default. The color intensity maps are interpolated by our proposed scheme explained in section 4.2. 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 4.1. 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 patches $\times$ 3 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 additional scene provided in the dataset. We trained our model using the TensorFlow open-source software library. The weights in our proposed *FVS-ResUNet* model are initialized following He et al. [21]. We used *Adam* gradient-based stochastic optimization algorithm [22] with the recommended hyperparameters for around 300 epochs with learning rate 0.0001, choosing a batch size of 16.

6. EXPERIMENTS

We simulate the retinal image by refocusing light field image to the gaze position using shift and add algorithm [26]. The light field

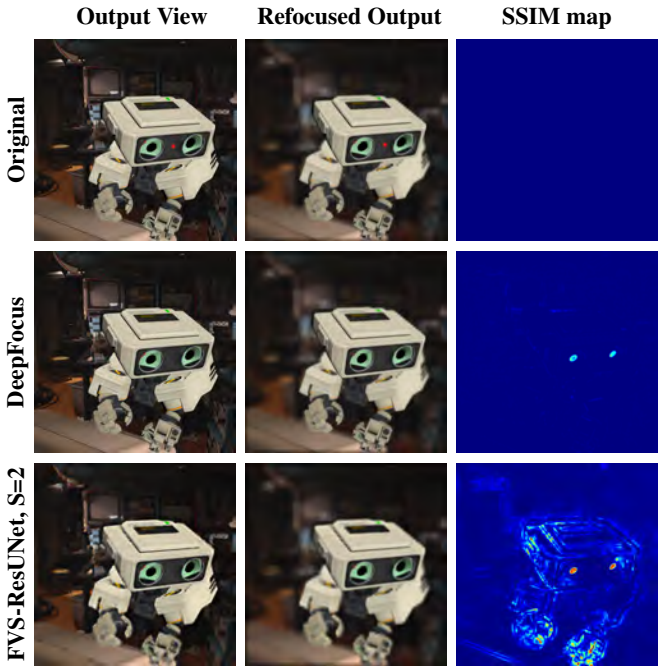


Fig. 5: Qualitative results at the near focus. Red dot in the original image shows gaze location (centre of the head).

views used for experiments have 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 evaluated SSIM for perceptual quality and PSNR for pixel-wise accuracy. 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 [11], 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. 4.

7. RESULTS & ANALYSIS

The performance of our proposed scheme is compared with latest work *DeepFocus* [10]. The visual results are depicted in Fig. 5 and Fig. 6. The quantitative results are shown in Table 1 and Table 4. The results reported here are obtained from a single trained model. The network differs in sampling patterns for the input.

To perform quantitative comparison with DeepFocus [10], we computed results considering 11.11% sampling (9 out of 81 selected light field views; 100% pixels of the selected full views similar to DeepFocus) of light field data, and achieves comparable quality results for the whole image. The critical advantage of our network FVS-ResUNet is that it performs both foveated reconstruction and view synthesis using as low as 0.38-1.18% of the total light field data with acceptable rendering quality. On the other hand, view synthesis methods like DeepFocus [10] are not designed, specifically, for foveated rendering. Therefore, it is not possible to render with DeepFocus [10] network architecture, considering very low sam-

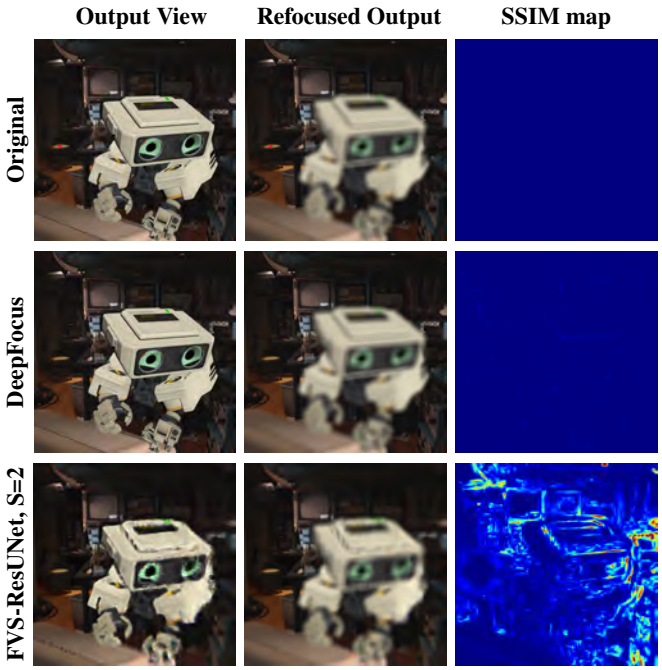


Fig. 6: Qualitative results at the far focus. Red dot in the original image shows gaze location (on the chair).

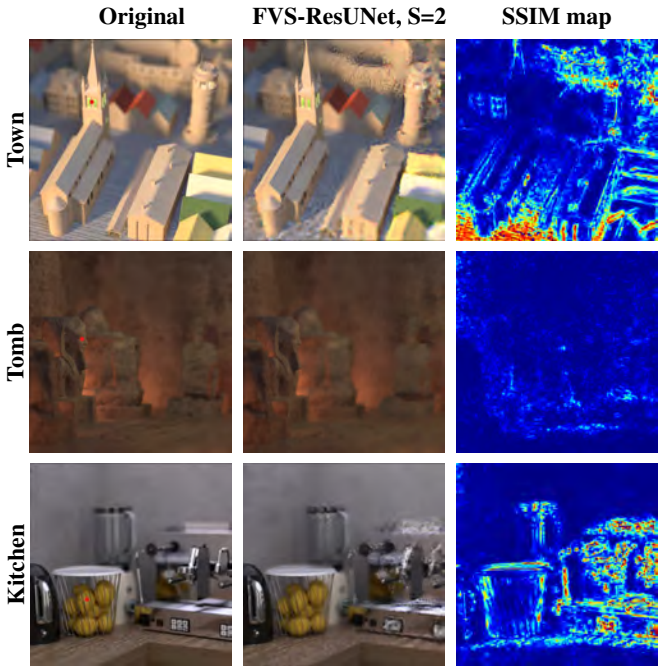


Fig. 7: Qualitative results for additional scenes. Red dots in original images point to the gaze locations.

pling rates. At both near and far focus, our model has comparable foveal SSIM score, when per view sampling is greater than 10.6%. The SSIM map in Fig. 5 and Fig. 6 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. 5-6 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.

The graphs analysing quality vs sampling rate using our proposed FVS-ResUNet architecture is shown in Fig 8. The plots for foveal quality depict that there is no significant drop in quality over various sampling rates. The SSIM and PSNR quality in fovea is high for most sampling rates. However, the overall quality decrease is mainly due to fall-off in quality of the peripheral region. Lower sampling rates result in lower peripheral quality, which is an expected result. Finding a trade-off solution in our settings requires a subjective user study to determine the quality thresholds detectable by human eyes in different regions of vision. This is a part of our future work. Based on our SSIM/PSNR analysis in Fig 8, a sampling rate can be chosen which does not produce any perceivable distortions in the image.

In Table 3, the runtime of our *FVS-ResUNet* CNN for performing both view synthesis and foveation rendering is reported. This is compared with *DeepFocus* [10] CNN runtime for performing view synthesis. The model's performance is tested on Nvidia GTX 1080 processor without TensorRT optimization. Our *FVS-ResUNet* runs slightly slower than [10]. 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. 7 and reported in Table 2 of our model using $S = 2$. These results are computed on additional scenes provided by [24], which consists of color(s) and depth map(s) obtained using the Blender renderer [25].

8. CONCLUSION

In this paper, we have proposed a novel flexible computational scheme that synthesizes realistic light field contents and foveated reconstruction using only modest RGB-D light field data. 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 and handling varying receptive field sizes. The results demonstrate that reconstruction in the foveal region is of high quality. Simultaneously, it avoids perceptible artifacts in the peripheral regions. Since light field facilitates natural defocus blur, and thus approximating full light field using modest inputs by FVS-ResUNet is critical for effectively addressing sharp foveated reconstruction, reproducing retinal defocus blur that drives the natural accommodation and computational tasks associated with high quality multi-view synthesis. We will further explore these aspects in-depth with new light field display tech with mixed reality applications.

9. 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.

Table 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
FVS-ResUNet (S=0, K=1)	11.11	0.9894 / 41.57	0.9819 / 40.704	0.9942 / 46.80	0.9912 / 44.42
FVS-ResUNet (S=0.75, K=1)	4.80	0.9866 / 39.35	0.9692 / 36.78	0.9804 / 38.02	0.9781 / 37.82
FVS-ResUNet (S=1, K=1)	3.28	0.9833 / 38.94	0.9603 / 34.86	0.9726 / 35.94	0.9705 / 35.86
FVS-ResUNet (S=2, K=3)	2.01	0.9854 / 40.45	0.9382 / 30.27	0.9569 / 33.16	0.9536 / 32.43
FVS-ResUNet (S=1.5, K=1)	1.82	0.9859 / 40.64	0.9407 / 30.83	0.9611 / 33.71	0.9577 / 33.04
FVS-ResUNet (S=2, K=1)	1.18	0.9892 / 41.81	0.9381 / 30.64	0.9585 / 32.98	0.9548 / 32.47
FVS-ResUNet (S=4, K=1)	0.38	0.9519 / 33.64	0.8368 / 25.08	0.9068 / 25.08	0.8914 / 26.73

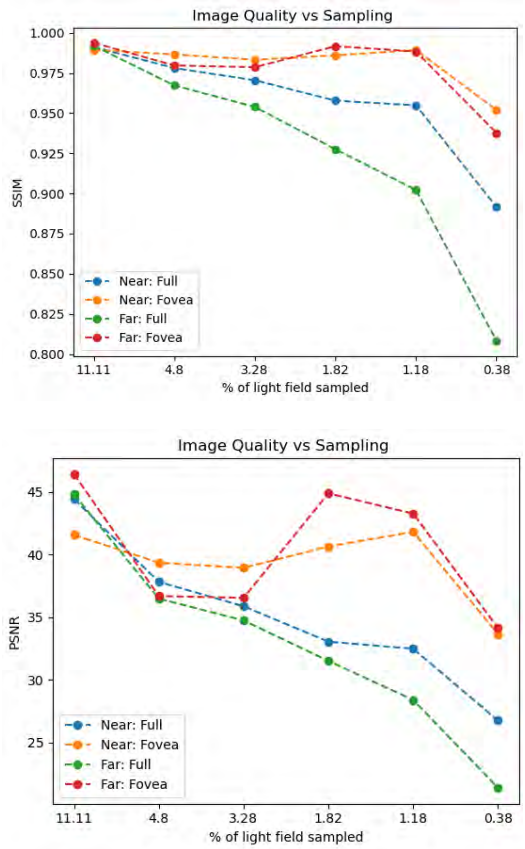


Fig. 8: Graphs showing trend between sampling rate and quality.

Table 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

Table 3: Comparison of CNN Runtime (in ms)

Resolution	DeepFocus (View Synthesis)	Ours (View Synthesis + Foveation Rendering)
512^2	733.3	1018.6
128^2	52.5	65.5

[2] Y. Itoh, T. Langlotz, D. Iwai, K. Kiyokawa and T. Amano, *Light Attenuation Display: Subtractive See-Through Near-Eye Display via Spatial Color Filtering*, in IEEE Transactions on Visualization and Computer Graphics, vol. 25, no. 5, pp. 1951-1960, May 2019.

[3] Douglas Lanman and David Luebke, *Near-eye light field displays*, ACM Trans. Graph. 32, 6, Article 220, 10 pages, 2013.

[4] Jen-Hao Rick Chang, B. V. K. Vijaya Kumar, and Aswin C. Sankaranarayanan, *Towards multifocal displays with dense focal stacks*, ACM Trans. Graph. 37, 6, Article 198, 13 pages, 2018.

[5] Kaan Akşit, Ward Lopes, Jonghyun Kim, Peter Shirley, and David Luebke, *Near-eye varifocal augmented reality display using see-through screens*, ACM Trans. Graph. 36, 6, Article 189, 13 pages, 2017.

[6] Olivier Mercier, Yusufu Sulai, Kevin Mackenzie, Marina Zannoli, James Hillis, Derek Nowrouzezahrai, and Douglas Lanman, *Fast gaze-contingent optimal decompositions for multifocal displays*, ACM Trans. Graph. 36, 6, Article 237, 15 pages, 2017.

[7] 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.

Table 4: 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
FVS-ResUNet (S=0, K=1)	11.11	0.9938 / 46.43	0.9915 / 43.92	0.9916 / 44.97	0.9920 / 44.81
FVS-ResUNet (S=0.75, K=1)	4.80	0.9797 / 36.68	0.9795 / 38.19	0.9711 / 37.47	0.9673 / 36.48
FVS-ResUNet (S=1, K=1)	3.28	0.9786 / 36.53	0.9719 / 37.35	0.9576 / 35.12	0.9539 / 34.73
FVS-ResUNet (S=2, K=3)	2.01	0.9891 / 43.59	0.9353 / 32.89	0.9321 / 31.41	0.9206 / 30.56
FVS-ResUNet (S=1.5, K=1)	1.82	0.9917 / 44.88	0.9408 / 33.27	0.9367 / 32.15	0.9274 / 31.49
FVS-ResUNet (S=2, K=1)	1.18	0.9883 / 43.27	0.9269 / 32.33	0.9145 / 28.88	0.9021 / 28.38
FVS-ResUNet (S=4, K=1)	0.38	0.9376 / 34.11	0.8932 / 30.30	0.8084 / 21.14	0.8081 / 21.35

[8] 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.

[9] Pratul P. Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng, *Learning to Synthesize a 4D RGBD Light Field from a Single Image*, ICCV 2017, pp. 2262–2270.

[10] 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.

[11] Brian Guenter, Mark Finch, Steven Drucker, Desney Tan, and John Snyder, *Foveated 3D graphics*, ACM Trans. Graph. 31, 6, Article 164, 2012, 10 pages.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] Sergey Ioffe, Christian Szegedy, *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*, arXiv:1502.03167, 2015.

[19] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter, *Fast and Accurate Deep Network Learning by Exponential Linear Units*, Conference on Learning Representations, ICLR 2016.

[20] Houdini Procedural Generator. Side Effects. 1996–2018. `https://www.sidefx.com/`

[21] K. He, X. Zhang, S. Ren and J. Sun, *Deep Residual Learning for Image Recognition*, CVPR, Las Vegas, NV, 2016, pp. 770-778.

[22] Diederik P. Kingma and Jimmy Ba, *Adam: A Method for Stochastic Optimization*, International Conference on Learning Representations, 2014.

[23] 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.

[24] 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, Lecture Notes in Computer Science, vol 10113. Springer.

[25] Blender Online Community, Blender - a 3D modelling and rendering package, Stichting Blender Foundation, Amsterdam, 2018, Available at: `http://www.blender.org`.

[26] Ren Ng, Digital light field photography, *Ph.D. Dissertation*, Stanford University, July 2006.

[27] Foivos I. Diakogiann, François Waldner, Peter Caccetta, Chen Wu, *ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data*, ISPRS Journal of Photogrammetry and Remote Sensing, Volume 162, April 2020, Pages 94-114.

[28] Jonghyun Kim, Youngmo Jeong, Michael Stengel, Kaan Akşit, Rachel Albert, Ben Boudaoud, Trey Greer, Joohwan Kim, Ward Lopes, Zander Majercik, Peter Shirley, Josef Spjut, Morgan McGuire, and David Luebke, *Foveated AR: dynamically-foveated augmented reality display*, ACM Trans. Graph., 38, 4, Article 99, 2019, 15 pages.