An Efficient Refocusing Scheme for Camera-Array Captured Light Field Video for Improved Visual Immersiveness

Nusrat Mehajabin
University of British Columbia
nusratm@ece.ubc.ca

Peizhi Yan
University of British Columbia
yanpz@ece.ubc.ca

Supreet Kaur
University of British Columbia
supreet4@ece.ubc.ca

Jingxiang Song
University of British Columbia
sjxca2021@ece.ubc.ca

Mahsa T. Pourazad
University of British Columbia
TELUS Communications Inc.
pourazad@ece.ubc.ca

Yixiao Wang
University of British Columbia
yixiaow@ece.ubc.ca

Hamidreza Tohidypour
University of British Columbia
htohidyp@ece.ubc.ca

Panos Nasiopoulos
University of British Columbia
panosn@ece.ubc.ca

Abstract

Light field video technology attempts to acquire human-like visual data, offering unprecedented immersiveness and a viable path for producing high-quality VR content. Refocusing that is one of the key properties of light field and a must for mixed reality applications has shown to work well for microlens based cameras, but as light field videos acquired by camera arrays have a low angular resolution, the refocused quality suffers. In this paper, we present an approach to improve the visual quality of refocused content captured by a camera array-based setup. Increasing the angular resolution using existing deep learning-based view synthesis method and refocusing the video using shift and sum refocusing algorithm produces over blurring of the in-focus region. Our enhancement method targets these blurry pixels and improves their quality by similarity detection and blending. Experimental results show that the proposed approach achieves better refocusing quality compared to traditional shift and sum method.

1. Introduction

Light field (LF) technology is the latest innovation in digital media, seen as an "upgrade" to the way we capture and reproduce visual information [1] [2]. This revolutionary technology is promising a more immersive and holistic imaging experience, allowing post-shoot refocusing, perspective change, depth of field change, and 3D-like content generation with great precision, mimicking the richness of human visual perception. In fact, refocusing is the most far-reaching potential application of light field technology and the foundation for more realistic mixed reality applications, as light field adds a real sense of reality. LFs are captured either by plenoptic cameras or camera arrays. Plenoptic cameras [1] have a primary lens and multiple micro-lenses, which are packed together with a very small distance between them (nanometers), making the different points of views very close together. As a result, they produce angularly dense LFs [3]. On the other hand, camera arrays [2] involve multiple cameras arranged on a rig to capture a scene. The distances between the lenses of the cameras (centimeters) are far greater than they are in plenoptic cameras. This means the point of views are further apart too. Hence, camera array's produce angularly sparse LFs. Here the number of cameras dictates the angular resolution/number of viewpoints of the LF. Each camera's photo-sensor is dedicated to capturing a single image, resulting in higher spatial resolution LFs. Regardless of the capturing method, LF content is an abundance of captured information that enables immersiveness through depth of field, refocusing at multiple levels, variable resolution object identification and all-around view perspective. It has been shown that for the case of microlenses, using the shift and sum algorithm [1], allows us to digitally refocus a sufficiently dense (LF from plenoptic cameras) LF content to a desired depth with excellent results, holding a great promise for mixed reality. Dayan et al. developed a way to train a specially designed deep convolutional neural network for refocusing through sparse LF data [4]. However, this method is also designed for microlense based LF content. A stereo-image refocusing method was introduced in [5]. This method selectively blurs the

URI: https://hdl.handle.net/10125/79417
978-0-9981331-5-7

image based on the estimated depth using the stereo image, to create a refocused effect. However, this method was not extended for more than two cameras. Similar problems can also be found in [6] and [7], the former creates a refocused image using three images, whereas the latter generates the entire light field from a single image. In summary, all the above-mentioned approaches are designed for microlense type of LF data and as such they do not translate well to the case of camera array based LFs. This is mainly due to the large shifts among the views. When we align the in-focus region from different views we end up with a large shift in the other regions. As a result, the refocused images have noticeable ghosting and blocky artifacts in the out of focus region [8]. Researchers have dealt with this problem in various ways. Xiao et al. [9] proposed a technique to identify angular aliasing by statistical analysis of the refocused LF and reduce the aliasing by using lower resolution versions of the refocused image from a Gaussian pyramid. Wang et al. proposed to render the not-in-focus region using bokeh rendering methods to avoid aliasing artifacts, then super resolve the in-focus region to create the refocused image [10]. LF reconstruction methods [11], [12], [13] and [14] are widely used to address the aliasing problem. These methods interpolate novel views between existing views and use the synthetic LF to refocus sub-images. Huang et al. [8] leveraged the fact that these novel views will never be seen or displayed therefore, avoided costly view interpolation techniques. They have also reduced memory access operations significantly. Methods [12], [13] can simultaneously increase the number of viewpoints and image resolution. Since there is a trade-off between the density of views and computational cost, these methods either suffer from a high computational burden, or retain aliasing artifacts to some degree. In summary, the holes caused by occluded areas and the inevitable approximation used in the generated views have shown to yield blurry refocused frames.

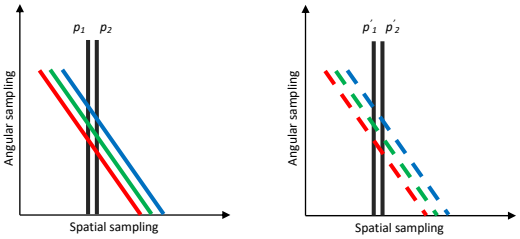
In this paper, we propose an efficient refocusing scheme for camera array LF content, which uses a deep learning network to synthesize an appropriate number of new views and a similarity-based enhancement technique for improving the overall visual quality of the refocused frame. This way we can mitigate the cost of synthesizing too many views in between existing views and get rid of the aliasing artifacts.

The rest of this paper is organized as follows. Section 2 gives an overview of the LF refocusing and shift-and-sum method designed for microlens based LF refocusing. In Section 3, we present our refocusing method. Experimental results are presented and discussed in Section 4. We conclude our work in Section 5.

2. Overview of Light-Field Refocusing

2.1. Light Field Parameterization

Space is filled with light rays of various intensities. The complete set of all the light rays in the physical world is known as the light field (LF). E. Adelson and J. Bergen [15] used the plenoptic function, to express LF. It is a function of the location of an eye/camera in the physical world, the angle at the center of the pupil/lens when the light ray passes through it, the wavelength of the light ray, and the time. M. Levoy and



(a) (b)

Figure 1. (a) Real focused imaged (b) synthetically refocused image

P. Hanrahan [16] assumed the radiance does not change along a light ray unless blocked and the space is free of occlusion. Thus, they further simplified the representation of LF as a 4D function presented in eq. 1.

$$LF = L(u, v, s, t) \tag{1}$$

where uv and st are two parallel planes, $(u, v)$is the intersection of a light ray with the first plane uv (perpendicular to the view direction) and (s, t) is the intersection with the st plane. Given u, v, s and t, the intensity $L(u, v, s, t)$ is sampled.

2.2. LF Refocusing

Let us assume there are $U \times V$ cameras and each image $I_{(u,v)}$ captured by the camera $\{u, v) \mid u \in U, v \in V$, has a $S \times T$ pixel resolution. During the refocusing process, one camera will be firstly picked as the reference camera. The baseline disparity shift (δ_u, δ_v) between the pixel $(s_0, t_0) \mid s \in S, t \in T$ in the reference view and the pixel $(s, t) \mid s \in S, t \in T$ in the other views is calculated at depth z_1 with a far-field depth z_0, where z_0 is arbitrarily large to minimize the reprojection distortion. For each camera, the disparity shift is constant for all the pixels according to the assumption that all the cameras are coplanar. Then, the

disparity shift at any depth z could be calculated as $d(z) * (\delta_u, \delta_v)$ as follows:

$$d(z) = \frac{\frac{1}{z} - \frac{1}{z_0}}{\frac{1}{z_1} - \frac{1}{z_0}} \quad (2)$$

The refocused image $I'_c(s,t,z)$ at depth z is computed by the shift-and-sum algorithm as follows:

$$s' = s + d(z)$$

$$t' = t + d(z)$$

$$I'_c(s,t,z) = \frac{\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} I_{(u,v)}(s' \cdot \delta_u, t' \cdot \delta_v)}{UV} \quad (3)$$

Pixels in all the other views are shifted based on the disparity, then averaged to get the refocused image [17]. Note that eq. 3 also follows the coplanar assumption, which states that for each camera, the disparity shift (δ_u, δ_v) is constant over the entire image.

2.3. Problem Definition

The aperture of an LF capture can be synthetically adjusted to refocus on an arbitrary focal plane (depth). However, the main difference with real aperture capture at that depth and a synthetic aperture is that the real one gathers all the light rays passing through the camera and the synthetic one acquires only a subset of those rays from all the cameras of the capture setup. Ideally an image I is formed by integrating all the light rays R entering the camera from the scene. However, if we have a subset of all the light rays coming from the scene the resultant image looks jagged. Let us analyze this problem for two points p_1 and p_2. Fig. 1(a) illustrates the situation for real focused capture and Fig. 1(b) shows what happens when we synthetically adjust the focal plane to refocus. For the real image both p_1 and p_2 receive angular samples of all three colors however, for the synthetic case p'_1 receives green and p'_2 gets red and blue angular samples. This sudden and abnormal behaviour in neighboring pixels causes aliasing artifacts in LF refocused images. This phenomenon effects the not-in-focus region severely as the samples are further scattered. It also effects the in-focus region however, due to the shifting of the shift-and-sum algorithm the samples become closer consequently reducing the aliasing. Later when the averaging of the angular samples for the same spatial sample takes place, it leaves the pixel blurry rather than in sharp focus. In the next section we propose an enhancement method to improve the quality (reduce blur) of the refocused frame.

3. Proposed Enhancement Method

Our proposed refocusing scheme for LF camera array videos uses a deep learning-based view synthesis method and a modified shift and sum approach combined with a unique similarity-based enhancement technique for the in-focus region. The following subsections describe our approach in detail.

3.1. View Synthesis

Assuming that the cameras in an LF camera array are aligned with equal spacing horizontally and vertically, we investigated the refocusing performance for two different patterns for view synthesis, namely sparse and dense as shown in Fig. 2. We define $n \in \mathbb{Z}^{0+}$ as the view synthesis factor where $n = 1$ is equivalent to one novel view in between two adjacent views. In the sparse pattern, *n* represents the number of synthesized views between every nearest pair of horizontal/vertical original views. In the dense pattern, we also synthesize *n* views between each pair of the horizontal/vertical synthesized views from the sparse pattern. We assume the camera array has an angular resolution of $N \times N$ ($N > 1$). Therefore, the total number of views is $2nN^2 - 2nN + N^2$ for the sparse patternand and $(nN - n + N)^2$ for the dense pattern.

The chosen deep learning network [18] leverages a pre-trained fully-convolutional encoder-decoder architecture (modeled after VGG-19 [19]) to synthesize the novel views. This network does not require any camera parameter information, and thus it

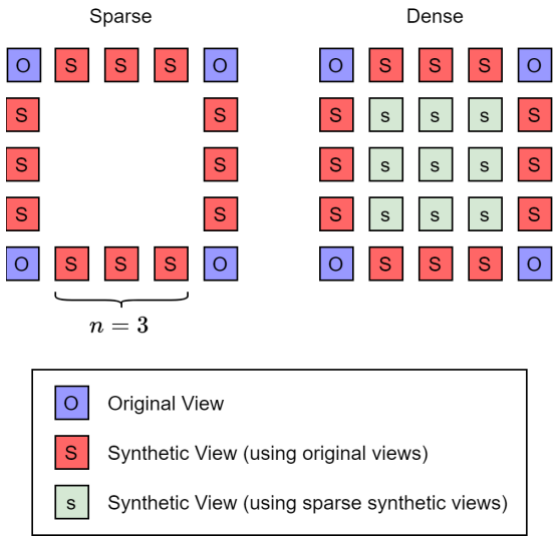


Figure 2. Examples (n = 3) of the proposed sparse and dense view interpolation patterns

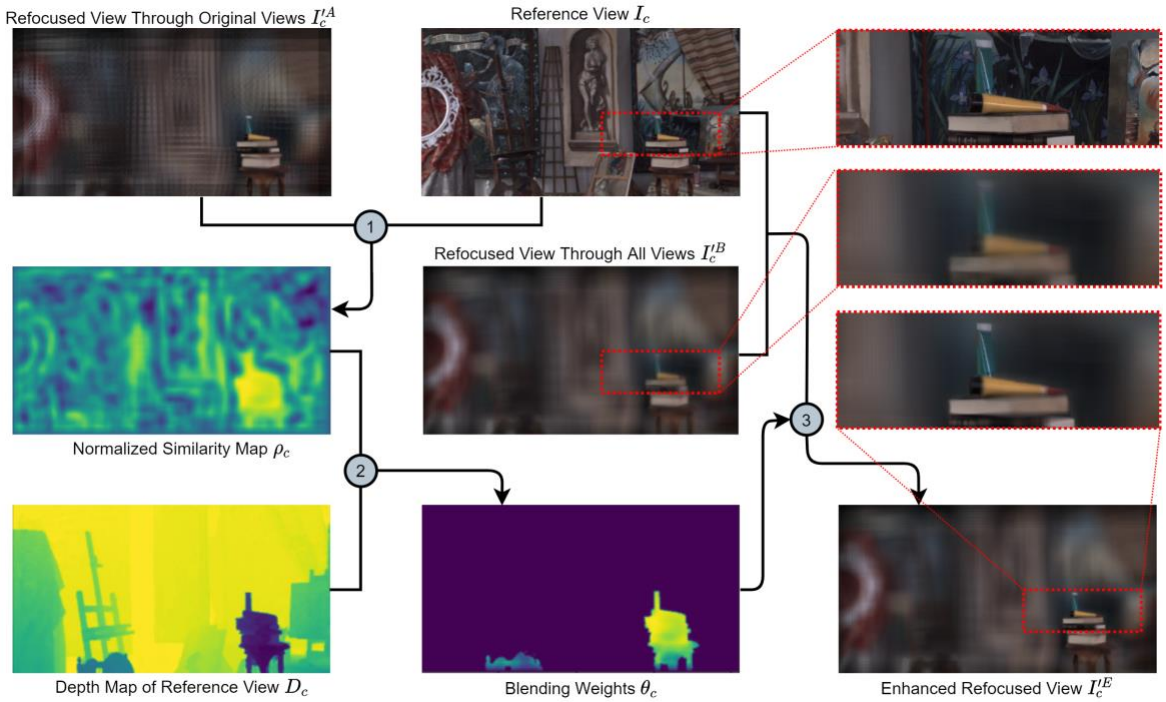


Figure 3. Flowchart of the proposed quality enhancement approach. Some details are enlarged to demonstrate the effectiveness of the approach.

can be generalized to any LF video dataset. Given two input views captured by two horizontally/vertically aligned cameras and the distance between those two cameras, the network can synthesize as many novel views between the input views as needed. First, we use the network to interpolate views for the sparse pattern, for n =1, 2, and 3, then we use it again to fill the "empty" view locations to achieve the dense pattern.

3.1.1. Network Overview. The network is a fully convolutional encoder-decoder architecture (see [18] Table 1 for detailed layer specification). The encoder is modeled after a VGG-19 variant [19]. In order to capture fine texture details the decoder consists of deconvolution layers with skip-connections from lower layers. To maintain spatial resolution and larger scene context it also has dilated convolutions [20] in the intermediate layers. Each layer is followed by a ReLU nonlinearity and layer normalization [21] however in the last layer the tanh activation function without layer normalization is applied. The system is implemented in TensorFlow [22]. It was trained using the ADAM solver [23] for 600K iterations with learning rate 0.0002, β1 = 0.9, β2 = 0.999, and batch size 1. The network was trained on images having a spatial resolution of 1024 × 576, but the model can be applied to arbitrary resolution at inference time in a fully convolutional manner. To tackle the large number of parameters produced by the fully connected layers the training images were cropped into 256×256 size patches. Training time was approximately one week on a Tesla P100 GPU with 32GB memory.

To refocus using the synthesized views, we need to calculate the disparity shifts of the synthetic views. We linearly interpolate these shifts using the disparity shifts of the two adjacent original views. For example, if $n = 1$ that is, we synthesize one view $I_{u,v}$ between every two original views $I_{u-1,v}$ and $I_{u+1,v}$, then we use the disparity shifts of views$I_{u-1,v}$ and $I_{u+1,v}$, to linearly interpolate the disparity shift of the synthesized view $I_{u,v}$. Finally, we apply the shift and sum algorithm using the interpolated and original disparities to produce the refocused video.

3.2. Similarity-based Enhancement Method

After performing the shift and sum refocusing algorithm using the original and synthesized views, we observed that the objects at the refocused plane appear to be blurry especially when they are close to the camera. The reason for this is that the synthesized views are not as accurate as original views could be, and as a result the average values for the in-focus pixels are not accurate either. To address this problem, we introduce a similarity-based enhancement method, which involves the 3 steps shown in Fig. 3: computation of a normalized similarity map (step 1), calculation of pixel blending weights (step 2), and generation of refocused view (step 3).

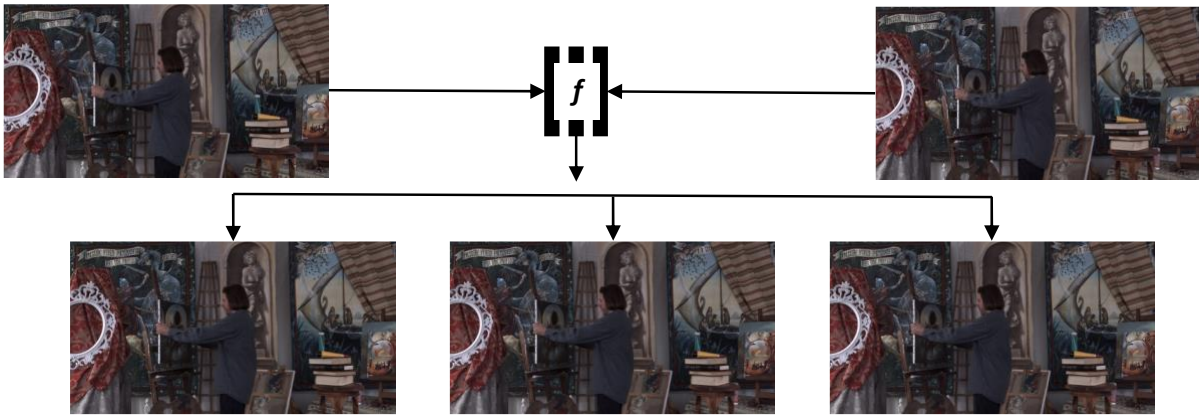


Figure 4. Dense view synthesis results for n=3. Here the top left and right views are original frames from camera 5 and 6. The bottom three views are the synthesized views in between them.

The task is to replace the in-focus pixel values at the refocused view with those of the reference view. To this end, we need to identify which pixels we want to copy. Moreover, for objects (or object parts) not exactly at the refocused plane but near the plane, we need to blend the corresponding pixel values from the reference view and the refocused view. Therefore, we need to find a blending weight for each pixel.

Let the reference view using reference camera c be I_c. For a given refocus depth z, we denote the refocused view using only original views as $I_c'^{A}$ and the refocused view using both original and synthetic views as $I_c'^{B}$. As an in-focus region in $I_c'^{A}$ resembles the corresponding area in I_c more than the not in-focus region, we first compute a pixel level similarity map based on $I_c'^{A}$ and I_c. This similarity map shows structure similarity rather than similar pairs of pixels in both views. Thus, we only compute the similarity maps of the Y channels ($Y_c'^{A}$ is the Y channel of $I_c'^{A}$ and Y_c is for I_c). To identify similar structures on a larger scale, we use a sliding window of size $K \times K$ and stride Δ to compute a similarity value between the corresponding pair of image patches on $Y_c'^{A}$ and Y_c. We used two different methods for computing the similarity values: Structural Similarity Index Measure (SSIM) [24] and Normalized Cross-Correlation (NCC) [25]. The SSIM will give a value in the range of [0, 1], higher SSIM indicates higher similarity. NCC will give a value in the range of [-1, 1], which indicates a correlation score. Since we do not care about the negative correlation, the negative part of the NCC range is replaced by zero. Now, we can interpret the value of NCC as a similarity score as well. To maintain the aspect ratio of the original views, we use zero paddings in $Y_c'^{A}$ and Y_c. Then, we use bi-cubic interpolation to resize the raw similarity map to the same size as the original view. Since all values in the similarity map are in the range of [0, 1], we name it the normalized similarity map, denoted by ρ_c. This is the first step of the proposed quality enhancement approach (see Fig. 3 step 1).

The resulting similarity map highlights the in-focus pixels. However, there could be some not in-focus pixels that still have high similarity scores in ρ_c. This could include false positives from repeated textures or homogeneous areas (e.g., sky) within the frame. To separate the latter from the actual in-focus pixels, we utilize the reference view's depth map D_c to accurately choose only the in-focus pixels. In this second step, D_c is estimated by the Depth Estimation Reference Software (DERS) [26]. Since the depth z of the focus plane is known, the idea is to set the values in ρ_c to be zero, if the corresponding depth in D_c is far from z. We impose γ meters tolerance threshold to calculate a pixelwise contribution mask for the refocused frame. The blending weights θ_c for this mask are computed as follows:

$$\theta_c(s,t) = \begin{cases} \rho_c(s,t) & |D_c(s,t) - z| \leq \gamma \\ 0 & |D_c(s,t) - z| > \gamma \end{cases} \quad (4)$$

Finally, in step 3, the enhanced refocused view $I_c'^{E}$ (Bottom right image in Fig. 3) is generated by blending I_c and $I_c'^{B}$ using θ_c as follows:

$$I_c'^{E}(s,t) = I_c(s,t)\theta_c(s,t) + I_c'^{B}(s,t)\overline{\theta_c}(s,t) \quad (5)$$

Here, $\overline{\theta_c}(s,t) = 1 - \theta_c(s,t)$. Note that blending is performed on each color channel separately.

4. Performance Evaluation and Discussions

We use the Interdigital LF video dataset [17] to evaluate the proposed approach. The LF videos were captured by a synchronized 4×4 camera array at 30fps.

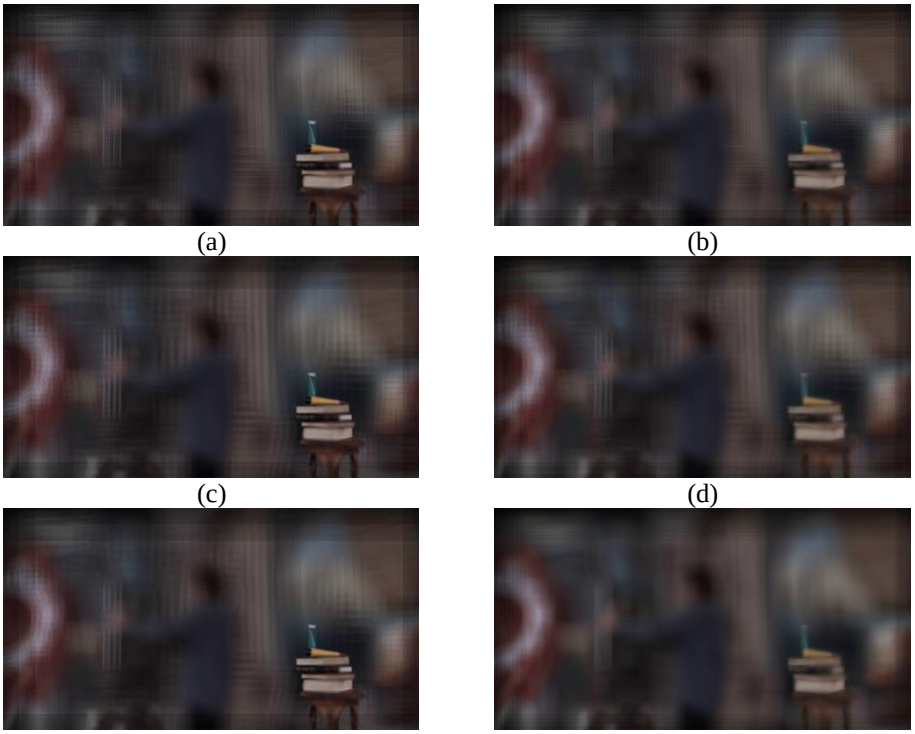


Figure 3. Sparse vs dense view synthesis. (a), (c) and (e) sparse pattern with n=1, 2 and 3 respectively. (b), (d) and (f) dense pattern with n = 1, 2 and 3 respectively.

The cameras are 70mm apart with 50°×37° field of view. Each LF video has 2048×1088 spatial resolution in raw 4:2:0 8bit YUV format and is 12.3 seconds long (i.e., 372 frames). To compare the two similarity methods used in our enhancement approach (SSIM and NCC), we experimented with sparse and dense view interpolation patterns with $n \in \{0, 1, 2, 3, 5, 10\}$, at various depth planes - $z \in \{1.9, 2.0, 3.0, 3.2, 4.0\}$. For the controlled parameters i.e., kernel size, stride and blending tolerance of the similarity computation we experimented with sliding kernel window sizes 10-90 at every 10 interval and have empirically is set $k = 60$, for the stride Δ we examined values from 10 to 15 and finalized 12 to be optimum, and finally for the tolerance threshold we varied γ from 0.1 to 1m at every 0.1m and found 0.5m to be acceptable.

A pretrained model [18] was used to render the synthesized views. This model was trained on YouTube videos which have multiple views of the same scene i.e., static scenes shots from a moving camera. For demonstration purposes of our method, we show the 90th frame of the "Painter" LF video sequence. Fig. 4 present the synthesized views using the deep learning network. We used the videos from camera 5 and 6 and synthesized 3 novel views between the two. As these two cameras are placed side by side horizontally, we can observe the synthesized views have new content appearing to the very right from the left most synthesized view to the right most and content disappearing from the left side. This is exactly what we were expecting. The overall quality of the synthesized views look good too. No visible holes, discontinuity or distortion were observed. Using a Tesla P100 GPU with 32GB memory, the network takes roughly 4.5

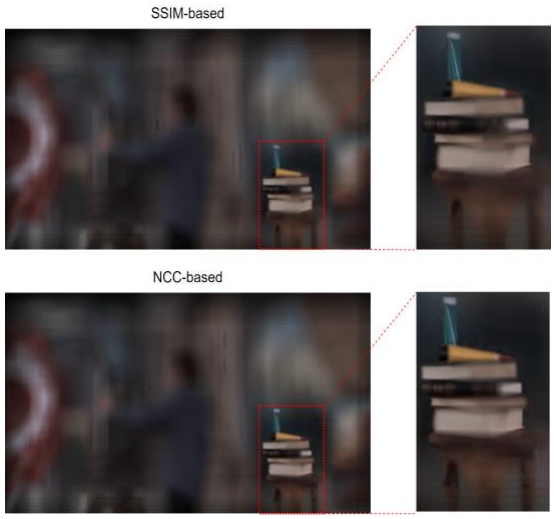


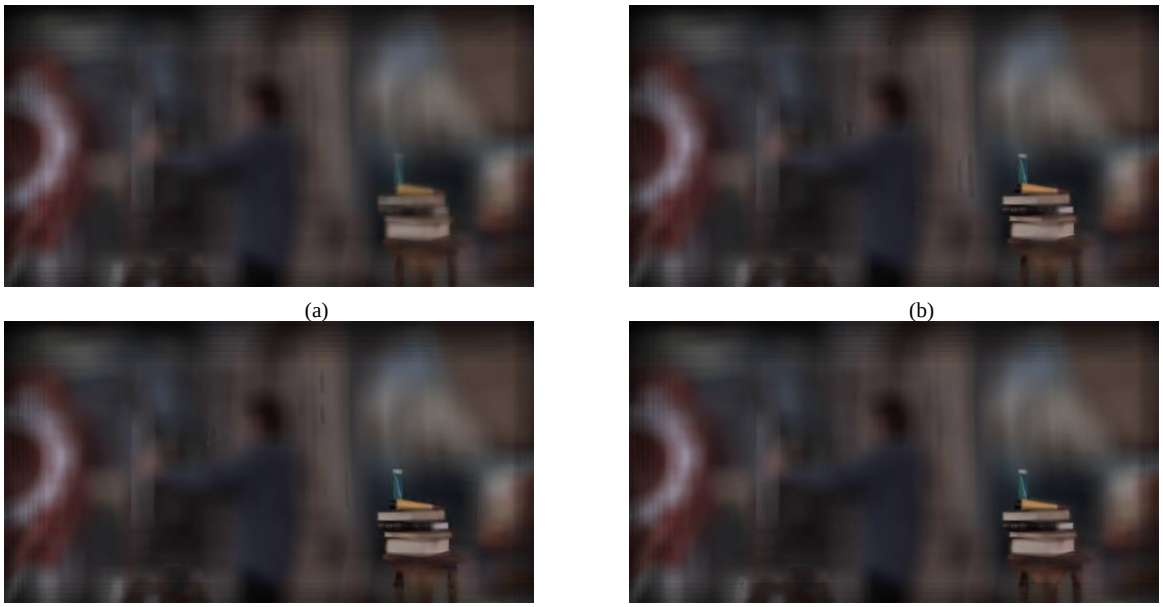
Figure 4. SSIM-based vs NCC-based enhancement method without blending weights.

minutes to generate the scene representation between two original views and 56 seconds to synthesize a novel view. For 16 original views we have 24 stereo pairs hence 24 unique scene representations. So, it takes 108 minutes to generate these scene representations. For *n=3* we have 169 views and 153 among them are synthesized and 16 are original. The view synthesis would take approximately 143 minutes. Therefore, the whole view synthesis process takes about 251 minutes.

We observed that as we increase the value of *n* for view synthesis, the refocusing results improve for both sparse and dense patterns. However, the refocusing quality for dense pattern improves drastically with increasing value of *n*. The dense pattern achieves visually acceptable results with $n = 3$ (total 169 views with 16 original views). These results are presented in Fig. 5 for side-by-side comparison of sparse and dense patterns for $n = 1, 2$, and 3. From the figure we can observe that, the sparse pattern still causes the ghosting artifacts in the not in-focus region even if we keep increasing *n*. Based on the above observations, from here on we fix $n = 3$ and only use the dense pattern to study SSIM-based and NCC-based enhancement methods. Results of both enhancement methods refocused at a 2m distance are shown in Fig. 6. We find that the NCC-based quality enhancement results are more visually pleasing and natural compared to SSIM based. The overall running time of the entire refocusing pipeline for any reference view is 252 minutes. It is worth mentioning that if we precompute the depth maps and synthesize the required novel views beforehand, the shift and sum refocusing, and the in-focus region enhancement can be done in near real-time. To study the effect of our blending approach, we use the normalized similarity map ρ_c as blending weights instead of θ_c. Without the blending weights, we see that both methods introduce blurriness. However, the SSIM-based method introduces larger and more noticeable artifacts than the NCC-based method. In addition, the transition from in-focus to not in-focus region looks abrupt and unnatural. For example, for the books on the table we expect a smooth transition from the in-focus region to the not in-focus region. Fig. 7 compares a refocused frame without any enhancement (Fig. 7 (a)), SSIM and NCC-based enhancement without blending weights (Fig. 7 (b) and (c) respectively) and NCC-based enhancement with blending weights (Fig. 7 (d)). The blending weights from the last step provide us with a smoother transition as seen in Fig. 7 (d). Based on our observation, we recommend using the NCC-based quality enhancement method.

5. Conclusion and Future Work

In this work, we presented an efficient refocusing scheme for camera array LF content, which uses a deep learning network to synthesize an appropriate number of new views and a similarity-based enhancement technique for improving the overall visual quality of



(a) (b) (c) (d)

Figure 5. LF post refocusing in-focus region enhancement (a) no enhancement, (b) SSIM-based enhancement, (c) NCC-based enhancement and (d) NCC-based enhancement with blending weights.

the refocused frame. We found that a dense pattern of synthesized views yields better visual results for the out of focus regions, while the quality enhancement approach improves the visual quality of the in-focus regions by replacing the blurry pixels with corresponding pixels from the reference view using similarity detection and blending. As a result, our method achieves visually acceptable and natural-looking refocused LF videos. To the best of our knowledge, this is the first method designed to efficiently enhance the quality of the refocused region of camera array LF content, offering unprecedented immersiveness and an excellent infrastructure for producing high-quality mixed reality content. Having said that one limitation of this work is the results are not reproducible in a short time frame. Our refocusing pipeline consists of a deep learning network. Depending on the spatial resolution of the LF content generating the required number of novel views might take a while. The code and an easy-to-use UI application for testing our approach are publicly available at GitHub accessible using the following link https://github.com/PeizhiYan/light_field_demo.

6. References

[1] R. Ng *et al.*, "Light Field Photography with a Hand-held Plenoptic Camera To cite this version : HAL Id : hal-02551481 Light Field Photography with a Hand-held Plenoptic Camera," 2005.

[2] B. Wilburn *et al.*, "High Performance Imaging Using Large Camera Arrays," 2005.

[3] ISO/IEC JTC1/SC29/WG1(JPEG) & WG11(MPEG) "Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications," June 2016, Geneva, Switzerland.

[4] S. Ben Dayan, D. Mendlovic, and R. Giryes, "Deep Sparse Light Field Refocusing," 2020.

[5] B. Busam, M. Hog, S. McDonagh, and G. Slabaugh, "SteReFo: Efficient image refocusing with stereo vision," *Proc. - 2019 Int. Conf. Comput. Vis. Work. ICCVW 2019*, pp. 3295–3304, 2019, doi: 10.1109/ICCVW.2019.00411.

[6] O. Cossairt, N. Matsuda, and M. Gupta, "Digital refocusing with incoherent holography," *2014 IEEE Int. Conf. Comput. Photogr. ICCP 2014*, 2014, doi: 10.1109/ICCPHOT.2014.6831819.

[7] Y. Bando and T. Nishita, "Towards Digital Refocusing from a Single Photograph."

[8] C. Huang, J. Chin, H. Chen, Y. Wang, and L. Chen, "Fast Realistic Refocusing For Sparse Light Fields," *2015 IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 1176–1180, 2015.

[9] Z. Xiao, Q. Wang, G. Zhou, and J. Yu, "Aliasing detection and reduction in plenoptic imaging," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 3326–3333, 2014, doi: 10.1109/CVPR.2014.425.

[10] Y. Wang, J. Yang, Y. Guo, C. Xiao, and W. An, "Selective Light Field Refocusing for Camera Arrays Using Bokeh Rendering and Superresolution," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 204–208, 2019, doi: 10.1109/LSP.2018.2885213.

[11] N. K. Kalantari, T. C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–10, 2016, doi: 10.1145/2980179.2980251.

[12] Y. Yoon, S. Member, H. Jeon, and S. Member, "Light-Field Image Super-Resolution Using Convolutional Neural Network," vol. 24, no. 6, pp. 848–852, 2017.

[13] R. A. Farrugia, C. Galea, and C. Guillemot, "Super Resolution of Light Field Images Using Linear Subspace Projection of Patch-Volumes," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 7, pp. 1058–1071, 2017, doi: 10.1109/JSTSP.2017.2747127.

[14] B. Mildenhall *et al.*, "Local light field fusion: Practical view synthesis with prescriptive sampling guidelines," *ACM Trans. Graph.*, vol. 38, no. 4, 2019, doi: 10.1145/3306346.3322980.

[15] E. H. Adelson and J. R. Bergen, "The Plenoptic Function and the Elements of Early Vision," *Comput. Model. Vis. Process.*, pp. 3–20, 2020, doi: 10.7551/mitpress/2002.003.0004.

[16] M. Levo and P. Hanrahan, "Light Field Rendering," 1996.

[17] N. Sabater *et al.*, "Dataset and Pipeline for Multi-view Light-Field Video," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2017-July, pp. 1743–1753, 2017, doi: 10.1109/CVPRW.2017.221.

[18] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: Learning view synthesis using multiplane images," *ACM Trans. Graph.*, vol. 37, no. 4, 2018, doi: 10.1145/3197517.3201323.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.

[20] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, vol. 40, no. 4, pp. 834–848, 2015.

[21] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer Normalization," 2016.

[22] J. D. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, M. K. Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, P. T. Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, and G. B. Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: A System for Large-Scale Machine Learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, 2016, doi:

10.1016/0076-6879(83)01039-3.

[23] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[24] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[25] C. Doutre and P. Nasiopoulos, “Color correction preprocessing for multiview video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 9, pp. 1400–1405, 2009, doi: 10.1109/TCSVT.2009.2022780.

[26] S. Rogge *et al.*, “MPEG-I Depth Estimation Reference Software,” in *2019 International Conference on 3D Immersion (IC3D)*, 2019, pp. 1–6, doi: 10.1109/IC3D48390.2019.8975995.