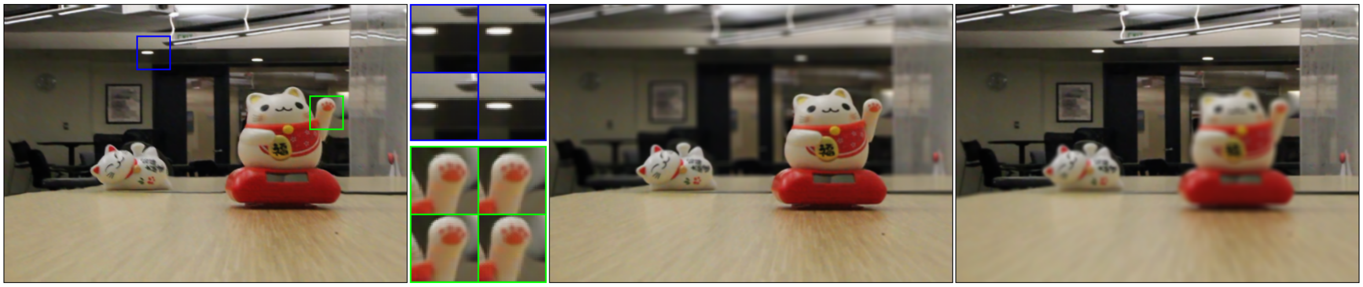
Synthesizing Light Field From a Single Image with Variable MPI and Two Network Fusion

QINBO LI, Texas A&M University
NIMA KHADEMI KALANTARI, Texas A&M University



Our novel view | Corner views | Ours focused on foreground | Ours focused on background

Fig. 1. We use our learning-based techniques to synthesize an 8 × 8 light field from an input image, captured with a standard Canon camera. One of our synthesized corner views along with all the corner views for two insets are shown on the left. On the right, we use our synthesized light field to generate two refocused images. The image is courtesy of Wang et al. [2017].

We propose a learning-based approach to synthesize a light field with a small baseline from a single image. We synthesize the novel view images by first using a convolutional neural network (CNN) to promote the input image into a layered representation of the scene. We extend the multiplane image (MPI) representation by allowing the disparity of the layers to be inferred from the input image. We show that, compared to the original MPI representation, our representation models the scenes more accurately. Moreover, we propose to handle the visible and occluded regions separately through two parallel networks. The synthesized images using these two networks are then combined through a soft visibility mask to generate the final results. To effectively train the networks, we introduce a large-scale light field dataset of over 2,000 unique scenes containing a wide range of objects. We demonstrate that our approach synthesizes high-quality light fields on a variety of scenes, better than the state-of-the-art methods.

CCS Concepts: • **Computing methodologies → Image-based rendering**.

Additional Key Words and Phrases: Light field, view synthesis, convolutional neural network

ACM Reference Format:
Qinbo Li and Nima Khademi Kalantari. 2020. Synthesizing Light Field From a Single Image with Variable MPI and Two Network Fusion. *ACM Trans. Graph.* 39, 6, Article 229 (December 2020), 10 pages. https://doi.org/10.1145/3414685.3417785

1 INTRODUCTION

4D light fields capture both the intensity and direction of light, enabling appealing effects such as viewpoint change, synthetic aperture, and refocusing. However, capturing a light field is difficult as it requires taking a set of images from different views at the same time. While off-the-shelf cameras such as Lytro and RayTrix capture light fields, they are not as widespread as standard digital and cellphone cameras. To make light fields widespread, our goal in this work is to synthesize 4D light fields from a single 2D RGB image.

With the rise of deep learning in recent years, the problem of single image view synthesis has received considerable attention [Liu et al. 2018; Niklaus et al. 2019; Park et al. 2017; Tulsiani et al. 2018]. However, these approaches are not designed to generate structured 4D light fields with small baselines. Specifically, they are not able to resolve the ambiguity in the scale of the scene's depth, often producing light fields with incorrect scale. The only exceptions are the approaches by Srinivasan et al. [2017] and Cun et al. [2019]. These methods first estimate the scene geometry at the new view and use it to warp the input and reconstruct the novel view image. Unfortunately, because of the per view estimation of the scene geometry, when trained on general scenes, these methods typically produce results with incorrect depth and inconsistent parallax.

To address these problems, we build upon multiplane image (MPI) representation [Zhou et al. 2018] that describes a scene through a series of RGBα images at fixed depths. Once this representation is estimated for a scene, novel view images can simply be reconstructed through alpha composition of the reprojected RGB images at different layers. Although MPI has been successfully used for multi-image view synthesis [Mildenhall et al. 2019; Srinivasan et al. 2019; Zhou et al. 2018], it produces sub-optimal results in our application of *single* image view synthesis with *small* baselines (See

Authors' addresses: Qinbo Li, Texas A&M University, lee@tamu.edu; Nima Khademi Kalantari, Texas A&M University, nimak@tamu.edu.

0730-0301/2020/12-ART229
https://doi.org/10.1145/3414685.3417785

Fig. 3). Therefore, we propose to extend this representation by allowing the disparity[1] of each layer to be inferred from the scene. Our representation, called *variable* multiplane image (VMPI), is able to better describe a scene as the disparity of the layers are set according to the distribution of the depth of the objects in the scene.

Moreover, we observe that reconstructing the visible and occluded regions from a single image requires two different processes. The content of the visible regions exist in the input image, but the occluded areas need to be hallucinated. Therefore, we propose to handle these two types of areas through two parallel convolutional neural networks (CNN). In our system, one network handles the visible areas, while the other is responsible for reconstructing the occluded areas. Specifically, the networks estimate two sets of VMPIs using a single image and its corresponding depth, obtained with a pre-trained network. These VMPIs are then used to reconstruct two novel view images which are in turn fused using a soft visibility mask to generate the final image. We train the visible network with an $L1$ loss, but use a perceptual loss to train the occluded network to hallucinate visually pleasing content in the occluded regions.

To effectively train our networks on general scenes, we introduce a new light field dataset of 2,000 unique scenes containing a variety of objects such as building, bicycle, car, and flower. Although all of our training light fields have an angular resolution of 8×8, we propose a training strategy to supervise our network for generating 15×15 light fields. Using light fields with a fixed baseline as our training data, our system learns to infer the scale of the scene's depth from the input data. We demonstrate that our approach is able to produce high-quality light fields that match the ground truth better than the state-of-the-art methods. Overall, our main contributions are:

- We present an extension of MPI scene representation and demonstrate its advantage over the original MPI for single image view synthesis with small baseline (Sec. 3.1).
- We propose to handle the visible and occluded regions through two parallel networks (Sec. 3.2).
- We introduce a new large-scale light field dataset containing over 2,000 unique scenes (Sec.3.3).

2 RELATED WORK

The problem of synthesizing novel views from one or more images has been studied for several decades. Here, we provide a brief overview of the previous approaches by classifying them into two general categories.

Multi-Image View synthesis. View synthesis approaches typically leverage the scene geometry [Chaurasia et al. 2013; Hedman et al. 2017; Hedman and Kopf 2018; Penner and Zhang 2017] to utilize the content of the input views and properly synthesize the novel view image. In recent years and with the emergence of deep learning, many approaches based on convolutional neural networks (CNN) have been proposed. Flynn et al. [2016] estimate the color and depth using two CNNs and synthesize the novel view using these estimates. Choi et al. [2019] propose to estimate depth probability volume to reconstruct an initial image and then refine it to produce the final image. Zhou et al. [2018] propose to magnify the baseline of a pair of images with small baselines by introducing multiplane image (MPI) scene representation. Srinivasan et al. [2019] use MPI for extreme view extrapolation, while Mildenhall et al. [2019] and Flynn et al. [2019] utilize it for large baseline view synthesis. We build upon the MPI representation, but use a single image as the input and tackle the specific problem of synthesizing structured light field images.

Several approaches have been specifically designed to synthesize light fields from a sparse set of input views. Kalantari et al. [2016] propose to reconstruct a light field from the four corner images by breaking the problem into disparity and appearance estimations. To handle scenes with non-Lambertian effects, Wu et al. [2017] propose to increase the angular resolution of a light field by operating on epipolar-plane images (EPI). Wang et al. [2018a] propose a similar approach, but use 3D CNN and 2D strided convolutions on stacked EPIs to avoid the pre- and post-processing steps of Wu et al.'s approach. These methods require at least four images to reconstruct a light field and are not able to work on a single image.

Single-Image View synthesis. A large number of methods have proposed to use CNNs to estimate novel views from a single image [Olszewski et al. 2019; Park et al. 2017; Rematas et al. 2016; Tatarchenko et al. 2015; Yan et al. 2016; Yang et al. 2015], but they are only applicable to a specific object or scene. Zhou et al. [2016] propose to synthesize novel views by first predicting a flow and then using it to warp the input image. Tulsiani et al. [2018] estimate layered depth image representation of a scene using view synthesis as a proxy task. Liu et al. [2018] estimate a set of homography transformations and masks and use them to reconstruct the novel view. Dhamo et al. [2019] use a conditional generative adversarial network to estimate novel views by breaking the scene into foreground and background layers. To generate 3D Ken Burn effects from a single image, Niklaus et al. [2019] use their estimated depth to map the input image to a point cloud and use a network to fill in the missing areas. Evain and Guillemot [2019] introduce a lightweight neural network and train it on a set of stereo images. Using an input RGB-D image, Shih et al. [2020] reconstruct a 3D photograph through layer depth image representation. Wiles et al. [2020] introduce a point cloud based view synthesis approach from a single image without needing 3D ground truth information for supervising the network during training. While these approaches work well for their intended applications, because of the scale ambiguity of the depth, they cannot properly reconstruct structured light fields with small baselines.

A couple of approaches propose to synthesize structured light field images from a single image. Srinivasan et al. [2017] use two sequential networks to perform disparity estimation and image refinement. The first network estimates a set of disparities at all the novel views. These disparities are then used to warp the input image to the novel view and they in turn are refined through the second network. Cun et al. [2019] propose to estimate a flow from the input image, depth, and the position of novel view. The flow is then used to warp the input image and reconstruct the novel view image. These approaches estimate the scene geometry independently for each view and, thus, produce results with inconsistent parallax and

[1] Disparity and depth are closely related in a structured light field and, thus, we use them interchangeably.

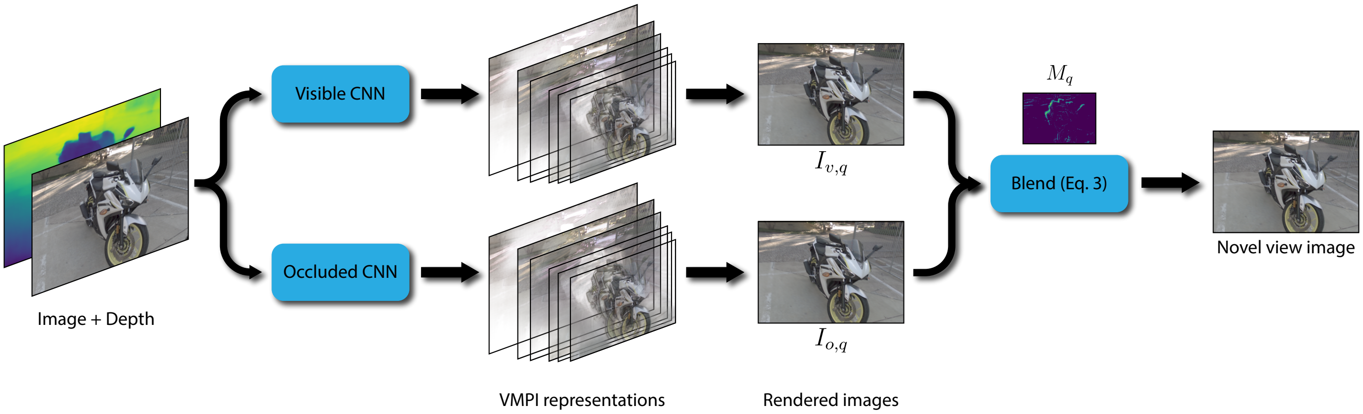


Fig. 2. Our system consists of two CNNs for handling the visible and occluded regions. Our networks take a single image along with its corresponding depth (estimated using Wang et al.'s approach [2018b]) as the input and estimate our proposed VMPI scene representation. The two VMPI representations are then used to reconstruct two images, which are then combined through a soft visibility mask to generate the final image. The visibility mask is calculated using the transparency layers of the visible network's VMPI (see Eq. 6).

inaccurate depth, when trained on general scenes. In contrast, we use a unified representation of the scene geometry to reconstruct all the views and are able to produce results with high quality.

Finally, the concurrent work by Tucker and Snavely [2020] also propose an MPI-based approach for single image novel view synthesis. However, they mostly perform large baseline view synthesis and only demonstrate their results for small baseline light fields on the dataset of Srinivasan et al. [2017], which is limited to flowers. In contrast, our goal is to develop a practical system for small baseline view synthesis with the ability to handle general scenes.

3 APPROACH

Given a single image I and the position of the novel view $\mathbf{q}$ (in u and v directions), our goal is to synthesize the novel view image I_q. Our system consists of two parallel networks that are responsible for generating results in the visible and occluded areas. To do this, each network estimates a novel layered representation of the scene from the input image and its corresponding depth. These representations are then used to render two images, which are in turn combined using a soft visibility mask to generate the final image.

To estimate the layered scene representation, our system needs to understand the geometry of the scene. In multi-image view synthesis where several images from different views are provided as the input to the system, the depth can be easily inferred through correspondences between the input images. However, in our application, only a single image is provided to the system and, thus, the geometry needs to be inferred through contextual information. Unfortunately, it is difficult to train a network to do so using limited light field training data. Therefore, we use the pre-trained single image depth estimation network by Wang et al. [2018b] to estimate the depth and use it along with the image as the input to our system.

We note that the input depth is relative as there is an inherent ambiguity in the scale of the estimated depth from a single image. However, by training our system on light fields with fixed baselines, our network can infer this scale from the input image. This is in contrast to the large baseline view synthesis approaches like the one by Niklaus et al. [2019], where the estimated depth is directly used to reconstruct the novel view images. Because of this, as shown in Sec. 4, Niklaus et al.'s method produces novel view and refocused images that do not match the ground truth.

In the following sections, we start discussing different components of our system by first explaining our new scene representation. The overview of our approach is given in Fig. 2.

3.1 Variable Multiplane Image Representation

Our method builds upon multiplane image (MPI) scene representation, proposed by Zhou et al. [2018]. MPI represents a scene at the input coordinate frame through a set of N fronto-parallel planes at fixed disparities $D = \{d_1, \cdots, d_N\}$. Each plane L_i consists of an RGB color image C_i and a transparency map α_i, i.e., $L_i = \{C_i, \alpha_i\}$. To reconstruct the image at novel view $\hat{I}_q$, we first translate the planes to the coordinate frame of the novel view based on their corresponding disparity as follows:

$$L_{i,q}(\mathbf{p}) = L_i(\mathbf{p} + d_i \mathbf{q}) \tag{1}$$

where $\mathbf{p}$ is the pixel position in x and y directions, while $\mathbf{q}$ is the novel view position in u and v directions. The final novel view image can then be reconstructed by alpha blending the color images at each layer from back to front through standard over operator [Porter and Duff 1984]. This can be formally written using the following recursive function:

$$I_{i,q} = (1 - \alpha_{i,q})\, I_{i-1,q} + \alpha_{i,q}\, C_{i,q}, \quad \text{where} \quad I_{1,q} = C_{1,q}. \tag{2}$$

where $I_{N,q}$ is the estimated novel view image $\hat{I}_q$. Note that, the transparency of the background layer α_1 is always 1 to ensure there are not any holes in the final image.

To utilize this representation in our single image view synthesis application, we can simply use a CNN to estimate the MPI of a scene from an input image and its corresponding depth. This network can be trained by minimizing the $L1$ loss between the rendered and ground truth novel view images. However, such a system is not able to produce satisfactory results, as shown in Fig. 3. When using a large number of planes ($N = 32$), the network tends to repeat the content over multiple planes and produce a blurry rendered image. This is mainly because the $L1$ loss does not sufficiently penalize blurriness to force the network to place the content on a single plane with the correct depth. On the other hand, with fewer layers ($N = 8$) the depth planes are sufficiently separated and the network can properly place the content on the correct planes. However, in cases where an object falls in between two planes, the network places the object on multiple planes to simulate the correct depth, producing results with ghosting artifacts. Note that, as discussed at the end of this section, this problem is specific to the use of MPI in our application of *single* image view synthesis with *small* baselines.

We address this problem by extending the original MPI representation to have layers with scene-dependent disparities. In our representation, called variable multiplane image (VMPI), the disparity of each layer is inferred from the input. The main advantage of our representation is that we can use fewer planes ($N = 8$), but place them more accurately throughout the scene and avoid ghosting and blurring artifacts, as shown in Fig. 3.

To estimate the VMPI representation, we need to estimate the disparity d_i (scalar) at each layer, in addition to the color image C_i and the transparency map α_i. To do this, our network estimates a 5-channel output for each layer; three channels for RGB, one channel for α, and one channel for disparity. The final disparity of each layer is then computed by averaging the pixel values in the disparity channel of the corresponding layer. We use the estimated disparity at each layer (instead of the fixed ones) to perform the warping in Eq. 1 and then blend the warped images through Eq. 2 to reconstruct novel view images. Note that since the estimated disparities could potentially be out of order, we first sort them before using Eq. 2 to blend the planes. Alternatively, we can force the network to estimate the disparities in order by introducing a penalty for out of order estimations during training.

We also experimented with estimating the disparity of each layer using a set of fully connected layers, but this strategy produced results with lower quality than our approach. This is mainly because the network needs to know the disparity of each VMPI layer to be able to appropriately estimate their RGBα values. However, in this case, the disparities are estimated by a separate fully connected branch and the main branch is unaware of the estimated disparities.

Discussion. It is worth noting that existing MPI-based techniques do not face the blurriness problem, discussed in this section. This is mainly because *all* of these approaches use a perceptual VGG-based loss [Zhou et al. 2018] between the rendered and ground truth novel view images to train their network. The VGG-based loss enforces the network to assign the content to one of the MPI layers, as blurriness is heavily penalized by this loss. This is even the case when using a small number of planes, as the VGG-based loss function favors slight misalignment over blurriness.

Unfortunately, when using the VGG-based loss in our application, the network tends to estimate the input image for all the novel views, i.e., $I_q = I \quad \forall q$. This is mainly because the VGG-based loss is robust to slight misalignment and favors sharpness. Since our baseline is small, the input and other novel views are relatively close. Therefore, the sharp and high-quality input image with slight misalignment produces lower error than what the network can potentially reconstruct from a single image.

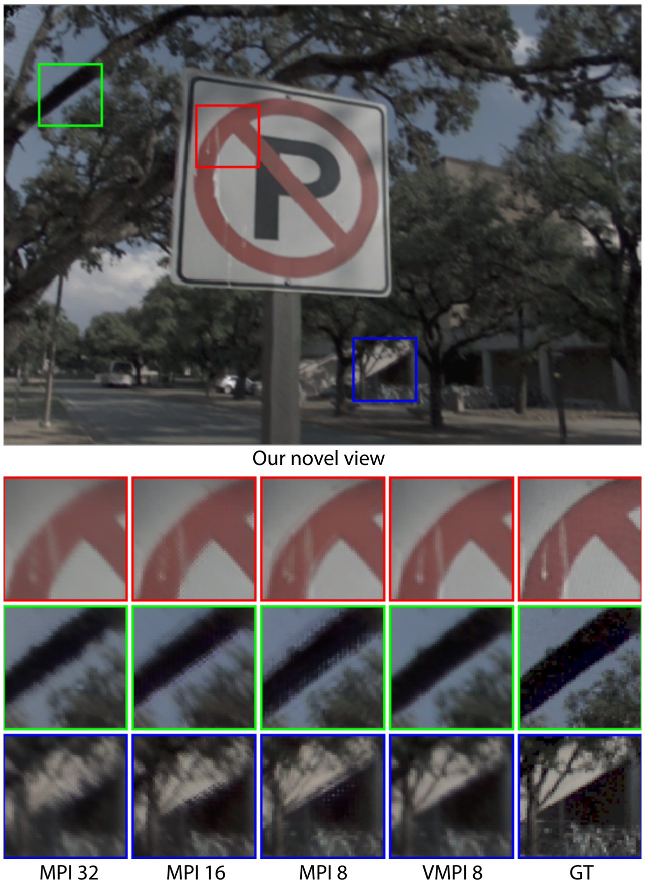


Fig. 3. Comparison of our approach with MPI and our proposed VMPI representations. MPI with a large number of planes produces results with blurriness, while MPI with a small number of planes produces ghosting artifacts. Our VMPI representation with a small number of planes can accurately represent the scene and produce high-quality results.

3.2 Two-Network Fusion

The network trained with $L1$ loss to estimate VMPI representation, as discussed in the previous section, is able to produce reasonable results, but has difficulty reconstructing the object boundaries, as shown in Fig. 4 (Visible). In these regions, which are occluded in the input view, the CNN simply replicates the visible content, producing results with ghosting artifacts. Our main observation is that the process of reconstructing these two areas is different. The content of the visible regions is available in the input image, but the occluded areas need to be hallucinated.

Therefore, we propose to handle these areas separately through two parallel networks (see Fig. 2). The visible and occluded networks first estimate two sets of VMPI representations. These VMPIs are then used to reconstruct the visible and occluded images. We then fuse reconstructed images using a soft visibility mask to generate the final novel view image. This can be formally written as:

$$\hat{I}_q = M_q \, I_{v,q} + (1 - M_q) \, I_{o,q}, \tag{3}$$

where $I_{v,q}$ and $I_{o,q}$ are the reconstructed images using the output of the visible and occluded networks, respectively. Moreover, M_q is a soft mask representing the visible regions, i.e., one in the fully

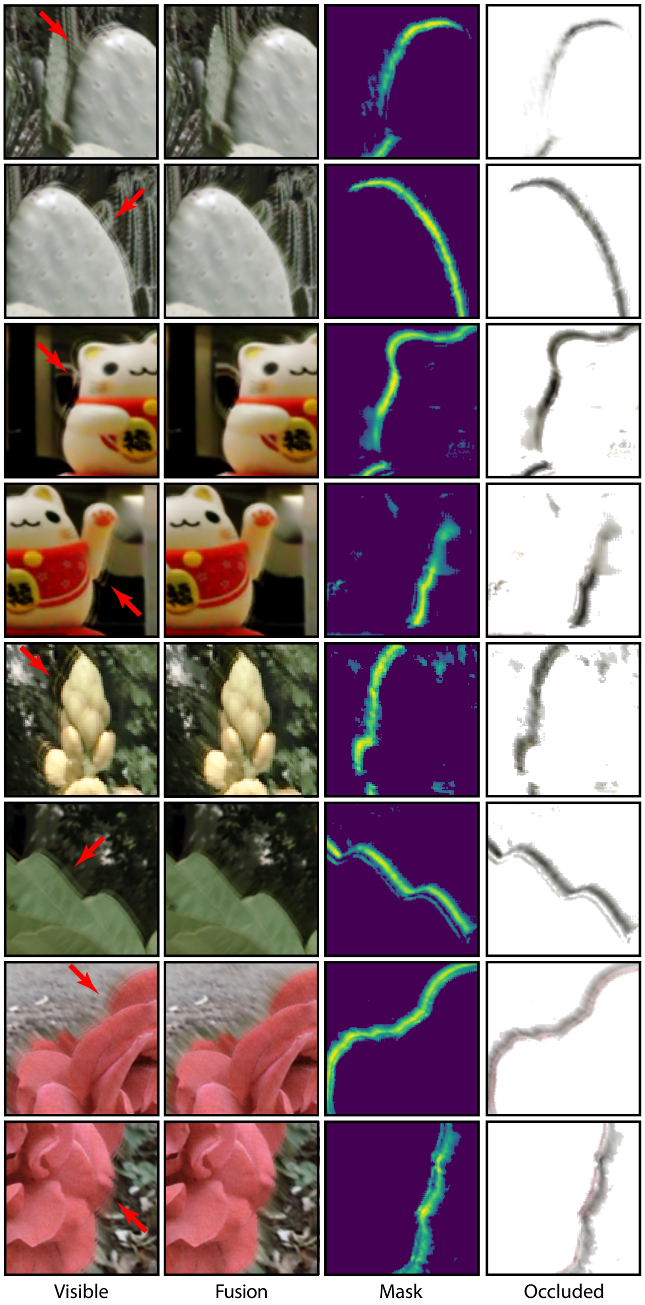


Fig. 4. Our single visible CNN (first column) is not able to properly reconstruct the occluded regions, producing results with ghosting artifacts as indicated by the red arrows. Our system with two parallel networks (second column) significantly reduces these artifacts. For each scene, we also show the mask M_q (third column), obtained through Eq. 6, as well as the occluded image. Note that for better visualization, we combine the occluded image with an all white image, i.e., $M_q\mathbf{1} + (1 - M_q)I_{o,q}$.

visible regions and zero in the completely occluded areas. This mask basically identifies the regions where the visible network has difficulty reconstructing high-quality results, since they are occluded in the input image.

We propose to compute the mask using the transparency maps of the VMPI representation, estimated by the visible network. Our high level idea is that the occluded regions are reconstructed using areas of the VMPI layers that are not visible in the input image. To compute the visibility of each VMPI layer, we first rewrite Eq. 2 in explicit form as follows [Flynn et al. 2019]:

$$I = \sum_{i=1}^{N} \beta_i C_i \quad \text{where} \quad \beta_i = \alpha_i \prod_{j>i} (1 - \alpha_j). \tag{4}$$

Note that, this equation is written in the input coordinate frame, and I is the reconstructed input image using the VMPI representation. Here, β_i basically indicates how much each color image C_i is visible in the input image. To measure the visibility in the novel view, we first warp β_i to the coordinate frame of the novel view:

$$\beta_{i,q}(\mathbf{p}) = \beta_i(\mathbf{p} + d\mathbf{q}). \tag{5}$$

The soft visibility mask M_q can then be computed as follows:

$$M_q = \min(\sum_{i=1}^{N} \beta_{i,q}, 1), \tag{6}$$

where the min operation ensures that the mask is clamped to one. In the occluded regions, all the β_i's are close to zero and, consequently, M_q has a value around zero. On the other hand, in the visible areas, β_i of at least one layer is close to one. Therefore, the visibility mask in the visible areas is roughly one.

Our network architecture is similar to that of Zhou et al. [2018], but with smaller number of filters in each layer (see Table 1). We use the same architecture for both visible and occluded networks. All the layers with the exception of the last layer are followed by a ReLU activation function and batch normalization [Ioffe and Szegedy 2015]. The last layer has a tanh activation function and outputs a tensor with 40 channels (8 VMPI layers each consisting of 5 channels for RGBα and disparity).

As discussed, we use $L1$ distance between the estimated and ground truth novel view images to train the visible network. However, we train the occluded network using the VGG-based perceptual loss [Zhou et al. 2018], as implemented by Koltun and Chen [2017], to produce visually pleasing results in the occluded regions.

Note that, Mildenhall et al. [2019] also propose a fusion strategy, but they fuse rendered images from different views. In contrast, our approach works on a single input image and the fusion is performed between the rendered images of the same view.

3.3 Dataset

To effectively train our system, we collect over 2,000 light fields (See Fig. 5) using a Lytro Illum camera from various indoor and outdoor scenes. Our indoor scenes contain a variety of objects such as tables, chairs, shelves, and mugs, while the outdoor scenes include objects like flowers, trees, signs, bicycles, cars, and buildings. We set the focal length as 35 mm for all the light fields, but choose the other camera parameters, such as shutter speed and ISO, manually based on the lighting condition for best quality.

We split this dataset into a set of 1950 scenes for training and 50 scenes for testing. We also use the Stanford Multiview Light Field dataset [Dansereau et al. 2019] for training and testing. This dataset has a set of 850 unique scenes each containing 3 to 5 light fields from

Table 1. Our network architecture. Here, k is the kernel size, s is the stride, and d is the kernel dilation. Moreover, "channels" refers to the number of input and output channels at each layer and '+' indicates concatenation.

Layer	k	s	d	channels	input
conv1_1	3	1	1	4/32	input + depth
conv1_2	3	2	1	32/64	conv1_1
conv2_1	3	1	1	64/64	conv1_2
conv2_2	3	2	1	64/128	conv2_1
conv3_1	3	1	1	128/128	conv2_2
conv3_2	3	1	1	128/128	conv3_1
conv3_3	3	2	1	128/256	conv3_2
conv4_1	3	1	2	256/256	conv3_3
conv4_2	3	1	2	256/256	conv4_1
conv4_3	3	1	2	256/256	conv4_2
conv5_1	4	.5	1	512/128	conv4_3 + conv3_3
conv5_2	3	1	1	128/128	conv5_1
conv5_3	3	1	1	128/128	conv5_2
conv6_1	4	.5	1	256/64	conv5_3 + conv2_2
conv6_2	3	1	1	64/64	conv6_1
conv7_1	4	.5	1	128/64	conv6_2 + conv1_2
conv7_2	3	1	1	64/64	conv7_1
conv7_3	3	1	1	64/40	conv7_2

different views. We use a set of 766 scenes for training containing roughly 3800 light fields. From the rest of the 84 scenes, containing around 400 light fields, we select 100 light fields for test. In summary, combining the Stanford and our dataset, we have a total of 1950 + 3800 light fields for training and 50 + 100 for testing.

The angular resolution of Lytro Illum light fields is 14×14, but we only use the center 8×8 views as the corner views are outside the aperture. To train our network, we randomly select one of the 4 corner sub-aperture images as the input. We then randomly select one of the remaining views as ground truth. With this simple strategy, we are able to supervise our system for generating light fields with angular resolution of 15×15 from only 8×8 views. Note that, this is not possible with Srinivasan et al.'s approach as their network estimates the entire 8×8 light field in a single pass.

We use the approach by HaCohen et al. [2011] to match the color of raw light fields to their processed version. We train our system on randomly cropped patches of size 192×192. We apply a series of data augmentations including randomly adjusting the gamma, saturation, hue, and contrast, as well as swapping the color channels to reduce the chance of overfitting.

3.4 Training

Training both networks by directly minimizing the loss in one stage is difficult. Therefore, we perform the training in two separate steps. In the first stage, we train the visible network by minimizing the l_1 loss between the reconstructed $I_{v,q}$ and ground truth I_q novel view images. At the end of this stage, the visible network is able to produce high-quality results in the visible areas, but produces ghosting artifacts in the occluded areas, as shown in Fig. 4. We use the output of this network to estimate the soft visibility mask, identifying the problematic occluded regions.

In the second stage, we freeze the weights of the visible network and only train the occluded network by minimizing the VGG-based perceptual loss between the final reconstructed $\hat{I}_q$ and ground truth I_q images. Note that, we compute the loss using the final fused image instead of the reconstructed image by the occluded network $I_{o,q}$. By doing so, we ensure that the synthesized content in the occluded regions blends with the reconstructed image in the visible areas in a coherent and visually pleasing manner. Fine-tuning both networks did not improve the results and, thus, we use the trained networks after the second stage to produce all the results.

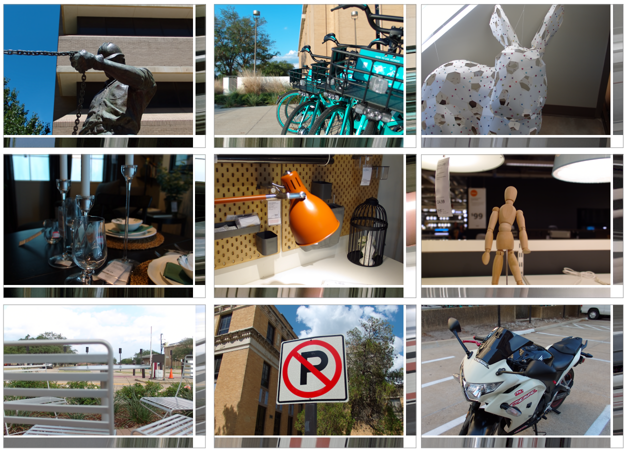


Fig. 5. We introduce a new light field dataset containing over 2,000 unique scenes covering a wide range of objects, nine of which are shown here. We capture our dataset using a Lytro Illum camera from various locations and under different lighting conditions. The epipolar images shown on the right and below each image demonstrate the depth complexity of our scenes.

Table 2. Comparison of our approach against two state-of-the-art view synthesis methods, as well as a version of their approach with the depth estimated by Wang et al. [2018b] as the input. We evaluate synthesized 8×8 and 15 × 15 light fields on three datasets in terms of PSNR and SSIM (higher is better in both cases). The best results are shown in bold. Note that, all the results are generated by training our method and both versions of Srinivasan et al.'s approach on the training set of Stanford/Ours dataset.

		8×8 LF		15×15 LF	
	Algorithm	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Stanford/Ours	Srinivasan	29.92	0.9177	NA	NA
	Srinivasan + Wang	30.03	0.9215	NA	NA
	Niklaus	24.00	0.7616	20.04	0.5835
	Niklaus + Wang	23.88	0.7864	21.02	0.6456
	Ours	**30.53**	**0.9306**	**28.52**	**0.9033**
Srinivasan	Srinivasan	24.94	0.7935	NA	NA
	Srinivasan + Wang	25.75	0.8182	NA	NA
	Niklaus	25.00	0.7639	22.06	0.6532
	Niklaus + Wang	26.38	0.8188	24.32	0.7409
	Ours	**26.78**	**0.8547**	**25.56**	**0.7990**
Kalantari	Srinivasan	26.96	0.8638	NA	NA
	Srinivasan + Wang	27.60	0.8829	NA	NA
	Niklaus	24.18	0.7948	22.04	0.6886
	Niklaus + Wang	24.94	0.8219	24.26	0.7791
	Ours	**29.03**	**0.9188**	**27.36**	**0.8841**

4 RESULTS

We implement our approach in PyTorch [Paszke et al. 2019] and train our networks using Adam [Kingma and Ba 2015] with the default parameters (β_1 = 0.9, β_2 = 0.999, ϵ = 1e-8). We use a learning

Table 3. We compare of our approach with VMPI using two-network fusion against our method with a single network as well as our technique with MPI representation with different numbers of planes. We evaluate synthesized 8×8 and 15 × 15 light fields in terms of PSNR and SSIM (higher is better in both cases). The best results are shown in bold.

	8×8 LF		15×15 LF	
Algorithm	PSNR↑	SSIM↑	PSNR↑	SSIM↑
MPI 8	29.57	0.9137	27.89	0.8856
MPI 16	29.39	0.9198	26.87	0.8856
MPI 32	29.28	0.9149	26.68	0.8626
VMPI 8 (Single)	30.53	0.9305	28.51	0.9033
VMPI 8 (ours)	**30.53**	**0.9306**	**28.52**	**0.9033**

rate of 2×10^{-4} and a batch size of 8 throughout the training. We perform both training and testing on a machine with an Nvidia GTX 1080Ti GPU with 11 GB of memory. Throughout this section, we compare our approach against the following state-of-the-art single image view synthesis approaches:

Srinivasan et al. [2017]. This approach uses a network to estimate the disparity at the novel view using the input image. The disparity is then used to backward warp the input image. Finally, this warped image is passed to a second network for refinement. In the original approach both networks are trained in an end-to-end fashion on a training set containing light fields of flowers. To ensure fairness, we retrain their networks on our training dataset, described in Sec. 3.3. We use the code provided by the authors for all the comparisons.

Srinivasan et al. [2017] + Wang et al. [2018b]. The only difference here is that in addition to the input image, we also provide the depth, estimated using the method of Wang et al. as the input to Srinivasan et al.'s disparity estimation network. By doing so, we decouple the effect of depth on the quality of the results since our approach uses the same depth as the input.

Niklaus et al. [2019]. This method first uses a network to estimate the depth given an input image. The depth is then used to forward warp the input image into the novel view. Note that, since the depth is computed on the input image, backward warping is not possible. The last stage of this approach is to use another network to inpaint the potential holes in the warped image. Again, we use the source code provided by the authors for all the comparisons.

Niklaus et al. [2019] + Wang et al. [2018b]. Here, we use Wang et al.'s depth instead of the depth estimated by Niklaus et al.'s depth estimation network. Again, this comparison will decouple the effect of the depth from the quality of the results.

To ensure fairness, we train both versions of Srinivasan et al.'s approach on our training set. However, since the main component of Niklaus et al.'s approach (forward warping) is non-learning we use their provided source code for comparisons.

4.1 Quantitative Comparison to Other Approaches

In Table 2, we compare our approach against the other methods for generating 8 × 8 and 15 × 15 light fields in terms of PSNR and SSIM [Wang et al. 2004]. Note that, since the views in Srinivasan et al.'s approach [2017] are hard-coded, this method can only generate 8 × 8 light fields. Since the test light fields are all 8 × 8, we perform the numerical evaluation for 15 × 15 light fields by using each one of the four corner views of an 8 × 8 light field as the input and reconstructing the rest of the views.

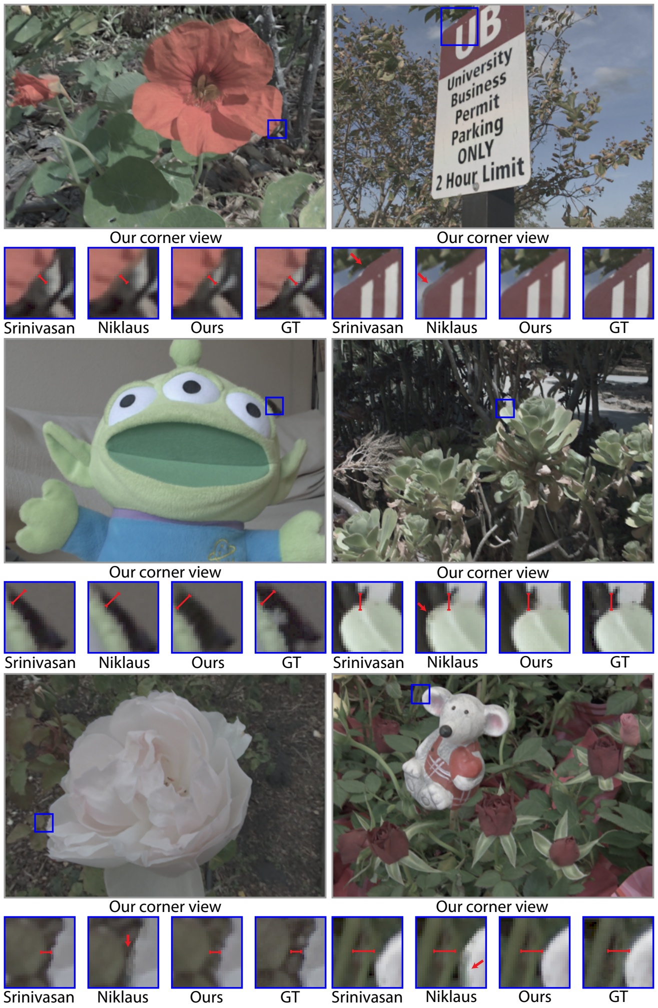


Fig. 6. Comparison against other approaches on six challenging scenes. The red bars are of the same length and show the distance between a foreground and background object.

We perform the comparisons on three different tests sets. Specifically, Stanford/Ours, Srinivasan et al., and Kalantari et al. [2016], containing 150, 100, and 30 light fields, respectively. Compared to the other approaches, our method produces significantly better results both in terms of PSNR and SSIM. Since both approaches by Srinivasan et al. and Niklaus et al. perform better using Wang et al.'s depth in almost all cases, we use this variant of their approaches for the rest of the comparisons in the paper. However, we refer to them as **Srinivasan** and **Niklaus** for simplicity.

We also compare our approach with VMPI representation (8 planes), against ours with MPI representation using 8, 16, and 32 planes in Table 3. Our VMPI representation with 8 planes produces better results than all the MPI variations. The gap in quality is even larger for synthesized 15 × 15 light fields. See Fig. 3 for visual comparison of VMPI and MPI representations.

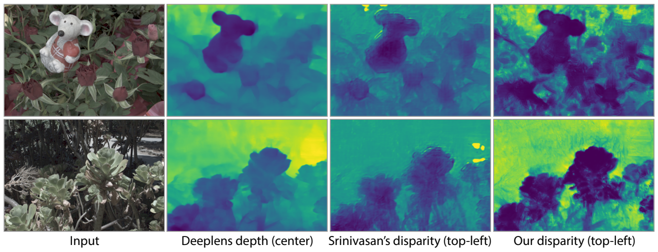


Fig. 7. We compare our disparity at the novel view against Srinivasan et al.'s disparity and Wang et al.'s estimated depth at the center view on two scenes from Fig. 6.

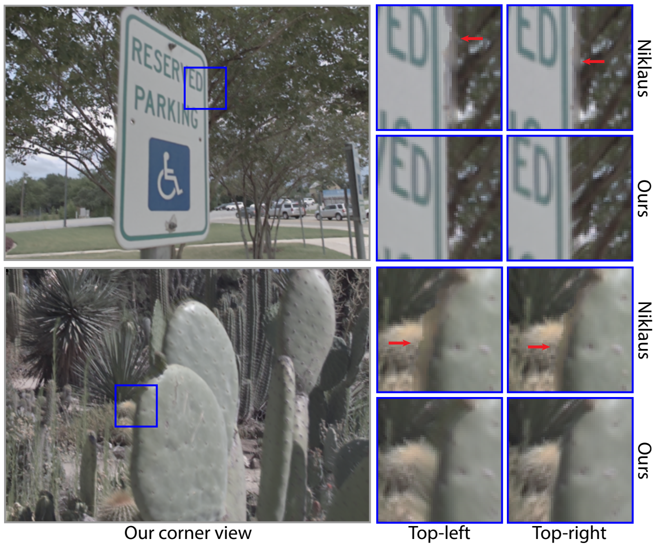


Fig. 8. Comparison against Niklaus et al.'s method for synthesizing 15 × 15 light fields on two challenging scenes.

Moreover, we compare our approach against our method with just the single visible network. As can be seen, quantitatively, the differences are minor as the occluded areas are typically only small regions around occlusion boundaries. However, as shown in Fig. 4, our approach with the two networks significantly improves the the results qualitatively.

4.2 Qualitative Comparison to Other Approaches

Here, we compare our approach against other methods on a variety of scenes. We first show comparisons on 8 × 8 light fields because of the restriction of Srinivasan et al.'s approach. In Fig. 6, we show comparison on six complex scenes captured with a Lytro Illum camera selected from the three datasets, discussed in the previous section. Here, we use the center view of a light field as the input to synthesize the full light field.

Srinivasan et al.'s approach is not able to properly estimate the disparity at the novel view even with the depth as the input, creating results with incorrect parallax, as indicated with the red bars. Niklaus et al. directly use the input depth to forward warp the input image. However, this depth is relative and, thus, this approach still generates incorrect parallax in most cases, as indicated by the red bars. Moreover, it cannot properly handle the object boundaries in some cases, as shown by the arrows. Furthermore, their forward warping scheme slightly overblurs the results in some cases (see supplementary video). In contrast, our approach generates high-quality results that are reasonably close to the ground truth.

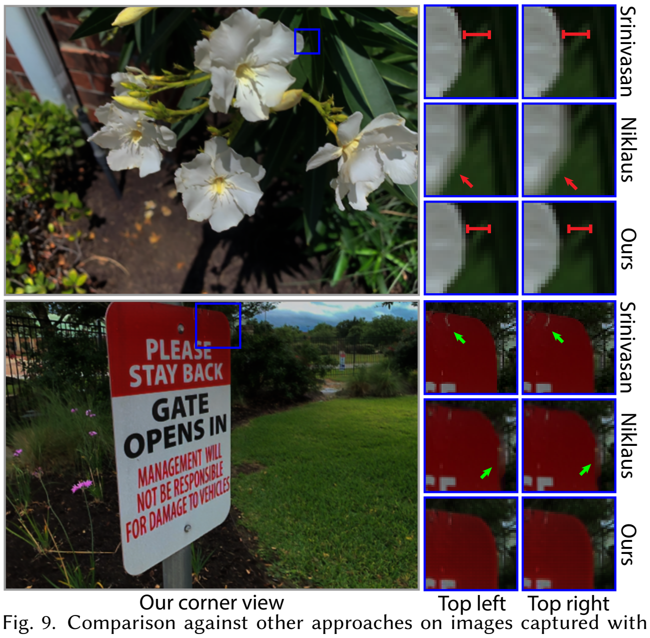


Fig. 9. Comparison against other approaches on images captured with standard cameras. We show insets of two synthesized corner views for each image. The red lines for each method have the same length and are used to better show the distance between a foreground and a background object in the two views. Our method correctly synthesizes both the foreground and background regions without objectionable artifacts.

In Fig. 7, we compare our estimated disparity against Srinivasan et al. and Wang et al.'s depth for two scenes from Fig. 6 (the bottom two on the right column). As seen, our disparity computed on the VMPI layers is overall sharper than the input depth by Wang et al. [2018b]. This is in part the reason for Niklaus et al.'s problem around occlusion boundaries. Note that, we also generated Niklaus et al.'s results with their depth estimator, but those were of lower quality. Moreover, although both our approach and Srinivasan et al. use Wang et al.'s depth as the input in this case, our disparity is considerably better than theirs. This is mainly because their network is trained to generate a disparity map for each view by maximizing the quality of that particular view. Therefore, each novel view image during the training process, provides a supervision for only the corresponding disparity map. In contrast, we use the same representation to generate all the novel view images. So each novel view image during training, provides a supervision for the same representation.

Furthermore, we compare our approach with Niklaus et al.'s method on synthesized 15 × 15 light fields, in the Fig. 8. As seen, unlike the result of Niklaus et al., ours do not suffer from the objectionable artifacts around occlusion boundaries. We also compare our approach against the other methods on two images captured with standard cameras in Fig. 9. Note that in these cases, ground truth novel view images are not available. Srinivasan et al. is not able to reconstruct the top scene with proper parallax. Moreover, in their results for the scene in the bottom row, the texture on the

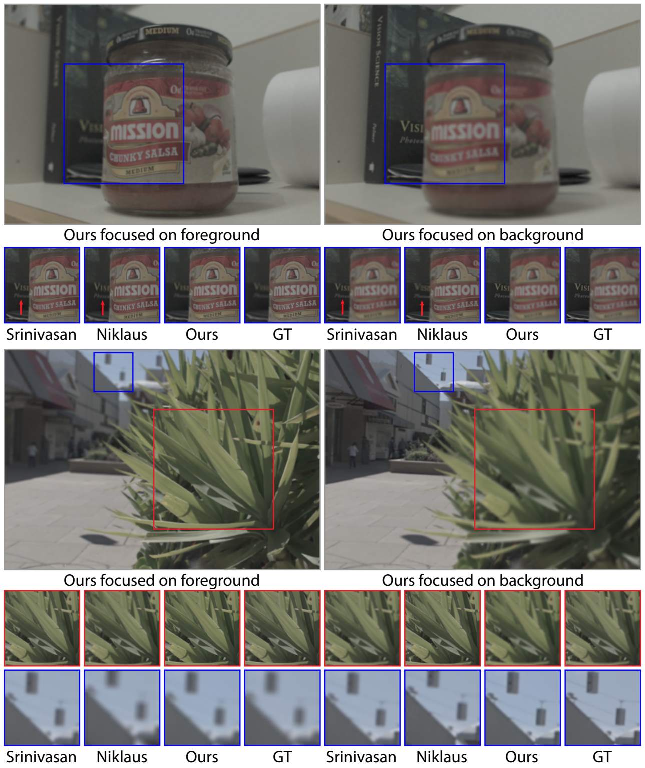

Ours focused on foreground
Ours focused on background
Srinivasan
Niklaus
Ours
GT
Srinivasan
Niklaus
Ours
GT
Ours focused on foreground
Ours focused on background
Srinivasan
Niklaus
Ours
GT
Srinivasan
Niklaus
Ours
GT

Fig. 10. Comparison of our refocusing results against the other methods. Our method produces results that are closer to the ground truth than the other methods.

sign moves between the two views because of inaccurate disparity. Furthermore, the results of Niklaus et al.'s approach for both scenes contain ghosting artifacts around the boundaries. Our method is able to produce reasonable results in both cases.

4.3 Comparison to Other Methods on Refocusing

Our synthesized light fields can be used to generate refocused images, as shown in Fig. 11. Here, we demonstrate the ability of our approach to focus on different areas of the scene by showing two refocused images for each scene. Although Srinivasan et al. handles the foreground regions, their two refocused images in the background areas have the same blurriness. This is mainly because their estimated disparity in the background areas is often inaccurate, as seen in Fig. 7 (see discussion in Sec. 4.2). Niklaus et al. produce results with incorrect defocus in both cases as they directly use the input depth which is relative and does not match the disparity of the ground truth. Our method produces considerably better results that are similar to the ground truth.

5 LIMITATIONS AND FUTURE WORK

Single image light field reconstruction is challenging and underconstrained and, thus, our results do not perfectly match with ground truth in all cases. Despite that, our synthesized light fields are visually pleasing and contain fewer objectionable artifacts than the other approaches. Moreover, in some cases the estimated depth map by Wang et al.'s approach is inaccurate and, thus, our approach is not able to synthesized high-quality novel views. However, as shown in Fig. 7, our VMPI still improves upon the input depth.

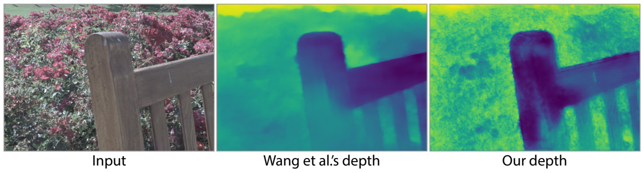


Fig. 11. Here, the input depth by Wang et al. [2018b] is inaccurate. Our method improves upon this input depth, but still cannot correctly estimate the disparity of the wooden bench. See supplementary video for the synthesized views and comparison to the other methods.

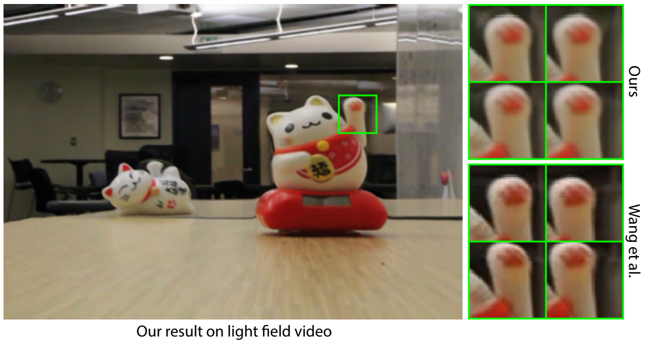


Fig. 12. On the left, we show a corner view of our synthesized light field frame from an input 2D video. On the right, we show an inset of all our synthesized corner views and compare them against Wang et al.'s approach [2017]. Note that, the approach by Wang et al. uses an additional light field video with a low frame rate as the input. Despite that, we produce light field frames with comparable quality. However, our light field video shows slight flickering as we synthesize each frame independently (see supplementary video).

Finally, our system can be used to synthesize a light field video from a standard video. This can be done by using our networks to synthesize a light field image for each frame of the video. While our synthesized light field frames are of high-quality (see Fig. 12), the video is not temporally coherent (see supplementary video). In the future, we would like to improve the quality of the synthesized videos, by enforcing the network to synthesize temporally coherent frames through a specially designed loss function.

6 CONCLUSION

We present a learning-based approach for synthesizing a light field from a single 2D image. We do this by introducing a new layered scene representation, called variable multiplane image (VMPI), where the disparity of each layer is inferred from the input image. Moreover, we propose to handle the visible and occluded regions separately, through the two parallel networks, to reduce ghosting artifacts in the occluded areas. In our system, each network synthesizes a novel view image and the two images are fused using a soft visibility mask to produce the final image. We introduce a light field dataset containing over 2,000 unique scenes for training. We show through extensive experiments that our approach produces better results than the other methods on view synthesis and refocusing.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments. This work was funded in part by a TAMU T3 grant 246451.

REFERENCES

Gaurav Chaurasia, Sylvain Duchene, Olga Sorkine-Hornung, and George Drettakis. 2013. Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG)* 32, 3 (2013), 1–12.

Qifeng Chen and Vladlen Koltun. 2017. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1511–1520.

Inchang Choi, Orazio Gallo, Alejandro Troccoli, Min H Kim, and Jan Kautz. 2019. Extreme View Synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*. 7781–7790.

X. Cun, F. Xu, C. Pun, and H. Gao. 2019. Depth-Assisted Full Resolution Network for Single Image-Based View Synthesis. *IEEE Computer Graphics and Applications* 39, 2 (March 2019), 52–64.

Donald G. Dansereau, Bernd Girod, and Gordon Wetzstein. 2019. LiFF: Light Field Features in Scale and Depth. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE.

Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. 2019. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters* 125 (2019), 333–340.

Simon Evain and Christine Guillemot. 2019. A Lightweight Neural Network for Monocular View Generation with Occlusion Handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2019), 1–14.

John Flynn, Michael Broxton, Paul Debevec, Matthew DuVall, Graham Fyffe, Ryan Overbeck, Noah Snavely, and Richard Tucker. 2019. DeepView: View synthesis with learned gradient descent. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2367–2376.

John Flynn, Ivan Neulander, James Philbin, and Noah Snavely. 2016. Deepstereo: Learning to predict new views from the world's imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5515–5524.

Yoav HaCohen, Eli Shechtman, Dan B Goldman, and Dani Lischinski. 2011. Non-rigid dense correspondence with applications for image enhancement. *ACM Transactions on Graphics (TOG)* 30, 4 (2011), 70.

Peter Hedman, Suhib Alsisan, Richard Szeliski, and Johannes Kopf. 2017. Casual 3D photography. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 1–15.

Peter Hedman and Johannes Kopf. 2018. Instant 3d photography. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–12.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* (2015).

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

Diederick P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Miaomiao Liu, Xuming He, and Mathieu Salzmann. 2018. Geometry-aware deep network for single-image novel view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4616–4624.

Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. 2019. Local Light Field Fusion: Practical View Synthesis with Prescriptive Sampling Guidelines. *ACM Transactions on Graphics (TOG)* 38, 4, Article 29 (July 2019), 14 pages.

Simon Niklaus, Long Mai, Jimei Yang, and Feng Liu. 2019. 3D Ken Burns Effect from a Single Image. *ACM Transactions on Graphics (TOG)* 38, 6, Article Article 184 (Nov. 2019), 15 pages.

Kyle Olszewski, Sergey Tulyakov, Oliver Woodford, Hao Li, and Linjie Luo. 2019. Transformable Bottleneck Networks. *arXiv preprint arXiv:1904.06458* (2019).

Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander C Berg. 2017. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3500–3509.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*. 8024–8035.

Eric Penner and Li Zhang. 2017. Soft 3D reconstruction for view synthesis. *ACM Transactions on Graphics (TOG)* 36, 6 (2017), 235.

Thomas Porter and Tom Duff. 1984. Compositing digital images. In *ACM Siggraph Computer Graphics*, Vol. 18. ACM, 253–259.

Konstantinos Rematas, Chuong H Nguyen, Tobias Ritschel, Mario Fritz, and Tinne Tuytelaars. 2016. Novel views of objects from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8 (2016), 1576–1590.

Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 2020. 3D Photography using Context-aware Layered Depth Inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8028–8038.

Pratul P Srinivasan, Richard Tucker, Jonathan T Barron, Ravi Ramamoorthi, Ren Ng, and Noah Snavely. 2019. Pushing the Boundaries of View Extrapolation with Multiplane Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 175–184.

Pratul P Srinivasan, Tongzhou Wang, Ashwin Sreelal, Ravi Ramamoorthi, and Ren Ng. 2017. Learning to synthesize a 4d rgbd light field from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*. 2243–2251.

Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. 2015. Single-view to Multi-view: Reconstructing Unseen Views with a Convolutional Network. *CoRR* abs/1511.06702 (2015).

Richard Tucker and Noah Snavely. 2020. Single-View View Synthesis with Multiplane Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 551–560.

Shubham Tulsiani, Richard Tucker, and Noah Snavely. 2018. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 302–317.

Lijun Wang, Xiaohui Shen, Jianming Zhang, Oliver Wang, Zhe Lin, Chih-Yao Hsieh, Sarah Kong, and Huchuan Lu. 2018b. DeepLens: Shallow Depth of Field from a Single Image. *ACM Transactions on Graphics (TOG)* 37, 6, Article 245 (Dec. 2018), 11 pages.

Ting-Chun Wang, Jun-Yan Zhu, Nima Khademi Kalantari, Alexei A Efros, and Ravi Ramamoorthi. 2017. Light field video capture using a learning-based hybrid imaging system. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 133.

Yunlong Wang, Fei Liu, Zilei Wang, Guangqi Hou, Zhenan Sun, and Tieniu Tan. 2018a. End-to-end view synthesis for light field imaging with pseudo 4DCNN. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 333–348.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. 2020. Synsin: End-to-end view synthesis from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7467–7477.

Gaochang Wu, Mandan Zhao, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. 2017. Light field reconstruction using deep convolutional network on EPI. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6319–6327.

Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. 2016. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*. 1696–1704.

Jimei Yang, Scott E Reed, Ming-Hsuan Yang, and Honglak Lee. 2015. Weakly-supervised disentangling with recurrent transformations for 3d view synthesis. In *Advances in Neural Information Processing Systems*. 1099–1107.

Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. 2018. Stereo Magnification: Learning View Synthesis Using Multiplane Images. *ACM Transactions on Graphics (TOG)* 37, 4, Article 65 (July 2018), 12 pages.

Tinghui Zhou, Shubham Tulsiani, Weilun Sun, Jitendra Malik, and Alexei A Efros. 2016. View synthesis by appearance flow. In *European Conference on Computer Vision*. Springer, 286–301.