Learning-Based View Synthesis for Light Field Cameras

Nima Khademi Kalantari\textsuperscript{1} Ting-Chun Wang\textsuperscript{2} Ravi Ramamoorthi\textsuperscript{1}

\textsuperscript{1}University of California, San Diego \textsuperscript{2}University of California, Berkeley

Abstract

With the introduction of consumer light field cameras, light field imaging has recently become widespread. However, there is an inherent trade-off between the angular and spatial resolution, and thus, these cameras often sparsely sample in either spatial or angular domain. In this paper, we use machine learning to mitigate this trade-off. Specifically, we propose a novel learning-based approach to synthesize new views from a sparse set of input views. We build upon existing view synthesis techniques and break down the process into disparity and color estimation components. We use two sequential convolutional neural networks to model these two components and train both networks simultaneously by minimizing the error between the synthesized and ground truth images. We show the performance of our approach using only four corner sub-aperture views from the light fields captured by the Lytro Illum camera. Experimental results show that our approach synthesizes high-quality images that are superior to the state-of-the-art techniques on a variety of challenging real-world scenes. We believe our method could potentially decrease the required angular resolution of consumer light field cameras, which allows their spatial resolution to increase.

Keywords: view synthesis, light field, convolutional neural network, disparity estimation

Concepts: Computing methodologies $\rightarrow$ Image manipulation; Computational photography;

1 Introduction

Light fields provide a rich representation of real-world scenes, enabling exciting applications such as refocusing and viewpoint change. Generally, they are obtained by capturing a set of 2D images from different views [Levoy and Hanrahan 1996; Wilburn et al. 2005] or using a microlens array [Adelson and Wang 1992; Ng et al. 2005; Georgiev et al. 2006]. The early light field cameras required custom-made camera setups which were bulky and expensive, and thus, not available to the general public. Recently, there has been renewed interest in light field imaging with the introduction of commercial light field cameras such as Lytro [2016] and Ray/Trix [2016]. However, because of the limited resolution of the sensors, there is an inherent trade-off between angular and spatial resolution, which means the light field cameras sample sparsely in either the angular or spatial domain. For example, Pelican cameras [Pelican Imaging 2016] have an array of $8 \times 8$ cameras.

To mitigate this problem, we propose a learning-based approach to synthesize novel views from a sparse set of input views captured using consumer light field cameras. Inspired by the recent success of deep learning in a variety of applications, such as image denoising [Burger et al. 2012], super-resolution [Dong et al. 2014], and deblurring [Sun et al. 2015], we propose to use convolutional neural networks (CNN) to predict novel views using the sparse input views and the position of the novel view in the light field. However, the major challenge is that training a single end-to-end CNN for this task is difficult, producing novel views that are quite blurry, as shown in Fig. 6.

Existing view synthesis approaches [Chaurasia et al. 2011; Chaurasia et al. 2013; Wanner and Goldluecke 2014] typically first estimate the depth at the input views and use it to warp the input images to the novel view. They then combine these images in a specific way (e.g., by weighting each warped image [Chaurasia et al. 2013]) to obtain the final novel view image. To make the learning more tractable, we build upon these methods and break down the task into disparity\textsuperscript{1} and color estimation components. The main contribution of our work is to use machine learning to model these two components and train both models by directly minimizing the error between the synthesized and ground truth images. In our system,

\textsuperscript{1}Because of the regularity of camera positions in structured light fields, depth and disparity are closely related and we use them interchangeably.
we use two sequential CNNs for estimating the disparity and the final pixel colors. Since our disparity estimation CNN is trained to directly minimize the synthesis error, our estimated disparities are more suitable for this application than existing disparity estimation techniques [Wanner and Goldluecke 2012; Tao et al. 2013; Wang et al. 2015; Jeon et al. 2015] (see Fig. 10). Moreover, since we train our system on the light fields generated by consumer light field cameras, it learns to model the noise and other inaccuracies of these cameras. Therefore, our method produces better results than the state-of-the-art optimization-based approach of Wanner and Goldluecke [2014], as shown in Fig. 2.

We demonstrate the performance of our approach using only the four corner sub-aperture views from 8 × 8 light fields captured by the Lytro Illum camera (see Fig. 1). Experimental results demonstrate that our method outperforms state-of-the-art schemes on challenging cases. Our method is two orders of magnitude faster than the recent learning-based DeepStereo method of Flynn et al. [2016], taking only 12.3 seconds to synthesize an image from four input views of size 541 × 376. Our system could potentially be used to decrease the required angular resolution of current cameras, which allows their spatial resolution to increase. Another application of our approach is to increase the baseline of current cameras and use our method on a subset of four angular views to synthesize the in between views. In summary, we make the following contributions:

- We present the first machine learning approach for view synthesis using consumer light field cameras. Our system consists of disparity and color estimation components which we model using two sequential CNNs. Note that although CNNs have been recently used for light field super-resolution [Yoon et al. 2015] and depth estimation [Heber and Pock 2016], these methods are not able to directly synthesize novel views at arbitrary locations.

- The output of our first network is disparity and typically we would need ground truth disparities to train this network.

However, we show how to train both networks simultaneously by directly minimizing the error between the synthesized and ground truth images.

- Since we train our disparity estimator in this way, our disparities are suitable for the view synthesis application. To the best of our knowledge, our method is the first to propose a disparity estimator which is specifically designed for this application.

### 2 Related Work

The problem of the light field’s limited resolution has been extensively studied in the past and several powerful methods for increasing the resolution in both angular [Levin and Durand 2010; Shi et al. 2014; Wanner and Goldluecke 2014] and spatial [Bishop et al. 2009; Cho et al. 2013] domains have been proposed. For brevity, we only focus on the approaches that are designed for angular super-resolution. We start by reviewing the algorithms that specifically work for light fields and then explain the approaches that perform view synthesis for general scenes and objects.

#### Light Field Super-resolution

Levin and Durand [2010] use a prior based on the dimensionality gap to reconstruct the full 4D light field from a 3D focal stack sequence. Shi et al. [2014] leverage sparsity in the continuous Fourier spectrum to reconstruct a dense light field from a 1D set of view points. Schedl et al. [2015] reconstruct a full light field using multidimensional patches from a sparse set of input views. These methods require the input samples to be captured with a specific pattern and are not able to synthesize novel views at arbitrary positions. Marwah et al. [2013] propose a dictionary-based approach to reconstruct light fields from a coded 2D projection. However, their method requires the light fields to be captured in a compressive way.

Mitra and Veeraraghavan [2012] introduce a patch-based approach where they model the light field patches using a Gaussian mixture model. However, this method is not robust against noise, and struggles on low-quality images taken with commercial light field cameras. Zhang et al. [2015] propose a phase-based approach to reconstruct light fields. However, their method is limited since it is designed for a micro-baseline stereo pair. Moreover, their approach...
is iterative, which is often slow and prevents its usage in practice. Yoon et al. [2015] perform spatial and angular super-resolution on light fields using convolutional neural networks (CNN). However, their method can only increase the resolution by a factor of two, and is not able to synthesize views at arbitrary locations. Layered patch-based synthesis has been proposed by Zhang et al. [2016] for various light field editing applications. Although they show impressive results for applications like hole-filling and reshuffling, their approach has limited performance for view synthesis and is not able to handle challenging cases as shown in Fig. 9.

Recently, Wanner and Goldluecke [2014] proposed an optimization approach to reconstruct images at novel views from an input light field. Given the depth estimates at the input views, they reconstruct novel views by minimizing an objective function which maximizes the quality of the final results. Although their method produces reasonable results on dense light fields, for sparse input views, it often produces results with tearing, ghosting, and other artifacts as shown in Fig. 2. We believe this is because of two main reasons. First, they estimate the disparity at the input views as a preprocess, independently of the view synthesis process. However, even state-of-the-art light field disparity estimation techniques [Wang et al. 2015; Jeon et al. 2015] are not typically designed to maximize the quality of synthesized views, and thus, they are not suitable for this application. Second, Wanner and Goldluecke’s approach has limited performance when the images are captured under ideal conditions. However, in practice, the images from consumer light field cameras are usually noisy and suffer from optical distortions.

View Synthesis for Scenes – View synthesis has a long history in both vision and graphics. One category of approaches [Eisemann et al. 2008; Goesele et al. 2010; Chaurasia et al. 2011; Chaurasia et al. 2013] synthesizes novel views of a scene in a two-step process. These methods first estimate the depth at the input views and use the depth to warp the input images to the novel view. They then produce the final image by combining these warped images. These approaches typically use multi-view stereo algorithms (e.g., PMVS by Furukawa et al. [2010]) to estimate depth and are not suitable for light fields with a narrow baseline. In our system, we also have depth and color estimation components. However, unlike these approaches, we use machine learning to model these two components. Furthermore, inspired by Fitzgibbon et al.’s approach [2003], we train both our disparity and color estimation models by directly minimizing the appearance error.

Another common approach is to synthesize images without explicitly estimating the geometry. For example, Mahajan et al. [2009] propose to move the gradients in the input images along a specific path to reconstruct the image at a novel view. Shechtman et al. [2010] propose a patch-based optimization framework to reconstruct images at novel views. However, these approaches are not able to utilize all the information available in light fields since they work on only two input images.

DeepStereo – Flynn et al. [2016] has recently proposed a deep learning method to perform view synthesis on a sequence of images with wide baselines. They first project the input images on multiple depth planes. They then estimate the pixel color and weight of the image at each depth plane from these projected images. Finally, they compute a weighted average of the estimated pixel colors to obtain the final pixel color. Comparing to this approach, our system has several key differences. First, our method is specifically designed for light fields, which have much narrower baselines and more regular camera positions. Second, unlike their approach, our system explicitly estimates the disparity which could potentially be used in other applications. Finally, our system is significantly faster than their method (several minutes vs. seconds). This shows the efficiency of our system, validating more practical usage.

View Synthesis for Objects – Since the recent release of large datasets of 3D shape models, synthesizing object views from a single image has become popular. Kholgade et al. [2014] transfer texture from the corresponding 3D model to render novel views of an object. However, manual annotation is required to specify the corresponding 3D model and its placement in the image. Su et al. [2014] resolve this limitation by selecting several similar models in the dataset and then interpolating between them. However, these methods heavily rely on the retrieval process and become vulnerable when a similar model cannot be found.

Recently, several algorithms have approached this problem by utilizing deep learning. Dosovitskiy et al. [2015] train a CNN which can render images of chairs once a graphics code containing the rendering details is given. Yang et al. [2015] expand this work and decode the implicit rendering information from the input image instead of representing it explicitly as the graphics code. They then apply the desired transformation and render the new view. Tatarchenko et al. [2015] also adopt a similar approach, but do not explicitly decouple the identity and the pose. Zhou et al. [2016] train a CNN to estimate appearance flow which is then used to warp the input image to the novel view. These methods are specifically designed to work on objects and do not work well on general scenes. Furthermore, they only use a single image, and thus, are not able to utilize all the images in light fields.

3 Proposed Learning-Based Algorithm

Given a sparse set of input views $L_{p_1}, \ldots, L_{p_N}$ and the position of the novel view $q$, our goal is to estimate the image at the novel view $L_q$. Formally, we can write this as:

$$L_q = f(L_{p_1}, \ldots, L_{p_N}, q),$$

(1)

where $p_i$ and $q$ refer to the $(u, v)$ coordinates of the input and novel view, respectively. Here, $f$ is a function which defines the relationship between the input views and the novel view. This relationship is typically very complex as it requires finding connections between all the input views, and collecting appropriate information from each image based on the position of the novel view. Inaccuracies such as noise and optical distortions in consumer light field cameras further add to the complexity of this relationship.

Therefore, we propose to learn this relationship. Inspired by the recent success of deep learning in a variety of applications, we propose to use convolutional neural networks (CNN) as our learning model. A straightforward way to do so is to directly model the function $f$ with a CNN. In this case, the CNN takes the input views as well as the position of the novel view and outputs the image at the novel view. However, as shown in Fig. 6, this naïve solution often produces blurry results. This is mainly due to the fact that the relationship is complex and requires the network to find connections between distant pixels, which makes the training difficult.

We make the training more tractable by following the pipeline of existing view synthesis techniques [Chaurasia et al. 2011; Chaurasia et al. 2013] and breaking the system down into disparity and color estimation components. Our main contribution is to use machine learning to model each component and train both models simultaneously by minimizing the error between the synthesized and ground truth images (see Sec. 3.3). In our system, we first estimate the disparity at the novel view from a set of features extracted from the sparse set of input views:

$$D_q = g_d(K),$$

(2)

where $D_q$ is the estimated disparity at the novel view, $K$ represents a set of features including the mean and standard deviation of warped images at different disparity levels (see Sec. 3.1). Moreover, $g_d$ defines the relationship between the input features and the disparity which we model using a CNN. The estimated disparity is then used to warp the input images to the novel view. Specifically, we perform a backward warp by sampling the input images based
on the disparity at the novel view (see Eq. 4). Finally, we estimate the image at the novel view using a set of input features including all the warped images, the estimated disparity, and the position of the novel view:

\[ L_q = g_c(H), \quad (3) \]

where \( H \) represents our feature set and \( g_c \) defines the relationship between these features and the final image. The overview of our system is shown in Fig. 3. In the next sections we describe the disparity estimator (Eq. 2) and the color predictor (Eq. 3) in detail.

### 3.1 Disparity Estimator

The goal of this component is to estimate the disparity at the novel view \( D_q \). For every pixel of the novel view image, this disparity points to the corresponding pixel in each input view:

\[ \hat{L}_{pi}(s) = L_{pi} \left[ s + (p_i - q) \right] D_q(s), \quad (4) \]

where \( s \) is a vector containing the pixel position in the \( x \) and \( y \) directions. Moreover, \( p_i \) and \( q \) are also vectors containing the position of input and novel views in the \( u \) and \( v \) directions. Here, \( L_{pi} \) is the image obtained by backward warping the input view \( L_{pi} \) using the disparity \( D_q \). If the disparity is accurate, it will point to the correct pixel in the input images, and thus, all the warped images would have the same color at each pixel. However, the disparity is not known a priori and we need to estimate it first.

To estimate the disparity, we first warp (backward) all the input images to the novel view using a set of predefined disparity levels \( d_1, \cdots, d_L \) as follows:

\[ L_{pi}^d(s) = L_{pi} \left[ s + (p_i - q) \right] d_i, \quad (5) \]

where \( i \in \{1, \cdots, N\} \) and \( l \in \{1, \cdots, L\} \). In our implementation we use \( L = 100 \) disparity levels in the range \([-21, 21]\) pixels. We use ideas from the depth estimation approach of Tao et al. [2011; Chaurasia et al. 2013; Wanner and Goldluecke 2014] have a specific way of combining these warped images and generating the final image. For example, Chaurasia et al. [2013] obtain the final image by computing the weighted average of all the warped images. However, these approaches are usually simple and do not properly model the relationship between the warped and final synthesized images which is complex because of occlusion.

In contrast, we propose to learn this relationship. We estimate the final image from a set of input features including all the warped images, the estimated disparity, and the position of the novel view. Specifically, our feature vector is \( H = \{L_{p_1}, \cdots, L_{p_N}, D_q, q\} \). Note that the disparity is useful to detect the occlusion boundaries and collect informative information from the warped images near these regions. Moreover, the position of the novel view can potentially be used to weight a particular image more in reconstructing the novel view. For example, if \( q \) is close to \( p_1 \), \( L_{p_1} \) should be heavily used in reconstructing the novel view at position \( q \). Although we do not explicitly model the occlusion, our system learns to reconstruct the final image by relying on the images with valid information in the occluded regions.

Here, we use a similar deep network as in Fig. 5 with different number of inputs and outputs. In this case, our input has \( 3N + 3 \) channels and the output is an RGB image which has 3 channels. In the next section, we discuss the details of training our system.

### 3.2 Color Predictor

The goal of this component is to estimate the final color using the disparity, estimated by the first CNN. The estimated disparity can be used to simply warp the input views to the novel view using Eq. 4. Existing view synthesis techniques [Chaurasia et al. 2011; Chaurasia et al. 2013; Wanner and Goldluecke 2014] have a specific way of combining these warped images and generating the final image. For example, Chaurasia et al. [2013] obtain the final image by computing the weighted average of all the warped images. However, these approaches are usually simple and do not properly model the relationship between the warped and final synthesized images which is complex because of occlusion.

In contrast, we propose to learn this relationship. We estimate the final image from a set of input features including all the warped images, the estimated disparity, and the position of the novel view. Specifically, our feature vector is \( H = \{L_{p_1}, \cdots, L_{p_N}, D_q, q\} \). Note that the disparity is useful to detect the occlusion boundaries and collect informative information from the warped images near these regions. Moreover, the position of the novel view can potentially be used to weight a particular image more in reconstructing the novel view. For example, if \( q \) is close to \( p_1 \), \( L_{p_1} \) should be heavily used in reconstructing the novel view at position \( q \). Although we do not explicitly model the occlusion, our system learns to reconstruct the final image by relying on the images with valid information in the occluded regions.

Here, we use a similar deep network as in Fig. 5 with different number of inputs and outputs. In this case, our input has \( 3N + 3 \) channels and the output is an RGB image which has 3 channels. In the next section, we discuss the details of training our system.

### 3.3 Training

In order to synthesize high-quality images that are close to the ground truth, we train the networks by minimizing the \( L_2 \) distance between the synthesized and ground truth images:

\[ E = \sum_k \left( \hat{L}_{q,k} - L_{q,k} \right)^2, \quad (7) \]

where the summation is over the RGB channels, \( L_{q,k} \) is the ground truth image at the novel view, and \( \hat{L}_{q,k} \) is our estimated image which is obtained by Eqs. 2 and 3. In order to use a gradient descent based technique to minimize our energy function, we need to compute the derivative of the error in Eq. 7 with respect to both networks’ weights, i.e., \( \partial E/\partial w_d \) and \( \partial E/\partial w_c \), where \( w_d \) and \( w_c \) are vectors and refer to all the weights of the disparity and color estimator networks, respectively.

Since the color predictor network directly outputs the synthesized image, \( \partial E/\partial w_d \) can be easily computed as in standard backpropagation [Rumelhart et al. 1986]. For \( \partial E/\partial w_d \), we use the chain rule to break down the derivative into three terms as follows:

\[ \frac{\partial E}{\partial w_d} = \sum_{k=1}^{3} \left( \frac{\partial E}{\partial L_{q,k}} \frac{\partial L_{q,k}}{\partial D_q} \frac{\partial D_q}{\partial w_d} \right). \quad (8) \]
vector. Here, the first term is the derivative of the color predictor where the summation is over the individual channels of the feature warped images, and thus, few other features to generate the final image.

Since our error is quadratic, the first term can be easily calculated. The last term is the derivative of the disparity estimation network’s output with respect to its weights which can be calculated as usual [Rumelhart et al. 1986]. The middle term is the derivative of the final image with respect to the estimated disparity. Note that the disparity is used to generate a set of features $H$ (see Sec. 3.2). These features are then used by the color estimator network to produce the final image. Therefore, we have:

$$\frac{\partial L_q,k}{\partial D_q} = \sum_{i=1}^{3N+3} \frac{\partial L_q,k}{\partial H_i} \frac{\partial H_i}{\partial D_q}$$  \hspace{1cm} (9)$$

where the summation is over the individual channels of the feature vector. Here, the first term is the derivative of the color predictor network’s output with respect to its input and is straightforward to calculate. For the second term, we need to investigate each channel separately. The first $3N$ channels of our input feature vector $H$ are the warped images, and thus, $\frac{\partial H_i}{\partial D_q}$ is basically the derivative of the warping function in Eq. 4. Fortunately, since we use bicubic interpolation to compute the color values, this function is differentiable. For simplicity of the implementation, we numerically calculate this gradient. The feature at the next channel $H_{3N+1}$ is the estimated depth and its derivative is equal to one. Finally, the last two channels are the position of the novel view which are independent of the disparity, and thus, their gradient is equal to zero.

At every iteration of the training, we use these gradients to update both networks’ weights in the opposite direction of the gradients. We used a set of 100 light fields captured with the Lytro Illum camera in our training set. To handle a diverse test set, we ensured our training set contained a variety of different scenes including bicycles, cars, trees, and foliage (see supplementary materials). We captured most of these images ourselves, and obtained some of them from Raj et al.’s dataset [2016]. These light fields have angular resolution of $8 \times 8$ from which we only used the four corner sub-aperture images as our input. For each light field, we randomly selected four novel view positions from the original $8 \times 8$ grid. For each novel view position we extracted a set of features (see Eq. 6) and used the original captured image at that position as the ground truth image.

Since training on the full images is slow, we extracted patches of size $60 \times 60$ with a stride of 16 pixels from the full images. This resulted in over 100,000 patches which we used to train our system. Note that for every input patch, our system outputs a patch of size $36 \times 36$ (reduced size is due to convolutions). These output patches are then compared to the ground truth patches and the error at each pixel is backpropagated to train the networks. Therefore, in practice, we had more than 100,000,000 examples which we found to be sufficient to properly train both networks. We used mini-batches of size 20 to have the best trade-off between speed and convergence. We randomly initialized our networks’ weights using the Xavier approach [Glorot and Bengio 2010] and trained our system using the ADAM solver [Kingma and Ba 2014], with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a learning rate of 0.0001.
Figure 7: We compare our approach against the method of Wanner and Goldluecke [2014] using several state-of-the-art light field disparity estimation methods as its input. For each scene, we also show our estimated disparity, where regions with darker color are closer to the camera. The PSNR and structural similarity (SSIM) values are listed below each result. Note that some artifacts are hard to see in still images, and thus, we refer the readers to also view the supplementary video.
Comparison Against Other Approaches

We implemented our approach in MATLAB and used MatConvNet [Vedaldi and Lenc 2015] for implementing our networks. All the results shown here are generated on light fields captured with a Lytro Illum camera. The angular resolution of the captured light fields is $8 \times 8$ from which we only use the four corner sub-aperture images as our input to generate the full light field. Note that our approach can generate any in-between views. However, we only generate the $8 \times 8$ views to be able to compare them against the ground truth images. Here, we only show one synthesized image ($5, 5$) for each scene, but videos showing all the views can be found in the supplementary video.

We begin by comparing the result of modeling the process using a single CNN in Fig. 6. Here, the network directly models the relationship between the input images and the novel view (see Eq. 1). However, the relationship is complex and requires the network to often connect distant pixels, which makes the training difficult. As a result, when compared to our architecture containing two sequential CNNs, the result of the single CNN is blurry and contains artifacts. For example, the single CNN is not able to connect the pixels of the white truck in the input views, and thus, generates a result with ghosting artifacts.

Comparison Against a Single Network

Next, we compare our method against Wanner and Goldluecke’s approach [2014]. They first compute the disparity for each input view using an existing technique. They then use the disparities within an optimization framework to obtain the novel view by minimizing an objective function. We adopt several state-of-the-art light field disparity estimation methods to generate the disparities required for Wanner and Goldluecke’s method. Specifically, we use the approaches by Wanner and Goldluecke [2012], Tao et al. [2013], Wang et al. [2015], and Jeon et al. [2015]. We evaluate the results numerically, in terms of PSNR and structural similarity (SSIM) [Wang et al. 2004]. SSIM produces a value between 0 and 1, where 1 indicates perfect perceptual quality with respect to the ground truth.

Table 1 shows the average PSNR and SSIM values for all the methods on 30 test scenes. To properly evaluate our system on challenging cases, we used images of foliage and flowers in about half of our test set. Note that we had completely separate training and test sets and none of the test scenes were part of the training set (see supplementary materials). As seen, our approach produces results that are significantly better than other methods. We show four of these scenes in Fig. 7. The FLOWER 1 scene demonstrates a flower in front of a truck, a building, and a tree (on the right). The flower and the leaves have complex structure which makes it hard for the other approaches to accurately estimate the disparity at the boundaries. Therefore, their results often contain artifacts around the occlusion boundaries. However, our approach produces a plausible result which is reasonably close to the ground truth image. Note, for example, that only our approach is able to properly reconstruct the truck’s roof (green inset) and the highlight (blue inset).

Next, we examine the CARS scene showing a tree branch in front of a street. Despite the simplicity of the scene, other approaches often are not able to accurately estimate the disparity around the boundaries of the branch from only four input images. Therefore, their result contains tearing artifacts which can be specifically seen in the blue inset. Moreover, the method of Wanner and Goldluecke [2014], which is used to synthesize the novel view, does not model the inaccuracies of consumer light field cameras which usually appear as discoloration in the results (see the colorful pixels in the red inset). Note that only our approach is able to reconstruct all the details around the occlusion boundaries such as the thin vertical line in the red inset.

The FLOWER 2 scene contains a flower with complex structure in front of a street. Our method produces a reasonable result that is better than other approaches. Note that only our method is able to faithfully reconstruct the challenging area between the flower stem and petal (blue inset). Finally, the ROCK scene is difficult for all the other approaches. They often are not able to accurately estimate the disparity around the boundaries of the rock which results in tearing artifacts. Meanwhile, we produce better results than the other methods relative to the ground truth.

Overall, all the other approaches show tearing, ghosting, and other artifacts around the occlusion boundaries which are important areas for the view synthesis application. The main reason is that these methods are not specifically designed for this application, and thus, they often have inaccuracies around these boundaries. Moreover, Wanner and Goldluecke’s approach [2014], which is used for generating the novel views, assumes the images are captured under ideal conditions, while this is not the case for consumer light field cameras. Our method, on the other hand, produces plausible results which are reasonably close to the ground truth. Numerically, our results are significantly better than the other approaches.
Timing — Our method takes around 12.3 seconds to generate a novel view from four input images of resolution $541 \times 376$ on an Intel quad-core 3.4 GHz machine with 16 GB of memory and a GeForce GT 730 GPU. Specifically, it takes 5.5 seconds to extract the features, 5.1 seconds to evaluate the disparity estimation network, 0.2 seconds for warping the four input images to the novel view, and 1.5 seconds to evaluate the color predictor network.

Analyzing Our System — We evaluate the effect of each component in our system. Figure 10 compares our estimated disparity against other approaches for the SEAHORSE scene (shown in Fig. 2). Although the disparities produced by some of the other approaches have higher quality than ours, their disparity often has artifacts around the occlusion boundaries which are the most important regions for view synthesis. For example, part of the tree trunk and the highlight which should be visible in the novel view. Therefore, as shown by the red arrows, the left warped images contain artifacts in these regions. Our color predictor network produces high-quality results by collecting appropriate information from all the warped images.

We compare our method against other approaches on a challenging scene in Fig. 8. This scene contains a significant number of occluded regions which are generally difficult for view synthesis. Therefore, even our approach fails to synthesize a high-quality image in the difficult regions (see Fig. 12). However, our result is reasonable overall and significantly better than all the other approaches. Note that the leaves have thin structure and only our approach is able to properly reconstruct them without introducing artifacts in the background (green and blue insets).

We also compare our approach against the recent method of Zhang et al. [2016] in Fig. 9. Note that their approach needs some user interaction, while ours is fully automatic. Moreover, their method requires the center view, and thus, uses five input images (instead of four). Nevertheless, even with user interaction, their approach is able to properly reconstruct the scene into different depth layers, resulting in tearing artifacts. As a result, they often produce artifacts in these regions.

For example, the leaf in the left input views on the top set blocks part of the tree trunk and the highlight which should be visible in the novel view. Therefore, as shown by the red arrows, the left warped images contain artifacts in these regions. Our color predictor network produces high-quality results by collecting appropriate information from all the warped images.

Our method, on the other hand, can produce a reasonable disparity in these regions. Note that our method does not always produce accurate disparity. However, our inaccuracies usually happen in the regions that are not important for view synthesis. For example, part

\[ \text{PSNR: 29.48 SSIM: 0.956} \]

\[ \text{PSNR: 31.87 SSIM: 0.970} \]
We have presented a novel learning-based approach for synthesizing novel views from a sparse set of input views captured with a consumer light field camera. Our system consists of disparity and color estimator components which we model using two sequential convolutional neural networks. We show the result of our approach on a variety of scenes using only the four corner sub-aperture images captured with a Lytro Illum camera. Experimental results show that our method outperforms state-of-the-art approaches.

In the future, we would like to investigate the possibility of using our system for generating high dynamic range light fields from a set of views with different exposures. Moreover, it would be interesting to extend our system to work with any number of input views. We are also interested in improving the speed of our algorithm to possibly work at interactive rates or even real-time. Finally, there is potential to use our system along with light field compression schemes [Tong and Gray 2003; Girod et al. 2003] to increase the compression ratio by, for example, generating the novel views from a sparse set and compressing the differences.

**Acknowledgments**

We would like to gratefully thank Alexei Efros for valuable discussions. We also thank the Stanford Computational Imaging group for the light field dataset, some of which were used in our training set. Fang-Lue Zhang ran his algorithm on our scenes. This work was funded in part by a Berkeley Fellowship, ONR grant N00014152013, NSF grants 1451830 and 1617234, and the UC San Diego Center for Visual Computing, as well as support from Draper Lab, Nokia, and a Google Research Award.

**References**


