U-shaped Networks for Shape from Light Field

Stefan Heber\textsuperscript{1}
stefan.heber@icg.tugraz.at
Wei Yu\textsuperscript{1}
wei.yu@icg.tugraz.at
Thomas Pock\textsuperscript{1,2}
pock@icg.tugraz.at

\textsuperscript{1} Graz University of Technology
Graz, Austria
\textsuperscript{2} Austrian Institute of Technology
Vienna, Austria

Abstract

This paper presents a novel technique for Shape from Light Field (SfLF), that utilizes deep learning strategies. Our model is based on a fully convolutional network, that involves two symmetric parts, an encoding and a decoding part, leading to a u-shaped network architecture. By leveraging a recently proposed Light Field (LF) dataset, we are able to effectively train our model using supervised training. To process an entire LF we split the LF data into the corresponding Epipolar Plane Image (EPI) representation and predict each EPI separately. This strategy provides good reconstruction results combined with a fast prediction time. In the experimental section we compare our method to the state of the art. The method performs well in terms of depth accuracy, and is able to outperform competing methods in terms of prediction time by a large margin.

1 Introduction

In this paper we investigate the problem of estimating depth information for given Light Field (LF) data. This problem is also referred to as Shape from Light Field (SfLF). A LF \([\mathbb{L}, \mathbb{W}]\) is a 4D function that provides in addition to the spatial information, that corresponds to the information of a traditional 2D image, also directional information. The additional directional information includes information about the geometry of the observed scene, and thus gave rise to interesting applications, like for instance digital re-focusing \([\mathbb{E}, \mathbb{W}]\), digital viewpoint manipulation \([\mathbb{E}]\), or depth estimation \([\mathbb{E}, \mathbb{W}, \mathbb{O}, \mathbb{I}, \mathbb{N}, \mathbb{F}]\). All of these tasks are basically impossible to realize given a single traditional 2D image, that only provides the spatial intensity information.

A LF is commonly described using the so-called two-plane parametrization. This type of parametrization defines a ray by the intersection points of two parallel planes. Those planes are referred to as image plane \(\Omega \subseteq \mathbb{R}^2\) and lens plane \(\Pi \subseteq \mathbb{R}^2\). Thus in mathematical terms the LF is given as

\[
L : \Omega \times \Pi \rightarrow \mathbb{R}, \quad (p, q) \mapsto L(p, q),
\]

where \(p = (x, y)^\top \in \Omega\) and \(q = (\xi, \eta)^\top \in \Pi\) represent the spatial and directional coordinates.
There are different ways of visualizing the 4D LF. In this work we use the so-called Epipolar Plane Image (EPI) representation. In terms of Equation (1) an EPI is obtained by holding one spatial and one directional coordinate constant. For instance by choosing a certain $y$ and a certain $\eta$ we restrict the 4D LF to the 2D function

$$\Sigma_{y,\eta} : \mathbb{R}^2 \rightarrow \mathbb{R}, \quad (x, \xi) \mapsto L(x, y, \xi, \eta),$$

that defines a horizontal EPI. In a similar way one can also define vertical EPIs. The EPI representation can be considered as a 2D slice through the 4D LF, and it illustrates the linear characteristic of the LF space. See Figure 1(a) for an illustration.

In this work we aim at automatically converting EPIs to corresponding disparity images. Our approach, based on fully Convolutional Neural Networks (CNNs) \[20\], consists of processing an EPI with a series of convolution operations, that are able to detect line orientations. Knowing the line orientations allows to reconstruct the geometry of the observed scene. The kernels used for the involved convolutions are learned by leveraging a LF dataset that was recently presented in \[11\]. The proposed data-driven approach has two main advantages compared to prevailing methods: First, it allows to learn necessary heuristics from the training data to cope with artifacts due to, for instance, occlusion and aliasing. Secondly, the convolutions can be implemented efficiently on the GPU allowing for fast prediction times.

## 2 Related Work

One of the most important research topics in LF image processing is the development of efficient and reliable shape extraction methods. Those methods are the foundation of various applications, like for instance digital refocusing \[13, 22\], image segmentation \[30\], or super-resolution \[2, 29\], to name but a few. The main focus of research regarding Shape from Light Field (SfLF) lies on developing methods to accurately reconstruct the observed scene at depth discontinuities or occlusion boundaries. For this purpose various approaches have been proposed, including specialized multi-view stereo techniques \[3, 12\] and methods based on an EPI analysis \[6, 28\]. Wanner and Goldluecke \[6, 28\] used for example the 2D structure tensor to measure the direction of each position in the vertical and horizontal EPIs. The results are then fused using variational methods by incorporating additional global visibility constraints. In \[11\] Heber et al. proposed a variational multi-view stereo method based on...
a technique called Active Wavefront Sampling (AWS). Tao et al. [27] proposed a fusion method that uses correspondence and defocus cues. Chen et al. [3] introduced a bilateral consistency metric on the surface camera to indicate the probability of occlusions, which was further used for LF stereo matching. Heber and Pock [10] proposed a variational method, that shears the LF by applying a low-rank assumption, where the depth information is provided by the amount of shearing. Jeon et al. [14] proposed an algorithm for SfLF, that utilizes phase shift based subpixel displacements. In [11] Heber and Pock presented a method for SfLF that applies a conventional CNN in a sliding window fashion. Up to this point deep learning techniques were barely used in LF image processing. Utilizing trained models for SfLF is an interesting idea to address certain limitations of previous methods. On the one hand a trained model has the ability to learn how to handle occlusion and aliasing artifacts, and on the other hand a CNN also allows faster computation times.

In this paper we seize ideas presented in [11]. Furthermore this work also builds upon fully convolutional networks [20] and up-convolution-based approaches [4, 20, 32], i.e. the proposed network architecture consists of a contracting and an expanding path, that involve only convolutional layers. The former path compresses the information and simultaneously captures context, and the latter path extracts the information and up-samples it to the original size. The expanding path is more or less symmetric to the contracting path, yielding a u-shaped architecture, that can be trained in an end-to-end scheme.

3 Contribution

The proposed method is inspired by the method of Heber and Pock [11], that uses a conventional CNN in a sliding window fashion to predict depth information. They showed that CNNs have a large capacity to learn from data to predict the orientation of the lines in the EPIs. However, due to the sliding window approach, their method suffers from considerable high computational costs. Compared to [11] we were able to significantly reduce the computation time by predicting complete EPIs at once using u-shaped networks. Besides drastically reducing the prediction time the proposed network architecture also allows to reduce the errors in homogeneous regions, because the proposed model can overcome the patch-nature of the network proposed in [11]. Our experiments demonstrate that the proposed method is able to predict an entire 4D disparity field within a few seconds. Moreover, due to the fact that our network architecture does not include any fully connected layer, our method also allows to process LFs with varying resolutions.

4 Methodology

In this section we describe the methodology of the proposed approach. The success of the proposed CNN depends on leveraging a set of recent improvements, that include up-convolutions [20], no explicit pooling [26], and the Adam optimization method [15]. The section starts with a short introduction to CNNs, followed by the description of the used u-shaped network architecture. At the end of the section we provide details regarding the network training and the leveraged dataset.

Convolutional Neural Networks. In the late 1980s, Yann LeCun et al. [17, 18] introduced a special type of multi-layer Neural Networks (NNs), where weights are shared across layers. By sharing the weights they were able to resemble an important operation in signal
processing known as convolution, leading to the CNN architecture. A CNN consists of several layers, where the different layers are connected such that layer $l$ creates the input for layer $l+1$. Layer $l$ can be seen as a multi-channel image of size $H_l \times W_l \times C_l$, where $H_l$, $W_l$ and $C_l$ denote image height, image width, and number of channels of the $l$th layer, respectively. The first and last layer are called input and output layers, respectively. Hence their size also corresponds to the input size and to the desired output size. Successive layers are connected via a convolutional mapping with an additional additive bias term, i.e. each channel of the layer $l+1$ is defined as a convolution with a kernel of size $k_h \times k_w \times C_l$ followed by the addition of a constant bias, where $k_h$ and $k_w$ denote the kernel width and height.

Yann LeCun [18] introduced CNNs trained in a supervised manner via back-propagation. Since Krizhevsky et al. [16] utilized CNNs effectively for the task of large-scale image classification the popularity of CNNs or deep learning techniques increased drastically in the computer vision literature. Nowadays CNNs are especially popular in image classification and objection recognition [9, 24]. The entire field of deep learning flourishes with innovations, one after another. However, the exploration of 4D LF data by CNNs is still limited.

Network Architecture. In contrast to methods that use natural images we are not able to exploit existing trained networks, i.e. we opt for designing our network entirely from scratch. However, not relying on pre-trained networks also allows to better adapt the network structure to the problem at hand. The proposed network is a fully convolutional network consisting of a contracting part and an expanding part. The first part acts as an encoder, that spatially compresses the image and thus reduces the input data to an essential feature representation. The bottom part processes the essential features, before the expanding part of the network decodes the simple feature representation to an output disparity image. The encoding and decoding parts of the network are basically symmetric leading to an u-shaped network architecture. An overview of the network structure is depicted in Figure 2, where the encoding and decoding parts of the network are highlighted in purple and green, respectively.

The u-shaped network uses down and up-convolutional layers for the encoding and decoding part, respectively. A down-convolution layer is obtained by increasing the stride of the convolution, i.e. it only computes a subset of all positions. This decreases the spatial resolution of the following layer, and simultaneously increases the spatial support of all subsequent layers. To increase the image resolution again we use so-called up-convolutional lay-
The basic building block of the overall network is a convolutional layer followed by a Rectified Linear Unit (ReLU) non-linearity \( \sigma(x) = \max(0, x) \). We combine two convolutional layers to one level. For the convolutional layers within one level, we use padding to compensate for the kernel size. This ensures that the output of one level has the same size as the input. In the first part of the network we use three of those levels, where we use down-convolutional layers after each level to increase the spatial support of subsequent layers. At each down-sampling step we double the number of feature channels, except for the last level. The bottom part of the network consists of another level, that processes the compressed data. The decoding part of the network uses again three levels, but now we utilize up-convolutional layers before each level. Hence we up-convolve the whole coarse feature maps allowing to transfer high-level information to the fine prediction, and finally increase the image resolution back to the original size. All the involved convolutions use kernels of size \( 3 \times 5 \), except for the down and up-convolutional layers that use \( 3 \times 3 \) kernels. We also use so-called pinhole connections between the encoding and decoding part of the network, i.e. we concatenate the input of each level in the decoding part with the corresponding output feature map from the encoding path. We want to emphasize that the network structure involves only convolutional layers, i.e. we are not using any fully connected layers nor any pooling operations. A main advantage of avoiding fully connected layers is the ability to process EPIs of arbitrary resolutions.

**Dataset.** In order to train the proposed u-shape network a large amount of labeled training data is needed. Fortunately, we were allowed to use the synthetic dataset proposed in [11]. This dataset was generated using POV-Ray [23] and comes with highly accurate ground truth depth fields. Moreover the dataset also provides a random scene generator that allows to generate the desired amount of LFs. We render 200 LFs with a spatial resolution of \( 640 \times 480 \) and a directional resolution of \( 11 \times 11 \), out of which 150 are used to generate training data and 50 are used for testing.

**Data Augmentation.** Data augmentation [5, 16] is a common way to combat overfitting and to improve the generalization of the trained model. It basically allows the model to become invariant to certain predefined image deformations. We perform excessive data augmentation, including brightness changes, color changes, and additive Gaussian noise. We also flip the EPIs horizontally and vertically, where each flipping results in a sign change of the disparity map. Our augmentation procedure results in 8 times the original amount of image pairs. Although they are heavily correlated they allow to increase the robustness of
the trained model. Figure 3 provides some augmentation examples.

Network Training. While NNs learned with back-propagation have been around for several decades [25], only recently the computational power and data has been available to fully exploit this training technique [14]. In order to train the proposed u-shaped network we use the tensorflow framework [1], where we use Adam [15] as the optimization method to minimize the $\ell_1$ loss. Out of the 150 rendered LFs used for training we extract $2 \times 10^3$ EPIs. The extracted samples are then increased eightfold using data augmentation. To monitor overfitting we use a test set of $10 \times 10^3$ samples. In deep networks with many convolutional layers a good initialization of the weights is extremely important. Ideally the weights in the network should be initialized such that each feature map has approximately unit variance. This can be achieved by drawing the initial weights of a given node from a Gaussian distribution with standard deviation $\sqrt{2/N}$, where $N$ denotes the number of incoming nodes [8]. After initializing the weights as suggested in [8] we train our model for 400 epochs, where we use a mini-batch size of $2^8$ samples.

5 Experiments

We have performed an extensive analysis of our proposed model. We conducted synthetic and real world experiments. For the synthetic evaluation we used a recently presented LF dataset [11], where all LF scenes within the dataset have a directional resolution of $11 \times 11$, and a spatial resolution of $640 \times 480$. For the real world evaluation we used a LF captured with a Lytro camera as well as LFs from the Stanford Light Field Archive (SLFA). The used Lytro data provides a spatial resolution of $328 \times 328$ and a directional resolution of $7 \times 7$. LFs within the SLFA are captured using a multi-camera array [31] and contain $289$ views on a $17 \times 17$ grid. We trained a u-shaped network based on the description in Section 4, where we use the same model for all the presented experiments. To obtain the final result we predict the horizontal and vertical EPIs and take the pointwise average of the two predictions.

We compare our model against the following state-of-the-art SfLF methods [10, 11, 14, 27, 28]. The method by Wanner and Goldluecke [28] analyzes the EPIs using the 2D structure tensor, before combining the obtained information using a variational framework. Tao et al. [27] proposed a fusion method that uses correspondence and defocus cues. Both local cues are combined to a global depth estimate by using a Markov Random Field (MRF) model. Heber and Pock [11] proposed a variational multi-view stereo model based on low rank minimization. This model includes a matching term based on Robust Principal Component Analysis (RPCA), that can be interpreted as an all vs. all matching term. Jeon et al. [14] proposed an algorithm for SfLF, that utilizes phase shift based subpixel displacements. Besides the use of the phase shift theorem the algorithm is quite straightforward. They first calculate various cost volumes, that are processed using edge-preserving filtering, before extracting a disparity map based on the winner-takes-all strategy. To correct the obtained disparity map in weak textured regions they proceed with a multi-label optimization using graph cuts. At the end they refine the discrete disparity map to a continuous one using an iterative refinement scheme. In [11] Heber and Pock presented the first attempt to predict depth information for given LF data by utilizing deep learning strategies. Their network was trained in a sliding window setup to predict for each imaged scene point the orientation of the corresponding 2D hyperplane in the domain of the LF. This corresponds to estimating the line orientations in the horizontal and vertical EPIs simultaneously. They also use a 4D regularization step to overcome prediction errors in textureless or uniform regions, where they
Figure 4: Comparison to state-of-the-art methods on the synthetic POV-Ray dataset. The figure shows the center view of the LF, the color-coded ground truth, the results for five state-of-the-art SfLF methods [10, 11, 14, 27, 28], followed by the result of the proposed method.

use a confidence measure to gauge the reliability of the estimate. This additional regularization step is not used in the following comparison, because such a post processing step can also be applied to the prediction of the proposed model. The method of Heber and Pock [11] works well but has drawbacks due to the sliding window scheme. First, the per-patch nature disallows to account for global output properties, and second, it leads to higher computational costs compared to the proposed approach. In what follows we will first provide some synthetic evaluations before presenting qualitative real world results.

**Synthetic Evaluation.** We start with the synthetic evaluation. Figure 4 provides a comparison of different state-of-the-art methods. Note that for all methods that rely on precomputed cost volumes [10, 14, 27, 28], the number of labels is set to 200. Moreover we also set the necessary known disparity range for those methods based on the ground truth data. We can see that, despite the complexity of the scene, our model is able to predict accurate disparity results, that are on par with the competing methods. When comparing the results of the proposed model to the predictions obtained by the conventional CNN used in [11], we see that the proposed model provides better results in textureless regions. Also note that the proposed model is barely affected by depth discontinuities.

Quantitative results in terms of RMSE and MAE are presented in Table 1. The table also shows the percentage of pixels with a relative disparity error larger than 0.2% and 0.5%. Besides the various disparity errors the table also provides the computation times for estimating a disparity map for one sub-aperture view of the LF. Moreover we also indication if a GPU implementation was used or not. The presented results represent the average over the 50 LFs used for testing. We observe that the proposed model was able to accurately
Table 1: Quantitative results for various SfLF methods averaged over 50 synthetic LFs. The table provides the RMSE, MAE, the percentage of pixels with a relative disparity error larger than 0.2% and 0.5%, and the computational time of the method.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>RMSE</td>
<td>3.91</td>
<td>2.33</td>
<td>2.50</td>
<td>2.49</td>
<td>1.87</td>
<td>0.80</td>
</tr>
<tr>
<td>MAE</td>
<td>2.94</td>
<td>1.06</td>
<td>0.79</td>
<td>0.75</td>
<td>1.13</td>
<td>0.35</td>
</tr>
<tr>
<td>0.5%</td>
<td>22.00</td>
<td>16.32</td>
<td>8.47</td>
<td>9.64</td>
<td>17.96</td>
<td>7.34</td>
</tr>
<tr>
<td>0.2%</td>
<td>35.22</td>
<td>28.48</td>
<td>13.20</td>
<td>16.46</td>
<td>31.61</td>
<td>14.76</td>
</tr>
<tr>
<td>Time</td>
<td>3min 18s</td>
<td>23min 4s</td>
<td>4min 44s</td>
<td>2h 12min 30s</td>
<td>35s</td>
<td>2s</td>
</tr>
<tr>
<td>GPU</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

learn the characteristics of this dataset. Furthermore, we also see that the proposed method is significantly better than all the competing methods in terms of the computation time. The presented method takes about 15 seconds to compute the disparity field for the entire LF (i.e., 121 views).

**Real World Evaluation.** We continue with the real world evaluation. Figure 5 provides a qualitative comparison to the methods by Tao et al. [27], Heber and Pock [10], and Jeon et al. [14]. To be able to compute results for the methods by Jeon et al. [14] and Tao et al. [27] in a reasonable time, it was necessary to reduce the directional resolution of the data to $11 \times 11$ and the number of labels to 75. The results show that although the proposed model was not trained on this dataset, nor have we performed any fine-tuning for this dataset, it allows to predict depth maps that are on par with the competing methods. However, the results are not perfect because the model produces streaking artifacts in homogeneous background regions. The main benefit of the proposed method is again the computational time of a few seconds. Also keep in mind that we are not using any post-processing, the results shown in the figure are the raw network predictions.

In Figure 6 we also present results for a LF captured with a Lytro camera. Note, that the Lytro data includes a significant amount of noise and outliers, for which the proposed u-shape network was not trained for. Nevertheless, the proposed model is able to predict a reasonable disparity field with clear depth discontinuities.

**6 Conclusion**

We have presented a novel end-to-end system for SfLF. Our model is based on stacked convolution operations, that result in a high efficiency. The model comprises an encoding and a decoding part. Those parts are symmetric resulting in a u-shaped network architecture. We avoided fully connected layers thus our model allows to process LFs of any resolution. Our results show that the proposed u-shaped network is able to predict disparity fields that are on par with the state of the art while maintaining a low computation time. We believe our proposed approach is an important step towards realtime LF image processing. We also want to emphasize that the results shown in the experimental section are the raw network predictions without any additional post-processing. Investigating suitable methods for post-processing the network output is left for future work.
Figure 5: Qualitative comparison for LFs from the SLFA. The figure shows from left to right the center view of the LF, followed by the results for the methods proposed by Tao et al. [27], Heber and Pock [10], and Jeon et al. [14]. The results to the right correspond to the proposed method.

Figure 6: Qualitative comparison for a LF captured with a plenoptic camera. The figure shows from left to right the center view of the LF, followed by the results for the methods proposed by Tao et al. [27], Heber and Pock [10], and Jeon et al. [14]. The result to the right corresponds to the proposed method.
Acknowledgment. This work was supported by the FWF-START project Bilevel optimization for Computer Vision, No. Y729 and the Vision+ project Integrating visual information with independent knowledge, No. 836630.

References


