Light field intrinsics with a deep encoder-decoder network

Anna Alperovich, Ole Johannsen, Michael Strecke and Bastian Goldluecke
University of Konstanz
Konstanz, Germany
anna.alperovich@uni-konstanz.de

Abstract

We present a fully convolutional autoencoder for light fields, which jointly encodes stacks of horizontal and vertical epipolar plane images through a deep network of residual layers. The complex structure of the light field is thus reduced to a comparatively low-dimensional representation, which can be decoded in a variety of ways. The different pathways of upconvolution we currently support are for disparity estimation and separation of the lightfield into diffuse and specular intrinsic components. The key idea is that we can jointly perform unsupervised training for the autoencoder path of the network, and supervised training for the other decoders. This way, we find features which are both tailored to the respective tasks and generalize well to datasets for which only example light fields are available. We provide an extensive evaluation on synthetic light field data, and show that the network yields good results on previously unseen real world data captured by a Lytro Illum camera and various gantries.

1. Introduction

Light fields have a complex, heavily redundant structure. In their two-plane parametrization [24], they are given as a dense, regularly sampled 2D grid of so-called subaperture views of a scene. When fixing a single vertical or horizontal line in the image plane and moving through the space of view points in the same direction, one obtains 2D slices in this four-dimensional space, which are called epipolar plane images (EPIs), see Figure 5. For scenes with purely diffuse reflection, these exhibit patterns of oriented lines of constant color. Each of these lines corresponds to the projection of a single 3D point in space, and its slope, called the disparity, is inversely proportional to the point's distance to the observer. Discontinuities in the pattern are caused by occlusions, as they cause transitions between multiple orientations at the occlusion edge [40], see Figure 2.

The situation also becomes less straightforward when reflection or glossy, non-Lambertian surfaces come into play,

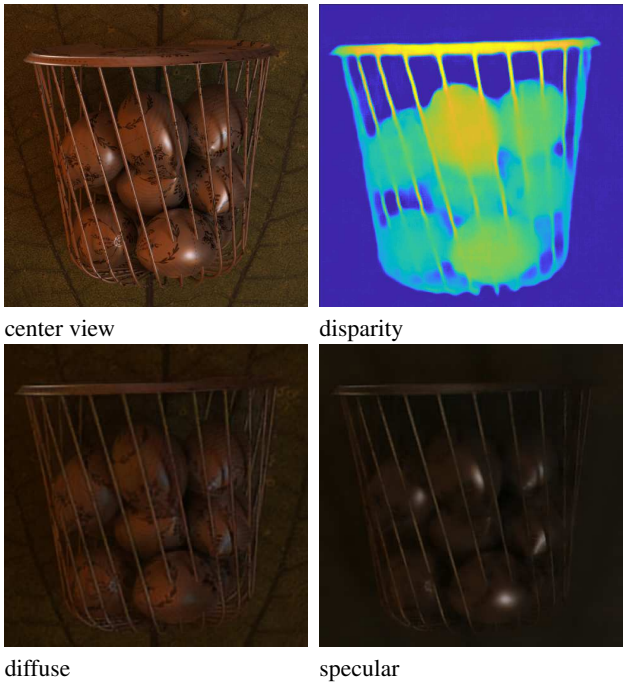

Figure 1. Our network jointly separates an input light field into diffuse and specular components, and computes a disparity map for the center view. This figure shows output on a previously unseen light field rendered with Blender.

as the EPIs then show superimposed patterns [19]. The orientation of the patterns corresponding to specular reflection does not correspond to disparity, but the specular flow direction, which depends on the intrinsic surface geometry. To distinguish between those two cases, one must know if a point exhibits diffuse or specular reflection. On the other hand, with known geometry, the specular flow can be directly estimated and reflection components can be separated [34]. In case that both shape and reflectance are unknown, it is hardly possible to tell which phenomena gave rise to a particular EPI.

Nevertheless, EPIs from natural light fields exhibit an overall regular structure, and it seems likely that they form

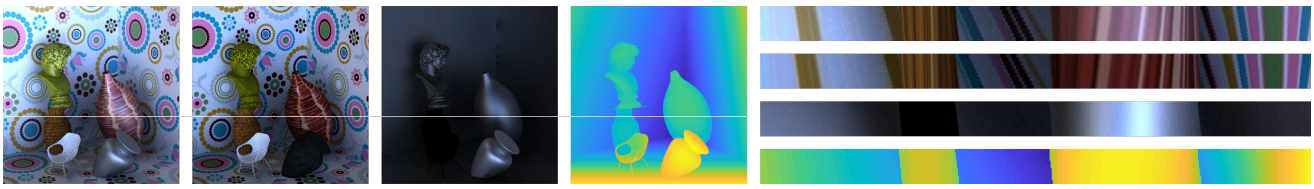

Figure 2. The four images to the left show, from left to right, the center view of the input light field, the diffuse component, the specular component (scaled for better visibility) and the disparity. The EPIs to the right are all taken from the same scan line in the light field, marked white. From top to bottom, they again show the input, the diffuse component, the specular component and the disparity. Since the diffuse component and the disparity correspond to the same projections of the same 3D points, they share the same pattern. However, the specular component behaves differently, as it follows the specular flow [34], which depends on the local surface geometry and view point change in a complex way. In particular, the orientation of the specular lobe in the EPI is different from that of the diffuse texture.

a comparatively low-dimensional manifold within all of epipolar plane image space. Furthermore, encoding an EPI well with only a few parameters is related to the difficult interrelated tasks, such as disparity estimation or separation of diffuse and specular components. Intuition suggests that if you learn how to do compression well, you will be able to better succeed at the other tasks. The idea of this paper is therefore to learn a low-dimensional representation of EPIs from arbitrary example light fields, but in a way that the latent variables can be used jointly to accurately solve various supervised tasks in light field analysis. For this, we propose an encoder-decoder neural network based on the concept of deep auto-encoders [14], which recently have been highly successful in finding meaningful manifold representations [28, 15].

Contributions. We introduce the first network architecture to jointly solve disparity regression and reflectance separation in light fields. Our fully-convolutional encoder-decoder network can be trained both unsupervised to just learn representations, as well as supervised to solve the above tasks based on the latent space. We employ 3D convolutions to compute features integrated over the whole range of both vertical and horizontal stacks to deal with complex occlusions and reflections. The network is trained on datasets rendered with Blender taken from the benchmark [16], as well as a custom random light field generator which in theory can synthesize an arbitrary amount of training data for reflection separation as well as disparity estimation. Currently we use dataset of 175 light fields, and will share rendering scripts and network code. We demonstrate in extensive comparisons that our method quantitatively and qualitatively outperforms existing light-field methods for diffuse and specular separation, and can robustly compute depth for highly specular scenes.

2. Related work

Encoding light fields. From the first introduction of light fields for image-based rendering [6, 25], light field compression has been an important topic due to the huge amount of data which needs to be stored. Early on, it has been noted that estimating disparity is necessary to exploit the redundancies in the different viewpoints [26]. This can be turned around, and sparse coding actually been used as a tool for disparity estimation - similar in spirit to what we are proposing here. In [11] they use the idea of redundancy of sub-aperture views and used sparsity of the RPCA as a new matching term. Likewise, [29] employ sparsity ideas to model light field patches as Gaussian random variables conditioned on its disparity value. They construct a patch prior and can estimate disparity by finding the nearest PCA subspace. In [19], EPI patches are encoded with a dictionary of patches with known slope, such that the coding coefficients give a disparity estimate. Notably, this method can recover disparity for multiple layers of a scene. Sparse coding is also used for compressive light field photography [27], which reduces the amount of data to be captured. Both sparse coding and low-rank constraints are also key to modern light field compression schemes [3, 18].

However, the idea of an auto-encoder we employ in this work is in some sense the exact opposite to sparse coding: instead of finding an overcomplete basis and represent patches with a sparse vector in a high-dimensional space, we want to find the best low-dimensional coding directly.

Reflection separation. The dichromatic reflection model proposed by Shafer [30] decomposes an input scene into diffuse and specular components. Based on this, [46] considers specularity removal as an image denoising problem and solves it with bilateral filtering. In [37, 36], Tan and Ikeuchi devise a method based on pure chromaticity analysis without any geometrical information. Kim *et al*. [21] used the fact that the dark channel can provide an approximately specular free image. In [1], Akashi and Okatani use sparse non-negative matrix factorization to jointly estimate body color and separate reflection components.

What makes reflection separation from a single image particularly difficult is that specularity is a view dependent phenomenon, and can hardly be recognized from a single view point. With multiple views available, changes in ob-

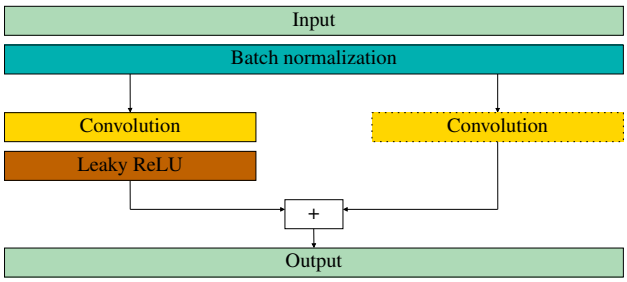

Figure 3. A single residual block of the network. After batch normalization, a first path leads through a (possibly strided) convolution layer and a leaky ReLU. A second path either keeps the input, or passes it through a strided convolution in case it needs to be resampled. Both paths are added together to produce the final output. The idea is that it is much easier for such blocks to learn the identiy transformation, or perform only small modifications to the input [10], which helps the encoder-decoder paths to gradually add details.

ject appearance can be tracked with respect to the viewing angle, which significantly simplifies the task of reflection separation. The behavior of specularity in static scenes with a moving camera is described by Swaminathan *et al*. [35]. They show how motion of specularity depends on object geometry and light source position, and propose a technique for specularity extraction from an image sequence.

Recent works by Gryaditskaya *et al*. [8] and Sulc *et al*. [34] explore the light field structure to edit appearance of specularity and estimate diffuse and specular components. Tao *et al*. [38] adapt the dichromatic reflection model to light fields and propose a depth estimation and specularity removal algorithm. Criminisi [4] studies the behavior of diffuse and specular components in EPIs and proposes several reflection separation techniques.

Neural networks for light field analysis. Deep neural networks are employed for all of the above tasks including light field analysis. Wang *et al*. [41] aim at material classification. They explore different light field representations that can be used to train a convolutional neural network. Heber and Pock [12, 13] apply an encoder-decoder architecture on 2D EPIs and later 3D EPI stacks to estimate depth. Kalantari *et al*. [20] and Srinivasan *et al*. [33] introduce view synthesis algorithms, which recover light fields from a sparse set of images or a single view. In a similar vein, [9] obtain compressive light field reconstructions from single coded 2D images using a joint autoencoder and 4D-CNN architecture. Recently, deep networks were also successfully applied for inverse rendering and intrinsic image problems [23, 31]. In contrast to the above approaches, our architecture is not limited to a single task, but can be trained to perform several of these jointly by implementing different decoder chains.

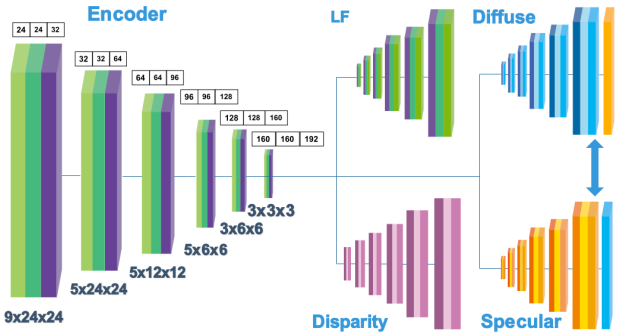

Figure 4. The pathways of our deep encoder-decoder network are organized in six groups of three residual blocks each. The first two blocks in each encoder group keep depth and resolution the same, the last block reduces resolution (shown on bottom, viewpoint $\times$ spatial coordinates), while increasing feature depth (shown on top) by 32. The decoder paths are exact mirrors of this chain. Disparity is only a 2D decoder, where the view point dimension of the shape is removed. To not overly clutter the figure, the visualization does not show that the encoder and 3D decoders actually operate on two EPI stacks in parallel, the horizontal and vertical one. The feature output of these is briefly joined on the bottom layer, and then decoded again into two separate chains.

3. Proposed network architecture

The key idea is to build the network around an autoencoder, so it can be trained unsupervised using just raw light fields. However, we add multiple pathways to decode the latent representation, which can be trained jointly with the autoencoder in a supervised manner, depending on which data is available in the current training example. Due to the combination of supervised and unsupervised training, we can make sure that the latent representation is both tailored to the desired tasks, such as depth reconstruction or intrinsic component representation, but can also generalize well to datasets for which no training information is available for these tasks. When the network is deployed, all decoder chains can be evaluated using just the light field data.

Encoder pathway. The input to the network is a pair of epipolar volumes, one sliced horizontally, the other one vertically, see Figure 5. Input patches are 48×48 RGB with a depth of nine views, larger light fields are segmented into these patches, so that our network can deal with lightfields of any shape.

The basic ingredient for the encoders and decoders are residual blocks. To decrease resolution, we employ strided convolutions instead of max-pooling, so the network is fully convolutional. See [10, 32] for justifications of this architecture. The residual blocks have a very simple structure and allow direct pass-through of the (batch normalized) input, see Figure 3.

In the encoder pathway, 18 residual blocks are chained

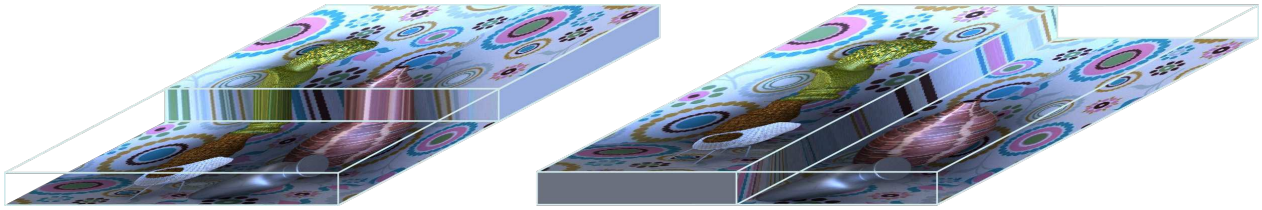

Figure 5. Visualization of horizontal (left) and vertical (right) EPI stacks used as input to our network. To achieve the actual spatial input resolution of 48×48, they need to be cut out from the above epipolar volumes. Note that although both stacks are three dimensional, they use images along different directions of view points. In effect, those two stacks assemble a crosshair of views around the center view, which is thus the only view present in both stacks.

together. Every third one reduces the patch resolution via strided convolution while increasing feature depth, with the overall goal of gradually reducing dimensionality. The final output has shape $3 \times 3 \times 3 \times 192$, for an overall reduction of the input to around 8.3% of its original size, see Figure 4. Horizontal and vertical epipolar volumes are encoded separately. As they have the exact same structure, we have them share the same filter kernels to reduce the number of network parameters. Since pathways like depth reconstruction require information from both horizontal as well as vertical epipolar volumes, their feature output at the representation level is concatenated. This is the final output of the encoder, and the bottleneck of the network.

Decoder pathways and output. After passing the bottleneck, the low-dimensional representation is decoded again by a chain of residual layers. The latent variables enter different decoder pathways. In this paper, we implement the auto-encoder path to reconstruct the input, two decoders for the diffuse and specular components, and a separate decoder for the disparity map. All decoder pathways use transpose convolutions to exactly revert the encoder on the corresponding level. However, the only link between them is through the latent representation, see Figure 4.

Lightfield, diffuse and specular components are reconstructed for the $17 = 9 + 9 - 1$ views in a crosshair around the center view, see Figure 5. The disparity map is computed for the center view only. We employ the dichromatic reflection model [30], whose adaptation to lightfields was discussed in detail in [38]. According to this model, the specular component is assumed to be independent from the diffuse one, which justifies the use of two separate decoder chains. However, they should also sum up to the input light field. To let the network better cope with this constraint, we append specular features to the diffuse ones and vice versa, but only for the input to the final layer. As disparity output is only 2D, we reduce the filter shape by the respective dimension. When tiling the output back together, we use overlapping patches and extract only the central 16×16 pixels, as data closer to the center is more accurate.

4. Network training

4.1. Training data

As input data for our algorithm we use a variety of publicly available datasets [45, 39, 16, 43] as well as scenes specifically created for the purpose of reflection separation.

4D light field benchmark [16]. The light field benchmark [16] offers 28 light fields rendered with Blender with ground truth disparity available. Their composition varies substantially, with many different materials, lighting conditions, and fine structures with complex occlusions. Their center view resolution is 512×512, but here and for all other datasets, we use only completely valid patches for training, in the sense that pixels shifted by their disparity always lie within all of the views. We use 48×48 pixel patches for training with 16 pixels of overlap, skipping a 16 pixel border region. In effect, this gives 900 training patches per light field for a total of around $25{,}200$ from the benchmark.

New light fields rendered with Blender. We generate data for specular and diffuse separation using the Blender addon provided with [16]. By randomizing scenes, we can generate a (theoretically) infinite amount of different light fields to ensure a large variety of data. We designed multiple scenes containing up to five objects of different scales and geometric complexity. Texture, the reflective properties and the environment map for lighting are chosen at random. Additionally, we randomly change the position and rotation of all objects and rotate the environment map, to prevent overfiting to certain geometries and lighting conditions. To ensure that the network can also deal with purely Lambertian materials, a certain percentage of objects have purely diffuse material. In total we used 36 pre-built scenes, 321 textures and 109 environment maps collected from different public sources. The 3D models we use are selected from Chocofur[1] and The British Museum[2]. We adapted the material properties to fit our needs and only used the mesh data.

Lightfields are rendered with the Cycles engine, and we

[1] http://www.chocofur.com

[2] https://sketchfab.com/britishmuseum

Dataset	L^2-loss times 100, validation data AE	diffuse	specular	disparity	L^2-loss times 100, training data AE	diffuse	specular	disparity
Synthetic								
Benchmark [16]	0.860	–	–	6.114	0.816	–	–	5.964
Ours	0.610	1.577	1.511	1.620	0.568	1.456	1.393	1.419
Real-world								
Lytro Illum	0.606	–	–	–	0.574	–	–	–
Stanford [39]	1.045	–	–	–	0.919	–	–	–
HCI [43]	1.230	–	–	–	1.150	–	–	–
Average	0.8702	1.577	1.511	3.867	0.8054	1.456	1.393	3.6915

Figure 6. Network losses for different groups of datasets at convergence. The datasets most difficult to fit for the autoencoder are the ones from gantries, perhaps due to minimally uneven sampling of viewpoints which has not been properly corrected. Depth reconstruction on our own synthetic dataset is surprisingly easier than for the benchmark datasets, although it has much stronger specularity. However, the geometry of our objects is also substantially simpler, and the datasets have large regions of easy to fit planes. Overall, disparity MSE on the benchmark validation is around the current benchmark average, which is 6.29. However, our model is not specifically optimized for depth reconstruction, and in particular trained for non-Lambertian scenes, on which it can perform much more robustly than competing methods, see Figure 7.

adapted the addon [16] such that it can output the intrinsic components. For both diffuse and specular passes, Cycles outputs the three different components color, direct lighting, and indirect lighting. Adding the direct and indirect light and multiplying it by the color yields the desired ground truth separation. Data is stored in high dynamic range to circumvent problems with saturated specularities. The size of these light fields is also $9 \times 9 \times 512 \times 512$. The 175 light fields we use for training contain around $160{,}000$ patches.

Real-world light fields. We have four sources for real world light fields for which no ground truth data is available. First, we use light fields captured with the Lytro Illum light field camera, calibrated and rectified using the light field toolbox from [5]. The size of the light fields is $9 \times 9 \times 434 \times 625$. We used 11 light fields for training and two for testing, which results in $10{,}175$ training examples. Second, we have a dataset built from the Stanford Light Field Archive [39] with six training data sets which is $6{,}816$ patches, and with two light fields held back for testing. Third, we captured a light field using an industrial camera mounted on a gantry we assembled ourselves. The size of the light field is $9 \times 9 \times 497 \times 710$ with a disparity range of $[-1.5, 1]$. The light field illustrates a non-Lambertian object, illuminated with approximately white light. Fourth, we use five real world light fields from the HCI benchmark [43] and we keep one for testing, which results in $16{,}016$ more training patches.

4.2. Network implementation and training strategy

From the training data, we set aside 5% for a validation set. Several light fields are also completely held back, and used only for testing, see above for details. We implement the network using Tensorflow in Python3, and train on an Intel Core i9 system with four nVidia Titan Xp, with the encoder/decoder chains distributed to different GPUs to satisfy memory requirements for training. All decoders are trained with an L^2-loss. In case a dataset does not provide ground truth for a certain pathway, that path is disabled during training. The autoencoder path can always be trained. Weights are initialized using the same strategy as for residual networks [10]. Stochastic optimization using the Adam optimizer [22] for twenty epochs of training data took roughly five days, after which loss for all pathways remained stable. The final losses over training and validation set are shown in Figure 6. While there is of course a slight gap between training and validation, performance on unseen data is not significantly worse, so overfitting does not seem to be an issue here.

Reconstruction of a single pathway during evaluation requires roughly 7 seconds on the above system for a light field with a center view resolution of 512×512, including tiling of the input light field, all transfers from CPU to GPU and back, and reassembling the output from the patches. The complete specular/diffuse decomposition with disparity estimation takes 19 seconds. We verify the quality of reflection separation and disparity estimation in detail in the next section.

5. Results

We compare our reflection separation with two algorithms designed for light fields. The first one by Sulc *et al*. [34] performs reflection separation based on specular flow. The second one is by Alperovich *et al*. [2] and performs intrinsic light field decomposition. In addition, we compare to the network proposed by Shi *et al*. [31], which uses a deep autoencoder for intrinsic images. However, it only works for standard 2D images. To compare to the full decomposition [2], where the authors decompose the input light field into albedo, shading and specularity, we compute the diffuse component by multiplying albedo and

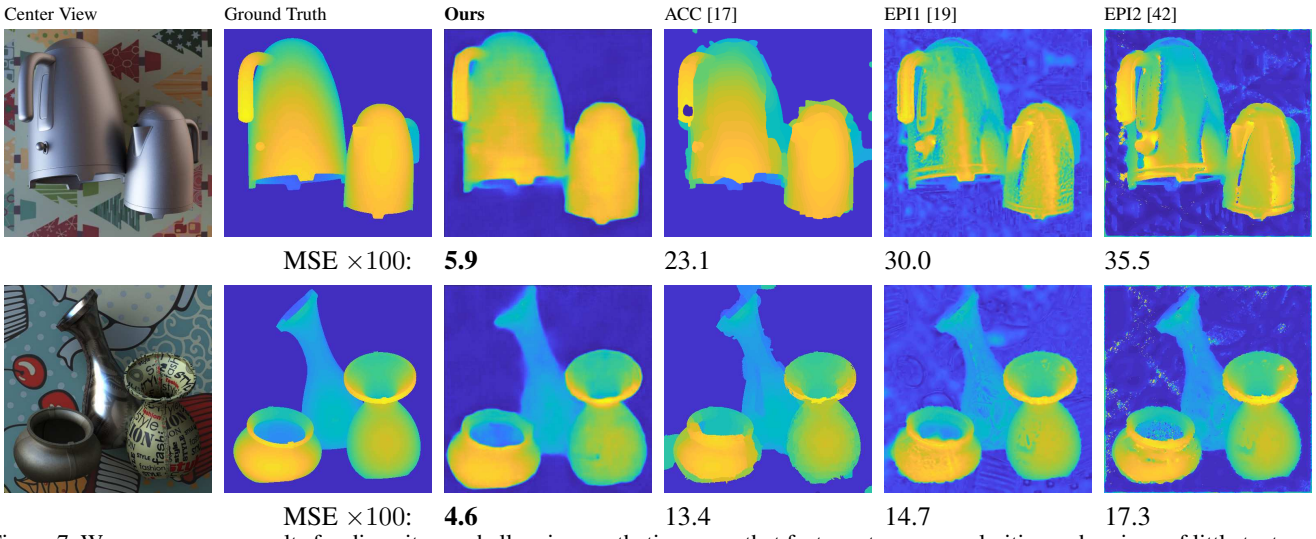

Figure 7. We compare our results for disparity on challenging synthetic scenes that feature strong specularities and regions of little texture against state of the art methods for depth estimation. Epecially in regions where the specularity dominates the texture, the other EPI based methods fail, while ACC due to its strong regularization can still yield pleasing (albeit oversmoothed) results. With respect to MSE, our approach outperforms the other methods significantly.

	LMSE ×100		GMSE ×100		SSIM ×100	
	diff.	spec.	diff.	spec.	diff.	spec.
Ours	0.15	**0.11**	0.28	**0.23**	**80.08**	**81.37**
Alperovich [2]	**0.12**	0.45	**0.22**	1.04	74.98	48.46
Sulc et al. [34]	**0.12**	0.47	0.24	1.01	75.43	47.25
Shi et al. [31]	0.34	0.15	0.5	0.39	63.02	73.71

Figure 8. Comparison of different error metrics for specular and diffuse components. Numbers show the average over nine previously unseen test datasets. See section 5 for a description of the metrics. Since Shi et al. [31] does not perform decomposition for the background, we multiply all results and ground truth with object mask before measuring the errors.

	LMSE ×100		GMSE ×100		SSIM ×100		MSE (depth) ×100	
	diff.	spec.	diff.	spec.	diff.	spec.	scene 1	scene 2
Original	**0.25**	**0.19**	**0.64**	**0.62**	**66.66**	**72.75**	**5.9**	**4.6**
48 x 48	0.33	0.33	0.74	0.73	56.67	59.07	192.7	167.9
9 x 24 x 24	0.28	0.35	0.69	0.85	57.72	62.87	55.87	19.31

Figure 9. Ablation study: Quantitative comparison of separation over nine previously unseen test datasets, and depth estimation for the two scenes from Figure 7. Note that we compute error for the whole center view, without object mask.

shading [7].

For quantitative results, we evaluate reflection separation on synthetic scenes and report the local mean-squared error (LMSE) [7] which we compute patch-wise. This error is scale invariant, since the brightness of the patches is adjusted to the ground truth. In our experiments, we use rectangular overlapping patches with a size of 20% of the total image size. To evaluate the errors that might me canceled by LMSE, we also compute global mean squared error (GMSE) that adjusts the brightness value for the whole image. We also measure the structural similarity index (SSIM). See Figure 8 for an overview of all numerical results, and Figures 10 and 11 for a visual comparison. We also compare performance of disparity map estimation for specular scenes to different other algorithms in Figure 7. As an ablation study, we performed two experiments. In the first case we trained network only for center view without any disparity information from sub-aperture views, in the second case we have reduced spatial patch size to 24×24. Both experiments lead to decrease in performance compared to the original network, see Figure 9 for the comparisons on the same data sets that are used in Figures 10, 8, 7 . Finally, results of our method on different datasets that are commonly used in the light field community [39, 43, 16] can be found in Figure 12. We refer to the supplementary material for more results for the real and synthetic scenes, and videos for diffuse and specular components that show angular consistency of the decomposition.

6. Conclusion

In this work, we propose a generative encoder-decoder architecture for patches taken from light field epipolar volumes. Using different decoder paths, we can achieve both intrinsic decomposition as well as disparity estimation with a unified network. Thanks to joint training of autoencoder and the supervised pathways, we can transform the input light field into a latent representation which is both much smaller and well adapted to the desired tasks.

Our method outperforms recent light field based methods [34, 2], and a single image deep network approach for intrinsic image decomposition [31]. Although we have only

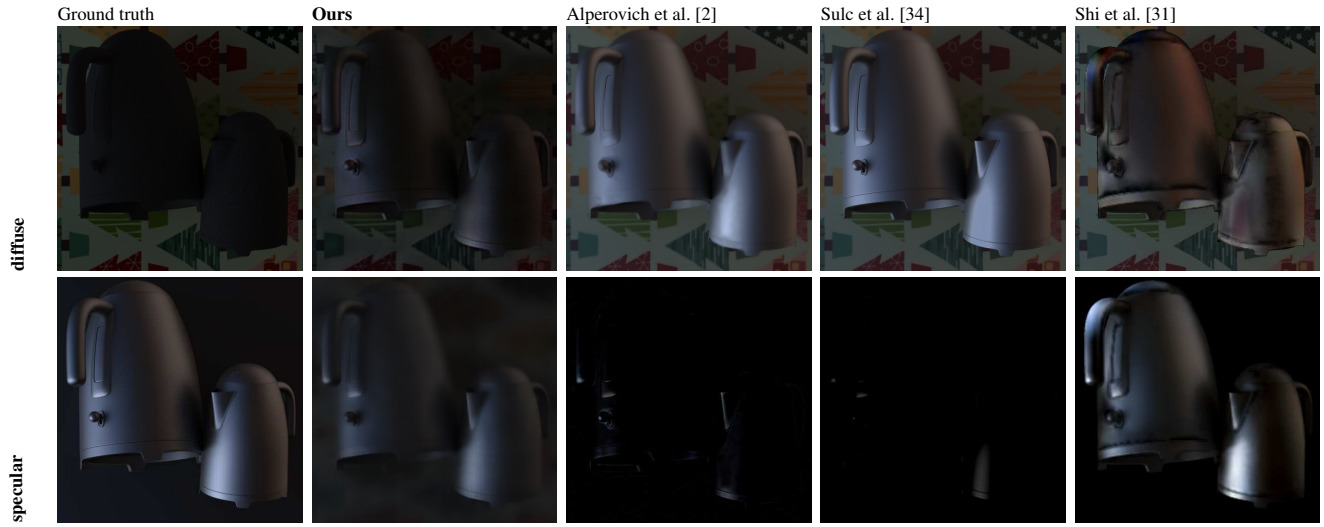

Figure 10. Comparison for a synthetic data set with two non-Lambertian objects with almost no texture, which is typically challenging for reflection separation. Both modeling approaches [2] and [34] fail to separate the specular component from the diffuse one. The CNN-based approach [31] successfully separates reflection components, but the diffuse one has some artifacts. In addition, the method requires an object mask, thus its application is limited to objects well separated from the background, which are rarely found in real world scenes.

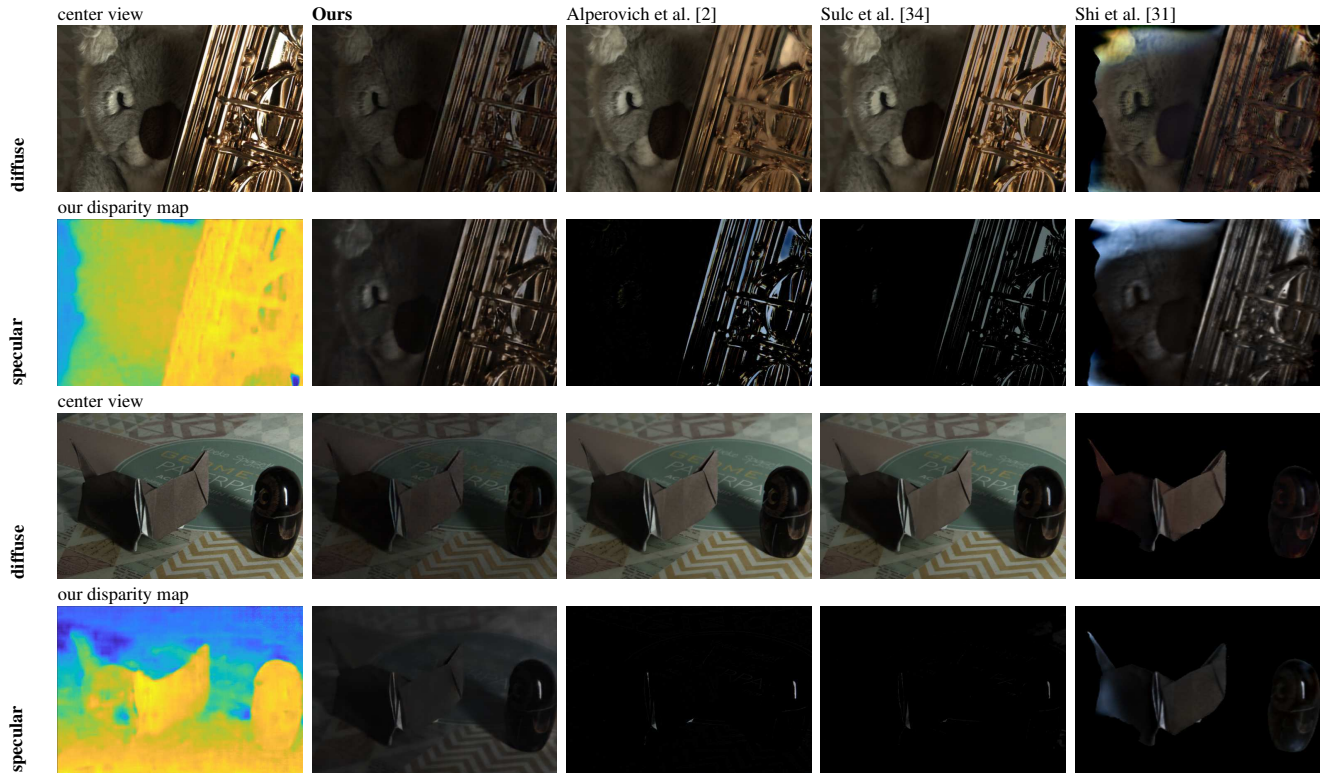

Figure 11. Two light fields captured with the Lytro Illum plenoptic camera. The first scene consist of a highly specular saxophone and an almost Lambertian koala. Our network successfully detects more specular parts of the saxophone compared to the other methods. While we mis-detect the koala as a specular object similar to [31], our method is the only one where the diffuse part behind the large specular spot on saxophone is not blurred. The second scene has two objects with very small saturated specularity, and only our method is the only one able to separate it. For all other methods, the specularity is still present in the diffuse component. Note that the single image CNN [31] does not perform decomposition for the background, thus it appears black in the visualization.

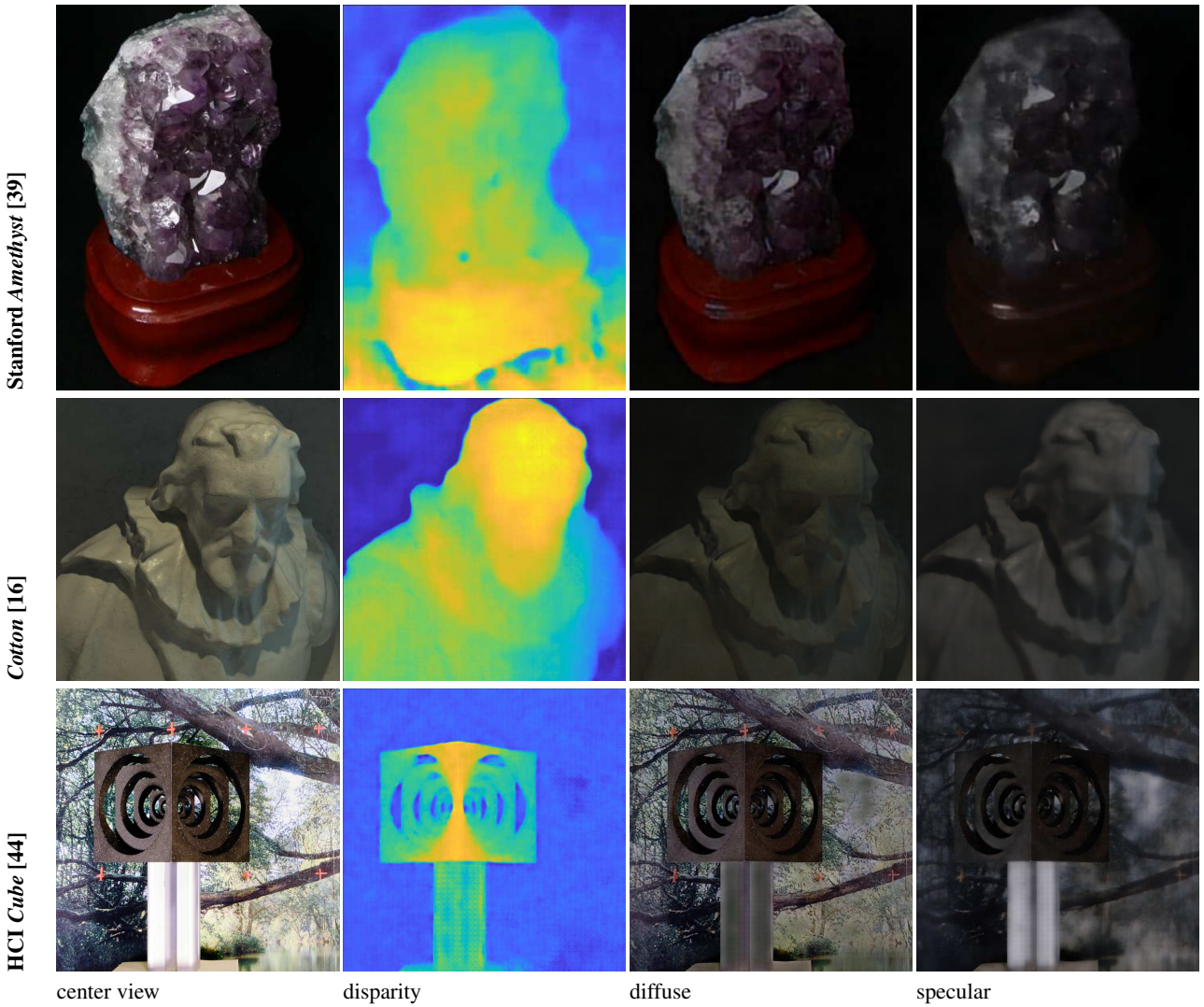

Figure 12. Results on unseen light fields from various sources. We show center views of the light fields with diffuse and specular components and estimated disparities. *Top:* lightfield from the Stanford data set [39], where we have chosen the most challenging case with respect to reflection separation and disparity estimation. Our network, while being trained on synthetic scenes, is able to generalize to real world examples with complicated geometry and reflection. *Middle:* synthetic scene from light field benchmark [16], where we have selected an object with small specular regions, to evaluate how the network will cope with it. Specularity is successfully from the diffuse part, while preserving texture. *Bottom:* an example data set from HCI benchmark [43].

average performance in depth reconstruction on datasets from the benchmark [16], in contrast to other methods, we still recover reliable depth in the presence of strong specularity. We also generalize well to real-world light fields captured with the Lytro Illum plenoptic camera or a gantry, although we do not have ground truth training data available for these. Despite being trained only on soft reflections, experiments with highly specular light fields show that we are robust against strong non-Lambertian effects. As the structures in epipolar volumes are both relatively characteristic and contain more information, we require only relatively few training examples (around 200 light fields), compared to single image approaches which use several millions of images.

Acknowledgments

This work was supported by the ERC Starting Grant "Light Field Imaging and Analysis" (LIA 336978, FP7-2014).

References

[1] Y. Akashi and T. Okatani. Separation of reflection components by sparse non-negative matrix factorization. *Computer Vision and Image Understanding*, 146:77–85, 2016.

[2] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke. Shadow and specularity priors for intrinsic light field decomposition. In *Int. Conf. on Energy Minimization Methods for Computer Vision and Pattern Recognition*, 2017.

[3] J. Chen, J. Hou, and L. Chau. Light field compression with disparity guided sparse coding based on structural key views. *IEEE Transactions on Image Processing*, epub2750413, 2017.

[4] A. Criminisi, S. Kang, R. Swaminathan, R. Szeliski, and P. Anandan. Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Computer vision and image understanding*, 97(1):51–85, 2005.

[5] D. Dansereau, O. Pizarro, and S. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 1027–1034, 2013.

[6] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, pages 43–54, 1996.

[7] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithm. In *Proc. International Conference on Computer Vision*, 2009.

[8] Y. Gryaditskaya, B. Masia, P. Didyk, K. Myszkowski, and H. P. Seidel. Gloss editing in light fields. In *Vision, Modelling and Visualization (VMV)*, 2016.

[9] M. Gupta, A. Jauhari, K. Kulkarni, S. Jayasuriya, A. Molnar, and P. Turaga. Compressive light field reconstructions using deep learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2016.

[11] S. Heber and T. Pock. Shape from light field meets robust PCA. In *Proc. European Conference on Computer Vision*, 2014.

[12] S. Heber and T. Pock. Convolutional networks for shape from light field. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2016.

[13] S. Heber, W. Yu, and T. Pock. Neural epi-volume networks for shape from light field. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[14] G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[15] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. *ACM Transactions on Graphics*, 35(4), 2016.

[16] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4D light fields. In *Asian Conf. on Computer Vision*, 2016.

[17] H. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. Tai, and I. Kweon. Accurate depth map estimation from a lenslet light field camera. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2015.

[18] X. Jiang, M. Pendu, R. Farrugia, and C. Guillemot. Light field compression with homography-based low-rank approximation. *IEEE Journal of Selected Topics in Signal Processing*, 11(7):1132–1145, 2017.

[19] O. Johannsen, A. Sulc, and B. Goldluecke. What sparse light field coding reveals about scene structure. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2016.

[20] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2016)*, 35(6), 2016.

[21] H. Kim, H. Jin, S. Hadap, and I. Kweon. Specular reflection separation using dark channel prior. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2013.

[22] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014.

[23] T. Kulkarni, W. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. *Advances in Neural Information Processing Systems*, 28:2539–2547, 2015.

[24] M. Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006.

[25] M. Levoy and P. Hanrahan. Light field rendering. In *Proc. SIGGRAPH*, pages 31–42, 1996.

[26] M. Magnor and B. Girod. Data compression for light field rendering. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(3):338–343, 2000.

[27] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar. Compressive light field photography using overcomplete dictionaries and optimized projections. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4):1–11, 2013.

[28] L. Mescheder, S. Nowozin, and A. Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *International Conference on Machine Learning (ICML)*, 2017.

[29] K. Mitra and A. Veeraraghavan. Light field denoising, light field superresolution and stereo camera based refocussing using a GMM light field patch prior. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 22–28, 2012.

[30] S. Shafer. Using color to separate reflection components. *Color Research & Application*, 10(4):210–218, 1985.

[31] J. Shi, Y. Dong, H. Su, and S. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *Proc. International Conference on Computer Vision and Pattern Recognition*, 2017.

[32] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. In *ICLR (workshop track)*, 2015.

[33] P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng. Learning to synthesize a 4D RGBD light field from a single image. In *Proc. International Conference on Computer Vision*, 2017.

[34] A. Sulc, A. Alperovich, N. Marniok, and B. Goldluecke. Reflection separation in light fields based on sparse coding and specular flow. In *Vision, Modelling and Visualization (VMV)*, 2016.

[35] R. Swaminathan, S. B. Kang, R. Szeliski, A. Criminisi, and S. K. Nayar. On the motion and appearance of specularities in image sequences. In *Proc. European Conference on Computer Vision*, volume I, pages 508–523, May 2002.

[36] R. T. Tan and K. Ikeuchi. Separating reflection components of textured surfaces using a single image. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(2):178–193, 2005.

[37] R. T. Tan, K. Nishino, and K. Ikeuchi. Separating reflection components based on chromaticity and noise analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(10):1373–1379, 2004.

[38] M. Tao, J. C. Su, T. C. Wang, J. Malik, and R. Ramamoorthi. Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 38(6):1155–1169, 2015.

[39] V. Vaish and A. Adams. The (New) Stanford Light Field Archive. `http://lightfield.stanford.edu`, 2008.

[40] T. Wang, A. Efros, and R. Ramamoorthi. Occlusion-aware depth estimation using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3487–3495, 2015.

[41] T. Wang, J. Zhu, E. Hiroaki, M. Chandraker, A. Efros, and R. Ramamoorthi. A 4D light-field dataset and CNN architectures for material recognition. In *Proc. European Conference on Computer Vision*, 2016.

[42] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4D light fields. In *Proc. International Conference on Computer Vision and Pattern Recognition*, pages 41–48, 2012.

[43] S. Wanner and B. Goldluecke. Reconstructing reflective and transparent surfaces from epipolar plane images. In *German Conference on Pattern Recognition (Proc. GCPR)*, 2013.

[44] S. Wanner, S. Meister, and B. Goldluecke. Datasets and benchmarks for densely sampled 4D light fields. In *Vision, Modelling and Visualization (VMV)*, 2013.

[45] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy. High performance imaging using large camera arrays. *ACM Transactions on Graphics*, 24:765–776, July 2005.

[46] Q. Yang, J. Tang, and N. Ahuja. "efficient and robust specular highlight removal". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(6):1304–1311, 6 2015.