IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 30, 2020, accepted October 14, 2020, date of publication October 22, 2020, date of current version November 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3033056

A Multispectral Light Field Dataset and Framework for Light Field Deep Learning

MAXIMILIAN SCHAMBACH AND MICHAEL HEIZMANN
Institute of Industrial Information Technology, Karlsruhe Institute of Technology, 76187 Karlsruhe, Germany
Corresponding author: Maximilian Schambach (schambach@kit.edu)

This work was financed by the Baden-Württemberg Stiftung gGmbH. The authors acknowledge support by the state of Baden-Württemberg through bwHPC as well as by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

ABSTRACT Deep learning undoubtedly has had a huge impact on the computer vision community in recent years. In light field imaging, machine learning-based applications have significantly outperformed their conventional counterparts. Furthermore, multi- and hyperspectral light fields have shown promising results in light field-related applications such as disparity or shape estimation. Yet, a multispectral light field dataset, enabling data-driven approaches, is missing. Therefore, we propose a new synthetic multispectral light field dataset with depth and disparity ground truth. The dataset consists of a training, validation and test dataset, containing light fields of randomly generated scenes, as well as a challenge dataset rendered from hand-crafted scenes enabling detailed performance assessment. Additionally, we present a Python framework for light field deep learning. The goal of this framework is to ensure reproducibility of light field deep learning research and to provide a unified platform to accelerate the development of new architectures. The dataset is made available under `dx.doi.org/10.21227/y90t-xk47`. The framework is maintained at `gitlab.com/iiit-public/lfcnn`.

INDEX TERMS Dataset, deep learning, disparity, light field imaging, multispectral imaging.

I. INTRODUCTION

Machine learning has gained a lot of attention in the image processing and computer vision community, in particular due to the recent advances regarding artificial neural networks and convolutional neural networks (CNNs). Whereas image-based deep learning, with applications such as classification, demosaicing, superresolution, denoising, etc., has a broad community with multiple tools and diverse datasets available, light field-related deep learning has a shorter history. There are many data-driven light field applications which have been recently discussed in the literature, ranging from disparity estimation [1]–[3], superresolution [4], [5], compression and compressed sensing [6], [7] to intrinsics estimation [8], to name a few. Many of these applications have significantly outperformed their respective conventional counterparts. For example, disparity estimation using CNNs [1] has outperformed conventional methods such as those based on the structure tensor or variational approaches [9], [10]. However, many of such architectures are supervised (with the notable exception of the work by Peng *et al.* [11]) and hence require synthetic light fields with corresponding ground truth labels such as disparity or surface normals. While there is a variety of synthetic light field datasets available, as we will discuss in detail in Section II, all of the available datasets consist of RGB light fields. Recently however, research interest regarding multispectral light field applications and camera designs has sparked.

A multispectral light field $L(u, v, s, t, \lambda)$ in the so-called plane-plane parametrization can be thought of as a multi-view collection of conventional multispectral subaperture images,

$$I_{uv}(s, t, \lambda) = L(u, v, s, t, \lambda), \tag{1}$$

for some fixed angular coordinate (u, v). Here, (s, t) denotes the spatial coordinate and λ the spectral dependency of the light field, which is replaced by a three-channel color index in the case of RGB light fields.

The associate editor coordinating the review of this manuscript and approving it for publication was Alex Noel Joseph Raj.

Various multispectral light field camera designs have recently been discussed in the literature, ranging from multi-camera arrays [12], [13] over single camera snapshot imagers [14]–[16] to more exotic designs using catadioptric mirrors [17], demonstrating an increased interest in the topic. Multispectral light fields have successfully been used in depth estimation [12], as well as shape and reflectance reconstruction [13], where they have shown superior results, particularly in specular regions, as compared to RGB light fields. Furthermore, combining traditional light field applications with methods from multi- and hyperspectral imaging offers new possibilities, for example with respect to material detection or classification. However, a multispectral light field dataset (neither real nor synthetic) to explore data-driven applications and to provide a common evaluation baseline is not available, hindering the advance of this emerging research field.

To overcome these limitations, we propose a new synthetic multispectral light field dataset with depth and disparity ground truth. By providing the abstract scene description of all light fields, rendering of additional ground truth labels is possible. The dataset contains light fields of shape $(11, 11, 512, 512, 13)$, rendered from 500 randomly generated scenes using a specifically designed scene generator to enable machine learning applications (for which the dataset is patched into roughly 80 000 light field patches), as well as seven handcrafted scenes to be used for a more detailed performance evaluation. The light fields are spectrally sampled in the visible range from 400 nm to 700 nm in steps of 25 nm resulting in 13 spectral samples.

Moreover, the community around light field-related deep learning is facing the lack of a common framework, resulting in many highly customized solutions and boilerplate code. This leads to published architectures being hard to re-evaluate or not being able to reproduce results at all. Furthermore, due to the specific requirements unique to light fields as compared to conventional image processing, which we will discuss in detail in Section III, these custom solutions may suffer from bad performance or, at worst, bugs. As a step towards a unifying platform on which light field deep learning applications are developed and investigated, ensuring reproducibility, we propose a new Python framework based on TensorFlow and the Keras API.

Summarizing, our contributions are as follows:

- We propose a novel synthetic multispectral light field dataset with depth and disparity ground truth to be used for light field-related deep learning applications. To the best of our knowledge, it is the first multispectral light field dataset of its kind.
- As part of the dataset, a collection of hand-crafted synthetic light fields is provided which can be used to assess the performance of an application with respect to specific challenges such as occlusion, shadows, noise, and more.
- The dataset is validated by comparing it to existing RGB light field datasets and multispectral image datasets.
- We propose a new Python framework for light field-related deep learning applications to unify and accelerate light field-related deep learning research. The framework is made publicly available.
- We evaluate the framework in terms of data processing speed and augmentation performance for three common light field deep learning applications: disparity estimation, superresolution and light field autoencoding.

II. MULTISPECTRAL LIGHT FIELD DATASET

With the advent of deep learning, the demand for training and test data, both labelled and unlabelled, has increased dramatically. In the case of RGB light fields, several synthetic datasets (of varying scope) with available disparity ground truth have been published, including the well known HCI benchmark dataset [18], the HCI specular light field dataset [8], the INRIA dataset [19], and the Graz University dataset [20]. Furthermore, numerous datasets of real light fields (without disparity ground truth) are available for tasks such as material classification, compression, superresolution and denoising [21]–[23]. However, a multispectral light field dataset is yet missing.[1] The proposed multispectral light field dataset shall fill this gap. To the best of our knowledge, this is the first multispectral light field dataset with ground truth depth and disparity labels.

A. DATASET PROPERTIES

The proposed dataset consists of multispectral light fields rendered from 500 randomly generated scenes as well as seven hand crafted scenes (to which we refer to as *challenges*) that can be used to assess a certain aspect of the final task's performance. The light fields are rendered with 16 bit unsigned integer precision with a shape of $(11, 11, 512, 512, 13)$. The spectrum is sampled from 400 nm to 700 nm in steps of 25 nm, resulting in 13 spectral channels. To each light field, we provide the depth as well as the disparity labels of every subaperture view with 32 bit floating point precision. To accommodate different camera designs, such as multi camera arrays or monocular systems, all light fields are rendered in two different camera settings: one corresponding to a plenoptic camera in the so-called unfocused design [24], such as the Lytro camera. Here, the main lens focal plane corresponds to a disparity value of zero. The other corresponding to a plenoptic camera in the unfocused design whose main lens is focused at infinity, which is effectively equivalent to a multi camera array with parallel optical axes. In this case, a disparity of zero corresponds to optical infinity. Therefore, in total $1000 + 14$ multispectral light fields including depth and disparity labels are provided. The rendered light fields are densely sampled with an effective baseline of about 1.8 mm, similar to the Lytro Illum camera.

[1] Xiong *et al.* [16] provide a set of three multispectral light fields which they captured using a multispectral light field camera prototype, however, due to its limited size (and missing disparity labels), we do not consider this to be a dataset usable for performance evaluation nor machine learning.

The dataset of each camera configuration is split 400 : 50 : 50 into a training, validation and test dataset to allow comparability across different applications and publications. Since deep learning applications are usually not trained using the full-sized light fields but smaller patches, we provide pre-patched versions of the datasets with patch sizes of (11, 11, 36, 36, 13) as well as (9, 9, 36, 36, 13), which is a commonly used angular resolution for many light field applications. However the larger angular resolution of (11, 11) may be more suitable for some tasks, in particular in the area of compressive light field imaging. Finally, we also provide an RGB conversion of the proposed dataset, converted using the CIE 1931 color matching functions and the CIE D65 midday light illuminant. For each light field we provide an abstract scene description file that can be used to access the camera parameters or to render additional ground truth data if needed. For example, the used ray tracer can trace surface normals or 3D coordinates. While it is also possible to render segmentation labels, the dataset is not suitable for contextual applications as the scene geometry is random and not contextually realistic.

B. RANDOM SCENE GENERATION

To create light fields that can be used for disparity estimation or shape reconstruction, the data has to be synthesized as there is no accurate enough reference measurement technique available. Usually ray tracers are used to obtain a physically correct light field rendering of a scene. Most of the available synthetic RGB datasets are rendered using Blender with a light field plugin provided by Honauer *et al.* [18]. Whereas Blender provides high photorealism, to our knowledge there does not (yet) exist a multispectral extension of the used ray-tracing engine Cycles. For this reason, we use a recently published ray tracer [25] which is capable of directly rendering multispectral light fields and the depth ground truth.

To obtain a dataset large and diverse enough for data-driven applications, a vast amount of light fields has to be rendered. Hand-crafting such a large amount of scenes is arguably impossible. Therefore, we choose to create the scenes automatically, employing certain geometric constraints. This approach is not new, as the RGB light field datasets by Alperovich *et al.* [8] as well as Heber and Pock [20] also use an automatic random scene generation. However, our approach differs in some details from both of the aforementioned ones.

To achieve diverse geometric properties of the scene, we place a random number $n = \lfloor N \rfloor$ of objects in the field of view of the virtual camera. Here, N is distributed according to a normal distribution with mean $\mu = 28$ and standard deviation $\sigma = 5$. We use both ideal geometric objects (such as spheres, cones, and planes), as well as 3D mesh models obtained from multiple open-source databases. Unlike Heber *et al.*, we do not place objects in the scene at three distinct distances (foreground, midground, background) but rather specify a disparity range in which objects should be placed uniformly. To this end, for each object we draw a uniformly distributed disparity $d \sim \mathcal{U}(-2.5\,\text{px}, 3\,\text{px})$ and calculate the corresponding distance from the camera (in the focused configuration) at which the object's center is then placed. A background object, either a large-diameter sphere or a possibly tilted plane, is placed at roughly $d = -2.5$ px. Doing so, the background does not posses a constant but slightly varying disparity, unlike the scenes generated by Alperovich *et al.* [8].

In order to obtain diverse spectral properties of the scene, we use real multispectral images from two datasets [26], [27], RGB images (which are converted to spectra by the ray-tracer), as well as noise textures with constant random spectra $\mathbf{s} \in [0, 1]^{13}$, where every spectral value $s_i \sim \mathcal{U}(0, 1)$ is independently drawn from a uniform distribution. This results in a mixture of realistic spectra (from the multispectral images), smooth spectra (from the RGB images) as well as uncorrelated, random spectra, which we believe to be a reasonable mix for machine learning applications and geometric light field applications. In Figure 1, the central views and disparities of two generated example light fields are shown. We will analyze the resulting disparity and spectral distribution of the created dataset in detail in Section IV-A.

C. CHALLENGES

To assess the performance of a specific light field application in detail, further data is needed. While a quantitative performance score can be calculated on the test dataset (with respect to one or multiple evaluation metrics) these values may only be used to quantitatively compare different architectures—their absolute values however are hard to interpret, in particular when the light fields are patched into smaller sizes for training and testing. Therefore, seven hand-crafted scenes, so-called *challenges*, are provided together with their respective ground truth disparities. These scenes should be used to further quantitatively and visually compare the obtained results. Moreover, they may be used to assess the performance with respect to a specific challenging aspect such as occlusion, shadows, detail, and noise. The proposed challenges consist of the following scenes: *Cabin*, *Elephant*, *Bust*, *Backgammon*, *Circles*, *Dots*, and *Wall*, the first six of which are shown in Figure 2. The reader familiar with the HCI benchmark dataset [18] will notice similarities between the two. In fact, the idea to pose additional challenges is heavily inspired by the so-called *stratified scenes* of the HCI benchmark dataset. Furthermore, the scenes *Bust*, *Backgammon*, and *Dots* are re-modeled according to scenes of the HCI benchmark dataset. The scene *Dots* is superposed with independent Gaussian noise whose variances differs across the eight identical patches of the scene, resulting in a block-wise PSNR of 45 dB (top left patch) and decreasing by 5 dB to 10 dB (bottom right patch). For each subaperture view, the noise is independent. While the first three challenges use high resolution 3D mesh models and show a realistic scene geometry, the latter are purely synthetic, utilizing ideal geometric shapes. The last scene, *Wall* (which is not shown), consists of a flat surface of constant disparity with a multispectral image texture. This scene is then rendered at

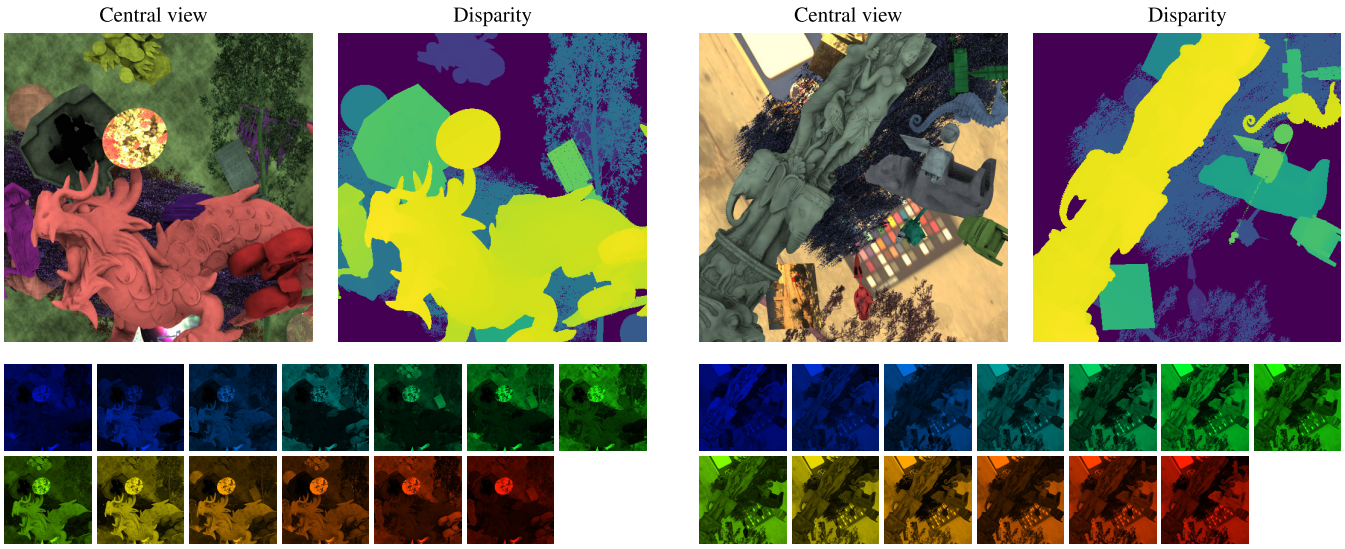

FIGURE 1. Two example light field's central views (top: converted to RGB, bottom: colored individual spectral channels) and corresponding central disparity maps. Note that not all details are visible in the disparity maps due to the large range of the colormap and a limited resolution of 8 bit used for this visualization.

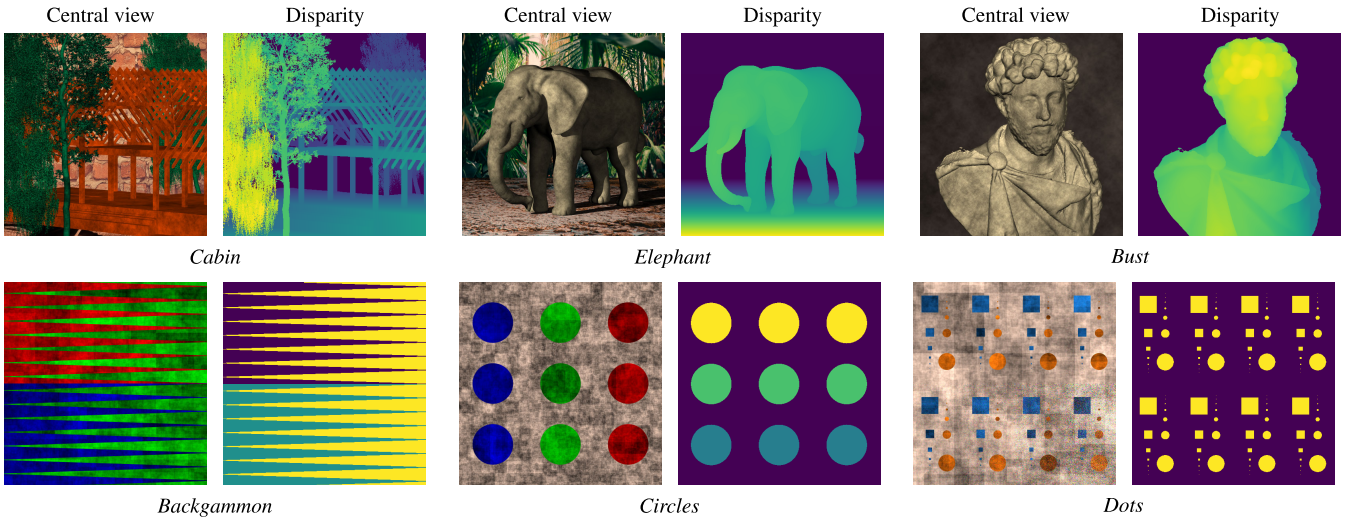

FIGURE 2. Central views (converted to RGB) and corresponding central disparity maps of 6 of the 7 dataset challenges.

different disparities, ranging from −1.5 px to 1.5 px in steps of 0.25 px which can for example be used to quantitatively compare multispectral-related performance vs. disparity. Hence, strictly speaking, the *Wall* challenge consists of 13 individually rendered light fields and their corresponding disparity ground truth.

III. DEEP LEARNING FRAMEWORK

In experimental research, both physical and numerical, reproducibility of the obtained results is of vast importance. Because often not all implementation details can be thoroughly presented in a written publication, especially in the case of machine learning where many hyperparameters would clutter the presentation, it is crucial to share the corresponding source code. For example, the source code holds valuable details of a presented deep learning architecture and training procedure. While this is already practiced by many researchers, especially in the case of high quality publications, there are frequently issues with the shared code in practice: first, the code may be implemented in a programming language that is not open source, making it impossible to reproduce the results without a corresponding license. In the academic world, this is often the case with code written in Matlab, which may additionally require licenses of certain proprietary toolboxes. While in academia, Matlab licenses are widespread, this might not be the case for the corresponding toolboxes. Second, the code may be poorly written, structured and/or documented, even in the case of

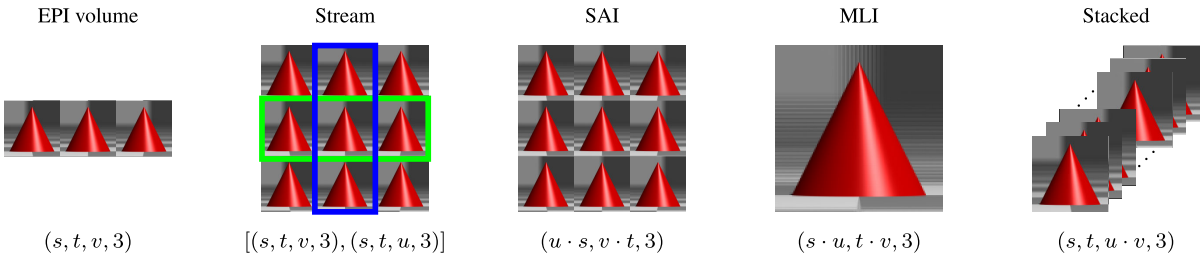

FIGURE 3. Different reshapes of a light field of shape $(u, v, s, t, 3)$ with corresponding resulting shapes. Note that, since the red cone is in focus, the shown MLI visually appears similar to the central subaperture view.

high quality publications (with notable exceptions). Often, published code requires a lot of manual tweaking and deeper knowledge of the source files to verify the presented experimental results and it may be impossible to adapt to one's own needs. Since a lot of times the code is written and updated under time pressure with certain experiments in mind, this is very understandable (and surely, researchers are not software developers). However, code shared this way is virtually impossible to run, which in turn makes it impossible to reproduce the corresponding experimental results or even look for the implementation details that were not presented in the paper. Concluding, more often than not, one is not able to reproduce state-of-the-art results of newly proposed architectures. Last, shared code in many cases contains a lot of custom boiler plate code to solve problems like data input and processing. Writing a lot of boiler plate code increases the overall time needed to implement new ideas and conduct the corresponding experiments, as well as the chance of bugs. A common framework mitigates these issues.

While projects such as TensorFlow (with the Keras API) and PyTorch (with the fastai API) succeed in providing open source frameworks for fast network training and validation workflows, especially in the case of image processing, light field-related applications pose additional challenges. Usually, data processing routines, network layers and convolution algorithms are only suited for 2D and 3D input (respectively 3D and 4D tensor operations on the mini batches). For 4D monochromatic or 5D color/multispectral light fields (respectively 5D and 6D mini batches), a lot of custom code modifications are necessary. Furthermore, due to the large memory requirement of light fields as compared to conventional 2D images (which becomes even more severe in the case of multispectral light fields), the data input and processing pipelines have to be implemented rather efficiently. Especially for smaller deep learning models, they may be the bottleneck of the network training. To this end we present a Python framework for light field-related deep learning applications, based on TensorFlow and the Keras API.

The contribution of the framework is mostly two-fold: First, an efficient and customizable data input and augmentation pipeline (including augmentation of labels such as disparity), is proposed, with special emphasis on reproducibility by proper random seeding. This input pipeline integrates well with the proposed pre-patched dataset. Second, we provide a wrapper around the Keras `Model` class, tightly integrating with the proposed data generators. In light field-based neural networks, there are multiple ways in which to input the light field. Since there is no native 4D convolution implementation in CUDA (and 4D convolution is computationally expensive), one usually does not use the standard light field as the input but rather a reshape or multiple streams extracted from it. For example, a common practice is to use an EPI volume or the so-called cross hair sections of the light field, corresponding to the vertical, horizontal and diagonal EPI volumes, resulting in a multi-input architecture. Alternatively, one performs a 2D or 3D reshape of the light field, for example using a subaperture image (SAI) or a microlens image (MLI) reshape, to feed into the network. Or one can simply stack the subaperture views channel-wise, partially loosing angular information. Depending on the reshape, spatial, angular or combined convolution can be performed by using a standard (possibly dilated and/or strided) 2D or 3D convolution on the reshaped light field. With this, an efficient pseudo (separable) 4D convolution can be achieved. For an overview of the most commonly used light field reshapes, see Figure 3. For this reason, a model architecture is tightly intertwined with the used light field input shape and hence with the used data processing pipeline, which previously resulted in a lot of custom, model-specific solutions. To this end, the proposed data generator can be easily extended to yield custom labels (such as disparity or segmentation labels) as well as a custom output shape. By default, we provide generators for light field outputs (used by autoencoders), disparity output (used by disparity estimators), downsampled light fields (used by superresolution architectures) and more, all using the same high level interface. These generators can then be combined with an arbitrary output reshape, for which we provide the previously mentioned ones. The user then only needs to specify the actual network architecture. We provide reference implementations of common light field-related architectures found in the literature such as the EPINET [1], VommaNet [2], and LFattNet [3] disparity estimators, the SAS-CONV [5] superresolution model and the encoder-decoder model by Alperovich *et al.* [8].

To ensure reproducibility, care has to be taken in the case of random operations such as data shuffling

and augmentation. This is particularly important for validation, testing, and in multiprocessing scenarios. To synchronize random seeds across multiple data generation processes during training, we use the current training epoch number as a random seed for data shuffling. As for random augmentation, the unique index of each light field in the corresponding dataset is used. To guarantee comparability, random augmentation and shuffling are turned off for validation and testing. Finally, the proposed framework integrates with Sacred [28] to log all parameters needed to be able to exactly reproduce experiments, however the framework can also be used without it.

Since the proposed network is based on TensorFlow and Keras, all loss functions and metrics provided there can be used for training and evaluation. However, we have re-implemented all of the commonly used losses to be used with light fields, such as the mean absolute error (MAE), mean squared error (MSE), structural similarity index metric (SSIM), its multiscale variant (MS-SSIM), and the peak signal-to-noise ratio (PSNR), as we have found them to be reduced differently over the mini batches (some are summed, some are averaged), which may be irritating in particular when combining and/or regularizing losses and in distributed (multi GPU) scenarios. Furthermore, we provide implementations of the Huber and pseudo Huber loss [29], [30] which combine advantages from both MAE and MSE. Tailored at disparity estimation, we provide implementations of the Total Variation and the BadPix (BP) metric [18]. In the case of multispectral applications, the Cosine Proximity and Spectral Information Divergence [31] are provided.

The framework supports multi-input, multi-label network architectures. Since the proposed framework is based (and tested) on the most recent TensorFlow release (v. 2.3), it supports all of its features such as mixed precision models and multi GPU training. All results presented in the paper are obtained using the proposed framework.

A. DATA AUGMENTATION

In order to increase the variance of the training data and to enhance certain invariances during the training, data augmentation techniques are commonly used in deep learning. Additional to conventional augmentations used in the case of 2D images, the 4D geometric structure of the light field has to be retained. In the proposed framework, we provide the following augmentations (including the corresponding disparity augmentation). It should be noted, that not all augmentations may be useful in different tasks, as we will discuss in more detail in Section IV-C. Therefore, the augmentations can be individually enabled.

Spectrum permutation: Randomly permutes the color/spectral channels of the light field. The disparity is not augmented, as it is spectrally invariant.

Spectral weighing: Randomly weighs each color/spectral channel of the light field with weights $w_\lambda \sim \mathcal{U}(0.75, 1.25)$. Again, the disparity is unaltered.

Gamma compression/stretching: A gamma compression/stretching is performed for a random $\gamma \sim \mathcal{U}(0.8, 1.2)$. Again, the disparity is unaltered.

Flipping: The input light field is flipped horizontally and/or vertically, each with a chance of 50%, in the spatial as well as the angular domain. The disparity is spatially flipped accordingly.

Rotation: Random rotation by 0°, 90°, 180° or 270°. Both the angular and the spatial coordinates have to be rotated. Hence, only rotations multiples of 90° can be performed without the need to interpolate (which is no problem in the spatial but in the angular domain). The disparity is spatially rotated accordingly.

Scaling: Random scaling with a factor calculated from the input light field shape and a specified target shape. Scaling is only performed in ranges that do not require boundary conditions. That is, if the spatial input shape is (36, 36) and the target shape is (32, 32), the scaling range is calculated to $[0.89, 1.11] = [32/36, 1 + (1 - 32/36)]$. Correspondingly, the disparity has to be scaled spatially as well as in its range since scaling effectively changes the light field's baseline.

Cropping: After scaling, the light fields have different spatial sizes. Therefore, a random spatial region is cropped from each light field (and the disparity) resulting in the final light field shape (which can individually be configured).

The augmentation is performed online upon each mini batch preparation. Since this task is being carried out by the CPU, certain bottlenecks have to be considered, depending on the batch size and the model under consideration. We will evaluate this in Section IV-B.

IV. EVALUATION

In the remainder, we evaluate the dataset in the setting corresponding to a light field camera with a focused main lens such that the disparity is zero at the focal plane. The configuration at which the camera is focused at infinity, corresponding to a multi camera array, is not further investigated here, as the properties only differ slightly.

A. DATASET VALIDATION

Validating the proposed dataset is not straightforward. Due to the novelty of multispectral light field research, there is no reference multispectral light field dataset nor benchmark applications available. To our knowledge, the only work using multispectral light fields directly is a recent paper by Zhou *et al.* [13]. However, in this instance, concentric light fields are employed, making the proposed disparity estimation incompatible with our dataset. On the other hand, we do not introduce new baseline algorithms for multispectral light field applications, as we believe this to be out of the scope of this paper. Overall, one faces a chicken-and-egg dilemma: to further advance multispectral light field applications, in particular data-driven ones, a sufficiently large dataset is needed. However, to validate the dataset directly, multispectral light field applications are needed. To overcome this dilemma, we validate the proposed dataset separately with respect to its

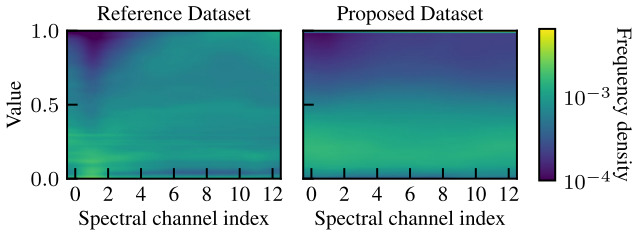

FIGURE 4. Spectral density of the reference multispectral and the proposed dataset. The density values are interpolated (bicubic) for visualization.

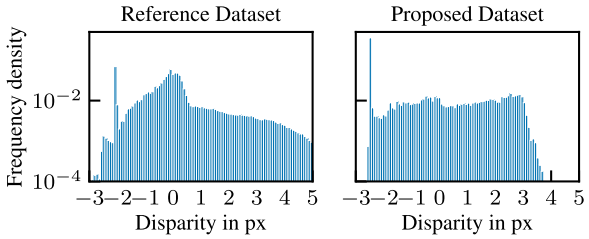

FIGURE 5. Disparity distribution of the reference RGB and the proposed dataset.

spectral distribution, its disparity distribution, and its applicability to light field deep learning.

First, we compare its spectral distribution with several datasets of multispectral images [26], [27], [32] which we downsampled to 13 spectral channels[2] and to which we collectively refer to as the ''reference'' dataset. Note that images of two of these datasets have also been used as multispectral textures in the rendering of our dataset. Both the proposed and the reference dataset have been normalized to a value range of $[0, 1]$ with 32 bit floating point precision. The resulting 2D histograms of the spectral distribution of the two datasets are shown in Figure 4. The proposed dataset shows a more balanced spectral distribution than the reference dataset, especially at lower spectral indices (corresponding to smaller wavelengths). While we do not want to judge which distribution is better suited for machine learning, it does reflect the design choices we have made upon the random scene generation as described in Section II-B. However, a peak at intensity values of one can be observed, likely stemming from overexposed regions of the used RGB image textures.

Second, we compare the dataset's disparity distribution with a dataset composed of previously published RGB light field datasets containing disparity ground truth [8], [18], [20] which we combined into a single dataset. Again, we refer to this composed RGB light field dataset as the ''reference'' dataset. The obtained histograms are shown in Figure 5. While the proposed dataset shows a stronger background peak at disparities around -2.5 px, the disparity distribution is overall more balanced and less biased towards a disparity of 0 px, corresponding to the focal plane. Again, this reflects the choices made in the scene generation, where object centers were placed uniformly in disparity.

Third, we evaluate three CNN architectures proposed in the literature for three distinct light field tasks: the EPINET model [1] for light field disparity estimation, the SAS-CONV model [5] for light field superresolution using separable angular-spatial convolution, and a CONV3D autoencoder model which is based on the autoencoder path of the model proposed by Alperovich *et al.* [8]. Additional to the distinct model tasks, each of these models also operates on different input shapes, reflecting the versatility of the proposed framework. Whereas the disparity and autoencoder models operate on (single or multiple) EPI volumes, the superresolution model uses the full light field as its input. Strictly speaking, the autoencoder model is hence an EPI volume autoencoder.

As mentioned before, all of these previous works are based on RGB rather than multispectral light fields. A direct evaluation of deep learning-based multispectral light fields is hence not possible, without introducing new baseline algorithms, wich is not within the scope of this paper. Each model is trained with both the reference and the proposed (RGB) dataset, and tested with both datasets' test data. The training duration is 150 epochs in the case of the proposed and 55 epochs in the case of the reference dataset. The discrepancy is due to the different sizes of the datasets: while the proposed dataset consists of roughly 80 000 light field patches, the reference RGB dataset contains about 215 000 patches and is hence larger by a factor of about 2.7. Hence, for comparability, we train the models such that they have been exposed to roughly the same amount of training data.

Throughout, we adopt the *1cycle* learning strategy recently proposed by Smith [33], [34]. In this cyclic learning approach, using a standard gradient descent (SGD) optimizer, the learning rate, starting with a pre-defined minimal value, is first increased to a comparably large value and again decreased to the initial value. The training concludes by further decreasing the learning rate to a very small value for fine-tuning the parameters. Together with this cyclic update of the learning rate, the momentum of the optimizer is updated inversely, starting from a large momentum, decreasing to a smaller momentum (to avoid divergence when the learning rate is large), back to a larger momentum. While Smith proposed to use a linear increase and decrease, we use a cosine annealing.

In order to find suitable values for the minimum and maximum learning rate (which depend on the model and the batch size under consideration), the learning range approach as described by Smith is employed: the corresponding model is trained for one epoch, starting with a very small learning rate of 10^{-7} and increasing the learning rate after each mini batch update, up to a large learning rate of 1. The resulting loss per mini batch is collected and plotted against the corresponding learning rate. The point of largest descent, *i.e.* the point of ''fastest'' learning, is used as the maximum learning rate. To this end, a moving average filter is applied to the resulting loss-over-learning rate graph. Finally, the smaller learning

[2] A Chebyshev type I filter of order eight was used as an anti-aliasing filter.

TABLE 1. Used batch sizes and learning rates for all models in the case of the RGB dataset validation and augmentation performance comparison.

Type	Model	Batch size	$\mathrm{lr}_{\mathrm{min}}$	$\mathrm{lr}_{\mathrm{max}}$	$\mathrm{lr}_{\mathrm{end}}$
Disparity	EPINET	256	$1 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	10^{-5}
Superresolution	SAS-CONV	32	$3 \cdot 10^{-4}$	$3 \cdot 10^{-3}$	10^{-6}
Autoencoder	CONV3D	256	$1 \cdot 10^{-3}$	$1 \cdot 10^{-2}$	10^{-5}

rate is chosen one order of magnitude smaller than the larger learning rate.

By using comparably large learning rates during the training, many applications have shown faster convergence and better generalization in the studies by Smith [33], [34]. In extreme cases, models were found to converge a factor of 10 faster when compared to regular training approaches, leading to *superconvergence*, as termed by Smith. While we did not observe extreme differences in convergence speed between the *1cycle* approach using SGD as compared to adaptive optimizers such as RMSprop or Adam, we did observe slightly better generalization. This is also in accordance to recent observations regarding adaptive optimization and generalization [35], [36]. Both the *1cycle* learning strategy as well as the learning rate finder are available in the proposed framework. We choose this training strategy because it is free of hyperparameters that have to be chosen carefully, requiring subjective judgment. For our application, we do not claim to train each model to their absolute best performance, but rather a converged state in which relative comparisons can be made. All models are trained employing the same cyclic learning strategy with individually chosen learning rate parameters and batch sizes which are shown in Table 1.

As the training loss, we employ the Huber loss with $\delta = 1$. The single element Huber loss is defined as

$$H_\delta(e_i) = \begin{cases} e_i^2\,, & e_i < \delta \\ 2\delta \cdot (e_i - \frac{1}{2}\delta)\,, & \text{else} \end{cases} \tag{2}$$

where $e_i = |\mathbf{y}_i^{\mathrm{true}} - \mathbf{y}_i^{\mathrm{pred}}|$ denotes the absolute prediction error of the i-th element for the vectorized prediction $\mathbf{y}_{\mathrm{pred}}$ with respect to the ground truth $\mathbf{y}_{\mathrm{true}}$. The overall Huber loss is then calculated as the mean of the element-wise Huber losses. Note that this definition deviates from commonly used definition (as for example implemented in TensorFlow) by a factor of two. We have chosen this, such that the Huber loss is truly identical to the MSE for errors smaller than δ, rather than MSE/2. Using this definition, for large errors and in the used case $\delta = 1$, the loss is identical to $2 \cdot \mathrm{MAE}$ and hence gradients are clipped by 2 instead of 1. Due to this effective gradient clipping, the Huber loss is less sensitive to outliers than the commonly used MSE.

Training was performed using a 32 GB Nvidia Tesla V100 GPU utilizing 10 cores and 96 GB RAM of an Intel Xeon Gold 6248-based computing node. Depending on the model, the training of each instance took between 12 h and 48 h.

The results of the dataset comparison are shown in Table 2. Here, the baseline test scores are calculated as the mean test scores of 10 runs when evaluating *untrained* instances of each model on the corresponding test datasets. The untrained model parameters are initialized using the method by He *et al.* [37]. This baseline can be used to more precisely judge the learning effect. The test scores evaluated on the reference dataset are indexed by ''ref'' whereas the ones evaluated on the proposed dataset by ''prop''. In the case of the autoencoder and superresolution models, the results are mostly independent of the used training dataset. While there is a slight generalization gap, *i.e.* testing on the complementary dataset results in slightly worse scores, it is not significant, validating the proposed dataset when compared to the reference dataset. It should also be noted, that the reference dataset contains almost three times as many unique training datapoints. In the case of disparity estimation, the generalization gap is much larger for both training datasets. Since this gap is more severe when evaluating with the MSE as compared to the MAE, we conclude that it is mostly caused by a few dominant outliers. This may be explained by the chosen Huber training loss. Another possible explanation of this gap may be the different disparity distributions of the two datasets. However, poor generalization can also be a flaw in the model's architecture. Still, overall, a significant learning effect can be observed.

Concluding, training artificial neural networks for different light field tasks is possible with the proposed dataset, achieving similar results as compared to a reference dataset that contained multiple of the previously published RGB light field datasets. Overall, we do not achieve state-of-the-art performance for any of these models, likely due to the comparably small number of training epochs or a sub-optimal training strategy (which also supports the previously made point about the difficulty of reproducing results on previously published network architectures). However, as already argued, the absolute performance of the networks is not our focus here but rather a relative comparison. Furthermore, we will investigate whether data augmentation can mitigate the generalization gap in the next section.

B. FRAMEWORK PERFORMANCE ANALYSIS

In the case of GPU-based machine learning, there are mainly three I/O-bound categories during training: (i) the data throughput when reading the data (either from disk or RAM), (ii) the speed of the preprocessing, including possible data augmentation and transferring of data onto the GPU, and (iii) the speed of processing one mini batch of samples (forward and backward pass) on the GPU. While for large models, the latter usually is the bottleneck upon the three categories, this may not always be the case. While bottlenecks of category (iii) are mitigated by either using faster GPUs, employing multi-GPU training, or using the hardware more efficiently (*e.g.* by using mixed precision training utilizing the so-called TensorCores of Nvidia GPUs), the first two categories can also be sped up by an efficient implementation

TABLE 2. Test performance evaluated on the proposed and the reference dataset for different training datasets in the case of autoencoder (AE), disparity estimation (DE), and superresolution (SR) models. The results denoted by the baseline datasets are obtained from untrained instances of the respective networks. Depticed values are mean squared error (MSE), mean absolute error (MAE), peak signal-to-noise ratio (PSNR) and the BadPix07 (BP07) metric.

Model	Dataset	MSE_{prop}	MSE_{ref}	MAE_{prop}	MAE_{ref}	$PSNR_{prop}$	$PSNR_{ref}$	$BP07_{prop}$	$BP07_{ref}$
AE	(Baseline)	$8 \cdot 10^6$	$5 \cdot 10^6$	1800	1530	4.7 dB	4.8 dB		
	Reference	0.0344	0.0207	0.1317	0.1037	14.8 dB	16.8 dB	-	-
	Proposed	0.0306	0.0224	0.1245	0.1133	15.4 dB	16.5 dB	-	-
DE	(Baseline)	3.9972	3.3319	1.8067	1.1382	-	-	98.23%	91.49%
	Reference	0.6490	0.2594	0.4041	0.1735	-	-	73.51%	48.00%
	Proposed	0.1480	1.2309	0.1385	0.4201	-	-	35.08%	62.90%
SR	(Baseline)	0.3149	0.2157	0.4117	0.3513	9.9 dB	11.0 dB	-	-
	Reference	0.0037	0.0022	0.0324	0.0283	27.9 dB	28.3 dB	-	-
	Proposed	0.0033	0.0021	0.0282	0.0258	28.1 dB	28.5 dB	-	-

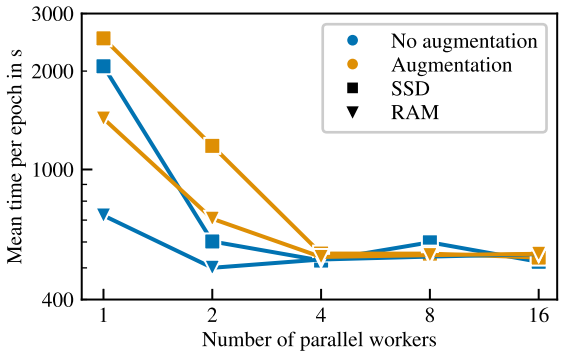

FIGURE 6. I/O-bound toy model speed evaluation reading from SSD or RAM, with and without augmentation.

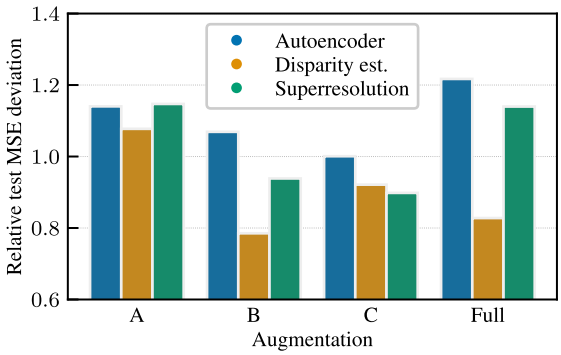

FIGURE 7. Test MSE when training with different data augmentations relative to a training without augmentation.

of the data input pipeline and augmentation. This is the case especially since common workstation hardware cannot hold full light field datasets (which easily reach several hundred GB in size) in the RAM. To this end, multiprocessing of the data generation is available in the proposed framework. To investigate the influence of multiprocessing and augmentation on the input speed performance, we train an input I/O-bound toy model, consisting of a single activation layer, for 11 epochs with a batch size of 128 light fields of shape (9, 9, 32, 32, 3), and measure the mean time per training epoch, disregarding the first epoch. We perform this training by both first loading the full dataset into RAM, as well as by streaming the data from an SSD directly, for the two cases with full and without any data augmentation. The results of this comparison are shown in Figure 6. As expected, using only a single process for data input and processing, the mean epoch time is much shorter when reading from RAM and when not performing augmentation. However, using only four parallel data generation processes, the mean epoch time in all cases is virtually the same. Of course, the absolute performance depends also on the used hardware, however, usually workstation CPU and RAM performance do not differ as significantly as the performance of the used GPU. As a recommendation for the proposed framework, the training should be performed using four data generation processes, maximizing the GPU workload.

C. DATA AUGMENTATION ANALYSIS

As we have previously illustrated, utilizing multiple processes, the implemented data augmentation does not have a significant impact on the training speed. However, the impact on the test performance has yet to be investigated. For this, we group the proposed augmentations into three categories: (A) those that are contiguous in memory (like gamma compression and channel weighting), (B) those that are not contiguous in memory but computationally trivial (such as flipping, rotation, and channel permutation), and (C) those that are computationally complex (*e.g.* scaling, which uses interpolation). Again using the previously introduced models for a light field autoencoder, disparity estimation and superresolution, we compare the relative test MSE of the instances when trained with different augmentations compared to the case without augmentation. The results, which are shown in Figure 7, are somewhat surprising. While category B (flipping, rotation, channel permutation) and C (scaling) in most instances show a lower (*i.e.* better) test MSE for most models, this is not the case for category A (gamma compression, channel weighting). Hence, the full augmentation mostly leads to a worse test MSE, depending on the application under consideration. Of course, this is the exact opposite behavior than which is expected but may be explained by the altered training set statistics, which are now expected to differ from the test dataset due to the data augmentation. However, a more

in-depth investigation of data augmentation in the case of light field-related deep learning applications is necessary, also with respect to non-synthetic light field data. As this is not within the scope of this contribution, and without further detailed insight, data augmentation should not be applied universally and carelessly.

V. CONCLUSION

We have proposed a novel multispectral light field dataset with depth and disparity ground truth as well as a new Python framework for light field deep learning. Comparing the dataset with existing RGB light field datasets and multispectral image datasets, we validated the proposed dataset's properties and applicability to the training of light field CNNs. We welcome contribution to the dataset, *e.g.* by providing newly rendered ground truth data (such as surface normals or 3D point clouds) or by extending the dataset with real-world multispectral light fields. Since the dataset is spectrally sampled in steps of 25 nm, off-the-shelf spectral bandpass filters can be employed *e.g.* in combination with a monochromatic light field camera or camera array to capture multispectral light fields in practice. We believe that the proposed dataset can accelerate the research in the emerging field of multispectral light field cameras and applications, in particular with respect to deep learning methods.

The proposed framework was investigated with respect to data processing and augmentation performance. Of course, the presented framework in its current form may not fit all light field-related deep learning applications. However, its object-oriented design is general and modular such that it can be easily extended and adapted–not only with respect to the network architecture, but also regarding the used data as well as custom training strategies. To this end, we invite the community to contribute modifications that may be necessary for workflows and architectures that we did not consider. It would be desirable to have a common light field deep learning framework such that research results can be easily reproduced and people can share new ideas and architectures or even contribute light field specific implementations such as a native 4D convolution.

With both the dataset and the framework, we believe we can mitigate or even eliminate the problems around reproducing light field deep learning applications which are often present in the current research landscape. By agreeing to a common yet flexible standard to develop and share new architectures, including the configuration and hyperparameters needed in order to reproduce a certain result, we are hopeful to enhance the overall quality and development speed of light field deep learning research.

REFERENCES

[1] C. Shin, H.-G. Jeon, Y. Yoon, I. S. Kweon, and S. J. Kim, "EPINET: A fully-convolutional neural network using epipolar geometry for depth from light field images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4748–4757.

[2] H. Ma, H. Li, Z. Qian, S. Shi, and T. Mu, "VommaNet: An End-to-End network for disparity estimation from reflective and textureless light field images," 2018, *arXiv:1811.07124*. [Online]. Available: http://arxiv.org/abs/1811.07124

[3] Y.-J. Tsai, Y.-L. Liu, M. Ouhyoung, and Y.-Y. Chuang, "Attention-based view selection networks for light-field disparity estimation," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12095–12103.

[4] Y. Wang, F. Liu, K. Zhang, G. Hou, Z. Sun, and T. Tan, "LFNet: A novel bidirectional recurrent convolutional neural network for light-field image super-resolution," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4274–4286, Sep. 2018.

[5] H. W. F. Yeung, J. Hou, X. Chen, J. Chen, Z. Chen, and Y. Y. Chung, "Light field spatial super-resolution using deep efficient spatial-angular separable convolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2319–2330, May 2019.

[6] N. Bakir, W. Hamidouche, O. Deforges, K. Samrouth, S. A. Fezza, and M. Khalil, "RDO-based light field image coding using convolutional neural networks and linear approximation," in *Proc. Data Compress. Conf.*, Mar. 2019, p. 554.

[7] M. Gupta, A. Jauhari, K. Kulkarni, S. Jayasuriya, A. Molnar, and P. Turaga, "Compressive light field reconstructions using deep learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 11–20.

[8] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke, "Light field intrinsics with a deep encoder-decoder network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9145–9154.

[9] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 41–48.

[10] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 606–619, Mar. 2014.

[11] J. Peng, Z. Xiong, D. Liu, and X. Chen, "Unsupervised depth estimation from light field using a convolutional neural network," in *Proc. Int. Conf. 3D Vis.*, Sep. 2018, pp. 295–303.

[12] K. Zhu, Y. Xue, Q. Fu, S. B. Kang, X. Chen, and J. Yu, "Hyperspectral light field stereo matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 5, pp. 1131–1143, May 2019.

[13] M. Zhou, Y. Ding, Y. Ji, S. S. Young, J. Yu, and J. Ye, "Shape and reflectance reconstruction using concentric multi-spectral light field," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 7, pp. 1594–1605, Jul. 2020.

[14] R. Horstmeyer, G. Euliss, and R. Athale, "Flexible multimodal camera using a light field architecture," in *Proc. IEEE Int. Conf. Comput. Photography*, Apr. 2009, pp. 1–8.

[15] J. Ye and F. Imai, "High resolution multi-spectral image reconstruction on light field via sparse representation," in *Imaging and Applied Optics*, 2015, Paper IT3A-4. [Online]. Available: https://www.osapublishing.org/abstract.cfm?uri=ISA-2015-IT3A.4

[16] Z. Xiong, L. Wang, H. Li, D. Liu, and F. Wu, "Snapshot hyperspectral light field imaging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3270–3278.

[17] Y. Xue, K. Zhu, Q. Fu, X. Chen, and J. Yu, "Catadioptric hyperspectral light field imaging," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 985–993.

[18] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 19–34.

[19] J. Shi, X. Jiang, and C. Guillemot, "A framework for learning depth from a flexible subset of dense and sparse light field views," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5867–5880, Dec. 2019.

[20] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3746–3754.

[21] V. Vaish and A. Adams, "The (new) Stanford light field archive," in *Computer Graphics Laboratory*, vol. 6, no. 7. Stanford, CA, USA: Stanford Univ., 2008.

[22] P. Paudyal, R. Olsson, M. Sjöström, F. Battisti, and M. Carli, "SMART: A light field image quality dataset," in *Proc. 7th Int. Conf. Multimedia Syst.*, 2016, pp. 1–6.

[23] T.-C. Wang, J.-Y. Zhu, E. Hiroaki, M. Chandraker, A. A. Efros, and R. Ramamoorthi, "A 4D light-field dataset and CNN architectures for material recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 121–138.

[24] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Comput. Sci. Tech. Rep.*, vol. 2, no. 11, pp. 1–11, 2005.

[25] T. Nürnberg, M. Schambach, D. Uhlig, M. Heizmann, and F. Puente León, "A simulation framework for the design and evaluation of computational cameras," *Proc. SPIE*, vol. 11061, Jun. 2019, Art. no. 1106102.
[26] S. M. C. Nascimento, K. Amano, and D. H. Foster, "Spatial distributions of local illumination color in natural scenes," *Vis. Res.*, vol. 120, pp. 39–44, Mar. 2016.
[27] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural RGB images," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 19–34.
[28] K. Greff, A. Klein, M. Chovanec, F. Hutter, and J. Schmidhuber, "The sacred infrastructure for computational research," in *Proc. 16th Python Sci. Conf.*, 2017, pp. 49–56.
[29] P. J. Huber, "Robust estimation of a location parameter," in *Breakthroughs in Statistics*. New York, NY, USA: Springer, 1992, pp. 492–518.
[30] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, Feb. 1997.
[31] C.-I. Chang, "Spectral information divergence for hyperspectral image analysis," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jun./Jul. 1999, pp. 509–511.
[32] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, "Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2241–2253, Sep. 2010.
[33] L. N. Smith, "Cyclical learning rates for training neural networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Mar. 2017, pp. 464–472.
[34] L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay," 2018, *arXiv:1803.09820*. [Online]. Available: http://arxiv.org/abs/1803.09820
[35] N. Shirish Keskar and R. Socher, "Improving generalization performance by switching from adam to SGD," 2017, *arXiv:1712.07628*. [Online]. Available: http://arxiv.org/abs/1712.07628
[36] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4148–4158.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

MAXIMILIAN SCHAMBACH received the B.Sc. degree in physics from Friedrich-Schiller-University Jena, in 2013, and the M.Sc. degree in physics from Leipzig University, in 2016. He is currently pursuing the Ph.D. degree with the Karlsruhe Institute of Technology, Germany. He is also a Research Associate with the Institute of Industrial Information Technology, Karlsruhe Institute of Technology. His current research interests include signal and image processing, computational imaging, and compressed sensing.

MICHAEL HEIZMANN received the M.Sc. degree in mechanical engineering and the Ph.D. degree in automated visual inspection from the University of Karlsruhe, Germany, in 1998 and 2004, respectively. From 2004 to 2009, he was a Postdoctoral Research Assistant with the Fraunhofer Institute of Optronics, System Technologies and Image Exploitation (IOSB), Karlsruhe, Germany, where he was the Head of the Department Systems for Measurement, Control and Diagnosis, from 2009 to 2016. From 2014 to 2016, he was a Professor of mechatronic systems with the Karlsruhe University of Applied Sciences. Since 2016, he has been a Full Professor of mechatronic measurement systems and has been the Director of the Institute of Industrial Information Technology, Karlsruhe Institute of Technology. His research interests include measurement and automation technology, machine vision and image processing, and image and information fusion.