Reconstruction from Spatio-Spectrally Coded Multispectral Light Fields

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

Maximilian Schambach, M.Sc.

geb. in Kassel

Tag der mündlichen Prüfung:	7. März 2022
Hauptreferent:	Prof. Dr.-Ing. Michael Heizmann, KIT
Korreferent:	Prof. Dr. rer. nat. Bastian Goldlücke, Uni Konstanz

Preface

My time at KIT was coined by many people whose help and support contributed to the successful completion of this thesis.

I want to thank Prof. Dr.-Ing. Fernando Puente León who was my supervisor in the first years. While creating space for personal research interests, he gave the right impulses and connections at the right time. Without his persistent pursuit for collaboration, this thesis, which at its core was embedded in a joint research project, would not have been possible. I will remember him as a free spirit, following his own (scientific) agenda and encouraging his students to do the same.

I thank Prof. Dr.-Ing. Michael Heizmann for taking over the supervision for the last two years. Despite the comparably short time, this period was a very productive one for me personally, embedded in the friendly and supportive environment that he created. Especially in a time of uncertainty, for me and IIIT in general, Prof. Heizmann came through for everyone and left no doubt about the continuation of all Ph.D. and research projects, for which I am very grateful.

I thank Prof. Dr. rer. nat. Bastian Goldlücke for co-reviewing this thesis. There likely is no one more active in the German light field research community und thus fitting to review this research.

Furthermore, I want to thank Prof. Dr. rer. nat. Uli Lemmer for co-initiating the joint research project, which enabled the research of this thesis. I hope the fruitful collaboration between IIIT and LTI continues.

There are many more people at IIIT I want to thank: I thank all colleagues, current and past, at IIIT for the friendly working environment and joyful moments, in particular those beyond work. In particular, I thank Johannes Anastasiadis, Matthias Bächle, Manuel Bihler, Muen Jin, Lanxiao Li, Markus Schwabe, Erik Tabuchi Barczak, David Uhlig, and Hannes Weinreuter for proofreading parts of the manuscript. I thank my students for contributing to this work by exploring different possible research directions and supporting me with their findings.

I want to thank the staff at IIIT, namely Dieter Brandt, Manuela Moritz, Patricia Nestl, and Marvin Winkler, for the organizational and technical support throughout the years. I am also grateful to the team at IIIT mechanical workshop, Alexander Hettinger, Jochen Ritter, Simon Rothardt, and Stefan Ziegler, for their excellent support and the fabrication of the spectral light field camera used in this thesis.

I thank the Baden-Württemberg Stiftung for financing the research project "compact multispectral light field camera", in which large parts of the presented research was carried out. Within this project, I particularly thank Qiaoshuang Zhang at LTI for the collegial and productive collaboration. In addition, I acknowledge support by the state of Baden-Württemberg through bwHPC.

Furthermore, I want to thank my friends, namely Janos Borst and Jonas Köhler, whose discussions about different topics within and beyond the scope of deep learning contributed significantly to this research.

Last but not least, I am deeply grateful to my family, in particular, my parents Petra and Udo, Julian, and Sabine, for their ongoing support.

Having spent a total of nearly 12 years at different universities, this thesis marks the end of a large chapter of my journey. Thanks to everyone who joined along the ride!

Karlsruhe, March 2022 Maximilian Schambach

Contents

Nomenclature

Common abbreviations

Abbreviation	Description
cf.	*Confer* (lat. refer to)
e.g.	*Exempli gratia* (lat. for example)
et al.	*Et alii* (lat. and others)
etc.	*Et cetera* (lat. and so forth)
i.e.	*Id est* (lat. that means)
AL	Auxiliary loss
AU	Arbitrary unit
BP	Bad-pix (metric)
CCD	Charge-coupled device
CNN	Convolutional neural network
CS	Compressed sensing
DCT	Discrete cosine transform
EPI	Epipolar plane image
GradSim	Gradient similarity
NormGradSim	Normalized gradient similarity
GT	Ground truth
MAE	Mean absolute error
MLA	Microlens array
MSE	Mean squared error
MT	Multi-task
PSNR	Peak signal-to-noise ratio
SA	Spectral angle
SID	Spectral information divergence
SSIM	Structural similarity index measure
ST	Single-task

Symbols

Latin letters

Symbol	Description
$\mathbf{1}$	Identity matrix
$\mathbf{x}$	Vector
$(\mathbf{x})_i$	Element i of $\mathbf{x}$
$\mathbf{A}$	Matrix
$\mathbf{A}_{ij}$	Element (i, j) of $\mathbf{A}$
$\boldsymbol{\mathcal{T}}$	Tensor, shape $(I, J, \dots, K)$
$\boldsymbol{\mathcal{T}}_{ij\dots k}$	Tensor, index notation
$\mathcal{T}[i, j, \dots, k]$	Tensor, functional notation
$\boldsymbol{\mathcal{D}}$	Central disparity map, shape (S, T)
$\boldsymbol{\mathcal{E}}_{u_0}$	Horizontal EPI volume, shape (V, S, T, Λ)
$\boldsymbol{\mathcal{E}}_{v_0}$	Vertical EPI volume, shape (U, S, T, Λ)
$\boldsymbol{\mathcal{I}}$	Central view, shape (S, T, Λ)
L	Loss function
$\boldsymbol{\mathcal{L}}$	Light field, shape (U, V, S, T, Λ)
$\boldsymbol{\mathcal{L}}^*$	Coded light field, shape (U, V, S, T, Λ)
$\boldsymbol{\mathcal{L}}^*_{\mathrm{p}}$	Coded and projected light field, shape (U, V, S, T)
$\boldsymbol{\mathcal{M}}$	Spectral coding mask, shape (S, T, Λ)
u, v	Angular light field coordinates
U, V	Angular light field resolution
s, t	Spatial light field coordinates
S, T	Spatial light field resolution

Greek letters

Symbol	Description
ϵ	Arbitrarily small number
λ	Spectral light field coordinate
Λ	Spectral light field resolution

Superscripts

Index	Description
$(\bullet)^{\mathrm{T}}$	Transposed

Subscripts

Index	Description
$(\bullet)_{\mathrm{aux}}$	Auxiliary
$(\bullet)_{\mathrm{cv}}$	Central view
$(\bullet)_{\mathrm{disp}}$	Disparity
$(\bullet)_{\mathrm{reg}}$	Regularization

Mathematical operators

Operator	Description
$**$	2D convolution
$\circ\!\!-\!\!\!-\!\!\bullet$	2D Fourier transform
$\odot$	Element-wise multiplication
$\otimes$	Kronecker product, tensor product
$\hat{\cdot}$	Estimated value
$\mathbb{E}[\cdot]$	Expected value
$\lVert\cdot\rVert$	Euclidian norm
$\lVert\cdot\rVert_{\mathrm{F}}$	Frobenius norm
$\lVert\cdot\rVert_{p}$	p-norm
$\lVert\cdot\rVert_{p,p}$	p, p-matrix norm
$\lVert\cdot\rVert_{0}$	l_0-"norm"
$\langle\cdot,\cdot\rangle$	Scalar product
$\lceil\cdot\rceil$	Ceiling function
$\lfloor\cdot\rfloor$	Floor function

1 Introduction

Vision is one of the core human senses and in many ways considered our primary. Using our powerful visual system, we are able to solve complex visual tasks intuitively and from an early age, ranging from recognition and classification to scene understanding and decision-making, to name a few. Throughout, we humans incorporate an immense amount of contextual information—visual information but also physical, historical, psychological, sociological, and cultural. For example, we can, with little effort, perceive and understand a complex inner-city crossing with cars, bicycles, pedestrians, and important visual cues, such as traffic lights, road signs, or crosswalks, even intuitively building a model to predict the configuration in the near future. To his end, we build on our large domain knowledge: We implicitly incorporate the corresponding physical laws, *e.g.* predicting the possible trajectories of the participants, even keeping track of temporarily occluded ones; we consider the applicable laws, fundamental ones and those communicated via traffic lights or road signs; we take into account sociological circumstances, for example children or cyclists may not act as expected by law; and we acknowledge cultural cues such as a car flashing or a person nodding to give way.

Nevertheless, we also suffer from some weaknesses, limiting our potential in a vision-based context. First, we cannot work arbitrarily fast, *i.e.* we are limited in our processing speed and in particular suffer from a significant delay—our reaction time. Second, we suffer from fatigue, *i.e.* we cannot work with full concentration for arbitrary periods of time. As we get tired or bored, our performance drops significantly, in particular when assigned to monotone, long-lasting, or undemanding tasks. And third, by definition, our visual sense is limited to the visible range of the electromagnetic spectrum. Therefore, we cannot process optical information in other ranges and depend on good visibility. In particular, visual perception is challenging in heavy rain, fog, snow, due to glare, or at night.

To overcome these limitations, scientists and engineers have developed technical solutions to visual perception. The corresponding research fields are broadly collected under the term *computer vision*, including a wide range of areas from hardware and sensor design and calibration, understanding and modeling of the corresponding optics, to image processing, in particular feature or object detection, pattern recognition, and other high-level applications related to scene understanding or geometry. In the past decades, computer vision applications have successfully been applied to numerous problems in medicine, *e.g.* imaging techniques such as computed tomography or the detection of skin cancer [125], engineering, *e.g.* in robotics and autonomous driving [94, 146], agriculture, *e.g.* remote sensing or smart farming [23, 195], surveillance, *e.g.* people detection or flow analysis [6], to industry, *e.g.* optical quality assurance or automated sorting [190], to name a few.

As a basis of computer vision applications, high-quality data is needed. To this end, several optical measurement techniques and sensors exist to capture the spatial and spectral characteristics of a scene. For example, conventional color cameras capture a comparably low-dimensional projection of the available information—detailed spectral and geometrical information, *e.g.* depth, are not directly available. In the context of computer vision, cameras, *i.e.* single- or multi-sensor systems providing contact-free optical measurements of a scene, can be roughly divided into two classes: Color or spectral cameras and imaging systems for distance measurement, which are referred to as 3D cameras in the following. Cameras that belong to both classes are referred to as spectral 3D cameras or spectral depth imagers. These are the focus of this thesis.

Spectral cameras

Spectral cameras capture the spatially resolved spectrum of a scene. Depending on the number of spectral channels, one distinguishes between multi- and hyperspectral cameras. Multispectral cameras capture three to about 15 color channels. Systems with a higher spectral resolution are referred to as hyperspectral, however, the transition between the two definitions is smooth. Since, within this thesis, the distinction between multi- and hyperspectral cameras or images is not of importance, they are collectively referred to as *spectral*. Color or RGB cameras, which mimic the imaging of the human visual system, are widely used in everyday life, in order to measure the true-color representation of a scene. Strictly speaking, they differ from spectral cameras in that they capture color, *i.e.* the sensory impression corresponding to human perception, and not the physical spectrum. Even for RGB cameras, true color sensors, *i.e.* sensors directly measuring the pixel-wise color value, are already the exception. Only Foveon, Inc., manufactured a consumer-grade sensor, measuring all three RGB values per pixel, using the different energies and corresponding absorption depths of the incoming photons. Most RGB cameras, however, are based on color filters in order to measure the three color channels separately or in a coded fashion. High-resolution cameras use beam splitters, color filters, and three separate sensors to measure an RGB image. Compact cameras, on the other hand, use color-coding masks integrated onto the sensor, the so-called color filter array (CFA), to measure a different part of the spectrum of the scene at each pixel. The Bayer pattern [16] is by far the most widely used. To obtain a complete color image, a so-called demosaicing has to be performed.

While RGB images offer the advantage of being intuitively interpretable by humans, some objects or materials cannot be distinguished with this coarse spectral resolution. In order to achieve a higher spectral resolution, various approaches exist. Spectral cameras widely used in industry and science are either spatially or spectrally *scanning* cameras. Common to these cameras is that the information to be captured is reduced by one dimension to increase the spatial or spectral resolution. The dimension reduced during acquisition is measured in a scanning process. For example, spectral line scan cameras, also known as push broom scanners, measure a 1D spatially resolved spectrum and capture the remaining spatial dimension in a scanning fashion. Using a prism or grating, the spectral dependence is coded onto one of the two spatial coordinates of the sensor, leaving the other to measure the 1D spatial dependence. On the other hand, spectrally scanning cameras capture each spectral channel separately, *e.g.* using a filter wheel and different bandpass filters. Spatio-spectrally scanning cameras also exist [74], however they are less common. These cameras usually offer high spatial and spectral resolutions. However, due to the scanning involved, they are only suitable to capture static or low-dynamic scenes.

For dynamic scenes, *snapshot* spectral cameras are required. These include cameras that generalize the three-channel CFA approach to the multispectral case. This leads to sensors with multispectral filter arrays (MFAs) [65, 148] and corresponding demosaicing methods [147] which have found their way into end-user products. Instead of using absorbing filters, as in the case of CFAs, interference-based filters, *e.g.* Fabry-Perot or dielectric stack filters, are generally used in the multispectral case to realize narrow spectral bandpass filters. Earlier approaches are based on large-area spectral filters, so-called tiling MFAs [64], and complex beam-splitting optics. Other conceptual approaches are based on Fourier filter arrays to measure the multispectral images by sampling in the Fourier rather than the conventional spectral domain [96].

Other spectral snapshot cameras are mostly based on complex optics to encode the spectral information on the sensor. In addition to non-compressive methods [20], compressive hyperspectral cameras have been the focus of research in the past decade. Here, following the principles of compressed sensing, the spectral information is encoded using elaborate optics prior to the acquisition. For example, the coded aperture spectral snapshot imager (CASSI) and its derivatives [10, 66, 119, 207]) have been thoroughly studied. These cameras exploit the compressibility of natural images to reconstruct data sampled well below the Nyquist frequency. This usually provides a good tradeoff between spatial and spectral resolution. What these approaches have in common, however, is the complexity in the optical design: Prisms, gratings, coded apertures, MFAs, and digital micromirror devices are often used for encoding, which makes the camera setup complex and non-portable, requiring elaborate calibration.

3D cameras

Early developments of 3D cameras, dating back to the 1940s, follow the principles of stereo vision of the human visual apparatus: Two lenses fixed at a distance of a few centimeters simultaneously take images of a scene [168]. These images then contain depth information due to the different viewing angles of the scene, the so-called parallax. The stereo image pairs could then be viewed using special stereoscopes [169]. This principle, which was initially developed for the consumer market, was further developed for scientific and industrial applications for which quantitative depth measurements are required. Many 3D cameras are based, like the stereo camera, on depth measurement via triangulation: A plane triangle is uniquely determined by two angles and a side length [18]. By identifying a point in the image pairs and knowing the distance between the lenses, the so-called baseline, the distance to the object point can be calculated. Numerous further developments, from point and line to area scanning using active illumination (*e.g.* structured light) in the infrared or visible range, which are used in current cameras, are based on triangulation.

An alternative method to capture 3D information is to measure the time-of-flight (ToF) of an emitted electromagnetic pulse, *e.g.* light or radar, which is reflected by an object and detected by the camera sensor. The distance to the object is calculated from the ToF and the speed of light. Due to the high demands on the time resolution of the sensors and signal processing chain, ToF cameras are an ongoing research subject [81, 186]. In particular, ToF cameras are well suited for applications where a high frame rate is required. Radar and lidar systems, on the other hand, which are likewise based on the ToF principle, are widely used in fields where high-precision depth, and possibly velocity, measurements are required.

In addition, there are techniques that enable monocular 3D depth imaging. For example, these include cameras with coded apertures, light field cameras, or cameras with specifically designed diffractive elements. Following either the depth-from-focus or depth-from-defocus principle, the depth dependence of the imaging properties of real lenses is exploited: objects in the focal plane are sharply imaged while those imaged out of focus are blurred. While monocular, some of the approaches require the measurement of a focal stack, *i.e.* an image series using different focal planes. However, with the help of specially designed apertures and/or color filters or a deep learning-based reconstruction, snapshot cameras also exist [15, 110, 144].

Light-field cameras, on the other hand, directly record the angular dependence of the light rays reaching the sensor. Being of main interest in this thesis, light field cameras are discussed in more detail in Chapter 2. The advantage of monocular 3D cameras over multi-camera systems is their compactness and mechanical robustness. Calibration usually has to be performed once under laboratory conditions.

Spectral 3D cameras

Spectral 3D cameras provide a near all-encompassing measurement of the optical and geometrical properties of a scene and are the subject of cutting-edge research—commercial solutions are currently not available. Yet, there are a number of approaches to spectral depth imaging.

One possibility is to use multi-camera arrays and spectrally code the individual cameras for example using optical bandpass filters. As it is comparably straightforward to implement, several prototypes have been proposed [67, 68, 236]. In fact, spectrally coded multi-camera arrays can be interpreted as measuring a spectrally coded light field, as detailed in Section 2.3. However, multi-camera setups are costly, bulky, and difficult to calibrate. Due to their comparably large dimension, they are usually only suited in laboratory or specific industrial environments. Furthermore, their complexity increases as a higher spectral resolution is needed since every spectral band is sampled by an individual camera.

Other approaches are based on hybrid camera systems, *e.g.* using both RGB and spectral cameras [224] or using a separate light field camera and a CASSI [221]. Recently, complex compressive methods, extending the CASSI approach to spectral depth imaging, have been investigated [135]. However, for now, these methods are only feasible under laboratory conditions and often require a complex optical setup.

Finally, multi-modal cameras such as spectral depth cameras can be realized by directly optimizing the design of the used optics in the digital domain, in terms of an end-to-end optimization. These approaches are collected under the term *deep optics* [216]. For example, a spectral depth camera using free-form-optimized diffractive lenses was recently proposed by Baek *et al.* [13]. To the best of the author's knowledge, this is the only previously proposed compact monocular spectral depth snapshot camera. In this thesis, an alternative approach to compact monocular spectral depth imaging is considered.

Following a concept briefly introduced by Ye and Imai [226], spectrally coded light fields, as captured by a camera with a spectrally coded microlens array (MLA), are investigated here. This approach is appealing in terms of its compactness and flexibility: The MLA and sensor are tightly integrated and embedded in an ordinary camera housing, making it mechanically robust. Therefore, the camera only requires one-time calibration under laboratory conditions. Combining spectral and depth measurements into a single monocular architecture offers several advantages and possible applications. First, the multi-modal measurements are aligned by design, making an additional registration or calibration step superfluous. Second, the different imaging modalities allow for a fusion of applications from depth and spectral imaging, *e.g.* segmentation, classification, scene understanding, surface reconstruction, *etc*. Furthermore, the additional spectral channels and angular views could be used to apply techniques from active imaging systems, such as structured illumination used by active depth imagers or directional illumination as used in photometric stereo—in particular in the non-visible near-infrared (NIR) range. This way, depth estimation or surface normal estimation on untextured or specular objects could possibly be improved.

However, this thesis is not concerned with a specific application but the general signal reconstruction problem arising when using a spectrally coded light field camera. In fact, the spectral domain is usually not taken into account in previous works on coded light fields, as elaborated in detail shortly. Furthermore, reference data, *i.e.* spectral light fields, possibly with additional depth ground truth, and baseline methods are not available. This thesis shall fill this void, providing a reference, strong baseline, and suitable datasets—hopefully, to spark interest to enhance or apply this novel spectral depth imaging technique.

1.1 Contributions

The main contributions of this thesis are as follows.

- Two approaches to reconstruction from spectrally coded light fields are investigated and extensively evaluated:
    - A reconstruction of the spectral light field within the compressed sensing framework is presented. To this end, a new tensor-based dictionary learning method is developed for spectral light fields, which is shown to outperform the conventional vectorized dictionary as well as the representation using fixed bases.

    - Furthermore, a novel deep learning-based approach is developed, directly estimating the spectral central view and its aligned disparity map from the coded measurements, which is shown to outperform the compressed sensing-based reconstruction and subsequent disparity estimation. High reconstruction qualities are achieved. Despite being estimated from coded measurements, the reconstructed disparity performs on-par or even better than state-of-the-art disparity estimation methods from uncoded RGB light fields.

- A novel regularization scheme based on adaptively weighted auxiliary losses is developed, using normalized gradient similarity, which is shown to enhance the performance of both the single-task and multi-task deep learning-based reconstruction in the considered case. The approach can be combined with adaptive multi-task training strategies, which is shown to further enhance the overall performance.

- Several spectral coding masks are investigated and a novel differentiable fractal coding mask generation is proposed, which can be optimized together with the deep learning-based reconstruction in an end-to-end fashion.

- A large synthetic spectral light field dataset with disparity ground truth is created. The dataset consists of spectral light fields from randomly generated scenes, suitable for the training of data-driven approaches as well as a quantitative evaluation of the investigated reconstruction using a test dataset. Furthermore, spectral light fields and their disparity rendered from hand-crafted scenes are provided to assess the performance in more detail.

- A real-world spectral light field dataset is created using a custom-built spectral light field camera, for which a novel radiometric calibration is developed. The fully sampled reference data allows for a quantitative evaluation of the investigated reconstruction methods.

- A refinement of the geometric (pre-)calibration of MLA-based light field cameras is developed, specifically taking into account natural and mechanical vignetting, which is shown to increase the accuracy of well-established subsequent calibration approaches. For its evaluation, a synthetic dataset of so-called white images is created with available ground truth microlens center coordinates.

Parts of several sections of this thesis have been previously published and/or presented at conferences:

- The synthetic dataset, as presented in Section 4.1, and a general deep learning framework, parts of which are presented in Section 3.2, were published in [A6].
- Contributions to the geometric calibration of MLA-based light field cameras, as presented in Section 4.2.3, were published in [A9].
- The reconstruction via multi-task deep learning, as presented in Section 3.2 and evaluated in Section 5.1, was published in [A1]. This also contained supplementary material regarding the camera prototype and its radiometric calibration, as presented in Section 4.2.2, as well as parts of the compressed sensing-based reconstruction, which are presented in Section 3.1 and evaluated in Section 5.1.

However, for the presentation in this thesis, most parts have been significantly modified and extended. In particular, the results presented here are more exhaustive as compared to the original publications.

Finally, it should be noted that large parts of the work presented here were conducted within a joint research project of the Institute of Industrial Information Technology and the Light Technology Institute of the Karlsruhe Institute of Technology. While the ultimate goal of this project is to build a working prototype of a light field camera with a spectrally coded MLA, the corresponding research and (preliminary) results are not presented here as they would not be the sole contribution of the author. In particular, a fully functional hardware integration has yet to be achieved.

1.2 General remarks

Digital supplement

A digital supplement to this thesis is made publicly available[1]. The supplement contains references to the developed Python frameworks, the created datasets, as well as the source code to reproduce the investigated experiments. Furthermore, as not all results could be visualized and shown here, an interactive Jupyter notebook is provided. This notebook gives full access to the obtained reconstruction results of all experiments for further evaluation. In particular, this includes the reconstruction results for all scenes of the evaluation datasets and additional evaluation metrics, which are not included in the results presented here for clarity.

RGB conversion

Since spectral images are difficult to visualize in general, usually their corresponding RGB conversions are shown throughout this thesis. To this end, RGB conversion from spectral data is performed according to the CIE 1931 standard. Here, the color matching functions and D65 standard illuminant provided by the Institute of Ophthalmology at University College London [184] are used. Subsequently, XYZ-to-RGB conversion is performed following the sRGB standard [5].

Preprints

It should be noted that, throughout this work, great care is taken to cite only relevant, well-established, and peer-reviewed research—except in those rare cases where it is inevitable to refer to a preprint. While it is usually not considered best practice to cite non-peer-reviewed research papers, in particular arXiv preprints, it cannot be avoided in some instances. For example, this is the case for recent works in the context of deep learning, which is extremely rapidly developing.

[1] https://maxschambach.github.io/thesis

1.3 Thesis outline

The remainder of this thesis is organized as follows.

Chapter 2 provides an introduction to light fields and their applications in general. Furthermore, different coding approaches, in particular the spatio-spectral coding of spectral light fields considered in this thesis, are discussed in detail. Finally, the camera model, using a spectrally coded MLA, and the corresponding signal model are introduced.

The reconstruction methods are developed in Chapter 3. This includes the reconstruction of the underlying spectral light field within the compressed sensing framework as well as a deep learning-based reconstruction of the spectral central view and its aligned disparity map. Within the compressed sensing-based framework, several approaches to sparsely represent the underlying spectral light fields are discussed. Besides the conventional methods using fixed bases, a vector-based dictionary learning approach is investigated and refined using a tensor-based separation of the angular, spatial, and spectral components of the light field. Moreover, a novel deep learning-based reconstruction is motivated and discussed in detail, in particular the challenges arising regarding the multi-task training as well as a regularization using auxiliary losses. To this end, a new adaptive auxiliary loss weighing is proposed. Finally, an approach to optimize the coding mask in an end-to-end fashion, called neural fractals, is developed.

In Chapter 4, the experimental setup is discussed. In particular, the created reference datasets—a synthetic and a real-world spectral light field dataset—are introduced. To capture the real-world dataset, a custom spectral light field camera was built, whose calibration is discussed in detail. Finally, the used evaluation metrics and hyperparameters are elaborated.

The reconstruction results are presented in Chapter 5. This includes the compressed sensing-based reconstruction as well as the one based on deep learning. In the case of the deep learning-based reconstruction, the multi-task and auxiliary loss training strategies are evaluated in detail. Furthermore, several ablation studies, investigating the dependence on noise, the angular resolution, and the coding mask, are performed.

In Chapter 6, the presented work is summarized and an outlook is presented, discussing possible future research directions.

2 Spectral Light Fields

The *plenoptic function* $P_{\lambda,t_0}(\mathbf{x}, \mathbf{\Omega}) : \mathbb{R}^5 \rightarrow \mathbb{R}$ describes the light flow at point $\mathbf{x} \in \mathbb{R}^3$ in direction $\mathbf{\Omega} \in \mathbb{R}^2$ at a time instance t_0 per unit wavelength λ. The value of the plenoptic function is the *spectral radiance* in units W/(sr m^2 nm). Sometimes, the plenoptic function is also referred to as the *5D light field*. The propagation of these light rays is described within the theory of geometrical optics, *i.e.* based on Fermat's principle, whereas the spectral radiance is a property emerging in wave optics. In that sense, the plenoptic function provides a heuristic extension of geometrical optics incorporating some properties from the higher-order theory of wave optics, namely color and light intensity. Other higher-order effects such as diffraction or polarization cannot be described. Nevertheless, the plenoptic function is well suited to describe macroscopic scenes, *e.g.* for novel view synthesis, or the image formation of imaging systems for which diffraction and polarization can be neglected. A schematic overview of such higher-order effects and the corresponding physical theories is given in Figure 2.1. In the remainder of this thesis, the explicit time dependence is neglected since it is irrelevant in the case of snapshot imagers that are considered here.

Most generally, the plenoptic function can be used to synthesize arbitrary views of a scene. However, its redundancy and high dimensionality pose great challenges in practice. To this end, obtaining the plenoptic function from a sparse set of measurements using efficient representations is an ongoing research endeavor. Recently, implicit representations via artificial neural networks, such as neural radiance fields [137, 143] or the neural lumigraph [101], have become the state-of-the art in the case of (implicit) scene representation and novel view synthesis.

When describing imaging systems, however, the volume in which the plenoptic function ought to be described is much smaller as only rays reaching the camera sensor are of interest. This allows for more specific and efficient representations. In the case of homogeneous media

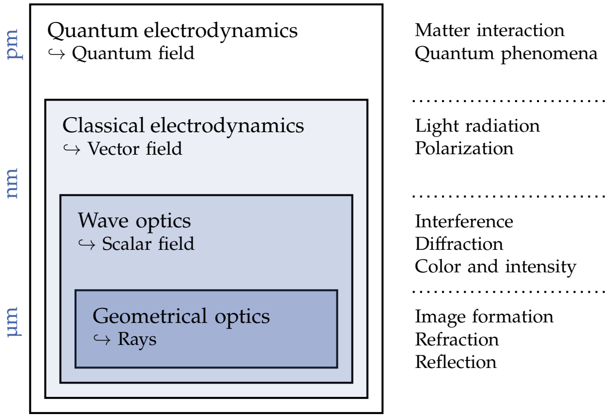


Figure 2.1 Overview of the different physical theories of optics together with their associated mathematical description and phenomena.

and non-occluding scenes, the spectral radiance along a given ray is constant. By dividing out the equivalence class of these constant-valued light rays, the domain of the plenoptic function can be reduced from five to four spatio-angular dimensions. The resulting *spectral 4D light field* $\mathcal{L}(u, v, s, t, \lambda)$ was first introduced by Moon and Spencer [149] and later coined by Levoy and Hanrahan [111], and is also sometimes referred to as the *lumigraph* [71]. As this thesis deals with spectral light fields, the spectral dependence is now made explicit while historically only the 4D spatio-angular coordinates were considered. The coordinates (u, v, s, t) correspond to a certain parametrization of the spatio-angular dependence of the light field of which there are numerous. In this thesis, as is common in the context of computational cameras, the so-called plane-plane parametrization is used: A light ray is uniquely described by the intersection points of two parallel planes at a given distance I using the angular coordinate (u, v) and the spatial coordinate (s, t). A schematic depiction of the plane-plane parametrization is given in Figure 2.2. Note that rays parallel to the planes cannot be parametrized. However, as the two planes are typically chosen to be parallel to the main lens and sensor plane, these rays do not contribute to the image formation and can be neglected.

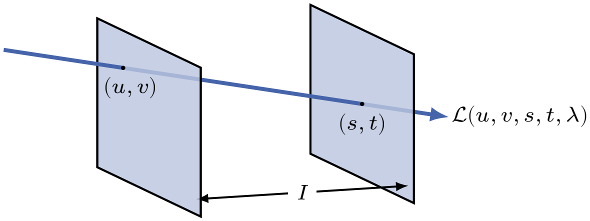
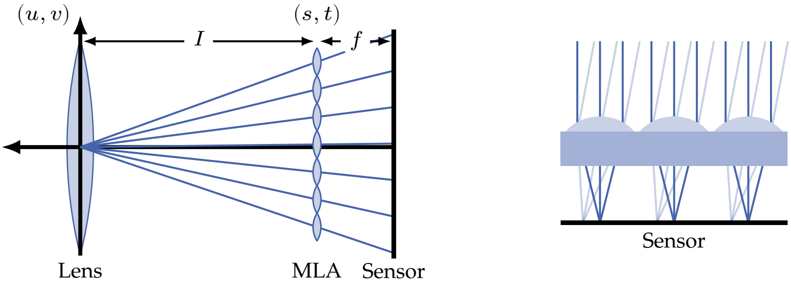


Figure 2.2 Plane-plane parametrization of the spectral 4D light field.

2.1 Light field acquisition

There are many camera designs to sample the continuous spectral light field $\mathcal{L}(u, v, s, t, \lambda)$. As opposed to conventional cameras, light field cameras also capture the angular dependence of a scene. In that sense, they can be regarded as a multi-view generalization of stereo cameras. Multi-camera arrays [217, 223] are hence a straightforward approach to achieve high-resolution light field imaging. However, camera arrays are bulky, expensive, and prone to mechanical changes that require recalibration.

As conventional CCD or CMOS imaging sensors are intrinsically two-dimensional, the five-dimensional spectral light field signal has to be coded onto the sensor in order to obtain a single-sensor light field camera. To this end, compact snapshot (RGB) light field cameras based on MLAs have been introduced by Adelson and Wang [1] and further compactified by Ng *et al.* [154], which were subsequently commercialized by Lytro, Inc. In this so-called *unfocused design*, an MLA is placed at the imaging distance of a regular main lens, and the imaging sensor is placed in the focal plane of the MLA. This way, the spatial dimension of the light field is sampled by the individual microlenses which can be interpreted as spatial macropixels. The angular dimension on the other hand is sampled by the sensor pixels underneath each microlens as depicted in Figure 2.3. Due to the small size of the microlenses, with typical diameters in the range of 20 to 60 µm, and their small focal lengths, the main lens can be viewed as being placed at optical infinity with respect to the microlenses. Therefore, incoming rays from a specific angular coordinate (u, v), parametrized at the main lens plane, are effectively parallel. These parallel rays are then focused onto the sensor pixels. For example,



(a) Side view.

(b) Detailed view of the MLA.

Figure 2.3 Model of the unfocused light field camera.

rays from the central angular coordinate, *i.e.* the main lens center, are focussed onto the central pixel underneath each microlens. Analogously, rays with a different incident angle are imaged onto pixels at an offset with respect to the central microlens, depending on the corresponding angular coordinate. In order to avoid angular crosstalk between different microlens images, the f-numbers of the main lens and the microlenses have to be matched [154]. This also effectively determines the field of view of the system. To be able to decode the multiplexed light field measurement, the individual microlens centers have to be detected such that the different angular views can be rearranged into a light field. Details on the geometric calibration of the camera and the subsequent light field decoding are presented in Section 4.2.3.

In this unfocused design, the spatial resolution is determined by the number of microlenses that are imaged onto the sensor, while the angular resolution is given by the number of pixels underneath each microlens. Hence, there is a fixed tradeoff between spatial and angular resolution for an imaging sensor of a given size and resolution.

Another MLA-based light field camera design is the so-called *focused design* which was first introduced by Lumsdaine and Georgiev [127] and later commercialized by Raytrix GmbH. In this design, the sensor is not directly placed in the focal plane of the MLA but at some offset that allows for a tunable tradeoff between the angular and the spatial resolution. The resulting spatio-angular coding of the light field, however, is very different from the unfocused design and not suited for the spectral coding which is considered in this thesis, as will be discussed shortly. Therefore, the focused design is not further elaborated.

After acquisition and decoding, one obtains the discretely sampled spectral light field $\mathcal{L}[u, v, s, t, \lambda]$. For any fixed angular coordinate (u_0, v_0),

$$\mathcal{I}_{u_0 v_0}[s, t, \lambda] = \mathcal{L}[u_0, v_0, s, t, \lambda] \tag{2.1}$$

is a conventional (spectral) image called a *subaperture view* of the light field. Hence, as previously noted, a light field can be viewed as a collection of subaperture views. A 2D section of the 4D light field, when fixing one angular and one spatial coordinate,

$$\mathcal{E}_{u_0 s_0}[v, t, \lambda] = \mathcal{L}[u_0, v, s_0, t, \lambda]\,, \quad \mathcal{E}_{v_0 t_0}[u, s, \lambda] = \mathcal{L}[u, v_0, s, t_0, \lambda]\,, \tag{2.2}$$

is a so-called *epipolar plane image* (EPI). An EPI may be horizontal or vertical, depending on which angular coordinate is fixed. Fixing solely the angular coordinate, one obtains a 3D section of the 4D light field, called an *EPI volume*,

$$\mathcal{E}_{u_0}[v, s, t, \lambda] = \mathcal{L}[u_0, v, s, t, \lambda]\,, \quad \mathcal{E}_{v_0}[u, s, t, \lambda] = \mathcal{L}[u, v_0, s, t, \lambda]\,, \tag{2.3}$$

which again may be horizontal or vertical. The EPIs and EPI volumes can for example be used to estimate the disparity from the light field as elaborated shortly.

Finally, in the discrete case, one may equivalently use a tensor-based notation and identify

$$\mathcal{L}[u, v, s, t, \lambda] = \boldsymbol{\mathcal{L}}_{uvst\lambda}\,, \quad \boldsymbol{\mathcal{L}} \in \mathbb{R}^{U \times V \times S \times T \times \Lambda}\,, \tag{2.4}$$

where (U, V) corresponds to the angular, (S, T) to the spatial, and Λ to the spectral resolution of the light field.

As an example, a synthetic RGB light field, rendered with a resolution of $(15, 15, 128, 128, 3)$, and two example EPIs, one horizontal and one vertical, are depicted in Figure 2.4. Here, for clarity, only the central and most-peripheral subaperture views of the light field are shown while the EPIs are obtained as sections from the full-resolution light field.

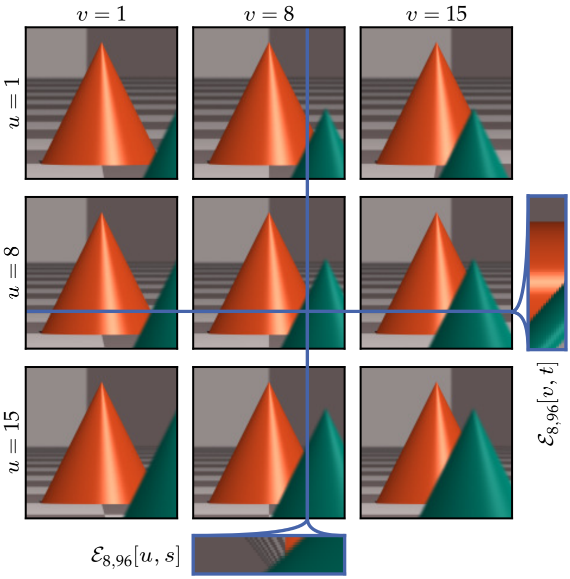


Figure 2.4 Selected subaperture views of an example light field together with one horizontal and one vertical EPI.

2.2 Light field applications

Compared to conventional images, light fields contain much more information of a captured scene. Besides the conventional spatial information, light field cameras also partially capture the scene geometry which allows for numerous applications such as post-capture refocusing [154], superresolution [19], segmentation [214], and saliency detection [114]. Moreover, one can extract reflectance properties such as the specular and diffuse components of a scene [4, 188], or use light fields for robust monocular visual odometry [46]. Recently, light fields have also been investigated in the context of monocular deflectometry [197].

As one of its core applications, light fields allow for a robust disparity estimation, which can be converted to the corresponding depth using a calibrated camera model as elaborated in Appendix A. To this end, a vast amount of work has been published [98], ranging from well-established computer vision methods based on the depth-from-defocus principle [189], feature-based matching [78, 80], using the focal stack [117, 185], or by local slope estimation in the EPIs [212, 213], to deep learning approaches, that have become the state of the art in recent years [176]. Intuitively, due to the dense angular sampling, diffuse features of a scene will be imaged onto constant-valued 2D planes in the 4D light field. For example, this results in the well-known line structures that are observed in the EPIs as previously shown in Figure 2.4. Using the EPIs, disparity estimation is basically equivalent to local line fitting [22] which is usually more robust than feature-based methods that are for example well-established in stereo imaging. As is the case for other passive depth imagers, such as stereo cameras, the scene needs to be sufficiently textured in order to allow for a dense disparity estimate. Estimating the disparity from non-textured diffuse objects via passive imaging techniques is intrinsically ill-defined and may only be achieved using higher-level contextual information (as does the human visual system). Hence, recent advances in deep learning show a great potential to enhance disparity estimation from light fields in challenging situations, *e.g.* for sparsely textured, specular, or reflective objects. For more details on light field imaging and applications, the reader is referred to the literature, *e.g.* the comprehensive reviews by Wu *et al.* [219] or Ihrke *et al.* [92].

Despite the versatile applications and possibilities in post-processing that light fields offer, they also show a strong redundancy [112]. This is particularly true for hand-held plenoptic cameras due to the inherently small baseline (and thus a strong similarity between adjacent subaperture views), which is usually in the range of millimeters, depending on the size of the main lens, the microlens radii, and the pixel pitch. The redundancy becomes even more severe in the case of spectral light fields in which the community has shown an increased interest [87, 221, 222, 236]. In some instances, spectral light fields outperform conventional RGB light fields, *e.g.* in depth estimation in specular regions [235] or profilometry [59]. Furthermore, spectral light fields offer new possibilities, combining methods from conventional light field imaging (*e.g.* disparity estimation) with those from spectral imaging (*e.g.* material classification). Therefore, in order to make use of the redundancy of light fields, one may consider more efficient measurement techniques by coding the light field, opening new possibilities in spectral light field imaging.

2.3 Coded light fields

There are many different possibilities to code light fields: in the angular, the spatial, or the combined spatio-angular domain [11]. In the realm of compressed sensing (*cf*. Section 3.1), the light field is typically coded in the spatio-angular domain by placing an attenuation mask in the light ray's pathway [138, 200, 206]. However, these methods usually do not explicitly account for the color or spectral domain of the light field and perform the reconstruction channel-wise, employing an additional Bayer mask in front of the sensor, and require demosaicing of the raw sensor image. This is for example the case in the fundamental works of Merwah *et al*. [138] but also in recent state-of-the-art deep learning-based frameworks [200]. A generalization to spectral light fields in these instances is generally not straightforward.

Analogously, for color or spectral coding of light fields, two approaches have been proposed: again, either coding the angular or the spatial component. While coding the angular component can naturally be achieved for camera arrays by placing a spectral filter in front of each individual camera [236], it is challenging for MLA-based cameras. Several studies have placed a spectral mask in the main lens plane [87, 139], however, alignment of the mask with the camera sensor is virtually impossible to achieve: Each spectral mask segment has to be imaged onto exactly one pixel since the pixels underneath each microlens code the angular component. Hence, when inevitably misaligned, the resulting coding is in fact not purely angular. Furthermore, the misalignment of the MLA and the sensor cannot be calibrated in a standard fashion, *i.e*. by raw sensor image interpolation [45] (*cf*. Section 4.2.3), since each microlens image is only sparsely sampled. On the other hand, spatial coding of the light field using a camera array can be achieved by placing the same spectral mask in front of each camera sensor, which however has not been considered in the literature. For MLA-based cameras in the unfocused design, the same can be accomplished by coding the MLA [226], resulting in a spatio-spectral coding which is elaborated shortly.

For completeness, in a sense, the conventional Bayer sensor of an MLA-based light field camera also captures a coded light field by applying a spatio-angular color mask at the sensor plane. However, the coding is not well adapted to the light field sampling in this case. The sensor image is usually first demosaiced and then decoded to an RGB light field. While demosaicing methods specific to MLA-based light field cameras exist [47, 230], the geometric properties of the full light field are not specifically taken into account. Furthermore, in the case of light field cameras, the Bayer approach cannot well be generalized to the multispectral case. While this is feasible in the case of conventional cameras [65], the number of measurements per microlens would simply be too sparse for standard sensor-based demosaicing. Hence, a more adapted spectral coding is needed. It is the author's opinion that, for hand-held MLA-based light field cameras, only the spectral coding of the MLA is truly practical as the coding naturally aligns with the discrete sampling of the light field by the MLA. The misalignment of the MLA and the sensor can be calibrated in complete analogy to the conventional RGB case since every microlens image is fully sampled and can be aligned with the sensor grid via standard procedures.

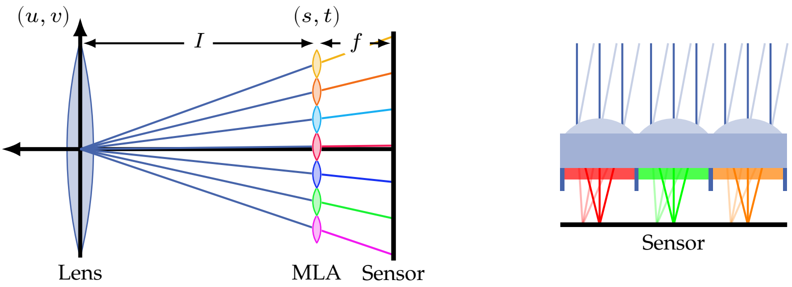
2.3.1 Spatio-spectrally coded light fields

Whereas the other coding schemes have been quite thoroughly discussed in the literature, spectral coding of the MLA has attracted only little attention. While spectral coding of the MLA was investigated in the context of multispectral imaging [178], in the case of light fields it was only briefly discussed in a paper by Ye and Imai [226], employing several restrictive constraints. While this might have been due to a challenging hardware realization, manufacturing of spectrally coded MLAs has become feasible using modern techniques such as inkjet printing of micro-optics [3, 43]. These processes can be generalized to spectral coding, for example using dyed inks for the individual microlenses or using printed dielectric stacks as interference filters.

The camera model of a light field camera in the unfocused design with a spectrally coded MLA is shown in Figure 2.5. By spectrally coding the individual microlenses, such that each microlens acts as a spectral bandpass filter, one obtains the spatio-spectrally coded light field

$$\mathcal{L}^{*}[u, v, s, t, \lambda] = \mathcal{M}[s, t, \lambda] \cdot \mathcal{L}[u, v, s, t, \lambda] \,. \tag{2.5}$$

Here, $\boldsymbol{\mathcal{M}} \in \{0, 1\}^{S \times T \times \Lambda}$ denotes the binary coding mask. The discrete spectral index λ denotes the index of the corresponding spectral filter.



(a) Side view.

(b) Detailed view of the MLA.

Figure 2.5 Model of the unfocused light field camera with a spectrally coded MLA.

Since only one filter is used in the imaging process at every spatial coordinate (s, t), the coding mask $\boldsymbol{\mathcal{M}}$ fulfills the summation constraint

$$\sum_{\lambda=1}^{\Lambda} \mathcal{M}[s, t, \lambda] = 1 \,. \tag{2.6}$$

That is, the coding mask is one-hot encoded in the spectral dimension. During the measurement, the coded light field is projected along the spectral dimension, obtaining

$$\mathcal{L}^*_{\mathrm{p}}[u, v, s, t] = \sum_{\lambda=1}^{\Lambda} \mathcal{L}^*[u, v, s, t, \lambda] \,. \tag{2.7}$$

When the coding mask $\boldsymbol{\mathcal{M}}$ is known, which can be achieved during calibration, the coded light field $\boldsymbol{\mathcal{L}}^*$ can easily be obtained from its projection $\boldsymbol{\mathcal{L}}^*_{\mathrm{p}}$ since for every pixel only one spectral channel has a non-zero value in $\boldsymbol{\mathcal{L}}^*$. Therefore, $\boldsymbol{\mathcal{L}}^*$ and $\boldsymbol{\mathcal{L}}^*_{\mathrm{p}}$ are considered equivalent in the following.

Here, it is assumed that the spectral filters are independent of the ray's incident angle and that the light fields are coded purely in the spatio-spectral domain. This is an approximation that may not hold in practice. Using absorption filters, *i.e.* by using colored inks, the light intensity is exponentially attenuated where the exponent is proportional to the thickness of the absorbing medium. This is known as the Beer-Lambert law. Therefore, rays with a larger incident angle are more strongly attenuated than rays with a perpendicular incident angle as they travel a longer distance in the absorbing medium. However, this only affects the overall attenuation and not the spectral characteristics of the filters. Hence, this effect can be compensated via whitebalancing and devignetting (*cf.* Section 4.2.2). This is not the case when using interference filters, *e.g.* thin-film filters. Here, the layer thickness specifies the central wavelength of the bandpass filter. The effective central wavelength of the filter is larger for rays with a larger incident angle as compared to perpendicular rays. (A similar argument can be made also in the case of multi-layer interference filters such as dielectric stack filters.) This effect cannot directly be incorporated in the discrete signal model. In principle, the model could be extended to include all central wavelengths of the filters associated with the discrete angular coordinates. However, this would drastically increase the dimensionality of the problem, requiring a different formulation of the reconstruction than those investigated here. In practice, the severity of this effect can be controlled by restricting the range of the incident angles. Using a narrow field of view, *i.e.* by using a large main lens focal length, the largest incident angle is decreased and the problem is mitigated. Therefore, the effect is neglected in the following and it is assumed that the filter characteristics are angle-independent.

With this coding scheme, every spectral subaperture $\mathcal{I}_{uv}$ is spectrally coded *using the same* coding mask $\mathcal{M}$. Intuitively, the spectral information of an object is hence spread out across the different subapertures of the coded light field: Given a Lambertian object with non-zero disparity, the object is imaged onto different pixels in each subaperture. Since the different pixels are differently coded, one actually obtains up to $U \times V$ sparse measurements of the object's spectrum, depending on the object's disparity and the coding mask. In a sense, one "simply" has to find this pixel-wise correspondence and join the individual measurements to obtain the original spectrum of the object. While this is trivial when the disparity is known or even constant, it is challenging in the general case.

2.4 Light field data

Throughout this thesis, light fields with different resolutions and properties are investigated: On the one hand a synthetic spectral light field dataset with available ground truth (GT) disparity (*cf.* Section 4.1), and

Table 2.1 Overview of the created light field datasets and their properties. The denoted sizes correspond to the size of a single light field at 32 bit resolution.

Name	Type	Resolution	GT disparity	Size	#
Train	Synthetic	$(9, 9, 32, 32, 13)$	Yes	4.3 MB	78400
Validation	Synthetic	$(9, 9, 32, 32, 13)$	Yes	4.3 MB	9800
Test	Synthetic	$(9, 9, 32, 32, 13)$	Yes	4.3 MB	9800
Challenge	Synthetic	$(9, 9, 512, 512, 13)$	Yes	1.1 GB	6 + 13
Evaluation	Real-world	$(9, 9, 400, 400, 13)$	No	674 MB	6

on the other hand a real-world dataset captured with a custom-built spectral light field camera (*cf.* Section 4.2). These datasets are introduced in more detail in the corresponding sections, however the resolution and properties are presented here as many difficulties arising in the context of spectral light field reconstruction are related to their comparably large dimensionality. Hence, having an overview of the used resolutions is useful when discussing the reconstruction methods in the following.

In all cases, the full (un-coded) spectral light fields are available for a quantitative evaluation of the reconstruction. For the spectral domain, all light fields are sampled in the visible range from 400 to 700 nm in steps of 25 nm resulting in 13 spectral bands. This sampling was chosen as off-the-shelf optical bandpass filters with these properties are available. Extending the sampled range is in this case not trivial due to the limited quantum efficiency of the used camera sensor. In the angular domain, a standard resolution of $(9, 9)$ is chosen as it is conventionally used in the light field community and corresponds to the angular resolution of the camera prototype that is available with sufficient quality. For evaluation, also lower resolutions are considered in an ablation study. In the spatial domain, different resolutions are used depending on the dataset type. For training, validation, and testing of the investigated reconstruction methods, synthetic light fields with a comparably small spatial resolution of $(32, 32)$ are used. To also allow for a quantitative full-sized evaluation, hand-crafted full-sized synthetic spectral light fields are used (which are refered to as *challenges*) with a spatial resolution of $(512, 512)$. Finally, the real-world light fields captured with the spectral light field camera have a spatial resolution of $(400, 400)$. An overview of the different datasets and their corresponding properties is given in Table 2.1.

3 Reconstruction from Coded Light Fields

3.1 Compressed sensing-based reconstruction

Traditionally, the full light field $\boldsymbol{\mathcal{L}}$ is recovered from the coded measurement $\boldsymbol{\mathcal{L}}^*$. For example, the Bayer-coded sensor image is demosaiced before the light field is extracted from it. From the fully recovered light field $\boldsymbol{\mathcal{L}}$, the desired information, such as the disparity, is subsequently estimated. When high compression ratios are involved, which is the case for coded light field cameras and in particular when considering spectral light fields, assuming that the measurement can be formulated via a linear operator, the reconstruction can be done in the compressed sensing framework [53, 58] by solving an optimization problem of the form

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{subject to} \quad \|\mathbf{l}_\mathrm{p}^* - \mathbf{M}\boldsymbol{\Psi}\mathbf{x}\|_2 < \epsilon \,. \tag{3.1}$$

Here, $\boldsymbol{\Psi}\mathbf{x} = \mathbf{l} \in \mathbb{R}^{UVST\Lambda}$ and $\mathbf{l}_\mathrm{p}^* \in \mathbb{R}^{UVST}$ denote the vectorized versions of $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{L}}_\mathrm{p}^*$, respectively, $\boldsymbol{\Psi}$ denotes a basis (or dictionary) in which $\mathbf{l}$ has the representation $\mathbf{x}$, and $\mathbf{M}$ corresponds to the coding and linear measurement. The product $\mathbf{M}\boldsymbol{\Psi}$ is also referred to as the *sensing matrix*. In the following, let $N = UVST\Lambda$ and $M = UVST$ denote the dimension of the vectorized light field and compressed measurement, respectively.

The core idea of compressed sensing, and its most significant distinction to conventional sensing and reconstruction in accordance with the Nyquist-Shannon sampling theorem, is that one is not interested in *any* recovered signal that is consistent with the observed compressed measurement (for which there would be many solutions as the system is underdetermined) but those signals that are *sparse* with respect to the basis $\boldsymbol{\Psi}$. Therefore, the l_0-"norm" $\|\mathbf{x}\|_0$, *i.e.* the number of non-zero elements of $\mathbf{x}$, is minimized while maintaining consistency with the observed measurement $\mathbf{l}_\mathrm{p}^*$. Note that, strictly speaking, the l_0-"norm" is not a norm as it is not homogenous, *i.e.* $\|\alpha\mathbf{x}\|_0 \neq |\alpha| \cdot \|\mathbf{x}\|_0$. However, the quotation marks are neglected in the following for simplicity.

The motivation behind this approach is that natural signals, such as audio recordings or images, have been shown empirically to be sparse in some domain—an observation that has found countless applications in signal and image processing, *e.g.* for compression, denoising, or inpainting [132]. One of the main challenges of compressed sensing, besides solving the optimization problem (3.1), is to find a suitable basis under which the signals of interest can be sparsely represented, and to specify which linear measurements guarantee a signal recovery. To this end, in the past 15 years a rigorous mathematical framework has been developed.

Assuming that the vectorized signal is k-sparse in the basis $\mathbf{\Psi}$, *i.e.*

$$\|\mathbf{x}\|_0 \leq k \tag{3.2}$$

for some $k < N$, one can show that the signal can be exactly recovered from $\mathcal{O}(k \lg N)$ measurements if the sensing matrix fulfills the so-called restricted isometry property (RIP) [30]. The RIP is not re-stated here as it is in fact NP-hard to compute and therefore quasi-impossible to verify in practice. Instead, the mutual coherence

$$\mu(\mathbf{A}) = \max_{1 \leq i,j \leq N} \left|\langle \mathbf{a}_i, \mathbf{a}_j \rangle\right| \leq 1 \tag{3.3}$$

can be used to estimate how well the sensing matrix $\mathbf{A} = \mathbf{M}\mathbf{\Psi}$ fulfills the RIP. Here, $\mathbf{a}_i$ and $\mathbf{a}_j$ denote the i-th and j-th column of $\mathbf{A}$ which are assumed to be normalized. The recovery of the original signal is then guaranteed *with high probability* from $\mathcal{O}(\mu(\mathbf{A})^2 k \lg N)$ measurements [30]. Hence, when designing the coding matrix $\mathbf{M}$, a low coherence of the sensing matrix $\mathbf{A}$ is desired. The mutual coherence of different basis matrices using the considered coding scheme is investigated in Section 3.1.1.

There are several approaches to solve the general compressed sensing reconstruction problem (3.1). In general, the minimization of the non-convex l_0-norm is NP-hard [58]. In the past decades, many different optimization techniques have been proposed to overcome this difficulty. On the one hand, greedy methods, such as matching pursuit (MP) and its variants [133, 152, 153, 159], directly tackle the l_0-norm minimization for k-sparse signals. However, these methods, with the notable exemption of the sparsity-adaptive MP [52], require that one specifies the sparsity value k which is unknown in practice. Due to their greedy nature, MP-based methods, while often faster than alternatives, usually converge to a worse minimum as compared to the following.

If the RIP of the sensing matrix is fulfilled, the optimization problem (3.1) is equivalent to the constrained convex l_1-minimization [58]

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{l}_\mathrm{p}^* - \mathbf{M}\mathbf{\Psi}\mathbf{x}\|_2 < \epsilon \,. \tag{3.4}$$

Furthermore, this constrained problem is equivalent to the unconstrained basis pursuit denoising [38]

$$\min_{\mathbf{x}} \|\mathbf{l}_\mathrm{p}^* - \mathbf{M}\mathbf{\Psi}\mathbf{x}\|_2^2 + \eta\|\mathbf{x}\|_1 \tag{3.5}$$

for some Lagrange multiplier $\eta \in \mathbb{R}$. In statistics, this is known as the *least absolute shrinkage and selection operator* (LASSO) [191]. This so-called *convex relaxation* opens the problem to well-scaling optimization techniques.

In the considered case regarding spectral light fields, one faces several challenges employing the compressed sensing framework. First, the considered signals are comparably high-dimensional. For example, even in the case of a small light field patch of shape $(9, 9, 32, 32, 13)$, which corresponds to the size of the light fields in the used test dataset, the vectorized dimension is already $N \approx 10^6$ and the basis $\mathbf{\Psi}$ alone requires more than 4.6 TB of memory at 32 bit resolution. This is unfeasible even for large-scale computers, not even considering the case of full-sized light fields. The same issue is generally also true for the coding matrix $\mathbf{M}$. However, in the considered case, due to the one-hot encoding in the spectral dimension, the coding mask takes a simple diagonal form

$$\mathbf{M} = \mathrm{diag}(\mathbf{m}) \ , \quad \mathbf{m} \in \{0, 1\}^N \ , \tag{3.6}$$

when *not* performing the projection along the spectral dimension. While this increases the memory requirements of the coded light field (because it is not spectrally projected), the memory requirements of the coding mask is reduced drastically. Furthermore, the coding can now be expressed as a simple element-wise multiplication instead of a matrix multiplication reducing the computational effort:

$$\mathbf{M}\mathbf{\Psi}\mathbf{x} = \mathbf{m} \odot \mathbf{\Psi}\mathbf{x} \,. \tag{3.7}$$

Therefore, the optimization problem (3.5) slightly simplifies to

$$\min_{\mathbf{x}} \|\mathbf{l}^* - \mathbf{m} \odot \mathbf{\Psi x}\|_2^2 + \eta \|\mathbf{x}\|_1 \, , \tag{3.8}$$

where $\mathbf{l}^* \in \mathbb{R}^{UVST\Lambda}$ denotes the vectorized version of the coded (but not spectrally projected) light field $\boldsymbol{\mathcal{L}}^*$. This can trivially be re-written as

$$\min_{\mathbf{x}} \, f(\mathbf{x}) + \eta \|\mathbf{x}\|_1 \, , \tag{3.9}$$

where

$$f(\mathbf{x}) = \|\mathbf{l}^* - \mathbf{M\Psi x}\|_2^2 = \|\mathbf{l}^* - \mathbf{m} \odot \mathbf{\Psi x}\|_2^2 \tag{3.10}$$

is convex and differentiable. Its derivative can be explicitly calculated as

$$\begin{aligned} \nabla f(\mathbf{x}) &= 2(\mathbf{\Psi}^{\mathrm{T}}\mathbf{M}^{\mathrm{T}}\mathbf{M\Psi x} - \mathbf{\Psi}^{\mathrm{T}}\mathbf{M l}^* \,) \\ &= 2(\mathbf{\Psi}^{\mathrm{T}}\mathbf{m} \odot \mathbf{\Psi x} - \mathbf{\Psi}^{\mathrm{T}}\mathbf{m} \odot \mathbf{l}^* \,) \, . \end{aligned} \tag{3.11}$$

Here, it was used that $\mathbf{M} = \mathrm{diag}(\mathbf{m})$ is an orthogonal projection, *i.e.* $\mathbf{M}^{\mathrm{T}} = \mathbf{M}$ and $\mathbf{M}^2 = \mathbf{M}$. This makes the problem suitable for the L-BFGS optimization [26] in the OWL-QN variant [7] which was developed specifically to scale well to high-dimensional problems. Moreover, using a non-matrix-based implementation of the basis transform $\mathbf{\Psi}$, saving of the basis matrix can be avoided and the approach becomes feasible even for full-sized spectral light field reconstruction.

Second, in the considered case, one is quite severely restricted in the design of the coding matrix M due to the constraints of the hardware realization. Specifically, the design is constrained by the physical coding model (2.5) and the summation constraint (2.6). However, it can be shown that random matrices, such as random Bernoulli oder Gaussian matrices, are largely incoherent to any orthonormal basis [58]. For this reason, in the compressed sensing context, random binary matrices are considered in the following. That is, for every spatial coordinate (s, t) a random wavelength channel index is drawn independently from the discrete uniform distribution on the set $\{1, 2, \ldots, \Lambda\}$,

$$\mathrm{z}_{s,t} \sim \mathcal{U}\{1, \Lambda\} \; , \tag{3.12}$$

which specifies the spectral coordinate of the one-hot encoding,

$$\mathcal{M}[s, t, \lambda] = \begin{cases} 1 & \text{if } \lambda = \mathrm{z}_{s,t} \, , \\ 0 & \text{otherwise.} \end{cases} \tag{3.13}$$

To obtain the vectorized $\mathbf{m}$, the full mask $\boldsymbol{\mathcal{M}}$ is subsequently vectorized and tiled UV-times. While this does not directly correspond to a random Bernoulli matrix, since the individual matrix entries are not independent along the spectral dimension due to the summation constraint, it is the closest mask that can actually be achieved with real hardware and the considered camera model. Also, it is not the main focus of this thesis to optimize the coding scheme in the context of compressed sensing. Different coding schemes however will be discussed in the context of the proposed principal reconstruction in Section 3.3.

And finally, third, it is in principle not known which bases are suitable to sparsely represent spectral light fields. To this end, the following two approaches are considered.

3.1.1 Fixed basis-based reconstruction

In the signal processing community, many methods have been discussed to sparsely represent natural (discrete) signals. Most prominently, the discrete Fourier transform and its real-valued analogue—the discrete cosine transform (DCT)—use global periodic functions as basis functions. The consequence of this is a fixed spatial frequency resolution. Using basis functions with a *compact support*, the so-called wavelets, the frequency decomposition structure can be altered or even made adaptive to a given signal. The resulting discrete wavelet transform (DWT) and its generalization—the wavelet packet transform (WPT)—have been applied to many problems in image processing and shown to be well suited to sparsely represent natural images [132]. Depending on the used decomposition structure, adaptive frequency resolutions are obtained. For example, the standard (multi-level) 2D-DWT has a high spatial resolution but low frequency resolution at high frequencies (the detail coefficients) and a low spatial resolution but high frequency resolution at low frequencies (the approximation coefficients). However, the wavelet transform is not anisotropic which is problematic at image edges and other curve-like discontinuities. This motivated further generalizations such as curvlets [29], ridgelets [28], and shearlets [57]. In principle, these anisotropic transforms would be well suited for sparse light field representations: Due to the epipolar geometry, diffuse points of a scene will be mapped to constant-valued 2D planes in the 4D light field. For example, this results in the well known line-like structure of the EPIs showing a 1D section of these 2D planes whose slope depends on the disparity of the scene object as previously discussed in Section 2.2. Hence, light fields show a strong correlation along these planes, and discontinuities along directions normal to them.

Nevertheless, only little work has been published regarding sparse light field representation using fixed bases. Recent works by Vagharshakyan *et al.* use the shearlet transform to sparsely code each individual EPI of the light field to synthesize intermediate views [201, 202] or use it for compressive light field coding [2]. However, it is not clear how this approach can be used in the context of light field compressed sensing. Similarly, a disparity-aware generalization of the DWT has been proposed [35, 69], however, the considered light fields have a much higher angular resolution than the ones considered in this thesis. In the considered case, one faces yet another challenge when transforming light fields into a sparse basis, namely the strong asymmetry between the spatial and the angular as well as the spectral resolution. That is, the angular and spectral resolution are comparably low. As previously noted, throughout this thesis the light fields have an angular resolution of $(9, 9)$ and a spectral resolution of (13) while the spatial resolution varies from $(32, 32)$ to a full-sized resolution of $(512, 512)$. These small resolutions make it quite difficult to employ more complex basis functions and hence limit the potential of the DWT and its anisotropic generalizations. For example, the well-known db2 Daubechies wavelet already has a filter sample length of four.

To investigate this in more detail, Figure 3.1 shows the mean mutual coherence of 100 sensing matrix realizations where the coding matrix is sampled according to (3.12) and (3.13). Here, the 5D-DCT, the 5D-DWT using two different mother wavelets, and a combined approach using the 3D-DCT in the angular and spectral and the 2D-DWT in the spatial domain are considered. For the (flattened) 5D transforms, only separable transforms are considered, *i.e.* the basis matrices are calculated as Kronecker products from the corresponding 1D transforms,

$$\mathbf{\Psi} = \mathbf{\Psi}_u \otimes \mathbf{\Psi}_v \otimes \mathbf{\Psi}_s \otimes \mathbf{\Psi}_t \otimes \mathbf{\Psi}_\lambda \,. \tag{3.14}$$

Note that in particular, the resulting 5D-DWT and 2D-DWT correspond to the so-called fully-separable DWT which has a slightly different fre-

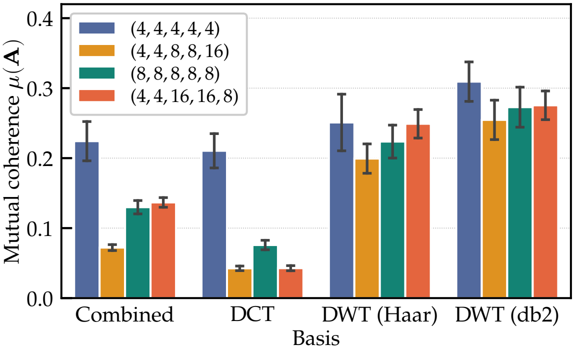


Figure 3.1 Mean mutual coherence of 100 measurement matrix realizations $\mathbf{A} = \mathbf{M}\boldsymbol{\Psi}$ using different samples of the coding matrices $\mathbf{M}$, basis matrices $\boldsymbol{\Psi}$, and light field sizes. The shown error bars indicate the 1σ interval.

quency decomposition than the one typically used in the 2D case. As previously argued, the 5D-DCT performs much better than the 5D-DWT for both the Haar as well as the db2 wavelet, in particular for higher resolutions. As expected, the db2 wavelet performs worse than the Haar wavelet due to the larger filter length and the small considered light field resolutions. The combination of a spatial 2D-DWT with the DCT does improve the performance compared to the plain DWT but it fails to outperform the 5D-DCT at the investigated sizes. Note that it was not possible to measure the mutual coherence for larger light field sizes due to the immense memory requirements of the basis matrices. Concluding, for the considered light field sizes and binary coding, the 5D-DCT results in the lowest coherence of the measurement matrix.

This observation is in accordance with a precursory study that was jointly conducted with Matthias Bächle [A5]. Here, it was found that the DWT, or even more generally the signal-adapted WPT, performs just on-par or worse than the DCT at high compression ratios in the case of compression and denoising of 4D (monochromatic) light fields. Similar observations have also been made by Marwah *et al.* [138].

Finally, these more elaborate transforms also come with a much higher computational complexity resulting in much slower calculation as compared to the DCT. This would drastically slow down the reconstruction which is already in the order of hours for the full-sized light fields.

Lower-dimensional DCTs, such as a subaperture-wise 3D-DCT, were considered in a precursory study [B6] but were found to be inferior to the full 5D-DCT. Concluding, for these reasons, only the 5D-DCT is considered in the fixed-basis approach.

Using the 5D-DCT as a basis for the spectral light field reconstruction and a functional (non-matrix-based) implementation of the DCT $\mathbf{\Psi}$ and its inverse $\mathbf{\Psi}^{\mathrm{T}}$, explicitly saving the matrix $\mathbf{\Psi}$ can be avoided and the approach becomes feasible even for full-sized light field reconstruction. With these modifications, in the considered case, the reconstruction of full-sized light fields with shape $(9, 9, 512, 512, 13)$ requires about 96 GB of RAM which is well within the capacity of high-end working stations or mid-range cluster computing nodes.

However, this DCT-based approach does not explicitly consider the epipolar geometry of the light field. As it will be shown in the evaluation, this results in decreased performance in the subsequent disparity estimation which of course is strongly dependent on the geometry. To make use of the light field geometry for a sparse representation, the following dictionary learning approaches are considered.

3.1.2 Dictionary-based reconstruction

To overcome the difficulty of explicitly choosing an appropriate basis, one can also learn a sparse representation of the spectral light fields using a suitable training dataset. This is referred to as *dictionary learning* [193]. Here, the goal is to find a dictionary $\mathbf{D} \in \mathbb{R}^{N \times kN}$, where again N is the vectorized light field's dimension and $k > 1$ is the so-called dictionary overcompleteness, such that the (possibly approximate) light field representation $\mathbf{x} \in \mathbb{R}^{kN}$, given by

$$\mathbf{l} = \mathbf{D}\mathbf{x}\,, \tag{3.15}$$

is sparse. The columns $\mathbf{d}_i, i = 1, \dots, kN$, of $\mathbf{D}$ are called the *atoms* of the dictionary. Obviously, obtaining $\mathbf{x}$ from a given light field $\mathbf{l}$ is an inverse problem (and in fact very similar to the reconstruction problem (3.1)) and may only be solved approximately. To learn a dictionary $\mathbf{D}$ from a training dataset $\mathbf{L} \in \mathbb{R}^{N \times L}$ containing L vectorized spectral light fields, the joint optimization problem

$$\min_{\mathbf{D},\mathbf{X}} \|\mathbf{L} - \mathbf{D}\mathbf{X}\|_{2,2}^2 + \eta \|\mathbf{X}\|_{1,1} \quad \text{subject to} \quad \mathbf{1} \odot \mathbf{D}^{\mathrm{T}}\mathbf{D} = \mathbf{1}\,, \tag{3.16}$$

has to be solved [131]. Here, $\mathbf{X} \in \mathbb{R}^{kN \times L}$ denotes the sparse representation of all light fields in the training dataset and $\|\cdot\|_{p,p}$ denotes the (p, p)-matrix norm, *i.e.* $\|\cdot\|_{2,2}$ corresponds to the Frobenius norm and $\|\cdot\|_{1,1}$ is equal to the sum of the absolute matrix elements. The constraint on $\mathbf{D}$ ensures that all atoms are normalized. The optimization is performed in an alternating manner, separating the non-convex problem (3.16) into two convex sub-problems for $\mathbf{D}$ and $\mathbf{X}$, respectively [109, 131]. Hence, for each optimization iteration step, first a sparse representation $\mathbf{X}$ of the light field batch is estimated using a *fixed* dictionary $\mathbf{D}$, which is initialized using a truncated normal distribution and succeeding atom normalization. Again, there are several approaches to estimate the sparse decomposition. Here, due to its fast convergence, the *fast iterative shrinkage/thresholding algorithm* (FISTA) [17] is used to solve

$$\min_{\mathbf{X}} \|\mathbf{L} - \mathbf{D}\mathbf{X}\|_{2,2}^2 + \eta \|\mathbf{X}\|_{1,1}\,. \tag{3.17}$$

In the second step, the sparse representation $\mathbf{X}$ is fixed, and the dictionary optimization

$$\min_{\mathbf{D}} \|\mathbf{L} - \mathbf{D}\mathbf{X}\|_{2,2}^2 \quad \text{subject to} \quad \mathbf{1} \odot \mathbf{D}^{\mathrm{T}}\mathbf{D} = \mathbf{1}\,, \tag{3.18}$$

is performed updating the dictionary atoms using gradient descent and succeeding atom normalization.

Over a fixed-basis approach, the dictionary has the advantage of implicitly taking the light field geometry and redundancy into account. However, one still faces the problem of large dimensionality. Due to the dictionary overcompleteness, the problem is even slightly more severe than for the basis decomposition.

In the context of light fields, some dictionary learning approaches have been investigated in the literature, however, most do not explicitly consider the color or spectral domain, performing the sparse coding channel-wise. For example, the dimensionality issue can be alleviated by using a disparity-aware dictionary to warp the central view [37]. Similarly, Johannsen *et al.* [97] learn a disparity-aware dictionary by learning atoms for the central view which are subsequently lifted to the full 4D light field using a predefined constant disparity. In this way, each atom is associated with a known disparity and disparity estimation can be performed by analyzing the coefficients of a light field in the dictionary representation. However, these approaches are problematic in the case of occluding scenes or patches. More closely related to the above problem statement, Marwah *et al.* [138] discuss a general dictionary learning approach for compressive light field imaging. In order to solve the problem of large dimensionality, the learned atoms are chosen to be of a smaller size, performing the full light field representation patch-wise. This approach is adopted here.

That is, the dictionary is learned with atoms of shape $(5, 5, 8, 8, 13)$. To encode an input light field using the dictionary, the light field is first patched into the corresponding shape, using a spatial overlap of $(4, 4)$ and an angular overlap of $(1, 1)$. To avoid edge defects, overlapping patches are averaged when de-patching. Patching in the spectral domain is not performed as the characteristics of the different spectral bands are assumed to be unique. Here, the shape of the light field atoms was chosen as large as possible while still resulting in a manageable dictionary size. Using a dictionary overcompleteness of $k = 2$, which in previous works has been argued to be suitable for light field dictionaries [138], the dictionary is roughly 3.5 GB in size. With the additional memory requirements of the gradient backpropagation and light field patching, this was just small enough to perform the dictionary learning on a 32 GB Nvidia Tesla V100 GPU.

Further, since the used training dataset contains more than 200 GB of light field data (*cf.* Section 4.1), it is obvious that (3.16) cannot be optimized at once. To overcome this, coresets have been discussed in the literature [60], which can be interpreted as a form of importance sampling. That is, using a specifically designed cost function, a suitable subset of the training dataset is chosen to perform the optimization task. However, due to the advances in computer hardware (in particular GPU memory) and the success of stochastic gradient descent (SGD) in machine learning, here SGD is used to solve (3.16) (or rather the alternating optimization of the two convex subproblems). This approach has been proposed already in 2009 by Mairal *et al*. [131]. Note that the atom shape $(5, 5, 8, 8, 13)$, while seeming small in the context of light fields, is actually comparably high-dimensional. For image-based dictionary learning, this corresponds to atoms of shape $(144, 144)$, since $144{\cdot}144 \approx 5{\cdot}5{\cdot}8{\cdot}8{\cdot}13$. However, even methods specifically adapted for high-dimensional image dictionary learning only reach a feasible atom size of $(64, 64)$ [187]. To further increase the individual atom size, the following dictionary decomposition approach is developed.

3.1.2.1 Dictionary tensor decomposition

As previously introduced, the compressed sensing framework is typically formulated in a vectorized fashion. This is useful since a common framework and generic algorithms can be developed regardless of the dimensionality of the specific problem, *e.g*. whether one deals with 1D time signals, 2D images, or 5D spectral light fields. Mathematically, one simply makes use of the fact that two vector spaces with the same (finite) dimension are isomorphic [83], *i.e*. in particular, for any $n, m \in \mathbb{N}$ that $\mathbb{R}^{n \times m} \simeq \mathbb{R}^{nm}$. From a practical standpoint, this corresponds to the fact that a block of computer memory associated with an nm-dimensional array can equivalently be viewed as an (n, m)-shaped tensor, related via a reshape and (possibly) a transpose.

However, in the vectorized dictionary learning approach it is not clear how to explicitly utilize the underlying geometric structure of the signal. To overcome this difficulty, the vectorized dictionary learning problem is "un-flattened" to regain the tensorial light field structure, *i.e*. the light field representation correspondence

$$(\mathbf{l})_i = \sum_j \mathbf{D}_{ij} (\mathbf{x})_j \quad \leftrightharpoons \quad \boldsymbol{\mathcal{L}}_{uvst\lambda} = \sum_j \boldsymbol{\mathcal{D}}_{juvst\lambda} (\mathbf{x})_j \tag{3.19}$$

is used, where now $\boldsymbol{\mathcal{D}} \in \mathbb{R}^{kN \times U \times V \times S \times T \times \Lambda}$ denotes the tensor dictionary (here, the atom axis is put as the first axis for simplicity) and again $\mathbf{l}$ corresponds to the vectorized version of $\boldsymbol{\mathcal{L}}$.

In the context of spectral light fields, Marquez *et al*. [134] introduce a dictionary learning approach based on the Tucker tensor decomposition. However, their approach is tightly coupled to their specific camera design and cannot be applied in the presented case. While some general tensor dictionary learning approaches and algorithms have been discussed in the literature [27, 63, 170], a more straightforward approach is proposed here. The proposed approach is based on simple matrix-tensor correspondences, while keeping the underlying algorithms identical to the vectorized case in order to be able to directly compare the tensor approach with the conventional approach. Via these correspondences, all matrix-vector operations that are used for the vector dictionary learning can now be translated to the corresponding tensor calculations. Most importantly, the already presented correspondence (3.19) of matrix-vector multiplication is used. Furthermore, for the used FISTA sparse decomposition algorithm, the largest eigenvalue of $\mathbf{D}^{\mathrm{T}}\mathbf{D}$ is calculated via the *von Mises iteration* [145] which is also known as the power method. To this end, the following correspondence is needed:

$$(\tilde{\mathbf{y}})_i = \sum_{jk} \mathbf{D}_{ji}\mathbf{D}_{jk}(\mathbf{y})_k \quad \leftrightharpoons \quad (\tilde{\mathbf{y}})_i = \sum_{juvst\lambda} \boldsymbol{\mathcal{D}}_{iuvst\lambda}\boldsymbol{\mathcal{D}}_{juvst\lambda}(\mathbf{y})_j \,. \tag{3.20}$$

With these correspondences, the standard dictionary learning can be directly adapted to use a tensor dictionary while keeping all numerical calculations identical. This now opens the problem to tensor decomposition techniques.

As a more parameter-efficient alternative to the standard vector dictionary learning approach, here it is proposed to factorize the tensor dictionary into three separate dictionaries for the angular, spatial, and spectral domain. This decomposition is similar to the one proposed by Caiafa and Cichocki [27] but does not further separate the 2D angular and spatial domains. That is, the light field is now represented as linear combinations of tensor products of the individual angular, spatial, and spectral atoms,

$$\boldsymbol{\mathcal{L}}_{uvst\lambda} = \sum_{abc} \boldsymbol{\mathcal{A}}_{auv}\boldsymbol{\mathcal{B}}_{bst}\boldsymbol{\mathcal{C}}_{c\lambda}\boldsymbol{\mathcal{X}}_{abc} \,. \tag{3.21}$$

Here, $\boldsymbol{\mathcal{A}}$ with shape (A, U, V), $\boldsymbol{\mathcal{B}}$ with shape (B, S, T), and $\boldsymbol{\mathcal{C}}$ with shape (C, Λ) denote the angular, spatial, and spectral dictionary with sizes $A = kUV$, $B = kST$, and $C = k\Lambda$, respectively. The coefficient tensor $\boldsymbol{\mathcal{X}}$ of shape (A, B, C) is hence of size $k^3 UVST\Lambda$ which is a factor of k^2 larger

Table 3.1 Overview of the used matrix-tensor and tensor factorization correspondences.

Type	Vector	Tensor (full)	Tensor (decomposed)
Light field	$\mathbf{l}, (N)$	$\boldsymbol{\mathcal{L}}, (U, V, S, T, \Lambda)$	$\boldsymbol{\mathcal{L}}, (U, V, S, T, \Lambda)$
Coefficients	$\mathbf{x}, (kN)$	$\mathbf{x}, (kN)$	$\boldsymbol{\mathcal{X}}, (A, B, C)$
Dictionary	$\mathbf{D}, (N, kN)$	$\boldsymbol{\mathcal{D}}, (kN, U, V, S, T, \Lambda)$	$\boldsymbol{\mathcal{A}}, (A, U, V)$ $\boldsymbol{\mathcal{B}}, (B, S, T)$ $\boldsymbol{\mathcal{C}}, (C, \Lambda)$
Decomp.	$\mathbf{l} = \mathbf{D}\mathbf{x}$	$\boldsymbol{\mathcal{L}}_{uvst\lambda} = \sum_{j} \boldsymbol{\mathcal{D}}_{juvst\lambda}(\mathbf{x})_j$	$\boldsymbol{\mathcal{L}}_{uvst\lambda} = \sum_{abc} \boldsymbol{\mathcal{A}}_{auv}\boldsymbol{\mathcal{B}}_{bst}\boldsymbol{\mathcal{C}}_{c\lambda}\boldsymbol{\mathcal{X}}_{abc}$
Eigenval.	$\tilde{\mathbf{y}} = \mathbf{D}^{\mathrm{T}}\mathbf{D}\mathbf{y}$	$(\tilde{\mathbf{y}})_i = \sum_{juvst\lambda} \boldsymbol{\mathcal{D}}_{iuvst\lambda}\boldsymbol{\mathcal{D}}_{juvst\lambda}(\mathbf{y})_j$	$\tilde{\boldsymbol{\mathcal{Y}}}_{abc} = \sum_{ijkuvst\lambda} \boldsymbol{\mathcal{A}}_{auv}\boldsymbol{\mathcal{A}}_{iuv}\boldsymbol{\mathcal{B}}_{bst}\boldsymbol{\mathcal{B}}_{jst}$ $\cdot \boldsymbol{\mathcal{C}}_{c\lambda}\boldsymbol{\mathcal{C}}_{k\lambda}\boldsymbol{\mathcal{Y}}_{ijk}$

than in the vectorized case. However, with this approach, the overall dictionary size is significantly reduced from

$$k(UVST\Lambda)^2 \tag{3.22}$$

to

$$k((UV)^2 + (ST)^2 + (\Lambda)^2)\,, \tag{3.23}$$

which frees up memory to use larger individual atoms as compared to the vectorized case. In the presented case, the atom size is increased from (5, 5, 8, 8, 13) to a (separated) size of (7, 7, 16, 16, 13). To complete the correspondences necessary for the FISTA decomposition,

$$\tilde{\boldsymbol{\mathcal{Y}}}_{abc} = \sum_{ijkuvst\lambda} \boldsymbol{\mathcal{A}}_{auv}\boldsymbol{\mathcal{A}}_{iuv}\,\boldsymbol{\mathcal{B}}_{bst}\boldsymbol{\mathcal{B}}_{jst}\,\boldsymbol{\mathcal{C}}_{c\lambda}\boldsymbol{\mathcal{C}}_{k\lambda}\,\boldsymbol{\mathcal{Y}}_{ijk} \tag{3.24}$$

is used to calculate the eigenvalues via the von Mises iteration. With these analogies, the learning of the three independent dictionaries $\boldsymbol{\mathcal{A}}$, $\boldsymbol{\mathcal{B}}$, and $\boldsymbol{\mathcal{C}}$ is performed in complete analogy to the conventional vector dictionary learning. In the machine learning context, this decomposition approach can also be interpreted as a form of parameter sharing. An overview of the used matrix-vector/tensor correspondences and tensor decomposition is given in Table 3.1.

3.2 Principal reconstruction via multi-task deep learning

Despite the versatile applications and flexibility that light fields offer, in the case of coded light fields, it seems superfluous to reconstruct the high-dimensional spectral light field from the low-dimensional compressed measurement, only to extract again low-dimensional (yet complex) information from it. For example, a spectral light field of shape $(9, 9, 512, 512, 13)$, which is the resolution that is used for the full-sized evaluation in this thesis, requires about 1.1 GB of memory when saved with 32 bit precision while its compressed measurement and spectral central view only require roughly 85 MB and 14 MB, respectively. Furthermore, the performance of subsequent light field applications, such as disparity estimation, may suffer due to the errors introduced by the reconstruction. Therefore, instead of reconstructing an intermediate full light field from the coded measurement, it is proposed to infer the desired properties from the coded light field directly. Here, this is referred to as *principal reconstruction*.

In this thesis, the focus lies on the reconstruction of the multispectral central view and its aligned disparity map but other reconstruction targets are also possible. That is, given the coded measurement $\boldsymbol{\mathcal{L}}^*$, estimating the central view $\mathcal{I}[s, t, \lambda]$ and the corresponding disparity map $\mathcal{D}[s, t]$ without the intermediate recovery of the full light field $\boldsymbol{\mathcal{L}}$. In this instance, the light field camera with a spectrally coded MLA can be interpreted as a monocular single-shot spectral depth camera. The central view and disparity map are chosen as the reconstruction targets because they represent a large amount of the full light field data and epipolar geometry. In fact, for non-occluding Lambertian scenes, they are equivalent to the full light field data. Of course, this may not be suitable for some applications such as specular component estimation or applications including strong occlusion. However, the reconstruction targets can in principle be adapted to those needs. That is, one could equally consider segmentation, saliency, or reflection properties of the light field. A schematic comparison of the conventional and the proposed principal reconstruction, in the case of the used reconstruction targets $\boldsymbol{\mathcal{I}}$ and $\boldsymbol{\mathcal{D}}$, is shown in Figure 3.2.

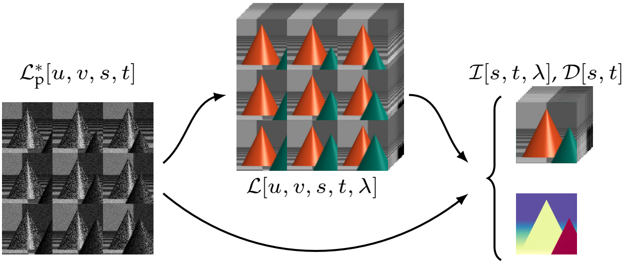

$\mathcal{L}^*_{\mathrm{p}}[u,v,s,t]$
$\mathcal{L}[u,v,s,t,\lambda]$
$\mathcal{I}[s,t,\lambda], \mathcal{D}[s,t]$

Figure 3.2 Schematic comparison of conventional, *e.g.* compressed sensing-based, (top) and principal reconstruction (bottom) from a spatio-spectrally coded light field.

Since artificial neural networks have become the state of the art for many computer vision tasks, the proposed principal reconstruction is performed using a supervised deep learning approach. In the context of light field deep learning, neural networks have significantly outperformed conventional methods, *e.g.* in the case of disparity estimation [176], light field superresolution [227, 228], intrinsics estimation [4], dense-from-sparse light field reconstruction [218], classification [208], and more.

3.2.1 Related work

Three publications resemble the proposed principal reconstruction. To some extent, the original work on coded MLAs by Ye and Imai [226] follows an approach that is similar to the proposed one. The authors reconstruct a super-resolved spectral central view from the coded light field formulated as a compressed sensing reconstruction. However, they heavily constrain the problem to scenes with a constant disparity (which is reasonable in the considered case of remote sensing). Doing so, the light field coding can actually be formulated in a linear fashion using solely the central view. Hence, its reconstruction is suitable for the compressed sensing framework. However, they do not discuss details on the estimation of the disparity from the coded data. While it seems feasible in the case of a constant disparity, it is certainly challenging in the general case due to the sparsity of the coded light field and therefore sparse observation of the epipolar geometry. Furthermore, in the case of a constant disparity, the reconstruction of the spectral central view actually becomes trivial as the individual subapertures can simply be warped onto the central view "filling up" the missing spectral measurements. Due to the constant disparity, the warping corresponds to a simple translation of the subaperture views. This approach was investigated in detail in a precursory study [A12] and is not further discussed here. The use of compressed sensing in this case therefore seems excessive. In this thesis, the constraint is lifted and the general case of scenes with arbitrary disparity is investigated.

Recently, Vadathya *et al.* [200] proposed a general framework for light field reconstruction from coded projections. In fact, similar to the proposed approach, they estimate an intermediate central view and disparity field directly from the coded measurement. However, there are some crucial differences to the work that is presented here: First, the estimated central view and the disparity field are used only intermediately to reconstruct the full light field from the coded measurement. In fact, the central view and disparity field estimation networks cannot be trained without the full light field reconstruction as the network design and (self-supervised) loss function are based on the full light field reconstruction. This directly opposes the proposed approach. However, due to the self-supervised architecture, their approach can be trained using real-world data and does not require synthetic disparity ground truth data, which is useful in practice. Since the full light field is obtained from the estimated disparity field and the central view, their approach is also not suitable to estimate different intermediate reconstruction targets from the coded measurements. While different targets are not explicitly considered in this thesis, the proposed approach can be adapted straightforwardly. Second, the coding schemes considered by Vadathya *et al.* include angular integration and are only valid for attenuation mask-based compressive light field imagers. In particular, this does not include MLA-multiplexed and coded light fields which are considered in this thesis. And most importantly third, their approach, as many compressive light field approaches before, does not consider the color or spectral domain of the light field. An RGB compressed light field is obtained by demosaicing the raw coded measurements of a Bayer pattern sensor, however, this is not explicitly discussed. An extension to multispectral light fields is likely to require significant changes in the network's architecture. The impact of demosaicing in compressive light field imaging remains unclear and has yet to be discussed in the literature. Hence, the framework cannot be applied to the spectrally coded light fields that are considered here.

Moreover, recent works by Baek *et al.* [13] propose a new spectral depth snapshot camera. The optics, a free-form diffractive lens, and reconstruction algorithm are trained in an end-to-end fashion which they refer to as *deep optics*. The considered reconstruction from the "compressed" measurement is almost the same as in the proposed case—namely a spectral image and its aligned depth map (instead of the disparity). In fact, the used reconstruction network is very similar to the one proposed here. The crucial difference between the two approaches is the angular component of the incoming signal (coded by the MLA) which can be considered to explicitly take into account the epipolar geometry. This is not explicitly the case in the work by Baek *et al.* as the whole optical setup is learned in an end-to-end manner. The explicit usage of the epipolar geometry might be beneficial for the quality of the estimated disparity map however a direct comparison of the two approaches is difficult. Furthermore, the approach presented in this thesis is based on a well-understood camera design. In particular, it is therefore possible to use well-established calibration schemes of the used light field camera. Hence, the proposed trained reconstruction network is independent of the used (calibrated) camera. On the other hand, the approach by Baek *et al.* is very general in nature and also considers higher spatial and spectral resolutions than the presented one. Furthermore, since the optics are optimized jointly with the reconstruction, it is possible to directly estimate the depth instead of the disparity values without the need of additional camera calibration. Overall, the two approaches cannot be directly compared in a meaningful quantitative way as they are conceptually quite different.

3.2.2 Network architectures

In recent years, deep learning has been tremendously successful in numerous scientific and engineering disciplines, ranging from computer vision (*e.g.* detection, semantic segmentation, or image synthesis [100, 166, 231]), to natural language processing (*e.g.* neural language modeling, text generation, or machine translation [24, 50, 123]), robotics (*e.g.* control [146]), chemistry (*e.g.* protein unfolding [99]), physics (*e.g.* efficient sampling of statistical ensembles [156]), and mathematics (*e.g.* solving differential equations [126]), to name a few.

As opposed to common image-based computer vision tasks, light field deep learning poses additional challenges. Similar to the case in light field compressed sensing, these are related to their comparably large dimensionality and the underlying epipolar geometry. In image-based deep learning, convolutional neural networks (CNNs) have been the go-to choice neural network architecture ever since the groundbreaking performance of AlexNet in the ImageNet Large Scale Visual Recognition Challenge in 2012 [107]. Since then, CNNs have pushed the state of the art in many image-based tasks, in particular with the introduction of downsampling, pooling, and residual convolutions to make the training of very deep architectures, such as VGG [179] and ResNet [76], feasible. In fact, it can be shown analytically that, in the discrete case, the convolution is the only linear operator that ensures translational equivariance by design [62], which is a useful inductive bias in many image-based computer vision tasks. In recent years, this has led to generalization of CNNs to spheres [42] and even arbitrary Riemannian manifolds [215].

Therefore, it would be the most straightforward approach to use 4D convolutions for light field deep learning applications. In the case of spectral light fields, 5D convolutions would be the natural choice. However, these high-dimensional convolutions are much more parameter-intensive and computationally complex than the standard 2D and 3D convolutions. Furthermore, there is no native implementation of 4D or 5D convolutions in CUDA which is necessary for the GPU-accelerated training of the neural networks using Nvidia GPUs. To overcome these challenges, the following approaches have been discussed in the literature.

By restricting the 4D convolutional kernels $\boldsymbol{\mathcal{K}}$ to those that are *separable* in the spatio-angular domain, *i.e.*

$$\boldsymbol{\mathcal{K}}_{uvst} = \boldsymbol{\mathcal{A}}_{uv} \otimes \boldsymbol{\mathcal{B}}_{st} \tag{3.25}$$

for some 2D kernels $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, one can achieve 4D convolutions by alternatingly applying 2D convolutions in the angular and spatial domain, respectively. Moreover, the number of parameters as well as the computational complexity is reduced as compared to a native 4D convolution.

The approach to separate high-dimensional filter kernels into lower-dimensional ones is quite common in image processing and has been around for decades. However, in the context of light field deep learning, it was first proposed by Yeung *et al.* [227]. Here, the spectral domain is not explicitly taken into account for the convolution but merely viewed as a "channel" axis, *i.e.* for every spectral dimension a different kernel is learned (as it is also common for 2D convolutions applied to RGB images). On a practical note, this approach is not as straightforward to implement as it seems since CUDA and common high-level GPU-accelerated tensor frameworks such as TensorFlow and PyTorch support 2D convolutions only on 4D inputs with one batch, channel, and two spatial axes. Hence, to perform 2D convolutions on a 6D mini-batch of spectral light fields directly, one has to either implement a corresponding CUDA kernel or a TensorFlow/PyTorch layer utilizing only the 2D convolution on 4D inputs. Furthermore, even though the spatio-angular separation decreases the complexity of 4D convolutions, it is still comparably expensive.

Therefore, there exist several other 2D and 3D convolution-based architectures in the context of light field deep learning. In order to employ the conventional 2D and 3D convolution of GPU-accelerated tensor frameworks, one has to reshape the batch of input light fields to 4D and 5D, respectively, or use a multi-input approach. A common practice is to use a single EPI volume or the so-called crosshair sections of the light field, corresponding to the vertical, horizontal, and diagonal EPI volumes, resulting in a multi-input architecture. For example, this approach is used by the well-known EPINET disparity estimation network [176] or the multi-task network proposed by Alperovich *et al.* [4]. Alternatively, stereo-view pairs [175] or either the full or a sparse subset of the light field are used [129, 160, 227, 228]. Furthermore, one can perform 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. Finally, one can also simply stack the subaperture views channel-wise, partially losing angular information. Depending on the reshape, spatial, angular, or spatio-angular convolution can be performed by using a standard (possibly dilated and/or strided) 2D or 3D convolution on the reshaped light field. An overview of the most commonly used light field reshapes is given in Figure 3.3.

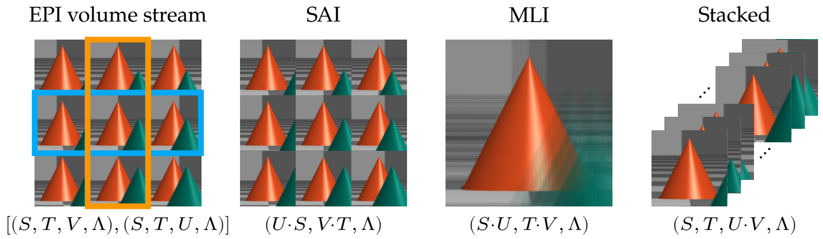

EPI volume stream
SAI
MLI
Stacked
$[(S,T,V,\Lambda),(S,T,U,\Lambda)]$
$(U{\cdot}S,V{\cdot}T,\Lambda)$
$(S{\cdot}U,T{\cdot}V,\Lambda)$
$(S,T,U{\cdot}V,\Lambda)$

Figure 3.3 Different reshapes of a light field of shape (U, V, S, T, Λ) with corresponding resulting shapes.

Inspired by compressed sensing and the success of autoencoder networks [82], a dual-stream U-net architecture [171], *i.e.* an encoder-decoder network with skip connections, is proposed here using multiple decoder streams to decode the spectral central view and disparity from the coded light field. The main idea of the encoder-decoder architecture is to map the coded light field to a well-adapted low-dimensional latent space. Then, using two jointly trained decoder paths, the central view and the disparity map are decoded from the latent representation. Ideally, this latent representation is invariant under the actual coding of the light field such that different codings of a light field are mapped to the same latent representation and subsequently decoded to identical central views and disparity maps. Additional to the standard encoder-decoder architecture, the U-net design introduces skip connections to share features between the encoder and decoder paths. In principle, the proposed method is not specific to the low-dimensional reconstruction target, *i.e.* it is straightforward to extend the encoder by an additional upsampling block to achieve superresolution in either one of the separate decoder paths, or to add (or replace) decoder paths to estimate different light field properties. Two different encoders are investigated in this thesis, one built upon separable 4D convolutions and one using 3D convolutions in the (reshaped) spatio-angular domain. The network architectures are shown in Figure 3.4 and Figure 3.5. In both cases, the full coded light field is used as the network's input, as it is important to use all of the available information in the case of a sparsely sampled input.

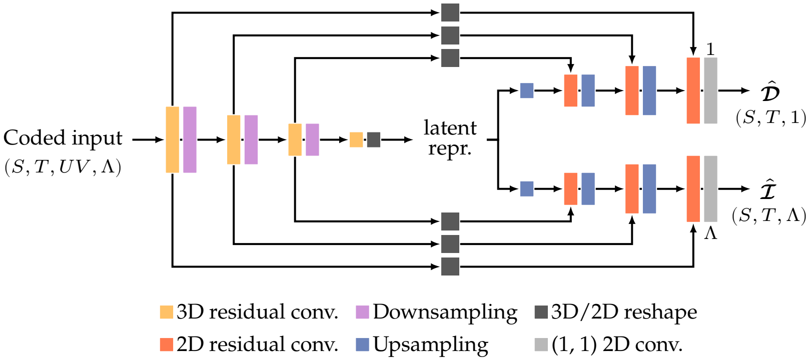


Figure 3.4 Schematic overview of the used dual-task encoder-decoder network based on 3D/2D residual convolutions. The depicted shapes neglect the batch axis. Details on the used blocks can be found in Figure 3.6.

For the proposed 3D convolution-based architecture, the subapertures of the full light field are stacked along a single axis, resulting in a 4D input of shape (S, T, UV, Λ). This way, 4D spatio-angular convolution can be approximated by using a 3D convolution. However, the angular information is lost to some extent because the flattened angular axis suffers from discontinuities in the epipolar geometry. For example, consider a light field with $(9, 9)$ angular resolution that is flattened to a single axis with 81 elements. The epipolar geometry is only consistent within blocks of length nine (the individual EPI volumes) after which a discontinuity occurs. The elements of these blocks do not need to be in consecutive order, as the horizontal, vertical, or diagonal EPI volumes can be built using different strides. Convolving a filter across these discontinuities will lead to artifacts which the network will have to learn to mitigate or circumvent, in particular using the higher-level features in deeper layers. For this reason, usually, the EPI volumes are fed into the network separately as previously noted. However, in this multi-input approach, not all light field information is available to every layer which is argued to be sub-optimal in the case of coded light fields. For example, using the proposed reshape and 3D convolution, the epipolar geometry in all angular directions can be utilized and not only the ones along a single angular axis as is the case for multi-input models. In fact, the proposed

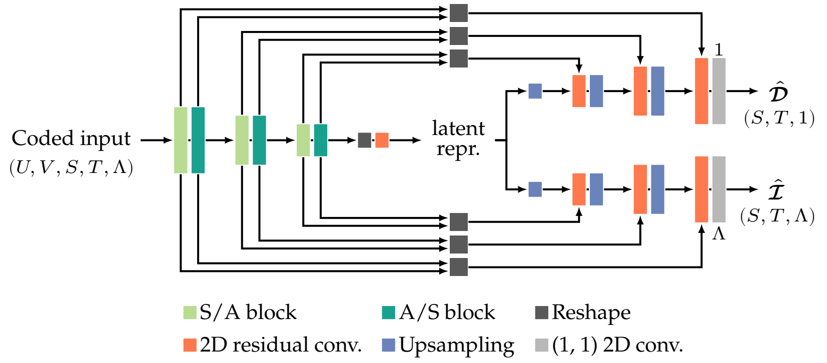


Figure 3.5 Schematic overview of the used dual-task encoder-decoder network based on separable 4D residual convolutions. The depicted shapes neglect the batch axis. Details on the used blocks can be found in Figure 3.6.

angular flattening in combination with 3D convolution turns out to be well-suited for the considered case as will be shown in the evaluation.

To extract the spatio-angular features, the full encoder path is then built upon 3D residual convolution blocks [76] which consist of two conventional convolution layers with intermediate batch normalization [93] and ReLU activation. The residual connection of the block, consisting of a convolution with a kernel size of one, allows for improved gradient backpropagation as compared to a conventional convolution layer. The number of used filters per layer is doubled after every downsampling layer starting from 24 in the first residual block to 192 in the latent-space residual block. All convolutions of the encoder are 3D convolutions with a kernel size of $(3, 3, 3)$, except for the residual convolutions which use a kernel size of $(1, 1, 1)$. Downsampling is performed via strided convolutions with a kernel size of $(3, 3, 3)$ and a $(2, 2, 2)$ stride. Doing so, the network remains fully convolutional, retaining the translational equivariance, as opposed to downsampling via pooling such as the often-used max-pooling [95].

The decoder is built symmetrically for both decoding paths, however 2D residual and transposed convolutions are used instead because the estimated disparity and central view do not have any angular dependence. In the last layer of each stream, a $(1, 1)$ convolution with either

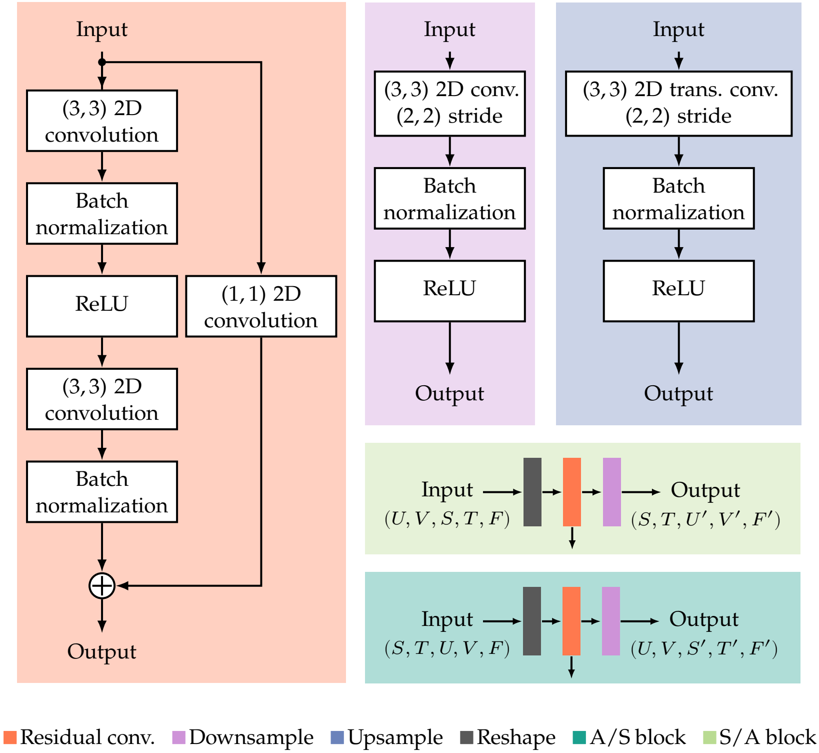


Figure 3.6 Schematic drawings of the used network blocks. In the case of 3D convolutions, all 2D convolutional blocks are replaced with 3D blocks as detailed in the main text.

13 or one features is performed to obtain the final shapes $(S, T, 13)$ and $(S, T, 1)$ for the reconstructed central view and disparity, respectively. The encoder features are joined with the decoder features via skip connections by concatenation. Therefore, 3D to 2D reshapes are necessary when connecting the encoder and decoders. The reshape converts a light field tensor of shape (S, T, UV, Λ) to $(S, T, UV\Lambda)$, ignoring the batch axis. This way, the proposed architecture combines a 3D U-net encoder [41] with dual-stream 2D U-net decoders [171]. The 3D convolution network architecture is depicted in Figure 3.4.

For the encoder based on separable 4D convolutions, the spatio-angular (S/A) and angular-spatial (A/S) blocks are used as depicted in Figure 3.6. Here, spatio-angular and angular-spatial reshapes, permuting the axes of the input light field of shape (U, V, S, T, Λ) to (S, T, U, V, Λ) and vice versa, are used. In the case of the S/A block, 2D residual convolution is then performed in the angular domain, while for the A/S block the convolution is performed in the spatial domain. After the residual convolution, the feature maps are downsampled. For the A/S block, the spatial downsampling is performed using (2, 2)-strided convolutions, similar to the 3D convolution-based encoder. In the case of the angular downsampling that is used in the S/A block, such a strided convolution is not suitable due to the low angular input resolution of (9, 9). Therefore, here an implicit downsampling is performed by using 2D convolution with a kernel size of (3, 3) in the angular domain *without padding*. This way, the angular resolution is reduced by one from both sides in each dimension. For example, the input angular resolution of (9, 9) is thus reduced to (7, 7) after the first S/A block. The number of features in the input layer is again 24 and doubled after each spatial downsampling. The dual-stream decoder is identical for both architectures. The 4D convolution architecture is shown in Figure 3.5.

As shown in Table 3.2, the number of trainable parameters of both architectures is roughly the same, with a slightly larger number in the case of the 4D convolutional architecture. In fact, both architectures can be considered small- to mid-sized with respect to established CNN architectures. For example, the smallest ResNet model, ResNet-50 with 25.5 million parameters, is of similar size, while the larger ResNet models, ResNet-101 and ResNet-152, already have 44.6 and 60.3 million trainable parameters, respectively [76, 77]. Other well-established architectures are even larger, *e.g.* VGG-16 already has 138.6 million trainable parameters [179], not even considering recent Transformer-based models such as the Vision Transformer which has 632 million parameters in its best-performing configuration [54]. Despite the comparably small size, in terms of trainable parameters, the memory requirements of the proposed architectures are considerable, in particular due to the used residual and skip connections. For these, the intermediate feature representations have to be saved temporarily, increasing the memory requirements as

Table 3.2 Complexity comparison of the two proposed network architectures. The shown GPU memory corresponds to the memory required during training using light fields of shape $(9, 9, 32, 32, 13)$, a mini-batch size of 64, and the adaptive Yogi [233] optimizer, which tracks three parameters per trainable network parameter.

Model	Parameters /M	FLOP /M	GPU memory /GB
3D Conv.	26.3	8.8	8.2
4D Conv.	29.7	12.3	17.7

compared to a similar architecture without residual and skip connections. Again due to the high dimensionality of the light fields, these feature representations are higher dimensional than in conventional image-based architectures for which the memory requirements of the intermediate representations is not as severe. Still, in precursory experiments it was found that introducing skip connections drastically improves the test performance and convergence, leading to shorter training times, which is in accordance with findings in the literature [55, 158]. Furthermore, larger models typically also need more training data to be trained to adequate test performance whilst avoiding overfitting. Therefore, for the considered case and the used training dataset (*cf*. Section 4.1), the size of the proposed architectures is argued to be adequate.

Finally, comparing the two proposed architectures in terms of their computational and memory complexity, as denoted in Table 3.2, the floating point operations (FLOP) of the separable 4D convolutional architectures as well as the necessary GPU memory during training is much higher compared to the architecture based on 3D convolutions, despite having a similar number of trainable parameters. The higher memory usage is caused by the increased dimensionality of the skip connections: Since both angular as well as spatial features are used for each skip connection, the number of features is doubled as compared to the 3D convolution architecture. Because these representations have to be saved intermediately in order to be concatenated to the corresponding features in the decoder, the memory overhead is much larger.

3.2.3 Training strategies

3.2.3.1 Multi-task training

Using the proposed network architectures, the training is inherently a multi-task problem: the disparity estimation and central view reconstruction are trained jointly. In the case of N tasks, the naive multi-task approach is to use a weighted sum of the individual task losses L_i as the overall training loss

$$L = \sum_{i=1}^{N} w_i \, L_i + L_{\text{reg}} \, . \tag{3.26}$$

Here, $w_i > 0$ are the task weights and L_{reg} is a task-independent regularization term (such as weight decay), which is neglected in the following but is implicitly assumed to be added to the final loss. The first challenge in this straightforward approach is to find suitable task weights w_i, which is time- and resource-intensive. Furthermore, it may not even be possible to find optimal *static* task weights. During the training, for each mini-batch the gradient of the loss with respect to the trainable network parameters $\mathcal{W}$,

$$\nabla_{\mathcal{W}} L = \sum_{i=1}^{N} w_i \nabla_{\mathcal{W}_i} L_i \, , \tag{3.27}$$

is calculated and used to update the network parameters via SGD. Here, the parameters $\mathcal{W}_i$ contain those shared across all tasks (the encoder parameters), as well as task-specific parameters (the individual decoder parameters). The shared parameters are hence updated based on the gradients from all tasks, which can be problematic: the gradients from the different tasks may be on different scales leading to a task imbalance during the update of the network parameters $\mathcal{W}$. Finally, the tasks may also be of different complexity, leading to different convergence speeds, which further enhances the task imbalance.

For example, consider a dual-task toy example where the two tasks are assumed to be similar in terms of complexity and scale. One task uses the mean absolute error (MAE) and the other task the mean squared error (MSE) as its loss function. While the MAE yields constant gradients, the gradients of the MSE-trained task will be very large in the beginning, dominating the update of the shared parameters. During training, the MSE-based gradients will decay, shifting the balance to the other task. Therefore, manually choosing the task weights w_i such that the training is balanced in the early stages will lead to an imbalance later in the training and vice versa.

To overcome these difficulties, some approaches to *dynamically* update the task weights w_i during training have been recently discussed. Here, the approach by Kendall *et al.* [102], using the multi-task uncertainty, as well as the GradNorm method by Chen *et al.* [39] are considered, both of which have shown good results in the context of computer vision.

Obtained from maximizing the log-likelihood of the model considering the single-task uncertainties σ_i, Kendall *et al.* propose to use the loss

$$L = \sum_{i=1}^{N} \frac{1}{2\sigma_i^2} L_i + \ln \sigma_i \tag{3.28}$$

instead of the naive loss (3.26). During training, the task weights

$$w_i = 1/\left(2\sigma_i^2\right) \tag{3.29}$$

are considered trainable parameters themselves and are updated in the same fashion as the remaining network parameters via backpropagation and SGD. Intuitively, the additional terms $\ln \sigma_i$ prevent the task weights from converging to zero, which would otherwise be a trivial solution of the minimization of the loss L. Despite its simplicity, low overhead, and comparably straightforward implementation, this approach has shown good performance in segmentation and monocular depth estimation [102].

Explicitly taking into account the tasks' gradient norms and convergence speeds, Chen *et al.* [39] propose the GradNorm method which introduces the *additional* loss

$$L_{\text{grad}} = \left| w_i \|\mathbf{G}_i\| - G_{\text{mean}} \cdot r_i^{\alpha} \right| \tag{3.30}$$

to optimize the (now trainable) weights w_i of the main loss (3.26). Here, the weights w_i are used to bring the single-task gradients $\mathbf{G}_i$ to a common scale using the mean weighted gradient norm

$$G_{\text{mean}} = \frac{1}{N} \sum_{i=1}^{N} w_i \|\mathbf{G}_i\| \,. \tag{3.31}$$

The single-task gradients

$$\mathbf{G}_i = \nabla_{\mathcal{W}_\mathrm{s}} L_i \tag{3.32}$$

are calculated for every mini-batch during training with respect to a set of shared network parameters $\mathcal{W}_\mathrm{s}$ to make sure that the individual gradients are elements of the same vector space and are hence comparable. Usually, the parameters of the last shared layer are used for the gradient calculations. The mean weighted gradient norm G_mean is weighted by the current relative inverse task learning rate r_i^α, where α is a hyperparameter that can be used to manually adjust the task training speed imbalance. The individual inverse task learning rates are calculated as

$$\tilde{L}_i = L_i \Big/ L_{\mathrm{init},i} \,, \tag{3.33}$$

where L_i and $L_{\mathrm{init},i}$ denote the task loss of the current and the initial mini-batch, respectively. Using the individual learning rates, the relative inverse task learning rate is calculated as

$$r_i = \tilde{L}_i \Big/ \frac{1}{N} \sum_{i=1}^{N} \tilde{L}_i \,. \tag{3.34}$$

With respect to the calculation of the gradients from (3.30), $\mathbf{G}_i$, G_mean, and r_i are considered constant, *i.e.* neither explicitly depending on the network's parameters nor the task weights w_i. The network parameters are updated using only the main loss (3.26).

Compared to the approach by Kendall *et al.*, the GradNorm method is computationally more expensive as it involves N additional gradient computations. Depending on the number of shared parameters $\mathcal{W}_\mathrm{s}$, with respect to which the individual task gradients are calculated, GradNorm may also be quite memory-intensive. For further technical details on these two approaches, the reader is referred to the original literature.

When the multiple tasks compete, it may be necessary to adapt methods from multi-objective learning [118, 174]. In the explored case of central view and disparity estimation, however, it was found that the used multi-task approaches perform on par or better than the corresponding single-task networks. Therefore, multi-objective approaches were not considered. However, it may become necessary when using different reconstruction targets. Also, grouping of multi-tasks in multi-objective and multi-task subgroups has also been recently investigated [183].

3.2.3.2 Auxiliary loss training

In computer vision tasks, often the MSE or MAE is used as a single-task loss function in regression tasks. The use of the MSE as the primary optimization and evaluation metric is well justified since it corresponds to the energy of the reconstruction error (*cf*. Section 4.3). Furthermore, the MSE is convex and continuously differentiable. However, both the MSE and the MAE are only evaluated pixel-wise and subsequently averaged. In particular, neither spatial nor spectral correlations are considered. Often, however, secondary quality metrics, such as the structural similarity index metric [210] to assess the spatial reconstruction quality or the spectral angle and the spectral information divergence [34] to evaluate the spectral reconstruction quality, are of key interest. To utilize these secondary metrics during optimization, ignoring the multi-task scenario for now, one can use the loss function

$$L = L_{\text{main}} + \sum_{j=1}^{N_{\text{aux}}} w_{\text{aux},j} L_{\text{aux},j} \tag{3.35}$$

using N_{aux} auxiliary loss functions $L_{\text{aux},j}$ to support the main loss L_{main}. In image processing, the combination of the MSE and SSIM has been shown to outperform the single-loss training [234], however, the loss weights were fine-tuned manually. In principle, the problems with this naive approach are similar to those in multi-task learning: namely, the manual (static) choice of the auxiliary loss weights $w_{\text{aux},j}$ and the possibly different scales of the loss gradients. Furthermore, the additional losses may even be adversarial to the main loss, leading to canceling gradients, and the overall loss landscape may suffer from high curvature. This has also been referred to as the *tragic triad* [229]. To mitigate this, the use of gradient similarity (GradSim) has been proposed Du *et al*. [56], however, similar approaches have also recently been discussed in the context of reinforcement learning [120] and meta-learning [122].

In the case of the GradSim approach by Du *et al*., the auxiliary loss weights are calculated as

$$w_{\text{aux},j} = \max\left\{0, \frac{\langle \mathbf{G}_{\text{main}}, \mathbf{G}_{\text{aux},j}\rangle}{\|\mathbf{G}_{\text{main}}\| \cdot \|\mathbf{G}_{\text{aux},j}\|}\right\}, \tag{3.36}$$

for each mini-batch during the training. The loss is based on the gradient similarity between the main and the auxiliary task gradients

$$\mathbf{G}_{\text{main}} = \nabla_{\mathcal{W}} L_{\text{main}}\,, \quad \mathbf{G}_{\text{aux},j} = \nabla_{\mathcal{W}} L_{\text{aux},j}\,, \tag{3.37}$$

calculated via the scalar product. The $\max$ operation ensures that only positive auxiliary gradient contributions are taken into account. While this resolves the issue concerning adversarial auxiliary losses and canceling gradients, the problem of different gradient norms persists. Furthermore, since the calculation is performed for each mini-batch individually, the updated weights $w_{\text{aux},j}$ may be quite noisy. Du *et al.* mention the possibility to use a moving average, however, no evaluation has been performed for this.

To overcome these limitations, it is proposed to use a modification of the above, named *normalized gradient similarity* (NormGradSim). Again, the main loss is extended, using N_{aux} auxiliary losses $L_{\text{aux},j}$, to

$$L = \left(L_{\text{main}} + \sum_{j=1}^{N_{\text{aux}}} \alpha_j \beta_j \, L_{\text{aux},j} \right) \Big/ \left(1 + \sum_{j=1}^{N_{\text{aux}}} \alpha_j \right) \tag{3.38}$$

with dynamic weights $\alpha_j, \beta_j > 0$ which are updated via SGD using the additional losses

$$L_\alpha = \sum_j \left| \alpha_j - \max\left\{ 0, \frac{\langle \mathbf{G}_{\text{main}}, \mathbf{G}_{\text{aux},j} \rangle}{\|\mathbf{G}_{\text{main}}\| \cdot \|\mathbf{G}_{\text{aux},j}\|} \right\} \right|, \tag{3.39}$$

$$L_\beta = \sum_j \Big| \beta_j \cdot \|\mathbf{G}_{\text{aux},j}\| - \|\mathbf{G}_{\text{main}}\| \Big| . \tag{3.40}$$

While α_j is used to weigh each auxiliary loss according to its gradient similarity with the main loss, β_j is used to bring the gradient of the auxiliary loss to the same scale as the main loss. The normalization term $(1+\sum_j \alpha_j)$ keeps the resulting gradient of the total loss L on the same scale as the original main loss L_{main}. This has the advantage that NormGradSim can be used as a drop-in replacement to the single loss training (without having to adapt hyperparameters such as the optimizer's learning rate) or combined with multi-task approaches such as the approach using multi-task uncertainty or GradNorm. When calculating the gradients of the main loss L, α_j and β_j are considered constant.

Finally, combining the multi-task scenario with the proposed NormGradSim method, one obtains the final overall loss function

$$L = \sum_{i=1}^{N} w_i \left(L_{\text{main}}^{(i)} + \sum_{j=1}^{N_{\text{aux}}^{(i)}} \alpha_j^{(i)} \beta_j^{(i)} \, L_{\text{aux},j}^{(i)} \right) \Big/ \left(1 + \sum_{j=1}^{N_{\text{aux}}^{(i)}} \alpha_j^{(i)} \right), \tag{3.41}$$

which is used in combination with the additional loss functions L_α, L_β.

3.3 Mask optimization via neural fractals

In the case of the proposed principal reconstruction, different (stochastic and regular) coding masks are investigated, as will be discussed in Section 4.4.2. To go one step further, one may optimize the used coding mask with respect to the considered downstream tasks, *i.e.* for the reconstruction of the central view and the disparity estimation in the case considered here. The design and optimization of coding masks have been discussed in the literature in several instances, however, it poses some challenges that have only been addressed partially in the past. The main challenge is that the coding mask is binary and therefore does not directly allow for standard optimization techniques such as gradient descent. Due to the immense dimensionality of the search space, in particular in the multispectral case, a brute force approach is also usually not suitable. In the past, coding masks were often designed based on expert knowledge, *e.g.*, in the case of color imaging, the well-known Bayer pattern [16] and its derivatives, or more recent proposals based on sparse color sampling [33]. Alternatively, heuristic optimization approaches have been utilized, *e.g.*, in the context of multispectral imaging, via binary tree search [141], genetic algorithms [220], or simulated annealing [177]. However, since the overall architecture of the proposed principal reconstruction is fully differentiable, it is desirable to also incorporate the mask generation into the framework and learn an optimal mask (with respect to the considered downstream tasks) in an end-to-end fashion. Such a framework has the additional benefit to be task-independent, *i.e.* different optimal masks could be obtained for different downstream tasks such as disparity estimation, segmentation, *etc*. To achieve a fully differentiable generation of binary coding masks, two approaches have recently been discussed in the literature.

First, the Deep Probabilistic Subsampling (DPS) approach by Huijben *et al.* [90] aims at learning a probability distribution from which a coding mask can be generated. In order to achieve a discrete distribution, the Gumbel-max trick is used [75]. That is, to parametrize the discrete categorical distribution, random noise is drawn from the Gumbel distribution and added to the trainable non-normalized logits. Subsequently, the temperature softmax is used as a differentiable approximation of the one-hot encoded argmax function, continuously relaxing the binary optimization as will be discussed shortly. To achieve a nearly discrete distribution, an additional loss is introduced penalizing high entropy, *i.e.* favoring "spiky", nearly one-hot encoded distributions. The entropy loss weight is linearly increased during training, which can be interpreted as a form of annealing. The approach has the inherent drawback of being stochastic, *i.e.* a mask distribution is learned instead of a static mask. However, in the considered case one is interested in a fixed mask that can be realized in hardware and which cannot be altered afterwards. To some extent, this issue is addressed in a recent generalization of the approach, called Active DPS [203], introducing inter-sample dependencies but retaining the probabilistic nature. Furthermore, the DPS approach considers a fixed spatial resolution that cannot be changed after training which is unsuitable in the presented case due to the smaller light field resolution of the training dataset. One could of course perform full-sized inference by patching the light field into smaller patches, similar to the compressed sensing-based approach using dictionaries, however, this would introduce a lot of computational overhead. Simultaneously to DPS, a similar formulation was published dubbed Concrete Autoencoders [14].

Second, Chakrabarti [32] proposes to learn the color filter layout of a color camera jointly with a deep learning-based demosaicing approach. Here, the color filter array is generated using a small patch of trainable weights $\mathcal{W}$ of shape (A, B, C) where (A, B) corresponds to the spatial size of the filter macropixel (*e.g.* $(2, 2)$ in the case of a conventional Bayer pattern) and C to the number of color channels (typically three in the case of RGB imaging, however, approaches using more color channels have also been considered in the literature). Similar to the DPS approach, the one-hot encoded argmax is approximated using the temperature softmax, *i.e.* the color filter is normalized along the channel axis via

$$\boldsymbol{\mathcal{M}}_{abc} = \mathrm{softmax}_\tau \boldsymbol{\mathcal{W}}_{abc} = \frac{\mathrm{e}^{\boldsymbol{\mathcal{W}}_{abc}/\tau}}{\sum_{c'} \mathrm{e}^{\boldsymbol{\mathcal{W}}_{abc'}/\tau}} \,. \tag{3.42}$$

In the limit $\tau \to 0$, the temperature softmax converges to the (non-differentiable) one-hot encoded argmax [130], which is one at the position of the largest elements along the last axis and zero otherwise. Chakrabarti anneals the temperature τ during training, instead of using the entropy along the channel axis to obtain nearly binary coding masks. The smaller the temperature, the smaller the bias of the gradient estimate, at the cost of a higher variance [90]. Furthermore, analogous to the DPS approach, the true argmax function is used in the forward pass and the temperature softmax is used only in the backward pass, *i.e.* to calculate the gradients via backpropagation. With this additional trick, the downstream task always "sees" a physically correct binary coding mask while the gradients can be calculated via backpropagation and the mask weights can be optimized via SGD. By spatially repeating the patch $\boldsymbol{\mathcal{M}}$, masks of arbitrary size can be generated which are then applied to the input image.

Here, the approach by Chakrabarti [32] is generalized. There are mainly three properties that a coding mask and the subsequent reconstruction should fulfill: the coding and reconstruction should ideally be translational, rotational, and scale equivariant, *i.e.* the reconstruction should not depend on the position and orientation of the camera with respect to the scene, as well as the sensor size, resolution, or pixel pitch. Furthermore, the coding and reconstruction should generalize when applied to images larger than those in the training dataset. Due to the periodic nature of conventional regular coding masks and a fully convolutional layout of the downstream task, as considered by Chakrabarti, translational and scale equivariance are approximately fulfilled. Rotational equivariance is typically achieved via online augmentation during training. Here, the focus lies on generalizing the optimization of the regular coding masks as proposed by Chakrabarti while maintaining the mentioned properties. To this end, the coding masks are generated and optimized as fractals, which fulfill scale invariance by design, while possibly being more expressive than regular coding masks.

First, as an illustration, the simple case of binary fractals is considered. Here, the fractals are formulated as a Lindenmayer- or L-system[1] [121]. A binary fractal mask is generated using two binary base patterns

$$p_0, p_1 \in \{0,1\}^{M\times M}, \tag{3.43}$$

where only square patterns of size $M \times M$ are considered for simplicity. The fractals are then created recursively using the base patterns. Starting with a root node $x_0 \in \{0,1\}$, the node is replaced by the corresponding base pattern, *i.e.* with p_0 if $x_0 = 0$ or p_1 if $x_1 = 1$. Then, in the recursion step, each pixel in the created mask is again replaced by the corresponding base pattern, depending on the binary pixel value. To formulate this in a more precise fashion, the base patterns are viewed as the image of the function

$$p : \{0,1\} \to \{0,1\}^{M\times M} \tag{3.44}$$

$$0 \mapsto p_0\,, \quad 1 \mapsto p_1\,. \tag{3.45}$$

For the recursive call, the function is generalized to

$$P_n : \{0,1\}^{M^n \times M^n} \to \{0,1\}^{M^{n+1}\times M^{n+1}} \tag{3.46}$$

$$\mathbf{x} \mapsto P_n(\mathbf{x})\,, \quad x_i \mapsto p(x_i)\,, \tag{3.47}$$

for any $n \in \mathbb{N}_+$, where x_i denotes the individual pixel value. The recursive generation can then simply be formulated as:

$$\begin{aligned} &\text{▪ choose root} && x_0 \in \{0,1\}\,, \\ &\text{▪ initialize} && \mathbf{x}_1 = P_0(x_0) \equiv p(x_0)\,, \\ &\text{▪ apply recursion} && \mathbf{x}_{n+1} = P_n(\mathbf{x}_n)\,. \end{aligned} \tag{3.48}$$

The generation of an exemplary fractal is depicted in Figure 3.7. Here, the root node

$$x_0 = 1 \tag{3.49}$$

is chosen arbitrarily.

[1] The idea to use fractals formulated via L-systems as coding masks and an initial implementation came up during a discussion with Jonas Köhler (Freie Universität Berlin) and cannot be considered the sole contribution of the author.

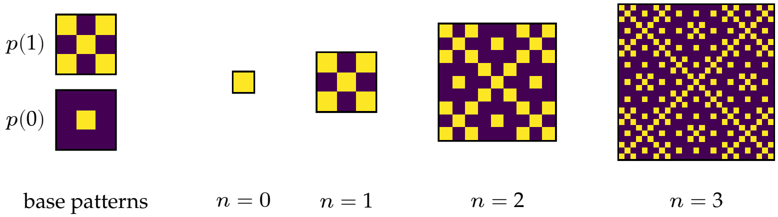


Figure 3.7 Generation of an exemplary binary fractal mask in the case of 3×3 base patterns. For visualization, the pixel size of the larger masks is decreased.

To generalize the binary approach to the multispectral case, the base patterns are generalized to

$$p : \{0,1\}^{\Lambda} \to \{0,1\}^{M\times M\times\Lambda}\,, \tag{3.50}$$

which are continuously relaxed to

$$p : \mathbb{R}^{\Lambda} \to \mathbb{R}^{M\times M\times\Lambda}\,. \tag{3.51}$$

In this continuous case, the base pattern function can now equivalently be viewed as the tensor

$$\boldsymbol{\mathcal{P}} \in \mathbb{R}^{M\times M\times\Lambda\times\Lambda}\,. \tag{3.52}$$

In order to achieve the desired one-hot encoding along the spectral axis of the mask, the base pattern tensor is obtained from a weights tensor of identical shape via

$$\boldsymbol{\mathcal{P}}_{ijkl} = \operatorname{argmax}_l \boldsymbol{\mathcal{W}}_{ijkl}\,. \tag{3.53}$$

Here, the tensor $\boldsymbol{\mathcal{W}}$ corresponds to the trainable parameters of the mask generation. Again, the non-differential argmax is approximated by the temperature softmax for the backward pass during training. The recursive generation of the mask, in complete analogy to the binary case, can now be formulated using the base pattern tensor as:

$$\begin{array}{ll} \blacksquare\ \text{choose root} & \boldsymbol{\mathcal{X}}^{(0)} \in \mathbb{R}^{1\times 1\times\Lambda}\,, \\ \blacksquare\ \text{apply recursion} & \tilde{\boldsymbol{\mathcal{X}}}^{(n+1)}_{abcij} = \sum_k \boldsymbol{\mathcal{P}}_{abkc} \boldsymbol{\mathcal{X}}^{(n)}_{ijk}\,, \\ \blacksquare\ \text{reshape} & \tilde{\boldsymbol{\mathcal{X}}}^{(n+1)} \to \boldsymbol{\mathcal{X}}^{(n+1)} \\ & (M, M, \Lambda, M^n, M^n) \to (M^{n+1}, M^{n+1}, \Lambda)\,. \end{array} \tag{3.54}$$

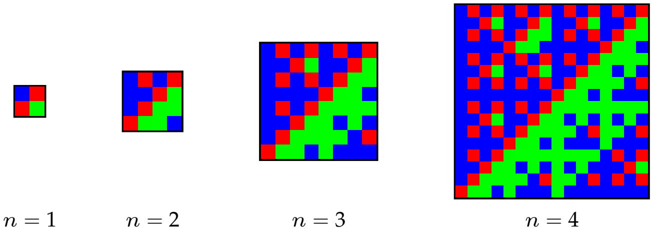


Figure 3.8 Generation of an exemplary fractal RGB mask $\boldsymbol{\mathcal{X}}^{(n)}$ using 2×2 base patterns.

Here, for simplicity, the root tensor is always chosen to be constant,

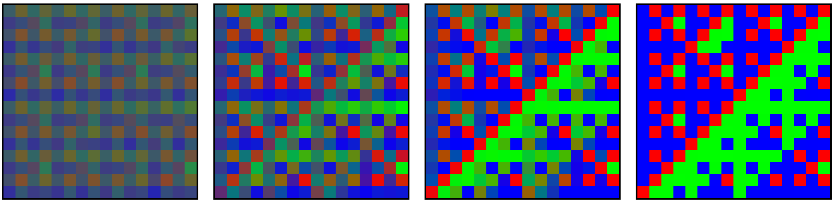
$$\boldsymbol{\mathcal{X}}^{(0)}_{ijk} = \begin{cases} 1 & \text{if } i = j = k = 0\,, \\ 0 & \text{otherwise.} \end{cases} \tag{3.55}$$

To create the full fractal coding mask, first, the recursion depth N is calculated from the spatial light field resolution and the predefined base pattern size M. Then, the recursion is applied until the calculated depth to obtain the coding mask $\boldsymbol{\mathcal{M}} = \boldsymbol{\mathcal{X}}^{(N)}$. If the base pattern size is not a true divider of the light field resolution, a larger mask is generated and centrally cropped to fit the target resolution. Since the mask generation is fully differentiable with respect to the parameters $\boldsymbol{\mathcal{W}}$, it can be easily integrated into the proposed principal reconstruction and jointly trained in an end-to-end fashion. Because the recursive mask generation can be interpreted as a neural network with N fully connected layers with argmax or softmax activation, it is proposed to refer to these generated fractal masks as *neural fractals*. During training, both the annealing of the temperature softmax as well as an entropy-based regularization are considered. To this end, the loss

$$L_{\mathrm{e}} = \operatorname{mean}\left(- \sum_{l} \boldsymbol{\mathcal{P}}_{ijkl} \ln \boldsymbol{\mathcal{P}}_{ijkl} \right) \tag{3.56}$$

is used whose loss weight is increased exponentially during training.

An example, using 2×2 base patterns and three color channels, is shown in Figure 3.8. Here, the true ("hard") forward pass, based on the argmax, is used. The mask is depicted using the corresponding RGB color associated with the pixel's filter. While in the remainder masks



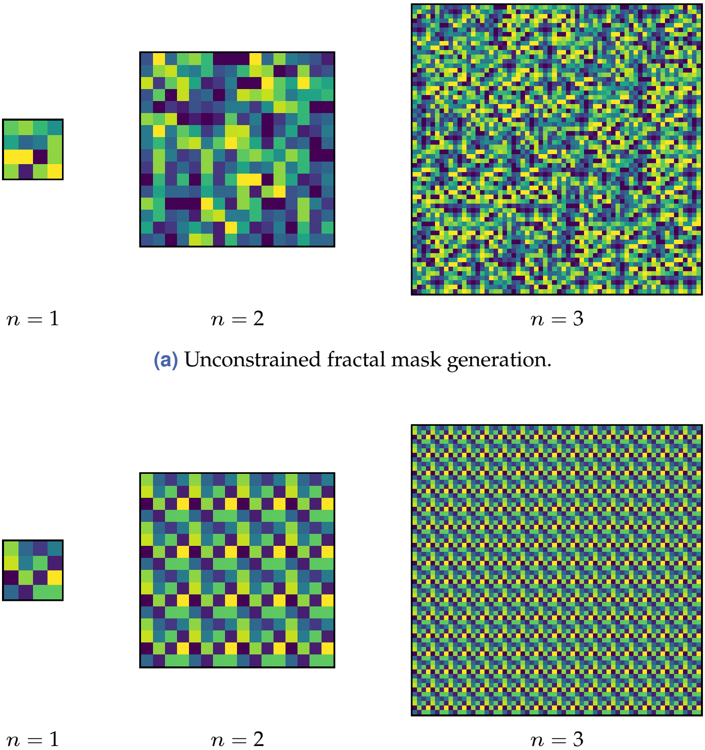
$\tau = 1$ $\tau = 1/3$ $\tau = 1/9$ $\tau = 1/81$

Figure 3.9 Continuous relaxation of the generated mask for different temperatures τ of the softmax used to generate the base patterns $\mathcal{P}$ from the trainable weights $\mathcal{W}$.

will be shown using the channel index, the RGB case has the advantage of being able to visualize the behavior of the softmax approximation. The same RGB mask, using a "soft" forward pass, is shown in Figure 3.9 for different temperatures of the softmax activation. It can clearly be observed that the mask becomes "spikier", *i.e.* more bandpass-like, the lower the temperature, as expected. This behavior is, of course, identical in the multispectral case which, however, is hard to visualize. As a final example, a multispectral fractal mask in the case of 4×4 base patterns using 13 spectral channels is given in Figure 3.10(a). While it is extremely difficult to understand the fractal nature in this high-dimensional case, it is obvious that these masks provide more powerful and flexible patterns as compared to conventional regular masks. However, note that conventional regular masks are also fractals. Using the proposed framework, regular masks are generated by constraining the weight tensor such that elements along the projecting axis are identical, *i.e.*

$$\mathcal{W}_{ijkl} = \mathcal{W}_{ijk'l} \tag{3.57}$$

for all k, k', effectively reducing the dimensionality of the pattern by a factor of Λ. Basically, the constraint means that, during generation, a pixel is always replaced with the same pattern, regardless of the pixel value. That is, the proposed fractal generation scheme is a true superset of the approach by Chakrabarti [32] and should therefore at least achieve on-par performance when properly optimized. An example of a regular coding mask generated with the proposed approach is shown in Figure 3.10(b). Concluding, despite the low dimensionality of the generating pattern, the neural fractal approach appears to be very expressive.

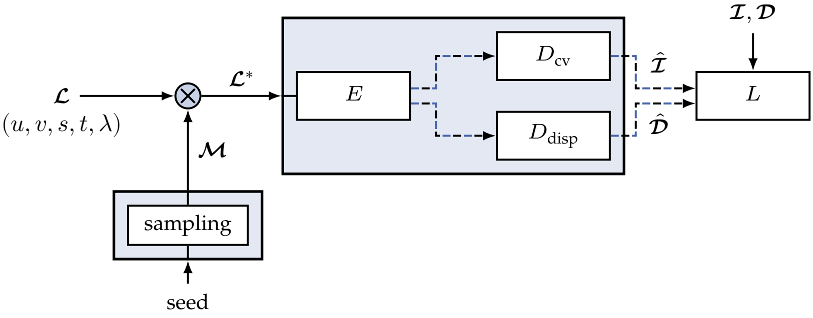


(a) Unconstrained fractal mask generation.

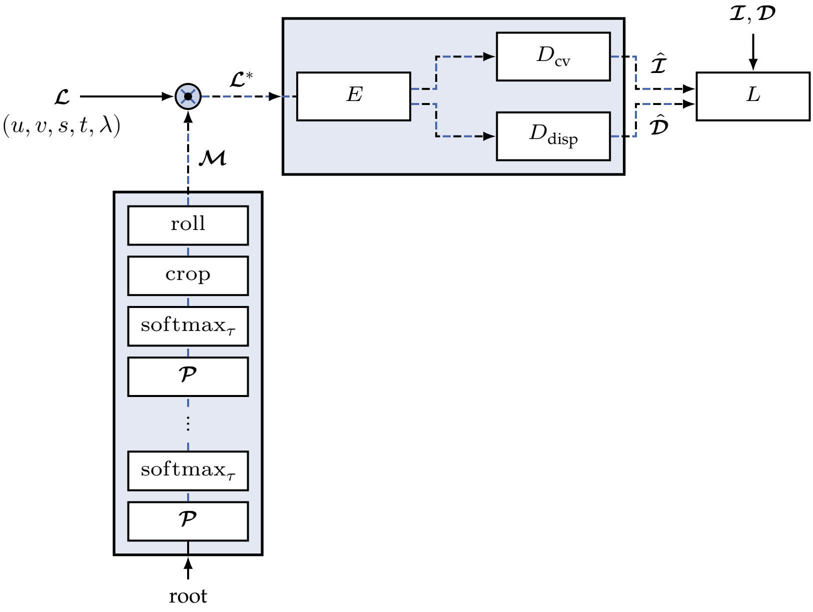
(b) Fractal mask generation constrained to regular masks.

Figure 3.10 Generation of two exemplary multispectral fractal masks $\boldsymbol{\mathcal{X}}^{(n)}$ with 13 spectral channels in the case of 4×4 base patterns using the hard forward pass. Here, the false-color corresponds to the index of the transmitting spectral channel. Dark values correspond to small and bright values to large channel indices.

A schematic comparison of training using static, predefined masks and end-to-end optimized masks via neural fractals is depicted in Figure 3.11. Here, E denotes the encoder whereas D_{cv} and D_{disp} denote the decoder for the central view and disparity, respectively. The trainable parameters of the encoder and decoders are not explicitly depicted for clarity. Furthermore, the shown loss L only corresponds to the reconstruction loss, *i.e.* regularization losses such as the entropy-based mask loss, are not depicted. In the case of using predefined masks (Figure 3.11(a)), the masks are randomly drawn from a predefined distribution (*cf.* Section 4.4.2) and multiplied element-wise with the input light field to simulate the coding. To optimize the neural fractal jointly with the downstream network, the gradients of the network and the parameters of the generating pattern $\boldsymbol{\mathcal{P}}$ are calculated via backpropagation. In this case, an element-wise multiplication with the input light field is not sufficient: the optimal mask would likely be purely constant, *i.e.* a tensor of all-ones. The gradients of the entropy-minimizing regularization loss and the main loss would likely cancel. Hence, the light field needs to be compressed after coding, *i.e.* projected along the spectral dimension analogously to the physical sampling by the camera according to (2.5). The element-wise multiplication and spectral projection correspond to a scalar product in the spectral domain. However, as argued previously, in this case, the network would have no information of the used masks, *i.e.* which pixels are associated with which spectral channel. In particular, this is problematic as the absolute position of the mask should not be learned by the network in order to generalize to larger spatial input sizes. To overcome this, compression, *i.e.* projection via the scalar product, is employed only in the backward path, while the element-wise multiplication is used for the forward path. This way, the network is always exposed to the full, non-projected, coded light field (and hence the associated spectral channel of each pixel), while the gradients are calculated using the projected measurement to avoid trivial solutions of constant-valued masks opposing the constraint of spectral one-hot encoding. This is symbolically denoted by the mixed scalar and element-wise multiplication in Figure 3.11(b). Intuitively, this approach is equivalent to passing the coded and projected light field $\boldsymbol{\mathcal{L}}^{*}$ to the network together with a static, non-differentiable binary mask indicating the indices of the used spectral channels in the coding mask.



(a) Training using a mask sampled from predefined distribution.



(b) Training using a differentiable mask generation via neural fractals.

Figure 3.11 Training with and without end-to-end optimization of the coding mask. Dashed lines indicate edges of backpropagation. The parametrizations of the encoder and decoders are neglected for clarity.

4 Experimental Setup

4.1 Synthetic dataset

With the advent of deep learning, the demand for training and test data, both labeled and unlabeled, has increased dramatically. In the case of RGB light fields, several synthetic datasets of varying scope with ground truth disparity labels have been published, including the well-known HCI benchmark dataset [85], the HCI specular light field dataset [4], the INRIA dataset [175], and the Graz University dataset [79]. However, a *spectral* light field dataset with disparity ground truth is yet missing. The newly created dataset fills this gap. To the best of the author's knowledge, this is the first spectral light field dataset with ground truth depth and disparity labels.

To create light fields that can be used for supervised data-driven disparity estimation methods, the data is usually synthesized as there is no suitable reference method to measure depth with sufficient accuracy. Furthermore, the ground truth labels of a test dataset can be used to quantitatively evaluate the disparity estimation. Usually, ray tracers are used to obtain a physically correct light field rendering and disparity maps of a scene. Most of the available synthetic RGB light field datasets were rendered using Blender with a light field plugin provided by Honauer *et al.* [85]. Whereas Blender provides high photorealism, there does not (yet[1]) exist a multispectral extension of the used ray tracing engine Cycles. The same also holds for other ray tracers that are often used in the computer vision community, such as POV-Ray[2] or the Mitsuba ray tracer [155]. For this reason, the IIIT-RayTracer is used since it is capable of spectral rendering. The IIIT-RayTracer, which is in large parts based on the PBRT

[1] An unofficial fork of Blender has been working on a spectral extension of Cycles since mid-2020. However, there is no stable release yet.

[2] http://povray.org

ray tracer [161], was initially developed by Thomas Nürnberg at IIIT and further extended in the context of this thesis to be able to directly render light fields. The used virtual light field reference camera, which was implemented in joint works with David Uhlig [A11], samples the light field in a fashion geometrically equivalent to a plenoptic camera in the unfocused design. Furthermore, the ray tracer can directly render the ground truth depth, surface normals, or segmentation labels of each light field subaperture. Using the specified camera parameters, the disparity can be calculated from the rendered depth as elaborated in Appendix A.

4.1.1 Dataset properties

The created dataset consists of multispectral light fields rendered from 500 randomly generated scenes as well as seven hand-crafted scenes, which pose specific challenges for the reconstruction task. In this regard, these hand-crafted scenes will be referred to as *challenges* in the following. The light fields are rendered with 16 bit unsigned integer precision and a resolution of $(11, 11, 512, 512, 13)$. The spectrum is sampled from 400 to 700 nm in steps of 25 nm, resulting in 13 spectral channels. This way, the synthetic dataset is sampled in full accordance with the real-world dataset as detailed in Section 4.2. For each light field, the ground truth depth is rendered with 32 bit floating point precision and subsequently converted to the corresponding disparity using the known camera parameters.

To accommodate different camera designs, such as multi-camera arrays or monocular systems, all light fields were rendered in two different camera settings: one corresponding to a plenoptic camera in the so-called unfocused design [154], 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 multispectral light fields including depth and disparity labels were synthesized (not including the dataset challenges). However, for the evaluation presented in this thesis, solely the dataset corresponding to the camera with a focused main lens is used. The camera parameters of the virtual camera were chosen in (rough) accordance with the Lytro Illum camera. This way, disparity ranges that are compatible with the newly created real-world dataset are achieved. Therefore, however, the dataset is not directly suited for large-baseline setups. (However, an adaption to those cases via interpolation and a corresponding disparity scaling should in principle be straightforward.)

The dataset of each camera configuration is split 400 : 50 : 50 into a training, validation, and test dataset. Since the deep learning applications are not trained using the full-sized light fields but smaller patches, the dataset is patched into small light fields with shape $(9, 9, 36, 36, 13)$, which are cropped to $(9, 9, 32, 32, 13)$ during training, validation, and testing. Prior to the development of the proposed methods, it was unclear what angular resolution of the coded light fields would be necessary. Therefore, the light fields are rendered with the comparably large angular resolution of $(11, 11)$. However, in the remainder of this thesis, only the smaller resolution of $(9, 9)$ is used as it turned out to be sufficient for the investigated task. An angular resolution of $(9, 9)$ is commonly used in the light field community and the resulting full-sized light fields are already 1.1 GB in size, since the data is converted to 32 bit for the GPU-based methods, as compared to 13.2 GB necessary for the full angular resolution. This is challenging even for recent GPUs when accounting for the additional need of the network parameters, gradients, and intermediate representations.

In total, the patched dataset consists of roughly 80 000 training, 10 000 validation, and 10 000 test light field patches. The detailed properties are shown in Table 2.1. This is comparable in size to popular deep learning image datasets such as CIFAR-10 and CIFAR-100 [106], containing 50 000 training and 10 000 test images, or MNIST [108], containing 60 000 training and 10 000 test images.

Finally, an RGB conversion of the created dataset and its patched variants are calculated in order to enable a comparison with state-of-the-art reference disparity estimation methods. For each light field, an abstract scene description file is provided that can be used to access the used camera parameters or to render additional ground truth data, such as surface normals, if needed. However, solely the disparity maps are utilized in this thesis. The dataset is made publicly available [A7].

4.1.2 Random scene generation

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, the scenes were created automatically, employing certain geometric constraints. This approach is not new, as the RGB light field datasets by Alperovich *et al.* [4] as well as Heber and Pock [79] also use an automated random scene generation. However, the presented approach differs in some details from both the aforementioned ones.

To achieve diverse geometric properties of the scene, a random number

$$\mathtt{n} = \max\left\{0, \lfloor \mathtt{N} \rfloor\right\}, \quad \mathtt{N} \sim \mathcal{N}(\mu = 28, \sigma = 5)\,, \tag{4.1}$$

of objects are placed in the field of view of the virtual light field reference camera. For these objects, both ideal geometric objects (such as spheres, cones, and planes), as well as 3D mesh models from multiple open-source databases are used, chosen by chance. Depending on the mesh resolution, the models are grouped into three categories: low-, mid-, and high-resolution, assigning lower probabilities to the higher-resolution groups. Furthermore, the maximum number of high-resolution mesh objects per scene is limited to two as they are much more resource-intensive during tracing, leading to significantly longer rendering times.

Unlike the work by Heber and Pock, here the objects are not placed in the scene at three distinct distances (foreground, midground, background). Instead, a disparity range is specified in which the objects are placed uniformly. Note that, due to the inverse relationship between depth and disparity (*cf.* Appendix A), this does not correspond to a uniform distribution of the object depth. To this end, a uniformly distributed disparity $\mathtt{d} \sim \mathcal{U}(-2.5\,\text{px}, 3.0\,\text{px})$ is drawn independently for each object. The corresponding distance from the camera (in the focused configuration) is calculated from the disparity, 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 $d = -2.5\,\text{px}$. Doing so, the background does not possess a constant but slightly varying disparity and possibly a non-trivial curvature, unlike the scenes generated by Alperovich *et al.*

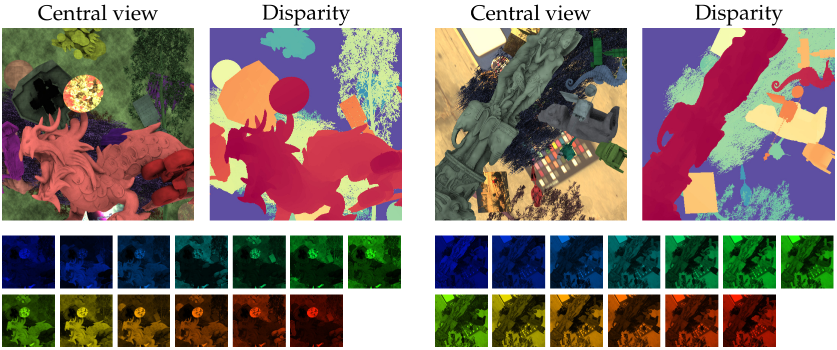


Figure 4.1 Central views of two randomly generated spectral light fields (top: converted to RGB, bottom: colored individual spectral channels) and corresponding central disparity maps. Note that not all details are visible in the disparity maps due to the large range of the colormap and a limited resolution of 8 bit used for this visualization.

In order to obtain diverse spectral properties of the scene, real multispectral images from two datasets [9, 151], freely licensed RGB images (which are spectrally converted by the ray tracer), as well as noise textures with constant random spectra $\mathrm{s} \in [0,1]^{13}$ were used. For the random noise textures, each spectral value $\mathrm{s}_i \sim \mathcal{U}(0,1)$ 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 is argued to be a reasonable mix for machine learning and geometric light field applications.

In all cases, objects are rendered as purely diffuse and highly textured. As almost no reference work regarding (coded) spectral light fields is available, this choice is made in order to not introduce unnecessary additional difficulties. However, in principle it is straightforward to render specular, reflective, and/or untextured objects. The dataset was rendered on a large-scale computing cluster. Each light field was rendered using 10 cores and 32 GB of a shared computing node. The rendering took 4 to 30 h per light field, depending on the scene complexity and in particular the resolution of the used 3D models.

The central views and disparities of two randomly generated light fields are shown in Figure 4.1. In the remainder of this thesis, the colorbar of all disparity plots is not explicitly denoted to not clutter the presentation. In all cases, the colorbar shown in Figure 4.6 is used, scaled to the minimum and maximum disparity value of the corresponding ground truth data as denoted in Table 4.1 for all considered datasets.

It should be noted that these randomly generated scenes are not semantically meaningful or natural. That is, objects can be arbitrarily oriented, scaled, or even overlapping. Therefore, the dataset statistics are in general assumed to be quite different from a real-world light field dataset. In particular, the dataset is not useful for applications in which a natural context is of interest, *e.g.* semantic segmentation or object detection. However, in the considered case, the most important feature of the dataset is its correct light field geometry, which the subsequent methods are relying on, as well as the spectral properties, and not the semantic context.

To visualize the properties of the training dataset, its disparity and spectral distributions are compared to two reference datasets. As there is no spectral light field dataset available, the spectral distribution is compared to reference multispectral image datasets and the disparity distribution is compared to reference RGB light field datasets.

First, to compare the spectral distribution, several datasets of multispectral images [9, 151, 225], which were re- and downsampled to the 13 considered spectral channels, are used, which are collectively referred to as the "reference" dataset. Note that images of two of these datasets have also been used as multispectral textures in the rendering of the created dataset. Both the new 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 distributions of the two datasets are shown in Figure 4.2. The created dataset shows a more balanced spectral distribution than the reference dataset, especially at lower spectral indices (corresponding to smaller wavelengths). While there is no point in arguing which distribution is better suited for data-driven applications, it does reflect the design choices made upon the random scene generation. However, a peak at intensity values of one can be observed, likely stemming from overexposed regions of the used RGB image textures. Second, to compare the dataset's disparity distribution, a dataset composed of previously published RGB light field datasets containing disparity ground truth [4, 79, 85], which are combined into a single dataset, is considered. Again, this composed RGB light field dataset is referred to

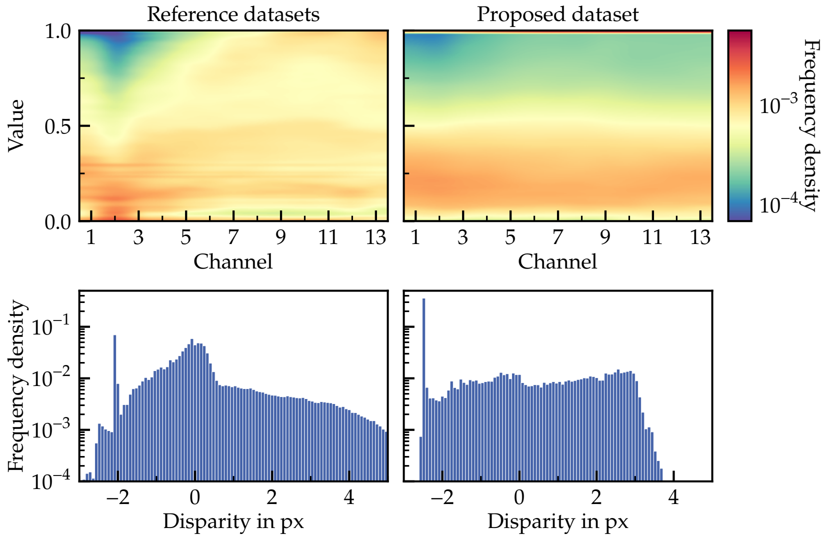


Figure 4.2 Spectral distribution (top) and disparity distribution (bottom) of the reference datasets (left) and the created synthetic training dataset (right). The spectral 2D histogram is interpolated for visualization.

as the “reference” dataset. The corresponding histograms are shown in Figure 4.2. While the created 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. Overall, the newly created synthetic training dataset shows the desired properties.

4.1.3 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 as discussed in Section 4.3) 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

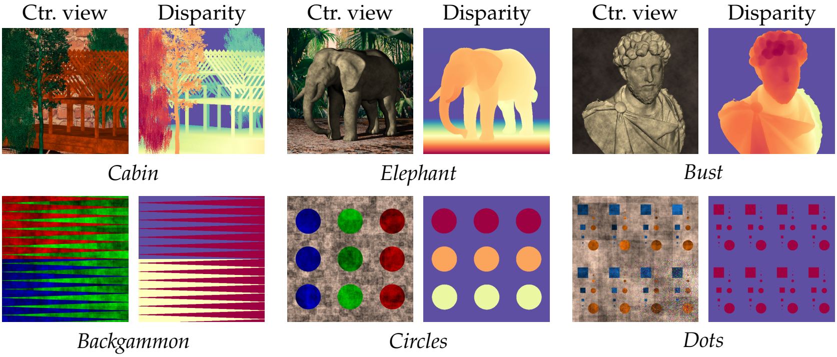


Figure 4.3 Spectral central views (converted to RGB) and corresponding central disparity maps of the dataset challenges. The disparity is shown with ranges as given in Table 4.1(a).

patched into smaller sizes for training and testing. Therefore, seven handcrafted scenes, so-called *challenges*, were created and rendered together with their respective ground truth disparities. These scenes are used to further quantitatively and visually compare the obtained results, using the full resolution of $(9, 9, 512, 512, 13)$. Moreover, the challenges may be used to assess the performance with respect to a specific challenging aspect such as occlusion, shadow, detail, or noise. The created challenges consist of the following scenes: *Cabin*, *Elephant*, *Bust*, *Backgammon*, *Circles*, *Dots*, and *Wall*, the first six of which are shown in Figure 4.3. The reader familiar with the HCI benchmark dataset [85] will notice some similarities. 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 contained in the HCI benchmark dataset.

While the first three challenges use high-resolution 3D mesh models and show a realistic scene geometry, the latter ones are purely synthetic, utilizing ideal geometric shapes. The three natural scenes *Cabin*, *Elephant*, and *Bust* roughly emphasize occlusion, shadow, and detail, respectively. On the other hand, the purely synthetic ones are designed to each assess a very specific aspect. The *Backgammon* scene consists of two flat surfaces at disparity −1 px and 0 px, as well as an occluding foreground at a disparity of 1 px with varying local width. The *Circles* scene consists of three groups of three circles—one red, one green, one blue—at disparities

−1 px, 0 px, and 1 px, respectively. This allows for a visual assessment of the reconstruction quality with respect to a spectral as well as a depth dependence. The scene *Dots* is superposed with independent Gaussian noise whose variances differ 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, overall distorting the light field geometry and posing difficult challenges on the reconstruction and disparity estimation. The last scene, *Wall* (which is not shown), consists of a flat surface of constant disparity with a multispectral image texture. This scene was then rendered at different disparities, ranging from −1.5 to 1.5 px in steps of 0.25 px which is used to quantitatively compare multispectral-related performance versus disparity. Hence, strictly speaking, the *Wall* challenge consists of 13 individually rendered light fields and their corresponding disparity.

4.2 Real-world dataset

To investigate the proposed reconstruction approaches also for real-world spectral light fields, a suitable dataset is needed. To the author's knowledge, the only public spectral light field dataset was created by Xiong *et al*. [221]. This dataset consists of three spectral light fields, *Boards*, *Toys*, and *Fruits*, with a resolution of $(11, 11, 270, 360, 25)$, captured using a gantry-mounted 2D spectrometer. These light fields are sampled from 450 to 690 nm in steps of 10 nm. While the imaging setup allows for a high spatial resolution and quality, the objects and background used in the dataset are not all textured, which is not ideal for the evaluation of the disparity estimation. Furthermore, the dataset is not properly white-balanced and shows some brightness flickering across the individually captured subapertures, indicating poor calibration. Overall, the dataset by Xiong *et al*., as shown in Figure 4.4, is not well suited for the evaluation performed in this thesis. However, the results from all evaluations presented in this thesis are also available for this dataset within the digital supplement. For this, the dataset is downsampled to 13 spectral channels and centrally cropped to a resolution of $(9, 9, 256, 360, 13)$ to be compatible with all investigated methods.

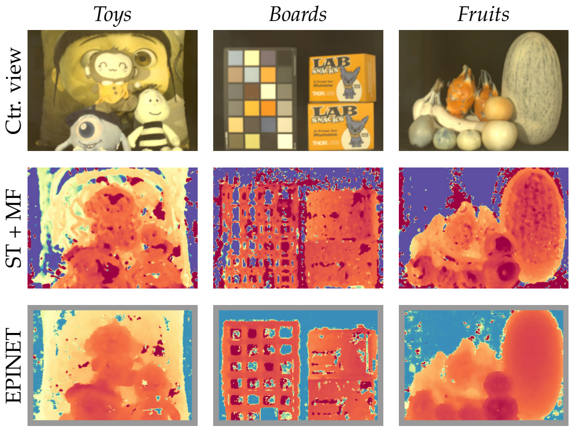


Figure 4.4 The three scenes from the spectral light field dataset by Xiong *et al.* [221]. White-balancing was performed using the white square of the color calibration chart. The disparity estimation from the RGB-converted light fields using the structure tensor (ST) with TV-L1 fusion and median filtering (MF) [213], as well as using EPINET [176] are shown using the disparity ranges given in Table 4.1(c).

To overcome the limitations of the dataset by Xiong *et al.*, a new spectral light field dataset was recorded using a custom-built spectral light field camera. The developed camera, based on a Lytro Illum light field camera and a spectral filter wheel holding 13 spectral bandpass filters, and its calibration are elaborated in detail in Sections 4.2.1 to 4.2.3. In total, three different scenes were captured—each using two different main lens focal lengths. To reduce the influence of the angular dependence of the interference filters, and to increase the disparity range, the scenes were captured using the comparably large 150 mm and 250 mm main lens focal length equivalents of the Lytro Illum camera. Since the disparity is proportional to the inverse of the depth, the disparity sensitivity is much higher for objects that are close to the camera. For the region beyond the focal plane, the sensitivity is comparably low (*cf.* Appendix A). Therefore, the main lens focal plane was placed roughly in the last third of the range between the first object and the background such that most of the imaged objects were contained within the high-sensitivity region.

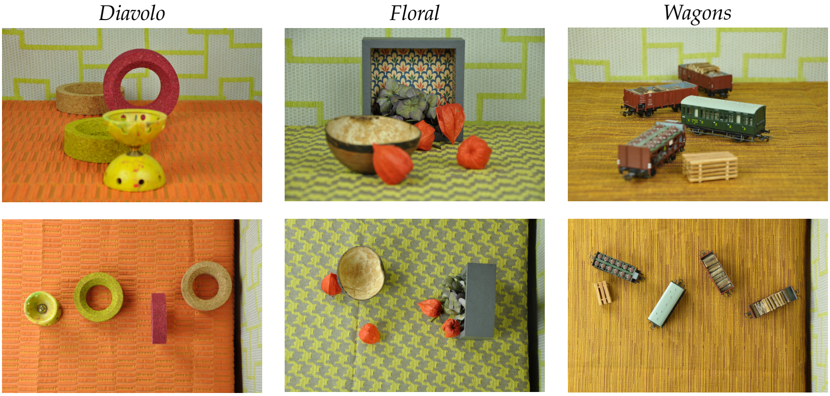


Figure 4.5 Front and top view of the captured scenes. Here, the images were taken using a conventional RGB DSLR camera.

The captured scenes were created with spectral variance and disparity estimation in mind. Therefore, highly textured, colorful backgrounds and scene objects were used, and the objects were placed as spread out as possible to obtain a diverse disparity distribution. In fact, for the shorter focal length equivalent of 150 mm, the objects were even further spread out, as compared to the setup used with the 250 mm focal length equivalent, resulting in slightly different arrangements than those used for the larger focal length. A front and top view of the captured scenes are given in Figure 4.5.

In the large-focal length setting used here, the f-number matching of the Lytro Illum camera is suboptimal, leading to a smaller angular resolution than with shorter focal lengths. Hence, the angular resolution of the captured light fields is centrally cropped to $(9, 9)$ which was found to still give reliable results. Spatially, the light fields are cropped to $(400, 400)$ to be compatible with all investigated methods—in particular the downsampling-based deep learning approach, and the used atom size in the case of the dictionary-based methods. Hence, the final resolution of the captured decoded light fields is $(9, 9, 400, 400, 13)$. These spectral light fields are reference light fields, *i.e.* they are not spectrally coded and can therefore be used to quantitatively evaluate the investigated reconstruction methods. To this end, the spectral coding is

Table 4.1 Disparity ranges (in px) of the created synthetic dataset challenges, the new real-world dataset captured at two different focal length equivalents F (in mm), and the dataset published by Xiong *et al.* [221].

(a) Disparity ranges of the created dataset challenges.

Scene	Disparity	
	$d_{\min}$	$d_{\max}$
Backgammon	−1.0	1.0
Bust	−2.2	1.4
Cabin	−2.4	2.8
Circles	−2.5	1.0
Dots	−0.5	0.5
Elephant	−0.9	0.6
Wall	−1.5	1.5

(b) Disparity ranges of the created real-world dataset.

F	Scene	Disparity	
		$d_{\min}$	$d_{\max}$
150	*Diavolo*	−0.2	1.8
	Floral	−0.2	1.5
	Wagons	−0.2	1.9
250	*Diavolo*	−0.3	2.3
	Floral	−0.3	2.0
	Wagons	−0.3	2.3

(c) Disparity ranges of the dataset by Xiong *et al.*

Scene	Disparity	
	$d_{\min}$	$d_{\max}$
Boards	−3.0	1.2
Fruits	−3.0	1.2
Toys	−3.0	1.2

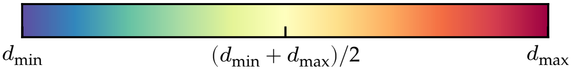


Figure 4.6 The disparity colormap used throughout with ranges given in Table 4.1.

performed in the digital instead of the optical domain, as discussed in Section 4.4.2. However, due to the used coding scheme, which allows for standard decoding of the raw sensor measurements as discussed in Section 2.3.1, the coding in the digital domain can be considered equivalent to the optical coding using a coded MLA. The dataset is made publicly available [A2].

As is common with real-world light field datasets, the captured dataset does not contain any depth or disparity reference. Hence, it cannot be used to *quantitatively* evaluate disparity estimation performance, however a qualitative comparison can of course be performed. Moreover, using the uncoded light fields, conventional state-of-the-art disparity estimation methods can be used as a reference, *e.g.* by converting the light fields to RGB. A comparison of the properties of the created synthetic and real-world datasets is given in Table 2.1.

4.2.1 Spectral light field camera

The spectral light field camera, developed and manufactured in joint works with the IIIT Mechanical Workshop, consists of a conventional Lytro Illum light field camera and a custom-built housing, enclosing a spectral filter wheel. Hence, the spectral light field is not captured at once but in a spectrally scanning fashion. The wheel holds 13 spectral bandpass filters with central wavelengths ranging from 400 to 700 nm in 25 nm steps, which also corresponds to the filter width. In this configuration, the light fields are spectrally sampled in complete analogy to the created synthetic dataset. The used filter transmissivities are depicted in Figure 4.8. To step the filter wheel automatically, the wheel is flange-mounted onto a stepper motor. The camera and stepper motor are synchronously controlled by a Raspberry Pi. For interaction, a Python-based backend as well as a web application GUI was developed. Further technical details of the camera are presented in Appendix C. An overview of the camera is given in Figure 4.7.

Since the Lytro Illum camera uses a Bayer pattern-based RGB sensor, the radiometric calibration of the camera is rather challenging and is presented in detail in Section 4.2.2. For the illumination of the scenes, two standard photo studio lights were used as shown in Figure 4.9. Each light consists of seven 50 W daylight lamps with a color temperature of 5500 K. These lights offer the advantage of providing a comparably strong, even, and diffuse illumination while being relatively inexpensive. However, in the case of spectral imaging, the black body-like emitted spectral flux of the lamps is not optimal because of a relatively low flux at low wavelengths (the blue and UV range) as well as large wavelengths (the red and NIR range). Therefore, the spectral channels of the green range will saturate much quicker than those of the blue and red range, assuming a constant exposure time. This effect is further amplified by the quantum efficiency of conventional CCD sensors, which usually follow the same trend, *i.e.* a relatively lower quantum efficiency at the blue and red as compared to the green wavelength range. Unfortunately, the quantum efficiency of the Lytro Illum camera is not known precisely. Approaches to overcome these challenges are discussed in the following.

Arguably, despite its high sensor resolution of 41 Mpx, the Lytro Illum camera is not the ideal camera for spectral light field imaging. First,

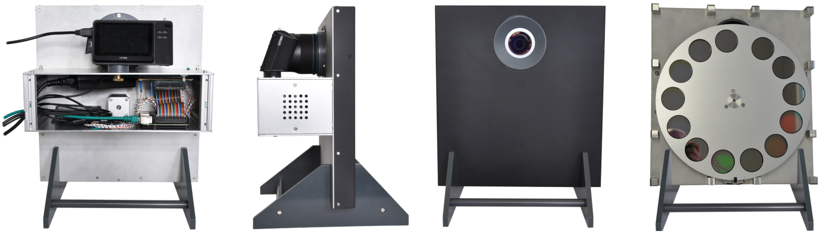


Figure 4.7 Custom-built spectral light field camera, using a Lytro Illum camera and a spectral filter wheel, jointly controlled by a Raspberry Pi.

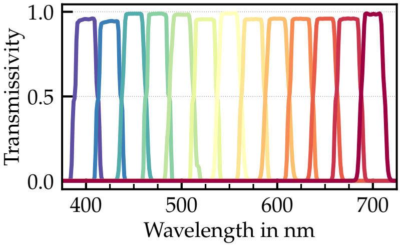


Figure 4.8 Filter transmissivities of the 13 used spectral bandpass filters according to manufacturer measurements.

Figure 4.9 The studio lights used for illumination while capturing the created real-world dataset.

the Lytro Illum camera was developed as a consumer camera. Hence, low-level interfacing (*e.g.* for remote triggering or camera control) is not as straightforward to achieve as with an industry-grade camera. All control of the camera is done via the Android Debug Bridge, which offers limited feedback and error handling. Also, it is not possible to extract a full light field from the measurements using the proprietary software provided by Lytro, only the super-resolved or the refocused images are available. Hence, the camera calibration, as well as the light field decoding, have to be implemented from scratch. Second, more severely, the Lytro Illum is a color camera, *i.e.* it employs additional RGB filters on the sensor, requiring extensive radiometric calibration which is further hampered by the non-detachable main lens, preventing direct access to the sensor. However, no off-the-shelf alternatives are available. While Raytrix offers monochromatic light field cameras, the optical design (and hence the light field coding) differs significantly from the Lytro's (*cf*. Section 2.1). Hence, Raytrix cameras are not suitable for the considered case. Alternatively, using a 2D spectrometer mounted on a gantry or robotic arm would be a good choice to capture high-quality spectral light fields, similar to the approach by Xiong *et al*. [221]. However, building such a reference camera is arguably much more expensive. Therefore, the Lytro Illum camera is chosen, shifting the challenges from the hardware to the software domain—in particular the radiometric calibration. Considering the hardware used, it is fair to argue that the used spectral light field imaging setup is among the least expensive ones possible.

4.2.2 Radiometric calibration

The spectral light field is measured in a spectrally scanning fashion where for each spectral channel a monochromatic light field is captured by the Lytro Illum camera, employing the corresponding spectral bandpass filter. Usually, the individual channels of a spectrally scanning camera are obtained using identical exposure times. However, this results in a strongly channel-dependent SNR due to the different spectral sensitivities, which are determined by the filter characteristics, the quantum efficiency of the sensor, as well as the light source. As previously noted, this effect is quite severe in the considered case due to the used photo studio illuminants, as well as the quantum efficiency and the additional RGB filters of the Lytro Illum camera. To overcome this problem, it is proposed to capture the spectral channels with their individually optimal exposure time. That is, for every imaged scene, first a channel-wise exposure time is determined to be as large as possible while avoiding oversaturation (*i.e.* clipping). In the presented case, this results in comparably short exposures (about 1 ms) in the highly sensitive green channels and comparably long exposures (up to 6 s) in the blue and red channels. This approach has the drawback that the camera needs to be radiometrically calibrated in order to scale the individual measurements to a common reference intensity, which is more cumbersome than the regular white balancing and de-vignetting. However, the radiometric calibration has the advantage that the additional Bayer pattern RGB filters that are present in the Lytro

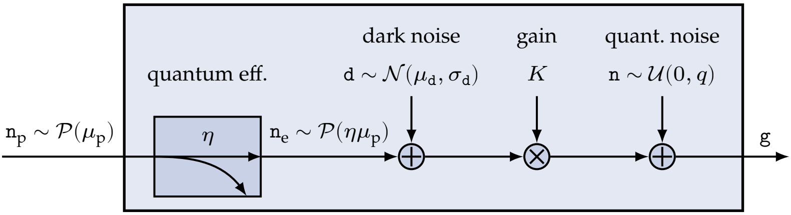

dark noise
gain
quant. noise
quantum eff.
$\mathtt{d} \sim \mathcal{N}(\mu_\mathtt{d}, \sigma_\mathtt{d})$
K
$\mathtt{n} \sim \mathcal{U}(0, q)$
$\mathtt{n_p} \sim \mathcal{P}(\mu_\mathtt{p})$
η
$\mathtt{n_e} \sim \mathcal{P}(\eta\mu_\mathtt{p})$
$\mathtt{g}$

Figure 4.10 Linear camera model according to the EMVA 1288 standard.

Illum camera, as well as the vignetting of the (non-detachable) main lens and the microlenses can be calibrated at once.

The radiometric calibration is performed using the linear camera model as specified by the European Machine Vision Association (EMVA) 1288 standard, which is depicted in Figure 4.10. Note that, since the main lens of the Lytro Illum camera is non-detachable, an absolute calibration of the sensor according to the standard is not possible here (nor is it necessary).

First, the camera's dark signal properties are estimated. Following the EMVA 1288 standard, the mean value of the dark signal $\mathtt{d}$ is given by

$$\mu_\mathtt{d}(t) = \mu_{\mathtt{d},0} + \mu_\mathtt{I}\, t\,. \tag{4.2}$$

The mean dark signal $\mu_\mathtt{d}$ is linear in the exposure time t with offset $\mu_{\mathtt{d},0}$ and slope $\mu_\mathtt{I}$ which is called the (mean) dark current. Since it is caused by thermally induced electrons, the dark current is temperature-dependent. Therefore, the calibration is only valid at a given temperature. For this reason, the calibration measurements and the dataset were captured on consecutive days of similar temperatures in the winter where the temperature was assumed to be approximately constant due to the constant heating of the laboratory. The offset and slope are estimated by measuring an exposure series of dark images, *i.e.* images without any illumination and varying exposure time. Here, an exposure series of five exposures, ranging from 4 to 10 s, was acquired. The mean values $\mu_\mathtt{d}(t)$ are approximated using the sample mean with a sample size of 10, *i.e.* 10 dark images were measured and averaged for each exposure time. The mean offset and dark current are estimated via a simple linear least-squares regression of the averaged exposure series using (4.2). Note that this calibration would have to be performed individually for every used camera gain. For this reason, the camera gain is set to ISO 80 throughout all measurements in the following. As only static scenes are imaged (and hence long exposure times are unproblematic), the value was chosen to be as small as possible in order to keep the sensor noise at the lowest possible value. Additionally, this minimizes the overall impact of the temperature dependence of the dark signal.

In order to radiometrically calibrate the camera, an exposure series of spectral bright images was collected. Here, a bright image refers to a spectral image of the used light source taken by the spectral light field camera through an optical diffuser, achieving an almost ideal diffuse white scene. The diffuser is positioned directly in front of the spectral filters. For a total of 30 different exposure times, ranging from 0.25 to 200 ms, five spectral bright images were measured and averaged. Hence, a total of 1950 greyscale images were captured for the bright image series. Note that, since the main lens is non-detachable, the following depends on the actual camera configuration, *i.e.* the camera gain and in particular the zoom and focus settings. Changing the zoom or the focus of the main lens changes the local intensity distribution of the imaged radiation and hence the effective spectral responsivity of the individual pixels. For this reason, the measurements of the spectral bright images and the radiometric calibration were repeated for the different camera settings that were used to capture the dataset.

Following the EMVA 1288 standard, each pixel is assumed to obey the linear camera model

$$\begin{aligned} \mu_{\mathtt{g}} &= \mu_{\mathtt{d}} + K\eta \frac{\lambda A}{hc} E t \\ &= \mu_{\mathtt{d},0} + \left(\alpha(\lambda) + \mu_{\mathtt{I}}\right) t \,. \end{aligned} \tag{4.3}$$

Here, g corresponds to the stochastic pixel grayscale value. The sensor's quantum efficiency is denoted by η and the system gain by K. The term $\frac{\lambda A}{hc} E$ corresponds to the mean number of photons reaching the pixel (depending on the light source and the used lenses) and is unified in the scalar $\alpha(\lambda)$, which incorporates all spectral characteristics of the light source. Overall, the mean pixel grayscale value is linear in the exposure time. Now, in the considered case, the camera employs additional spectral filters as well as a Bayer-pattern RGB sensor. Since the filters have the same effect on the incoming photons as the sensor quantum efficiency, *i.e.* a photon is either transmitted with a certain probability or not, the filters can be multiplicatively[3] incorporated into (4.3) via

$$\begin{aligned} \mu_{\mathtt{g}} &= \mu_{\mathtt{d},0} + (b_n\,\varphi_\lambda\,\alpha_\lambda + \mu_{\mathtt{I}})\,t\,, && (4.4)\\ n &= 1,2,3\,, && \text{(RGB filter)}\\ \lambda &= 1,2,\dots,13\,, && \text{(spectral filter)} \end{aligned}$$

where b_n denotes the effective RGB filter and φ_λ the spectral filter transmissivity. Therefore, for an arbitrary spectral pixel (x, y, λ), denoting the two spatial and one spectral coordinate of the raw measurement, the linear model can be written in a tensor-like fashion,

$$\mu_{\mathtt{g},xy\lambda i} = \mu_{\mathtt{d},0} + (b_{n(x,y)}\,\varphi_\lambda\,\alpha_\lambda + \mu_{\mathtt{I}})\,t_i\,, \quad \text{or} \tag{4.5}$$

$$\tilde{\mu}_{\mathtt{g},xy\lambda i} := \mu_{\mathtt{g},xy\lambda i} - \mu_{\mathtt{d},0} - \mu_{\mathtt{I}}\,t_i = b_{n(x,y)}\,\varphi_\lambda\,\alpha_\lambda\,t_i\,, \tag{4.6}$$

where $i = 1, \dots, 30$ denotes the index of the exposure time series. The index $n(x, y)$ of each RGB filter $b_{n(x,y)}$ is uniquely determined by the pixel (x, y) and is neglected in the following. In principle, (4.6) can be used to calibrate each pixel individually, viewing $c_{xy\lambda} = b_n\,\varphi_\lambda\,\alpha_\lambda$ as the single model parameter. In fact, this simple linear least-squares approach can be analytically solved for $c_{xy\lambda}$. However, this resulted in highly noisy light fields in precursory experiments. This is not too surprising, as in some cases only a few (below five) measurements are available per pixel, depending on its sensitivity, due to overexposure as discussed shortly. For a more parameter-efficient and smooth estimate, it is proposed to factorize the linear dependence into its spatial and spectral components,

$$\tilde{\mu}_{\mathtt{g},xy\lambda i} = v_{xy} r_\lambda^{(n)}\,t_i\,. \tag{4.7}$$

Here, v_{xy} denotes all spatial dependencies such as the natural and mechanical main lens and microlens vignetting, and r_λ denotes the spectral responsivity, which depends on the pixel's color filter type $n \in \{\mathrm{R}, \mathrm{G}, \mathrm{B}\}$ and the used bandpass filter λ. The goal is now to estimate v_{xy} and $r_\lambda^{(n)}$

[3] This is a direct result of the so-called thinning property of the Poisson process when composed with a Bernoulli experiment.

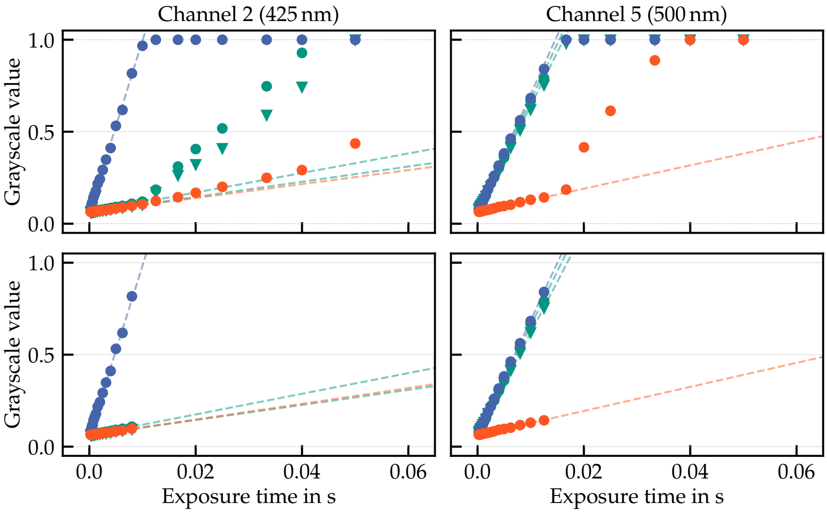
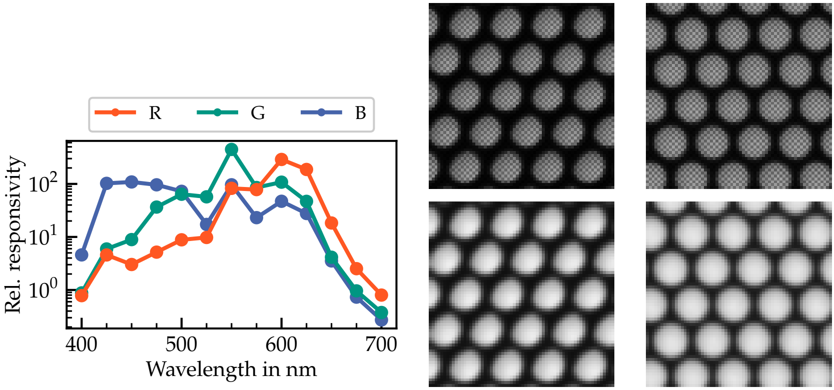


Figure 4.11 Greyscale values of a 2×2 px crop from a spectral bright image exposure series. The individual measurements are colored corresponding to their respective RGB filter type. Top row: unfiltered exposure series with a linear fit of the first 10 values for reference. Bottom row: filtered with neighbor overexposure compensation and linear fit of the full data.

from the measured spectral bright image exposures series. Doing so, the model parameters v_{xy} and r_λ are jointly estimated from all available measurements and not just a single pixel. Since the relative constant scaling between v_{xy} and $r_\lambda^{(n)}$ is arbitrary, it is fixed here by assuming $v_{xy} \in [0, 1]$ which interprets the vignetting as a form of attenuation.

In order to be able to process the full spectral exposure series tensor $\tilde{\mu}_{\mathrm{g},xy\lambda i}$ at once, which makes GPU acceleration (*e.g.* via PyTorch or TensorFlow) straightforward, certain care has to be taken to mask out overexposed pixels. Since the full exposure series is measured for all spectral channels λ, overexposure is unavoidable. For example, the green channels are much more sensitive than the red or blue channels and will therefore saturate much more quickly, while longer exposure times are necessary to achieve a reliable estimate of the less sensitive channels. However, overexposure is easy to handle by masking out all pixels with pixel values larger than 0.985, which corresponds to the threshold given



(a) Estimated spectral responsivity $\hat{r}_\lambda^{(n)}$.

(b) Estimated vignetting $\hat{v}_{xy}$ (bottom) and a raw bright image for reference (top).

Figure 4.12 Spectral responsivity and vignetting (top left and central crop) estimated from a spectral bright image exposure series in the case of the 250 mm main lens focal length equivalent (and a fixed focus setting).

by the four least significant bits of the 10 bit sensor[4] of the Lytro Illum camera. However, this simple masking is not sufficient. As shown in Figure 4.11, overexposed pixels influence neighboring pixels. In CCD sensors, the charges from saturated pixels overflow to neighboring pixels, in particular along the line at which the CCD sensor is readout. In the top left of Figure 4.11 one can observe that the blue pixels saturate first, leading to a change of the green pixels' sensitivity, since they are now also registering the overflown charges from the blue pixel. This is referred to as *blooming* in CCD sensors [18]. However, the red pixels' sensitivity only changes slightly. This is likely because in a standard Bayer pattern the red and blue pixels are only neighbored diagonally, whereas the green pixels are direct neighbors to both blue and red pixels. So when also the green pixels saturate, the red pixels' sensitivity changes abruptly. To obtain a reliable estimate of the true sensitivity, it is clear that these measurements also have to be masked out. This is achieved by using the available

[4] $(1111110000)_2/(1111111111)_2 = 1008/1023 \approx 0.985$.

saturation mask and extending it to direct neighbors as well as neighbors in the 5 px neighborhood along the line along at which the CCD sensor is readout. The filtered results are shown in Figure 4.11 (bottom). Furthermore, since the exposure times of the camera are logarithmically more densely sampled at short exposure times, it is proposed to weigh the measurements logarithmically. Finally, an estimate of the vignetting and responsivity is obtained using the model (4.6) and a weighted least-square fit. Since the model parameters are coupled, an analytic solution is not available. Instead, the weighted mean squared error is minimized using PyTorch and the Adam optimizer [103]. An example result of the fitted vignetting and responsivity is given in Figure 4.12. Since the full spectral exposure series cannot be loaded onto the GPU at once (a single spectral bright image exposure series is about 65 GB in size), the measurements are spatially patched using 64 windows with 50 % overlap. In overlapping regions, the obtained estimates are averaged.

Now, using the estimated dark offset, dark current, vignetting, and spectral responsivity, a raw spectral light field measurement $\tilde{\boldsymbol{\mathcal{R}}}_{xy\lambda}$, where each spectral channel λ is measured using its optimal exposure time t_λ, is calibrated, in accordance with (4.6), via

$$\boldsymbol{\mathcal{R}}_{xy\lambda} = \frac{\tilde{\boldsymbol{\mathcal{R}}}_{xy\lambda} - \hat{\mu}_{\mathrm{d},0}}{\left(\hat{v}_{xy}\hat{r}_{\lambda}^{(n)} + \hat{\mu}_{\mathrm{I}}\right) t_\lambda} \, . \tag{4.8}$$

From the spectrally calibrated raw measurement, the spectral light field is decoded as follows.

4.2.3 Geometric calibration

Each raw measurement corresponds to a 2D multiplexed version of the 4D light field. In order to decode the measurements, as will be discussed in more detail in Section 4.2.4, the camera has to be calibrated. In particular, as the basis of virtually all model-based calibration methods, the individual microlens centers have to be detected. Usually, a regular grid is then estimated that best approximates the detected centers. Here, these calibration steps are referred to as *pre-calibration*. A full calibration of the remaining intrinsic and extrinsic parameters is not discussed here as there are several established and well-tested approaches [21, 40, 45, 84].

In spite of the importance of this pre-calibration, the literature focuses mostly on the camera models and decoding but pays little or no attention to the necessary details emerging in the pre-calibration, most importantly non-trivial effects such as mechanical and natural vignetting. While for a correct pre-calibration, the detection of the perspectively projected microlens centers is necessary, as discussed shortly, all methods proposed in the literature rely on estimating the center of each microlens image brightness distribution, approximating the orthogonally projected centers. Due to natural and mechanical vignetting, this results in severe deviations from the true projected centers, in particular in off-center microlenses.

4.2.3.1 Camera model

The pre-calibration of MLA-based cameras is usually performed using so-called white images—images of a white scene, for example taken using an optical diffuser. As opposed to the spectral bright image exposure series used for the radiometric calibration, the white images are not taken through the additional spectral filters, as the lens geometry remains unchanged in that case. A single white image per camera configuration, *i.e.* zoom and focus, is sufficient. To increase the robustness against sensor noise, a single white image is obtained as the mean of 10 individual measurements here.

To estimate the grid parameters of the MLA, the previously introduced camera model is slightly generalized to the one depicted in Figures 4.13 and 4.14. In this model, the camera consists of a main lens and a collection of microlenses, arranged in a hexagonal grid, which may be rotated (not depicted in the figures) and tilted. All lenses are modeled as thin lenses. As is usual in the focused design, f-number matching of the main lens and microlenses is assumed [154]. To model irregularities of the grid, independent uncorrelated Gaussian noise ϵ is added to the ideal grid point coordinates. Finally, an object-side aperture with variable entrance pupil is placed at a distance a to the main lens to account for mechanical vignetting effects.

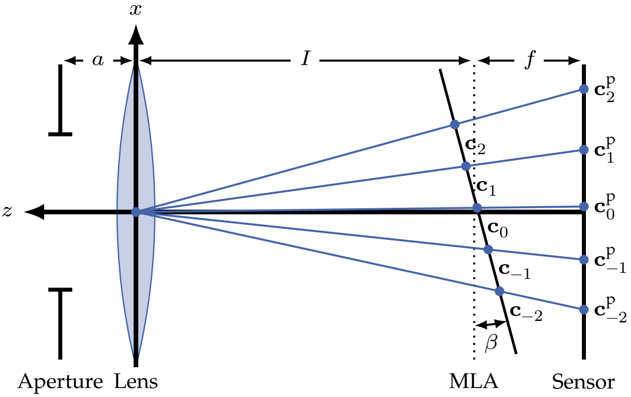

x
a
I
f
z
$\mathbf{c}_2$
$\mathbf{c}_1$
$\mathbf{c}_0$
$\mathbf{c}_{-1}$
$\mathbf{c}_{-2}$
$\mathbf{c}^{\mathrm{p}}_2$
$\mathbf{c}^{\mathrm{p}}_1$
$\mathbf{c}^{\mathrm{p}}_0$
$\mathbf{c}^{\mathrm{p}}_{-1}$
$\mathbf{c}^{\mathrm{p}}_{-2}$
β
Aperture
Lens
MLA
Sensor

Figure 4.13 Schematic side view of the unfocused light field camera model with MLA tilt and perspectively projected microlens centers.

In the case of a hexagonal microlens arrangement, the ideal unrotated, untilted and unshifted microlens center coordinates are given by

$$\mathbf{c}^{\mathrm{id}}_{i,j} = \mathbf{o}_{\mathrm{g}} + \begin{pmatrix} \left(i + \frac{1}{2} j \bmod 2\right) d_x \\ j d_y \\ 0 \end{pmatrix} + \boldsymbol{\epsilon}_{i,j}\,, \tag{4.9}$$

for $(i,j) \in \mathbb{Z}^2$. Here, d_x and d_y denote the horizontal and vertical ideal grid spacing, $\mathbf{o}_{\mathrm{g}} = (o_{\mathrm{g},x}, o_{\mathrm{g},y}, 0)^{\mathrm{T}}$ the grid offset, and $\boldsymbol{\epsilon} = (\epsilon, \epsilon, 0)^{\mathrm{T}}$ the grid noise with standard deviation σ_{g}. The ideal hexagonal grid is determined by a single grid spacing d via

$$d_x = d\,, \quad \text{and } d_y = d \cdot \sqrt{3}/2\,. \tag{4.10}$$

The microlens radius is given by $r = d/2$. The ideal grid points are then rotated in the xy-plane by α, rotated around the y-axis by β, rotated around the x-axis by γ, and shifted to $z = -I$, where I denotes the imaging distance with respect to the main lens. Hence, one obtains the final grid point coordinates

$$\mathbf{c}_{i,j} = \mathbf{R}_{x,\gamma}\,\mathbf{R}_{y,\beta}\,\mathbf{R}_{z,\alpha}\mathbf{c}^{\mathrm{id}}_{i,j} + (0,0,-I)^{\mathrm{T}}\,. \tag{4.11}$$

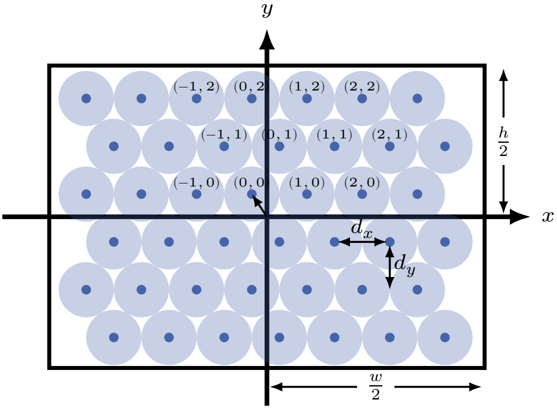


Figure 4.14 Schematic front view of the MLA in the case of a hexagonal layout together with the used microlens indexing scheme.

At times, the grid point coordinates will be simply referred to as $\mathbf{c}_k$ for $k \in \mathbb{Z}$, when one does not need to specify the re-indexing $(i, j) \mapsto k$. The size (w, h) of the MLA is chosen such that the projection of the grid, after rotation α and tilt (β, γ), covers the full sensor of size (s_x, s_y), *i.e.* it can be calculated via

$$\mathbf{R}_{x,\gamma}\mathbf{R}_{y,\beta}\mathbf{R}_{z,\alpha}\begin{pmatrix} w \\ h \\ 0 \end{pmatrix} = \begin{pmatrix} s_x \\ s_y \\ z \end{pmatrix}, \tag{4.12}$$

where z is arbitrary. The perspective projection of the microlens centers (4.11) from the center $(0, 0, 0)^{\mathrm{T}}$ of the exit pupil onto the sensor is given by

$$\mathbf{c}^{\mathrm{p}}_{i,j} = \zeta_{i,j}\mathbf{c}_{i,j} \tag{4.13}$$

with scaling factor $\zeta_{i,j}$ such that

$$\left(\mathbf{c}^{\mathrm{p}}_{i,j}\right)_z = \left(\zeta_{i,j}\mathbf{c}_{i,j}\right)_z = -I - f, \tag{4.14}$$

where f denotes the ideal microlens focal length. Therefore, using (4.11), one obtains

$$\zeta_{i,j} = \frac{-I-f}{\left(\mathbf{R}_{x,\gamma}\,\mathbf{R}_{y,\beta}\,\mathbf{R}_{z,\alpha}\mathbf{c}^{\mathrm{id}}_{i,j}\right)_z - I} \,. \tag{4.15}$$

The orthogonally projected centers $\mathbf{c}^{\mathrm{o}}_k$ are simply obtained from the grid coordinates $\mathbf{c}_k$ by setting their z-value to $(-I-f)$.

4.2.3.2 Microlens array accuracy estimates

To simplify some of the model parameters, the following estimates are considered. Assuming an ideal grid, $0 = \alpha = \beta = \gamma$, the focal length f_k of a microlens has to be accurate within

$$\Delta_f < pf/d \tag{4.16}$$

such that the disk of confusion lies within a pixel with pixel pitch p [154]. Deviations from this constraint will lead to blur in the decoded image which cannot be compensated. Following the same argument, the rotation α and tilt (β, γ) have to be constrained such that the maximum change in distance Δ_z to the sensor fulfills the same restriction. To estimate this, the outermost point $(w/2, h/2, 0)^{\mathrm{T}}$ of the MLA is used and rotated and tilted via $\mathbf{R}_{x,\gamma}\,\mathbf{R}_{y,\beta}\,\mathbf{R}_{z,\alpha}$. The resulting z-component then yields the maximum change of distance of the MLA to the sensor. Using (4.12), one finds

$$\Delta_z = s_x(\,\tan\beta/\cos\gamma) + s_y \tan\gamma < pf/d \,. \tag{4.17}$$

Note that the result does not depend on the rotation α. To obtain a common upper bound Δ_δ for the accuracies of the tilt angles, a Taylor series expansion in $\beta, \gamma = 0$ is performed,

$$\Delta_z \approx s_x\left(\beta + \beta^3/3 + \beta\gamma^2/2\right) + s_y\left(\gamma + \gamma^3/3\right) \,. \tag{4.18}$$

Inserting $\beta = \gamma \equiv \Delta_\delta$, one can solve for

$$\Delta_z \approx \Delta_\delta(s_x + s_y) + \Delta_\delta^3(5s_x/6 + s_y/3) < pf/d \,. \tag{4.19}$$

Additionally, the tilt introduces geometric distortions in the perspectively projected grid. That is, the regular grid $\{\mathbf{c}_k \colon\ k \in \mathbb{Z}\}$ with constant grid spacing d will be projected onto an irregular grid $\{\mathbf{c}^{\mathrm{p}}_k\colon\ k \in \mathbb{Z}\}$ with a local grid spacing

$$d_{x,i,j} = \left\| \mathbf{c}^{\mathrm{p}}_{i,j} - \mathbf{c}^{\mathrm{p}}_{i-1,j} \right\| , \tag{4.20}$$

$$d_{y,i,j} = \left\| \mathbf{c}^{\mathrm{p}}_{i,j} - \mathbf{c}^{\mathrm{p}}_{i,j-1} \right\| \cdot \sqrt{3}/2 \,. \tag{4.21}$$

The maximum difference in local grid spacing is given using the largest microlens indices $i_{\mathrm{max}} = \lceil s_x/2d_x \rceil$ and $j_{\mathrm{max}} = \lceil s_y/2d_y \rceil$ by

$$\Delta_{d,x,\mathrm{max}} = \left| d_{x,i_{\mathrm{max}},j_{\mathrm{max}}} - d_{x,-i_{\mathrm{max}},-j_{\mathrm{max}}} \right| , \tag{4.22}$$

$$\Delta_{d,y,\mathrm{max}} = \left| d_{y,i_{\mathrm{max}},j_{\mathrm{max}}} - d_{y,-i_{\mathrm{max}},-j_{\mathrm{max}}} \right| . \tag{4.23}$$

This formula can be used to estimate whether the tilt (if within the constraint (4.19)) is detectable in the microlens image.

In the case of a Lytro Illum camera, with its fixed $f/2$ lens, the microlens focal lengths have to be accurate within $\Delta_f < 2.8\,\mu\mathrm{m}$ according to (4.16). Furthermore, using (4.19) and assuming ideal microlens focal lengths, the tilt has to be accurate within $\Delta_\delta < 0.0088°$. The accuracy will have to be even higher in order for the combined $\Delta_f + \Delta_z$ to fulfill the constraint. For the maximum geometrical distortion of the projected grid within these constraints, following (4.22), one obtains

$$\Delta_{d,x,\mathrm{max}} \approx \Delta_{d,y,\mathrm{max}} = 0.006\,\mathrm{px}\,, \tag{4.24}$$

assuming a 30 mm main lens focal length equivalent for which the scaling factor ζ and distortion effects are the largest. The geometric distortion hence is negligibly small and undetectable in the white image. Therefore, in the following, the model is simplified assuming zero tilt, *i.e.* $\beta = 0 = \gamma$.

4.2.3.3 Proposed pre-calibration

The main purpose of the pre-calibration is to estimate a regular grid approximating the perspectively projected microlens centers $\mathbf{c}^{\mathrm{p}}_k$, which correspond to the coordinates of the central rays of the target light field. In the further decoding pipeline, the estimated grid is used to align the lenslet image with the sensor and slice it to a 4D light field.

Usually, the microlens grid is estimated by detecting the microlens centers from the corresponding white image [21] and building a regular

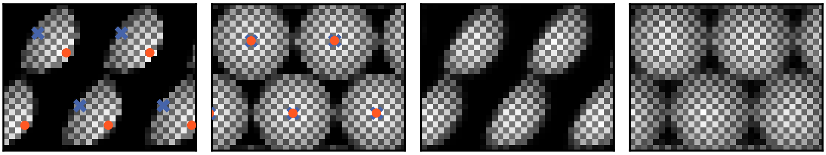
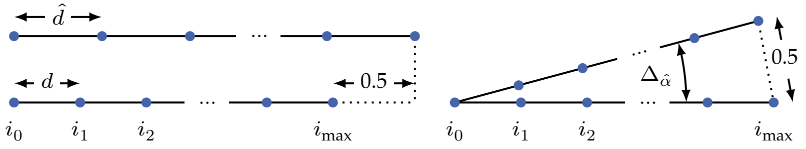


Figure 4.15 Off-center and central image crops from synthetic (left) and real Lytro Illum (right) raw white images. The ground truth perspectively projected centers $\mathbf{c}_k^{\mathrm{p}}$ (●) and orthogonally projected centers $\mathbf{c}_k^{\mathrm{o}}$ (●) are depicted in the case of the synthetic data. Details on the rendering of the synthetic white images are presented in Appendix B.

grid best approximating the detected centers [45] or by directly estimating a regular grid from the white image [40]. Challenges in the detection are versatile: On the one hand, the sheer amount of microlenses in MLA used in practice limits the algorithm's complexity. On the other hand, the geometry of the MLA is not trivial and usually slightly irregular. Furthermore, main lens and microlens vignetting influences the (local) shape and brightness of the microlens images, particularly of those that are close to the sensor edge: while microlens images close to the optical axis are circular and brightest in the center, microlens images at the sensor edge are cat-eye-shaped and show an off-center brightest pixel, as shown in Figure 4.15 for a synthetic and real-world example. There are mainly two methods proposed in the literature: Cho *et al*. [40] first compensate the rotation using an estimate obtained in the Fourier transform of the white image. In the spatial domain, they perform a grayscale erosion and clustering of the demosaiced white image. To estimate the microlens centers, they use a parabolic least-squares regression of the clustered microlenses. In the decoding pipeline by Dansereau *et al*. [45], which is implemented in the *Matlab Light Field Toolbox*, the de-facto open source standard, the raw white image is convolved with a disk kernel. The microlens centers are then estimated by finding the local maxima in the filtered image. This does not result in subpixel precision. However, in the succeeding pre-calibration, the grid parameters are estimated with subpixel precision.

None of the previous algorithms consider vignetting. Taking into account the natural and mechanical vignetting by estimating the microlens grid parameters and coordinates in the spatial domain of the white im-



(a) Grid spacing accuracy requirement. **(b)** Grid rotation accuracy requirement.

Figure 4.16 Schematic depiction of the accuracy requirements for the grid spacing estimate and the grid rotation estimate. Here, i_0 corresponds to the 1D central microlens index, and $i_{\max}$ to the index of the outermost microlens.

age is extremely challenging. Local circle search algorithms have been proposed [142] which show good performance in the image's central regions and in cases of strong vignetting close to the sensor edge, but mediocre performance in cases of only slight mechanical vignetting. Lytro supposedly uses a similar local arc fitting to account for mechanical vignetting [116], but since the software is closed source this is speculative.

The grid parameters (grid spacing, rotation, and offset) have to be estimated with very high accuracy. Assuming that the maximum deviation from a grid point of the estimated regular grid to a true grid point may not exceed 0.5 px, one can estimate upper bounds of the individual accuracy requirements, as depicted in Figure 4.16. Assuming that the grid offset is estimated perfectly and the grid matches the real grid at the sensor center, in order for the grid points furthest from the center to be within 0.5 px of the true grid centers, the accuracy $\Delta_{\hat{d}}$ of the estimated grid spacing $\hat{d}$ has to be accurate within

$$|\Delta_{\hat{d}}| < \frac{0.5}{l_{\max}} = 0.0018\,\text{px} \tag{4.25}$$

in the case of the Lytro Illum camera. Here,

$$l_{\max} = \max\{i_{\max}, j_{\max}\} = \max\{\lceil 2s_x/d_x \rceil, \lceil 2s_y/d_y \rceil\} \tag{4.26}$$

is determined by the longer side of the sensor. Following a similar argument, the accuracy $\Delta_{\hat{\alpha}}$ of the estimated grid rotation $\hat{\alpha}$ has to satisfy

$$\sin|\Delta_{\hat{\alpha}}| < \frac{0.5 \cdot p}{i_{\max} \cdot d} \implies |\Delta_{\hat{\alpha}}| < \arcsin \frac{0.5 \cdot p}{i_{\max} \cdot d} = 0.0074°\,. \tag{4.27}$$

Finally, the grid offset leads to a global shift of the estimated grid, and should hence at least be accurate within

$$|\Delta_{\hat{\mathbf{o}}}| < 0.5\,\text{px}\,. \tag{4.28}$$

These accuracy estimates pose challenging requirements on the estimation algorithms, in particular on the estimation of the grid spacing. Here, a novel algorithm is proposed, which operates in the Fourier domain to estimate the grid spacing and rotation, and in the spatial domain to estimate the grid offset. The estimation takes into account the natural and mechanical vignetting present in the white images.

4.2.3.4 Grid rotation and spacing estimation

A white image can be interpreted as an (approximately) regular structure: ignoring natural and mechanical vignetting, a (continuous) white image is made up of a texture element (texel) $e(\mathbf{x})$, which is arranged in a grid spanned by the vectors $\mathbf{b}_1$ and $\mathbf{b}_2$. For example,

$$\mathbf{b}_1 = (0, 2r), \mathbf{b}_2 = (\sqrt{3} \cdot r, r) \tag{4.29}$$

for a perfect hexagonal grid with a microlens radius $r = d/2$. Formulating the white image as a periodic texture via 2D convolution, denoted by $**$, it can be written as

$$g_{\text{ideal}}(\mathbf{x}) = e(\mathbf{x}) ** \sum_{i,j \in \mathbb{Z}} \delta(\mathbf{x} - i\mathbf{b}_1 - j\mathbf{b}_2)) \tag{4.30}$$

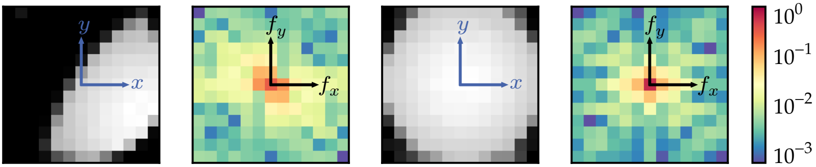
$$\circ\!\!-\!\!\!-\!\!\bullet$$

$$G_{\text{ideal}}(\mathbf{f}) \propto E(\mathbf{f}) \cdot \sum_{i,j \in \mathbb{Z}} \delta(\mathbf{f} - i\mathbf{f}_1 - j\mathbf{f}_2)\,, \tag{4.31}$$

where $E(\mathbf{f})$ denotes the Fourier transform of $e(\mathbf{x})$. The frequency basis vectors $\mathbf{f}_k = (f_{k,x}, f_{k,y})$, are given by [18]

$$\begin{pmatrix} b_{1,x} & b_{2,x} \\ b_{1,y} & b_{2,y} \end{pmatrix} = \begin{pmatrix} f_{1,x} & f_{1,y} \\ f_{2,x} & f_{2,y} \end{pmatrix}^{-1} , \tag{4.32}$$

where $b_{k,x}, b_{k,y}$ are the components of the vectors $\mathbf{b}_k$. Ideally, one could estimate the grid spacing and rotation by detecting the peaks corresponding to $\mathbf{f}_1, \mathbf{f}_2$ (and their multiples) in the absolute value of the Fourier transform $G_{\text{ideal}}(\mathbf{f})$.



$e_{-i_{\max},j_{\max}}(\mathbf{x})$ $|E_{-i_{\max},j_{\max}}(\mathbf{f})|$ $e_{0,0}(\mathbf{x})$ $|E_{0,0}(\mathbf{f})|$

Figure 4.17 Off-center and central local texels of a white image with the corresponding (normalized) Fourier transform magnitudes.

Introducing vignetting does not change the grid vectors but instead modulates the (now local) texel: in the image center, the texel is not altered, but deviating from the center the brightness distribution of the texel changes due to natural vignetting. Furthermore, due to mechanical vignetting, some pixels of the texel are blocked. Therefore, the texel $e_{i,j}(\mathbf{x})$ is different at every grid position (i, j). As an example, two local (discrete) texels are shown in Figure 4.17. One obtains the white image

$$
\begin{aligned}
g(\mathbf{x}) &= \sum_{i,j\in\mathbb{Z}} e_{i,j}(\mathbf{x}) ** \delta(\mathbf{x} - i\mathbf{b}_1 - j\mathbf{b}_2)) && (4.33)\\
&\begin{array}{c}\circ\\[-6pt] |\\[-6pt] \bullet\end{array} \\
G(\mathbf{f}) &\propto \sum_{i,j\in\mathbb{Z}} E_{i,j}(\mathbf{f}) \cdot \mathrm{e}^{-2\pi\mathrm{i}\,(i\mathbf{b}_1 + j\mathbf{b}_2)\cdot\mathbf{f}}\,, && (4.34)
\end{aligned}
$$

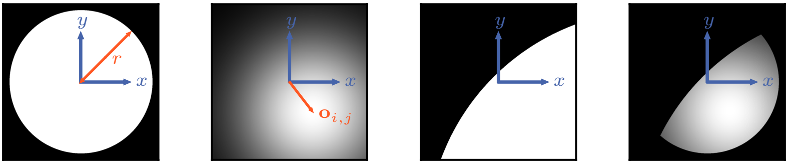
where $E_{i,j}(\mathbf{f})$ denotes the Fourier transform of $e_{i,j}(\mathbf{x})$. As every texel is different, (4.34) cannot be simplified to a Dirac comb as the ideal case (4.31).

Now, it is proposed to model the local texels as

$$
e_{i,j}(\mathbf{x}) = e(\mathbf{x}) \cdot m^{\mathrm{nv}}_{i,j}(\mathbf{x}) \cdot m^{\mathrm{mv}}_{i,j}(\mathbf{x})\,, \tag{4.35}
$$

where $e(\mathbf{x})$ is a binary circular mask with microlens radius r, $m^{\mathrm{nv}}_{i,j}(\mathbf{x})$ is the modulation due to natural vignetting, whose shape does not need to be specified explicitly but could for example be modeled as a wide Gaussian bell or an approximation of the $\cos^4$ law, and $m^{\mathrm{mv}}_{i,j}(\mathbf{x})$ describes the modulation due to mechanical vignetting, which can be modeled again as a binary circular mask of large radius with non-zero center. In more detail, the natural vignetting can be written as

$$
m^{\mathrm{nv}}_{i,j}(\mathbf{x}) = m^{\mathrm{nv}}_{0,0}(\mathbf{x} - \mathbf{o}_{i,j})\,, \tag{4.36}
$$



$e(\mathbf{x})$ $m^{\mathrm{nv}}_{i,j}(\mathbf{x})$ $m^{\mathrm{mv}}_{i,j}(\mathbf{x})$ $e_{i,j}(\mathbf{x})$

Figure 4.18 Local texel model consisting of an ideal circular microlens image and modulation due to natural and mechanical vignetting.

where $\mathbf{o}_{i,j}$ is the distance of the perspectively projected microlens center to the orthogonally projected one, since natural vignetting causes the brightest pixel to be at the orthogonally projected center but the modulation shape does not change otherwise. A schematic drawing of the local texel model is shown in Figure 4.18.

The Fourier transform of the local $e_{i,j}(\mathbf{x})$, using (4.35) and (4.36), is

$$\begin{aligned} E_{i,j}(\mathbf{f}) &= E(\mathbf{f}) ** M^{\mathrm{nv}}_{i,j}(\mathbf{f}) ** M^{\mathrm{mv}}_{i,j}(\mathbf{f}) \\ &= E(\mathbf{f}) ** M^{\mathrm{nv}}_{0,0}(\mathbf{f}) \cdot \mathrm{e}^{-2\pi \mathrm{i}\, \mathbf{o}_{i,j} \cdot \mathbf{f}} ** M^{\mathrm{mv}}_{i,j}(\mathbf{f}) \, , \end{aligned} \tag{4.37}$$

where both the Fourier transform $E(\mathbf{f})$ and $M^{\mathrm{mv}}_{i,j}(\mathbf{f})$ are Airy discs of different widths (and phases). It was observed that natural vignetting causes the periodic peaks in the Fourier transform of the white image to shift. This is likely due to the underlying periodic structure of the modulation which manifests itself in a phase factor: the natural vignetting of the overall white image can be seen as a regular texture which is arranged in the hexagonal grid of the orthogonally projected microlens centers (instead of the perspectively projected ones). To eliminate this effect, it is proposed to perform a strong gamma compression to effectively eliminate the modulation $m^{\mathrm{nv}}_{0,0}(\mathbf{x})$. That is, one defines

$$\tilde{g}_\gamma(\mathbf{x}) = g^\gamma(\mathbf{x}) \approx \sum_{i,j \in \mathbb{Z}} e(\mathbf{x}) \cdot m^{\mathrm{mv}}_{i,j}(\mathbf{x}) ** \delta(\mathbf{x} - i\mathbf{b}_1 - j\mathbf{b}_2) \tag{4.38}$$

$$\begin{array}{c}\circ \\ | \\ \bullet\end{array}$$

$$\tilde{G}_\gamma(\mathbf{f}) \approx \sum_{i,j \in \mathbb{Z}} E(\mathbf{f}) ** M^{\mathrm{mv}}_{i,j}(\mathbf{f}) \cdot \mathrm{e}^{-2\pi \mathrm{i}\, (i\mathbf{b}_1 + j\mathbf{b}_2) \cdot \mathbf{f}} \, , \tag{4.39}$$

for $\gamma \ll 1$ such that

$$\left(m^{\text{nv}}_{0,0}(\mathbf{x})\right)^{\gamma} \approx 1 \;\circ\!\!-\!\!\bullet\; \delta(\mathbf{f})\,. \tag{4.40}$$

Note that since $e(\mathbf{x})$ and $m^{\text{mv}}_{i,j}(\mathbf{x})$ are binary they are unchanged from the gamma compression. The gamma compression has the additional advantage that the algorithm can equivalently operate on the raw, mosaiced white image, since the compression will push all gray values to one, mitigating the possible effects introduced by demosaicing algorithms, which are particularly severe in the case of microlens images [47]. Hence, the proposed method is applied directly to the raw white images.

In practice, the texels will deviate from the texel model (4.35), in particular, $e(\mathbf{x})$ and $m^{\text{mv}}_{i,j}(\mathbf{x})$ will not be exactly binary. The gamma compression then would have no effect as all non-zero pixels would be mapped to one. To make sure that the texel model (4.35) is acceptable, an adaptive contrast stretching prior to gamma compression is performed. That is, for some $q \in (0, 1)$, values in the range $[q, 0.99]$ are linearly mapped to $[0, 1]$ and subsequently clipped to the target range. The value of q depends on the white image and the camera parameters. The resulting white image is denoted by $\tilde{g}_{\gamma,q}$. Furthermore, since the mechanical vignetting effects off-center texels $(i, j) > (I, J)$ for some $I, J \in \mathbb{Z}$, the white image is windowed using a rotationally symmetric Gaussian window $w_\sigma(\mathbf{x})$ prior to calculating the Fourier transform. This suppresses off-center texels, which are distorting the ideal spectrum due to mechanical vignetting. The standard deviation σ of the Gaussian window is chosen such that

$$\begin{aligned}\tilde{g}_{\sigma,\gamma,q}(\mathbf{x}) &= w_\sigma(\mathbf{x}) \cdot \tilde{g}_{\gamma,q}(\mathbf{x}) \\ &\approx w_\sigma(\mathbf{x}) \cdot \left(e(\mathbf{x}) ** \sum_{i,j\in\mathbb{Z}} \delta(\mathbf{x} - i\mathbf{b}_1 - j\mathbf{b}_2) \right)\,.\end{aligned} \tag{4.41}$$

Hence, its Fourier transform is approximately

$$\begin{aligned}\tilde{G}_{\sigma,\gamma,q}(\mathbf{f}) &\approx W_\rho(\mathbf{f}) ** \left(E(\mathbf{f}) \cdot \sum_{i,j\in\mathbb{Z}} \delta(\mathbf{f} - i\mathbf{f}_1 - j\mathbf{f}_2) \right) \\ &= \sum_{i,j\in\mathbb{Z}} W_\rho(\mathbf{f} - i\mathbf{f}_1 - j\mathbf{f}_2) \cdot E(i\mathbf{f}_1 + j\mathbf{f}_2)\,,\end{aligned} \tag{4.42}$$

where the $\mathbf{f}_i$ are given by (4.32) and the Fourier transform $W_\rho(\mathbf{f})$ of the Gaussian window is again a Gaussian window with a standard deviation

$\rho = 1/(2\pi\sigma)$. Therefore, the Fourier transform $\tilde{G}_{\sigma,\gamma,q}(\mathbf{f})$ is given by a sum of shifted Gaussians centered at linear combinations of the grid frequency vectors $\mathbf{f}_i$. Since the standard deviation σ of the window in the spatial domain is much larger than the grid spacing, the standard deviation ρ of the window in the Fourier domain is much smaller than the grid frequency spacing. More specifically, in the case of the Lytro Illum camera, using a standard deviation of $\sigma = 100\,\text{px}$ and a hexagonal grid spacing of 15 px, one finds a standard deviation of $\rho = 0.0016\,\text{px}^{-1}$ and a smallest distance of frequency basis vectors of 0.0770 px^{-1}. Therefore, the center of each Gaussian in (4.42) lies outside the 51σ neighborhood of the closest neighboring Gaussian. The local maximum of each Gaussian is accordingly virtually undisturbed by neighboring ones. Hence, the peaks in $\hat{G}_{\sigma,\gamma,q}(\mathbf{f})$ approximate well the linear combinations of the grid frequency spacing vectors $\mathbf{f}_i$. In the discrete case, a rotationally symmetric Hann window whose width is determined by the length of the smaller dimension of the white image is applied to reduce spectral leakage.

By estimating the peaks in the spectrum $\hat{G}_{\sigma,\gamma,q}(\mathbf{f})$ of the contrast-stretched, gamma-compressed and windowed white image, the grid spacing of the underlying perspectively projected microlens centers can be estimated via (4.32). The basis vectors $\mathbf{f}_i$ are estimated in the Fourier domain by finding the local maxima in the magnitude of the Fourier-transformed white image that correspond to the first n multiples of the frequency basis vectors $n\mathbf{f}_i$. The number n of detected maxima depends on the camera used. In the case of the Lytro Illum camera, one finds up to $n = 5$ values per frequency basis vector. Zero padding and a centroid calculation are used to estimate the sub-pixel coordinates of these frequency vectors, *i.e.* for every cluster $\mathcal{C}_{i,n}$ around a peak corresponding to $n\mathbf{f}_i$ one calculates

$$\hat{\mathbf{f}}_{i,n} = \frac{\sum_{j \in \mathcal{C}_{i,n}} \mathbf{f}_j \, \tilde{G}_{\sigma,\gamma,q}(\mathbf{f}_j)}{\sum_{j \in \mathcal{C}_{i,n}} \tilde{G}_{\sigma,\gamma,q}(\mathbf{f}_j)} \,. \tag{4.43}$$

In total, the proposed method depends on three hyperparameters, σ, γ, and q. In order to determine these parameters appropriately (*i.e.*, such that the made approximations hold), a certain measure that does not depend on any prior knowledge (*e.g.* the microlens centers or the underlying grid spacing) is needed. To this end, the pair-wise distances of the estimated frequencies $\hat{\mathbf{f}}_{i,n}$

$$\hat{d}_{i,n} = \|\hat{\mathbf{f}}_{i,n+1} - \hat{\mathbf{f}}_{i,n}\| \tag{4.44}$$

is calculated, which should ideally be constant in the case of a regular grid for $i = 1, 2$, respectively. Hence, the estimated standard deviation $\hat{s}_i$ of the samples $\hat{d}_{i,n}$ should ideally be zero. Therefore, the hyperparameters γ, q, and σ are optimized by minimizing the sample standard deviation $\hat{s}_i$. For a specific task, *e.g.* given a light field camera with a fixed focal length lens, this could be done manually. But for the Lytro camera, the white images corresponding to the different focal lengths show different characteristics in terms of vignetting, grid spacing, and brightness distribution. Therefore, and to obtain an automated calibration process, optimization via differential evolution on a predefined search space of the hyperparameters is used to minimize the estimated standard deviation $\hat{s}_i$ of the Fourier grid spacing. Using the final estimated grid frequency vectors $\hat{\mathbf{f}}_k$ with the corresponding estimated grid spacing vectors $\hat{\mathbf{b}}_k$, one obtains the estimates of the grid spacing and rotation of the perspectively projected microlens grid. An overview of the proposed grid estimation is shown in Figure 4.19(a).

4.2.3.5 Grid offset estimation

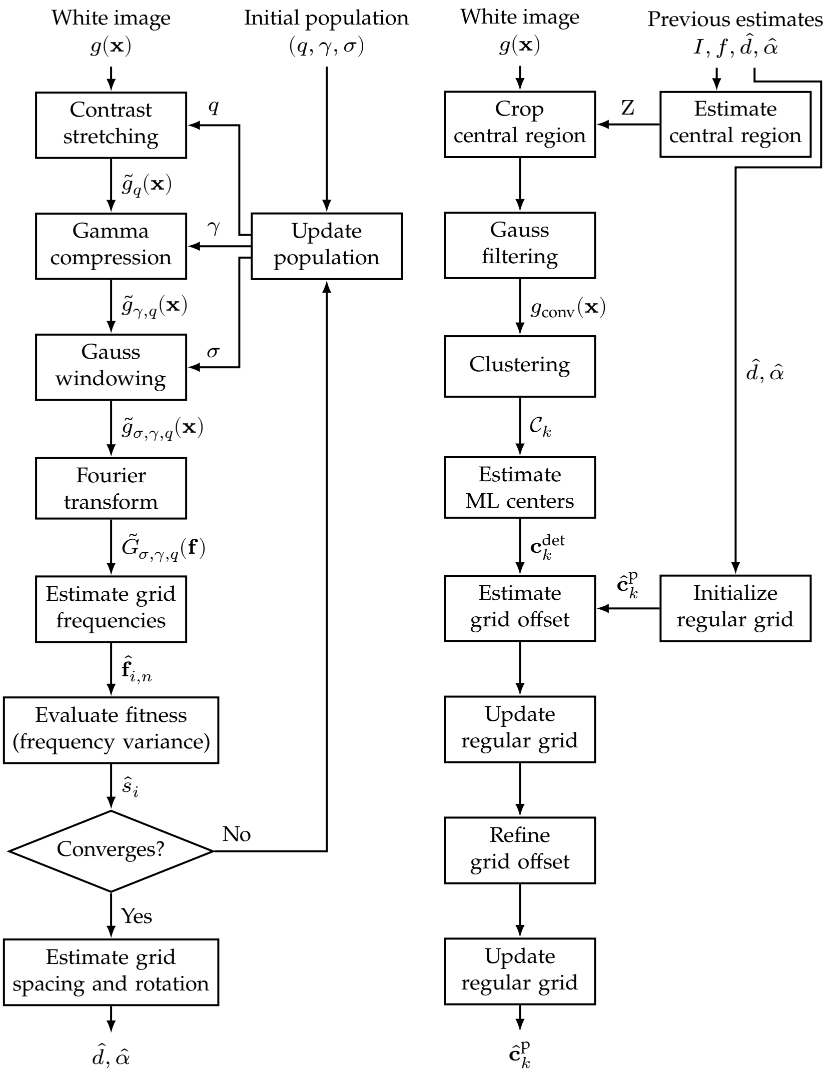
Having estimated the grid rotation and spacing, the overall grid offset remains to be estimated. This is done in the spatial domain. Dansereau *et al.* [45] estimate the offset by building an initial regular grid, using the previously estimated grid spacing and rotation, and measuring the median distance of the regular grid points to the previously detected microlens centers. Here, this approach is refined. First, the microlens centers are estimated only in the central region of the image where the expected difference of perspectively to orthogonally projected centers is less than 0.5 px. This increases the accuracy of the detection since the orthogonally projected centers are easier to detect due to the natural vignetting. That is, the detection region Z is restricted such that

$$\left\|\mathbf{c}^{\mathrm{p}}_{i_{\max}0} - \mathbf{c}^{\mathrm{o}}_{i_{\max}0}\right\| = d \cdot i_{\max}(\zeta_{\max 0} - 1) < 0.5\,. \tag{4.45}$$

Since $d_x > d_y$, this suffices to fulfill the analogous constraint in the y-direction as well. A rough estimate of the factor $\zeta \approx (I + f)/I$ obtained from (4.15) is used, given a rough estimate of the image distance I, the microlens focal length f, and the previously estimated MLA grid spacing $\hat{d}$. The restricted region Z will be comparatively small, depending on the factor ζ and hence on the main lens focal length, consisting of as few as 50×50 microlenses in the case of a 30 mm main lens. While this small region is not suited to estimate the grid spacing with high accuracy, estimating the overall offset can be done with much fewer measurements, *i.e.* fewer available microlens centers. Having restricted the detection region, low pass filtering is performed using a Gaussian kernel to reduce noise. In a second step, the image is clustered using local thresholding (by local Gaussian weighted mean with a block size of 17 px), to find areas around local peaks, and a standard cluster labeling algorithm. Each cluster represents exactly one microlens. Finally, the microlens centers are estimated from the detected clusters. That is, for each detected cluster $\mathcal{C}_k$, the center of mass, analogously to the calculation (4.43) in the Fourier domain, is calculated as

$$\mathbf{c}_k^{\text{det}} = \frac{\sum_{n \in \mathcal{C}_k} \mathbf{x}_n \, g_{\text{conv}}(\mathbf{x}_n)}{\sum_{n \in \mathcal{C}_k} g_{\text{conv}}(\mathbf{x}_n)} \,, \tag{4.46}$$

where g_{conv} denotes the region-restricted, Gauss-filtered white image. To estimate the grid offset, analogously to Dansereau *et al.* [45], the median distance of the initialized regular grid points $\hat{\mathbf{c}}_k^{\text{p}}$ to the detected microlens centers $\mathbf{c}_k^{\text{det}}$ is calculated. In addition, as a refinement step, a *weighted* median distance of the updated regular grid point to the detected microlens centers is calculated, assigning a higher weight to those microlens centers that are more central. The weights are chosen as a symmetric Gaussian distribution. Since the detection inaccuracies due to natural vignetting will be smaller in the image center, this should yield a more reliable final result of the estimated grid offset. Using the estimated grid spacing, rotation, and offset, the final estimated regular hexagonal grid $\{\hat{\mathbf{c}}_k^{\text{p}} \colon \, k \in \mathbb{Z}\}$ approximating the perspectively projected microlens centers $\mathbf{c}_k^{\text{p}}$ can be constructed. An overview of the proposed offset estimation method is depicted in Figure 4.19(b). The proposed pre-calibration is evaluated in Appendix B as it is not within the main focus of this thesis.



(a) Grid spacing and rotation estimate.

(b) Estimate of the projected regular grid.

Figure 4.19 Flowcharts of the proposed grid estimation algorithms based on grid spacing and rotation estimate in the Fourier domain and full grid estimate in the spatial domain. Detailed steps for the optimization via differential evolution are omitted.

4.2.4 Light field decoding

To decode a light field $\boldsymbol{\mathcal{L}}$ from the raw measurements $\tilde{\boldsymbol{\mathcal{R}}}$, the well-known decoding pipeline by Dansereau *et al.* [45] is used. Other schemes follow a similar approach and differ only in detail [21, 40, 47]. In the considered multispectral case, the conventional decoding scheme is applied to the individually captured channels.

First, white balancing and devignetting are applied to the raw measurement. In the conventional RGB case, this is performed using a single bright image and subsequent demosaicing. However, in the presented case, as discussed, this is achieved using the radiometric calibration via (4.8). Using the estimated regular microlens center grid, the calibrated raw measurement is aligned with the grid. That is, the raw measurement is translated by the estimated grid offset $\mathbf{o}$, rotated to compensate the estimated grid rotation $\hat{\alpha}$, and upscaled such that the upscaled grid spacing is integer-valued. This way, the aligned microlens centers fall exactly onto pixel centers. The decoding can now be achieved by a simple slicing of the aligned, calibrated raw measurement. For example, the central subaperture view is obtained by collecting all pixels that correspond to the microlens centers. Adjacent subapertures are decoded using the neighboring pixels. This way, the light field is decoded subaperture-wise. For light field cameras with a hexagonal MLA, such as the considered Lytro Illum camera, the obtained light field is spatially sampled on a hexagonal grid. In this case, the light field has to be resampled to a rectangular grid as a final step. After decoding, the light field can be rectified using a calibrated camera model [45]. As the decoding of raw measurements from an MLA-based light field camera is a standard procedure, the reader is referred to the literature for details.

However, it should be noted that, in the case of a light field camera with a spectrally coded MLA, the final resampling to a rectangular grid cannot be performed at this point as the sparse measurement does not allow for the necessary interpolation. For this reason, in this thesis, the coding is simulated in the light field domain according to (2.5) and not at the level of the raw measurements, which would ideally be preferable. In principle, it would be possible to code the raw measurements and perform the resampling together with the reconstruction, with additional challenges. For example, to adapt the proposed principle reconstruction

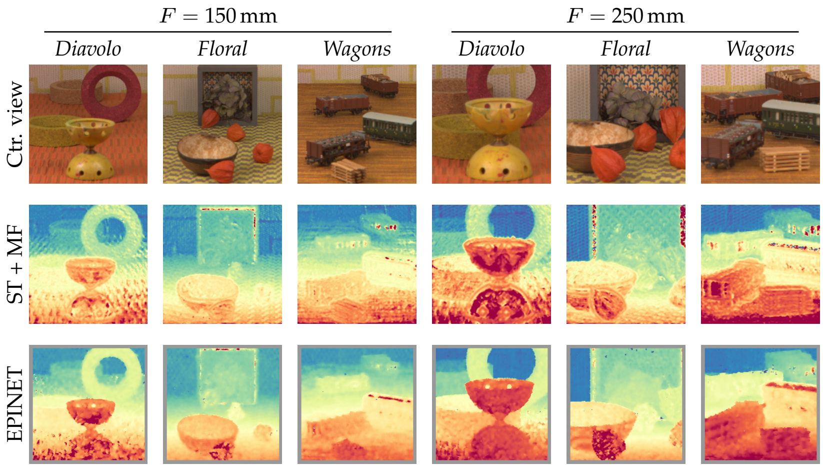


Figure 4.20 Central views of the captured real-world dataset consisting of three scenes at two different focal length equivalents. The disparity estimation from the RGB-converted light fields using the structure tensor (ST) with TV-L1 fusion and median filtering (MF) [213], as well as using EPINET [176] are shown using the disparity ranges given in Table 4.1(b).

to hexagonally sampled light fields, one needs to replace the used spatial convolutions with hexagonal convolutions [86, 128]. In the case of 3D convolutions, it is probably necessary to separate the convolution into a 2D hexagonal convolution and a 1D spectral convolution. This way, the network could perform the reconstruction and disparity estimation *as well as* the resampling. Of course, this would require both hexagonally and the corresponding rectangularly sampled light fields as training data. It is therefore reasonable to first consider the case using a rectangular MLA and exclude these additional challenges. Once the challenges regarding a full hardware prototype and the reconstruction of the correspondingly coded light fields have been resolved, one can investigate the case of hexagonally sampled coded raw measurements.

The calibrated and decoded real-world dataset is shown in Figure 4.20. Visually, the quality of the captured spectral light fields is adequate. Judging by the qualitative performance of the shown disparity estimates, the epipolar geometry of the decoded light field seems to be consistent, indicating a good geometric calibration of the camera. It should be noted that recent model-free geometric calibration methods (*e.g.* the so-called generalized camera model [73, 198]) outperform the model-based ones used here. Using a generalized camera calibration, each individual pixel is assigned a ray corresponding to its line of sight with very high accuracy, for example using elaborate phase shift coding strategies. While in many applications it is sufficient to describe the camera using a set of rays (also called raxels), these generic models are usually not suited to obtain an image or light field in the conventional sense, *i.e.* sampled on a regular grid. Recently, however, a method to resample the raxels of an MLA-based light field camera to a regularly sampled light field was developed by Uhlig and Heizmann [199], which likely further improves the quality of the reconstructed light fields, in particular with respect to the epipolar geometry most notably in peripheral subapertures. However, in the considered case, the reconstruction quality is assumed to be limited also by the spatial noise introduced by the RGB filters and the radiometric calibration and not solely by the geometric calibration of the camera. This becomes evident when investigating the spectral channels of the decoded light fields in more detail. For example, Figure 4.21 shows five out of the 13 spectral channels of the central view of the *Floral* scene. Due to the very low sensitivity in the case of very small and very long wavelengths (*i.e.* channels one and 13), the exposure times are chosen to be very long in those instances, as previously elaborated. Here, one observes increased impulse-like sensor noise, likely due to defective pixels. Furthermore, some texture noise can be observed which likely stems from the least sensitive pixels, *i.e.* the red color filter pixels in the case of channel one (400 nm) and the blue color filter pixels in the case of channel 13 (700 nm). This visible noise is reduced in the RGB converted case due to the involved integration. In principle, impulse noise can be reduced using a median filter. However, median filtering is also likely to reduce the available details in the light field in noise-free regions. Therefore, the decoded light fields are used as-is without any additional smoothing. Concluding, the overall quality of the decoded spectral light fields is adequate and well suited for the evaluation of the considered reconstruction methods while incorporating a non-negligible amount of noise which limits the performance of the reconstruction and disparity estimation.

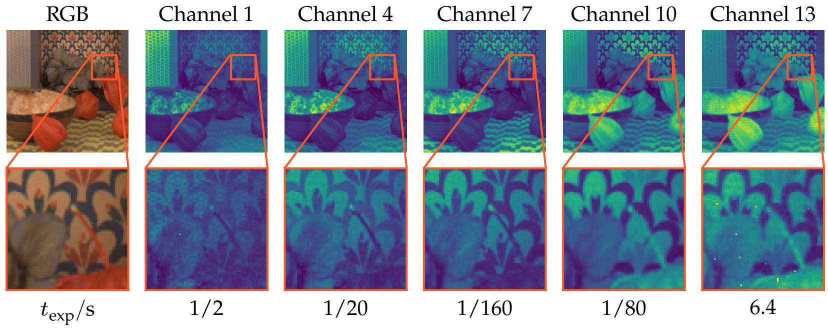


Figure 4.21 False-color representation of five out of the 13 spectral channels from the central view of the calibrated, decoded light field of the *Floral* scene imaged using the 250 mm focal length equivalent. The listed exposure times correspond to the individual exposure times that were used to capture the raw measurement of the corresponding channel. For visualization, the individual channels are normalized to maximize the contrast.

4.3 Evaluation metrics

Both the reconstructed central view and the estimated disparity are evaluated using the ground truth data and several quality metrics. For the spectral central view, the peak signal-to-noise ratio (PSNR) in dB,

$$\mathrm{PSNR}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = 10 \lg \frac{1}{\mathrm{MSE}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}})}\,, \tag{4.47}$$

is used to assess the overall quality of the reconstruction. In all cases, the ground truth central view is normalized to the range $[0, 1]$. Here,

$$\mathrm{MSE}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \frac{1}{ST\Lambda} \sum_{st\lambda} \left(\boldsymbol{\mathcal{I}}_{st\lambda} - \hat{\boldsymbol{\mathcal{I}}}_{st\lambda}\right)^2 \tag{4.48}$$

denotes the mean squared error (MSE) of the spectral reconstruction, which corresponds to the energy of reconstruction error. For a perfect reconstruction, the MSE is zero and the PSNR diverges. Despite its simplicity, the PSNR has been shown to correlate comparably well with the perceived visual quality in many cases [163]. However, the PSNR only takes into account pixel-wise differences and is not well suited to assess the spatial and spectral reconstruction in more detail.

To evaluate the spatial reconstruction, the structural similarity index metric (SSIM), as introduced by Wang *et al.* [210], is used. For two monochromatic image patches $\mathbf{a}$ and $\mathbf{b}$, the structural similarity is defined as

$$\mathrm{SSIM}(\mathbf{a}, \mathbf{b}) = \frac{(2\mu_{\mathbf{a}}\mu_{\mathbf{b}} + c_1)(2\sigma_{\mathbf{ab}} + c_2)}{(\mu_{\mathbf{a}}^2 + \mu_{\mathbf{b}}^2 + c_1)(\sigma_{\mathbf{a}}^2 + \sigma_{\mathbf{b}}^2 + c_2)} \,. \tag{4.49}$$

Here, $\mu_{\mathbf{a}}$ and $\sigma_{\mathbf{a}}^2$ denote the sample mean and variance of $\mathbf{a}$, respectively (with an analogous definition for $\mathbf{b}$). The sample covariance of $\mathbf{a}$ and $\mathbf{b}$ is denoted by $\sigma_{\mathbf{ab}}$. For numerical stability, the small constants c_1 and c_2 are introduced, which are set to

$$c_1 = 1 \times 10^{-4} \,, \quad c_2 = 9 \times 10^{-4} \tag{4.50}$$

as in the original paper. For a perfect reconstruction, the SSIM is one. To calculate the structural similarity for two full-sized images, a small window is slid across the images to extract the patches on which the local SSIM values are calculated. In practice, a Gaussian filter is used to estimate the local averages, variances, and covariance via convolution. The obtained local SSIM map is then averaged to obtain a single quality metric. As in the original paper, a kernel size of $(11, 11)$ is used here in the case of the full-sized light fields. For the light fields in the training, validation, and test dataset, with the smaller spatial resolution of $(32, 32)$, a kernel size of $(5, 5)$ is used instead. To calculate the SSIM for color or spectral images, there are mainly two possibilities. Ideally, color images are converted to a luminance-based color space such as YUV or YCbCr. Then, the SSIM is calculated either using only the luma values or using both the luma and chrominance with subsequent averaging. In the case of spectral images, this approach is computationally quite expensive and would drastically slow down the training of the investigated deep learning models. Therefore, the SSIM is calculated channel-wise and averaged, which is common not only for spectral but also for color images.

There also exists a multi-scale generalization of the structural similarity (MS-SSIM) which was introduced by Wang *et al.* [209]. Here, the SSIM is calculated on different layers of an image pyramid containing the original and downsampled images. The MS-SSIM is then defined as the weighted mean of the individual SSIM values on the different scales.

Wang *et al.* show that the MS-SSIM is slightly more in accordance with the perceived visual quality as compared to the SSIM. Typically, the MS-SSIM is calculated on five layers using the weights 0.0448, 0.2856, 0.3001, 0.2363, and 0.1333, which are proposed in the original paper. This approach is adopted here in the case of the full-sized light fields. For the smaller resolution of the light fields in the training, validation, and test dataset this approach is not feasible. In this case, the MS-SSIM is calculated on three layers using the weights 0.5, 0.3, and 0.2 and a reduced filter size of $(3, 3)$. In the context of this thesis, it was observed that the SSIM and MS-SSIM did not yield significantly different results. Hence, throughout this thesis, only the SSIM is presented. However, the MS-SSIM is also evaluated for all experiments and available within the digital supplement.

Finally, to evaluate the quality of the reconstructed spectra, two metrics are used. The spectral angle (SA), which is sometimes also referred to as the spectral angle mapper, is defined via the mean cosine similarity of the individual ground truth and reconstructed spectra,

$$\mathrm{CS}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \frac{1}{ST} \sum_{st} \frac{\langle \mathbf{i}_{s,t}, \hat{\mathbf{i}}_{s,t} \rangle}{\|\mathbf{i}_{s,t}\| \cdot \|\hat{\mathbf{i}}_{s,t}\|} \,. \tag{4.51}$$

Here, $\mathbf{i}_{s,t}$ denotes the ground truth spectrum at (s, t), *i.e.*

$$(\mathbf{i}_{s,t})_\lambda = \mathcal{I}[s, t, \lambda] \,, \tag{4.52}$$

with the analogous definition for the reconstructed spectra $\hat{\mathbf{i}}_{s,t}$. To obtain the SA, the cosine similarity is then converted to the corresponding angle,

$$\mathrm{SA}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \cos^{-1} \mathrm{CS}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) \,. \tag{4.53}$$

For a perfect spectral reconstruction (modulo scaling), the CS is one and the SA is zero. Note that, due to the nonlinearity of the cosine, the above definition is slightly different from first calculating the spectral angle of the individual spectra and averaging *subsequently*, which is sometimes also used in the literature.

As an alternative to the SA, interpreting the spectral pixels as random variables, the spectral information divergence (SID) was proposed by Chang [34]. The SID is defined as

$$\mathrm{SID}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \frac{1}{ST} \sum_{st\lambda} \left(\boldsymbol{\mathcal{P}}_{st\lambda} \ln \frac{\boldsymbol{\mathcal{P}}_{st\lambda}}{\boldsymbol{\mathcal{Q}}_{st\lambda}} + \boldsymbol{\mathcal{Q}}_{st\lambda} \ln \frac{\boldsymbol{\mathcal{Q}}_{st\lambda}}{\boldsymbol{\mathcal{P}}_{st\lambda}} \right), \tag{4.54}$$

where $\boldsymbol{\mathcal{P}}$ and $\boldsymbol{\mathcal{Q}}$ correspond to the spectrally normalized ground truth and reconstructed central view, respectively, *i.e.*

$$\boldsymbol{\mathcal{P}}_{st\lambda} = \boldsymbol{\mathcal{I}}_{st\lambda} \Big/ \sum_{\lambda'} \boldsymbol{\mathcal{I}}_{st\lambda'}, \quad \boldsymbol{\mathcal{Q}}_{st\lambda} = \hat{\boldsymbol{\mathcal{I}}}_{st\lambda} \Big/ \sum_{\lambda'} \hat{\boldsymbol{\mathcal{I}}}_{st\lambda'}. \tag{4.55}$$

For each pixel (s, t), $\boldsymbol{\mathcal{P}}_{st\lambda}$ and $\boldsymbol{\mathcal{Q}}_{st\lambda}$ can be viewed as a discrete probability distribution. Therefore, the SID basically corresponds to the mean Jensen–Shannon divergence, a symmetric generalization of the Kullback-Leibler divergence, of the normalized ground truth and the estimated spectra. For a perfect reconstruction, the SID is zero. Note that, for a simpler implementation, the natural logarithm is used for the calculation of the Kullback-Leibler divergence. Hence, throughout this thesis the SID is evaluated in units of nat rather than bit. However, it is in principle straightforward to convert to bit through division by a factor of $\ln 2$.

In the case of the estimated disparity, three quality metrics are evaluated if the corresponding ground truth data is available. As is common in the light field community, the MSE,

$$\mathrm{MSE}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = \frac{1}{ST} \sum_{st} \left(\boldsymbol{\mathcal{D}}_{st} - \hat{\boldsymbol{\mathcal{D}}}_{st} \right)^2, \tag{4.56}$$

as well as the mean absolute error (MAE),

$$\mathrm{MAE}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = \frac{1}{ST} \sum_{st} \left| \boldsymbol{\mathcal{D}}_{st} - \hat{\boldsymbol{\mathcal{D}}}_{st} \right|, \tag{4.57}$$

are used to assess the overall quality of the estimated disparity. For both the MSE and the MAE, a perfect estimate results in a value of zero, however the MSE is more sensitive to outliers.

Furthermore, the BadPix metric, as proposed by Honauer *et al.* [85], is used for the evaluation. The BadPix corresponds to the percentage of pixels for which the estimated disparity deviates from the ground truth by more than a specified value. More precisely, for any $q > 0$, the BadPix is defined as

$$\mathrm{BadPix}_q(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = 100 \cdot \frac{\left|\left\{(s,t) \in \mathcal{M} : |\boldsymbol{\mathcal{D}}_{st} - \hat{\boldsymbol{\mathcal{D}}}_{st}| > q\right\}\right|}{|\mathcal{M}|}, \tag{4.58}$$

where $\mathcal{M} = \{1, 2, \dots, S\} \times \{1, 2, \dots, T\}$, $|\mathcal{M}| = ST$, denotes the set of all pixels. As is common in the community, the BadPix metric is evaluated for q values 0.01 px, 0.03 px, and 0.07 px which are abbreviated as BP01, BP03, and BP07 in the remainder. Again, no crucial qualitative difference between the different BadPix metrics was found during the evaluation. Hence, only the values of BP07 are presented in this thesis while the remaining ones are available within the digital supplement.

It should be noted that the evaluation metrics in the case of the disparity estimation are hard to compare among different scenes. Unlike for the central views, which are always normalized to values within the range $[0, 1]$, the disparity range is basically unconstrained. A deviation of, *e.g.*, 0.01 px of the estimated disparity from the ground truth is relatively less severe for larger disparity values than for smaller ones. Due to the nonlinear character of the disparity with respect to the depth (*cf.* Appendix A), small deviations for positive disparity values (*i.e.* for objects close to the camera) may be more severe than for negative disparity values, for which the disparity is less sensitive. However, these details are lost when averaging over the disparity map or even a whole dataset for the calculation of the MAE and the MSE, which do not take the disparity range into account. For similar reasons, the BadPix metric suffers from the same problems when compared among different scenes. The mean absolute percentage error (MAPE) is often used to explicitly take into account the scale of the ground truth value. However, the MAPE cannot be applied in the case of disparity estimation due to the common occurrence of small and zero values. To overcome the limitations of the MAPE, the mean absolute scaled error (MASE) was introduced by Hyndman and Koehler [91] in the context of time series forecasting. While it may also be suitable in the context of disparity estimation, it is not within the scope of this thesis to investigate its applicability in general. Therefore, while opening an interesting future investigation, the MASE is not considered here. Hence, the standard metrics mentioned above are used also when evaluating the performance on the validation and test dataset since no quality metric overcoming these limitations has been discussed in the literature. While the evaluation of these metrics may not be particularly nuanced in this case, an absolute comparison is possible without the limitations mentioned above when comparing these metrics for a fixed scene, *e.g.* a dataset challenge. Therefore, the performance of a given reconstruction method should not alone be judged by the performance on the test dataset but also on the individual dataset challenges.

4.4 Training and implementation details

4.4.1 Loss functions and weights

In the case of the proposed principal reconstruction, for both the central view and the disparity, the Huber loss [89] is used as the main loss function. For any $\delta > 0$, the single-element Huber loss is defined as

$$H_\delta(e) = \begin{cases} e^2 & \text{if } e < \delta, \\ 2\delta \cdot (|e| - \frac{1}{2}\delta) & \text{otherwise.} \end{cases} \tag{4.59}$$

Here, e denotes the pixel-wise prediction error with respect to the ground truth data. The overall loss is then calculated as the mean of the element-wise Huber losses, *i.e.*

$$L_{\mathrm{H},\delta}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \frac{1}{ST\Lambda} \sum_{st\lambda} H_\delta(\boldsymbol{\mathcal{I}}_{st\lambda} - \hat{\boldsymbol{\mathcal{I}}}_{st\lambda}) \,, \tag{4.60}$$

$$L_{\mathrm{H},\delta}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = \frac{1}{ST} \sum_{st} H_\delta(\boldsymbol{\mathcal{D}}_{st} - \hat{\boldsymbol{\mathcal{D}}}_{st}) \,. \tag{4.61}$$

The Huber loss is equivalent to the MSE for errors smaller than δ and proportional to the MAE for larger errors. Therefore, during training, the Huber loss is equivalent to the MSE with element-wise gradient clipping. This helps to reduce the influence of outliers when estimating the gradient during training, which is sometimes problematic in the case of the MSE, in particular for the multi-task optimization. As opposed to a pure MAE, which is often used in the case of deep disparity estimation, the MSE is convex and yields gradients that are proportional to the estimation error whereas the MAE always gives constant gradients. In the remainder, $\delta = 1$ is chosen for both the central view as well as the disparity loss. While this is well-motivated in the case of the central view, as the ground truth values are constrained to the range $[0, 1]$, the choice is somewhat arbitrary for the disparity.

For training strategies that use additional auxiliary loss terms (*cf*. Section 3.2.3), two auxiliary loss functions are considered for both the central view and the disparity estimation. To enhance the *spatial* reconstruction quality of the central view, the SSIM-based loss

$$L_{\text{SSIM}}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \frac{1}{2}\Big(1 - \text{SSIM}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}})\Big) \tag{4.62}$$

is used. To enhance the *spectral* reconstruction quality, the loss

$$L_{\text{CS}}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}}) = \frac{1}{2}\Big(1 - \text{CS}(\boldsymbol{\mathcal{I}}, \hat{\boldsymbol{\mathcal{I}}})\Big) \tag{4.63}$$

is considered, maximizing the mean cosine similarity of the spectra. Hence, the loss minimizes the spectral angle of the reconstruction.

For the disparity task, a total variation-based smoothness-enhancing auxiliary loss is developed. Using the total variation

$$\text{TV}(\hat{\boldsymbol{\mathcal{D}}}) = \sum_{st} \left(|\partial_s \hat{\boldsymbol{\mathcal{D}}}_{st}| + |\partial_t \hat{\boldsymbol{\mathcal{D}}}_{st}|\right) \tag{4.64}$$

to regularize the estimated disparity is common within the light field community. Here, the (informal) notation of the discrete partial derivative corresponds to

$$\partial_s \hat{\boldsymbol{\mathcal{D}}}_{st} = \hat{\boldsymbol{\mathcal{D}}}_{s+1\,t} - \hat{\boldsymbol{\mathcal{D}}}_{st} \tag{4.65}$$

with an analogous definition for $\partial_t \hat{\boldsymbol{\mathcal{D}}}_{st}$. However, minimizing the standard total variation often results in overly smooth estimates, in particular at disparity edges, *i.e*. occlusion boundaries. To overcome this drawback, in the case of self-supervised disparity estimation from stereo view pairs, Repala and Dubey [167] propose to weigh the total variation using the gradients of the corresponding images. This way, the total variation is weighted less in regions with large image gradients, assuming that these correlate with large gradients in the disparity, *i.e.* edges, that should be conserved. However, this approach is problematic in textured regions with constant disparity. Since passive stereo or light field disparity estimation heavily relies on texture, one can argue that this approach is not optimal probably everywhere. While no alternative exists in the self-supervised case, which Repala and Dubey consider, the ground truth disparity is available in the supervised case. Therefore, it is proposed to use the gradients of the ground truth disparity to weigh the total variation of the estimated disparity,

$$L_{\mathrm{TV}}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = \frac{1}{ST} \sum_{st} \left(\left| \partial_s \hat{\boldsymbol{\mathcal{D}}}_{st} \cdot \mathrm{e}^{-|\partial_s \boldsymbol{\mathcal{D}}_{st}|} \right| + \left| \partial_t \hat{\boldsymbol{\mathcal{D}}}_{st} \cdot \mathrm{e}^{-|\partial_t \boldsymbol{\mathcal{D}}_{st}|} \right| \right), \quad (4.66)$$

such that noise is reduced while edges in the disparity are preserved.

Furthermore, the mean disparity normal similarity, which was introduced by Hu *et al.* [88],

$$\mathrm{NS}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = \frac{1}{ST} \sum_{st} \frac{\langle \mathbf{n}_{s,t}, \hat{\mathbf{n}}_{s,t} \rangle}{\|\mathbf{n}_{s,t}\| \cdot \|\hat{\mathbf{n}}_{s,t}\|}, \quad (4.67)$$

is used by minimizing the corresponding loss

$$L_{\mathrm{NS}}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}}) = \frac{1}{2}(1 - \mathrm{NS}(\boldsymbol{\mathcal{D}}, \hat{\boldsymbol{\mathcal{D}}})). \quad (4.68)$$

The disparity surface normals are calculated as

$$\mathbf{n}_{s,t} = [-\partial_s \boldsymbol{\mathcal{D}}_{st}, -\partial_s \boldsymbol{\mathcal{D}}_{st}, 1]^{\mathrm{T}}, \quad (4.69)$$

and analogously for the estimated disparity.

Finally, as a global regularization term, layers that are preceded by batch normalization are regularized during training using the l_2-norm of the corresponding weights, coupled with a factor of 1×10^{-5}. This l_2-regularization is also known as *weight decay* in the deep learning community or as ridge regression and Tikhonov regularization in the statistics community. While weight decay has been common among practitioners for some time, it has only recently been shown empirically to be related to improved generalization in the deep learning regime [150, 164]. However, weight decay is yet not understood to full extent and is still the subject of current research [70, 124].

As detailed in Section 3.2.3, different training strategies employing different loss functions and weights are investigated for the proposed principal reconstruction. In the case of the naive multi-task approach, each task is weighted equally, *i.e.* $w_i = 0.5$, $i = 1, 2$, whereas the single-task networks are obtained by setting $w_i = \delta_{ij}$. For the remaining strategies, the weights are adaptively updated during training. An overview of the used loss weights for the different investigated training strategies is given in Table 4.2.

Concluding, as an elaborate example, the full loss function in the case of the proposed NormGradSim auxiliary loss training strategy in combination with the adaptive multi-task approach with uncertainty by Kendall *et al.* [102] is

$$L = \frac{1}{2\sigma_0^2} L_{\text{cv}} + \frac{1}{2\sigma_1^2} L_{\text{disp}} + \ln \sigma_0 + \ln \sigma_1 + L_\alpha + L_\beta + L_{\text{reg}}\,, \tag{4.70}$$

where

$$L_{\text{disp}} = \left(L_{\text{H}} + \alpha_1^{(2)} \beta_1^{(2)} L_{\text{TV}} + \alpha_2^{(2)} \beta_2^{(2)} L_{\text{NS}} \right) \Big/ \left(\sum_i \alpha_i^{(2)} \beta_i^{(2)} \right), \tag{4.71}$$

$$L_{\text{cv}} = \left(L_{\text{H}} + \alpha_1^{(1)} \beta_1^{(1)} L_{\text{SSIM}} + \alpha_2^{(1)} \beta_2^{(1)} L_{\text{CS}} \right) \Big/ \left(\sum_i \alpha_i^{(1)} \beta_i^{(1)} \right) \tag{4.72}$$

denote the individual task losses (the dependence on the prediction and ground truth has been omitted for clarity). Here, the task weights σ_i as well as the auxiliary loss weights $\alpha_i^{(j)}$ and $\beta_i^{(j)}$ are considered trainable parameters. However, $\alpha_i^{(j)}$ and $\beta_i^{(j)}$ are considered constant with respect to the individual task losses and are trained solely using the losses L_α and L_β as defined in (3.39) and (3.40). The l_2-norm weight regularization is denoted as L_{reg}.

4.4.2 Mask generation, augmentation, and seeding

4.4.2.1 Mask generation

As previously discussed, in the case of the compressed sensing-based reconstruction, random coding masks are used. For each pixel, a spectral channel index is drawn from a uniform distribution, which specifies the coordinate of the one-hot encoding of the mask, *cf.* (3.12) and (3.13).

Table 4.2 Comparison of the used multi-task weights w_{cv}, w_{disp}, and the auxiliary loss weights $w_{\mathrm{aux},j}$ for the investigated single task (ST), multi-task (MT), and auxiliary loss (AL) training strategies.

Method	w_{cv}	w_{disp}	$w_{\mathrm{aux},j}$
ST central view	1.0	0.0	–
ST disparity	0.0	1.0	–
MT naive	0.5	0.5	–
MT Uncertrainty	adapt.	adapt.	–
MT GradNorm	adapt.	adapt.	–
AL (Norm)GradSim, ST central view	1.0	0.0	adapt.
AL (Norm)GradSim, ST disparity	0.0	1.0	adapt.
AL (Norm)GradSim, MT naive	0.5	0.5	adapt.
AL (Norm)GradSim, MT Uncertrainty	adapt.	adapt.	adapt.

In the case of the proposed principal reconstruction, several masks are investigated. First, to be directly comparable with the compressed sensing approach, the same random masks are considered. However, despite the fully convolutional architecture, some care has to be taken to be able to generalize from the small spatial resolution of $(32, 32)$ of the training dataset to the full resolution of the dataset challenges as well as the real-world data. To this end, during training as well as for validation and testing, a different realization of the random coding is used for every light field in a mini-batch. This way, ideally, the latent representation only depends on the mask statistics and not a particular realization.

Second, constrained random masks are considered. For every 4×4 macropixel, a random mask is created by shuffling all available color channels. Since only 13 channels are available for the 16 possible pixels, three randomly chosen channels are repeated. This way, as opposed to the fully random mask, the maximum distance between two identical channels is seven because every channel is guaranteed to appear at least once within each macropixel.

Third, regular masks are considered, *i.e.* masks for which a fixed 4×4 macropixel is repeated. A naive regular mask, placing the channels in the macropixel consecutively, as well as a mask with an optimized macropixel layout are used. Again, because only 13 spectral channels are considered, three channels have to be repeated. Here, similar to the Bayer pattern in

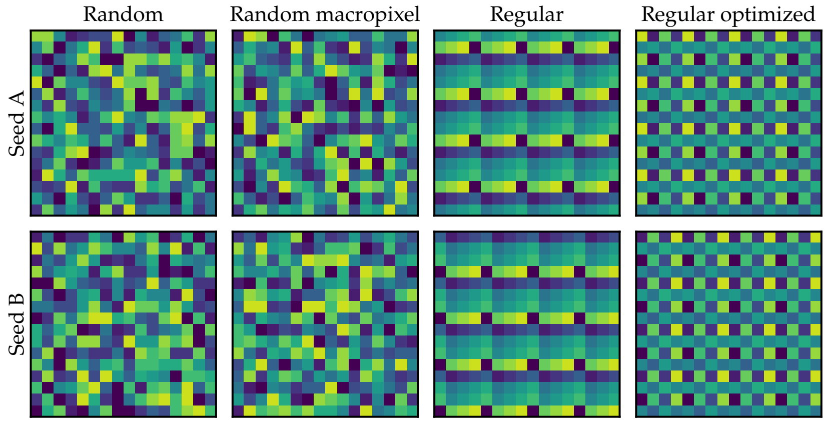


Figure 4.22 Schematic false-color crops of the investigated masks. Dark values correspond to small and bright values to large channel indices.

the RGB case, the "green" spectral channels are repeated because both the human visual system as well as the custom-built camera are most sensitive to those. The optimized macropixel layout as proposed by Shinoda *et al*. [177] is adopted here with a small modification to account for the double occurrences of the green channels. Intuitively, this optimized layout has a higher total variation than the conventional regular mask, ensuring that neighboring pixels are coded with central wavelengths that are spread out as much as possible. In the case of the regular (non-stochastic) masks, the translational equivariance is enhanced by shifting the mask by a random amount. This is done for every light field in a mini-batch. An overview of the used masks is given in Figure 4.22.

4.4.2.2 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. In the context of computer vision, common augmentations are scaling, cropping, rotation, gamma compression or stretching, color channel weighting, and color channel permutation. As opposed to conventional augmentations used in the case of 2D images, the epipolar geometry of the light field has to be retained when applying augmentation to light fields. For example, when the input light field is rescaled, the corresponding disparity has to be scaled accordingly—both spatially and in its range because the scaling effectively changes the baseline between the subapertures. That is, when spatially upsampling the light field by a factor of two, the disparity values have to be doubled. While many investigations have been conducted in the case of conventional image augmentation, leading to popular elaborate techniques such as AutoAugment [44], no such approaches exist in the case of light field deep learning. Furthermore, an adequate choice of augmentation may also depend on the investigated task, *i.e.* whether one considers classification, encoding, or disparity estimation. In fact, in a precursory study, it was found that in some instances applying augmentation may lead to decreased generalization performance [A6]. To exclude non-essential techniques with unknown outcomes, augmentations are not used in this thesis, except shuffling of the training dataset in each training epoch and a random crop to the target spatial resolution of $(32, 32)$,

4.4.2.3 Random seeds

To ensure reproducibility and comparability within the evaluation, care has to be taken in the case of random operations such as data shuffling, cropping, and light field coding. Furthermore, since the preparation of the mini-batches is typically a bottleneck during training, they are parallelized via multi-processing. To synchronize random seeds across the different data generation processes during training, the current training epoch number is used as a random seed to shuffle the dataset before each epoch. As for the random coding mask generation and cropping, the unique index of each light field in the corresponding dataset is used as the corresponding random seed during validation and testing. In particular, this guarantees that the effective test dataset, despite using a random crop and coding, is identical in all instances during the evaluation. This holds, regardless of whether the multispectral dataset or the RGB-converted dataset is used, *e.g.* when training and evaluating with state-of-the-art disparity estimation from RGB light fields. That is, the effective multispectral and RGB test datasets are geometrically identical.

4.4.3 Training and implementation

The proposed deep learning architectures, based on 3D or 4D convolutions, were trained for 170 epochs. In the case of the 3D convolution-based architecture, a mini-batch size of 128 was used, while the mini-batch size was decreased to 64 for the 4D convolutional network due to the increased memory requirements. In both cases the Yogi optimizer [233] was used, which has shown to outperform the commonly used Adam optimizer and its derivatives [103, 165] in many deep learning tasks. The learning rate was decayed from $5{\times}10^{-3}$ to $1{\times}10^{-4}$ using a sigmoid decay. In precursory experiments, which are not presented, the sigmoid decay has shown to be superior to linear, exponential, and step decay as well as the cyclical learning rate scheduling proposed by Smith [181], for the considered architectures.

The dictionary learning approach, which is investigated in the context of the compressed sensing-based reconstruction, was trained using the standard stochastic gradient descent optimizer with a learning rate of one, a momentum of 0.95, and a mini-batch size of 32. For all compressed sensing-based methods, the coupling constant (*cf*. (3.8)) was set to $\eta = 0.01$, which has shown to yield good results in preliminary experiments. Of course, the quality of the reconstruction depends to some extent on this sparsity regularisation and may not be optimal in all cases considered. However, finetuning this hyperparameter for every evaluation separately is arguably unfeasible.

For the training of EPINET, the experimental setup in the original paper [176] was followed as closely as possible. That is, the training was performed using the RMSProp optimizer with a learning rate of $1{\times}10^{-4}$ for 400 epochs. The MAE was used as the loss function. However, deviating from the original, no augmentation other than random cropping was performed in accordance with the proposed methods, as previously discussed. Naturally, EPINET was trained using the RGB-converted dataset instead of the multispectral one. Otherwise, however, the training, validation, and test data were fully equivalent to the multispectral case.

All learning-based methods, *i.e.* the proposed principal reconstruction, the disparity estimation using EPINET, and the dictionary learning-based reconstruction via compressed sensing, were trained using a single 32 GB Nvidia Tesla V100 GPU and 10 cores with 96 GB RAM of a shared GPU computing node. Inference with the deep learning methods was carried out using the same hardware. The full-size inference of the synthetic dataset challenges, as well as the real-world data in the case of the compressed sensing-based methods, could not be carried out on the GPU due to the large memory requirements. Instead, they were performed on a 40-core/80-thread computing node with 192 GB of RAM. Training took between four and seven days in the case of the 3D convolution-based architecture, depending on the used training strategy, and up to 12 days for the 4D convolutional architecture. The training of EPINET took about one day. Finally, the dictionary training converged after one epoch in about two days for both the vectorized as well as the tensor-decomposed dictionary.

All methods presented within this thesis were implemented from scratch in Python and made publicly available. This includes a general Python framework for light fields[5], for both conventional mono or RGB light fields as well as spectral light fields. The framework provides standard light field operations and conventional disparity estimation algorithms, spectral-to-color conversion, as well as calibration and decoding methods for several light field cameras including the Lytro Illum camera and the custom-built spectral light field camera. Furthermore, a TensorFlow-based framework for general light field deep learning is made available[6], containing all investigated deep learning architectures, multi-task and auxiliary loss training strategies, as well as the dictionary learning approaches. The source files of all conducted experiments can be found in the digital supplement.

[5] https://gitlab.com/iiit-public/plenpy

[6] https://gitlab.com/iiit-public/lfcnn

5 Results

As previously noted, the evaluation of all considered methods is performed on the synthetic test dataset, with a spatial resolution of $(32, 32)$. Furthermore, results are visualized for the full-sized dataset challenges with a spatial resolution of $(512, 512)$ as well as the real-world dataset with a spatial resolution of $(400, 400)$. If not specified otherwise, the angular resolution is $(9, 9)$ and the spectral resolution is (13) in all considered cases. For the sake of clarity, only a single challenge and a single scene from the real-world dataset are evaluated and shown here while the evaluation of the remaining ones can be found in the digital supplement. Here, the *Elephant* scene is chosen, as it shows an average complexity across the different challenges. For the real-world example, the *Floral* scene is used, imaged using the 250 mm focal length equivalent. For the visualization, the evaluation metrics are the PSNR in dB, the SA in °, the MSE in px^2, and the BP07 in %. The units are omitted in the plots for clarity. Moreover, the spectra of the two marked points are depicted in blue and orange together with the ground truth in grey. Note that the ground truth spectra are individually normalized for visualization. Throughout, disparity maps will be visualized using the ranges given in Table 4.1 and the colormap as shown in Figure 4.6. In particular, all disparities are shown using the same reference range and colormap within a single plot. Since no disparity ground truth data is available for the real-world dataset, the prediction of EPINET using the uncoded and RGB-converted reference light field is shown for comparison. Note that EPINET does not use zero-padding for its convolutions and the predictions are therefore cropped by 11 px from all sides. The corresponding plots hence show a grey 11 px wide margin. Furthermore, this means that for the test dataset, which only has a spatial resolution of $(32, 32)$, only the central 10×10 px patch is evaluated in the case of all EPINET-based disparity estimations, while for the proposed principal reconstruction methods the patch is estimated and evaluated at full resolution.

5.1 Compressed sensing-based reconstruction

First, the three investigated compressed sensing-based methods are evaluated. That is, the reconstruction using a fixed 5D-DCT basis, as well as the dictionary learning-based methods using the conventional vectorized as well as the proposed tensor-decomposed approach. Using the reconstructed light fields, the disparities are subsequently estimated via EPINET. As EPINET uses RGB light fields as its input, the reconstructed spectral light fields are converted to RGB beforehand.

The performances of the considered methods on the test dataset are given in Table 5.1. The overall quality of the reconstructed light fields is respectable, with a PSNR of around 28 dB for both the 5D-DCT as well as the vector dictionary approach while the proposed tensor dictionary approach performs slightly better, in particular also in the secondary metrics (SSIM and SA). This is quite remarkable, considering the huge difference in the dictionary sizes: while the conventional vector dictionary has over 865 million trainable parameters, using a patch size of $(5, 5, 8, 8, 13)$, the tensor dictionary contains only about 136 *thousand* parameters, despite the increased (separated) patch size of $(7, 7, 16, 16, 13)$. In fact, the smaller number of parameters might actually be a reason for the better performance, as the complexity of the optimization problem during the training of the dictionary is reduced drastically. As previously discussed, the two approaches are otherwise algorithmically equivalent.

However, as compared to the vanilla EPINET prediction using the uncoded reference light fields, a significant drop in the quality of the disparity estimation can be observed using the reconstructed light fields for all compressed sensing-based methods. While this drop is particularly severe in the case of the 5D-DCT-based reconstruction, also the dictionary-based methods perform poorly, however, the quality of the estimated disparities in those instances is considerably better in comparison. This is likely due to the fact that the dictionaries implicitly learn a representation of the spectral light fields that is in accordance with the underlying epipolar geometry which the 5D-DCT does not consider at all.

A similar trend can be observed in the case of a dataset challenge and a real-world scene as shown in Figure 5.1. Yet, in particular, the disparity estimation of the dictionary-based methods is much better than what is to be expected from the test dataset performance with an MSE of around

Table 5.1 Test dataset performance of the compressed sensing (CS)-based reconstruction methods. The performance of EPINET using the original, uncoded data is shown for reference but excluded from the best-value comparison.

Method	Central view			Disparity		
	PSNR/dB	SSIM	SA/°	MSE/px^2	MAE/px	BP07/%
EPINET [176] (uncoded)	-	-	-	0.0881	0.0626	6.28
CS 5D-DCT + EPINET	27.48	0.75	7.39	1.3671	0.4875	41.65
CS Vec.-D. + EPINET	27.86	0.74	7.25	0.6050	**0.2939**	**32.34**
CS Tens.-D. + EPINET	**29.39**	**0.78**	**6.81**	**0.5913**	0.2959	35.28

0.02 px^2 for both considered methods. (However, this is not the case for all dataset challenges.) While these disparities still show some artifacts, in particular a noticeable speckle, the results are much better as opposed to the disparity estimated from the 5D-DCT-based reconstruction. In the case of the real-world example, the compressed sensing-based methods surprisingly show much higher qualities of the reconstructed central views with a PSNR of up to 37 dB. For both the dataset challenge and the real-world scene, the 5D-DCT and the vector dictionary-based reconstruction show a noticeable blur while the reconstruction using the tensor dictionary appears much sharper which is also reflected in the higher SSIM value. While both considered dictionary-based methods perform similarly in the case of the test dataset and the synthetic dataset challenge, with respect to the subsequent disparity estimation, the proposed tensor dictionary-based reconstruction results in a qualitatively better disparity estimation in the case of the real-world example. (This can also be observed for those scenes which are not shown here.) This effect may be caused by the larger effective patch size which should be able to adequately represent the epipolar geometry on a larger scale.

Overall, the proposed approach using a tensor dictionary performs best among the compressed sensing-based methods and is therefore used for comparison in the remainder. Finally, it should be noted that the reconstruction of a single full-sized dataset challenge takes between 3 to 5 h, depending on the used method, despite a highly parallelized implementation using a 40-core computing node with 192 GB of RAM. GPU-based inference times using the test dataset will be compared for all investigated methods shortly.

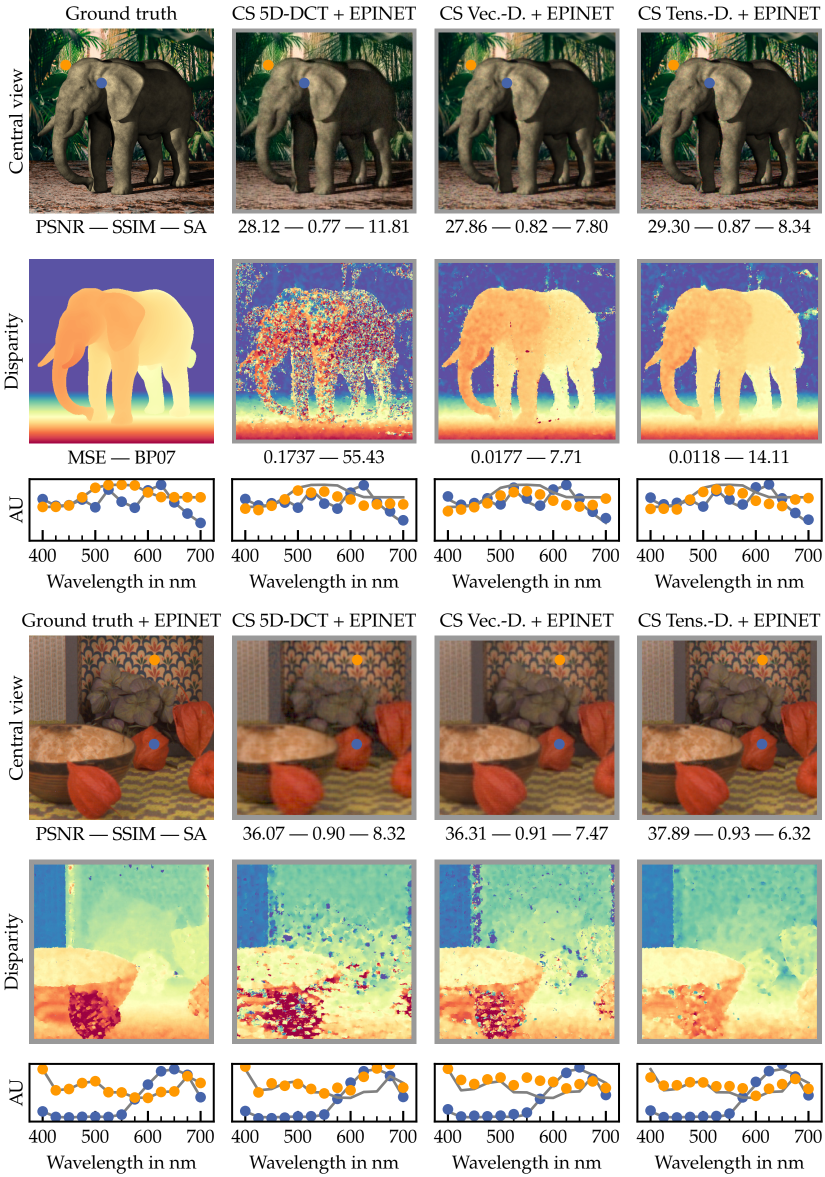


Figure 5.1 Performance comparison of the compressed sensing-based reconstructions for a synthetic dataset challenge (top) and a real-world example (bottom).

5.2 Principal reconstruction

Next, the proposed principal reconstruction is evaluated and several ablation studies are performed. First, the different training strategies are compared using random coding masks, identical to those used for the compressed sensing-based methods. Using the best training strategy, further investigations, such as the dependence on noise or the angular resolution, are presented. Second, different coding schemes are evaluated using the best-performing training strategy. Throughout, results are obtained using the 3D convolution-based network if not mentioned otherwise. The network based on 4D convolutions is evaluated using only the best training strategy. Due to the large number of investigated strategies, as well as the long training time of the 4D convolution network as compared to the one based on 3D convolutions, evaluating both networks for all considered methods is arguably excessive. Since the overall difference in performance between the 4D and the 3D convolution-based architectures is comparably small, as will be shown, the full evaluation would be unreasonable also in terms of the used computational resources and energy: Conservatively estimating the energy usage by only considering the consumption of the GPU, which is 300 W in the case of the used Nvidia V100, assuming a mean training time of seven days in the case of the 4D convolutional network, training of all twenty investigated methods would require more than 1000 kW h. This corresponds to over one-third of the yearly power consumption of an average household in Germany in 2018 [25]. This is also the reason why the investigated networks are only trained and evaluated once, if not stated otherwise. Of course, it would be good practice to perform training and evaluation multiple times, *e.g.* by using k-fold cross-validation, and presenting averages and possibly standard deviations. However, as outlined above, the required computational and energy resources are out of proportion. Moreover, besides the presented results, some networks were trained and evaluated multiple times showing only negligible fluctuations of the results, suggesting that the convergence does not strongly depend on the initialization and other random operations during training in the considered case. While the ecological and economic impact of deep learning is a research area of its own [173] and is not further discussed here, it is worth keeping its order of magnitude in mind.

Table 5.2 Test dataset performance of the adaptive auxiliary loss training strategies for both single-task and multi-task inference.

Method	Central view			Disparity		
	PSNR / dB	SSIM	SA / °	MSE / px^2	MAE / px	BP07 / %
ST Naive	32.68	0.92	5.39	0.0607	0.0679	14.23
ST AL GradSim [56]	32.83	**0.95**	4.66	0.0866	0.0911	19.09
ST AL NormGradSim	**33.47**	0.94	**4.52**	**0.0562**	0.0679	14.93
MT Naive	27.70	0.85	8.85	0.0626	0.0697	14.92
MT AL GradSim [56]	29.54	0.93	6.63	0.0727	0.0847	18.67
MT AL NormGradSim	28.52	0.89	7.77	0.0569	**0.0660**	**14.04**

5.2.1 Adaptive auxiliary loss approaches

Several ablation studies are performed in the case of the proposed principal reconstruction, investigating single-task versus multi-task performance, adaptive multi-task training strategies, and adaptive auxiliary loss strategies, as introduced in Section 3.2.3. Throughout, methods names are chosen to reflect the used training setup: ST for single-task, MT for multi-task, and AL for auxiliary loss training strategies.

First, the auxiliary loss methods are investigated. The test performance of the investigated auxiliary loss approaches for both the single- and the multi-task case are shown in Table 5.2. Note that, while the single-task performances are presented in a single column, the results are obtained using two separately trained networks—one for estimating the central view and one for the disparity. It can be observed that both auxiliary loss methods, the GradSim approach by Du *et al*. [56] and the proposed NormGradSim approach, lead to an improved quality of the reconstructed central view both in the single-task and the multi-task case. In particular, it is remarkable that the auxiliary losses not only lead to an improvement of the secondary metrics, *i.e*. the SSIM and the SA, which the auxiliary losses are based on, but also of the primary metric, corresponding to the main loss, as reflected by an improvement of the PSNR as compared to the naive baselines. That is, the introduction of suitable auxiliary losses together with an adaptive weighting scheme does show the desired regularizing effect.

In the case of the estimated disparity, however, the GradSim approach performs significantly worse than the naive single-task baseline. As this effect is alleviated by the proposed normalized approach, the auxiliary losses likely yield gradients with a much larger norm than the main disparity loss which disturbs the overall gradient estimate leading to a worse overall performance. The proposed NormGradSim approach resolves this issue, however, in the case of the estimated disparity, the improvements over the baseline are small to moderate. Still, using auxiliary losses, the main disparity loss is also improved, as reflected by the MSE scores which are improved by about 7 to 8 %. Yet, the situation is slightly different from the central view estimate: the auxiliary losses do not directly correspond to secondary metrics that one might be interested in, as is the case for the central view using auxiliary losses based on the SSIM and the SA directly. More precisely, recall that the auxiliary disparity losses are based on the total variation and the normal similarity which are not directly reflected in the evaluation metrics. Still, while the auxiliary losses only moderately improve the disparity estimate, the overall results are very good, similar to the performance of the vanilla EPINET as shown in Figure 5.1. In fact, the disparity estimate via principal reconstruction from spectrally coded light fields even outperforms EPINET in terms of the MSE, however at the cost of a higher BP07 score. Therefore, the proposed approach shows more but less severe outliers than the estimate from EPINET. Again, recall that the results for EPINET are obtained from uncoded RGB light fields and only evaluated on the central 10×10 px patch as opposed to the proposed approach.

In the case of the multi-task approaches using auxiliary losses, the effect is slightly different. While the GradSim approach again shows a significant improvement of the primary as well as the secondary metrics in the case of the estimated central view, this again comes at a cost of a significantly worse disparity estimate. This is resolved once more with the developed normalized approach which improves all primary and secondary metrics as compared to the naive multi-task baseline. In the case of the estimated disparity, the multi-task approach using auxiliary losses with NormGradSim even performs slightly better than the corresponding single-task disparity method.

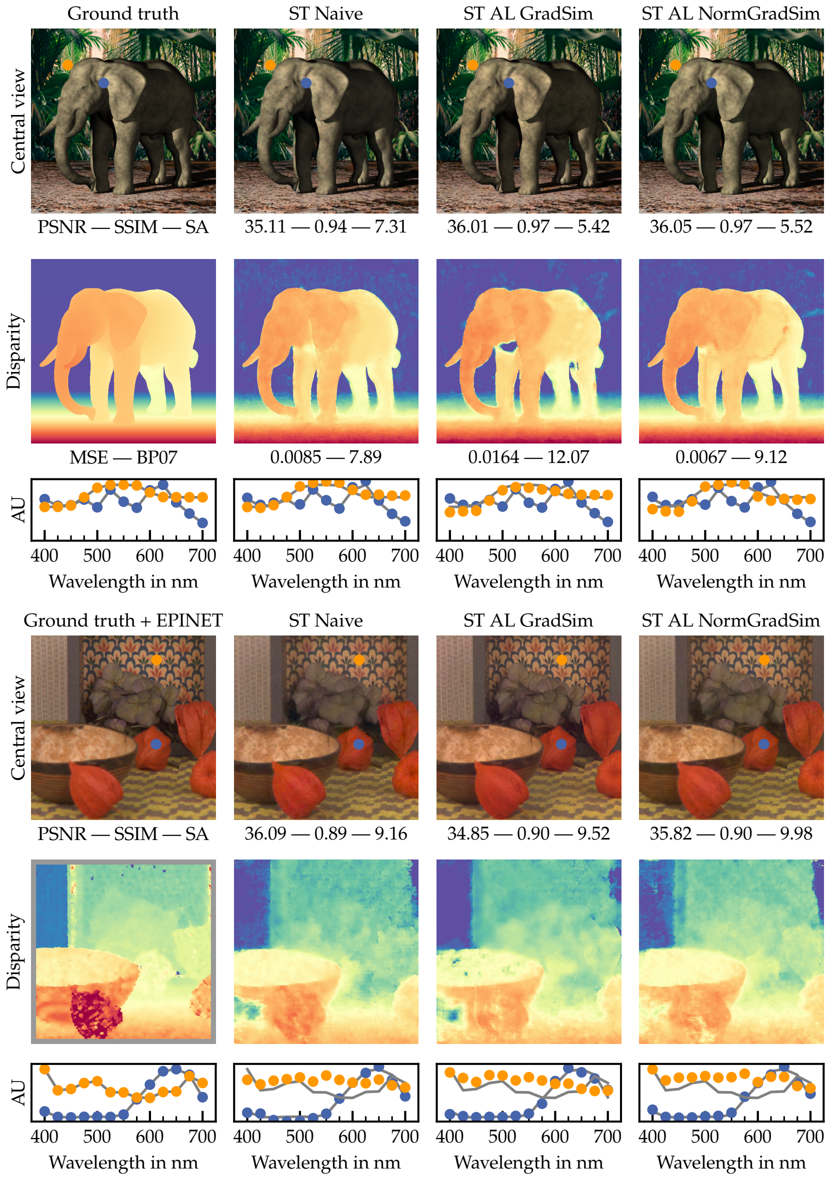


Figure 5.2 Performance comparison of the single-task reconstruction with adaptive auxiliary loss training for a synthetic challenge (top) and a real-world example (bottom).

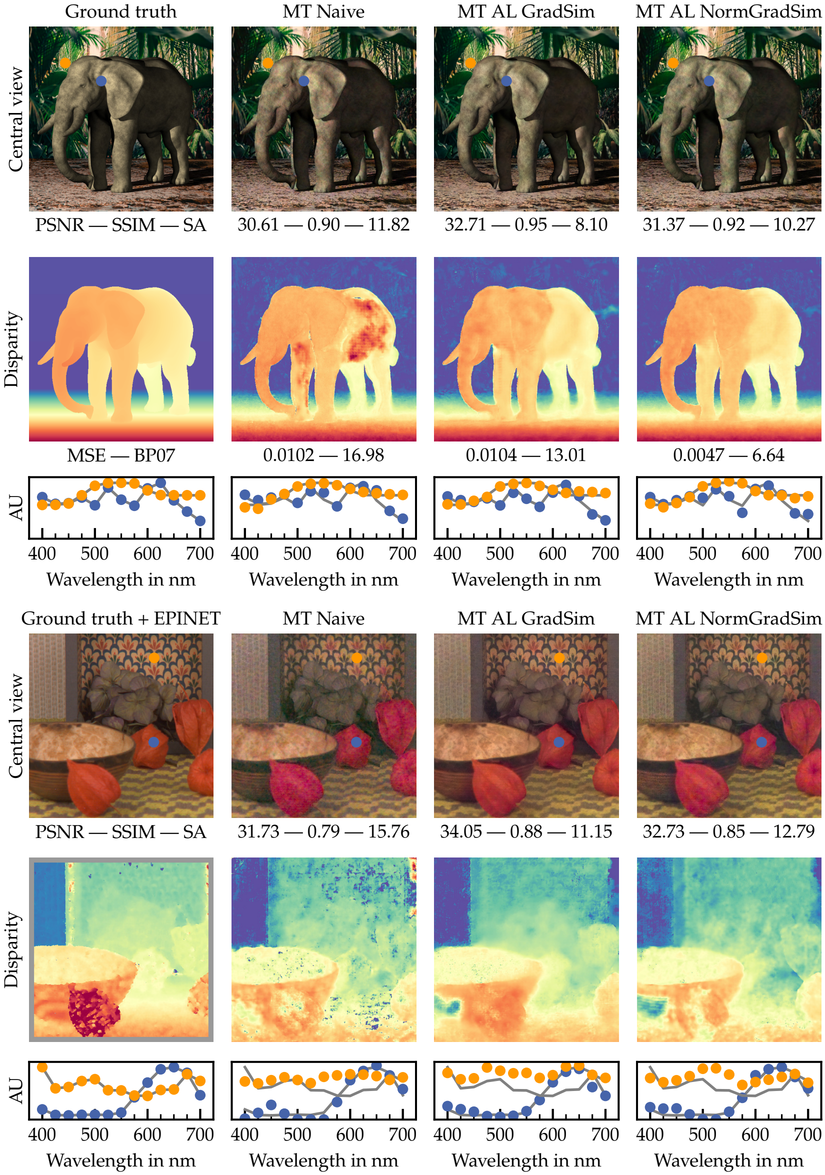


Figure 5.3 Performance comparison of the multi-task reconstruction with adaptive auxiliary losses for a synthetic challenge (top) and a real-world example (bottom).

The performances using a synthetic dataset challenge as well as a real-world example are shown in Figure 5.2 for the single-task and in Figure 5.3 for the multi-task scenario. The results are similar to those achieved on the test dataset: Using auxiliary losses improves the central view estimate with respect to the primary as well as the secondary evaluation metrics, achieving a central view PSNR of up to 36 dB for the considered dataset challenge as well as the real-world light field, and a disparity MSE of around 0.005 to 0.001 px^2 which, again, is even lower than what is expected from the results on the test dataset. In the case of the real-world example, the improvements over the baseline are particularly strong in the multi-task scenario. Again, the GradSim approach, while improving the central view estimate, yields worse disparity estimates as compared to the baseline whereas the proposed NormGradSim approach resolves this issue and yields the overall best result, in particular in the multi-task case.

Overall, however, the multi-task approaches show a significant drop in performance with respect to the estimated central view as compared to the single-task estimates, regardless of the used dataset. To this end, the adaptive multi-task training strategies are investigated in the following.

5.2.2 Adaptive multi-task approaches

To improve the multi-task performance, two methods are considered, as discussed in Section 3.2.3: the multi-task learning approach using uncertainty by Kendall *et al*. [102] and the GradNorm method by Chen *et al*. [39]. The results for the test dataset are given in Table 5.3 and visualized for the dataset challenge and the real-world example in Figure 5.4. Both approaches improve the performance of the central view as well as the disparity estimation significantly as compared to the naive multi-task approach. While, on the test dataset, the GradNorm method performs slightly better with respect to the estimation of the central view, the approach using uncertainty performs very well in the case of disparity estimation—even outperforming the single-task network—while also improving the central view estimate as compared to the naive multi-task baseline. Judging by the quality of the estimates using the synthetic dataset challenge as well as the real-world example, the situation is slightly different. In these cases, the GradNorm approach shows very

Table 5.3 Test dataset performance of the adaptive multi-task training strategies.

Method	Central view			Disparity		
	PSNR/dB	SSIM	SA/°	MSE/px^2	MAE/px	BP07/%
ST Naive	**32.68**	**0.92**	**5.39**	0.0607	0.0679	14.23
MT Naive	27.70	0.85	8.85	0.0626	0.0697	14.92
MT GradNorm [39]	31.95	0.92	5.80	0.0692	0.0815	19.07
MT Uncertainty [102]	31.18	0.91	6.27	**0.0594**	**0.0665**	**13.87**

good estimates, both for the central view and the disparity, while the approach using uncertainty suffers from some artifacts in the estimated disparity. It should be noted that this is not consistent throughout all dataset challenges and real-world examples, as shown only in the digital supplement. While GradNorm performs better in some, the approach using uncertainty outperforms GradNorm in others. However, again, both methods show significant improvements over the naive multi-task baseline, almost achieving the single-task performance in the case of the estimated central view and slightly outperforming the single-task disparity estimate.

Finally, it should be pointed out that GradNorm introduces an additional asymmetry hyperparameter which was manually fine-tuned for the obtained results by evaluating multiple training runs using different choices of this parameter. The approach using uncertainty on the other hand is hyperparameter-free which is favorable in practice, saving both time and computational resources. Therefore, in the following, only the approach using uncertainty by Kendall *et al.* [102] is used as the adaptive multi-task training strategy.

5.2.3 Reconstruction using random coding masks

To draw a first conclusion, the best-performing reconstruction methods, using random coding masks, are compared. In addition to the previously discussed methods, the multi-task training using uncertainty is combined with the proposed NormGradSim approach utilizing auxiliary losses. The results are given in Table 5.4 and visualized partially in Figure 5.5.

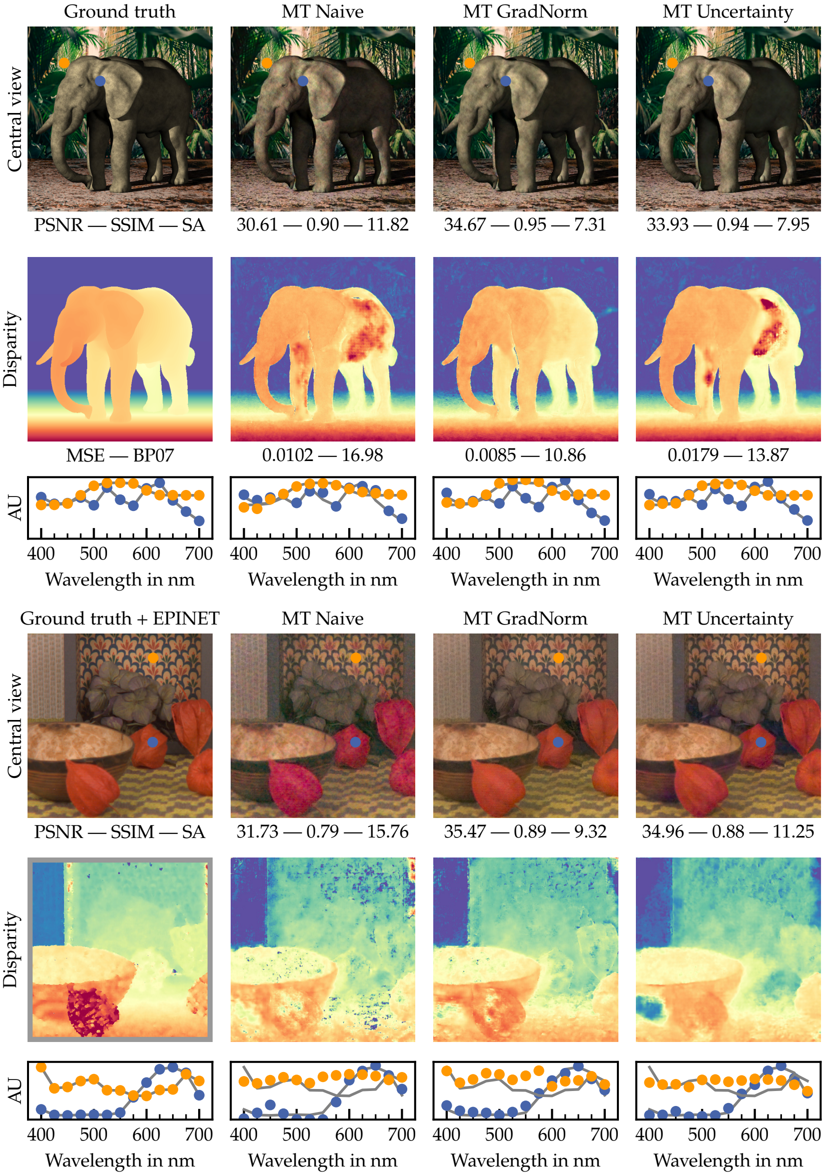


Figure 5.4 Performance comparison of the multi-task reconstruction with naive and adaptive weighting for a synthetic challenge (top) and a real-world example (bottom).

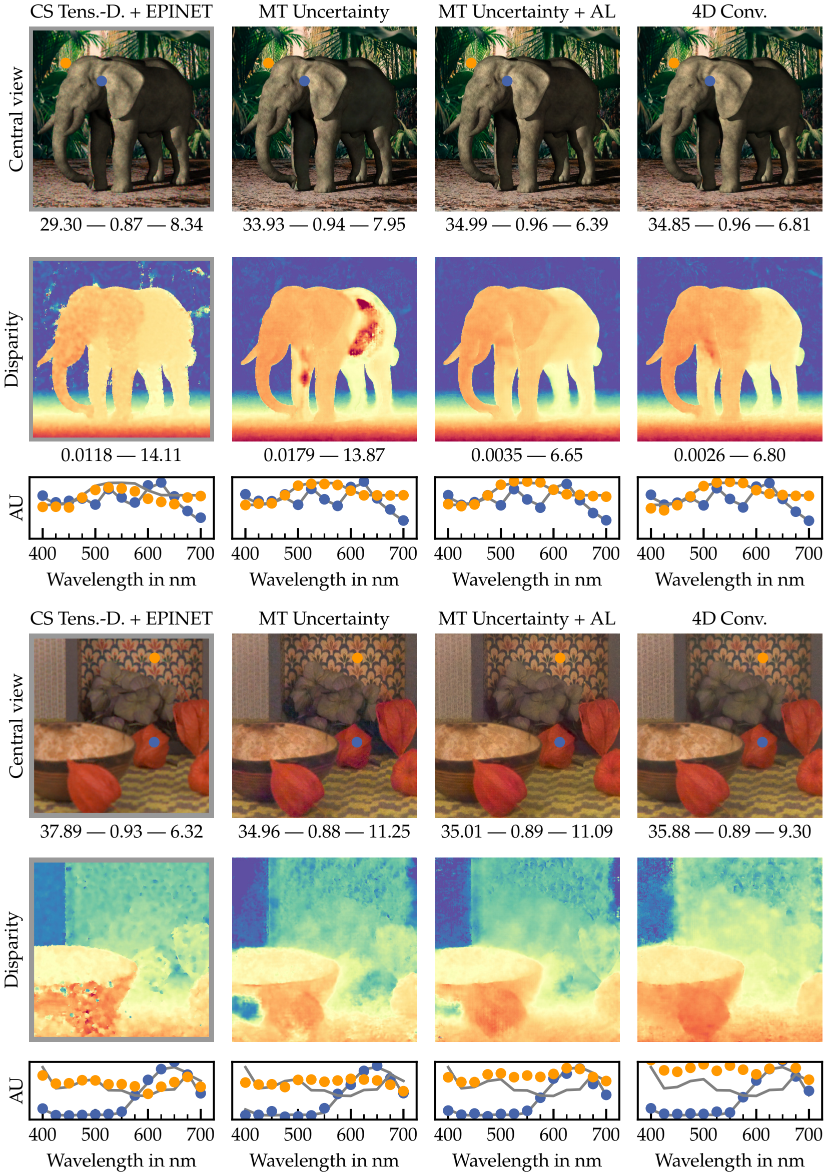


Figure 5.5 Performance comparison of the best reconstruction methods using random coding masks. The two multi-task results are obtained using the 3D convolution network.

Generally, the proposed principal reconstruction outperforms the best compressed sensing-based reconstruction using a tensor dictionary with subsequent disparity estimation in all considered evaluation metrics. In particular, the quality of the estimated disparity is far superior to those estimated from the compressed sensing-reconstructed light fields. While this holds not only for the test dataset, the differences in the case of the dataset challenge and the real-world example are not as severe. In fact, for the real-world example, the compressed sensing-based reconstruction yields the overall best central view estimate, reaching a PSNR of nearly 38 dB, and a visually sharper but more disturbed subsequent disparity estimate. However, this gap will be closed by using non-random coding masks in the case of the proposed principal reconstruction, which is evaluated in Section 5.3.

Considering the different investigated training strategies of the principal reconstruction methods, the combined adaptive multi-task training using uncertainty together with the proposed adaptive auxiliary loss strategy using normalized gradient similarity yields the best reconstruction among the multi-task approaches—even outperforming the single-task approach with respect to the estimated disparity while only performing slightly worse in the case of the estimated central view. Using the elaborate adaptive multi-task and auxiliary loss weighting, the multi-task performance is nearly on-par with the naive single-task performance and performs only slightly worse than the single-task network using adaptive auxiliary losses in the case of the estimated central view. However, the multi-task approaches are much more parameter-efficient and show shorter inferences times as compared to the two separately trained single-task networks as discussed shortly. This result is quite remarkable, considering that the deep learning-based approaches differ only in the used training strategy while the network architectures are identical. While, in the applied computer vision community, deep learning-based applications are often trained using comparably naive loss functions and training strategies, in particular in multi-task applications, these results demonstrate the potential of more elaborate adaptive training strategies.

Table 5.4 Test dataset performance comparison of the best-performing methods using random coding masks.

Method	Central view			Disparity		
	PSNR / dB	SSIM	SA / °	MSE / px^2	MAE / px	BP07 / %
EPINET [176] (uncoded)	-	-	-	0.0881	0.0626	6.28
CS Vec.-D. + EPINET	27.86	0.74	7.25	0.6050	0.2939	32.34
ST AL NormGradSim	**33.47**	**0.94**	**4.52**	**0.0562**	0.0679	14.93
MT Uncertainty [102]	31.18	0.91	6.27	0.0594	0.0665	13.87
MT U. + AL NormGradSim	31.94	0.94	5.36	0.0567	**0.0656**	**13.58**

Table 5.5 Test dataset performance comparison of the two different investigated network architectures. In both cases, training was performed using the adaptive multi-task training with uncertainty and the proposed NormGradSim auxiliary loss method.

Method	Central view			Disparity		
	PSNR / dB	SSIM	SA / °	MSE / px^2	MAE / px	BP07 / %
3D Conv.	31.94	0.94	5.36	0.0567	0.0656	**13.58**
4D Conv.	**32.55**	**0.94**	**5.03**	**0.0491**	**0.0636**	13.69

5.2.3.1 Network architecture

Using the best-performing training strategy, the performance of the architecture based on 3D convolution, which was exclusively considered so far, is compared to the one based on 4D convolution, as introduced in Section 3.2.2. Both architectures are trained with the same strategy, *i.e.* the multi-task training using uncertainty together with the auxiliary losses using NormGradSim. The results are depicted in Table 5.5 and in Figure 5.5. Generally, the performance of the two considered architectures is quite similar, in particular considering the reconstructed central view. However, the architecture based on 4D convolution performs noticeably better with respect to the estimated disparity, improving the test dataset MSE by more than 10 %, from 0.0567 px^2 to 0.0491 px^2. While this can also be observed using the dataset challenge as well as the real-world example, the differences are not as severe in these instances. Investigating the real-world example, the 4D convolution network seems to estimate a visually sharper disparity map, especially for objects close to the camera. However, this is difficult to judge quantitatively due to the lack of disparity ground truth data.

While the network based on 4D convolution can be considered superior to the one based on 3D convolution, with respect to both the reconstructed central view and especially the estimated disparity, it is also more resource-demanding. This results in considerably longer training and inference times as discussed shortly. Therefore, in the following, the 4D convolutional architecture is not further evaluated. However, depending on the application, and its corresponding accuracy and speed requirements, the 4D architecture may be more suitable in some cases.

5.2.3.2 Generalization gap

While the performance of the proposed principal reconstruction approaches is in general very good, all deep learning-based methods suffer from a generalization gap when performing inference on the real-world data. While this is not specific to the proposed approach, it seems to be more severe as compared to, for example, the EPINET prediction. Since this gap is likely due to noise, both in the conventional sense, *i.e.* sensor noise, as well as distortions in the epipolar geometry, *i.e.* "calibration noise", it is understandable that the effect is less severe in the case of the EPINET prediction as the data is converted to RGB beforehand, smoothing both the sensor and calibration noise. In any case, closing this generalization gap in data-driven approaches that are trained using synthetic data is non-trivial. While it is straightforward to introduce sensor noise statistics to the training process, either via data augmentation or, more generally, via noise injection techniques [72] such as DropOut [182] or WhiteOut [115], taking into account calibration inaccuracies is much more difficult. To this end, self-supervised approaches, directly using (non-labeled) real-world data, offer a great advantage as the statistics of the training data and the data that is used for the inference in practice are identical by design. While self-supervised approaches to light field disparity estimation exist, *e.g.* using a self-consistency loss obtained by warping the central view onto the original subapertures using the estimated disparity [160], these approaches are problematic at occlusion boundaries. Furthermore, it is not clear how a self-supervised approach could be formulated using spectrally coded light fields. Concluding, closing the generalization gap between synthetic and real-world data inference in the context of light field deep-learning remains an open challenge. The influence of sensor and calibration noise on the used reconstruction method is investigated in more detail in Section 5.2.4.

5.2.3.3 Inference times

Finally, the inference times for the different investigated methods are compared. As previously hinted, the inference time for the different investigated methods cannot be fairly compared in the case of full-sized inference since the compressed sensing-based methods cannot be evaluated on a GPU due to the immense memory requirements. However, for light fields in the test dataset, with a smaller resolution of $(9, 9, 32, 32, 13)$, a comparison can be made for all but the 5D-DCT-based reconstruction for which a GPU implementation was not created. The mean inference times for a single test dataset light field are shown in Table 5.6. The values are collected across the full test dataset using an individually chosen mini-batch size to maximize the GPU load. To calculate the average, the corresponding mini-batch size and the total number of batches are taken into account. Overall, while the proposed tensor dictionary approach is about four times faster than the conventional vector-based method, it can be observed that the deep learning-based methods are orders of magnitudes faster than the reconstructions based on compressed sensing. This is not particularly surprising, as the compressed sensing-based methods reconstruct the full multispectral light field which is a higher-dimensional problem than estimating the central view and disparity from the coded measurement directly. Furthermore, the deep learning-based approaches are likely to gain more from the GPU acceleration as opposed to the compressed sensing methods, in particular due to the involved light field patching which is time- and memory-consuming. While the inference using the vanilla EPINET, which solely predicts the disparity from uncoded RGB light fields, is the fastest, the proposed principal reconstruction performs adequately, especially considering the much higher complexity of the reconstruction. For the principal reconstruction, minor differences between the single-task and the multi-task models can be observed. In particular, the multi-task inference is more than 40 % faster than the combined single-task prediction of the central

Table 5.6 Mean inference time per light field across the test dataset for the different investigated methods using an Nvidia V100 GPU.

Method	Inference time / ms
EPINET [176] (uncoded)	0.34
CS Vec.-D.	945.41
CS Tens.-D.	233.39
ST 3D Conv. (central view)	1.67
ST 3D Conv. (disparity)	1.40
MT 3D Conv.	1.75
MT 4D Conv.	7.03

view and the disparity map. Comparing the multi-task networks based on 3D and 4D convolution, an increase of about 400 % can be observed which is more than what is expected from the previous analysis as shown in Table 3.2.

However, it should be noted that the measurements also include the time needed for the data generation as well as copying the data from the CPU RAM to the GPU RAM to some extent. While the data generation is performed via multiprocessing, the CPU and PCIe transfer speed may become the bottleneck due to the large size of the spectral light fields, increasing the effective inference times. Due to the CPU multiprocessing and the GPU-based inference, this effect is difficult to quantize but it is assumed to be small considering the significant difference in the inference between the 3D and the 4D convolution-based reconstruction which should show a similar overhead. Moreover, this effect is likely much smaller in the case of the vanilla EPINET whose prediction uses RGB light fields which are more than four times smaller in size compared to their multispectral counterparts, reducing the overhead caused by the data generation.

Concluding, the proposed principal reconstruction using the architecture based on 3D convolution, trained with an adaptive multi-task strategy using uncertainty combined with the proposed adaptive auxiliary loss strategy NormGradSim, yields the overall best result—in terms of the considered evaluation metrics, as well as the computational effort as reflected by the shorter inference times. Hence, it is the only approach considered in the following.

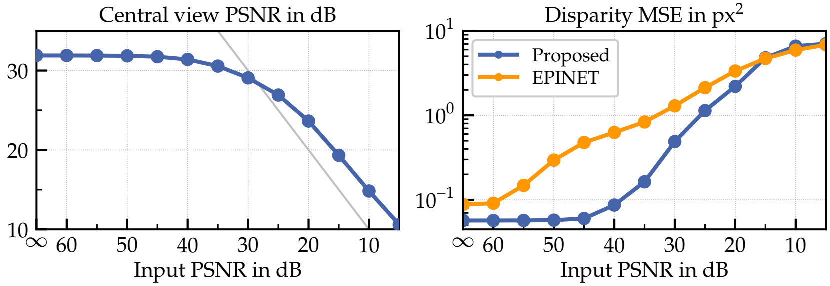


Figure 5.6 Test dataset performance for the proposed principal reconstruction from spectrally coded light fields as well as EPINET, using uncoded RGB light fields, in the case of varying input noise levels. The symbolic input PSNR of infinity corresponds to the original (possibly coded) input without additional noise.

5.2.4 Noise, angular resolution, and depth

5.2.4.1 Dependence on sensor noise

As previously noted, the generalization gap between synthetic and real-world data may stem from the disregard of noise during training. While inaccuracies of the geometric calibration are difficult to simulate, investigating the dependence on sensor noise is feasible. To this end, pixel-wise independent Gaussian noise with different standard deviations is added to the input light fields and the reconstruction performance is measured using the test dataset. In the case of the estimated disparity, the results are also compared to the EPINET prediction from noisy input data. Note that the trained network instances are the same as before, *i.e.* the networks were not exposed to any additional noise during training. The results are shown in Figure 5.6 for both the proposed approach using spectrally coded light fields and EPINET in the case of uncoded RGB light fields. Generally, the proposed approach is quite robust with respect to input noise for input PSNRs of 40 dB and higher. For lower values, the performance suffers, both for the central view and the disparity estimate. Nevertheless, especially for very high noise levels, with corresponding PSNRs below 30 dB, the proposed network actually improves the overall PSNR of the central view which can be seen as a form of denoising. This is not particularly surprising. In fact, the reconstruction from spec-

Table 5.7 Test dataset performance of the principal reconstruction for different angular resolutions of the input light field.

Ang. resolution	Central view			Disparity		
	PSNR/dB	SSIM	SA/°	MSE/px^2	MAE/px	BP07/%
3×3	31.76	0.93	5.44	0.0722	0.0849	21.19
5×5	32.41	0.94	**5.13**	0.0631	0.0798	20.23
7×7	**32.49**	**0.94**	5.13	0.0595	0.0712	15.61
9×9	32.28	0.94	5.20	**0.0569**	**0.0675**	**14.14**

trally coded light fields can be interpreted as a severe form of denoising, where the input light fields are corrupted by strong impulse noise, *i.e.* pepper noise. On the other hand, the quality of the estimated disparity quite rapidly decreases for input PSNRs below 40 dB. Still, the proposed approach is much more robust against noise as compared to EPINET whose performance rapidly declines for PSNRs below 60 dB. Again, this probably stems from the fact that the proposed network, unlike EPINET, is in fact exposed to a form of noise during training, even though the noise statistics are severely different from the ones explicitly modeled here. Overall, the results suggest that some potential exists to further increase the network's robustness, potentially increasing the performance on real-world data, *e.g.* by incorporating sensor noise into the training, as discussed previously. However, the real-world performance is more likely limited by the inaccuracies of the calibration as shown in the following.

5.2.4.2 Dependence on angular resolution

Second, the dependence of the principal reconstruction on the input's angular resolution is investigated. To this end, the proposed network is trained and evaluated using different angular resolutions of the coded input light fields, ranging from 3×3 to 9×9, which are centrally cropped from the higher-resolution original dataset. The results are given in Table 5.7 and visualized in Figure 5.7.

While the disparity shows a decreasing estimation error with larger angular resolutions, which is expected as more geometric information is available with higher angular resolutions, the central view estimate is mostly independent of the used angular resolution, in particular for

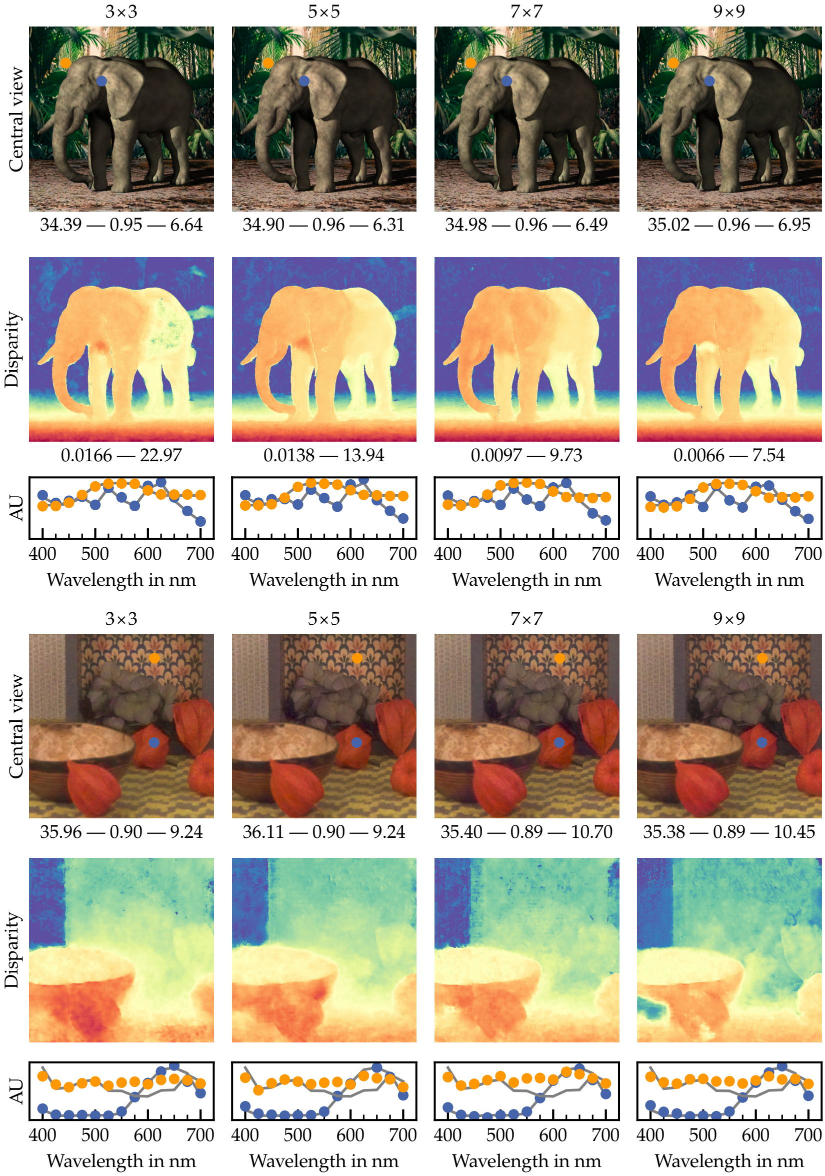


Figure 5.7 Performance comparison of the reconstruction for different angular resolutions of the coded input light field.

those larger than 3×3, in the case of the synthetic test dataset and the dataset challenge. Considering the sparsity of the input light fields, it is quite remarkable that the central view and disparity can be estimated with respectable quality even for the comparably low angular resolution of 3×3. However, this is presumably also dependent on the number of used spectral channels as the sparsity increases when more channels are considered. In the case of the used 13 spectral channels, an angular resolution of 9×9 yields good results for both the reconstructed central view and the estimated disparity. The benefits of further increasing the angular resolution are likely outweighed by the increased computational and memory requirements.

In the case of the real-world example, however, the lower angular resolutions of 3×3 and 5×5 actually show the best performance. While the central view estimates are mostly independent of the angular resolution, analogous to the evaluation of the synthetic data, the quality of the disparity estimate seems superior in those cases. (Of course, since no ground truth disparity is available, this judgment can only be made qualitatively.) This result supports the previous claim that the reconstruction performance in the case of the real-world example is likely limited by inaccuracies of the geometric calibration. Naturally, these inaccuracies are more severe in peripheral subapertures (*cf.* Appendix B) leading to a worse performance when using larger-resolution crops of the original real-world light fields. While decreasing the angular resolution mitigates this issue, it is not really an option in practice as only a fraction of the true sensor resolution would be utilized. Since the calibration inaccuracies mostly stem from deviations of the used camera model at microlens boundaries, decreasing the actual angular resolution of the camera, *e.g.* by decreasing the microlens radius or increasing the pixel size, is not a viable solution. Hence, to fully leverage the potential of the proposed approach in practice, a more accurate geometric calibration is needed. While recent model-free generalized calibration methods, as shortly discussed in Section 4.2.3, are promising, it is not clear how the ray resampling in the case of a spectrally coded light field camera can be achieved. Therefore, this opens new possible directions for future research, which are not within the scope of this thesis.

5.2.4.3 Dependence on scene depth

Finally, the dependence on the scene's depth—or, equivalently, its corresponding disparity—is investigated. To this end, two approaches are considered. On the one hand, the *Wall* challenges are evaluated, which are explicitly designed to test performance versus disparity. Recall, the *Wall* challenges consist of a perfectly flat surface with a multispectral image texture that is rendered at several constant distances corresponding to disparities from −1.5 to 1.5 px in 0.25 px steps.[1] Therefore, the *Wall* challenges consist of 13 separate light fields that are reconstructed and evaluated independently, each corresponding to a single ground truth disparity. On the other hand, a similar evaluation can be performed using a single light field with available disparity ground truth. To this end, the pixel-wise reconstruction errors, for both the central view and the disparity, are grouped using the underlying ground truth disparities. The mean errors are subsequently calculated on the separate groups allowing for a disparity-dependent evaluation. The grouping is performed using disparity bins with a width of 0.1 px and bin centers coinciding with the ones used for the *Wall* challenges. For example, all pixels with a ground truth disparity between −0.05 px and 0.05 px are grouped and their corresponding (pixel-wise) reconstruction errors are collected and averaged to yield the measurement for the bin corresponding to a disparity of 0 px. While this allows for an evaluation using a scene with a more realistic geometry, it should be noted that the resulting statistics are disparity-dependent. In particular, there may be bins with significantly fewer measurements than others, depending on the ground truth disparity distribution. The results are shown in Figure 5.8 for the *Wall* challenge and the *Cabin* challenge which has the largest disparity range amongst the dataset challenges.

Investigating the result for the *Wall* challenges, a clear effect can be observed: Both the central view and the disparity estimates are considerably worse in the case of a ground truth disparity of exactly 0 px while the performances for the remaining disparities are similar, with a small positive trend towards larger absolute disparities in the case of the esti-

[1] The challenge in the case of a disparity of 1 px is excluded from the evaluation because the corresponding light field was corrupted during rendering.

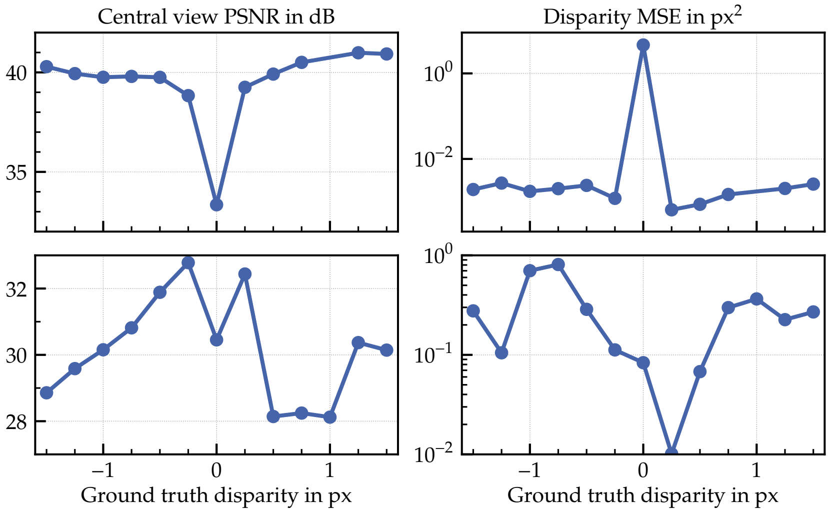


Figure 5.8 Dependence of the reconstruction performance on the ground truth disparity in the case of the *Wall* challenges (top) and the *Cabin* challenge (bottom).

mated central view. In fact, this effect is expected from the camera design: Since the used coding of the microlenses leads to an angular-independent coding of the light field, all subaperture views are coded using the same spatio-spectral mask. However, in the case of a scene with a disparity of exactly zero, all subapertures are in fact identical. Therefore, the available information in the coded light field is reduced to a minimum and the reconstruction problem becomes the most challenging. Strictly speaking, this argument is valid for the reconstruction of the central view, while the disparity estimation should still be feasible. After all, the observation of identical subaperture views directly leads to the corresponding disparity estimate—at least this would be expected in the case of uncoded light fields. Using coded light fields, however, the disparity estimation becomes less intuitive due to the sparse observation of the epipolar geometry. For example, one cannot directly observe lines in the epipolar images (or planes in the epipolar volumes) as neighboring pixels in the light field are coded using different spectral channels. Therefore, the disparity is extracted from the latent representation in some other way.

However, when the overall information of the coded light field is reduced, as is the case of exactly zero disparity, it may be that the encoder is incapable of mapping the input to a suitable, *i.e.* geometrically correct, latent space representation. That is, the latent space representation may not be adapted to this extreme kind of sampling, especially since it is also not only used to represent the abstract geometric but also the spatial and spectral information in the case of the multi-task approach. This then would lead to the observed degraded performance for both the central view and the estimated disparity. Nevertheless, the severity of this effect in the case of the disparity estimate is surprising.

In the case of the *Cabin* challenge, showing a more realistic scene geometry with non-constant disparities, the situation is quite different. Here, the effect occurring at zero disparity cannot be observed, or merely slightly in the case of the estimated central view. Rather, the performance of the reconstruction is strongly dependent on the local geometry: The *Cabin* scene features trees in both the fore- and the background, resulting in a complex local geometry with many occlusion boundaries. Hence, the quality of the reconstruction, both of the central view and the disparity, is relatively worse for very small and very large ground truth disparities, while the midground shows better estimates. This result shows that, in the realistic case of non-constant disparities, the reconstruction of zero-disparity regions is effectively regularized by neighboring regions with non-zero disparity, while the overall quality is limited by the scene's complexity.

Overall, these results suggest two conclusions in practice. First, when considering scenes with nearly constant disparity, as may be the case in industrial applications, *e.g.* investigating flat objects such as circuit boards or wavers on lab tables or conveyor belts, the quality degrades severely for objects with zero disparity, *i.e.* for objects that are in focus. However, the solution is straightforward: By focussing the camera's main lens to optical infinity, the zero-disparity plane is moved to optical infinity as well. Therefore, objects are always imaged out-of-focus, avoiding the discussed problem. (One could also move the focal plane to the foreground, however, the resulting disparity sensitivity is worse for objects beyond the focal plane, which is unfavorable in most cases.) However, due to the trivial disparity distribution, the advantage of a (spectral) depth camera over a conventional (spectral) camera is questionable.

Second, in the case of geometrically non-trivial scenes with a diverse disparity distribution, the reconstruction quality is limited by the scene complexity and mostly disparity-independent. Hence, the camera focus should be set such that the depth range of interest is imaged with sufficient sensitivity which depends on the actual camera configuration, as detailed in Appendix A.

5.3 Mask optimization

Up to now, solely random coding masks have been investigated. While the use of random masks is well-motivated in the compressed sensing approach (*cf*. Section 3.1), the influence of the coding mask in the case of the deep learning-based reconstruction is less obvious. This is especially true since the reconstruction targets, in this case, are different from those considered in the compressed sensing-based reconstruction. In particular, the reconstruction of the multispectral central view and its aligned disparity map cannot be formulated in a linear fashion within the compressed sensing framework. Therefore, similar analytic arguments regarding the properties of the coding mask, based on the mutual coherence of the sensing matrix, cannot be made and transferred to the deep learning-based approaches. However, to empirically investigate the influence of the coding mask on the proposed principal reconstruction, two approaches are evaluated in the following.

5.3.1 Predefined coding masks

First, different predefined coding masks, as introduced in Section 4.4.2, are investigated, including fully random masks, masks with random macropixels, as well as regular masks. The results using the test dataset are shown in Table 5.8. To this end, 10 different inference runs, using different initial random seeds, are evaluated and averaged. Investigating the case of using fully random masks during training but different masks at inference, it can be observed that the performance is quite robust with respect to the used mask. In particular, the mask using random macropixels,

Table 5.8 Test dataset performance of the principal reconstruction using different coding masks during training and at inference. The results are obtained as averages from 10 inference runs using different initial random seeds.

Mask		Central view			Disparity		
Training	Inference	PSNR/dB	SSIM	SA/°	MSE/px^2	MAE/px	BP07/%
Random	Random	31.94	0.94	5.36	0.0564	0.0655	13.56
	Rand. macropx	32.44	0.94	5.11	0.0560	0.0652	13.50
	Regular	29.67	0.93	6.37	0.0570	0.0658	13.63
	Regular opt.	27.07	0.91	8.25	0.0579	0.0669	14.12
Rand. macropx	Rand. macropx	33.00	0.94	4.85	0.0573	0.0681	14.03
Regular	Regular	35.88	0.96	4.13	**0.0533**	0.0652	**13.06**
Regular opt.	Regular opt.	**36.20**	**0.96**	**4.10**	0.0563	**0.0649**	13.50

despite its different statistics, performs on-par (or even slightly better) with the original fully random mask used during training. Similarly, using regular coding masks during inference, the performance is only slightly worse in the case of the reconstructed central view while the disparity estimate is mostly mask-independent. This is quite remarkable, considering the severely different statistics of the masks in the regular case. Overall, the approach to use a different mask for each light field in a mini-batch during training in order to obtain a mask-independent central view reconstruction and disparity estimate seems to perform as intended. However, also using the corresponding mask during training yields significantly better performance as compared to training using a random mask, in particular in the case of the considered regular masks. Here, the regular mask with the optimized layout by Shinoda *et al*. [177] yields the overall best performance, reaching a PSNR of over 36 dB of the reconstructed central view and similar quality of the estimated disparity as compared to the other used masks. Furthermore, also the secondary central view metrics, *i.e*. the SSIM and the SA, are significantly improved over the random mask. However, the differences between the naive and the optimized regular mask layout can be considered negligible here.

Similar trends can also be observed using the synthetic dataset challenge and the real-world example, as shown in Figure 5.9. Here, the run closest to the average performance across the 10 different inferences is chosen for visualization. For the dataset challenge, a central view PSNR of over 40 dB and an SSIM of 0.99 is achieved in the case of the regular mask with the optimized layout—by far exceeding all previous methods. Similar to the test dataset performance, the estimated disparity shows only minor differences between the different masks. Investigating the mask dependence using the dataset challenge in more detail, Figure 5.10 shows all individually seeded inference runs for the considered masks. Here, it can be seen that the performance does in fact not depend significantly on the actual realization of the mask, again validating the used training approach. Recall, the different random seeds at inference lead to different realizations of the random masks and different shifts of the regular masks (*cf.* Section 4.4.2). Overall, the regular mask using an optimized layout outperforms the other approaches. In particular, the optimized regular mask performs significantly better than the naive regular mask in the case of the disparity estimation. This is likely due to the larger total variation of the coding mask. That is, the mask has a "spikier" layout and neighboring pixels are more spread out in the spectral domain. This way, different spatio-spectral features may be observed with a higher effective sampling rate, leading to a more robust disparity estimate as compared to the naive regular mask using a consecutive filter layout.

In the case of the real-world data, however, the differences between the naive and the optimized regular masks do not seem to be significant. Yet, the disparities estimated by the models trained with regular masks appear to be sharper than those using random masks. Furthermore, a particularly challenging part of the scene (the front left part of the coconut) that leads to distortions in the estimated disparity for most previously considered methods is estimated correctly by the approaches using regular masks. Analogous to the synthetic case, the quality of the reconstructed central view in the case of regular coding masks outperforms all previous approaches. Concluding, the principal reconstruction using regular coding masks in an optimized layout [177] performs best overall. To visualize its quality in more detail, the ground truth and reconstructed central view are depicted in Figure 5.11. Overall, the proposed principal reconstruction using regular coding masks with an optimized layout is able to reconstruct the spectral central view with high quality and detail. Even for the low-intensity and noisy channels of very short and very long wavelengths, the reconstruction performs remarkably well, to some extent de-noising the noisy ground truth data. In the mid-range channels, the reconstructed central view is visually almost indistinguishable from the ground truth data. Hence, the overall performance, as previously discussed, is likely limited by the quality of the ground truth data, both with respect to the sensor noise as well the inaccuracies of the calibration. Yet, to possibly further enhance the reconstruction, the end-to-end optimization of the coding mask is considered in the following.

It should be noted that, using regular coding masks, the reconstruction of the central view can actually be performed using naive subaperture-wise multispectral demosaicing. However, this naive approach has two drawbacks in practice. First, a subaperture-wise reconstruction does not take into account the light field geometry. Therefore, the subsequent disparity estimation using the reconstructed spectral light field would likely perform poorly. Explicitly taking the geometry into account during demosaicing is not possible in the naive approach, as the disparity, which is unknown, would be needed. Implicitly, it could be achieved in a compressed sensing-based approach, requiring random masks as previously discussed. Furthermore second, in the multispectral case, it has been shown by Degraux *et al.* [48, 49] that compressed sensing approaches outperform conventional multispectral demosaicing using regular masks. In the spectral light field case, it was found in precursory experiments that a 3D subaperture-wise compressed sensing-based reconstruction performs worse than the considered 5D reconstruction [B6]. Therefore, a naive demosaicing approach is not evaluated here, as it is most likely to perform poorly, and in particular worse than the considered compressed sensing-based approaches as well as the principal reconstruction.

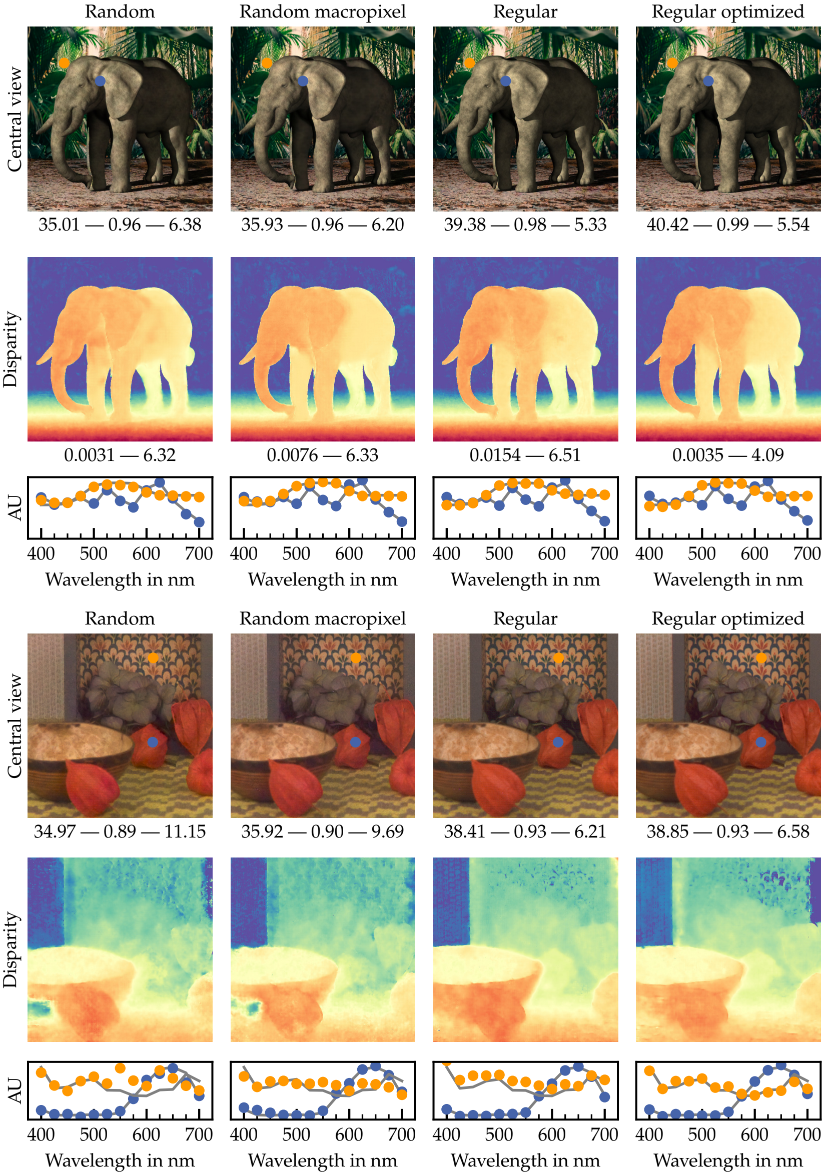


Figure 5.9 Performance comparison of the reconstruction using different coding masks during both training and inference.

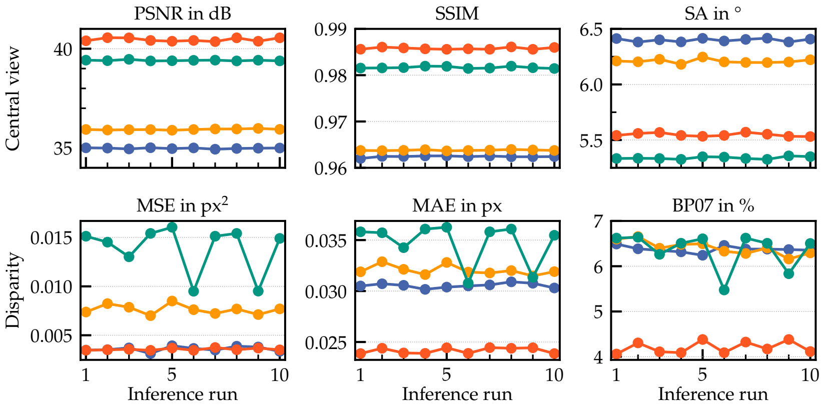


Figure 5.10 Performance for the *Elephant* challenge using different masks during training and at inference. Ten runs, using different random seeds, are shown for the masks: random (●), random macropixel (●), regular (●), and regular optimized (●).

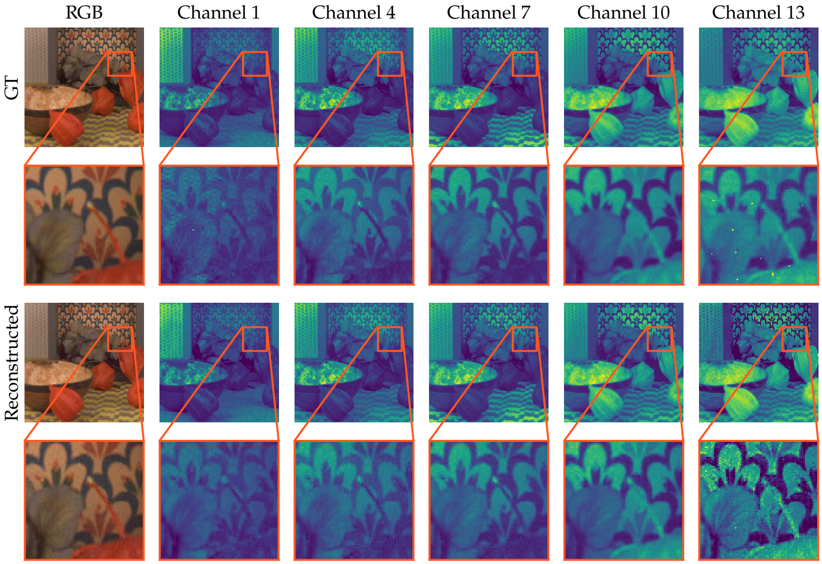


Figure 5.11 False-color representation of five out of the 13 spectral channels from the ground truth and the reconstructed central view using a regular coding mask with an optimized layout [177]. Note that the individual channels are normalized to a common reference to maximize the contrast for visualization.

Table 5.9 Test dataset performance of the investigated end-to-end optimized masks in the case of optimization from pre-trained models as well as optimization from scratch. The previously considered predefined masks are shown as a reference.

Optimization	Mask	Central view			Disparity		
		PSNR/dB	SSIM	SA/°	MSE/px^2	MAE/px	BP07/%
(Predefined)	Random	31.94	0.94	5.36	0.0564	0.0655	13.56
	Regular opt.	**36.20**	**0.96**	**4.10**	0.0563	0.0649	13.50
From pre-trained	Fractal 4×4	33.32	0.94	4.95	0.0566	0.0691	14.58
	Fractal 5×5	32.29	0.93	5.45	0.0563	0.0673	14.03
	Fractal 6×6	32.35	0.93	5.43	0.0558	0.0680	14.15
	Regular 4×4	31.44	0.94	6.23	0.0554	0.0672	14.18
	Regular 5×5	27.42	0.91	10.38	0.0564	0.0681	14.36
	Regular 6×6	29.88	0.93	7.45	0.0558	0.0684	14.52
From scratch	Fractal 4×4	33.24	0.95	4.63	**0.0544**	**0.0643**	**13.32**

5.3.2 End-to-end optimized coding masks

Finally, an end-to-end optimization of the coding masks using neural fractals, as proposed in Section 3.3, is performed. To this end, two training strategies are used. First, a pre-trained model, fully trained using random coding masks, is used to bootstrap the mask optimization. Here, the optimization is performed in three phases: a warmup phase of five epochs optimizing solely the mask while fixing the parameters of the pre-trained reconstruction network; a joint optimization phase of 20 epochs, training both the mask and the reconstruction network; and a final finetuning phase of 10 epochs, training solely the reconstruction network while fixing the mask parameters. Throughout the three phases, the learning rate is decayed from 5×10^{-2} to 5×10^{-5}. The main idea is to make use of the mostly mask-independent reconstruction performance of the pre-trained model, significantly reducing the training time of the mask optimization. However, as a comparison, also a fully joint optimization, trained from scratch in accordance with the previous approaches, is evaluated. As before, the adaptive multi-task training strategy and the proposed adaptive auxiliary losses are used in both training scenarios. Both unconstrained fractal masks as well as masks constrained to a regular layout, as discussed in Section 3.3, are investigated with sizes ranging from 4×4 to 6×6 px. The test dataset results are given in Table 5.9.

Overall, the results are mostly underwhelming. However, some useful insights can still be gained from them. First, investigating the cases trained from the pre-trained model, it can be observed that the proposed fractal optimization does in fact perform better than the constrained optimization of regular masks, suggesting that the fractal masks are more expressive and performant than regular ones, as desired. On the other hand, the optimized masks only perform slightly better than the baseline using random masks (which was used to bootstrap the optimization) in the case of the fractal masks. However, this improvement may also stem from the additional training, in particular the finetuning phase. More severely, the masks constrained to regular layouts actually perform worse than the baseline in the case of the reconstructed central view. All methods perform similarly with regard to the estimated disparity. Similarly, the mask optimization trained from scratch does not perform significantly different from the one bootstrapped using the pre-trained model. All considered end-to-end optimized approaches perform significantly worse than the predefined regular mask in the optimized layout by [177] in terms of the reconstructed central view while showing some minor improvements in the estimated disparity.

In particular, this is disappointing since the predefined regular mask, considered previously, is in fact an element of the search space (or, more precisely, the generated image of the search space) of the mask optimization. Hence, in principle, a properly optimized neural fractal should at least achieve on-par performance. Therefore, the issue is most likely related to the actual optimization. In fact, unlike previous works investigating optimization of binary coding masks [32, 90], the downstream task is much more complex and parameter-intensive in the considered case. Since the gradients have to back-propagate all the way through the reconstruction network to optimize the mask parameters, issues related to vanishing or canceling gradients may also play a role here. Despite a lot of effort spent investigating several annealing strategies for the softmax temperature or the entropy loss as well as experimenting with hard and soft binarization, the problem could not be resolved and the performance of the predefined coding mask could not be achieved let alone improved. It may be advisable to investigate the fractal-based binary mask optimization in a simpler scenario, *e.g.* for RGB demosaicing, as

Chakrabarti [32], or importance sampling, as Huijben *et al.* [90]. However, this is not within the main scope of this thesis. While successful end-to-end optimizations of binary coding masks within complex optical systems have been recently reported [136, 204], the optimization of the binary mask remains a challenge in the considered case, even with the discussed tricks. Moreover, recent works by Metz *et al.* [140] discuss the problem arising in optimizing (partially) chaotic systems. The authors argue that, despite being differential, (partially) chaotic systems lead to poor gradient estimates and flat loss landscapes that are difficult to optimize over. In particular, the proposed recurrent fractal mask generation (3.54) fits the definition of partially chaotic recurrent systems by Metz *et al.* However, while the end-to-end optimization of the coding mask could not be achieved here, precursory experiments directly optimizing the neural fractal with respect to a predefined layout were in fact successful, as shown in the digital supplement. Hence, it is unclear to what extent the problem is practical, possibly solvable with further engineering, or systematic in the sense as discussed by Metz *et al.*

Nevertheless, some interesting observations can be made. Lifting the constraint of always using the hard argmax (*cf.* Section 3.3), *i.e.* using a temperature softmax not only for the backpropagation but also for the forward pass, the resulting training metrics are depicted in Figure 5.12. Here, near-binarization is achieved by minimizing the mask entropy whose weight is increased during training, as shown. That is, early on in the training, the coding masks are spectrally "continuous" while the minimization of the mask entropy (along the spectral dimension) leads to nearly binary, spiky masks. Interestingly, it can be seen that the disparity estimation is independent of the mask entropy during training, *i.e.* whether the masks are spectrally continuous or nearly binary, leading to estimates on-par with previous results. This is remarkable, again suggesting that the proposed network architecture is reasonably robust. However, for the reconstruction of the central view, this is clearly not the case. Here, very high training PSNRs of up to 42 dB, as well as high SSIM and low SA scores, can be achieved for high-entropy masks, while the central view quality drastically degrades when the mask entropy regularization strength is increased, favoring nearly binary masks. Since the spectral dimension of the non-binary coding mask can be interpreted

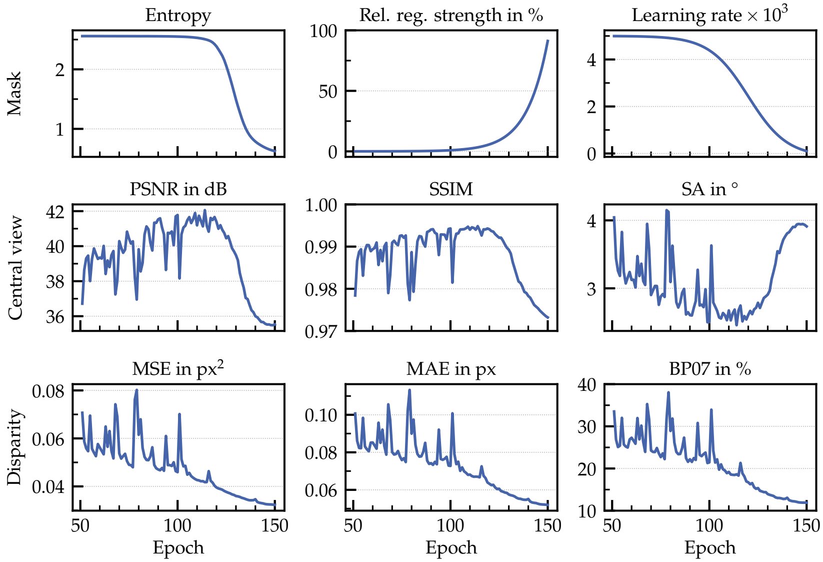


Figure 5.12 Training metrics of the end-to-end mask optimization using a soft forward pass when trained jointly with the principal reconstruction from scratch.

as the transmittance of an optical filter, which is multiplied and spectrally projected with the spectral dimension of the input light field via the scalar product, optimizing non-binary, spectrally continuous masks can be interpreted as optimizing the spectral characteristics of the used optical filters. Therefore, the results suggest that end-to-end optimization of the used spectral filter characteristics and layout may be a promising way to further enhance the quality of the reconstructed spectral central view—possibly even in other than the considered application involving spectral light fields. While the optimization of filter characteristics has been investigated to some extent in the case of color imaging [61], this is not the case in multispectral imaging, to the best of the author's knowledge. Furthermore, continuous end-to-end optimization of the filter characteristics and layout would allow for additional regularization, possibly solving some issues encountered in practice. For example, using the proposed spectral coding of the MLA, one encounters a problem regarding the severely different spectral sensitivities of the different spatial pixels (*i.e.* the microlenses). In particular, similar to the case regarding the custom-built spectral light field camera, pixels associated with green light will saturate much quicker than those in the blue and the red or NIR range. As previously discussed in Section 4.2.2, this effect is further enhanced by the quantum efficiency and the spectral distribution of natural light. Since the proposed camera is a snapshot camera, this will lead to very poor SNRs in the less sensitive channels, as each raw image has to be captured such that the most sensitive channels do not saturate. Using absorbance filters, this issue could be resolved by optimizing the thickness of each filter, such that the overall spectral radiant energy reaching the pixels is roughly constant, regardless of the used filter. This way, it would even be possible to account for the non-constant spectral radiance of the used light source (*e.g.* natural sunlight) and the quantum efficiency of the used camera. However, using interference filters, this approach is not possible and would require additional neutral density filters for each individual microlens. With an end-to-end optimization of the continuous filter characteristics, such constraints could easily be included via additional loss terms. Furthermore, the filters could be optimized for different specific downstream tasks. This way, parts of the downstream tasks, such as classification, may already be solved in the optical domain by the end-to-end optimized filters. To some extent, similar approaches have for example been applied in works regarding classification or abundance estimation [105].

Nevertheless, a practical hardware realization of freely optimized spectral filter transmittance is challenging. While, for example, spectral filters with arbitrary transmittance using acousto-optic tunable filters were achieved by Yushkov and Molchanov [232], in the considered case, the spectral filters would need to be integrated with an MLA at micrometer scale. Acousto-optic tunable filters are therefore not appropriate. It is not clear which technology may be suitable to realize spectral filters with arbitrary transmittance at micrometer scale. Possible recent technologies involve photonic crystal slabs [211], which have been successfully used at sensor-level spectral imaging, or optical microcavities such as whispering gallery mode resonators [36, 172]. Since photonic crystals are based on the same physical principle as acousto-optic tunable filters, *i.e.* a spatially periodic modulation of the refraction index, it may be possible to achieve arbitrary transmittance in a way analogous to the works by Yushkov and Molchanov. Furthermore, other interference-based approaches, such as Fabry-Pérot filters or Mach–Zehnder interferometers, have been successfully integrated at pixel-level [49, 65, 104] and may be suitable to achieve arbitrary transmittance, even though they are typically operated as bandpass filters. Finally, end-to-end optimized diffractive elements may be suitable, in line with recent advances in deep optics [216]. Of course, these approaches require a differentiable forward model of the filter optics, *e.g.* free-form diffractive lenses as used in the approach by Baek *et al.* [13] which was previously discussed (*cf.* Section 3.2). Recently, this principle has been applied successfully at nanometer scale to achieve high-quality nano-optics which can be fabricated as a nano-post array with existing lithographic methods and were optimized using a differentiable forward model [196]. Overall, the results suggest that there is room for improvements utilizing end-to-end optimization techniques, jointly optimizing both the optics and the downstream reconstruction.

6 Conclusion

6.1 Summary

In this thesis, the reconstruction from spatio-spectrally coded light fields, as taken by a camera with a spectrally coded microlens array, was investigated. Mainly, two approaches were discussed: First, a reconstruction of the full spectral light fields was investigated. To this end, different approaches within the compressed sensing framework were developed, either using fixed 5D-DCT bases or learned dictionaries to sparsely represent the spectral light fields. In this case, the conventional vector-based dictionary learning was refined to a tensor-based approach, separating the angular, spatial, and spectral dependence of the light field. From the reconstructed spectral light fields, the disparities were estimated using a state-of-the-art reference method.

Second, a direct reconstruction of the spectral central view and its aligned disparity map from the coded light field was proposed, dubbed principal reconstruction. Here, the desired information is directly estimated from the coded measurements using a dual-task encoder-decoder CNN architecture. In this instance, the camera is interpreted as a monocular spectral depth snapshot camera. To optimize the performance, two multi-task training strategies were investigated. Furthermore, a new approach to use auxiliary losses was developed, called normalized gradient similarity, based on previous works employing gradient similarity, to adaptively weigh auxiliary losses to regularize the training. The auxiliary loss training strategies were also combined with the adaptive multi-task training approaches to further enhance the reconstruction.

Several kinds of coding masks were investigated. In the case of the compressed sensing-based reconstruction, random coding masks were used because they ensure a low mutual coherence of the sensing matrix, which is required to enable the reconstruction from the sparse measurements. For the principal reconstruction, random coding masks were also investigated to be able to directly compare the results with the compressed sensing-based approaches. Furthermore, constrained random masks, as well as regular masks, were considered. Finally, an approach called neural fractals was developed to optimize the coding mask in an end-to-end fashion. Here, the coding masks are formulated via a fractal generation process, which was continuously relaxed to allow for a joint optimization with the reconstruction network.

To evaluate the methods, several datasets were created. A large synthetic spectral light field dataset with disparity ground truth was rendered using a custom camera plugin for the IIIT ray tracer. Consisting of randomly generated scenes, this dataset was split into a training, validation, and test dataset and used to quantitatively evaluate the investigated methods and to train all data-driven approaches. To evaluate the performance also for full-sized spectral light fields, a synthetic dataset of seven hand-crafted scenes was created. For these so-called *challenges*, the ground truth disparity is also available. Finally, a real-world spectral light field dataset was captured using a custom-built spectral light field camera. The radiometric and geometric calibration of the camera, consisting of a Lytro Illum light field camera and a filter wheel with 13 spectral filters, was developed and discussed in detail.

In principle, high reconstruction qualities could be achieved with all investigated approaches. In the compressed sensing case, the proposed tensor-based dictionary outperforms both the conventional vector dictionary as well as the reconstruction using fixed 5D-DCT bases. However, a subsequent disparity estimation, using the state-of-the-art EPINET disparity estimation network, suffered as compared to the performance using the original uncoded light fields. This suggests that the reconstruction suffers from artifacts, in particular with respect to the light field geometry. Furthermore, the compressed sensing-based reconstruction is time- and resource-consuming.

These limitations were overcome with the proposed principal reconstruction. Here, high reconstruction and disparity estimation qualities were achieved. In particular, it was shown that adaptive multi-task training strategies in combination with the proposed auxiliary loss regularization significantly improve the reconstruction as compared to the naive baselines—performing on-par or even outperforming their single-task analogues. The performance was further increased by using regular coding masks in an optimized layout previously proposed in the literature. Unfortunately, the end-to-end optimization of the coding mask via neural fractals did not outperform the regular masks, despite including them in its solution space. To overcome these difficulties, several possible solutions and future investigations were outlined.

Since comparable works and datasets regarding (coded) spectral light fields have previously been missing in the literature, it is the firm belief of the author that this thesis provides a strong reference and a new baseline for their reconstruction and disparity estimation, which may encourage the community to further research.

All datasets and methods proposed within this thesis are made publicly available:

- The created synthetic spectral light field dataset with disparity ground truth is available at https://dx.doi.org/10.21227/y90t-xk47.

- The created real-world dataset is available at https://dx.doi.org/10.35097/500.

- The framework implementing all considered deep learning-based approaches, in particular the investigated training strategies and the proposed NormGradSim method, is available at https://gitlab.com/iiit-public/lfcnn.

- The framework implementing the proposed radiometric and geometric calibration as well as the light field decoding is available at https://gitlab.com/iiit-public/plenpy.
  This framework also includes conventional disparity estimation methods as well as the used spectrum and color conversions.

- The created dataset containing synthetic white images of the Lytro Illum camera with ground truth microlens centers, which was used to evaluate the proposed geometric calibration, is available at https://dx.doi.org/10.21227/msck-x083.

- In addition, a digital supplement to this thesis is available at https://maxschambach.github.io/thesis.

6.2 Limitations and outlook

Despite its good performance, some limitations of the proposed dataset and principal reconstruction exist. Foremost, the created synthetic and real-world datasets are highly textured and contain solely diffuse objects. While this choice was made deliberately to exclude unnecessary challenges in this novel context, it also limits the potential of the proposed reconstruction. In fact, it may be one of the strengths of spectral light fields to provide a robust depth estimation even for untextured or specular objects [235]. Furthermore, the spectral approach opens new possibilities to combine passive light field imaging with active illumination. For example, the additional spectral channels could be used in the near-infrared to use structured illumination, possibly enhancing depth estimation in untextured regions. Also, spectrally coded positional illumination could be used, allowing for surface normal estimation similar to photometric stereo. Finally, all of these techniques could be combined into a single architecture, possibly having a regularizing effect on each other. Being monocular, additional registration and alignment would not be necessary, which is usually the case for multi-modal techniques.

Second, due to the unfocused design of the light field camera, the spatial resolution is small compared to the original sensor resolution. While a direct tradeoff cannot be calculated, as not only a spectral image but also its disparity map are estimated, the overall efficiency may be improved in the future. Given the results regarding the dependence of the reconstruction on the angular resolution, as previously discussed, there likely exists some potential to superresolution. For example, the proposed architecture could be extended by a superresolution component, which could be optimized separately or even jointly. Furthermore, as previously discussed, the overall efficiency and reconstruction quality can likely be improved by developing a more accurate geometric calibration, utilizing the peripheral microlens image measurements, for example using a generalized camera model. A more accurate geometric calibration is also likely going to reduce the observed generalization gap when applying the proposed reconstruction to real-world data, as previously discussed. The overall efficiency could also be improved by using a polar coordinate-based representation of the angular dimension of the light fields, as discussed by Uhlig and Heizmann [199]. However, this would require adaptions of the proposed reconstruction layout, in particular, a replacement of the standard 2D and 3D convolutions would be necessary. (For example, 2D polar coordinate convolutions could be realized via multiplication in the Fourier domain using the 2D polar coordinate Fourier transform [12].) Furthermore, the decoding would be independent of the used microlens array layout, *i.e.* the rectangular or hexagonal arrangement of the microlenses. As previously noted, the hexagonal layout requires a resampling step during decoding, which cannot be performed in the case of coded light fields. Therefore, the de-hexing could be included in the reconstruction network, requiring adaptions to account for the hexagonal sampling. Nevertheless, in the case of a generalized camera model, it is not obvious how to resample the calibrated raxels onto a regular grid, given the sparse nature of the coded light field.

Third, with regard to the proposed reconstruction network, it may be interesting to use the sparsity of the coded input more efficiently. For example, sparse convolutions, as used in the context of point cloud-based deep learning, could be investigated. To some extent, spectrally coded light fields are quite similar to point clouds. In this analogy, the spectral dimension corresponds to the depth of the points. However, unlike point clouds, this spatio-spectral space is regularly sampled, which for point clouds is part of the preprocessing for many approaches (*i.e.* the so-called *voxelization*). Sparse convolutions are then applied to the sparsely sampled measurements in this voxel space. Analogously, sparse convolutions could be applied in the sparsely sampled spatio-angular space. However, it is not straightfoward to include the angular dimension and further research is required. Recent works also explore the direct usage of the disparity map in point cloud-based classification [113]. In this analogy, the disparity map corresponds to the coding mask index map, indicating which measurement belongs to which spectral channel. Therefore, it might be an efficient way of incorporating the mask into the latent representation while feeding the monochromatic projected light fields as input. However, the analogy breaks down when moving from classification to the reconstruction of the spectral central view, for which the analogue in the point cloud case would be more similar to a radiance field, *i.e.* densely sampled in the depth.

Fourth, the proposed training strategy using normalized gradient similarity is computationally and memory-intensive since the gradients for each auxiliary loss have to be calculated. Depending on the size of the network as well as the number of auxiliary losses, it can have a significant impact on the training time. For the considered reconstruction, using the 3D convolutional architecture and four auxiliary losses in total, the training time was increased by about a factor of two compared to the naive training. In precursory experiments, gradient subsampling was investigated, using both static and stochastic subsets of the network weights to calculate the gradients. Despite showing improvements over the baseline, the performance gains of the full gradient-based method were not achieved. To reduce the computational requirements of the proposed training strategy, it is an interesting future investigation to employ dimensionality reduction methods, such as binary random projections, to the approach. However, the required (quasi)orthogonal binary random projections have not yet been investigated in the literature. Furthermore, by design, current graph-based auto differentiation frameworks such as TensorFlow or PyTorch do not allow for gradient calculation with respect to an arbitrary subset of the parameters but only with respect to a full layer, likely limiting the performance gains in practice.

Finally, it may be interesting to employ implicit neural representations for the reconstruction. While implicit representations have been previously discussed within the context of signal representation [180] and novel view synthesis from sparse observations [137, 143], they have not been applied to compressive measurement techniques. In this context, the reconstruction could also be interpreted as a fixed-angle novel view synthesis from sparse samples in the angular, spatial, and spectral domain. In this instance, superresolution would be a “free” byproduct. While it is likely possible to reconstruct the spectral central view via implicit representations, it is unclear how the disparity estimation could be achieved without implicitly representing the full scene geometry. Furthermore, the drawback of these implicit neural representations is that they do not learn any domain knowledge that could be transferred to new data, because the representation is always optimized using a single instance of measurement data.

As a more general note on light field deep learning, it is possible that newer architectures outperform the well-established CNN architectures investigated here. Despite the usefulness of the translational equivariance of CNNs, Transformer models were recently adapted for computer vision tasks. Transformers are based on the core concept of multi-head attention [205] which was originally developed within the natural language processing (NLP) community. Here, it has had a tremendous impact and was quickly adopted by many architectures such as the well-known BERT [50] (and its derivates) or GPT-3 [24], pushing the state of the art of many natural language processing tasks. While Transformers are well suited for sequence models (which is what they were designed for), their application to images (or other higher-dimensional signals) is not straightforward. The Vision Transformer architecture [54] was one of the first to introduce the Transformer concept to image-based tasks. Here, the input image is first patched into a fixed number of patches, which are subsequently flattened and embedded by a linear layer. The representation of the input patches is then simply viewed as a sequence of input tokens analogously to the case in NLP. In the short time since their introduction, Vision Transformers have already shown exciting results, in particular in self-supervised scenarios [31]. However, the Transformer approach has several drawbacks when applied to images. First, the input image resolution is fixed (as is the patch size). Second, as a consequence of the patching and flattening, the positional relation between pixels is, to some extent, lost. While this is alleviated by the positional encoding, which is a standard procedure for Transformer architectures, the full 2D spatial relation is not encoded. Finally, Transformer models scale quite badly with the input size. To be precise, for images of resolution $N \times N$ the time and memory complexity of a Vision Transformer is $\mathcal{O}(N^4)$ [157]. This has recently led to image-specific adaptions of the Transformer model such as the Cross-Covariance Image Transformer [157] which scales linearly with the (flattened) input dimension, $\mathcal{O}(N^2)$.

Therefore, as of early 2022, it is likely that Vision Transformers (or other computer vision-specific Transformer derivatives) are going to gain traction in computer vision deep learning research. But, as is the case with all recent developments in deep learning, it is unclear if they are here to last or simply be superseded by yet another architecture. For example, even fully-connected architectures, operating on image patches, have recently celebrated a comeback to computer vision deep learning [192, 194]. In fact, recent research suggests that the success of Vision Transformers may, at least partially, be attributed to the patching rather than the multi-head attention [8]. In other vision applications, the well-established generative adversarial networks have been outperformed by diffusion models [51]. Therefore, also in the context of light field deep learning, newer approaches such as Transformer architectures are likely to further push the state of the art, despite the fact that the aforementioned challenges are even more severe in this high-dimensional case.

Overall, it is an exciting time to observe and contribute to the rapidly developing field that is computer vision and deep learning.

Appendix

A Light Field Camera Depth and Disparity

In the following, the relationship of disparity and depth in the case of the used unfocused light field camera is derived. In particular, this applies to the virtual light field reference camera of the IIIT ray tracer that is used to render the synthetic light fields presented in this thesis. This reference camera samples the light field in a fashion corresponding to an MLA-based light field camera in the unfocused design with a perfectly aligned MLA and using a thin main lens. In this ideal case, the MLA corresponds to a virtual sensor sampling the spatial dependence of the light field with an effective pixel size given by the microlens diameter. The angular coordinate is sampled by sampling the main lens aperture with a predefined resolution. For an MLA-based light field camera, this angular resolution corresponds to the number of pixels underneath each microlens. The camera parameters, such as the main lens radius R, the microlens radius r, the main lens focal length f, the focus distance g, and the angular resolution N_{ang}, are predefined by the user. Using these parameters, the goal is now to convert depth to disparity and vice-versa.

The geometrical construction of the following derivation is illustrated in Figure A.1. Here, the object G' is imaged out-of-focus onto the MLA plane positioned at the imaging distance b from the main lens. Here, g' denotes the object distance. The true and virtual image distances are denoted by b and b', respectively, and can be obtained via the thin lens equation,

$$\frac{1}{f} = \frac{1}{g} + \frac{1}{b}\,, \quad \text{and} \quad \frac{1}{f} = \frac{1}{g'} + \frac{1}{b'}\,. \tag{A.1}$$

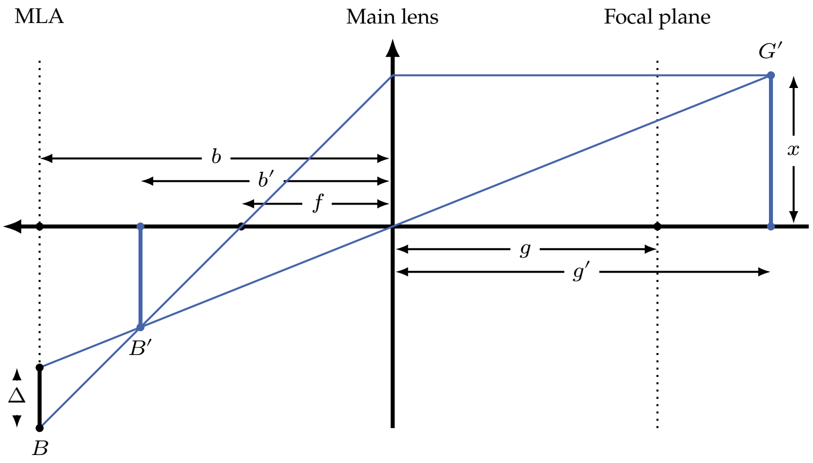


Figure A.1 Schematic drawing for the derivation of the disparity calculated from the object depth in the case of an unfocused plenoptic camera.

The effective baseline of the camera is denoted by x and can be calculated using the main lens radius and the angular resolution via

$$x = \frac{2R}{N_{\text{ang}}}, \tag{A.2}$$

i.e. the effective baseline corresponds to the sampling period of the main lens aperture. Now, the disparity Δ can be derived using the intercept theorem, *i.e.* one finds the relation

$$\frac{-\Delta}{x} = \frac{b - b'}{b'}. \tag{A.3}$$

Note that here the sign is chosen such that objects in front of the focus plane, *i.e.* closer to the camera, have a positive disparity while objects that are beyond the focal plane have a negative disparity. By inserting the effective baseline x, the image distances b and b' as obtained from the thin lens equation, and solving for the disparity, one obtains

$$\Delta = -\frac{2Rf(g - g')}{g'(f - g)N_{\text{ang}}}. \tag{A.4}$$

Here, the disparity value is still given in the used SI units, *e.g.* in meters. To obtain the disparity value in pixels between two subaperture images, one has to divide by the effective spatial pixel size. As previously noted, the effective pixel size is given by the microlens diameter $2r$, as every microlens samples one spatial (s, t) coordinate. Therefore, the disparity d in px is given by

$$d = \frac{\Delta}{2r} = -\frac{Rf(g - g')}{rg'(f - g)N_{\text{ang}}} \,. \tag{A.5}$$

In the case $g' = g$, *i.e.* for an in-focus object, one directly finds $d = 0\,\text{px}$, as expected. Solving (A.5) for the object's depth g', one obtains

$$g' = \frac{Rfg}{Rf - rd(f - g)N_{\text{ang}}} \,. \tag{A.6}$$

The depth-from-disparity and disparity-from-depth in the case of the camera parameters that were used to create the synthetic light field training, validation, and test data are depicted in Figure A.2. As previously noted, the disparity has a very high sensitivity (with respect to depth) for objects that are close to the camera and low sensitivity for objects that are far away. In fact, for any camera configuration there exists a minimum disparity that can be achieved with it: From (A.5) one obtains

$$d \xrightarrow{g' \to \infty} d_{\text{min}} = \frac{Rf}{r(f - g)N_{\text{ang}}} \,. \tag{A.7}$$

E.g., in the case of the used camera parameters, one finds $d_{\text{min}} \approx -2.72\,\text{px}$.

To validate (A.5), in particular in correspondence with the implementation of the reference light field camera of the ray tracer, a test scene is rendered which is designed to precisely evaluate the imaged disparities. The scene consists solely of a white plane spanning one quadrant at a constant distance from the camera. Hence, the central subaperture image shows a white corner spanning a quarter of the image and is otherwise black while the adjacent subapertures will image the corner at an offset corresponding to its disparity. The distance at which the plane is placed is calculated via (A.6) from a target disparity value. The scene is rendered with a light field resolution of (32, 32, 32, 32, 1) at different depths calculated from the disparity values –1 px, –0.5 px, 0 px, 0.5 px, and 1 px.

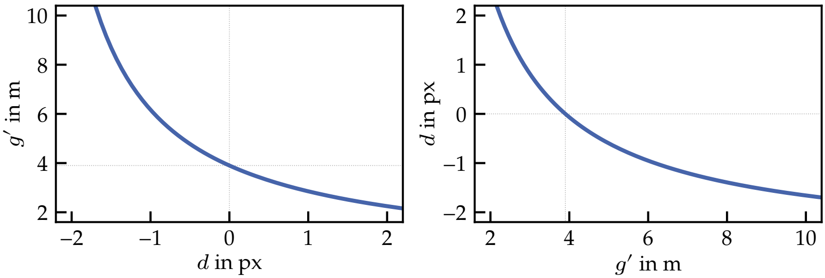


Figure A.2 Disparity and depth relationship from (A.5) and (A.6) in the case of the camera parameters that were used to create the synthetic training, validation, and test dataset.

With this test light field, the image disparity values can be evaluated using the horizontal and vertical EPIs

$$\mathcal{E}_{u_0 s_0}[v, t] \,, \quad \text{and} \quad \mathcal{E}_{v_0 t_0}[u, s] \,. \tag{A.8}$$

This way, both the horizontal and the vertical epipolar geometry is asserted. Here, the angular coordinates $u_0 = 16 = v_0$ and spatial coordinates $s_0 = 8 = t_0$ are fixed. As an example, the test light field for a target disparity of $d = 1$ px is shown in Figure A.3. Note that here, due to the even number of subapertures per angular dimension, there is no exact central view. Hence, for the chosen subaperture, the edge of the plane is imaged onto exactly the middle of the camera pixels. Therefore, during tracing, rays from both the background as well as the plane are sampled and the resulting values are averaged, obtaining a value of 0.5 for the edges and 0.25 for the corner. The horizontal and vertical EPIs for all rendered test scenes are shown in Figure A.4. The results are as anticipated: For a target disparity of $d = 0$ px, the EPIs show a perfectly vertical line since all subapertures are identical. On the other hand, in the case of the target disparities $d = \pm 1$ px, the EPIs show perfect diagonals which is expected since every subaperture is shifted by ±1 px with respect to its adjacent subaperture. Finally, in the case of the target disparities $d = \pm 0.5$ px, the slope of the resulting line in the EPI is exactly two, as expected. Concluding, the derived formula (A.5) and its inverse (A.6) are validated, in particular in combination with the reference light field camera of the used ray tracer.

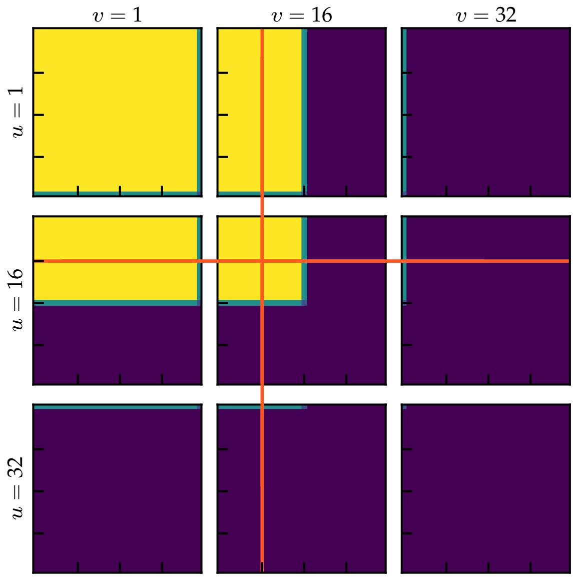


Figure A.3 False-color images of the subapertures $\mathcal{I}_{uv}$ from the central and outermost angular coordinates of the test light field in the case of a target disparity $d = 1\,\mathrm{px}$. The orange lines indicate the sections for the corresponding horizontal and vertical EPIs.

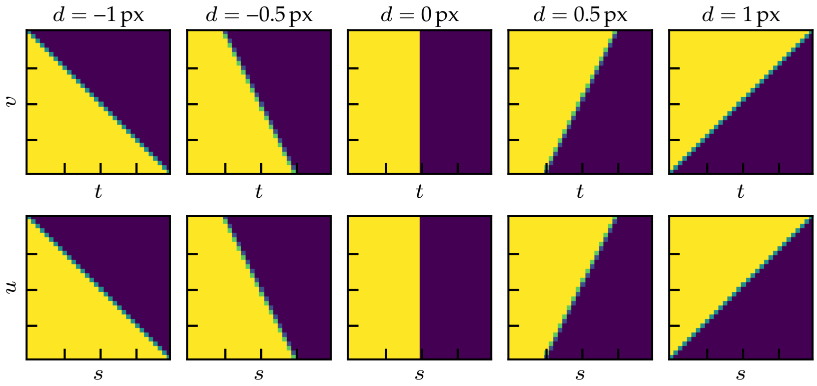


Figure A.4 False-color images of a horizontal (top) and a vertical (bottom) EPI of the test light field rendered at different depths with corresponding target disparities d.

B Geometric Calibration Evaluation

In order to quantitatively evaluate the performance of the MLA grid estimation algorithms (*cf*. Section 4.2.3), appropriate reference data is needed. Of course, real-world white images, as for example provided by the Lytro cameras, are unsuited since the actual microlens centers are unknown. Therefore, reference data has to be synthesized. Previously, Hog *et al*. [84] used a simple addition of three 2D cosine waves to synthesize a white image with known parameters. However, the results are too crude for a precise evaluation of the estimation algorithms. In particular, they neither account for natural nor mechanical vignetting of the main lens and the microlenses. To overcome these shortcomings, the IIIT-RayTracer [A11] is used to synthesize the reference data. To this end, the ray tracer was extended by the camera model of the unfocused plenoptic camera (*cf*. Figure 4.13) to render a multitude of white images with precisely known microlens centers. Mechanical vignetting is implemented using an aperture with variable distance while natural vignetting is implemented by using the $\cos^4$ law and the ray's incident angle. Note that systematic, non-rigid deformations of the MLA, *e.g*. as considered by Pitts *et al*. [162], are not explicitly modeled. It is argued that these irregularities should be eradicated in the manufacturing process of high-quality MLAs as they introduce irreducible blur in the light field.

B.1 Parameter choice

For the evaluation, all parameters are chosen according to a Lytro Illum light field camera. That is, a sensor of size 7728×5368 px with a pixel pitch of 1.4 µm, a resolution of 10 bit, and a gamma factor of 0.4 is used. The microlenses have an approximate diameter $d = 20\,\mu\text{m}$ and a fixed f-number of $f/2$, hence an ideal focal length f of 40 µm. The microlenses are arranged in a hexagonal grid with an estimated grid noise standard deviation of 0.1 % of the microlens diameter, *i.e.* $\sigma_{\mathrm{g}} = 0.0143$ px. Unfortunately, it was not possible to find manufacturer specifications on the grid spacing accuracies so they had to be roughly estimated. The main lens of the Lytro Illum camera is a zoom lens with a focal length equivalent of 30 to 250 mm. The Lytro Illum camera provides a set of 33 different white images by default, taken at 10 different zoom settings and different focus settings. In order to be able to compare the synthetic results to actual white images, four main focal lengths are chosen for which a corresponding white image is provided by the camera. In particular, the focal lengths F of 30 mm, 47 mm, 117 mm, and 249 mm are chosen to fully cover the zoom range of the Lytro Illum camera. For every white image, three different aperture settings were simulated, ranging from no mechanical vignetting to strong vignetting, where the object-side aperture is chosen such that the resulting vignetting effect is visually comparable with the Lytro white image of the corresponding zoom setting and a focus setting showing the strongest vignetting. The remaining parameters, such as the grid rotation α and offset $\mathbf{o}_{\mathrm{g}}$, were varied to obtain a collection of different white images in order to increase the statistical significance of the evaluation. A total of 240 white images were ray-traced. The synthetic white images were then mosaiced using a Bayer pattern with color response according to the Lytro Illum camera. Furthermore, Gaussian noise with a standard deviation σ_{n} of four different levels was added to the white images to investigate the robustness of the grid estimation algorithms with respect to image noise. Hence, a total of 960 different white images are evaluated. A comparison of a synthesized and a Lytro Illum white image was previously shown in Figure 4.15. The synthesized image incorporates all relevant characteristics of the real one, in particular natural vignetting, which causes off-center brightest pixels, and mechanical vignetting resulting in the characteristic cat-eye shape of the projected microlens images close to the sensor edges. The dataset is made publicly available [A10].

B.2 Evaluation metrics

To quantitatively measure the performance of the grid estimation algorithms, the following quality measures are used. The overall grid estimation accuracy Q_g is measured by calculating the root mean square distance of the estimated to true grid points,

$$Q_\text{g} = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\|\hat{\mathbf{c}}_k^\text{p} - \mathbf{c}_k^\text{p}\|^2}\,. \tag{B.1}$$

Here, M denotes the number of grid points in the estimated grid. When grid noise has been added to the ideal grid points, higher values of Q_g are to be expected. Ideally, one would estimate the perfect regular grid that the grid points are derived from, *i.e.*

$$\mathtt{Q}_\text{g, ideal} = \sqrt{\frac{1}{M}\sum_{k=1}^{M}\mathtt{e}_k^2} = \frac{\sigma_g}{\sqrt{M}}\sqrt{\sum_{k=1}^{M}\left(\frac{\mathtt{e}_k}{\sigma_\text{g}}\right)^2} =: \frac{\sigma_g}{\sqrt{M}}\cdot\mathtt{X}\,, \tag{B.2}$$

where $\mathtt{e}_k \sim \mathcal{N}(0,\sigma_\text{g})$ and hence $\mathtt{X}$ is distributed according to the chi distribution with M degrees of freedom. Therefore, one finds the expected value

$$\overline{Q}_\text{g, ideal} := \mathbb{E}\left[\mathtt{Q}_\text{g, ideal}\right] = \sigma_\text{g}\cdot\sqrt{\frac{2}{M}}\cdot\frac{\Gamma((M+1)/2)}{\Gamma(M/2)}\,, \tag{B.3}$$

where Γ denotes the gamma function. Using the identity

$$\lim_{n\to\infty}\frac{\Gamma(n+\gamma)}{\Gamma(n)n^\gamma} = 1\,, \quad \text{for all } \gamma \in \mathbb{C}\,, \tag{B.4}$$

with $n = M/2$ and $\gamma = 1/2$, one obtains the approximation

$$\overline{Q}_\text{g, ideal} \approx \sigma_\text{g} \tag{B.5}$$

for a large number of detected grid points M. This is viewed as the ideal mean grid estimation accuracy. To gain further insight into the grid estimation performance, the mean absolute difference of estimated to true grid rotation α as well as the mean absolute difference of estimated to true grid spacing d,

$$Q_\text{s} = \left|\hat{d} - d\right|\,, \tag{B.6}$$

$$Q_\text{r} = |\hat{\alpha} - \alpha|\,, \tag{B.7}$$

are also calculated. Furthermore, the runtime of each grid estimation algorithm is measured. The evaluation was carried using eight cores of a shared computing node utilizing multithreading where possible.

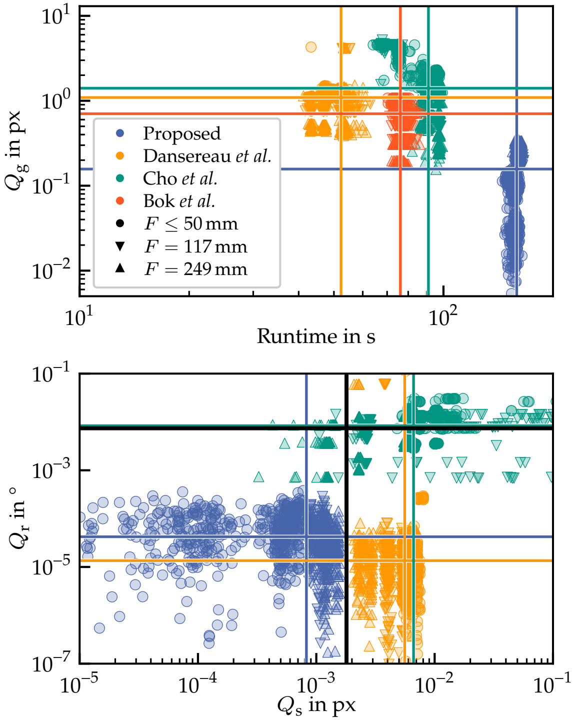


Figure B.1 Performance comparison of the different MLA grid estimation algorithms. The drawn colored lines show the median values of the corresponding datasets. The solid black lines denote the accuracy requirements as stated in Section 4.2.3.3.

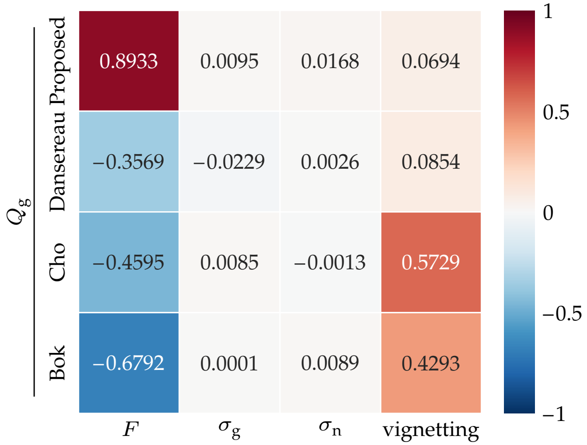
B.3 Results

B.3.1 Grid estimation accuracy

For all 960 white images, a regular grid is estimated with the different estimation algorithms and the overall grid accuracy Q_g as well as the spacing and rotation accuracies Q_s, Q_r are calculated using the ground truth microlens centers and grid parameters. A detailed comparison of the overall grid estimation performances is shown in Figure B.1. Note that, since the calibration by Bok *et al.* [21] does not utilize a regular grid but the individually detected centers, the grid spacing and rotation accuracies $Q_\mathrm{s}, Q_\mathrm{r}$ cannot be specified in this case. It can be observed that only the proposed algorithm satisfies the accuracy requirements as stated in Section 4.2.3.3. While the algorithm by Dansereau *et al.* [45] performs very well in estimating the grid rotation, its spacing estimation is robust but limited in accuracy, in particular for short main lens focal lengths. The proposed method outperforms the others in the overall grid estimation accuracy. The improved accuracy stems from the more accurate grid spacing estimation while the performance regarding the rotation accuracy is slightly worse than the algorithm by Dansereau *et al.* Still, the rotation estimation is performed with high accuracy of about 0.0001°. Conversely, the method proposed by Cho *et al.* [40] does not yield a robust estimation of the grid spacing and rotation. In terms of the grid accuracy Q_g, the proposed method yields results of about one order of magnitude better than the other methods while occasionally performing even two orders of magnitude better. The estimation proposed by Bok *et al.* slightly outperforms the well-established one by Dansereau *et al.*

Even though the proposed method has a longer runtime, with an average of about 160 s per white image compared to an average of about 75 s in the case of the method by Bok *et al.* and about 50 s in the case of the method by Dansereau *et al.*, this is still feasible, as the calibration usually only has to be executed once per camera configuration. Furthermore, the runtime will be shorter in practice using a desktop PC with a clock speed higher than the 2.4 GHz of the used server CPU.

Investigating the results in more detail, as shown in Figure B.2, it can be observed that all methods are insusceptible to image and grid noise. While the method proposed by Cho *et al.* and the method by Bok *et al.* show a strong dependence on the mechanical vignetting present in the white image, the proposed method and the method by Dansereau *et al.* do not show such a correlation. On the other hand, there seems to be a strong dependence on the used focal length for all pre-calibration methods. (To some extent, this is also observable in Figure B.1.) While the methods by Bok *et al.*, Cho *et al.*, and Dansereau *et al.* perform increasingly better with larger image distances, the accuracy of the proposed method decreases. This is further analyzed in Table B.1. Since the scaling factor ζ of the perspective projection (*cf.* Section 4.2.3) converges to one when the image distance increases, the influence of natural vignetting



Q_g	F	σ_g	σ_n	vignetting
Proposed	0.8933	0.0095	0.0168	0.0694
Dansereau	−0.3569	−0.0229	0.0026	0.0854
Cho	−0.4595	0.0085	−0.0013	0.5729
Bok	−0.6792	0.0001	0.0089	0.4293

Figure B.2 Pearson correlation of the grid estimation accuracy Q_g with the different focal lengths F, grid noises σ_g, image noises σ_n and (mechanical) vignetting for the different grid estimation algorithms.

in the white image decreases. That is, with larger image distances, the orthogonally projected centers and the perspectively projected centers coincide. Hence, the methods relying on the local brightness distribution of every microlens, such as the method by Bok *et al*. or Dansereau *et al*., show an increase in accuracy. On the other hand, the proposed method shows extremely accurate estimates, close to the expected ideal mean accuracy $\overline{Q}_{g,\,\text{ideal}}$, in the case of a 30 mm and 47 mm main lens but a decreasing performance in the case of the 117 mm and the 249 mm lens. This is likely due to the characteristics of the mechanical vignetting: While for shorter main lens focal lengths, the mechanical vignetting only influences microlenses very close to the sensor edge but with a sharp cut-off, the vignetting is more spread out in the case of a longer focal length. Therefore, the proposed algorithm is likely to use a smaller window size which decreases the estimation accuracy, since the effective resolution of the Fourier-transformed image is decreased. Nevertheless, the performance in those cases is better than the estimation accuracy reached by Bok *et al*., Dansereau *et al*. or Cho *et al*. Also, using a high-quality main

Table B.1 Mean grid estimation accuracy for the different algorithms for different image distances F (in mm). All other quantities in pixel.

F	$\sigma_{\mathrm{g}} = \overline{Q}_{\mathrm{g,\,ideal}}$	Q_{g}			
		Dansereau [45]	Cho [40]	Bok [21]	Proposed
30	0	1.2850	2.4631	0.9724	**0.0865**
	0.0143	1.2855	2.6117	0.9723	**0.0881**
47	0	1.1323	1.8162	0.7124	**0.0498**
	0.0143	1.1075	1.6608	0.7126	**0.0561**
117	0	0.9418	2.8216	0.5056	**0.1973**
	0.0143	0.9420	2.8486	0.5057	**0.1990**
249	0	0.8238	0.6398	0.4339	**0.2949**
	0.0143	0.7613	0.6369	0.4340	**0.2913**

lens for long focal lengths should mitigate the effects of the mechanical vignetting and lead to higher estimation accuracies.

Overall, the proposed grid estimation algorithm outperforms the ones by Dansereau *et al*. [45], Cho *et al*. [40], and Bok *et al*. [21]. As the method by Cho *et al*. could not provide reliable results, it is excluded from the remaining evaluation.

B.3.2 Calibration accuracy

To quantitatively evaluate the influence of the MLA grid estimation accuracy on the calibration, a Lytro Illum camera, set to a focal length equivalent of 30 mm and focused at infinity, is used. For the full geometric calibration, two well-established methods that have been proposed in the literature are used: Namely, the calibration by Dansereau *et al*. [45], using corner features in the decoded light fields, and the calibration by Bok *et al*. [21], which directly utilizes the raw lenslet images and line features. Since the different methods rely on different features during the calibration, different optimized datasets are used for each. While the calibration by Dansereau *et al*. profits from many corners being present in the calibration images (and hence from smaller grid sizes), the method by Bok *et al*. performs better with larger grid sizes that show more line features in the raw lenslet images.

For the calibration using line features, a dataset containing 10 images of a checker grid with a baseline of 15.57 mm is created. The calibration is performed using the Matlab code provided by Bok *et al.* [21], which was slightly modified to use the individually estimated microlens centers. Overall, the calibration is performed with the original code, as well as with the microlens centers estimated using the method by Dansereau *et al.* as well as the proposed method.

For the calibration using corner features, a dataset containing 10 images of a checker grid is created, now with a baseline of 6.23 mm. The calibration is performed using the Matlab *Light Field Toolbox* by Dansereau. Again, the code was slightly modified such that the different MLA grid estimations could also be used.

The calibration results are shown in Table B.2 and in more detail in Figure B.3 in the case of the calibration using line features by Bok *et al.* The projection and re-projection RMSEs are calculated across all images and light field subapertures. Both calibration methods, the calibration using line features as well as the calibration using corner features, profit from the improved accuracy of the proposed MLA grid estimation algorithm. Depending on the used calibration, the overall ray reprojection RMSE is improved by about 0.02 mm to 0.04 mm which is an improvement by about 15 to 20 % compared to previous methods. The gain in accuracy is larger in peripheral subapertures (as shown in Figure B.3). This likely reflects the more accurate grid spacing estimation leading to a higher quality in the decoded peripheral subapertures.

Additionally, in the case of the calibration using line features, it is observed that the methods utilizing a regular grid to estimate the microlens centers (namely the method by Dansereau *et al.* as well as the proposed method) result in a higher calibration accuracy than the method by Bok *et al.* which uses individual microlens center estimates. To some extent, this is surprising, as the previous results showed a slightly better microlens center estimation performance in the case of the algorithm used by Bok *et al.* [21]. This suggests two conclusions: First, the microlens centers are more robustly estimated when approximated by a regular grid. Estimating a regular grid, a multitude of measurements are fused to obtain an accurate and robust result. Second, systematic, non-rigid deformations of the MLA are likely negligible for the Lytro Illum camera.

Table B.2 Root mean square errors (RMSE) across all subapertures of all calibration images for the different investigated calibration and grid estimation methods.

Method		RMSE	
Calibration	Grid estimation	Re-projection/mm	Projection/px
Bok *et al.* [21]	Bok	0.1409	0.4747
	Dansereau	0.1263	0.4208
	Proposed	**0.1118**	**0.3719**
Dansereau *et al.* [45]	Dansereau	0.2583	-
	Proposed	**0.2196**	-

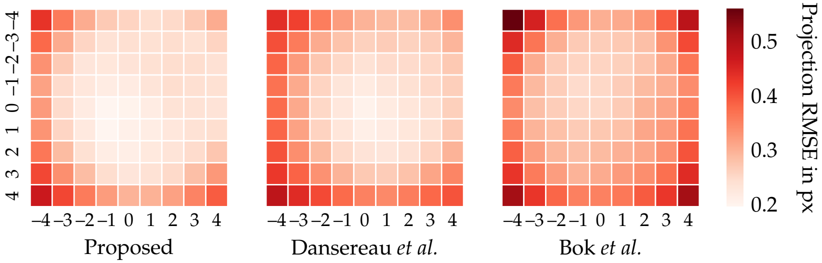


Figure B.3 Projection RMSE across all calibration images for the different subaperture indices (u, v) for the investigated MLA grid estimation algorithms using the calibration by Bok *et al.* [21] in the case of a Lytro Illum camera.

These irregularities should well be detected by the algorithm by Bok *et al.* which however shows a worse calibration. This reinforces the decision to not include these deformations in the camera model.

Overall, the proposed pre-calibration method leads to a high accuracy of the estimated MLA grid parameters, outperforming previously proposed methods. The higher accuracy in turn results in a more accurate geometric calibration of the light field camera. Nevertheless, the peripheral subapertures have a lower accuracy than the central ones, limiting the performance of subsequent light field applications, such as disparity estimation. As previously discussed (*cf.* Section 4.2.4), generic camera models are therefore likely better suited to further enhance the quality of the decoded light fields in the case of MLA-based light field cameras.

C Spectral Light Field Camera Technical Details

As introduced in Section 4.2.1, the custom-built spectral light field camera consists of a housing, enclosing a filter wheel, and a Lytro Illum camera. The camera was designed and developed in cooperation with Stefan Ziegler and built by his team at IIIT Mechanical Workshop. The core of the housing is a 8 mm thick steel plate to which all vital parts are directly connected. In particular, a stepper motor is directly connected to the plate via four bolts. The plate was chosen comparably thick and heavy, as to maximize its inertia and minimize possible abrupt shaking when the motor is stepped. The front cover of the housing consists of a single thin aluminum sheet, which is cut and bent into form. To reduce stray light, the front cover is black anodized. All metal parts are CNC-milled with 0.1 mm precision. This allows for a tight fit of all components and a close placement of the camera and the plastic exit aperture, again minimizing possible stray light. The housing is built to be modular. That is, the housing can be used with arbitrary cameras. To this end, the camera's main lens is fit tightly into a custom plastic enclosure, which is connected with the main housing. To use a different camera, *e.g.* a conventional monochromatic camera, solely a custom lens enclosure and a small plastic plateau has to be build to level the optical axis with that of the filter wheel. A front and back view of the computer-aided design (CAD) model of the camera housing are shown in Figures C.2 and C.3, respectively.

The filter wheel is flange-mounted onto the stepper motor, which is controlled by a Raspberry Pi 4. Unlike servo motors, stepper motors do not offer control over an absolute notion of the angular state of the rotor. In order to calibrate the orientation of the filter wheel, an off-the-shelf forked light barrier is positioned at the bottom of the housing. The light barrier is then used to detect a hole in the filter wheel, which is placed directly opposite to the center of the first filter. To reset the position of the filter wheel, the motor is stepped as long as no event from the light barrier is detected. Because the used stepper motor offers 200 steps per rotation (without microstepping), the filter centers have to be located at integer multiples of 1.8°. Therefore, the 13 filters are each separated by an angle of 27° resulting in a slightly larger angle of 36° between the last and the first filter. To reduce abrupt high torque, microstepping is used, combined with an acceleration and deceleration phase when stepping the motor.

The spectral bandpass filters are off-the-shelf Edmund Optics Techspec hard-coated OD 4 interference filters mounted in a 50 mm diameter black aluminum ring. The central wavelengths are spread out across the visible range, from 400 to 700 nm in 25 nm steps, as discussed previously. The used motor and motor driver are a NEMA 17 2-phase hybrid stepper motor with 70 N cm holding torque and its according driver. To convert the 3.3 V output level of the Raspberry Pi's GPIO pins to the 5 V input needed by the motor driver, a level shifter is used, connected to the 3.3 V and 5 V power supply pins of the Raspberry Pi's 40-pin connector. Furthermore, to make the light barrier event detection more robust, high frequency noise is filtered using a small-capacity capacitor, connected in parallel to the light barrier's phototransistor. All electrical components, besides the light barrier, are soldered onto a small circuit board, which is incorporated into the housing.

For the Raspberry Pi, the ArchLinux ARM Linux distribution is used. The backend and frontend of the spectral light field camera control are written in Python. A full list of the used hardware, the system configuration of the Raspberry Pi, and the Python frontend and backend can be found in the digital supplement. A schematic drawing of the board circuit is shown in Figure C.1.

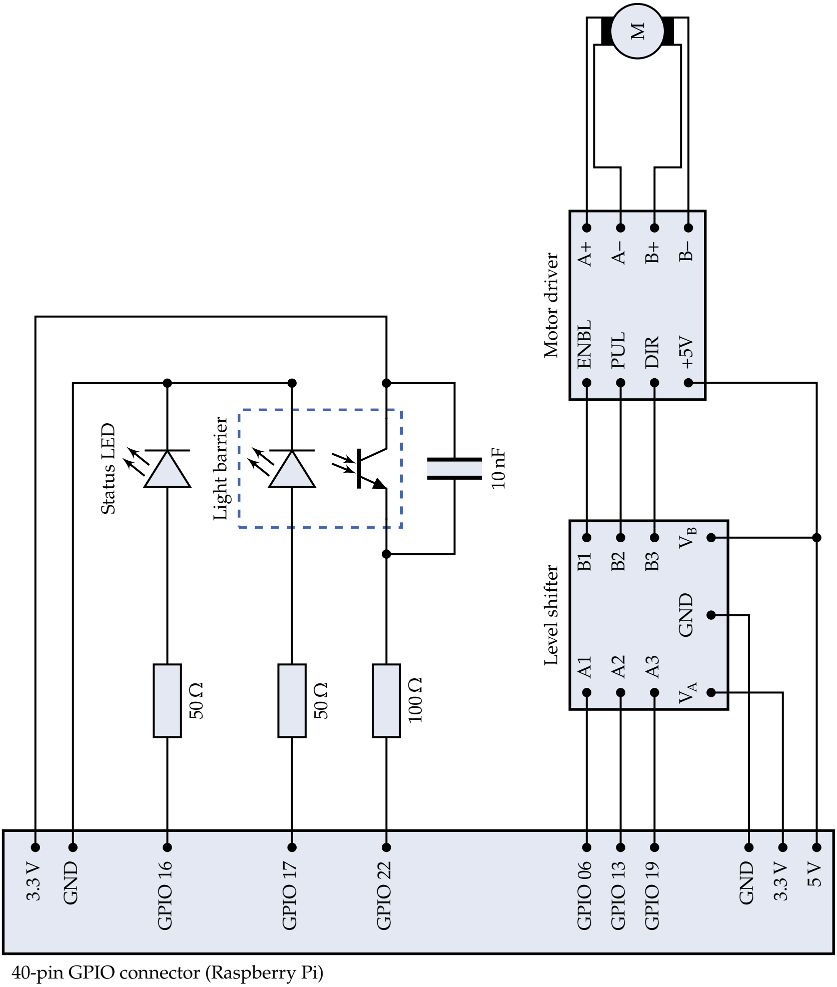


Figure C.1 Schematic drawing of the wiring diagram for the status LED, the light barrier, and the motor control. The 5 V USB power supply of the Raspberry Pi and the 24 V power supply of the motor driver are ommitted for clarity. The shown GPIO pins are the ones used here, however, they are of course freely configurable.

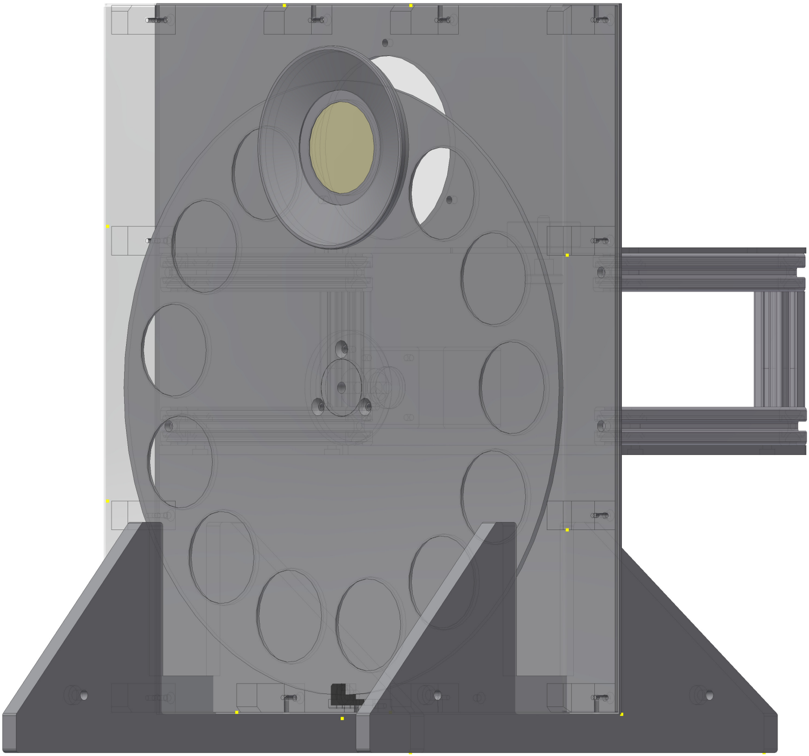


Figure C.2 Front view rendering from the CAD model of the custom-built camera housing. The CAD model was created by Stefan Ziegler at IIIT Mechanical Workshop.

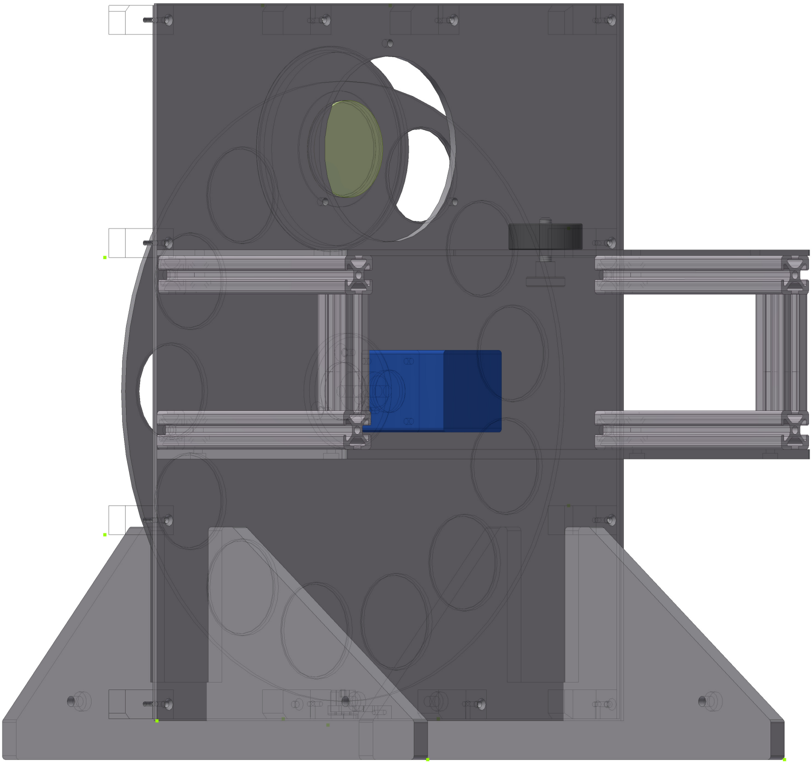


Figure C.3 Back view rendering from the CAD model of the custom-built camera housing. The CAD model was created by Stefan Ziegler at IIIT Mechanical Workshop.

Bibliography

[1] **E. H. Adelson and J. Y. A. Wang**. *Single lens stereo with a plenoptic camera*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 99–106.

[2] **W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson**. *Shearlet transform-based light field compression under low bitrates*. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 4269–4280.

[3] **J. Alamán, R. Alicante, J. I. Peña, and C. Sánchez-Somolinos**. *Inkjet printing of functional materials for optical and photonic applications*. In: *Materials* 9.11 (2016), p. 910.

[4] **A. Alperovich, O. Johannsen, M. Strecke, and B. Goldlücke**. *Light field intrinsics with a deep encoder-decoder network*. In: *Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9145–9154.

[5] **M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes**. *Proposal for a standard default color space for the internet–sRGB*. In: *Color and Imaging Conference*. Vol. 1996. 1. Society for Imaging Science and Technology. 1996, pp. 238–245.

[6] **E. L. Andrade, S. Blunsden, and R. B. Fisher**. *Hidden Markov models for optical flow analysis in crowds*. In: *International Conference on Pattern Recognition*. 2006, pp. 460–463.

[7] **G. Andrew and J. Gao**. *Scalable training of L_1-regularized log-linear models*. In: *International Conference on Machine Learning*. Vol. 24. 2007, pp. 33–40.

[8] **Anonymous**. *Patches are all you need?* In: *OpenReview ICLR 2022 submission* (2021). https://openreview.net/forum?id=TVHS5Y4dNvM.

[9] **B. Arad and O. Ben-Shahar**. *Sparse recovery of hyperspectral signal from natural RGB images*. In: *European Conference on Computer Vision*. 2016, pp. 19–34.

[10] **G. R. Arce, D. J. Brady, L. Carin, H. Arguello, and D. S. Kittle**. *Compressive coded aperture spectral imaging: An introduction*. In: *IEEE Signal Processing Magazine* 31.1 (2013), pp. 105–115.

[11] **A. Ashok and M. A. Neifeld**. *Compressive light field imaging*. In: *Defense, Security, and Sensing*. SPIE. 2010, 76900Q.

[12] **A. Averbuch, R. R. Coifman, D. L. Donoho, M. Elad, and M. Israeli**. *Fast and accurate polar Fourier transform*. In: *Applied and Computational Harmonic Analysis* 21.2 (2006), pp. 145–167.

[13] **S.-H. Baek, H. Ikoma, D. S. Jeon, Y. Li, W. Heidrich, G. Wetzstein, and M. H. Kim**. *Single-shot hyperspectral-depth imaging with learned diffractive optics*. In: *International Conference on Computer Vision*. 2021, pp. 2651–2660.

[14] **M. F. Balın, A. Abid, and J. Zou**. *Concrete autoencoders: Differentiable feature selection and reconstruction*. In: *International Conference on Machine Learning*. Vol. 36. 2019, pp. 444–453.

[15] **Y. Bando, B.-Y. Chen, and T. Nishita**. *Extracting depth and matte using a color-filtered aperture*. In: *ACM SIGGRAPH Asia*. 2008, pp. 1–9.

[16] **B. E. Bayer**. *Color imaging array*. US Patent 3,971,065. 1976.

[17] **A. Beck and M. Teboulle**. *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*. In: *SIAM Journal on Imaging Sciences* 2.1 (2009), pp. 183–202.

[18] **J. Beyerer, F. Puente León, and C. Frese**. *Machine vision. Automated visual inspection: Theory, practice and applications*. Springer, 2015.

[19] **T. E. Bishop, S. Zanetti, and P. Favaro**. *Light field superresolution*. In: *International Conference on Computational Photography*. 2009, pp. 1–9.

[20] **A. Bodkin, A. Sheinis, A. Norton, J. Daly, S. Beaven, and J. Weinheimer**. *Snapshot hyperspectral imaging: The hyperpixel array camera*. In: *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*. Vol. 7334. SPIE. 2009, 73340H.

[21] **Y. Bok, H.-G. Jeon, and I. S. Kweon**. *Geometric calibration of micro-lens-based light field cameras using line features*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.2 (2017), pp. 287–300.

[22] **R. C. Bolles, H. H. Baker, and D. H. Marimont**. *Epipolar-plane image analysis: An approach to determining structure from motion*. In: *International Journal of Computer Vision* 1.1 (1987), pp. 7–55.

[23] **M. Borengasser, W. S. Hungate, and R. Watkins**. *Hyperspectral remote sensing: Principles and applications*. CRC Press, 2007.

[24] **T. Brown *et al.*** *Language models are few-shot learners*. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901.

[25] **Bundesministerium für Wirtschaft und Energie and Statistisches Bundesamt**. *Jährlicher Stromverbrauch eines privaten Haushaltes in Deutschland in den Jahren 1991 bis 2018*. Statista, Statista GmbH. 2019.

[26] **R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu**. *A limited memory algorithm for bound constrained optimization*. In: *SIAM Journal on Scientific Computing* 16.5 (1995), pp. 1190–1208.

[27] **C. F. Caiafa and A. Cichocki**. *Block sparse representations of tensors using Kronecker bases*. In: *International Conference on Acoustics, Speech and Signal Processing*. 2012, pp. 2709–2712.

[28] **E. J. Candès and D. L. Donoho**. *Ridgelets: A key to higher-dimensional intermittency?* In: *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 357.1760 (1999), pp. 2495–2509.

[29] **E. J. Candès and D. L. Donoho**. *Curvelets: A surprisingly effective non-adaptive representation for objects with edges*. In: *International Conference on Curves and Surfaces*. 2000, pp. 105–120.

[30] **E. J. Candès and M. B. Wakin**. *An introduction to compressive sampling*. In: *IEEE Signal Processing Magazine* 25.2 (2008), pp. 21–30.

[31] **M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin**. *Emerging properties in self-supervised Vision Transformers*. In: *International Conference on Computer Vision*. 2021, pp. 9650–9660.

[32] **A. Chakrabarti**. *Learning sensor multiplexing design through back-propagation*. In: *Advances in Neural Information Processing Systems*. Vol. 29. 2016, pp. 3081–3089.

[33] **A. Chakrabarti, W. T. Freeman, and T. Zickler**. *Rethinking color cameras*. In: *International Conference on Computational Photography*. 2014, pp. 1–8.

[34] **C.-I. Chang**. *An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis*. In: *IEEE Transactions on Information Theory* 46.5 (2000), pp. 1927–1932.

[35] **C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod**. *Light field compression using disparity-compensated lifting and shape adaptation*. In: *IEEE Transactions on Image Processing* 15.4 (2006), pp. 793–806.

[36] **Y. K. Chembo, D. V. Strekalov, and N. Yu**. *Spectrum and dynamics of optical frequency combs generated with monolithic whispering gallery mode resonators*. In: *Physical Review Letters* 104.10 (2010), p. 103902.

[37] **J. Chen and L.-P. Chau**. *Light field compressed sensing over a disparity-aware dictionary*. In: *IEEE Transactions on Circuits and Systems for Video Technology* 27.4 (2015), pp. 855–865.

[38] **S. Chen and D. Donoho**. *Basis pursuit*. In: *Asilomar Conference on Signals, Systems and Computers*. Vol. 28. 1994, pp. 41–44.

[39] **Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich**. *GradNorm: Gradient normalization for adaptive loss balancing in deep multitask networks*. In: *International Conference on Machine Learning*. Vol. 35. 2018, pp. 794–803.

[40] **D. Cho, M. Lee, S. Kim, and Y.-W. Tai**. *Modeling the calibration pipeline of the Lytro camera for high quality light-field image reconstruction*. In: *International Conference on Computer Vision*. 2013, pp. 3280–3287.

[41] **Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger**. *3D U-Net: Learning dense volumetric segmentation from sparse annotation*. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2016, pp. 424–432.

[42] **T. S. Cohen, M. Geiger, J. Köhler, and M. Welling**. *Spherical CNNs*. In: *International Conference on Learning Representations*. 2018.

[43] **W. R. Cox, T. Chen, and D. J. Hayes**. *Micro-optics fabrication by ink-jet printers*. In: *Optics and Photonics News* 12.6 (2001), pp. 32–35.

[44] **E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le**. *AutoAugment: Learning augmentation strategies from data*. In: *Conference on Computer Vision and Pattern Recognition*. 2019, pp. 113–123.

[45] **D. G. Dansereau, O. Pizarro, and S. B. Williams**. *Decoding, calibration and rectification for lenselet-based plenoptic cameras*. In: *Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1027–1034.

[46] **D. G. Dansereau, I. Mahon, O. Pizarro, and S. B. Williams**. *Plenoptic flow: Closed-form visual odometry for light field cameras*. In: *International Conference on Intelligent Robots and Systems*. 2011, pp. 4455–4462.

[47] **P. David, M. Le Pendu, and C. Guillemot**. *White lenslet image guided demosaicing for plenoptic cameras*. In: *International Workshop on Multimedia Signal Processing*. 2017.

[48] **K. Degraux, V. Cambareri, L. Jacques, B. Geelen, C. Blanch, and G. Lafruit**. *Generalized inpainting method for hyperspectral image acquisition*. In: *International Conference on Image Processing*. 2015, pp. 315–319.

[49] **K. Degraux, V. Cambareri, B. Geelen, L. Jacques, and G. Lafruit**. *Multispectral compressive imaging strategies using Fabry–Pérot filtered sensors*. In: *IEEE Transactions on Computational Imaging* 4.4 (2018), pp. 661–673.

[50] **J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova**. *BERT: Pre-training of deep bidirectional transformers for language understanding*. In: *Conference of the North American Chapter of the Association for Computational Linguistics*. 2019.

[51] **P. Dhariwal and A. Nichol**. *Diffusion models beat GANs on image synthesis*. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[52] **T. T. Do, L. Gan, N. Nguyen, and T. D. Tran**. *Sparsity adaptive matching pursuit algorithm for practical compressed sensing*. In: *Asilomar Conference on Signals, Systems and Computers*. Vol. 42. 2008, pp. 581–587.

[53] **D. L. Donoho**. *Compressed sensing*. In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.

[54] **A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby**. *An image is worth 16×16 words: Transformers for image recognition at scale*. In: *International Conference on Learning Representations*. 2021.

[55] **M. Drozdzal, E. Vorontsov, G. Chartrand, S. Kadoury, and C. Pal**. *The importance of skip connections in biomedical image segmentation*. In: *Deep Learning and Data Labeling for Medical Applications*. Springer, 2016, pp. 179–187.

[56] **Y. Du, W. M. Czarnecki, S. M. Jayakumar, M. Farajtabar, R. Pascanu, and B. Lakshminarayanan**. *Adapting auxiliary losses using gradient similarity*. In: *arXiv preprint arXiv:1812.02224* (2018).

[57] **G. Easley, D. Labate, and W.-Q. Lim**. *Sparse directional image representations using the discrete shearlet transform*. In: *Applied and Computational Harmonic Analysis* 25.1 (2008), pp. 25–46.

[58] **Y. C. Eldar and G. Kutyniok**. *Compressed sensing: Theory and applications*. Cambridge University Press, 2012.

[59] **V. Farber, Y. Oiknine, I. August, and A. Stern**. *Spectral light fields for improved three-dimensional profilometry*. In: *Optical Engineering* 57.6 (2018), p. 061609.

[60] **M. Feigin, D. Feldman, and N. Sochen**. *From high definition image to low space optimization*. In: *International Conference on Scale Space and Variational Methods in Computer Vision*. 2011, pp. 459–470.

[61] **G. D. Finlayson and Y. Zhu**. *Designing color filters that make cameras more colorimetric*. In: *IEEE Transactions on Image Processing* 30 (2020), pp. 853–867.

[62] **M. A. Finzi, R. Bondesan, and M. Welling**. *Probabilistic numeric convolutional neural networks*. In: *International Conference on Learning Representations*. 2021.

[63] **Y. Fu, J. Gao, Y. Sun, and X. Hong**. *Joint multiple dictionary learning for tensor sparse coding*. In: *International Joint Conference on Neural Networks*. 2014, pp. 2957–2964.

[64] **B. Geelen, N. Tack, and A. Lambrechts**. *A snapshot multispectral imager with integrated, tiled filters and optical duplication*. In: *Advanced Fabrication Technologies for Micro/Nano Optics and Photonics VI*. Vol. 8613. SPIE. 2013, p. 861314.

[65] **B. Geelen, N. Tack, and A. Lambrechts**. *A compact snapshot multispectral imager with a monolithically integrated per-pixel filter mosaic*. In: *Advanced Fabrication Technologies for Micro/Nano Optics and Photonics VII*. Vol. 8974. SPIE. 2014, p. 89740L.

[66] **M. E. Gehm, R. John, D. J. Brady, R. M. Willett, and T. J. Schulz**. *Single-shot compressive spectral imaging with a dual-disperser architecture*. In: *Optics Express* 15.21 (), pp. 14013–14027.

[67] **N. Genser, J. Seiler, and A. Kaup**. *Camera array for multi-spectral imaging*. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 9234–9249.

[68] **I. Gheța, M. Mathias, M. Heizmann, and J. Beyerer**. *Fusion of combined stereo and spectral series for obtaining 3D information*. In: *Multisensor, Multisource Information Fusion: Architectures, Algorithms, and Applications*. Vol. 6974. SPIE. 2008, p. 697406.

[69] **B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu**. *Light field compression using disparity-compensated lifting*. In: *International Conference on Acoustics, Speech, and Signal Processing*. Vol. 4. 2003, pp. 760–763.

[70] **A. S. Golatkar, A. Achille, and S. Soatto**. *Time matters in regularizing deep networks: Weight decay and data augmentation affect early learning dynamics, matter little near convergence*. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019, pp. 10678–10688.

[71] **S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen**. *The lumigraph*. In: *ACM SIGGRAPH*. 1996, pp. 43–54.

[72] **Y. Grandvalet, S. Canu, and S. Boucheron**. *Noise injection: Theoretical prospects*. In: *Neural Computation* 9.5 (1997), pp. 1093–1108.

[73] **M. D. Grossberg and S. K. Nayar**. *A general imaging model and a method for finding its parameters*. In: *International Conference on Computer Vision*. Vol. 2. 2001, pp. 108–115.

[74] **S. Grusche**. *Basic slit spectroscope reveals three-dimensional scenes through diagonal slices of hyperspectral cubes*. In: *Applied Optics* 53.20 (2014), pp. 4594–4603.

[75] **E. J. Gumbel**. *Statistical theory of extreme values and some practical applications*. Vol. 33. National Bureau of Standards Applied Mathematics Series, 1954.

[76] **K. He, X. Zhang, S. Ren, and J. Sun**. *Deep residual learning for image recognition*. In: *Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[77] **K. He, X. Zhang, S. Ren, and J. Sun**. *Identity mappings in deep residual networks*. In: *European Conference on Computer Vision*. 2016, pp. 630–645.

[78] **S. Heber and T. Pock**. *Shape from light field meets robust PCA*. In: *European Conference on Computer Vision*. 2014, pp. 751–767.

[79] **S. Heber and T. Pock**. *Convolutional networks for shape from light field*. In: *Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3746–3754.

[80] **S. Heber, R. Ranftl, and T. Pock**. *Variational shape from light field*. In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. 2013, pp. 66–79.

[81] **F. Heide, W. Heidrich, M. Hullin, and G. Wetzstein**. *Doppler time-of-flight imaging*. In: *ACM Transactions on Graphics* 34.4 (2015), pp. 1–11.

[82] **G. E. Hinton and R. R. Salakhutdinov**. *Reducing the dimensionality of data with neural networks*. In: *Science* 313.5786 (2006), pp. 504–507.

[83] **K. Hoffman and R. Kunze**. *Linear algebra*. Prentice-Hall, 1971.

[84] **M. Hog, N. Sabater, B. Vandame, and V. Drazic**. *An image rendering pipeline for focused plenoptic cameras*. In: *IEEE Transactions on Computational Imaging* 3.4 (2017), pp. 811–821.

[85] **K. Honauer, O. Johannsen, D. Kondermann, and B. Goldlücke**. *A dataset and evaluation methodology for depth estimation on 4D light fields*. In: *Asian Conference on Computer Vision*. 2016, pp. 19–34.

[86] **E. Hoogeboom, J. W. T. Peters, T. S. Cohen, and M. Welling**. *HexaConv*. In: *International Conference on Learning Representations*. 2018.

[87] **R. Horstmeyer, G. Euliss, and R. Athale**. *Flexible multimodal camera using a light field architecture*. In: *International Conference on Computational Photography*. 2009, pp. 1–8.

[88] **J. Hu, M. Ozay, Y. Zhang, and T. Okatani**. *Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries*. In: *Winter Conference on Applications of Computer Vision*. 2019, pp. 1043–1051.

[89] **P. J. Huber**. *Robust estimation of a location parameter*. In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[90] **I. A. M. Huijben, B. S. Veeling, and R. J. G. van Sloun**. *Deep probabilistic subsampling for task-adaptive compressed sensing*. In: *International Conference on Learning Representations*. 2020.

[91] **R. J. Hyndman and A. B. Koehler**. *Another look at measures of forecast accuracy*. In: *International Journal of Forecasting* 22.4 (2006), pp. 679–688.

[92] **I. Ihrke, J. Restrepo, and L. Mignard-Debise**. *Principles of light field imaging: Briefly revisiting 25 years of research*. In: *IEEE Signal Processing Magazine* 33.5 (2016), pp. 59–69.

[93] **S. Ioffe and C. Szegedy**. *Batch normalization: Accelerating deep network training by reducing internal covariate shift*. In: *International Conference on Machine Learning*. Vol. 32. 2015, pp. 448–456.

[94] **J. Janai, F. Güney, A. Behl, and A. Geiger**. *Computer vision for autonomous vehicles: Problems, datasets and state of the art*. In: *Foundations and Trends in Computer Graphics and Vision* 12.1–3 (2020), pp. 1–308.

[95] **K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun**. *What is the best multi-stage architecture for object recognition?* In: *International Conference on Computer Vision*. 2009, pp. 2146–2153.

[96] **J. Jia, C. Ni, A. Sarangan, and K. Hirakawa**. *Fourier multispectral imaging*. In: *Optics Express* 23.17 (2015), pp. 22649–22657.

[97] **O. Johannsen, A. Sulc, and B. Goldlücke**. *What sparse light field coding reveals about scene structure*. In: *Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3262–3270.

[98] **O. Johannsen, K. Honauer, B. Goldlücke, A. Alperovich, F. Battisti, Y. Bok, M. Brizzi, M. Carli, G. Choe, M. Diebold, M. Gutsche, H.-G. Jeon, *et al.*** *A taxonomy and evaluation of dense light field depth estimation algorithms*. In: *Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 82–99.

[99] **J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žı dek, A. Potapenko, *et al.*** *Highly accurate protein structure prediction with AlphaFold*. In: *Nature* (2021), pp. 1–11.

[100] **T. Karras, S. Laine, and T. Aila**. *A style-based generator architecture for generative adversarial networks*. In: *Conference on Computer Vision and Pattern Recognition*. 2019, pp. 4401–4410.

[101] **P. Kellnhofer, L. C. Jebe, A. Jones, R. Spicer, K. Pulli, and G. Wetzstein**. *Neural lumigraph rendering*. In: *Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4287–4297.

[102] **A. Kendall, Y. Gal, and R. Cipolla**. *Multi-task learning using uncertainty to weigh losses for scene geometry and semantics*. In: *Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491.

[103] **D. P. Kingma and J. Ba**. *Adam: A method for stochastic optimization*. In: *International Conference on Learning Representations*. 2015.

[104] **D. M. Kita, B. Miranda, D. Favela, D. Bono, J. Michon, H. Lin, T. Gu, and J. Hu**. *High-performance and scalable on-chip digital Fourier transform spectroscopy*. In: *Nature Communications* 9.1 (2018), pp. 1–7.

[105] **W. Krippner, J. Anastasiadis, and F. Puente León**. *Robust iterative estimation of material abundances based on spectral filters exploiting the SVD*. In: *Algorithms, Technologies, and Applications for Multispectral and Hyperspectral Imagery XXV*. Vol. 10986. SPIE. 2019, 109861T.

[106] **A. Krizhevsky and G. Hinton**. *Learning multiple layers of features from tiny images*. In: *Technical Report* (2009).

[107] **A. Krizhevsky, I. Sutskever, and G. E. Hinton**. *ImageNet classification with deep convolutional neural networks*. In: *Advances in Neural Information Processing Systems*. Vol. 25. 2012, pp. 1097–1105.

[108] **Y. LeCun, C. Cortes, and C. J. C. Burtes**. *The MNIST database of handwritten digits*. http://yann.lecun.com/exdb/mnist/, last accessed 11/27/2021 .

[109] **H. Lee, A. Battle, R. Raina, and A. Y. Ng**. *Efficient sparse coding algorithms*. In: *Advances in Neural Information Processing Systems*. Vol. 19. 2007, pp. 801–808.

[110] **A. Levin, R. Fergus, F. Durand, and W. T. Freeman**. *Image and depth from a conventional camera with a coded aperture*. In: *ACM Transactions on Graphics* 26.3 (2007), p. 70.

[111] **M. Levoy and P. Hanrahan**. *Light field rendering*. In: *ACM SIGGRAPH*. 1996, pp. 31–42.

[112] **M. Levoy and P. Hanrahan**. *Light field rendering*. In: *Conference on Computer Graphics and Interactive Techniques*. ACM. 1996, pp. 31–42.

[113] **L. Li and M. Heizmann**. *2.5D-VoteNet: Depth map based 3D object detection for real-time applications*. In: *British Machine Vision Conference*. 2021.

[114] **N. Li, J. Ye, Y. Ji, H. Ling, and J. Yu**. *Saliency detection on light field*. In: *Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2806–2813.

[115] **Y. Li and F. Liu**. *Adaptive Gaussian noise injection regularization for neural networks*. In: *International Symposium on Neural Networks*. 2020, pp. 176–189.

[116] **C.-K. Liang and Z. Wang**. *Calibration of light-field camera geometry via robust fitting*. US Patent 9,420,276. 2016.

[117] **H. Lin, C. Chen, S. B. Kang, and J. Yu**. *Depth recovery from light field using focal stack symmetry*. In: *International Conference on Computer Vision*. 2015, pp. 3451–3459.

[118] **X. Lin, H.-L. Zhen, Z. Li, Q. Zhang, and S. Kwong**. *Pareto multi-task learning*. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[119] **X. Lin, Y. Liu, J. Wu, and Q. Dai**. *Spatial-spectral encoded compressive hyperspectral imaging*. In: *ACM Transactions on Graphics* 33.6 (2014), p. 233.

[120] **X. Lin, H. S. Baweja, G. Kantor, and D. Held**. *Adaptive auxiliary task weighting for reinforcement learning*. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[121] **A. Lindenmayer**. *Mathematical models for cellular interactions in development I. Filaments with one-sided inputs*. In: *Journal of Theoretical Biology* 18.3 (1968), pp. 280–299.

[122] **S. Liu, A. J. Davison, and E. Johns**. *Self-supervised generalisation with meta auxiliary learning*. In: *Advances in Neural Information Processing Systems*. Vol. 32. 2019.

[123] **Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer**. *Multilingual denoising pre-training for neural machine translation*. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 726–742.

[124] **E. Lobacheva, M. Kodryan, N. Chirkova, A. Malinin, and D. P. Vetrov**. *On the periodic behavior of neural network training with batch normalization and weight decay*. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[125] **G. Lu and B. Fei**. *Medical hyperspectral imaging: A review*. In: *Journal of Biomedical Optics* 19.1 (2014), p. 010901.

[126] **L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis**. *Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators*. In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229.

[127] **A. Lumsdaine and T. Georgiev**. *The focused plenoptic camera*. In: *International Conference on Computational Photography*. 2009, pp. 1–8.

[128] **J. Luo, W. Zhang, J. Su, and F. Xiang**. *Hexagonal convolutional neural networks for hexagonal grids*. In: *IEEE Access* 7 (2019), pp. 142738–142749.

[129] **H. Ma, H. Li, Z. Qian, S. Shi, and T. Mu**. *Fast and accurate 3D measurement based on light-field camera and deep learning*. In: *Sensors* 19.20 (2019), p. 4399.

[130] **C. J. Maddison, A. Mnih, and Y. W. Teh**. *The concrete distribution: A continuous relaxation of discrete random variables*. In: *International Conference on Learning Representations*. 2017.

[131] **J. Mairal, F. Bach, J. Ponce, and G. Sapiro**. *Online dictionary learning for sparse coding*. In: *International Conference on Machine Learning*. Vol. 26. 2009, pp. 689–696.

[132] **S. Mallat**. *A wavelet tour of signal processing*. Elsevier, 1999.

[133] **S. G. Mallat and Z. Zhang**. *Matching pursuits with time-frequency dictionaries*. In: *IEEE Transactions on Signal Processing* 41.12 (1993), pp. 3397–3415.

[134] **M. Marquez, H. Rueda-Chacon, and H. Arguello**. *Compressive spectral light field image reconstruction via online tensor representation*. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 3558–3568.

[135] **M. Marquez, P. Meza, F. Rojas, H. Arguello, and E. Vera**. *Snapshot compressive spectral depth imaging from coded aberrations*. In: *Optics Express* 29.6 (2021), pp. 8142–8159.

[136] **J. N. P. Martel, L. K. Mueller, S. J. Carey, P. Dudek, and G. Wetzstein**. *Neural sensors: Learning pixel exposures for HDR imaging and video compressive sensing with programmable sensors*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.7 (2020), pp. 1642–1653.

[137] **R. Martin-Brualla, N. Radwan, M. S. M. Sajjadi, J. T. Barron, A. Dosovitskiy, and D. Duckworth**. *NeRF in the wild: Neural radiance fields for unconstrained photo collections*. In: *Conference on Computer Vision and Pattern Recognition*. 2021, pp. 7210–7219.

[138] **K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar**. *Compressive light field photography using overcomplete dictionaries and optimized projections*. In: *ACM Transactions on Graphics* 32.4 (2013), pp. 1–12.

[139] **L. Meng and K. Berkner**. *System model and performance evaluation of spectrally coded plenoptic camera*. In: *Imaging Systems and Applications*. OSA. 2012, JW1A–3.

[140] **L. Metz, C. D. Freeman, S. S. Schoenholz, and T. Kachman**. *Gradients are not all you need*. In: *arXiv preprint arXiv:2111.05803* (2021).

[141] **L. Miao and H. Qi**. *The design and evaluation of a generic method for generating mosaicked multispectral filter arrays*. In: *IEEE Transactions on Image Processing* 15.9 (2006), pp. 2780–2791.

[142] **L. Mignard-Debise**. *Tools for the paraxial optical design of light field imaging systems*. PhD thesis. Université de Bordeaux, 2018.

[143] **B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng**. *NeRF: Representing scenes as neural radiance fields for view synthesis*. In: *European Conference on Computer Vision*. 2020, pp. 405–421.

[144] **Y. Ming, X. Meng, C. Fan, and H. Yu**. *Deep learning for monocular depth estimation: A review*. In: *Neurocomputing* 438 (2021), pp. 14–33.

[145] **R. von Mises and H. Pollaczek-Geiringer**. *Praktische Verfahren der Gleichungsauflösung*. In: *ZAMM – Zeitschrift für Angewandte Mathematik und Mechanik* 9.1 (1929), pp. 58–77.

[146] **V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*** *Human-level control through deep reinforcement learning*. In: *Nature* 518.7540 (2015), pp. 529–533.

[147] **Y. Monno, M. Tanaka, and M. Okutomi**. *Multispectral demosaicking using guided filter*. In: *Digital Photography VIII*. Vol. 8299. SPIE. 2012, pp. 1–7.

[148] **Y. Monno, S. Kikuchi, M. Tanaka, and M. Okutomi**. *A practical one-shot multispectral imaging system using a single image sensor*. In: *IEEE Transactions on Image Processing* 24.10 (2015), pp. 3048–3059.

[149] **P. Moon and D. E. Spencer**. *The photic field*. MIT Press, Cambridge, 1981.

[150] **P. Nakkiran, G. Kaplun, Y. Bansal, T. Yang, B. Barak, and I. Sutskever**. *Deep double descent: Where bigger models and more data hurt*. In: *International Conference on Learning Representations*. 2020.

[151] **S. M. Nascimento, K. Amano, and D. H. Foster**. *Spatial distributions of local illumination color in natural scenes*. In: *Vision Research* 120 (2016), pp. 39–44.

[152] **D. Needell and J. A. Tropp**. *CoSaMP: Iterative signal recovery from incomplete and inaccurate samples*. In: *Applied and Computational Harmonic Analysis* 26.3 (2009), pp. 301–321.

[153] **D. Needell and R. Vershynin**. *Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit*. In: *Foundations of Computational Mathematics* 9.3 (2009), pp. 317–334.

[154] **R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan**. *Light field photography with a hand-held plenoptic camera*. In: *Stanford Computer Science Technical Report* 2 (2005), pp. 1–11.

[155] **M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob**. *Mitsuba 2: A retargetable forward and inverse renderer*. In: *ACM Transactions on Graphics* 38.6 (2019), pp. 1–17.

[156] **F. Noé, S. Olsson, J. Köhler, and H. Wu**. *Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning*. In: *Science* 365.6457 (2019).

[157] **A. El-Nouby, H. Touvron, M. Caron, P. Bojanowski, M. Douze, A. Joulin, I. Laptev, N. Neverova, G. Synnaeve, J. Verbeek, and H. Jegou**. *XCiT: Cross-covariance image Transformers*. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[158] **E. Orhan and X. Pitkow**. *Skip connections eliminate singularities*. In: *International Conference on Learning Representations*. 2018.

[159] **Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad**. *Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition*. In: *Asilomar Conference on Signals, Systems and Computers*. Vol. 27. 1993, pp. 40–44.

[160] **J. Peng, Z. Xiong, D. Liu, and X. Chen**. *Unsupervised depth estimation from light field using a convolutional neural network*. In: *International Conference on 3D Vision*. 2018, pp. 295–303.

[161] **M. Pharr, W. Jakob, and G. Humphreys**. *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.

[162] **C. Pitts, T. J. Knight, C.-K. Liang, and Y.-R. Ng**. *Compensating for variation in microlens position during light-field image processing*. US Patent 8,831,377. 2014.

[163] **N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti**. *TID2008 – A database for evaluation of full-reference visual quality assessment metrics*. In: *Advances of Modern Radioelectronics* 10.4 (2009), pp. 30–45.

[164] **A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra**. *Grokking: Generalization beyond overfitting on small algorithmic datasets*. In: *International Conference on Learning Representations, Mathematical Reasoning in General Artificial Intelligence Workshop*. 2021.

[165] **S. J. Reddi, S. Kale, and S. Kumar**. *On the convergence of Adam and beyond*. In: *International Conference on Learning Representations*. 2018.

[166] **J. Redmon, S. Divvala, R. Girshick, and A. Farhadi**. *You only look once: Unified, real-time object detection*. In: *Conference on Computer Vision and Pattern Recognition*. 2016, pp. 779–788.

[167] **V. K. Repala and S. R. Dubey**. *Dual CNN models for unsupervised monocular depth estimation*. In: *International Conference on Pattern Recognition and Machine Intelligence*. 2019, pp. 209–217.

[168] **S. I. Rochwite**. *Rochwite camera*. US Patent D150,517. 1948.

[169] **S. I. Rochwite**. *Stereoscope*. US Patent 2,484,591. 1949.

[170] **F. Roemer, G. Del Galdo, and M. Haardt**. *Tensor-based algorithms for learning multidimensional separable dictionaries*. In: *International Conference on Acoustics, Speech and Signal Processing*. 2014, pp. 3963–3967.

[171] **O. Ronneberger, P. Fischer, and T. Brox**. *U-net: Convolutional networks for biomedical image segmentation*. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. 2015, pp. 234–241.

[172] **A. A. Savchenkov, V. S. Ilchenko, A. B. Matsko, and L. Maleki**. *Tunable filter based on whispering gallery modes*. In: *Electronics Letters* 39.4 (2003), pp. 389–391.

[173] **R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni**. *Green AI*. In: *Communications of the ACM* 63.12 (2020), pp. 54–63.

[174] **O. Sener and V. Koltun**. *Multi-task learning as multi-objective optimization*. In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018, pp. 525–536.

[175] **J. Shi, X. Jiang, and C. Guillemot**. *A framework for learning depth from a flexible subset of dense and sparse light field views*. In: *IEEE Transactions on Image Processing* 28.12 (2019), pp. 5867–5880.

[176] **C. Shin, H.-G. Jeon, Y. Yoon, I. So Kweon, and S. Joo Kim**. *EPINET: A fully-convolutional neural network using epipolar geometry for depth from light field images*. In: *Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4748–4757.

[177] **K. Shinoda, T. Hamasaki, M. Hasegawa, S. Kato, and A. Ortega**. *Quality metric for filter arrangement in a multispectral filter array*. In: *Picture Coding Symposium*. 2013, pp. 149–152.

[178] **R. Shogenji, Y. Kitamura, K. Yamada, S. Miyatake, and J. Tanida**. *Multispectral imaging using compact compound optics*. In: *Optics Express* 12.8 (2004), pp. 1643–1655.

[179] **K. Simonyan and A. Zisserman**. *Very deep convolutional networks for large-scale image recognition*. In: *International Conference on Learning Representations*. 2015.

[180] **V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein**. *Implicit neural representations with periodic activation functions*. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020.

[181] **L. N. Smith**. *Cyclical learning rates for training neural networks*. In: *Winter Conference on Applications of Computer Vision*. 2017, pp. 464–472.

[182] **N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov**. *Dropout: A simple way to prevent neural networks from overfitting*. In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1929–1958.

[183] **T. Standley, A. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese**. *Which tasks should be learned together in multi-task learning?* In: *International Conference on Machine Learning*. Vol. 37. 2020, pp. 9120–9132.

[184] **A. Stockman and L. T. Sharpe**. *The spectral sensitivities of the middle-and long-wavelength-sensitive cones derived from measurements in observers of known genotype*. In: *Vision Research* 40.13 (2000), pp. 1711–1737.

[185] **M. Strecke, A. Alperovich, and B. Goldlücke**. *Accurate depth and normal maps from occlusion-aware focal stack symmetry*. In: *Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2814–2822.

[186] **S. Su, F. Heide, G. Wetzstein, and W. Heidrich**. *Deep end-to-end time-of-flight imaging*. In: *Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6383–6392.

[187] **J. Sulam, B. Ophir, M. Zibulevsky, and M. Elad**. *Trainlets: Dictionary learning in high dimensions*. In: *IEEE Transactions on Signal Processing* 64.12 (2016), pp. 3180–3193.

[188] **A. Sulc, A. Alperovich, N. Marniok, and B. Goldlücke**. *Reflection separation in light fields based on sparse coding and specular flow*. In: *Conference on Vision, Modeling and Visualization*. 2016, pp. 137–144.

[189] **M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi**. *Depth from combining defocus and correspondence using light-field cameras*. In: *International Conference on Computer Vision*. 2013, pp. 673–680.

[190] **P. Tatzer, M. Wolf, and T. Panner**. *Industrial application for inline material sorting using hyperspectral imaging in the NIR range*. In: *Real-Time Imaging* 11.2 (2005), pp. 99–107.

[191] **R. Tibshirani**. *Regression shrinkage and selection via the lasso*. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 58.1 (1996), pp. 267–288.

[192] **I. Tolstikhin, N. Houlsby, A. Kolesnikov, L. Beyer, X. Zhai, T. Unterthiner, J. Yung, A. P. Steiner, D. Keysers, J. Uszkoreit, M. Lucic, and A. Dosovitskiy**. *MLP-mixer: An all-MLP architecture for vision*. In: *Advances in Neural Information Processing Systems*. Vol. 34. 2021.

[193] **I. Tošić and P. Frossard**. *Dictionary learning*. In: *IEEE Signal Processing Magazine* 28.2 (2011), pp. 27–38.

[194] **H. Touvron, P. Bojanowski, M. Caron, M. Cord, A. El-Nouby, E. Grave, G. Izacard, A. Joulin, G. Synnaeve, J. Verbeek, and H. Jégou**. *ResMLP: Feedforward networks for image classification with data-efficient training*. In: *arXiv preprint arXiv:2105.03404* (2021).

[195] **P. Tripicchio, M. Satler, G. Dabisias, E. Ruffaldi, and C. A. Avizzano**. *Towards smart farming and sustainable agriculture with drones*. In: *International Conference on Intelligent Environments*. 2015, pp. 140–143.

[196] **E. Tseng, S. Colburn, J. Whitehead, L. Huang, S.-H. Baek, A. Majumdar, and F. Heide**. *Neural nano-optics for high-quality thin lens imaging*. In: *Nature Communications* 12.6493 (2021).

[197] **D. Uhlig and M. Heizmann**. *Multi-stereo deflectometry with a light-field camera*. In: *tm – Technisches Messen* 85.1 (2018), pp. 59–65.

[198] **D. Uhlig and M. Heizmann**. *A calibration method for the generalized imaging model with uncertain calibration target coordinates*. In: *Asian Conference on Computer Vision*. 2020.

[199] **D. Uhlig and M. Heizmann**. *Model-independent light field reconstruction using a generic camera calibration*. In: *tm – Technisches Messen* 88.6 (2021), pp. 361–373.

[200] **A. K. Vadathya, S. Girish, and K. Mitra**. *A unified learning-based framework for light field reconstruction from coded projections*. In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 304–316.

[201] **S. Vagharshakyan, R. Bregovic, and A. Gotchev**. *Accelerated shearlet-domain light field reconstruction*. In: *IEEE Journal of Selected Topics in Signal Processing* 11.7 (2017), pp. 1082–1091.

[202] **S. Vagharshakyan, R. Bregovic, and A. Gotchev**. *Light field reconstruction using shearlet transform*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.1 (2017), pp. 133–147.

[203] **H. Van Gorp, I. Huijben, B. S. Veeling, N. Pezzotti, and R. J. G. Van Sloun**. *Active deep probabilistic subsampling*. In: *International Conference on Machine Learning*. Vol. 38. 2021, pp. 10509–10518.

[204] **E. Vargas, J. N. P. Martel, G. Wetzstein, and H. Arguello**. *Time-multiplexed coded aperture imaging: Learned coded aperture and pixel exposures for compressive imaging systems*. In: *International Conference on Computer Vision*. 2021, pp. 2692–2702.

[205] **A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin**. *Attention is all you need*. In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017, pp. 5998–6008.

[206] **A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin**. *Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing*. In: *ACM Transactions on Graphics* 26.3 (2007), p. 69.

[207] **A. Wagadarikar, R. John, R. Willett, and D. Brady**. *Single disperser design for coded aperture snapshot spectral imaging*. In: *Applied Optics* 47.10 (2008), B44–B51.

[208] **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: *European Conference on Computer Vision*. 2016, pp. 121–138.

[209] **Z. Wang, E. P. Simoncelli, and A. C. Bovik**. *Multiscale structural similarity for image quality assessment*. In: *Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. 2003, pp. 1398–1402.

[210] **Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli**. *Image quality assessment: From error visibility to structural similarity*. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612.

[211] **Z. Wang, S. Yi, A. Chen, M. Zhou, T. S. Luk, A. James, J. Nogan, W. Ross, G. Joe, A. Shahsafi, K. X. Wang, M. A. Kats, and Z. Yu**. *Single-shot on-chip spectral sensors based on photonic crystal slabs*. In: *Nature Communications* 10.1 (2019), pp. 1–6.

[212] **S. Wanner and B. Goldlücke**. *Globally consistent depth labeling of 4D light fields*. In: *Conference on Computer Vision and Pattern Recognition*. 2012, pp. 41–48.

[213] **S. Wanner and B. Goldlücke**. *Variational light field analysis for disparity estimation and super-resolution*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.3 (2013), pp. 606–619.

[214] **S. Wanner, C. Straehle, and B. Goldlücke**. *Globally consistent multi-label assignment on the ray space of 4D light fields*. In: *Conference on Computer Vision and Pattern Recognition*. 2013, pp. 1011–1018.

[215] **M. Weiler, P. Forré, E. Verlinde, and M. Welling**. *Coordinate independent convolutional networks – Isometry and gauge equivariant convolutions on Riemannian manifolds*. In: *arXiv preprint arXiv:2106.06020* (2021).

[216] **G. Wetzstein, A. Ozcan, S. Gigan, S. Fan, D. Englund, M. Soljačić, C. Denz, D. A. Miller, and D. Psaltis**. *Inference in artificial intelligence with deep optics and photonics*. In: *Nature* 588.7836 (2020), pp. 39–47.

[217] **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*. In: *ACM SIGGRAPH*. 2005, pp. 765–776.

[218] **H. Wing Fung Yeung, J. Hou, J. Chen, Y. Ying Chung, and X. Chen**. *Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues*. In: *European Conference on Computer Vision*. 2018, pp. 137–152.

[219] **G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu**. *Light field image processing: An overview*. In: *IEEE Journal of Selected Topics in Signal Processing* 11.7 (2017), pp. 926–954.

[220] **R. Wu, Y. Li, X. Xie, and Z. Lin**. *Optimized multi-spectral filter arrays for spectral reconstruction*. In: *Sensors* 19.13 (2019), p. 2905.

[221] **Z. Xiong, L. Wang, H. Li, D. Liu, and F. Wu**. *Snapshot hyperspectral light field imaging*. In: *Conference on Computer Vision and Pattern Recognition*. 2017, pp. 3270–3278.

[222] **Y. Xue, K. Zhu, Q. Fu, X. Chen, and J. Yu**. *Catadioptric hyperspectral light field imaging*. In: *International Conference on Computer Vision*. 2017, pp. 985–993.

[223] **J. C. Yang, M. Everett, C. Buehler, and L. McMillan**. *A real-time distributed light field camera*. In: *Rendering Techniques* 2002 (2002), pp. 77–86.

[224] **M. Yao, Z. Xiong, L. Wang, D. Liu, and X. Chen**. *Spectral-depth imaging with deep learning based reconstruction*. In: *Optics Express* 27.26 (2019), pp. 38312–38325.

[225] **F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar**. *Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum*. In: *IEEE Transactions on Image Processing* 19.9 (2010), pp. 2241–2253.

[226] **J. Ye and F. Imai**. *High resolution multi-spectral image reconstruction on light field via sparse representation*. In: *Imaging Systems and Applications*. OSA. 2015, IT3A–4.

[227] **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*. In: *IEEE Transactions on Image Processing* 28.5 (2019), pp. 2319–2330.

[228] **Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. S. Kweon**. *Light-field image super-resolution using convolutional neural network*. In: *IEEE Signal Processing Letters* 24.6 (2017), pp. 848–852.

[229] **T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, and C. Finn**. *Gradient surgery for multi-task learning*. In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 5824–5836.

[230] **Z. Yu, J. Yu, A. Lumsdaine, and T. Georgiev**. *An analysis of color demosaicing in plenoptic cameras*. In: *Conference on Computer Vision and Pattern Recognition*. 2012, pp. 901–908.

[231] **Y. Yuan, X. Chen, and J. Wang**. *Object-contextual representations for semantic segmentation*. In: *European Conference on Computer Vision*. 2020, pp. 173–190.

[232] **K. B. Yushkov and V. Y. Molchanov**. *Acousto-optic filters with arbitrary spectral transmission*. In: *Optics Communications* 355 (2015), pp. 177–180.

[233] **M. Zaheer, S. J. Reddi, D. Sachan, S. Kale, and S. Kumar**. *Adaptive methods for nonconvex optimization*. In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.

[234] **H. Zhao, O. Gallo, I. Frosio, and J. Kautz**. *Loss functions for image restoration with neural networks*. In: *IEEE Transactions on Computational Imaging* 3.1 (2016), pp. 47–57.

[235] **M. Zhou, Y. Ding, Y. Ji, S. S. Young, J. Yu, and J. Ye**. *Shape and reflectance reconstruction using concentric multi-spectral light field*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.7 (2020), pp. 1594–1605.

[236] **K. Zhu, Y. Xue, Q. Fu, S. B. Kang, X. Chen, and J. Yu**. *Hyperspectral light field stereo matching*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.5 (2019), pp. 1131–1143.

List of publications

[A1] **Maximilian Schambach**, Jiayang Shi, and Michael Heizmann. *Spectral reconstruction and disparity from spatio-spectrally coded light fields via multi-task deep learning*. In: *International Conference on 3D Vision (oral)*. 2021.

[A2] **Maximilian Schambach** and Michael Heizmann. *A highly textured multispectral light field dataset*. Dataset accompanying [A1]. RADAR4KIT, Karlsruhe Institute of Technology, DOI: 10.35097/500. 2021.

[A3] Theresa Panther, **Maximilian Schambach**, and Michael Heizmann. *Improving light efficiency in multispectral imaging via complementary notch filters*. In: *Automated Visual Inspection and Machine Vision IV*. Vol. 11787. SPIE. 2021, 117870K.

[A4] **Maximilian Schambach**, Qiaoshuang Zhang, Uli Lemmer, and Michael Heizmann. *Automated quality assessment of inkjet-printed microlens arrays*. In: *tm – Technisches Messen* 88.6 (2021), pp. 342–351.

[A5] Matthias Bächle, **Maximilian Schambach**, and Fernando Puente León. *Signal-adapted analytic wavelet packets in arbitrary dimensions*. In: *European Signal Processing Conference (oral)*. Vol. 28. 2021, pp. 2230–2234.

[A6] **Maximilian Schambach** and Michael Heizmann. *A multispectral light field dataset and framework for light field deep learning*. In: *IEEE Access* 8 (2020), pp. 193492–193502.

[A7] **Maximilian Schambach** and Michael Heizmann. *A multispectral light field dataset for light field deep learning*. Dataset accompanying [A6]. IEEE Dataport, DOI: 10.21227/y90t-xk47. 2020.

[A8] **Maximilian Schambach**, Qiaoshuang Zhang, Uli Lemmer, and Michael Heizmann. *Automated quantitative quality assessment of printed microlens arrays*. In: *Forum Bildverarbeitung*. KIT Scientific Publishing, 2020, pp. 51–62.

[A9] **Maximilian Schambach** and Fernando Puente León. *Microlens array grid estimation, light field decoding, and calibration*. In: *IEEE Transactions on Computational Imaging* 6 (2020), pp. 591–603.

[A10] **Maximilian Schambach** and Fernando Puente León. *Synthetic whiteimages for the Lytro Illum light field camera with ground truth microlens center coordinates*. Dataset accompanying [A9]. IEEE Dataport, DOI: 10.21227/msck-x083. 2019.

[A11] Thomas Nürnberg, **Maximilian Schambach**, David Uhlig, Fernando Puente León, and Michael Heizmann. *A simulation framework for the design and evaluation of computational cameras*. In: *Automated Visual Inspection and Machine Vision III (oral)*. Vol. 11061. SPIE. 2019, p. 1106102.

[A12] **Maximilian Schambach** and Fernando Puente León. *Reconstruction of multispectral images from spectrally coded light fields of flat scenes*. In: *tm – Technisches Messen* 86.12 (2019), pp. 758–764.

[A13] **Maximilian Schambach** and Fernando Puente León. *Algorithms for microlens center detection*. In: *Forum Bildverarbeitung (oral)*. KIT Scientific Publishing, 2018, pp. 229–240.

[A14] **Maximilian Schambach** and Ko Sanders. *The Proca field in curved spacetimes and its zero mass limit*. In: *Reports on Mathematical Physics* 82.2 (2018), pp. 203–239.

List of supervised theses

[B1] **J. Shi**. *Reconstruction and disparity estimation from coded light fields via multi-task deep learning*. Master thesis. IIIT, KIT, 2020.

[B2] **F. Nonnenmacher**. *Disparitätsschätzung und Rekonstruktion aus spektral codierten Lichtfeldern mit künstlichen neuronalen Netzen*. Master thesis. IIIT, KIT, 2020.

[B3] **H. Sheng**. *Depth estimation and color reconstruction from color-coded RGB light fields using artificial neural networks*. Master thesis. IIIT, KIT, 2019.

[B4] **A. Zimmer**. *Tiefenberechnung aus Lichtfeldern mit künstlichen neuronalen Netzen*. Master thesis. IIIT, KIT, 2019.

[B5] **J. Peng**. *Compressed sensing methods for multispectral imaging*. Master thesis. IIIT and MRT, KIT, 2019.

[B6] **W. Yi**. *Reconstruction from spectrally coded light fields using DCT bases*. Master thesis. IIIT, KIT, 2019.

[B7] **G. Yang**. *Algorithms for microlens center detection*. Master thesis. IIIT, KIT, 2018.

[B8] **H. Tu**. *Rekonstruktion von Hyperspektralbildern aus spektral kodierten Lichtfeldern*. Master thesis. IIIT and MRT, KIT, 2018.

[B9] **F. Fayala**. *Decodierung von Lichtfeldern für Mikrolinsenarray-basierte Lichtfeldkameras*. Bachelor thesis. IIIT, KIT, 2018.