ULB ECOLE POLYTECHNIQUE DE BRUXELLES

Depth Estimation from Structured Light Fields

Thesis Presented by Yan LI

in fulfilment of the requirements of the PhD Degree in Engineering Sciences and Technology ("Docteur en Science de l'Ingénieur et Technologie")
Academic year 2019-2020

Supervisor: Prof. Gauthier LAFRUIT

Thesis jury

Olivier DEBEIR (Université Libre de Bruxelles, Chair)
Dragomir MILOJEVIC (Université Libre de Bruxelles, Secretary)
Gauthier LAFRUIT (Université Libre de Bruxelles)
Adrian MUNTEANU (Vrije Universiteit Brussel)
Lu ZHANG (INSA-Rennes)

July 9, 2020

DOCTORAL THESIS

By YAN LI

A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy

in the

Laboratory of Image Synthesis and Analysis

in the

UNIVERSITÉ LIBRE DE BRUXELLES

July 9, 2020

ABSTRACT

Light fields have been populated as a new geometry representation of 3D scenes, which is composed of multiple views, offering large potentials to improve the depth perception in the scenes. The light fields can be captured by different camera sensors, in which different acquisitions give rise to different representations, mainly containing a line of camera views - 3D light field representation, a grid of camera views - 4D light field representation. When the captured position is uniformly distributed, the outputs are the structured light fields.

This thesis focuses on depth estimation from the structured light fields. The light field representations (or setups) differ not only in terms of 3D and 4D, but also the density or baseline of camera views. Rather than the objective of reconstructing high quality depths from dense (narrow-baseline) light fields, we put efforts into a general objective, i.e. reconstructing depths from a wide range of light field setups. Hence a series of depth estimation methods from light fields, including traditional and deep learning-based methods, are presented in this thesis. Extra efforts are made for achieving the high performance on aspects of depth accuracy and computation efficiency.

Specifically, 1) a robust traditional framework is put forward for estimating the depth in sparse (wide-baseline) light fields, where a combination of the cost calculation, the window-based filtering and the optimization are conducted; 2) the above-mentioned framework is extended with the extra new or alternative components to the 4D light fields. This new framework shows the ability of being independent of the number of views and/or baseline of 4D light fields when predicting the depth; 3) two new deep learning-based methods are proposed for the light fields with the narrow-baseline, where the features are learned from the Epipolar-Plane-Image and light field images. One of the methods is designed as a lightweight model for more practical goals; 4) due to the dataset deficiency, a large-scale and diverse synthetic wide-baseline dataset with labeled data are created. A new lightweight deep model is proposed for the 4D light fields with the wide-baseline. Besides, this model also works on the 4D light fields with the narrow baseline if trained on the narrow-baseline datasets.

Evaluations are made on the public light field datasets. Experimental results show the proposed depth estimation methods from a wide range of light field setups are capable of achieving the high quality depths, and some even outperform state-of-the-art methods.

ACKNOWLEDGEMENT

Firstly, I would like to express my gratitude to my supervisor Prof. Gauthier Lafruit, who has given me an opportunity to start and complete my PhD in Laboratory of Image Synthesis and Analysis (LISA) of ULB. With his guidence, I go from the start of learning the light field technique to the end of fulfilling the doctoral thesis writing. During this PhD life, I was given much patience, many practical and helpful comments from him for the scientific publications and thesis. I was also offered the free environment for promising research directions, and the chances to go abroad for meaningful scientific meetings.

Secondly, I would like to express my thanks to all jury members for helping me to better finalize the thesis, the colleagues in our LISA for helping me to solve the problems, and the seminars held by LISA for widening the research horizon of mine. I also would like to give my sincere thanks to the LISA lab and VUB cluster Hydra for allowing me to get access to the compute nodes. Further, I want to thank my colleagues, friends in Belgium for sharing the practical news with me and offering me assistances in the daily life.

Next, I will thank China Counsel Scholarship (CSC) and Van Buuren-Jaumotte-Demoulin funding so much for giving me the financial support for my doctoral life and my doctoral thesis respectively.

Last but not least, I will thank my wife Dr. Qiong Wang so much for giving encouragement and unconditional support to me, having scientific discussions and cooperations with me, and giving birth to our newborn baby YiKe after my private defense. I also would like to give many thanks to my parents for their understanding and support in the past five years.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Depth Estimation

1.1.1 Background

Humans are aware of how far away the real-world objects in 3D scene, i.e. the *approximate* depth, from themselves because of the parallax from the separated left and right eyes. Researchers, in the 3D vision community, have been attempting to perceive/estimate the *accurate* depth by measuring the physical distance of real-world points from the sensors. The perceived depth from the sensor (aka range camera) is usually stored as an 8-bit channel image or more (e.g., 16 bits), which has been applied into a variety of research fields, such as segmentation, view synthesis, 3D modeling, autonomous driving, etc.

Over the last few decades, depth estimation has been progressed with fruitful approaches. According to whether the light source (illumination) is emitted or not, the depth perception approaches can be categorized into the active and passive depth sensing [1–4]. With respect to active methods, there are a variety of sensors, including Structured Light sensor (measuring depth from deformed light pattern projected by an infrared laser, e.g, Microsoft Kinect v1 [1]), Time of Flight sensor (measuring depths by calculating the round trip of light beams in the entire scene, e.g., Microsoft Kinect v2 [1]), and LiDAR sensor (measuring depths from reflected light beams by a rotating laser, e.g., Faro scanner [2]). Passive methods are mostly related to the triangulation, in which the depth is generated by finding the correspondence from the (RGB) images. The passive depth sensors mainly consist of stereo camera (e.g., ZED camera [3]) and multi-camera, light field-camera. Researchers have been attracted for more effectively modeling the depth estimation and enhancing the depth estimation performance.

1.1.2 Passive Depth Sensing

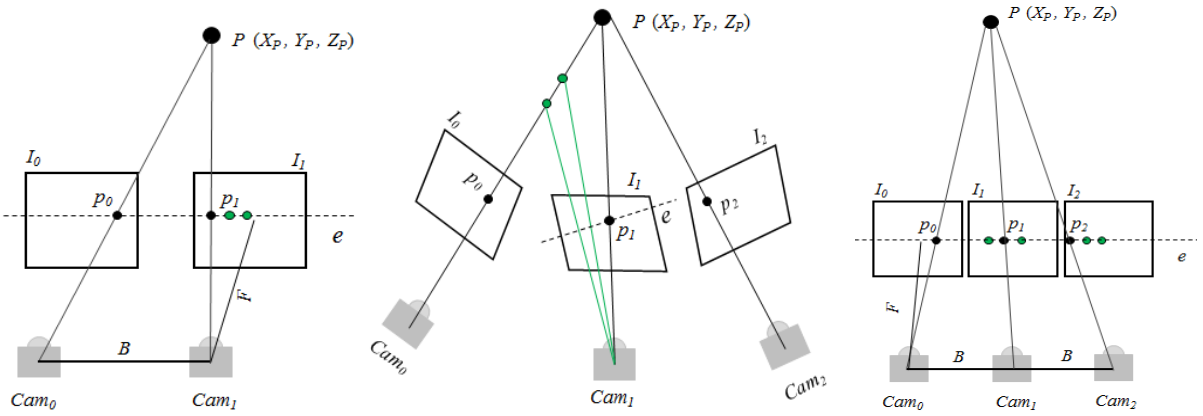
In this section, we will briefly describe the passive methods for depth estimation since it is mostly related to the thesis. We gradually introduce the basic knowledge of depth

1. https://www.xbox.com/en-us/kinect/ 2. https://www.faro.com/ 3. https://www.stereolabs.com/

estimation from two-view stereo, multi-view stereo to light field images, which differ in terms of the ***input*** representation. In fact, the light field inputs could be thought of as the extension of the two-view stereo and multi-view stereo inputs. Among them, there exist the common knowledge that the depth estimation is formulated into the disparity estimation, where the disparity is searched from the corresponding points relying on the epipolar geometry [5], cf. Fig. 1.1. With respect to the disparity property, when the object is near to the cameras, the disparity will be large, and vice versa. Note that, for the two-view stereo and light fields, the disparity is exchangeable to the depth from now on since the depth can be calculated/triangulated by Eq. 1.1,

$$Z = B \times F/D \tag{1.1}$$

where F is the focal length, B is the camera baseline, D is the disparity and Z is the depth.



(a) two-view stereo (b) multi-view stereo (c) light fields

Figure 1.1: Epipolar geometry in different scenarios. I, P and p represent the image plane, the 3D world point and the 2D point projected in the image plane respectively. e indicates an epipolar line, black points p are corresponding points and green points denote the points in the search space.

From two-view stereo: in an earlier stage, researchers imitated the human depth-perception mechanism by placing two cameras in 3D scene, and deduced the disparity from the corresponding points in a rectified/structured image pair (aka. stereo matching), as is shown in Fig. 1.1 (a). The rectified image pair is obtained from two cameras, where the epipolar line is rectified to be parallel and horizontal. The corresponding points search is limited to 1D space, i.e. the horizontal epipolar line. In Fig. 1.1 (a), the left view I_0 is referred to as the reference image I_R and the right view I_1 as the target image I_T. The principle of predicting the disparity for the reference view is to find correspondences from the target view. The traditional pipeline of searching correspondences typically consists of the cost calculation, cost aggregation, regularization or optimization, post-processing (we refer the interested readers to [6] for a detailed review).

For the deep learning-based methods, the feature extraction, cost volume generation, cost aggregation, followed by upsampling via bilinear interpolation are employed in the pipeline, which borrow the knowledge from the traditional methods.

From multi-view stereo: two-view stereo was extended to multi-view stereo on the unstructured camera setups to address the challenging concerns (e.g., the occlusion). In fact, multi-view stereo (MVS) has been a commonly used term in 3D reconstruction [7], in which the multiple cameras are placed at the arbitrary locations. The related works for this camera setup are not only used to recover the depth, but also are used for reconstructing the mesh and the point cloud, going beyond the scope of the thesis. As with depth estimation from the multi-view stereo, at least two unstructured views, i.e., all image views are not rectified, are employed. The parallax lies in the non-horizontal epipolar line instead, and the correspondence search is carried on along this line. In Fig. 1.1 (b), the being estimated view I_0 is referred to as the reference image I_R and several neighboring views (I_1 and I_2) as the target images I_T. MVS methods typically take as input all images and their corresponding camera parameters, and then reconstruct the 3D representation of the scene from all input views. In the traditional depth estimation pipeline, the plane-sweep technique is often firstly used to project the neighboring images onto a number of virtual depth planes, and then calculate the matching cost among multiple views, followed by aggregating or refining the costs. We refer the interested readers to [7] for the detailed developments of the previous traditional methods. For the deep learning-based MVS methods, the pipeline is quite similar to that of the stereo matching [8]. Unlike stereo matching, the input images involved with arbitrary camera locations might pose a tricky issue in using the deep learning technique [9].

From Light fields: the textureless, occlusion regions in two-view stereo and multi-view stereo are often the troublesome issues of being estimated to be the *real* disparity, causing degradations in depth accuracy. 4D light fields, i.e. the compact representation from the plenoptic function [10], record a large amount of information of the scene. The (4D) light fields are typically captured as multiple images/videos from the multi-view setup, which was initially aimed at improving image-based-rendering without the explicit geometry. Light fields came to the computer vision community very early, and were used for reconstructing the depth, which went from a niche research topic to an active topic. Depth estimation from light fields is closely related to that from multi-view stereo (MVS) in the computer vision community, but actually there exists a difference, i.e. the light fields might be densely and regularly sampled [10], which is not the case in multi-view stereo. Since light fields consist of the more (structured) views, this enhances the more potentials of addressing the textureless and occlusion issues than that in the two-view stereo and multi-view stereo. The light fields, in the literature, are mainly structured, being comprised of rectified images. In Fig. 1.1 (c), the parallel cameras are

placed in the scene. The leftmost view I_0 is referred to the reference image I_R and the other views are the target views I_T. One strategy of searching correspondences for the reference view is to integrate the intermediate findings from all reference and target image pairs using the two-view stereo or multi-view stereo methods. Moreover, given that the structured light fields exhibit a several of properties: Epipolar-Plane-Image (EPI, as is shown in Fig. (1.2)), refocusing and symmetry, these are usually are taken into consideration to improve the correspondence search. For instance, the property of EPI is that the slope of its EPI-line is inversely proportional to the disparity, as given in Eq. (1.2).

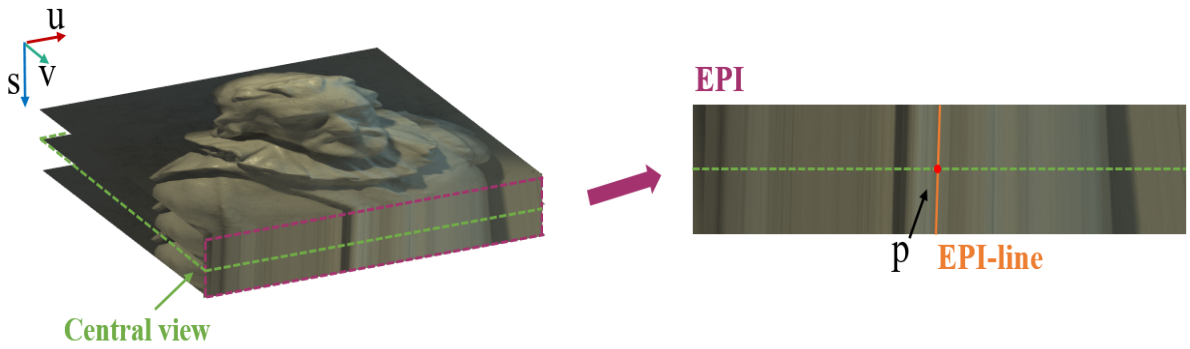


Figure 1.2: Epipolar plane image (EPI). The EPI is constructed by stacking a sequence of epipolar lines in the same image scanline. The line in orange is the EPI-line where the pixel p of the central view (yellow) lies. The slope of this EPI-line is inversely proportional to the *real* disparity.

$$\frac{\Delta s}{\Delta u} = \frac{1}{d} \tag{1.2}$$

Here d represents the disparity of the pixel, Δs represents camera intervals and Δu is the horizontal disparity.

1.2 Light Fields

A light ray in the real-world space can be parameterized by the 3D spatial position for every 2D direction, corresponding to the (5D) plenoptic function in [11]. Due to the storage and computational burden, the plenoptic function is reduced to 4D function under the free space assumptions (free of occluders), referred to as 4D light fields [10] or Lumigraph [12]. 4D light fields are parameterized by two planes, i.e., the camera/angular and image/spatial planes. With respect to the recording of 4D light fields, light rays coming from different directions are split to different pixels on the sensors, which is more distinguished than the integration of rays from different directions as done in the conventional camera. When only a line (e.g. a horizontal or vertical line) is kept

on the camera plane, the 3D light fields can be constructed. In practice, the 3D or 4D light fields are captured by a different number of sampled camera views with different baselines.

1.2.1 4D Light Field Representation

The 4D light field is represented by two-plane parametrization (2PP) in which a camera plane is parametrized by the coordinate system (s, t) and the image plane (u, v). Then it could be simply seen as a collection of a plane of views (cf. Fig. 1.3) with radiance values r in the RGB color space, described as $R = L(u, v, s, t)$, in which (s, t) represents a camera coordinate and (u, v) indicates a coordinate of a pixel on the image plane. The light field view, which is being estimated, is denoted by R_{s^*,t^*}. Then, according to this view, a radiance set $R_{u,v,s,t}(d)$ is easily built by assigning a hypothetical disparity d to a light ray, as given in Eq. 1.3:

$$R_{u,v,s,t}(d) = \{L(u + d * (s^* - s), v + d * (t^* - t), s, t) \\ |s = 1, 2, ..., M; t = 1, 2, ..., N\} \tag{1.3}$$

where d is the disparity in some range and (M, N) denotes the angular resolution of the light field. The subscript (u, v) that corresponds to the pixel or light ray in a view is replaced with p in the following texts for simplicity.

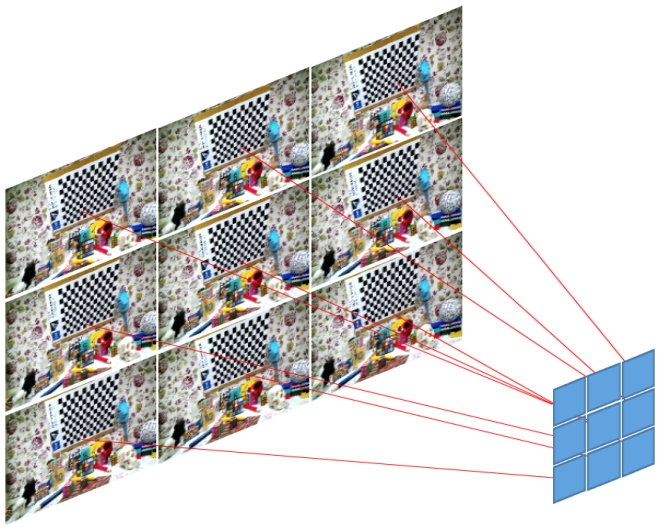


Figure 1.3: Light field images are captured from a equally spaced 2D camera array.

1.2.2 3D Light Field Representation

The 3D light fields are typically a collection of a horizontal line of views with radiance values r in the RGB color space, described as $R = L(u, v, s)$, in which s represents

a camera coordinate and (u,v) indicates a coordinate of a pixel on the image plane. The radiance value set $R_{u,v,s}(d)$ for the reference view can be built by Eq. 1.4, which is similar to that in 4D light fields.

$$R_{u,v,s}(d) = \{L(u + d * (s^* - s), v, s) | s = 1, 2, ..., M\} \tag{1.4}$$

Note that the subscript (u, v) is also removed in the following texts for simplicity.

1.2.3 Baseline

There exists various light field acquisition setups to acquire the light fields, which mainly differs in terms of the *baseline* (or *density*) [13]. The ***baseline*** indicates the inter-camera distance, which is closely related to the disparity range of the scene: the wider baseline corresponds to a higher disparity range and vice versa. Specifically, the narrow-baseline light fields have the low disparity range, e.g., less than 1.5 pixels, whereas the disparity range in the wide-baseline light fields is always much larger than 1.5 pixels. Note that, in the light field community, the term *baseline* is usually exchanged for the *density*, where the sparse sampled light fields indicate or accompany the wide-baseline, while the dense sampled light fields denote or are coupled with the narrow-baseline.

1.3 Motivation

To date, the research community has achieved appealing performances in depth accuracy but are limited to good settings (e.g., densely sampled light fields). Indeed, the densely sampled (narrow-baseline) light fields are capable of enhancing the potentials of high quality, however, the over-sampling or redundancy might occur. Therefore, the trade-off between the redundancy and the quality is necessary. As a fact, the dense sampling or narrow-baseline for the scene capture, was paid more attention to in the past, but now the same scene viewed as the sparse (wide-baseline) light fields with less redundancy seems much more attractive. Therefore, it seems worthy taking this setup into consideration for the new algorithms.

From the perspective of the algorithm, the existed algorithms can be classified into the traditional algorithms and the CNN-based algorithms. Both algorithms have their own pros and cons. In terms of the image features, traditional methods manually engineer the features (aka hand-crafted features), such as the edges and histograms, while CNN-based methods automatically learn features from the data. In terms of the complexity, traditional methods are only related to an algorithmic complexity, while for CNN-based algorithms, the complexity is not only related to the algorithmic complexity,

but also the creation/collection of the training datasets with the high quality, the long-time training, and storing parameters in space. In terms of the computation efficiency, the CNN-based methods is capable of being accelerated by GPUs, while this is not always the case in the traditional methods. In general, for the CNN-based methods, the price to pay is higher, nevertheless, it still be acceptable if the quality-complexity-efficiency trade-off is getting better.

In this thesis, we intend to conduct the more meaningful explorations under a range of settings, and attempt to present new algorithms (including the traditional and CNN-based algorithms) taking care of the potential issues occurred in previous methods on aspects of the *depth accuracy*, *computation efficiency* and *dataset sufficiency*, as shown in Fig. 1.4.

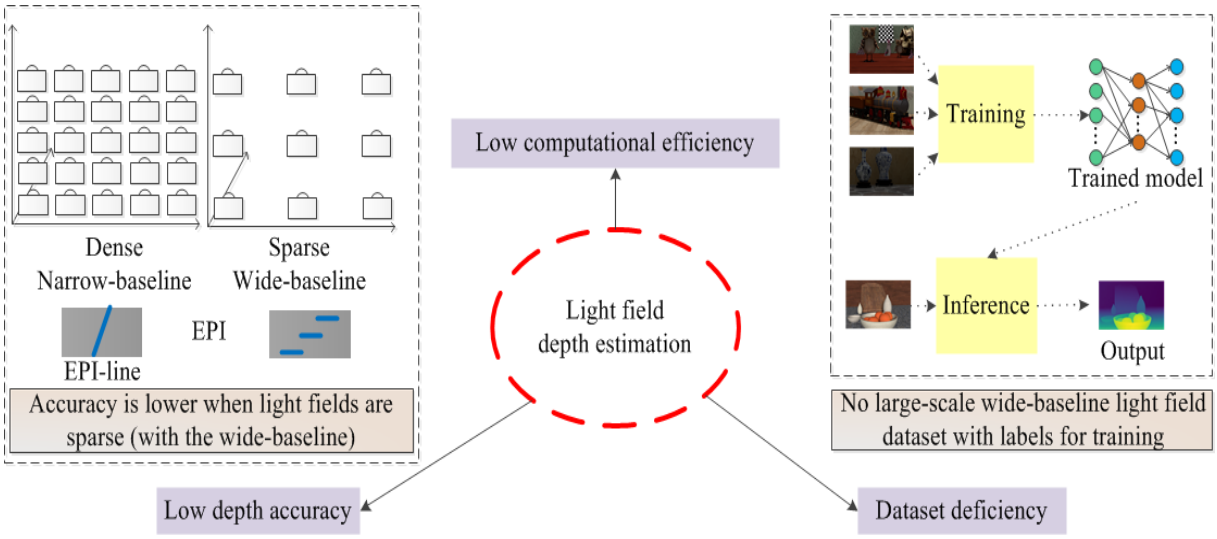


Figure 1.4: Illustration of the potential issues in light field depth estimation.

Depth accuracy: some state-of-the-art methods observe the gradual degradations in extracting depths when the light fields are sparser and/or the baseline is wider. So we might ask at which camera density and/or baseline the quality of depth maps is still acceptable? Or is it possible for a framework to obtain high quality depths that is independent of the density or baseline of the light fields? Note that the sparse light fields reduce not only the budget of light field setup but also the elapsed computation time, therefore it is worth of making explorations toward sparse light fields.

Computational Efficiency: since the number of angular images in light fields is usually an order of magnitude more than two views, most of state-of-the-arts methods relying on the full-shape or star-shape light fields spend a large amount of time in estimating depths for one camera view. If the spatial resolution of light fields goes larger, the computational time might be a nightmare for users, impeding the future potential applications. Therefore it is essential to give attention to the high computational

efficiency algorithms.

Dataset Sufficiency: as is well known, the deep learning has witnessed a fruitful progress in a variety of vision tasks, including depth estimation from two views. The deep learning requires a great deal of perfect labelled data in general, however, the light field research community has the limited public datasets with labels for the supervised depth estimation learning tasks. Though there exist a large number of real-world light fields configured with the wide-baseline, there are no available large-scale datasets with labels for measurement or supervised training. Thus it is of significance to involve such dataset with labelled data.

1.4 Contribution

The thesis has made several contributions for depth estimation from structured light fields. These contributions come from different perspectives, being classified into the traditional perspective and the CNN-based perspective. Actually, we firstly focus on the traditional algorithms that are distributed to **Part I**, and then move on to the CNN-based algorithms distributed to **Part II**. In general, the proposed CNN-based algorithms outperform the proposed traditional algorithms in the depth accuracy and computational efficiency (using GPU accelerations is a precondition of getting higher efficiency in CNN-based algorithms, otherwise it is not true). Whereas, in contrast with the proposed traditional algorithms, the proposed CNN-based algorithms have to store the extra models with a large number of parameters in space. The detailed contributions are summarized below, and some related visulizations are shown in Fig. 1.5, Fig. 1.6, Fig. 1.7, and Fig. 1.8.

Traditional algorithms 1) A robust depth estimation framework for 3D sparsely-sampled (wide-baseline) light fields (1x10) (*R3DE*) is presented, achieving high quality depth in real-world datasets. 2) A scalable framework based on the 4D light fields (*S-R4DE*) is presented, which allows to accurately predict depths from the dense (9x9) or sparse (3x3) light fields with different baselines.

CNN-based algorithms 3) A couple of the CNNs (*HFNet* and *MANet*) are proposed, which improve the depth accuracy on the 4D light fields with the narrow-baseline. While the *LLF-Net* is proposed to perform well on both the narrow- and wide-baseline 4D light fields. 4) The three proposed CNNs for light field depth estimation require a much lower computational overhead than the traditional methods, especially the runtime of the *MANet* and *LLF-Net* is both less than 1 second. 5) Two lightweight CNNs (*MANet* and *LLF-Net*) with less than 2 million parameters are presented. The *MANet* has around 1.6M parameters that achieves the state-of-the-art accuracy on the narrow-baseline light field datasets, while the *LLF-Net* with a bit more parameters, i.e. 1.8M,

achieving state-of-the-art accuracy on both the narrow- and wide-baseline datasets. 6) Considering that the light field community lacks of the synthetic dataset with wide baseline, the new *W*ide-baseline *L*ight *F*ield dataset *WLF* is introduced for the first time (to the best of our knowledge) to fill in this gap.

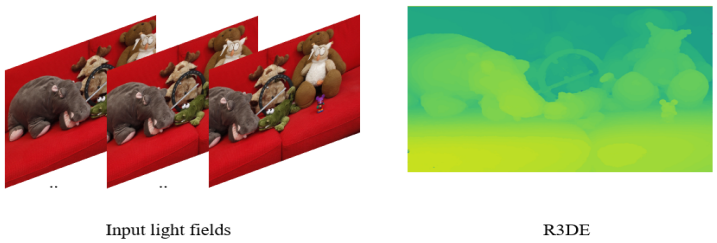


Figure 1.5: Example of the depth map from 3D light fields by the *R3DE* in Chapter 3.

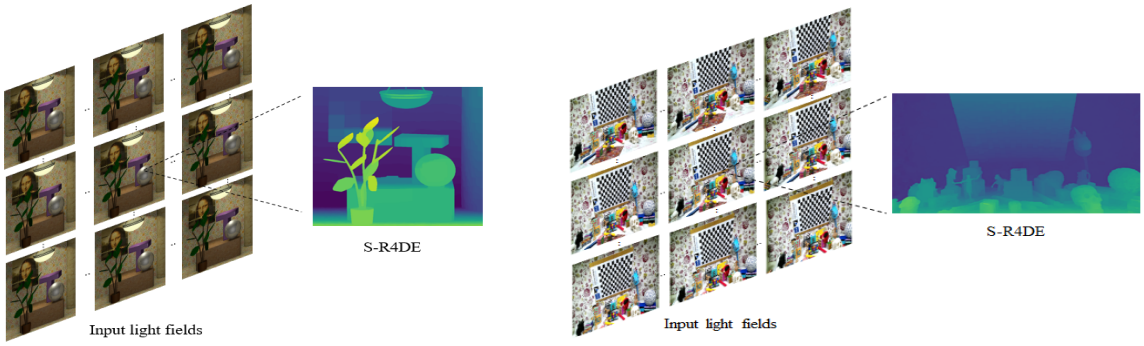


Figure 1.6: Example of the depth maps by the *S-R4DE* in Chapter 4: the scene from the left to right is from the narrow- and wide-baseline 4D light fields respectively.

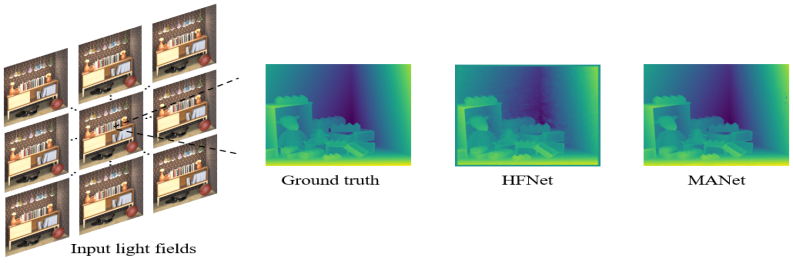


Figure 1.7: Example of the depth maps from the narrow-baseline 4D light fields by *HFNet* and *MANet* in Chapter 5

1.5 Outline

The thesis introduces several methods to recover the depth from the structured light fields, and includes seven chapters in total, and is organized as in Fig. 1.9 (excluding the first chapter).

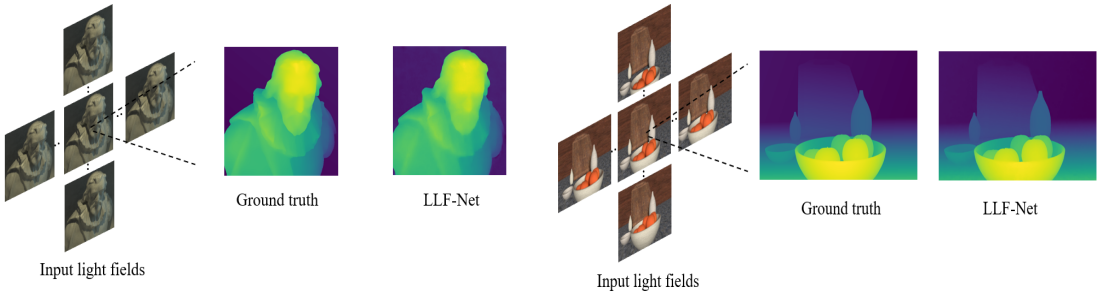


Figure 1.8: Example of the depth maps by the *LLF-Net* in Chapter 6: the scene from the left to right is from the narrow-baseline 4D light fields and *WLF* respectively.

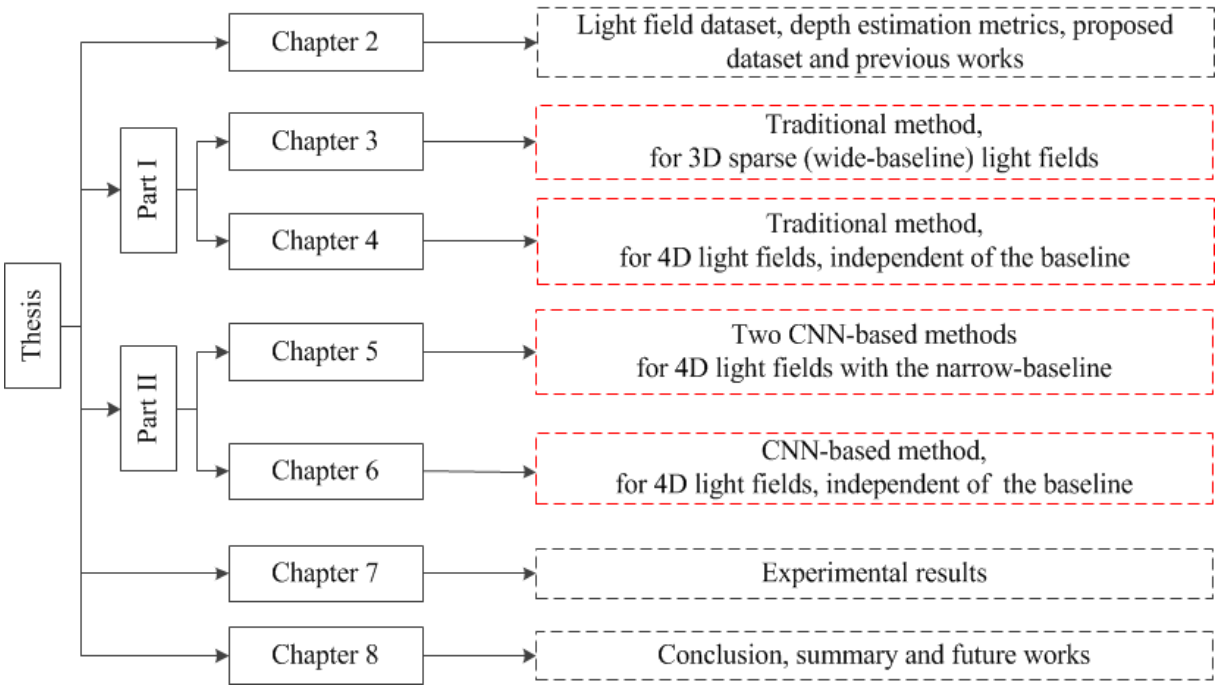


Figure 1.9: The outline of the following text in the thesis. The proposed depth estimation methods and/or datasets are in red dashed rectangles.

Chapter 2 describes the acquisitions of the light fields by hardware and software. Considering that the target of the thesis is focused on the depth estimation task, the detailed information of the available light field datasets for depth estimation is given, in which the classification, exemplar scenes statistics, and challenge attributes are included. Next, a large-scale synthetic wide-baseline dataset (*WLF*) with labeled data is introduced in order to train and validate the CNN models, and the testing dataset for evaluations are given. Afterwards, the metrics used for evaluating or comparing the competing depth estimation methods for light fields are given. Finally, the previous depth estimation works from 3D and 4D light fields are reviewed in detail.

Chapter 3 presents a robust 3D light field depth estimation framework (*R3DE*) to derive the depth from the sparse sampled (wide-baseline) 3D light field images (10 images in total).

Chapter 4 presents an extension of the framework in Chapter 3 to the 4D light fields (*S-R4DE*), which is scalable to the light fields with the different densities or baselines.

Chapter 5 alternatively puts forward two end-to-end convolution neural networks (CNN) (*HFNet* and *MANet*) sequentially for estimating depths from the 4D light fields with the narrow-baseline, in which the deep features, instead of hand-crafted features in Chapter 3 and 4, are extracted. Besides, the *MANet* is designed as a lightweight network.

Chapter 6 explores the feasibility and capability of the CNN in estimating depth from the 4D light fields with the wide-baseline. A novel end-to-end lightweight CNN, called *LLF-Net*, is built.

Chapter 7 displays the depth estimation results of the proposed traditional algorithms and the CNN-based algorithms from the 3D light fields and the 4D light fields respectively.

Chapter 8 concludes with the summary and potential future works.

CHAPTER 2

LIGHT FIELD DATASETS, METRICS AND PREVIOUS WORKS

2.1 Datasets

In recent years, the more number of light field datasets have been emerging, and also accessible to the public. These datasets have played a key role in the rapid development of new solutions using light field techniques to the problems in various vision or image processing tasks [14]. Most of the datasets, in general, are served to assess the performance of competitive solutions/algorithms, pushing the research field toward the more troublesome and challenging issues. To date, there has been a number of public datasets generated for the light field depth estimation. In the following we will give detailed descriptions of these datasets from various aspects, encompassing the acquisition, classification, scene illustration, statistics, and challenge attributes. In addition, the proposed dataset is introduced in the following text.

2.1.1 Acquisition

Acquiring the light fields could trace back to more than a hundred of years ago. To now, there are a variety of ways for capturing the 3D or 4D light fields. One way is to photographically capture the light fields by using the camera sensors, mainly containing the plenoptic camera, the camera gantry and the camera array. Another way of the capture is using 3D computer graphics software to render the 3D models with the environmental maps. The captured light fields from this software look not as physical as that from the camera sensors, but this way could help to reduce the research cost and serve as a complement for providing the ground truth that is hard to obtain in practice.

Plenoptic Camera

Plenoptic camera (aka light field camera) typically consists of a conventional camera with a matrix of lenslet array, and captures the 4D light fields by placing a lenslet array in front of the conventional image sensor [15]. One popular plenoptic camera

prototype is *Lytro Illum* camera [16], belonging to the plenotic camera 1.0 (defined by [17]) or standard plenoptic camera (defined by [18]). Fig. 2.1 shows the appearance of this prototype camera and the corresponding schematic. As seen in the schematic, a lenslet/micro-lens array (referred to as *uv* plane) is placed at the focal plane of the main lens (referred to as *st* plane) and one micro-lens focal length away from the image sensor. This layout results in its maximal angular resolution and minimal spatial resolution, where the angular resolution is relative large (i.e. with a *dense* set of views) at the sacrifice of the spatial resolution of the conventional photograph. Meanwhile, the *baseline* is limited by the aperture size of the main lens, thus it is always very narrow. In the figure, the light rays (in blue), for instance, are emitted from a point on the object, which will converge at a micro-lens. Then the micro-lens separates the directional/angular light rays to be imaged as a sub-image to the sensor behind the micro-lens. Actually, this sub-image is equivalent to a collection of pixels at the same (u, v) but at the different (s, t). The light rays (in red) pass through different image pixels at (u, v) that come from the same sub-aperture (s, t) on the main-lens, and the resulted image is called sub-aperture image. Therefore, providing the raw data from this plenoptic camera, we typically extract the same position pixel under each micro-lens to obtain the sub-aperture images as the input of the task.

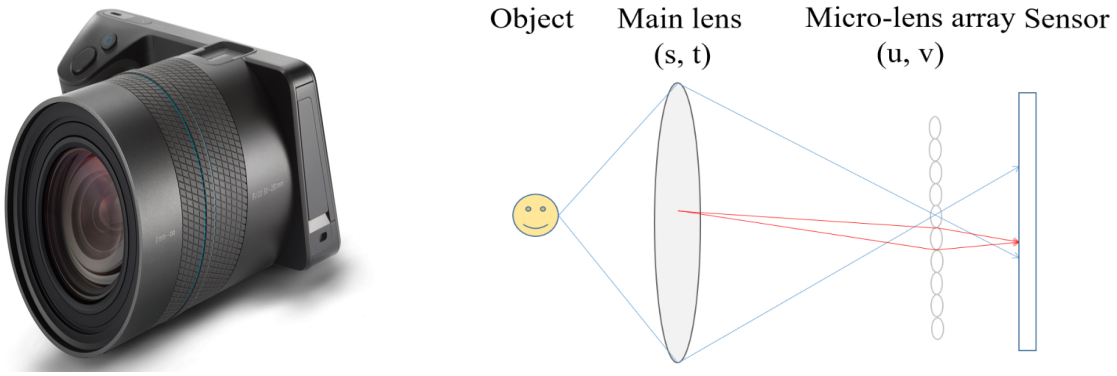


Figure 2.1: Left: Lytro Illum camera, Right: the corresponding schematic.

Camera Gantry

A linear or planar camera gantry is often used to capture the static 3D or 4D light fields since it is an effective device for acquiring the light fields under flexible configurations. The acquisition setup is comprised of a conventional camera, gantry, motor and computer etc. A user places a camera on the gantry, and the camera uniformly moves from one end to another end during which the movement is controlled by a motor and computer. According to whether the camera moves along a line or plane, the captured light fields are divided into the 3D light fields and 4D light fields respectively, as demonstrated in Fig. 2.2. Note that for this setup, the camera baseline, spatial and

angular resolution of captured light fields are flexible or selectable, and configured by the users. The minimal baseline is usually similar in size to that in plenoptic cameras, while the spatial and angular resolution of light fields are usually much larger than that of plenoptic cameras. The existing datasets involves the 3D light fields with 101 or 151 views [19] and the 4D light fields with 17x17 views [1], 21x21 views [20], 141x141 views [21]. In short, due to the flexibility, this setup is often used to explore capabilities of the light fields for different purposes in the research community.

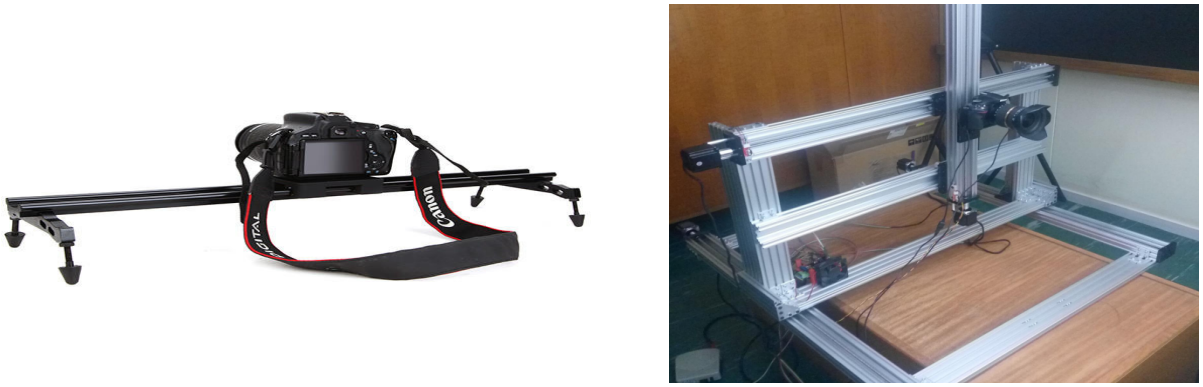


Figure 2.2: Camera gantries for capturing the 3D Light fields (left) and 4D light fields (right).

Camera Array

Camera array setup is used to capture the static and non-static 3D or 4D light fields (see Fig. 2.3). This setup differs from the camera gantries in that it is able to capture the movements in the scene. Moreover, the *baseline* is often much wider than that in the plenoptic cameras and the minimal baseline of the camera gantries, and the density/number of angular views is usually equal to or less than that of the plenoptic cameras. A disadvantage is that the whole capture is more expensive, tedious and challenging since all cameras need to be synchronized, and the focal length, aperture of all cameras need to be kept same and fixed. For the actual acquisition, it is mostly staged in a controlled laboratory environment, and is also controlled by the computer. In the past, there occurred a number of arrangements using conventional cameras: 1x100 camera array [2], 4x4 camera array [22], 2x3 camera array [23], 8x8 cameras [24], 8x12 cameras [25]. With respect to recent camera array setups, the number of views make almost no changes: 5x5 camera [26], 4x4 camera array [27], 3x5 camera array [21], but the spatial resolution might be increased.

1. http://lightfield.stanford.edu/lfs.html 2. http://www.fujii.nuee.nagoya-u.ac.jp/multiview-data/

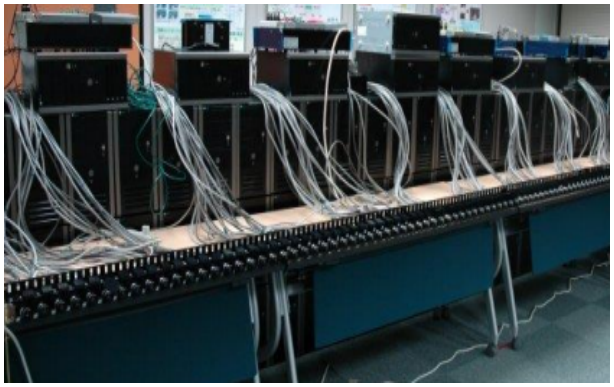



Figure 2.3: Camera array for capturing the 3D Light fields (left) and 4D light fields (right).

Computer Graphics Software

Acquiring (perfect) light fields by the camera sensors is usually difficult and expensive. Another effective and popular way is to create the light fields by using advanced 3D computer graphics softwares, e.g., open source software Blender [3], Unreal [4], Grand Theft Auto V game engine [5], etc. The creation mainly involves the collections of 3D computer-aided design (CAD) models and environmental maps on the Internet, artistically arranging the scenes and rendering photorealistic or non-photorealistic light field images, as is shown in Fig. 2.4. In addition to the elaborate arrangement of scenes (i.e., mimicking our real-world scenes), we might alternate to put randomly flying objects in a fixed 3D cube or others, which is also found effective [28]. They always take into account lighting, shading variations in order to reduce the gaps between the synthetic and real light fields. The graphics softwares not only are easy to configure the acquisitions (with different *density* and different *baseline*), but also are able to provide ground truth disparity, flow and object segmentations that are difficult to obtain for real-world light fields. In general, the software seems enough in creating the convincing light fields for the research purpose, and meanwhile tackles the time and cost issues that occur in acquisitions from the aforementioned light field setups.

In order to validate the generative performance of the proposed algorithms, the light field datasets generated from both the camera sensor (i.e. the Plenoptic Camera, Linear and Planar Camera Gantry and Camera Array) and the graphics software are taken into use for the assessments and comparisons.

2.1.2 Classification

There are a large number of light field datasets for the support of different tasks, including depth estimation. We will mainly classify the datasets that are used in light field

3. https://www.blender.org/
4. https://www.unrealengine.com/en-US/?lang=en-US
5. https://www.rockstargames.com/V/

Figure 2.4: An example of 3D graphics software for rendering light fields.

depth estimation literature here. Based on the type of light field setups, the existing light field datasets can also be grouped into the plenoptic (micro-lens array) camera, camera gantry and camera array dataset. Based on whether the light fields are photographically captured in the scene, these datasets can be classified into the real-world dataset and synthetic dataset. The detailed classification is summarized in Table 2.1. From Table 2.1, we find that there are no available synthetic datasets for camera array setups.

Table 2.1: Classification of current frequently-used light field datasets in previous works.

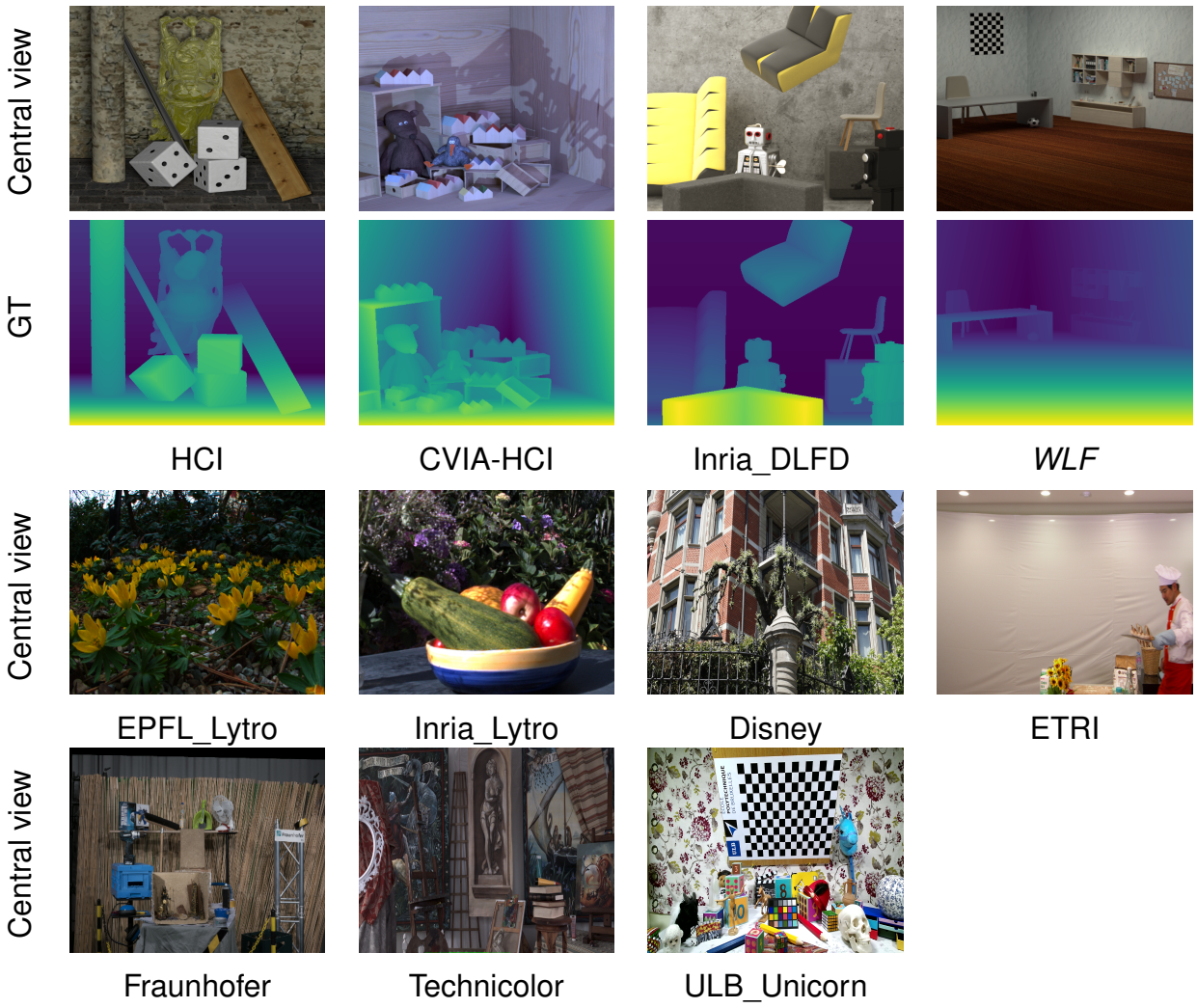
Type	Dataset
Synthetic Micro-lens array	HCI [29], CVIA-HCI [30], Inria-DLFD, Inria-SLFD [31]
Synthetic Camera array	-
Real-world Micro-lens array	EPFL-Lytro [32], Inria-Lytro [33]
Real-world Camera gantry	Disney [19], Fraunhofer [20], ULB_Unicorn [21]
Real-world Camera array	Google [26], ETRI [34], Technicolor [27], ULB [35]

2.1.3 Scene Illustration

Various categories of objects, including animals, vegetables, building and food, are uniformly distributed in the current light field datasets. To illustrate scenes, some examples from the aforementioned datasets are shown in Fig. 2.5. We can notice from

this figure that the synthetic datasets are mainly designed as the indoor scenes, while the real-world datasets are comprised of both the indoor and outdoor scenes.

Figure 2.5: Scene illustration.



2.1.4 Statistics

Dataset statistics are given in Table 2.2. We can observe that the (synthetic and real-world) micro-lens array datasets have much lower spatial resolution than that of the camera gantry datasets and the camera array datasets. The angular resolution of the camera gantry dataset is larger than that of the other two setups. The baseline of datasets from the micro-lens array camera and the camera gantry is narrow (an interval of more or less than 1 millimeter), and the related disparity range is quite limited. In contrast, the baseline of the camera array dataset is wide (an interval of several centimeters) and and the related disparity range is also large. The synthetic datasets for the micro-lens array camera are the only datasets to provide the ground truth depth maps, and some of them are utilized as the training set for learning-based algorithms.

Table 2.2: Datasets statistics of current frequently-used light field datasets for the depth estimation task. GT: ground truth, AR: angular resolution, SR: spatial resolution.

Dataset	#train	#test	AR	SR	scene	baseline	#GT
HCI		7	9x9	768x768	image	narrow	7
CVIA-HCI	16	12	9x9	512x512	image	narrow	28
Inria_SLFD	44	-	9x9	512x512	image	-	53
Inria_DLFD	-	-	9x9	512x512	image	narrow	39
EPFL_Lytro		118	15x15	434x625	image	narrow	✗
Disney		5	101 or 151	2622×1718	image	narrow	✗
Fraunhofer		9	21x21	3976x2656	image	narrow	✗
ULB_Unicorn		1	141x141	1920x1080	image	narrow	✗
Google		6	5x5	[1024, 1764]	image	wide	✗
ETRI_Chef		300	5x5	1920x1080	video	wide	✗
Technicolor_Painter		372	4x4	2048x1088	video	wide	✗
ULB_BabyUnicorn		300	3x5	3712x2064	video	wide	✗

Note: For the HCI dataset, only the scene with the full ground truth is counted. For the Disney dataset, the smallest spatial resolution is shown here since it varied in different scenes. For the Google dataset, the spatial resolutions consists of the 1024x1024 and 1764x1764. The Technicolor dataset is specified to the Technicolor_Painter.

As a fact, these are designed for the narrow-baseline scenario, however, there are no available synthetic datasets with the ground truth for the wide-baseline scenario. Meanwhile, the ground truth depth maps are not provided the existed real-world datasets.

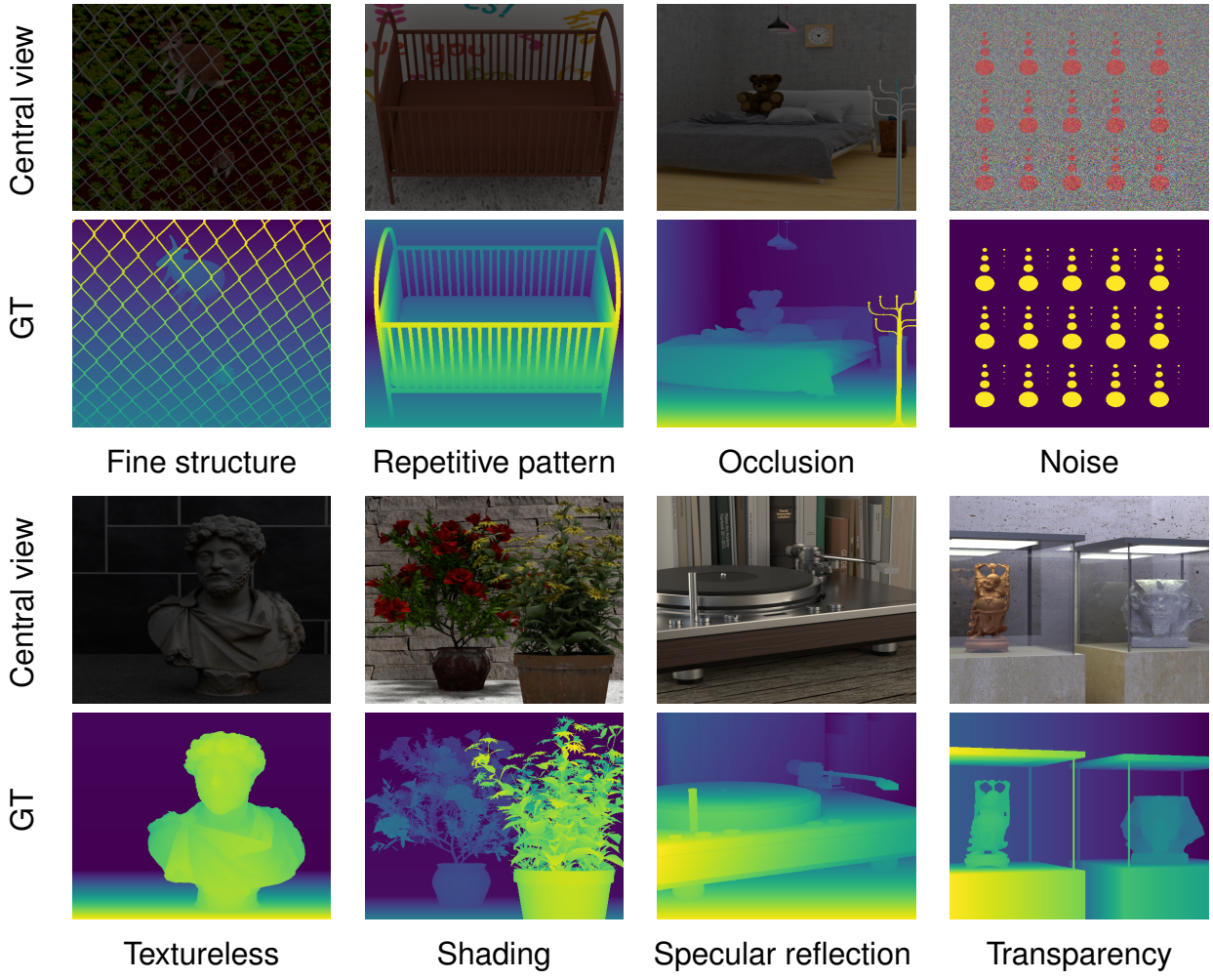
2.1.5 Challenge Attribute

For the more in-depth analysis, it would be essential to summarize the challenge attributes occurred in light field depth estimation, as are given in Table 2.3 and demonstrated in Fig. 2.6. In Table 2.3, a list of the challenge attributes and the brief descriptions are shown. The fine structure, textureless and occlusion are the most frequency issues considered in the literature, which might be put down to the majority of such image regions in the existed light field datasets. Likewise, in this thesis, these three issues are also paid more attention to. Some other issues, e.g., the noise or **non-lambertian** (the captured radiance changes with the camera viewpoint), were explicitly taken care of by previous works but less (implicitly) considered or not considered in the proposed algorithms. As with the creation of training dataset, it is worthy of including these attributes completely in order to prevent the model from over-fitting the specific issues.

Table 2.3: Challenge attributes.

Attribute	Description
Misalignment	Calibration and/or rectification error
Fine structure	Thin segment or object, e.g., fence, fur
Repetitive pattern	Repetitive texture, e.g., checkerboard
Occlusion	Foreground objects occlude background, e.g., flower, tree
Noise	The mechanical error
Textureless	Object or background with low/no texture, e.g., sky, wall
Shading	Non-lambertian: light
Specular reflection	Non-lambertian: metal, water, mirror
Transparency	Non-lambertian: glass, plastics

Figure 2.6: Visualizations of challenge attributes.



2.1.6 Proposed WLF Dataset

As shown in Table 2.2, most of available datasets belong to the narrow-baseline, which are composed of a grid of 9x9 light field image views and with the small disparity range [-4, 4] (HCI [29], CVIA-HCI [30], and DLFD [36]). The CVIA-HCI includes 16 frames with available ground truth depths that are provided for training. Models trained on the CVIA-HCI and/or even other similar datasets are not able to infer depth well for wide-baseline datasets due to source and target disparity range issues. The available wide-baseline light field datasets are rare. Moreover, training CNNs requires a large amount of labelled data, but there were no large-scale public wide-baseline light field datasets for this purpose.

For a new dataset creation, a straightforward way is to collect real data and label them through physical depth sensing devices (e.g., structure light sensor or LiDAR). However, it is difficult, tedious and expensive: structure light sensor is cheap but usually produces inaccurate depth which may cause performance degradation in CNNs models, while LiDAR offers incomplete accurate depth but is unaffordable. Similar to narrow-baseline scenario, we put efforts into building a synthetic wide-baseline dataset with accurate (ground truth) depths, aiming at training and evaluating CNN models, inferring depth for real-world datasets, and serving to research community for future promising researches. We use 3D computer graphics software to create a large-scale, synthetic *W*ide-baseline *L*ight *F*ield dataset with diversities, called ***WLF***.

Specifically, we construct a large-scale, wide-baseline synthetic multicamera light field capture dataset *WLF*. The total number of the light fields is 381, which is around 14 times larger than that of the popularly-used dataset CVIA-HCI. Each light field provides 9x9 angular (RGB) images and ground truth disparities as similar to the CVIA-HCI dataset. The light fields involve high resolution (1920x1080) and low resolution (512x512) images.

To enrich the dataset diversity, the *WLF* dataset is constructed in two scenarios: Hand-designed and Flying-objects. The scenes in Hand-designed and Flying-objects scenarios are rendered by the Cycle engine in open source software Blender [6]. The statics of the *WLF* dataset is given in Table 2.4, and Fig. 2.7 shows the rendered samples from these two scenarios.

Hand-designed Scenario: We carefully collect free 3D models from different websites [7] with free licenses and elaborately assemble them to create physically plausible and meaningful scenes. Each scene contains more than two challenges in depth estimation: fine structure, repetitive pattern, occlusion, shading, glossy appearance. The hand-designed scenario counts the aesthetic impression, but the manual design of 3D scenes is tedious and expensive, which causes difficulties to generate a large size

6. https://www.blender.org/ 7. https://chocofur.com, https://sketchfab.com, https://free3d.com

Table 2.4: Datasets statics of *WLF*

Dataset	#train	#test	spatial resolution	disparity range
Flying-objects	345		512x512	[0, 50]
Hand-designed	24	12	1920x1080	[0, 50]

dataset. This subset includes 36 scenes, and is split into 24 training scenes and 12 test scenes.

Flying-objects Scenario The richness of the dataset content is significant, therefore we attempt to render new scenes with flying objects in a faster way, which is inspired by recent advances of synthetic scenes with flying objects [28, 37, 38] in deep learning methods. Specifically, we carefully collect a large number of 3D models from the websites[7] and [39], and collect the texture images and environmental maps from Google Image. We then make a 3D cube in 3D space of Blender software, and the surfaces of cube are randomly textured. Next, a number of objects, which vary from 2 to 20, are randomly and automatically put in the cube, including 1-15 static objects and 1-5 random moving objects. The objects are randomly scaled, rotated and translated. Moreover, the light intensity is random, and the virtual light field cameras are slightly translated. This subset includes 345 scenes, and is provided for training models.

2.1.7 Testing Datasets

3D Light Fields

The Disney dataset [19] is chosen from the datasets listed in Table 2.2, since this was specially built for the 3D light fields. This dataset includes the densely sampled light fields with the narrow-baseline, and contains challenging content, such as textureless regions and occlusion regions. We choose 10 angular views from the dense light fields for test, where 10 or 15 views are skipped to obtain the wide-baseline.

4D Light Fields

The narrow-baseline dataset and the wide-baseline dataset are both employed as the test set. Note that the test set is held out from the whole dataset for the sake of an unbiased evaluation of a model trained on the training set.

Narrow-baseline Datasets: we choose the frequently-used synthetic datasets in previous works for qualitative and quantitative comparisons, and the real-world (narrow-baseline) dataset for quantitative comparisons only. We use the 7 test scenes from the HCI synthetic dataset [29] and the 8 test scenes from the CVIA-HCI synthetic dataset. The photorealistic and non-photorealistic scenes are encompassed in the synthetic

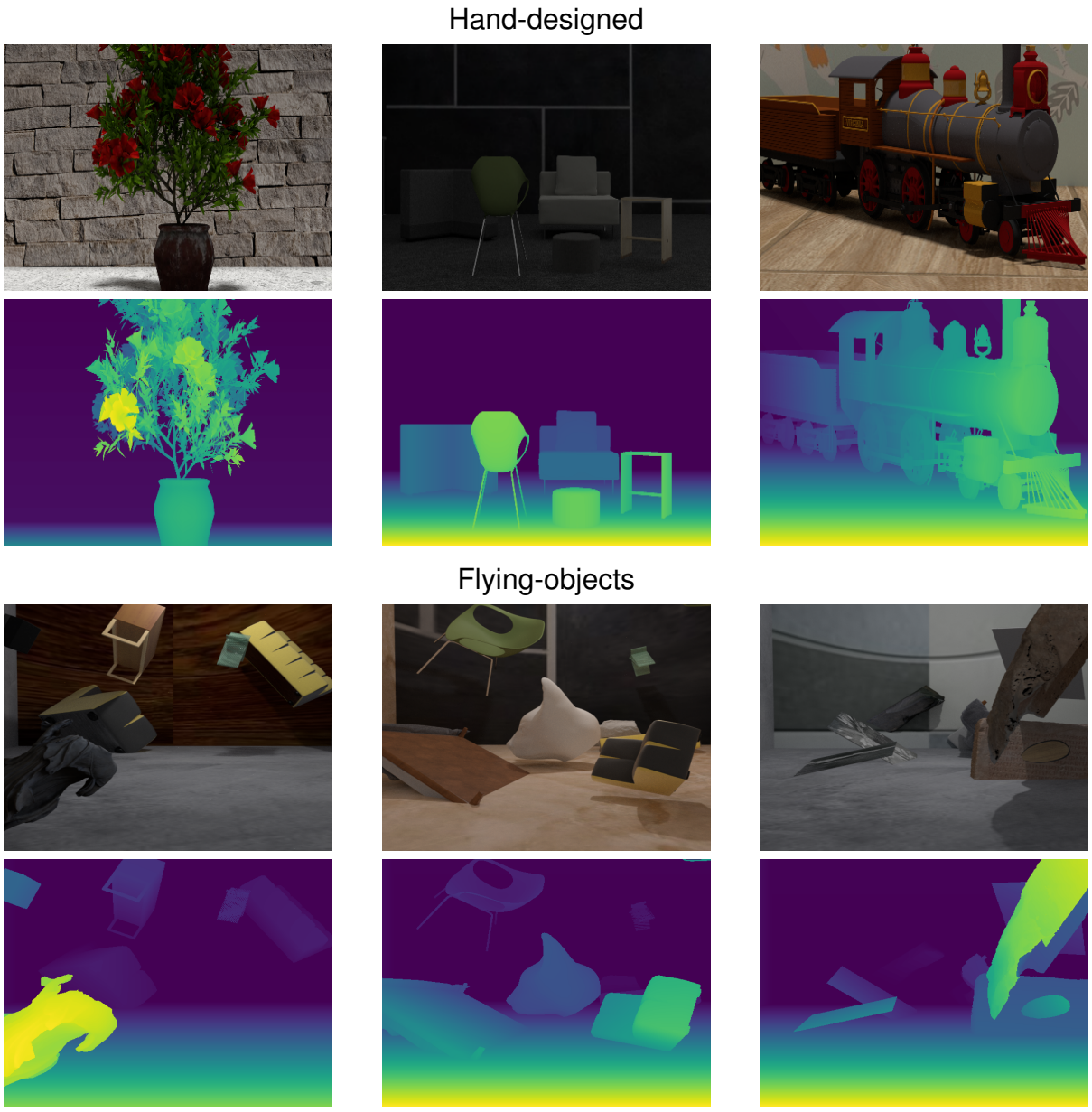


Figure 2.7: Examples of *WLF* dataset: the central view and colored ground truth disparity map are shown.

datasets, where four scenes in the CVIA-HCI are non-photorealistic. As with the real-world dataset, we use the EPFL-lytro [32] dataset for purpose. This dataset only contains the raw data from the Lytro Illum camera, we extract the sub-aperture images from these data through a light field toolbox [14].

Wide-baseline Datasets: we choose the proposed synthetic datasets *WLF* for qualitative and quantitative comparisons. Specifically, we use all test scenes (12 in total) of *WLF*, comprised of the challenging photorealistic scenes. As with the real-world dataset, we use the Google and ULB_Unicorn dataset, which contains most of the challenging attributes in Table 2.3. For ULB_Unicorn dataset, a number of views are selected by skipping 15 views to reach the wide-baseline.

2.2 Metrics

For depth estimation evaluation, various metrics are exploited by measuring the similarity between the generated disparity map *D* and the ground truth *G*, which are categorized into the quantitative metric and the qualitative (visual) metric in the literature. As with the quantitative metric, the Mean Square Error and Bad pixel are adopted in the thesis, since these were the widely-used metrics in depth estimation literature and benchmarking websites [30, 40, 41].

- Mean Square Error (MSE): is computed as the average square difference between all pixels in *D* and *G*.

$$\mathrm{MSE} = \frac{1}{h_1 \times w_1} \sum_{i=1}^{h_1 \times w_1} (D(i) - G(i))^2 \tag{2.1}$$

  where h_1 and w_1 represent the height and the width of the predicted depth map respectively. A smaller MSE value means a higher similarity and a better performance. The MSE is displayed by its numerical value multiplied by 100 in comparisons hereafter.

- Bad Pixel: is computed as the percentage of the absolute difference between *D* and *G* that is greater than a threshold.

$$\mathrm{Bad-t} = \frac{|\{|D(i) - G(i)| > t, i \in h_1 \times w_1\}|}{h_1 \times w_1} \tag{2.2}$$

  Based on the benchmark [30] and existing methods, various thresholds *t* are used in assessment. A smaller BadPix value means a better performance.

With respect to the Bad Pixel metric, the thresholds differ on the *baseline*. 1) Narrow-baseline: the two thresholds 0.1 and 0.07 are used as the metrics, which are defined

in [29, 30, 42]. 2) Wide-baseline: to measure the accuracy of reconstructed depth from wide-baseline light fields, the larger threshold of the bad pixels are set to 0.15, 0.3, 0.6 and 1.

2.3 State-of-the-art

To date, a large number of works have been put into efforts for improving depth prediction from light fields. The works could be categorized into depth estimation from 3D light fields and depth estimation from 4D light fields on aspect of the light field input representation. With respect to the 4D light fields scenario, the works could be further classified into the traditional methods and deep learning-based methods. In this section, we will review most of the related works in detail in order to make it self-contained and readers better understand the development of depth estimation from light fields and our proposals in the following chapters.

Before going into the reviews, we firstly starts with a description of the terminologies that frequently appear in the related works. These terminologies or techniques could be classified into four categories: representation, cost function, aggregation and optimization, as are given in Table 2.5. Here we mainly explain the representation with the more details. What is different from the two-view stereo and multi-view stereo is that there are more representations available for extracting depth from light fields, including Defocus, Epipolar plane image (EPI), Focal stack, Multi-view stereo and SCAM. One or two representations are usually exploited in most of the related works. Specifically, **Defocus**, i.e. a integration of multiple images of light fields focused at different depths, is not sensitive to the repetitive patterns or noises dues to its blurriness artifact. **EPI** is constructed by a stack of image scan-lines from a line of views, and when the light fields are densely-sampled, the EPI-line is a continuous line, in which the slope of EPI-line is proportional to the disparity value. Since the depth estimation is reformulated into the slope calculation, the non-lambertian issue is somewhat alleviated, thus the EPI-based representation is widely used in previous works for (narrow-baseline) light fields. **Focal stack**, i.e. a sequence of images captured with different focus, exhibits local color symmetry for texture boundary pixels regardless of the noise or the changes of spatial resolution or angular sampling rate. Meanwhile, this will partially disappear for pixels on the occluder and disappear for pixels at the true depth on the occluded surface [43]. **MVS**, one technique adopted in the classical 3D reconstruction, is used to seek corresponding pixels from all sub-aperture images or views of the light fields. This technique allows the estimate of large disparities and a considerable or good estimate with few sampled views. **SCAM** (or angular patch) is constructed by the 2D points at different angular positions, projected from the 3D point. Since the edge in SCAM has

the same orientation as the occlusion edge in the spatial domain, it is a good candidate to handle occlusions, but is conditioned that the resolution of SCAM should be large enough.

Secondly, the photo-consistency will be explained since it is an important assumption considered in depth estimation from light fields. This assumption is that the same 3D point is seen from the different directional rays as the same color. Actually, this assumption does not hold when the point on a non-lambertian surface or occluded surface. Thus this issue is carefully taken care of in the state-of-the-arts.

Table 2.5: A summary of the terminologies or techniques used in light field depth estimation methods.

	Terminology	Description
Representation	Defocus	Multiple image exposures focused at different depths
	EPI	The slope of EPI-line is inversely proportional to the disparity
	Focal stack	A sequence of refocused images
	MVS	Multiple stereo, the displacement of same points in each pair is the disparity
	SCAM	An angular sampling image or angular patch
Cost function	Angular Entropy	The light radiance randomness of the angular patch
	KDE	Kernel density estimation used for computing the depth probability
	SAD	The sum of absolute differences
	SPO	Spinning parallelogram operator for locating EPI-lines and calculating orientations
	SSD	The sum of squared difference
	Structure tensor	The second-moment matrix used for estimating the slope of EPI-line
	Variance	The expectation of the squared deviation of pixels differences in SCAM
	ZSSD	The zero-mean SSD
Aggregation	BF	Bilateral filtering (an edge-preserving filter) used for filtering cost slices
	GF	Guided filer (an edge-preserving filter) used for filtering cost slices
	MWBM	Multiple window block matching using elongated windows with different orientations
	Sum	A sum of costs of pixels in a window
Optimization	Least square	CO*, Minimizing the energy function by a close-form solution
	SGM	DO*, Semi-global matching computed by different directions with different passes
	MRF	DO*, Markov Random Field solved by graph cuts [44] or belief propagation [45]
	Variational	CO*, Total generalized variation solved by functional lifting [46]

CO*: continuous optimization. DO*: discrete optimization.

2.3.1 Depth From 3D Light Fields

The previous works for 3D light fields are modeled in a traditional way, and a comprehensive overview of some related works are give in Table 2.6.

Kim et al. [19] made an early attempt to compute depths from the 3D light fields, where a multi-scale framework taking as input of the light fields with high spatial reso-

Table 2.6: Overview of the state-of-the-art 3D light field depth estimation methods ordered by date.

Method	Year	#Views	Baseline	Representation	Cost calculation	Optimization
Kim et al. [19]	2013	$\geq$ 100*	Narrow	MEPI*	KDE	-
Yu et al. [47]	2013	$\geq$ 2	Narrow, Wide	MVS	SSD	MRF
Lv et al. [48]	2015	$\geq$ 100	Narrow	EPI	SAD, BF	Least square
Huang et al. [49]	2016	$\geq$ 100	Narrow	EPI	KDE	SGM
Jorissen et al. [50]	2016	10	Wide	EPI	KDE, Sum, SURF	-

100*: this work mainly uses more than 100 views, but only using 10 views is still able to reconstructing good depth maps. MEPI*: this represents the multi-scale EPIs.

lution was proposed. To cope with the multi-scale EPIs, a fine-to-coarse (**FTC**) strategy was put forward to progressively estimate depths. Specifically, the estimation starts at the fattening (horizontal) edges of the highest scale level first, and then proceeds to the fattening (horizontal) edges and/or non-fattening (horizontal) edges at the lower scales. Note that at each scale level, the pixel-based matching cost is calculated using kernel density estimation (KDE). Since a large number of views (more than 100) with a narrow-baseline are employed, the high quality depth is reconstructed, even without global optimization that was commonly used in a late step of depth estimation pipeline. Besides, when reducing the number of views to 10 views without changing the baseline, the quality of depth map is still found acceptable.

Yu et al. [47] presented a single-scale framework based on the multi-view stereo (MVS) method for the 3D light fields. Hundreds of line segments are detected by the line segment detector, and then encoded as the hard constraints into the global optimization, i.e. the line assisted graph cuts (**LAGC**) to improve depth estimation. With respect to the line segment, it is a double-edged sword: when it works well, this contributes to the disparity-preserving at occlusion regions; while large errors are inevitably occurred if the depth of the line segment is incorrectly estimated.

Huang et al. [49] followed the work [19] and presented a modified framework, in which both the horizontal and vertical edge in EPI were proposed to calculate matching costs. Another main modification from this work is that the fine-to-coarse estimation is replaced with the semi-global matching (SGM), which globally optimizes the depths.

The work by Lv et al. [48], i.e. one of the mostly related work to the proposed *R3DE* in the Chapter 3, proposed a 1D window-based cost aggregation approach to select the optimal orientation (being equivalent to the disparity) for each pixel in EPI, where the truncated sum of absolute differences (SAD) of both the radiance and gradient are calculated, followed by a weighted sum of costs of all pixels in the horizontal edge of EPI using bilateral filtering. Besides, the sub-pixel estimation based on quadratic poly-

Table 2.7: Overview of the state-of-the-art 4D light field traditional depth estimation methods ordered by date.

Method	Year	Shape of views	Baseline	Representation	Cost calculation	Optimization
Wanner et al. [51]	2012	Cross-hair	Narrow	EPI	Structure tensor	Variational
Tao et al. [52]	2013	Full*	Narrow	Defocus, MVS	Laplacian, Variance	MRF
Chen et al. [53]	2014	Full*	Narrow, Wide	MVS	KDE, GF	Least square
Jeon et al. [54]	2015	Full*	Narrow	MVS	Phase shift, SAD	MRF
Wang et al. [55]	2015	Full*	Narrow	SCAM	Variance, SSD	MRF
Lin et al. [43]	2015	Full*	Narrow	Focal stack	KDE	MRF
Zhang et al. [56]	2016	Cross-hair	Narrow	EPI	SPO, GF	MRF*
Zhu et al. [57]	2016	Star	Narrow	EPI	Structure tensor	-
Williem et al. [58]	2017	Full*	Narrow	Defocus, MVS	Entropy, SAD	MRF
Navarro et al. [59]	2017	Cross-hair	Narrow, Wide	MVS	ZSSD, MWBM	Variational
Zhu et al. [60]	2017	Full*	Narrow, Wide	SCAM	Structure tensor	MRF
Huang et al. [61]	2019	Full*	Narrow, Wide	MVS	GSM	MRF
Mishiba et al. [62]	2020	Full*	Narrow, Wide	MVS	SAD, Sum	WMF

Full*: the full grid light field views, MRF*: this is turned on for real-world light fields.

nomial interpolation is utilized for addressing the quantization issue caused in previous steps. Finally, the re-projection is used to handle occlusion and then the reliable depth is propagated to fill the depth holes using least square based optimization.

Jorissen et al. [50] also followed the work [19] and made attempts to modify the framework in order to adapt the framework to the challenging scenario: few light field views with the wide-baseline (including 10 views in total, and each view is sampled from every 10 or 15 views). The pixel-based matching cost is replaced with the window-based matching cost, while the fine-to-coarse estimation is replaced with the SURF-based cost aggregation. This change indeed makes this framework better reconstruct the depth from the sparse sampled light fields.

To conclude, most of the previous works somewhat rely on a large number of light field views with the narrow-baseline to well recover depths, but the quality of recovered depth maps becomes much worse when the number of views are drastically decreased and/or the baseline is much wider. Thus a new solution to achieve high depth accuracy in the sparse (wide-baseline) 3D light fields is desired.

2.3.2 Depth From 4D Light Fields

Until now, there exists a great number of works that are dedicated to depth estimation from 4D light fields, which includes traditional and deep-learning methods. The representative related works will be reviewed in detail below.

Traditional Methods

Depth estimation from light fields begins with the the traditional way, on which the occlusion is paid much attention in previous works. Actually, these works somewhat shares some techniques in common while the specific techniques are used and customized for their purpose. The comprehensive overview of the related works are give in Table 2.7.

Wanner et al. [51] introduced an EPI-based framework for 4D (narrow-baseline) light field depth estimation, where the local estimate and the global optimization were sequentially carried on the horizontal and vertical EPIs of light fields. Specifically, the structure tensor technique was proposed to estimate the slope of EPI-line and the confidence of local estimate, which was solved by [63]. Note that the structure tensor was computed on 3x3 kernels, on condition that the displacement between neighboring views should be less than two pixels. Besides, at occlusion regions, it was difficult for this tensor to locate the EPI-line and obtain the slope of this line, which easily leads to over-smoothing results in such regions. Then the estimates from the two directional EPIs were separated by a variational-based energy function, which showed a higher accuracy than one integrated estimate from two local estimates. Since the disparity estimate is formulated into computing the slope of EPI-line, the non-lambertian points or textureless points are more or less well coped with.

Tao et al. [52] presented a multiple cues-based framework by the defocus and correspondence cues. For the defocus cue, the sheared EPIs are integrated across one dimension, followed by the Laplacian operator being applied onto a window of pixels around each current pixel in EPI. Since the defocus blurs the image regions, this cue makes the depth estimation less sensitive to the repetitive patterns and noises. For correspondence cue, the window-based matching cost is computed, where the variance metric is used to measure the cost for each pixel. Finally, the global optimization based on MRF is employed to remove the ambiguities resulted from the local estimate.

Chen et al. [53] proposed a bilateral consistency metric to select the visible pixels for explicitly handling the occlusion issue. This metric relies on the matching cost calculated beforehand, which is implemented by the Gaussian kernel metric. Then the bilateral filter is used to estimate the probability of each pixel in different viewpoints and apply a threshold to the probabilities to determine if the pixel is visible in more than half of the views. Then the Gaussian kernel metric is reused onto the visible pixels to obtain the matching costs. After that, the guided filer and the local confidence measure is used to correct the wrong estimated pixels in textureless regions, and then the reliable depth of pixels is propagated to fill the close-by unreliable pixels using least square based optimization. This work tends to handle the heavy occlusion well, however, as reported by the author, this causes the limitation, i.e. this method will not work well if a

small set of light field views (e.g., less than 10) are used as inputs.

Jeon et al. [54] presented a sub-pixel interpolation based framework (dubbed as **LF**), in which the phase-shift interpolation was used for sub-pixel shifts. The phase-shift was implemented by the discrete 2D Fourier transform and inverse transform, and led to the sharper depth than linear interpolation. The similar window-based cost aggregation with [48] were employed, and then the weighted median filtering were used to remove the noise in the local estimate. Finally, the local estimate were optimized by minimizing the global energy function, followed by being refined from the discrete depth after the multi-label optimization to a continuous disparity depth while keeping depth discontinuities.

Wang et al. [55] proposed a SCAM-based occluder model (dubbed as **LF_OCC**) for explicit occlusion handling, based on the observation that the edge in SCAM has the same orientation as the occlusion edge in the spatial domain. This model is built as a single occluder model, assuming that the pixel on an occlusion edge is only occluded by a single occluder. Under this assumption, only one of two regions spit by the occlusion edge shows photo-consistency, and the occluded pixels are easily focused to correct depths, which improves estimation at occlusion areas. For the edge, the Canny operator is used for detecting edges and getting orientation, and then the detected edge is dilated in order to include the pixels that are un-occluded in the current view but occluded in other views. The similar cost measure with [52] is used for calculating matching cost. The depth, refocus and correspondence cues are used for extracting occlusion boundary. In the last step, the matching cost and the occlusion boundary are integrated into a global energy function for the final depth map.

Lin et al. [43] proposed a focus stack based framework, in which the images from in-focus to out-of-focus by the max focal shift were generated. From the focal stack, the observation is made that the non-occluding pixel along the focus dimension exhibits the symmetry centered at the in-focus slice. The in-focus cost and the multi-view cost calculated by the Gaussian kernel onto the radiance and gradients are integrated into MRF optimization, which is also solved by graph cuts.

Zhang et al. [56] proposed a spinning parallelogram operator (SPO) onto both horizontal and vertical EPI-lines to estimate the depth. The operator was used to locate the EPI-line and obtain the EPI-line orientation by calculating the cost of histograms in two regions of spinning parallelogram, which were spit by the EPI-line. The guided filter was also used to aggregate the costs in order to remove the noise or ambiguities caused in the SPO operation. Until this step, the depths from synthetic light fields could be enough recovered, but this is not the case in the real-world light fields. Therefore, the author further applied the global optimization proposed by [54] to refine the depth map.

Zhu et al. [57] proposed a multiple directional EPIs based framework that took as

input the EPIs sliced from the horizontal, vertical, left diagonal and right diagonal views respectively. For one pixel at occlusion regions, the EPI-line might be difficultly located in one of EPIs as occurred in [51], but might be possible to be detected in other EPIs. Considering that the foreground that occludes the background always has a small depth, the largest depth estimated from four EPIs will be assigned to the occluded (background) pixel for tackling the occlusion issue.

Based on the works by [52], [54] and [55], Park et al. [58] proposed new cost matching measures, in which the occlusion-aware angular entropy and adaptive defocus costs were calculated in order to be robust to occlusions and noises. For the angular entropy measure, it was found capable to generate a unary minimum cost for the occluded pixel. This measure was computed for each channel, and then a weighted sum of the costs from max pooling and averaging over three channels was made. For the defocus measure, instead of the direct computation of defocus on a whole patch, the patch (with the size 15 × 15) was divided into 9 (5x5) sub-patches, and the defocus was computed for each sub-patch that was less affected by the blurring. Besides, the color similarity constraint used for distinguishing the ambiguity between the occluder and occluder regions was enforced and combined with the defocus from the sub-patches for being less sensitive to the noise and occlusion.

Navarro et al. [59] proposed a multiple window-based framework (dubbed as MWBM) to search correspondences across multiple scales in order to better recover the depth at depth discontinuity regions. Specifically, they downsize every sub-aperture image via bicubic interpolation to generate the image pyramid with three scales, and then progressively estimate depth from the coarse to fine scale where the estimation at a coarser scale is used as initialization in the next finer scale. This coarse-to-fine strategy helps to reduce the search space in finding the correct correspondence, and predict the disparity in low-texture regions well without losing depth discontinuity.

Following the work [55], Zhu et al. [60] proposed a multiple-occluder model since when a multiple-occluder appears, the work [55] cannot work well because the single-occluder assumption in [55] does not hold. An un-occluded view selection and re-selection scheme were adopted. Since its accuracy relies on the occlusion boundary, the depth edge map was combined with an edge map to improve occlusion boundary detections. Finally, the occlusion boundary was integrated into a MRF-based energy function, which was solved by graph cuts.

Huang et al. [61] presented an empirical Bayesian framework (named **RPRF**) for robust depth estimation, in which the scene-dependent (MRF) parameters are estimated before the following global optimization as adopted by most of previous works (e.g., [54], [55], etc). Specifically, the soft expectation-maximization (EM) was proposed to estimate the MRF (data and smoothness term) parameters for a good distribution fitting of pseudo-likelihood that was separated from global likelihood. Then the hard EM

is used in constructing the data and smoothness energy. The global energy is solved by belief propagation [45] for the final depth map.

Mishiba et al. [62] presented a fast depth estimation framework, in which the cost calculation and optimization were reasonably simplified for fast computations. The sum of the pixel difference in horizontal and vertical directions were calculated to obtain the matching cost using only one-bit feature for each pixel, and then the (fast) box filtering was utilized to aggregate a window of costs to generate the cost volume. The interpolation was done for the matching cost that had the minimum cost among all cost slices of the volume so that the total number of candidate cost slices can be reduced. Afterwards, a fast weighted median filter (WMF) is used in a coarse-to-fine manner to speed up the convergence in minimizing the global energy function. Note that, the authors also proposed the meaningful (avoid the occlusion and redundancy) viewpoints off-line selection from all viewpoints to accelerate the whole processing time.

To conclude, most of the previous works for 4D light fields also require a large number of light field views with the narrow-baseline to well recover depths. When the number of views is decreased and/or the baseline is wider, the (heavy) occlusion is still difficult to be handled. Besides, the detection of the occlusion boundary is still not accurate enough, which will result in the over-smoothness after global optimization. Thus, a new solution to achieve high depth accuracy in the both the dense (narrow-baseline) and sparse (wide-baseline) 4D light fields are desired.

Deep learning-based methods

In recent years, deep learning-based methods have gained much attention in estimating depth from light fields. With the creation of light field training datasets that contain ground truth depths or disparities, the deep learning techniques are applied to solve the depth prediction problem from the statistical perspective, having shown appealing performance in both depth accuracy and speed. When the number of training examples is sufficient, the input-output relation is more possible to be well learned. Most of the related works for 4D light fields are reviewed in this section, and an overview of these works is given in Table 2.8.

As is shown in Table 2.8, most of previous works are focused onto the 4D narrow-baseline light fields. Parts of works are based on the EPI-line property, and the related CNNs are proposed to learn the relationship between the EPI-line and the labeled data. For instance, Heber et al. [64], Luo et al. [66] and Feng et al. [67] feed the input of (horizontal and vertical) EPI patches to the CNN where the network learns the proportional relation between the slope of the EPI-line and depth (cf. Fig. 1.2 in Chapter 1). Heber et al. [65] and Shin et al. [68] feed one or more 3D EPI-volumes to the network, and attempt to let the network learn the EPI-line property or the epipolar geometry property.

Table 2.8: Overview of the state-of-the-art 3D light field depth estimation methods ordered by date.

Method	Year	Shape of views	Baseline	Representation	Formulation	Optimization
Heber et al. [64]	2016	Cross-hair	Narrow	EPI	Classification	Variational
Heber et al. [65]	2017	Cross-hair	Narrow	EPI-volume*	Regression	-
Luo et al. [66]	2017	Cross-hair	Narrow	EPI	Classification	MRF
Feng et al. [67]	2018	Cross-hair	Narrow	EPI	Classification	Variational
Shin et al. [68]	2018	Star	Narrow	EPI-volume	Regression	-
Zhou et al. [69]	2019	Full*	Narrow	Focal stack	Classification	-
Leistner et al. [70]	2019	Cross-hair	Narrow, Wide	Shifted EPI	C+R*	-
Shi et al. [31]	2019	Diagonal	Narrow, Wide	MVS	Regression	RefineNet

FULL*: the full grid light field views, EPI-volume*: the horizontal EPI slices of all scan-lines, C+R*: Classification+Regression

Next, the details with respect to these works will be described below.

Heber et al. [64] made an early attempt to utilize deep learning techniques to do depth estimation from 4D (narrow-baseline) light fields. The shallow (five-layer) network was proposed to learn the input-output relation (i.e. the proportional relation between the slope of the EPI-line and depth), where four convolutional layers and one fully-connected layer were included. The network takes as inputs the concatenated two EPI patchs (with a small size 31 × 11), and then predicts the depth by classifying the EPI patches. Since the depth map produced by the network is noisy or unreliable at textureless regions, a global optimization Total Generalized Variation (TGV) introduced by Bredies et al. [71] is employed, in which the primal-dual algorithm proposed by Chambolle et al. [72] is adopted to find a global minim solution.

Heber et al. [65] presented a novel U-shaped deep network based on a modified version of a Fully Convolutional Network (FCN) [73]. The changes from his previous work [64] are that the last fully-connected layer is replaced with the convolutional layer, and the (discrete) classification is thus reformulated into the (continuous) regression for inferring the depth. Another change is to use 3D convolutions to process the whole 3D EPI-volumes of the 4D light fields instead of (2D) EPIs. In addition, the time-consuming global optimization is removed in this network since the quality of depth map inferred by the network is good enough.

Luo et al. [66] presented a similar shallow network with Heber et al. [64]. One of the main difference is that two EPI patches (the horizontal and vertical EPIs) are independently fed to the network. With respect to the network architecture, several more Convolutional layers than that in [64] are employed. Though the more layers are used, the depth map output from the network is still not acceptable, a different global optimization (i.e. graph cuts) is utilized to refine the depth map for the high quality.

Feng *et al.* [67] put forward a similar approach to the work by Luo et al. [66]. One of the differences is that a shallower network is considered and the output pixels after the fully-connected layer are more than one pixel. Another difference is to use the variational technique for optimization to remove the noise resulted from the network, in which the modified Alternating Direction Method of Multipliers (ADMM) proposed by Liu et al. [74] is used to obtain the optimal solution.

Shin *et al.* [68] presented a four-stream network, called **EPINET**, which takes as inputs the horizontal, vertical, left diagonal and right diagonal image views, instead of EPI patches. The EPINET [68] was proposed as a DispNet-like network, i.e. convolutions are only calculated at the full-scale of light field images. Each of four streams of EPI-volumes are fed to the three convolutional blocks respectively, and then a concatenation of the feature maps from the four streams are given to the last eight blocks to regress the final depth map. With the use of various data augmentations, the EPINET achieves top-performing performance on the public 4D light field benchmark (for the narrow-baseline scenario).

Zhou et al. [69] proposed a two-pathway CNN to learn the semantic features and the low-level structure information from the representation *focal stack* (being composed of a sequence of refocused images) and the central view respectively. One pathway, called depth-semantics pathway, is comprised of a dozen of 3D convolution layers, which takes as input the focal stack patches. Specifically, the layers include six layer-wise convolutional layers and six point-wise convolutional layers. Another pathway, called structure-information pathway, is comprised of several identical Inception blocks (includes 2D convolutional layers). Then, the depth map will be classified by the last two fully-connected (FC) layers. Finally, the parabolic interpolation and a median filter are applied to remove unreliable pixels.

These above-mentioned convolutional neural works mainly focus on depth estimation from 4D narrow-baseline light fields, however, it might fail in 4D wide-baseline light fields without changing the network architecture, which might be ascribed to the EPI-line being discontinuous, etc. To deal with this limitation, Leistner et al. [70] proposed to shift the EPI from wide-baseline image views into the narrow-baseline, while Shi et al. proposed a stereo-based deep network without taking into account EPIs.

To be specific, Leistner et al. [70] presented an EPI-Shift network, where light field stacks were shifted from the wide-baseline to the narrow-baseline, and then used to predict depths by trained models from the narrow-baseline data. The EPI-Shift is designed as an end-to-end trainable network, and uses the plane-sweep method to shift light field inputs, and then input the shifted images to a U-Net architecture to obtain a classification output and a continuous regression output. The classification output corresponds to a (large) integral disparity, and the regression out is responsible for the fractional disparity within 0.5 pixel. This solution allows the network to work on both the

narrow-baseline and wide-baseline light fields.

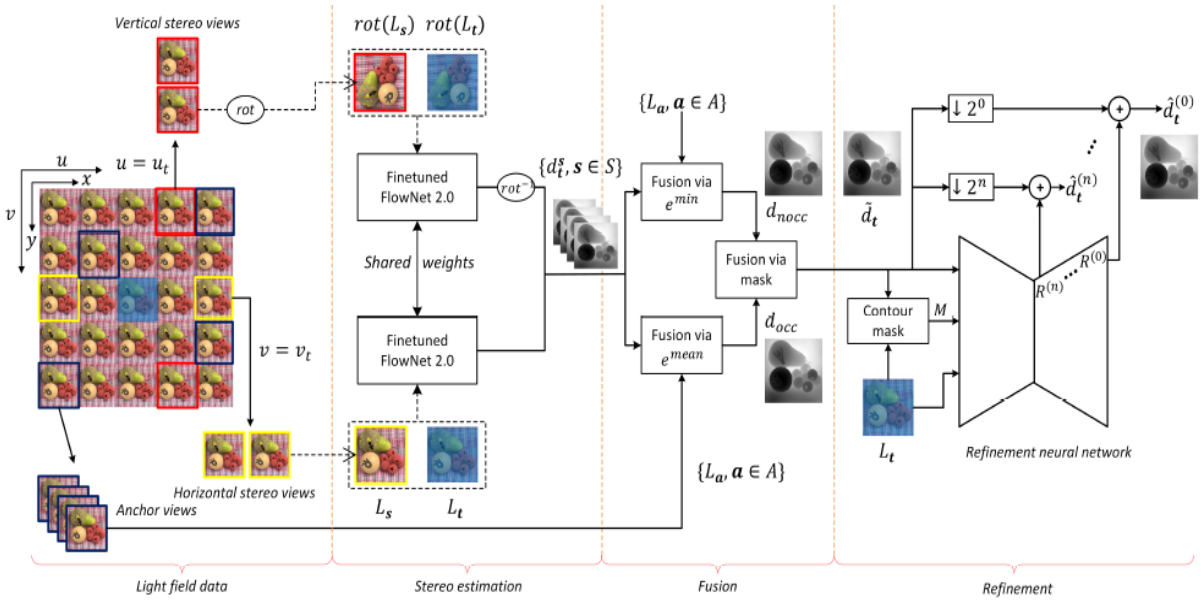


Figure 2.8: The architecture of LBDE-E, figure courtesy of [31].

Shi et al. [31] proposed a MVS-based deep network (dubbed as **LBDE-E**) for light field depth estimation, consisting of the three steps (subnetworks), as is shown in Fig. 2.8. This work starts at fine-tuning the FlowNet 2.0 [37] model onto narrow-baseline and wide-baseline image pairs and then infer four image-pairs using the fine-tuned model to obtain initial depth maps. The next step is to fuse the candidate depth maps into a single depth map by minimizing backward warping errors and handling occlusions. Finally, the fused map is refined by a modified multi-scale residual learning network (RefineNet) [75] to get the final depth map.

Though these two methods can work onto both the narrow-baseline and wide-baseline light fields, the two models are heavyweight and not suitable for practical applications. Besides, the LBDE-E [31] is not trained end-to-end so that the sub-optimal results might be obtained. Thus it is necessary to develop a new network for better performance.

PART I

Traditional Algorithms

Table 2.9: Reference of the terms of Part I.

Term	Definition
Photo-consistency	The same 3D point is seen as the same color by any cameras
Bivariate Kernel Density Estimation	Estimate probability density function with two variables
Epanechnikov Kernel	A parabolic kernel for estimating density functions of variables
Bilateral Filter	A filter for blurring images but persevering edges
Guided Filter	A more efficient and effective extension from Bilateral Filter
Weighted Guided Filter	An extension from Guided Filter
Speeded Up Robust Features (SURF)	A descriptor for a robust description of image features
Superpixel	A cluster of similar pixels
Cost Volume	A 3D volume that stores costs of all hypothetical depths
Matting Laplacian Matrix	An affinity matrix for associating similar colors with similar depths
Markov Random Field (MRF)	A graphical model of a joint probability distribution
Graph Cuts	Global optimization by estimating maximum a posteriori of MRF

CHAPTER 3

DEPTH ESTIMATION FROM WIDE-BASELINE 3D LIGHT FIELDS

There exists a few traditional methods regarding depth estimation from the (structured) 3D light fields. Due to the record of more data in the 3D light fields, the previous studies have achieved the improvements in depth accuracy, in which a large number of views with a narrow-baseline are employed. Some previous works also attempt to make evaluations on a small number of views with the same baseline, but the results witness no small degradations on aspect of accuracy. Moreover, the depth performance is still not high when the baseline is changed to be wider.

In this chapter, the sparse sampled (wide-baseline) 3D light fields are taken into consideration. Toward the goal of high depth accuracy in this scenario, a robust depth estimation method for sparse 3D light fields is proposed, dubbed *R3DE*. Bivariate Kernel Density Estimation functions are built to tackle the noise and radiance changes. The image is decomposed into edge regions and non-edge regions to deal with the occlusion issue. Additionally, a Weighted Guided Filter that is insensitive to the noise and textureless regions is applied to filter the cost volume of each image region. Finally, a confidence measure detects unreliable pixels with false disparities, to which a disparity refinement is applied.

3.1 Introduction

The (structured) 3D light fields has been actively used for depth estimation. When compared with the two-view stereo methods, the more viewpoints are available in light fields for potentially enhancing the depth accuracy. When compared with the multi-view stereo methods, the characteristic of the epipolar lines being parallel in light field views relaxes the correspondence search in the horizontal or vertical image scan-line with unknown camera poses. In general, the existed depth estimation methods from the 3D light fields have improved the depth accuracy so far, where the 3D light fields are comprised of a large number of sampled views with the narrow-baseline. Stacking the sampled views on top of each other generates a 3D imaging volume, and then the so-called Epipolar Plane Images (EPI) [76] can be constructed by vertically slicing

this 3D volume. The related methods take advantages of one property of the 3D light fields, i.e., the slope of the (continuous) EPI-line is proportional to the disparity, to help generate the high accuracy depth maps.

However, an EPI will be constructed as the discontinuous strips rather than the continuous EPI-lines when the 3D light fields are captured with the wide-baseline camera setup. Moreover, with the decrease of the sampled views, the robustness might be impaired. kim et al. [19] demonstrates its attractive performance on a large amount of views, but it fails to perform well on the fewer multi-views with the wide-baseline since it is not robust against noise and occluded regions. Lv et al. [48] is in the same boat with [19], which cannot work well either due to the obvious degradations seen in their produced depth maps. Jorissen et al. [50] enhances the robustness in this camera setup, but still fails to preserve the discontinuity well in occlusion regions with low contrast and is sensitive to severe occlusions.

To handle this issue, a robust depth estimation method for sparse light fields with the wide-baseline is proposed, which will be detailed described in Section 3.2.

3.2 Methodology

Fig. 3.1 shows an overview of the proposed method (in 3DTV-CON). The 3D light fields are taken as the input, and firstly preprocessed by an image filter. The edge map and non-edge map are separated from the image by gradient operators (Sec 3.2.2), and then the cost volumes are initially generated by the calculation of the bivariate kernel density function (Sec 3.2.1), followed by the window-based filtering (Sec 3.2.3). The winner-take-all strategy is applied to the filtered cost volumes to obtain the edge disparity map and the non-edge disparity map. Next, the fused disparity map and unreliable disparity map are input to an energy function in order to assign the correct disparity to these unreliable pixels by the global optimization (Sec 3.2.4-Sec 3.2.6).

For preprocessing, the bilateral filter [77] is employed to remove the noise in the light field images prior to the depth computation for pixels.

3.2.1 Bivariate Kernel Density Estimation

To obtain an accurate initial depth, the cost function BKDE using the radiance and the relative gradient is proposed. Though it is not computationally complex, it is nevertheless effective - compared to radiance-only approaches - w.r.t. more disturbing issues in detailed regions of the scene, as exemplified in Fig. 3.2.

The Relative Gradient (RG) deals with the radiance change problem and is typically used as a cost function [78] in stereo matching. The relative gradient for each pixel is

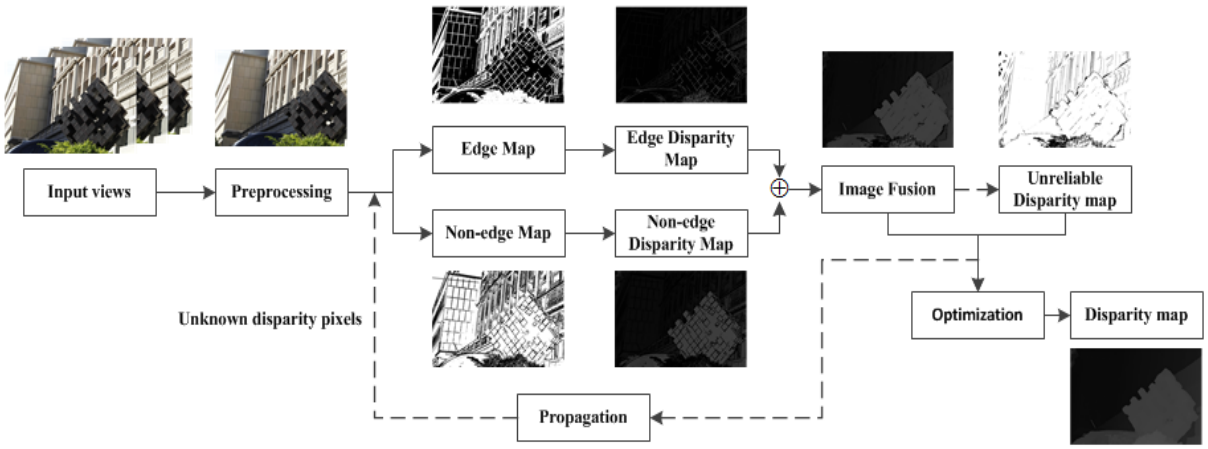


Figure 3.1: The proposed framework of depth estimation on the 3D light fields.

calculated as follows:

$$RG = \frac{Grad}{Grad_{max} + \delta} \tag{3.1}$$

where RG is the relative gradient, $Grad$ is the gradient of the image that is computed by the gradient operator (such as the sobel operator, difference operator etc.), and the $Grad_{max}$ is the maximum gradient of a pixel in its centered 3×3 window. To avoid the denominator being zero, a very small value δ is added.

Here the RG is not used to compute the cost function directly, but is combined with the radiance to build the BKDE function. The reason to build the BKDE is that we observe the RG is more vulnerable to the noise than the radiance, while it is more robust to the radiance changes. The specific BKDE function is represented as follows:

$$C_p(\hat{d}) = \frac{1}{|\Omega|} \sum_{s \in \Omega} K_{h_1} \left(R_{p,s^*} - R_{p,s}(\hat{d}) \right) * K_{h_2} \left(RG_{p,s^*} - RG_{p,s}(\hat{d}) \right) \tag{3.2}$$

where $C_p(\hat{d})$ is the cost of the pixel p at the candidate disparity $\hat{d}$ where the maximum value corresponds to the true disparity in the cost volume, and Ω represents the number of valid views for cost computations. For the cost volume, it indicates a 3D array $(u, v, \hat{d})$ that stores the costs/probabilities of candidate disparities $\hat{d}$ for all pixels of the reference view. $R_{p,s}$ and $RG_{p,s}$ denote the radiance and RG value of the pixel p in the target views respectively, while R_{p,s^*} and RG_{p,s^*} denote the radiance and RG value of the pixel p in the reference view respectively. K_h corresponds to the Epanechnikov kernel that is given below in Eq. 3.3. h_1 and h_2 are band width parameters for the radiance (h_1=0.05, h_2=0.1) and RG values respectively, which control the accuracy of density estimation.

$$K_h(x) = \begin{cases} 1 - \left\| \frac{x}{h} \right\|^2 & \left\| \frac{x}{h} \right\| \leq 1 \\ 0 & otherwise \end{cases} \tag{3.3}$$

Figure 3.2: An example of comparison between Radiance-only (Left) and BKDE (Right) using the Statue scene from the 3D light field dataset.

3.2.2 Edge Map and Non-edge Map

We first extract a thick edge map composed of both edges (containing contours and their similar regions) and others, using a 2D cross window with horizontal and vertical arms, yielding fattening edges. These are computed through the L1-norm, following Eq. 3.4,

$$F_e = \sum_{n=-N}^{n=N} |R_{s^*} - R_n| \geq \alpha_e \tag{3.4}$$

where F_e represents the fat edge, composed of the horizontal edge and vertical edge. N denotes the length of both the horizontal and vertical arm (N=4). R_n represents the neighboring pixels of the current pixel R_{s^*}. α_e is a threshold value (α_e=0.12).

Since the fat edge involves the non-edge regions (the regions excluding the edges), the extraction of the contour map (containing only the thin edges relative to the thick edges) is performed to help obtaining the target edge map. We observe that the RG operator extracts the edges well when regarded as the gradient operator. Hence it is chosen here as it has been computed beforehand. Likewise, the pixel with a high relative gradient, i.e., $M_e = RG \geq \beta_e$ (β_e is set to 0.7), is assigned a mask, and then combined with $\{|R_{s^*} - R_n| \geq \gamma_e | n = 1, 2, ...N\}$ to remove the non-edge pixels using a 2D cross window (N=10, γ_e=0.3).

After the extraction of edge maps in the image, the remaining regions are automatically non-edge maps.

3.2.3 Cost Volume Filtering

The costs from the BKDE usually fail to have the unique maximum, especially in the low texture regions. Moreover, the computed costs sometimes give the wrong maximum in the noisy or occluded regions. To deal with this problem, we therefore impose the constraint of nearby pixels to have similar disparities. To achieve this goal, the weighted guided image filter [79], an O(N) edge-aware preserving filter with smoothing properties, is chosen as a local filter. It is an extension of an edge-preserving filter, i.e. the guided image filter [80], which showed higher performance for stereo cameras [81].

The filtering is computed as follows:

$$\widetilde{C}_p(\hat{d}) = \sum_q W_{pq} C_q(\hat{d}) \tag{3.5}$$

where $\widetilde{C}_p(\hat{d})$ is the filtered cost of $C_p(\hat{d})$ and $C_q(\hat{d})$ is the cost of the neighboring pixel q in a window. With respect to the weight of the filter, it is given as below,

$$W_{pq} = \frac{1}{|\omega|^2} \sum_{k:(p,q)\in\omega_k} \left\{ 1 + \frac{(I_p - \mu_k)(I_q - \mu_k)}{{\sigma_k}^2 + \frac{\varepsilon}{\chi(p)}} \right\} \tag{3.6}$$

$$\chi(p) = \frac{1}{N} \sum_{k=1}^{N} \frac{{\sigma_p}^2 + \eta}{{\sigma_k}^2 + \eta} \tag{3.7}$$

where $W_{p,q}$ with its control parameter ϵ (ε=0.01) gives a higher weight to the pixel on the same side of the edge and a lower weight to the pixel on two sides of the edge. As the edge and non-edge regions are processed individually, the $|\omega|$ of their window ω is set to 25^2 and 51^2 respectively in our experiments. I is the guided image, μ and σ are the mean and variance of the window in I respectively. $\chi(i)$ with its control parameter η is used for mitigating the ambiguities (η= 0.05), penalizing the cost of the pixel that lies around the edge, especially in occluded regions, as shown in Fig. 3.3.

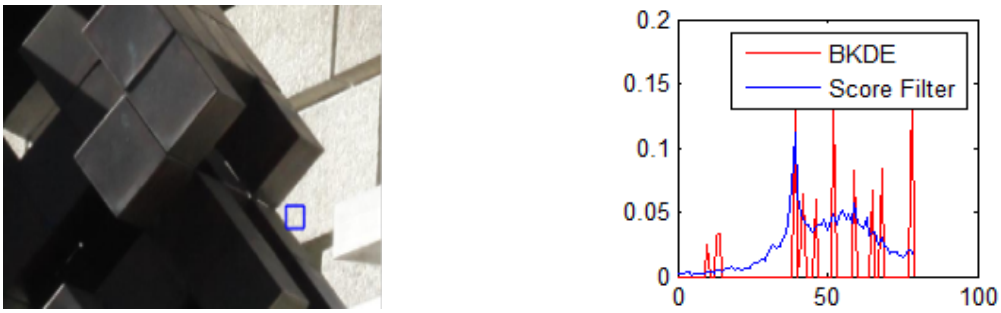


Figure 3.3: An example of the distribution of the cost values for a pixel in occluded regions using the Statue scene from the 3D light field dataset. Note that the cost is obtained after cost volume filtering.

3.2.4 Confidence Computation and Depth Fusion

Though previous steps provide most often correct depth, few outliers with wrong depth remain, especially in occluded regions, as a consequence of the foreground/background ambiguity. We compute the confidence of the pixels having the true depth through the neighboring Gaussian kernel function, where the pixels with the low confidence will be regarded as the outliers. With respect to the neighboring Gaussian kernel function, it is formulated by checking the similarity of corresponding pixels at the estimated depth, which is given as follows:

$$M_u = \frac{1}{|\omega_{s^*}|} \sum_{s^* \in \omega_{s^*}} \left\{ \frac{1}{|\Omega|} \sum_{s \in \Omega} exp \left\{ -\frac{\Delta}{\varphi^2} \right\} \right\} \leq \tau \tag{3.8}$$

where M_u denotes the mask of the low confidence pixel ($M_u = 0$ if low confidence, otherwise 1), $|\omega_{s^*}|$ is the length of a 3x3 window ω_{s^*} of the view, $|\Omega|$ is the length of valid pixels in a radiance set Ω, $\Delta = |R_{s^*} - R_s|$ (Δ does not exceed 0.15), φ and τ is a parameter and a threshold value respectively ($\varphi = 1.0/255$ and τ=0.04).

When the outlier/unreliable map is obtained, the edge depth map is merged with the non-edge depth map, and then remove outliers to get the initial depth map.

3.2.5 Propagation

The depths of the high confidence pixels can be propagated to the corresponding pixels in the other views. Considering that a few pixels are not visible in all views, the propagation is constrained by a simple metric, namely $|R_{s^*} - R_s| \leq \tau$ that belongs to the part of Eq. 3.8. Note that we only propagate the depths of the pixels to the unprocessed views, as the views closer to the center view tend to have more high confidence pixels. The pixels with low confidence will be accurately coped within the optimization step, explained below.

3.2.6 Optimization

Now, the fused depth map for the reference view still contains the (unreliable) pixels without being assigned the correct depth. Assuming that the pixels with similar radiance will have similar depth in a small neighborhood (referred to as local constraints), we utilize the neighboring pixels having the high confidence depth to deduce the depth for the remaining unreliable depth pixels. A least square energy function that consists of a smoothness and a data term term is thus built,

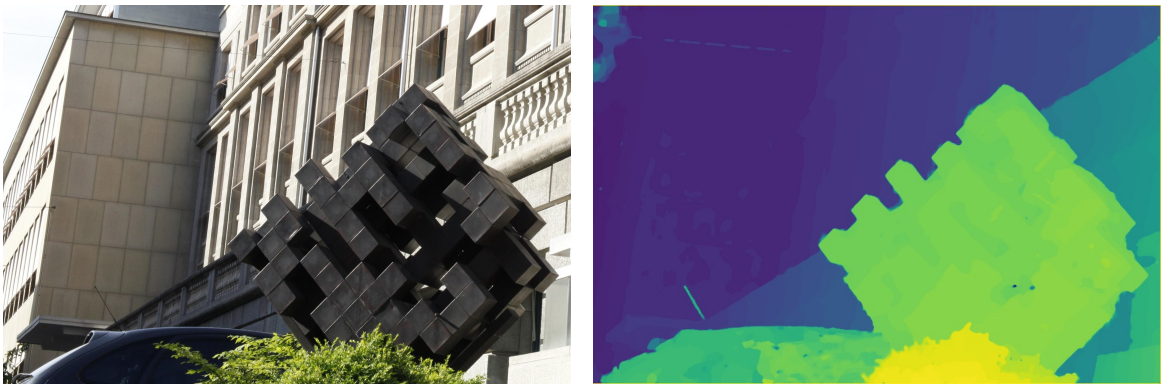
$$E(\hat{d}) = \hat{d}^T L \hat{d} + \lambda (\hat{d} - \hat{d}_{init})^T M (\hat{d} - \hat{d}_{init}) \tag{3.9}$$

where $\hat{d}_{init}$ is the initial depth map, $\hat{d}$ is the desired depth map, and λ is a large number (λ=100). The first term is denoted as the smoothness term, where L is formed as the matting Laplacian matrix proposed by [82], satisfying the local smooth constraints. The second term, i.e., the data term, encodes the relation between the initial depth map and the desired depth map via the unreliable depth map M. The minimization of the energy function is solved by making the first derivative of Eq. 3.9 zero, which gives a closed-form solution. Computing this solution directly is a time- and memory-consuming task, therefore we resort to an alternative way, i.e. adopting the O(N) large kernel-based approach [83], to solve this large and sparse linear system. The specific form of L can

be found in [82–84].

3.3 Exemplar Results

Figure 3.4 depicts the exemplar result of our method, which is tested on the sparse 3D light fields with the wide-baseline. Though this outdoor scene contains challenging contents, e.g. occlusions, our method is still able to reconstruct a high quality depth map. For the more (comparative) results, we refer the reader to the Chapter 7 for details.



Central view *R3DE*

Figure 3.4: An visual example of the proposed depth estimation result.

CHAPTER 4

DEPTH ESTIMATION FROM 4D LIGHT FIELDS

The 3D light fields, used in Chapter 3, are typically captured along the horizontal or vertical path, which can be regarded as being linearly sampled from the 4D light fields. In contrast with the 4D light fields, the 3D light fields (e.g. composed of a horizontal line of views) seem a bit weaker since the 3D real-world point is occluded in their views but might be un-occluded in other angular views of the 4D light fields. In this chapter, we seek an extension of the proposed method in Chapter 3 to use the more angular views in the 4D light fields against the occlusion issue when estimating the depth. With respect to the state-of-the-art methods for the 4D light fields, they have improved the depth accuracy with the dense sampled light field views (e.g. 9x9 light fields), but this is usually not the case with the sparse sampled light fields views. Thus, a scalable 4D light field depth estimation framework, called *S-R4DE*, is proposed to work well on both the dense and sparse 4D light field images. The proposed framework is mainly realized by leveraging multiple edge cues to occlusion detection and then integrate it with local costs into an energy function.

4.1 Introduction

Depth estimation from the 4D light fields has been studied for a long time, and has achieved a significantly high accuracy when the baseline (or disparity range) between the adjacent images is narrow or the light field is densely sampled (e.g. 9x9 light field images within a range of 4 disparity pixels). However, we observe that the accuracy still remains an issue in the state-of-the-art traditional methods, when the baseline is wider or the light field is sparser (e.g. 3x3 light field image views within a range of 16 disparity pixels). Actually, depth estimation with the aid of the angular information but with as few angular views as possible is more attractive in the practical applications.

To cope with this issue, a scalable framework for light field depth estimation is proposed in this chapter. More specifically, the kernel density estimation and the size-adaptive window filter are introduced to locally estimate disparities in which an adaptive size is considered. Since there are more ambiguities at occlusion areas, an occlusion

handling method, i.e., occlusion detection and cost-volume recomputation, are proposed, followed by using an occlusion-aware optimization to improve depth-continuity and enforce global consistency.

4.2 Methodology

Fig. 4.1 shows an overview of our method (in WSCG). Taking a central view of the 4D light fields for instance, the local disparity map (LDM) is initially produced from a winner-take-all strategy onto cost volume computations (Sec 4.2.1). Then a disparity edge map (DEM), canny edge map (CEM), superpixel edge map (SEM), occluded pixels map (OPM) are put into the occlusion handling site to extract an occlusion boundary map (OBM) (Sec 4.2.2). With the aid of these occlusion detection results, the final disparity map (FDM) is better generated under optimizations when compared with the LDM (Sec 4.2.3).

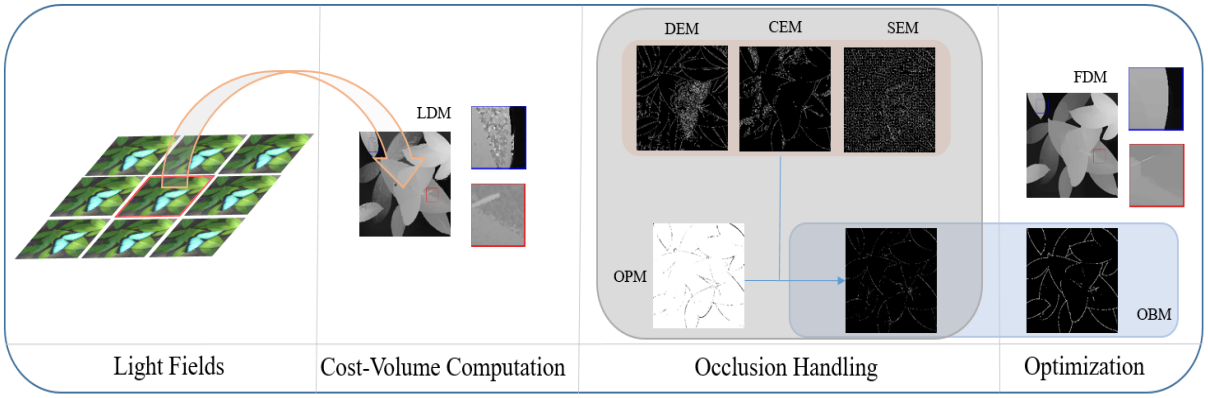


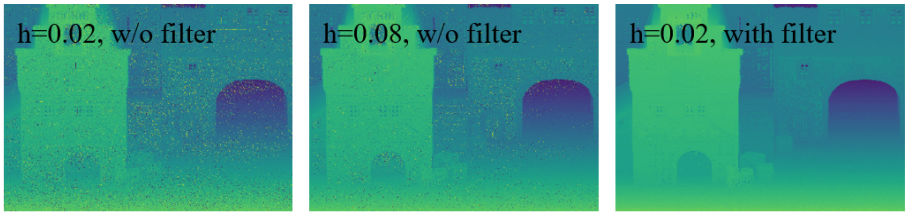
Figure 4.1: The proposed framework of depth estimation on the 4D light fields.

4.2.1 Cost Volume Computation

The proposed cost volume computation is composed of two steps: 1) initially computing the cost volume, 2) filtering the cost volume.
A kernel density estimation function is employed to the initial cost volume calculations, which is formulated as follows:

$$C_p(\hat{d}) = \frac{1}{|\Omega|} \sum_{s,t \in \Omega} K_h \left(R_{p,s^*,t^*} - R_{p,s,t}(\hat{d}) \right) \tag{4.1}$$

where $C_p(\hat{d})$ is the cost of the pixel p at the candidate disparity $\hat{d}$, and Ω represents a number of valid views for cost computations. $K_h(\cdot)$ corresponds to the Epanechnikov



MSE 0.175 MSE 0.084 MSE 0.013

Figure 4.2: Compared with the increase of h, the edge-preserving filter demonstrates its higher ability (a lower MSE) to remove the noises without losing fine details on the Medieval scene from the 4D light field dataset.

kernel that is given in Eq. 3.3 and h is its bandwidth parameter (= 0.02), which controls the accuracy of the density estimation. Actually, a higher value of h increases the accuracy and robustness to noise. However, it will lose fine details.

As a complement, the guided filter [80], rather than the weighted guided filter, is introduced to filter out some noises by Eq. 3.5, which is simpler but can work well here. With respect to the weight of this filter, it is given as below,

$$W_{pq} = \frac{1}{|\omega|^2} \sum_{k:(p,q)\in\omega_k} \left\{ 1 + \frac{(I_p - \mu_k)(I_q - \mu_k)}{{\sigma_k}^2 + \varepsilon} \right\} \tag{4.2}$$

where $W_{p,q}$ gives a higher weight to the pixel on the same side of the edge and a lower weight to the pixel on opposite sides of the edge in a window ω_k centered at the pixel k. The radius of this window ω_k is adaptive with the spatial resolution (w, h) of the light field, i.e., $max(\lfloor max(w,h)^2/(256 * min(w,h)) \rfloor, 3)$. I is a guided image, namely the light field view R_{s^*,t^*} that is being estimated; μ_k and σ_k are the mean and variance of the window ω_k in I respectively; ϵ is set to 0.01. The more effectiveness of this technique than the only increase of the h is shown in Fig. 4.2, clearly reducing the speckle noise.

4.2.2 Occlusion Handling

Assuming that the scene in light fields is lambertian, the scene point that is seen from different viewpoints shares the same color, exhibiting the photo-consistency. However, this is not true for the point that is occluded. Some pixels from such a point in the cost-volume computation step might be correctly estimated due to the edge-aware cost volume computation. Nevertheless, the disparities of pixels at heavy occlusion regions still remain difficult to be well-estimated due to ambiguities. As a result, a pixel with a wrong disparity may be assigned a highest cost. To handle this issue, the occluded pixel detection (OPD), occlusion boundary detection (OBD) and cost-volume recomputation (CVR) are proposed.

Occluded Pixel Detection

Some pixels disappear in parts of the views due to occlusions, breaking off the photo-consistency. Assuming that the scene is lambertian, a simple thresholding technique could be applied to detect these occluded pixels, as given in Eq. 6.

$$C_p(\hat{d}) = \frac{1}{|\Omega|}\sum_{\Omega}(1 - exp(-|R_{p,s^*,t^*} - R_{p,s,t}(\hat{d})|)) \tag{4.3}$$

where $C_p(\hat{d})$ indicates the occlusion confidence of a pixel p at the estimated disparity $\hat{d}$. If the confidence of a pixel is larger than a specified threshold τ (= 0.05), it is masked as an occluded pixel $(OP = 1)$; otherwise it is un-occluded $(OP = 0)$.

Occlusion Boundary Detection

Occlusion boundary detection is a significant step for the occlusion handling as its accuracy makes differences for the following disparity re-estimation and occlusion-aware optimization. To guarantee its precision, multiple edge cues are leveraged to detect occlusion boundaries.

Firstly, a fact to be known is that there always exist edges between an occluder and an occluded region, which is ascribed to lighting changes in-between. Thus the following lemma is given.

Lemma I. An occlusion boundary set OB_s is a proper subset of an edge set EG_s.

The edge set is approximately constructed in our work for efficiency, i.e., a union of edge points and edge lines,

$$EG_s \simeq EG_{point} \cup EG_{region} \tag{4.4}$$

where EG_{point} denotes the edge points that are acquired by an edge detector, and EG_{region} indicates the edges from a region/superpixel detector [85]. Note that the region size is set to a smaller value so as to be not much larger than the objects in the scene. Additionally, a small region used in a superpixel detector could boost the edge accuracy.

The occlusion boundaries that belong to the occlusion boundary set OB_s are taken from the approximated edge set. Firstly, a disparity edge map DEM is computed from a relatively reliable local disparity map LDM using a canny edge detector [86], and an edge map EM is intersected by the canny edge map CEM and the superpixel edge map SEM. Then we calculate an intersection of DEM and EM to get an initial occlusion boundary map Occ_b^i. Furthermore, the disparity variance in a window and the difference operator are computed as masks to update the difference between Occ_b^i and themselves in order to remove edge point outliers,

$$EM^u = M_{disp} * (EM - Occ_b^i)$$
$$DEM^u = M_{\nabla} * (DEM - Occ_b^i) \tag{4.5}$$

where M_{disp} and M_{∇} denote the disparity mask and the difference mask respectively. If the pixel has a disparity variance beyond a threshold φ that is adaptive to the disparity range, M_{disp} is assigned 1, otherwise 0. Similarly, if the pixel has a difference beyond a specified threshold $\nabla(= 0.05)$, M_{∇} is assigned as 1, otherwise 0.

Finally, a union of multiple maps are used to produce the occlusion boundary map $OBM = Occ_b^i \cup DEM^u \cup EM^u$.

Cost Volume Recomputation

The cost volume recomputation consists of two steps: 1) computing the disparity bound, 2) cost-volume computation, targeting the improvement of the occluded pixel disparity estimation.

Disparity Bound The new upper bound ub and the lower bound lb in disparity are determined by the disparities of pixels in their neighborhood beforehand. The upper and lower bound are assigned to the maximum and minimum disparity of neighboring pixels respectively.

Cost-Volume Computation The procedure in the previous cost-volume computation is reused here, but there exists two differences. The first difference is that a disparity bound, i.e, a half-closed interval $[lb, ub)$, is utilized for computing the occluded pixel cost $OccS_p(\hat{d})$ for the pixel p at a candidate disparity $\hat{d}$. The second difference is that the visible views Ω_{vis} for photo-consistency are selected. More specifically, the relative location of the occluded pixel to the occlusion boundary from OBM (with rare negative occlusion boundaries) is used to simply select the visible views.
At the end of the occlusion handling flow, the occlusion boundary map with a high accuracy can be extracted and the cost of the occluded pixel will be lessen, which are beneficial to the following optimization step.

4.2.3 Optimization

Our disparity estimation is optimized by minimizing a Markov Random Field-based energy function, as given in Eq. 4.6.

$$E = \lambda * \sum_p E_{data}(p, d(p)) + \sum_{q \in N_p} E_{smooth}(p, q, d(p)) \tag{4.6}$$

where N_p is a 4-neighborhood of the pixel p, q represents one of the neighboring pixels and $d(p)$ denotes a disparity that is mapping to an integer. Herein λ (= 10) is introduced to balance the ratio of the data term and the smoothness term.
The data term in the energy function is built by weighting the cost $\widetilde{S}$ and the occlusion cost $OccS$,

$$E_{data}(p, d(p)) = \kappa - \alpha * \widetilde{C}_p(\hat{d}) - (1 - \alpha) * OccS_p(\hat{d}) \tag{4.7}$$

where E_{data} measures the photo-consistency for the pixel p, α is a weighting coefficient (= 0.6) and κ is a large constant $(= 10)$.
The smoothness term is computed by a weighted neighboring function,

$$E_{smooth}(p, q, d(p)) = w_{p,q} * min(|d(p) - d(q)|, \Gamma) \tag{4.8}$$

$$w_{p,q} = exp(-\frac{||R_{s^*,t^*,p} - R_{s^*,t^*,q}||^2}{\psi^2} - \frac{|OB_p - OB_q|}{\phi^2} - \frac{|OP_p - OP_q|}{\phi^2}) \tag{4.9}$$

where Γ represents a truncated threshold that is set to 10; ψ and ϕ is set to 1/9 and 1 respectively; OB is an occlusion boundary mask from the occlusion boundary map OBM and OP is an occluded pixel mask from the occluded pixel map OPM that are enforced as constraints. If an occlusion boundary exists in-between two pixels or one of two neighbouring pixels is an occluded pixel, the strength of smoothness will be reduced. Besides, the color in R_{s^*,t^*}, is encoded as a constraint in which two pixels with different colors will decrease smoothness. To solve the proposed occlusion-aware energy function, the graph cuts algorithm [44] is used, which performs the max-flow/minimum cut onto the edges in a graph (comprised of the source and sink terminals, the image pixels) to find the minimum edge costs among all cuts (cf. 8.2 for more details), and is implemented in gco-v3.0 [1] here. As a consequence, the proposed occlusion metrics especially help a lot to avoid over-smoothing, hence preserving sharp edges, see Fig. 4.3.

4.3 Exemplar Results

In this section, the exemplar results of the proposed *S-R4DE* are presented, which are evaluated on the 4D light field dataset with the narrow-baseline and wide-baseline respectively. Our method is capable of recovering correct depths at a huge amount of image regions, as is shown in Figure 4.4. Besides, the reconstructed depths look very

1. https://vision.cs.uwaterloo.ca/code/

w/o occ with occ Ground truth

Figure 4.3: Comparisons between without (w/o) and with occlusion detection results (occ) in the energy function. It demonstrates that the proposed occlusion-aware energy function contributes to a higher accuracy (a lower MSE 0.010) without over-smoothing the sharp edges on the Medieval scene from the 4D light field dataset.

close to the ground truth, and indeed are well recovered at occlusion regions. We will make further evaluations and comparative experiments, and arrange the corresponding text in the later chapter (also the Chapter 7) for unification.

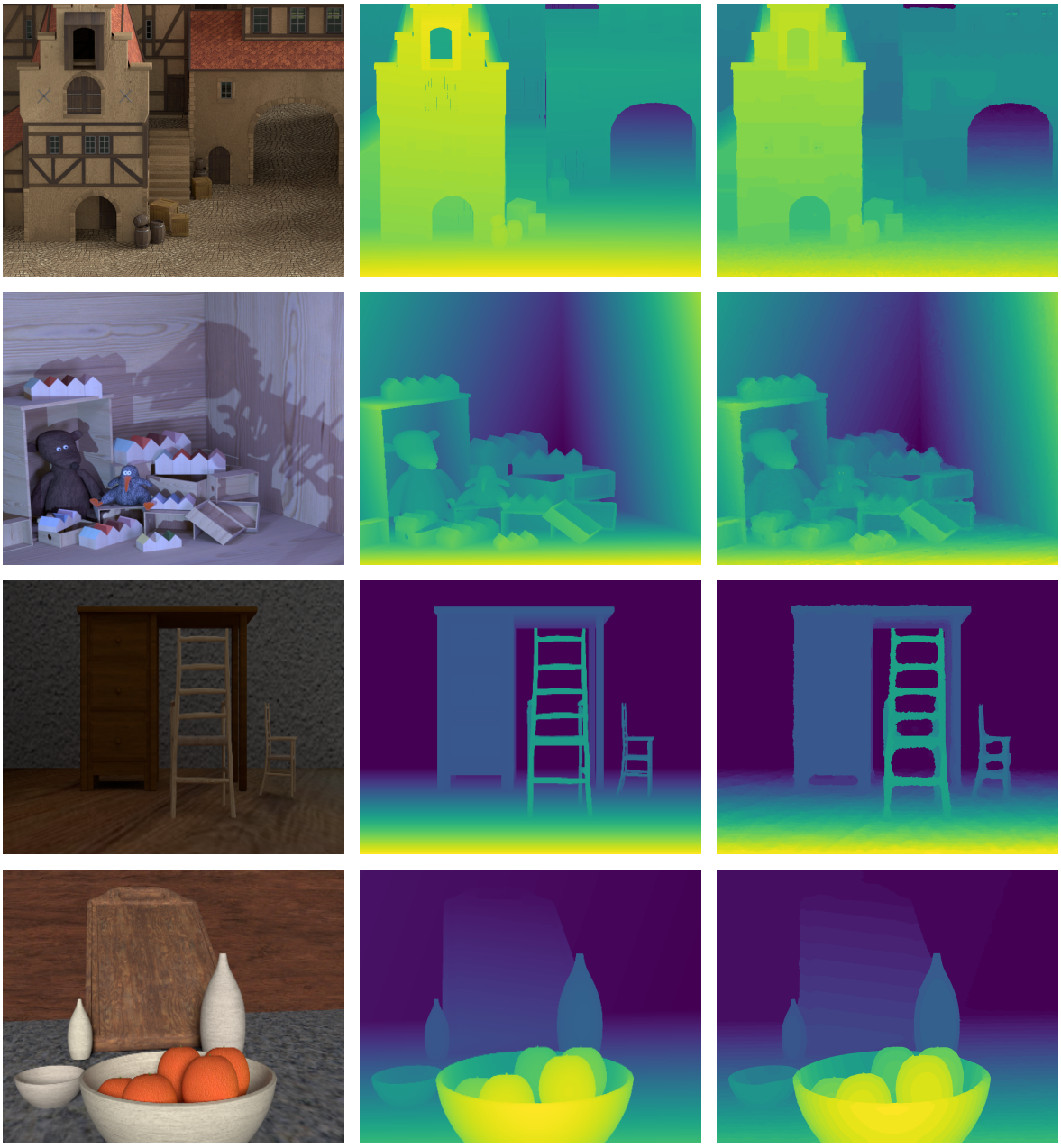


Figure 4.4: An example of visual depth estimation results from the proposed method on different-baselines light fields. The scenes in the top two rows belong to the narrow-baseline, and the scenes in the bottom two rows belong to the wide-baseline.

PART II

CNN-based Algorithms

Table 4.1: Reference of the terms of Part II.

Term	Definition
Batch Normalization	A technique that improves the convergence of neural networks
ReLU	Rectified Linear Unit, i.e. a type of activation function
Residual Learning	Using skip connections to learn the residual in case of gradient exploding
Encoder-decoder network	Encoder maps inputs to latent spaces that are mapped to outputs by decoder
Stacked Hourglass Network	A network comprised of the repeat of pooling and upsampling layers
Adam Optimizer	An optimization algorithm that updates the weights to minimize the loss
Data Augmentation	A technique to increase the quantity and diversity of training data
Ablation Study	A way of removing one component of the model to see the influence

CHAPTER 5

DEPTH ESTIMATION FROM NARROW-BASELINE 4D LIGHT FIELDS

In the previous two chapters, the traditional depth estimation methods for the 3D or 4D light fields are described, in which a variety of hand-crafted features (e.g. focus/defocus, epipolar plane image, and surface camera) are typically extracted, and then feature matching is conducted, followed by a time-consuming global optimization.

In this chapter, the hand-crafted features are replaced with the more discriminative features learned by the convolutional neural networks (CNN). Specifically, two end-to-end convolution neural networks, i.e., *HFNet* and *MANet*, are sequentially presented for estimating depths from the 4D light fields. The *HFNet* is proposed as a hybrid feature network that combines the Epipolar Plane Image and epipolar properties, learning more generic feature representations. The *MANet* is a parameter-effective and efficient multi-scale aggregated network. The two networks are performed for estimating depth from the plenoptic cameras, and experimental results show that the proposed *MANet* outperforms state-of-the-art the traditional and CNN-based methods on HCI, CVIA-HCI and EPFL Lytro light field datasets, and run much faster than the traditional methods. The code and models are available at https://github.com/YanWQ/MANet.

5.1 Introduction

The proposed traditional method in the previous chapter has been shown that it is able to recover the depth well from 4D light fields, however, the powerfulness of (engineered) feature representation is still limited. In fact, the low-level features (e.g. the edge, corner, etc) used by the traditional methods and the high-level features (e.g. the semantics), which are superior to either of the two features in deducing disparities, are in demand. The more important issue is that the proposed method suffers from the computational burden since the global optimization solved by graph cuts is used, impeding the real-time possibilities. To tackle the limitations, we move on to the CNN-based methods that learn low-to-high features, support parallel computing and having achieved the success in similar vision tasks: stereo matching, optical flow, etc.

In recent years, hand-engineered features have been replaced with deep features in

some of current state-of-the-art algorithms [64, 66–68]. The learnt multi-level features in CNN exhibit the invariance to intensity changes in images, which are beneficial to the feature matching in light field depth estimation. The current state-of-the-art learning-based methods opt to learn such features based on the traditional EPI or epipolar representation. When compared with the traditional algorithms using similar characteristics [19, 51, 56], the CNN-based methods achieve higher accuracy depth maps, which are derived from the more discriminative features that the CNN learn. Specifically, some of CNN-based works [64, 66, 67] propose to learn representative features by inferring the epipolar line orientations on EPI while the other [68] propose to stack epipolar images in channel to make CNN learn features through epipolar property (cf. Fig. 1.2). The former takes advantage of the EPI property, i.e., the disparity is a function of the line orientation on EPI, for depth inference, but it seems sensitive to the texture-less regions so that the inferred disparities look noisy. Thus the computational-burden optimization is often used as a post-processing step to remove the noises. The latter takes advantage of the context or structure information in the scene [87] and draws on the effectiveness of the epipolar property for depth predictions [28], but seems vulnerable at depth-discontinuity image regions. In general, the current state-of-the-art CNN-based works have surpassed the traditional algorithms in **depth accuracy** and **computation efficiency**, but still have some issues, which might be resulted from the less discriminative features.

Recently, EPINET [68], a network trained end-to-end without post-processing, achieves state-of-the-art accuracy onto the HCI and CVIA-HCI datasets [6, 30]. However, these CNN-based approaches are modeled by heavyweight networks (e.g., [36] has around 199M parameters).

5.2 Related Work

To better understand the potential contributions, the recent developments of relevant techniques based on deep learning for depth estimation (since they surpass the traditional methods in accuracy and speed) are discussed in this section.

Stereo matching Neural networks are more exploited in stereo matching, compared to light field depth estimation. DispNet [28] adopts a deep encoder-decoder architecture (U-net [88] alike architecture) in which the left and right images are simply concatenated to extract deep features, followed by a number of 2D convolution layers to aggregate the context (2D aggregation) and regress the disparity maps. DispNetC [28] makes use of a 3D cost volume (a standard component in traditional stereo matching [6] with the size $H \times W \times C$ by correlating left and right image features, in which H, W and C represent the image height, image width and the number of channels,

respectively.
For the sake of large context aggregation, GC-Net [89] introduces a new dimension, i.e, disparity, to build a 4D cost volume $L \times H \times W \times C$, followed by a number of 3D convolutions and deconvolutions (3D aggregation). Recent works follow the idea of GC-Net [89] but make efforts to improve the accuracy by replacing some of its components. For example, PSMNet [90] replaces the simple 3D encoder-decoder network in GC-Net with a 3D stacked hourglass network; PDSNet [91] computes an expectation around the disparity with minimum matching cost in the sub-pixel MAP approximation of GC-Net; GwcNet [92] changes the concatenation-based cost volume into a group-wise cost volume to make full use of both correlation and concatenation. Note that the output resolution of these ConvNets is a half (GC-Net) or quarter (the others) of the input resolution (due to the GPU memory limitation), and the output are finally bilinearly upscaled back to the original resolution.

5.3 Review of EPINET

In this section, we will review the EPINET [68] that achieved top-performing performance for the narrow-baseline scenario, in order to tell why the new or proposed CNNs are still needed. Firstly, we will describe the model architecture of EPINET, as is shown in Fig. 5.1. The EPINET is designed as a four-stream network, taking as inputs the horizontal, vertical, left diagonal and right diagonal image views that are similar to inputs in the traditional work [57]. The light field image features are always calculated by convolutional blocks on a single-scale and the depth map is also predicted on a single-scale. Actually, one advantage of the CNN is the ability of learning the richer feature representation. From this point of view, the multi-scale feature representations are semantically strong at all scale levels, enhancing the potentials in fitting the relation between the inputs and outputs well. Besides, it might be possible to combine the EPI-volumes used by EPINET with the EPI patches to learn the more discriminative features. Another motivation is that we found from the experiments that the high performance is relevant to the padding being not used in the convolutional layers. However, the 22 side-length pixels are sacrificed in the final depth map, which might be not acceptable in some applications.

5.4 Methodology - I

In this section, the details of the proposed *HFNet* for light field disparity estimation will be presented.The proposed *HFNet* mainly consists of multiple subnetworks: EPI patch subnetwork (EPSNet), context-aware subnetwork (CASNet) and fusion subnet-

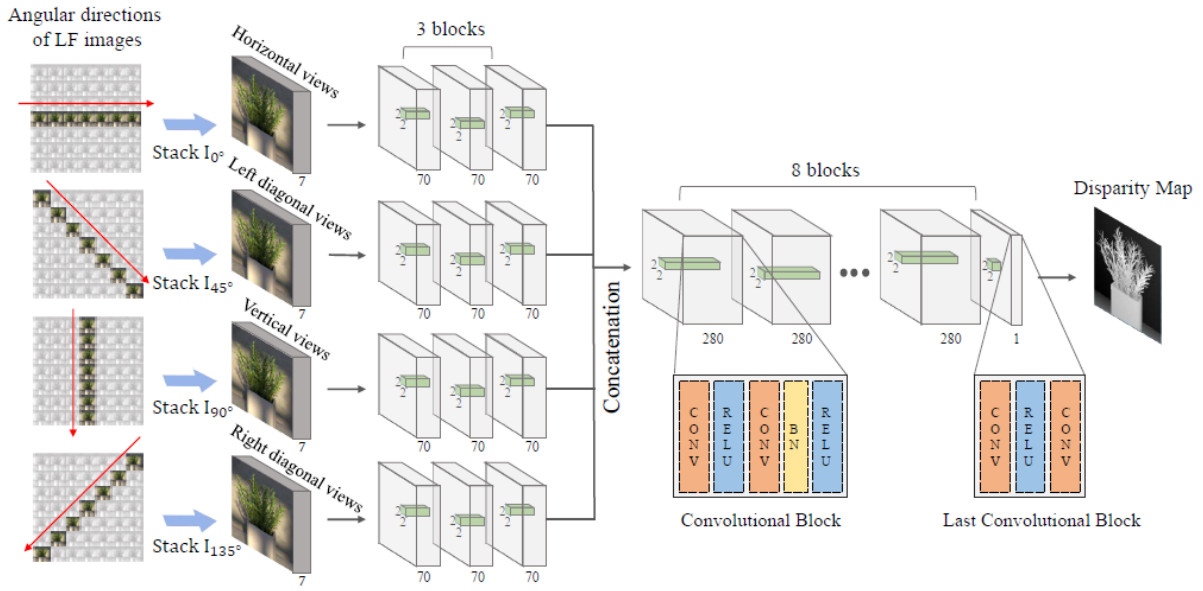


Figure 5.1: The architecture of EPINET, figure courtesy of [68].

work (FSNet). The first two subnetworks are responsible for collecting the EPI and context information of the light field respectively, and the last aims to collect the generic information that are more representative than [66–68]. The proposed network is trained end-to-end on a public light field dataset without the need of any pre-training. The whole architecture of the proposed *HFNet* is illustrated in Fig. 5.2.

5.4.1 EPI Patch Subnetwork

The EPSNet is designed as a pixel regression network to learn the relationship between the slope of the epipolar line at EPI patch and the ground truth (GT) disparity. A straightforward choice for the purpose is to build a standard CNN architecture, formulating this problem as a task of classification, as is done in [66, 67]. Another choice is to replace the fully-connected layer with the convolutional layers to get a fully convolutional network and make regressions. We have attempted the two choices in our case but do not observe much difference. Considering the number of parameters, the latter is exploited here. Since the existing light field datasets are sparsely sampled, the angular resolution of the light field (i.e., the number of sub-aperture views) is not large enough such that one dimensional size of EPI patch is limited. To address this issue, a shallow fully convolutional network with 8 layers (Conv-Bn-Relu, Conv-Relu, and Conv blocks) and without any pooling is used, and in each layer the spatial kernel of the convolutional filter is set to 2x2 for guaranteeing the sub-pixel precision. In Fig. 5.2, the detailed structure of the EPSNet is shown. The EPSNet takes as input the cross EPI patches (HEPP and VEPP) to learn richer representations. Each HEPP and VEPP at

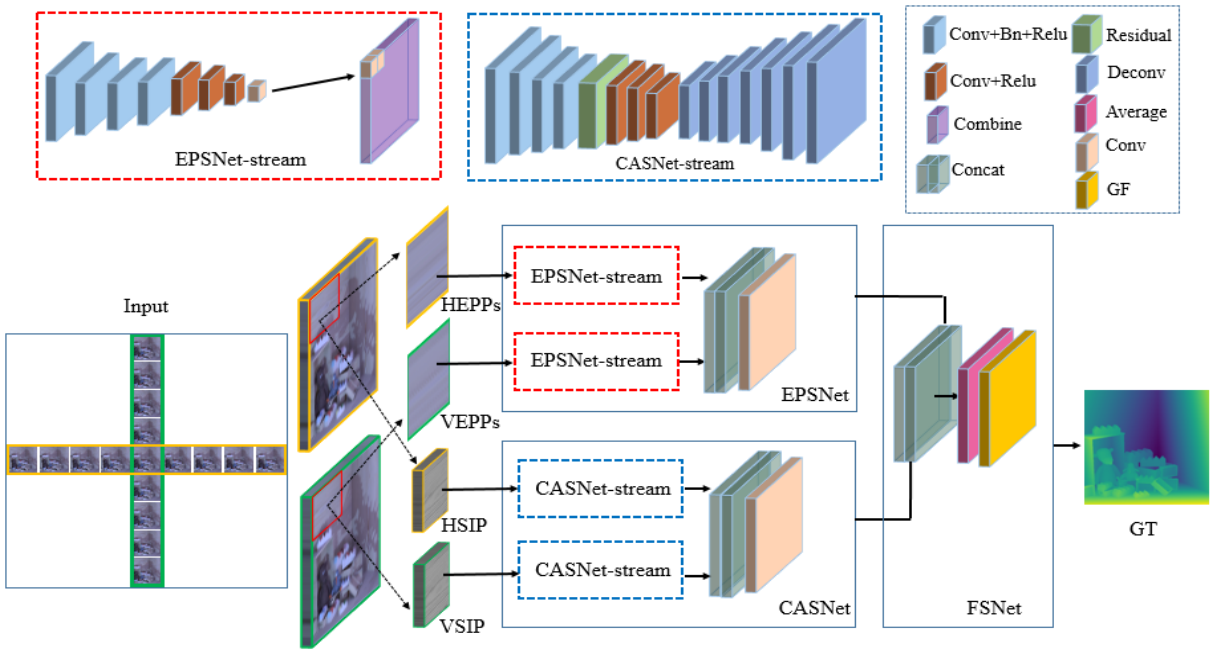


Figure 5.2: The proposed fully convolutional neural network for light field disparity estimation: HFN. The Horizontal EPI Patches (HEPPs) and Vertical EPI Patches (VEPPs) that are sliced from stacked images are fed into the EPSNet-streams, and the Horizontal Stacked Image Patches (HSIP) and Vertical Stacked Image Patches (VSIP) go to the CASNet-streams. After the high-level feature fusion, the disparity maps are obtained.

EPI pixel p with the size EPP_hxEPP_w are pixel-wisely sampled from the horizontal and vertical EPIs respectively, in which the subscript h and w denote the height and width of EPI patch repsectively. The features for the EPI pixel p can be learned from two streams (EPSNet-H for HEPP and EPSNet-V for VEPP). Then the learned features at a higher level from each stream are concatenated, followed by 1x1 convolutional layer for the disparity regression.

5.4.2 Context Subnetwork

The CASNet is designed as an encoder-decoder regression network for dense disparity estimation through learning epipolar property, i.e., searching correspondences constrained by epipolar geometry. This subnetwork takes as input the cross stacked image patches (HSIP and VSIP). For the horizontal stream (CASNet-H), it attempts to regress the horizontal disparities from the HSIP, while the vertical stream (CASNet-V) tries to regress the vertical disparities from the VSIP. The feature representations that are separately learned from each stream are kept in order to fuse and get the more discriminative features at a later stage. In this subnetwork, the HSIP and VSIP have the same dimension IP_hxIP_wxN in which the IP_h, IP_w and N indicate the height, width of image patch and the number of image patches respectively. On the encoder side, it

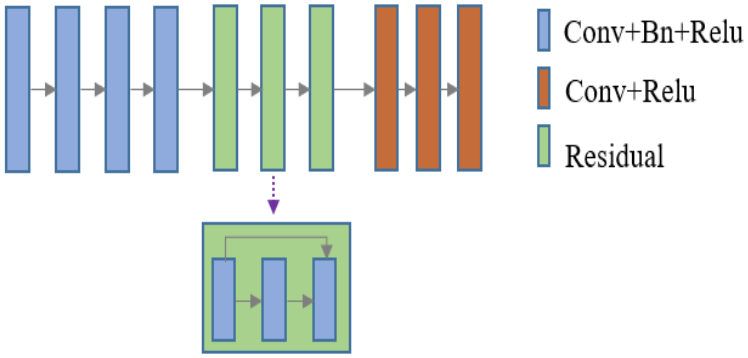


Figure 5.3: The encoder part of the CASNet.

mainly consists of Conv-Bn-Relu and Conv-Relu blocks without any pooling operations. To further boost dense prediction accuracy, we attempt to increase the depth of the network. However, the network becomes difficult to be optimized. To handle this problem, we draw upon the success of residual learning [93] in other computer vision applications, and add several residual blocks in-between Conv-Bn-ReLU and Conv-ReLU blocks, which passes information from a layer to one next layer via skip connections. The overall architecture of the encoder part is shown in Fig. 5.3. On the decoder side, several transposed convolution layers are applied to increase the dimensional information that are lost in the encoder part. Note that we still employ the 2x2 convolutional filters on both sides for accuracy. At the end of the CASNet, the high-level features from two streams are concatenated, followed by 1x1 convolutional layer for the disparity regression. In contrast with EPSNet, this network tends to learn the more structure of the scene, and provides more context information thanks to a larger receptive field that a deeper network leads to.

5.4.3 Fusion Subnetwork

Though the EPSNet and CASNet are able to infer the disparities, a more discriminative network is worth being generated in order to enhance the prediction accuracy. A simple way for the purpose is to score the probability of each subnetwork or similar operations, gathering high confidences from all subnetworks [94, 95]. Instead, in our work, the FS-Net for joint training is proposed, which is superior to that simple fusion approach. Note that the EPSNet performs pixel-wise predictions while the CASNet patch-wisely infers the disparities. Therefore, to fuse the two subnetworks, we firstly combine IP_hxIP_w feature maps with the size 1x1 from the EPSNet into a feature map with the size IP_hxIP_w in order to match the same size of feature map from the CASNet. Note that the spatial locations between two feature maps should be aligned. Then 1x1 convolution and filter operations guided by the original view (GF) [96] are subsequently employed. Though there are a few parameters, the accuracy is significantly boosted.

5.4.4 Training Loss

To guarantee each subnetwork learns the corresponding relationship, the EPSNet, CASNet and *HFNet* are all supervised by the ground truth disparities. The total loss is a weighted sum of the three losses, and each loss is calculated by the mean absolute error that is robust to outliers. The loss weights are set to 0.5, 0.5 and 1.0 respectively.

5.4.5 Implementation Details

Data augmentation: It is an effective technique to enlarge the limited training datasets, which prevents the CNN from overfitting the given data. Considering the light field dataset for training is not large enough, a variety of augmentations have been applied. Firstly, the flipping and rotation are sequentially applied onto the EPI patches to increase the different tilt angle of the EPI-lines, and also applied onto the image patch counterpart. Note that the vertical flipping on image patches will not be influenced since the corresponding points are locally searched in a line here. Specifically, the EPI patch and image patch are horizontally and vertically flipped, rotated by 90, 180 and 270 degrees respectively. Note that the flipping and rotation for EPI patches and image patches of light fields differ from the standard flipping and rotation on images due to the angular property. The sign of the disparity value are reversed during the flipping. When the EPI patches or image patches of light fields are rotated, the horizontal EPI patches or image patches will be turned into the vertical ones since the disparity direction changes.

Training: The proposed *HFNet* is trained on the synthetic CVIA-HCI training dataset, which provides dense ground truth disparity and depth maps. The dimension of stacked image patches in grayscale is chosen to 32x32x9, which corresponds to the height of image patch, the width of the image patch and the number of angular image patches. With respect to the EPI patches in RGB, the dimensions of both the horizontal and vertical EPI patches (HEPPs and VEPPs) are chosen to 32x32x9x9x3. The first two dimensions represent the height and width of the image patch respectively, and the last three dimensions indicate the size of the EPI patch. All the training patches are randomly sampled from the whole training dataset, but the reflection and textureless patches are excluded due to ambiguous disparity estimation. We use the adam optimizer with the default parameter values β_1 = 0.9 and β_2 = 0.999 for the training phase, starting the learning rate 1e-4 and then gradually divided by a half in-between epochs. The model is trained for 12 epochs with 60k iterations, and in each iteration the mini-batch size is 2.

Inference: The angular resolution 9x9 of each light field dataset is used. The size of patch in the inference phase is alternatively chosen to 128x128, considering both the speed and accuracy of disparity predictions. During the inference, the discontinuity

effects came out at the border of patches. To eliminate these effects, the regions with the size 112x112 are selected from that patch.

5.4.6 Ablation Study

We have taken full use of both the EPI-line property and epipolar geometry so that the high quality of depth maps can be generated. For ablation experiments, the same training dataset is used here, but for inference, eight scenes of the test dataset with available ground truth are utilized.

We perform ablation experiments for two subnetworks (EPSNet and CASNet, being responsible for learning the EPI-line orientation and epipolar geometry respectively) and the *HFNet*. For depth predictions, the EPSNet performs well at discontinuity regions but is sensitive to smooth regions, whereas the CASNet is on the opposite. So, in order to integrate the pros of each subnetwork, two ways are explored for fusing the models from each net: one trains the main network and fusion network separately; another trains the main network and fusion network jointly (online fusion). Table 5.4.6 gives the average numerical values from the MSE metric. From the table we learn that any fusion of the two networks gets large gains, explaining the necessities of learning from the two representations. Besides, the online-fusion of the two networks is able to achieve the 16.0% gain, and what is more, supervising the EPSNet and CASNet separately is found better than that only a single supervision of the *HFNet*.

Table 5.1: Performance comparison of ablation components.

Method	MSE
EPSNet	3.01
CASNet	2.95
HFNet	**1.99**
HFNet Online fusion (single loss)	2.268
HFNet Offline fusion (single loss)	2.69

5.5 Methodology - II

Based on the analysis from previous works, we design an end-to-end neural network (*MANet*) for depth estimation from 4D narrow-baseline light fields.

5.5.1 Network Architecture

Fig. 5.4 demonstrates the network architecture of the proposed *MANet* (in ICASSP) for predicting the disparities of the central view.

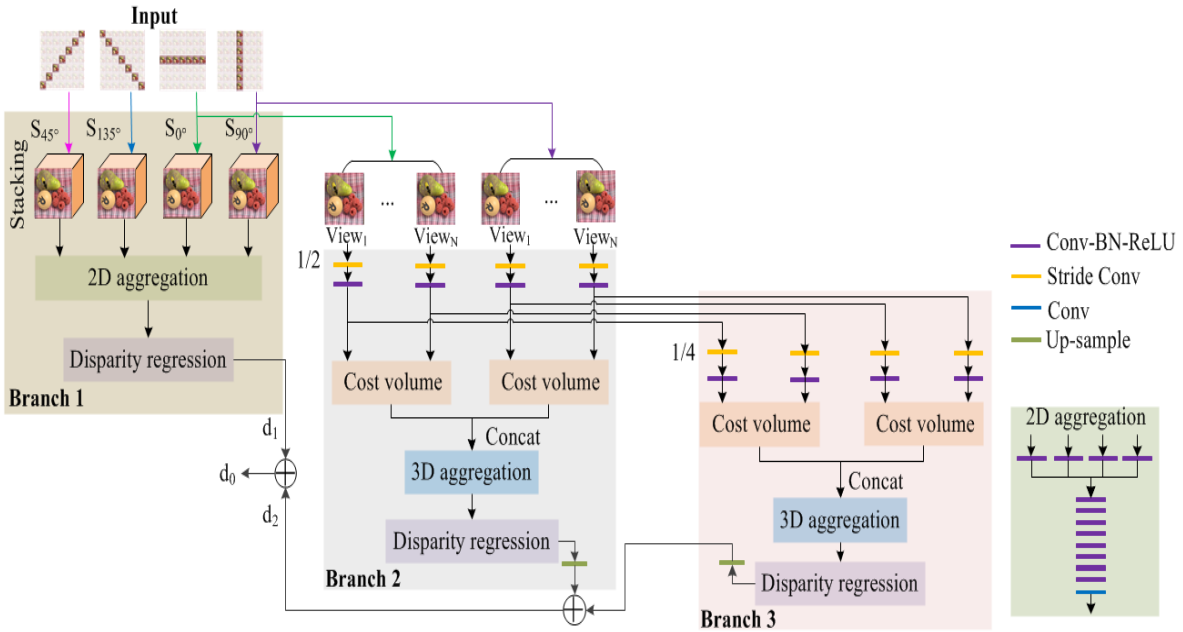


Figure 5.4: The architecture of the proposed *MANet*.

Table 6.1 describes the proposed network *MANet* details. 'ConvBnR' means a convolution layer followed by a batch normalization (Bn) layer and a Relu layer. The prefixes '3D' and '3De' represent 3D convolution and 3D transposed convolution respectively, and the suffixes 'K' and 'S' denote the spatial kernel size and stride respectively. 'x8' means 'ConvBnR' is repeated by 8 times. 'Conv' denotes a convolution layer. 'Upsampling' is a nearest sampling layer. H and W are the image height and width respectively, and H' and W' at $Branch1$ are the reflectively padded image height and width respectively.

The *MANet* consists of three branches with different scales. The basic idea behind it is that the deep lower scale features can bring in large context information, but the information gets lost due to the down-sampling and up-sampling operations, causing the details difficult to be densely recovered. Besides, considering that the disparity space in light fields is continuous, it is significant to maintain full-scale feature representations. The cost volume with few parameters is introduced and combined with 3D aggregation at low scale levels. To compensate for the expensive memory cost in the *cost volume* module, 2D aggregation with low memory (but more parameters) is utilized.

Specifically, instead of using all views for the prediction, our *MANet* employs four streams of views, i.e., horizontal (S_{0°), vertical (S_{90°), left and right diagonal views (S_{45°) and (S_{135°). To the extent, the number of angular views are capable of accurately es-

Table 5.2: The details of the proposed network architecture.

Layers	Output size	Input layer name	Output layer name
$Branch1$			
ConvBnR_K2S1	$H' \times W' \times 40$	$S_{90°}$	Cbr1
ConvBnR_K2S1	$H' \times W' \times 40$	$S_{0°}$	Cbr2
ConvBnR_K2S1	$H' \times W' \times 40$	$S_{45°}$	Cbr3
ConvBnR_K2S1	$H' \times W' \times 40$	$S_{135°}$	Cbr4
Concatenation	$H' \times W' \times 160$	Cbr1,Cbr2,Cbr3,Cbr4	Concat1
ConvBnR_K2S1 $\times$ 8	$H \times W \times 160$	Concat1	Cbr5
Conv_K2S1	$H \times W \times D$	Cbr5	C1
SoftArg	$H \times W \times 1$	C1	d_1
Feature Extraction - Half resolution			
Conv_K2S2	$H/2 \times W/2 \times 8$	$\{v\_1, ..., v\_N\}_{S\_0°}$	$C2\{v\_1, ..., v\_N\}\_S\_0°$
ConvBnR_K2S1	$H/2 \times W/2 \times 8$	$C2\{v\_1, ..., v\_N\}\_S\_0°$	$Cbr6\{v\_1, ..., v\_N\}\_S\_0°$
Conv_K2S2	$H/2 \times W/2 \times 8$	$\{v\_1, ..., v\_N\}\_S\_90°$	$C3\{v\_1, ..., v\_N\}\_S\_90°$
ConvBnR_K2S1	$H/2 \times W/2 \times 8$	$C3\{v\_1, ..., v\_N\}\_S\_90°$	$Cbr7\{v\_1, ..., v\_N\}\_S\_90°$
$Branch2$			
Cost volume	L/2×H/2×W/2× 56	$Cbr6\{v\_1, ..., v\_N\}\_S\_0°$	CV1
Cost volume	L/2×H/2×W/2× 56	$Cbr7\{v\_1, ..., v\_N\}\_S\_90°$	CV2
Concatenation	$L/2 \times H/2 \times W/2 \times 112$	CV1,CV2	Concat2
3DConvBnR_K3S1	$L/2 \times H/2 \times W/2\times$ 16	Concat2	3Cbr1
3DConvBnR_K3S1	L/2×H/2×W/2× 16	3Cbr1	3Cbr2
3DConvBnR_K3S2	L/4×H/4×W/4× 32	3Cbr2	3Cbr3
3DeConvBnR_K3S2	L/2×H/2×W/2× 16	3Cbr3	3DCbr1
3DeConvBnR_K3S1	L/2×H/2×W/2× 1	3DCbr1	3DCbr2
SoftArg	$H/2 \times W/2 \times 1$	3DCbr2	$d\_2^{\hat{}}$
Feature Extraction - Quarter resolution			
Conv_K2S2	$H/4 \times W/4 \times 16$	$Cbr6\{v\_1, ..., v\_N\}\_S\_0°$	$C4\{v\_1, ..., v\_N\}\_S\_0°$
ConvBnR_K2S1	$H/4 \times W/4 \times 16$	$C4\{v\_1, ..., v\_N\}\_S\_0°$	$Cbr8\{v\_1, ..., v\_N\}\_S\_0°$
Conv_K2S2	$H/4 \times W/4 \times 16$	$Cbr7\{v\_1, ..., v\_N\}\_S\_90°$	$C5\{v\_1, ..., v\_N\}\_S\_90°$
ConvBnR_K2S1	$H/4 \times W/4 \times 16$	$C5\{v\_1, ..., v\_N\}\_S\_90°$	$Cbr9\{v\_1, ..., v\_N\}\_S\_90°$
$Branch3$			
Cost volume	L/4×H/4×W/4× 112	$Cbr8\{v\_1, ..., v\_N\}\_S\_0°$	CV3
Cost volume	L/4×H/4×W/4× 112	$Cbr9\{v\_1, ..., v\_N\}\_S\_90°$	CV4
Concatenation	L/4×H/4×W/4× 224	CV3,CV4	Concat3
3DConvBnR_K3S1	L/4×H/4×W/4× 32	Concat3	3Cbr4
3DConvBnR_K3S1	L/4×H/4×W/4× 32	3Cbr4	3Cbr5
3DConvBnR_K3S2	L/8×H/×W/8× 64	3Cbr5	3Cbr6
3DeConvBnR_K3S2	L/4×H/4×W4× 32	3Cbr6	3DCbr3
3DeConvBnR_K3S1	L/4×H/4×W/4× 1	3DCbr3	3DCbr4
SoftArg	$H/4 \times W/4 \times 1$	3DCbr4	$d\_3^{\hat{}}$
Upsampling	$H/2 \times W/2 \times 1$	$d\_3^{\hat{}}$	$d\_3^{\hat{}\prime}$
Average	$H/2 \times W/2 \times 1$	$d\_2^{\hat{}}$,$d\_3^{\hat{}\prime}$	$d\_2^{\hat{}\prime}$
Upsampling	$H \times W \times 1$	$d\_2^{\hat{}\prime}$	d_2
Average	$H \times W$	d_1,d_2	d_0

timate depth, and the computational overhead, the memory footprint could be considerably reduced accordingly. Each stream of views contains a line of views with the dimension HxWxN, where H, W and N denote the height, width and number of views respectively (N is 7, same with EPINET). *Branch 1* adopts 2D aggregation and disparity regression modules in full-resolution, taking as input four streams of stacking views. *Branch 2* and *Branch 3* both employ the cost volume, 3D aggregation and disparity regression modules, fed by half-resolution and quarter-resolution feature maps respectively. Finally, the outputs from three branches are averaged in a coarse-to-fine manner.

5.5.2 Modules

2D aggregation Each stack, concatenated from each stream of views, is fed into a Conv-BN-ReLU block (i.e., convolutional layer followed by batch normalization [97] and ReLU layers) without sharing parameters. Then the outputs from four stacks are concatenated, followed by embedding the grouped outputs into eight Conv-BN-ReLU blocks and one Conv layer. The spatial kernel size and stride for all convolutional filters are chosen to be 2 $\times$ 2 and 1, respectively. The output M_a of this module is represented as:

$$M_a = T_{1,...,9}(\text{Concat}\{T(S_{90^\circ}), T(S_{0^\circ}), T(S_{45^\circ}), T(S_{135^\circ})\}) \tag{5.1}$$

where T indicates the transformation, i.e., the Conv-BN-ReLU block or Conv layer. In order to maintain the full resolution, reflective padding is used on the input images.

Cost volume The feature maps f^l with the size ($H^l \times W^l \times C^l$) for each scale level l (half-resolution and quarter-resolution maps) are generated beforehand and fed to build the cost volumes. Specifically, a series of shared weights layers are employed on each view for the horizontal and vertical streams. The first layer starts by a convolutional layer with stride of 2 for down-sampling. Then a Conv-BN-ReLU block is used to enrich unary view features. Thus the half-resolution feature maps are generated. Two more layers/blocks (one convolutional layer with stride of 2 and one Conv-BN-ReLU block) are used to get the quarter-resolution feature maps.

The feature maps of sub-aperture views are firstly shifted based on the central view for each disparity level $\hat{d}$, in which the bilinear sampling are used for interpolations. Then the shifted unary feature maps from stream α of views at a scale level l are concatenated to form a 4D cost volume ($D^l \times H^l \times W^l \times (N \times C^l)$), as given in Eq. (6.2) and Eq. (6.1):

$$C^l(\hat{d}, v, u, c) = \text{Concat}\{f^l(v + \hat{d}(t^* - t_\alpha), u + \hat{d}(s^* - s_\alpha), c_i), (i = 1, ..., N)\} \tag{5.2}$$

$$\hat{d} = d_{min} + n \times (d_{max} - d_{min})/L, (n \in \{0, 1, ..., L-1\}) \tag{5.3}$$

where the d_{min}, d_{max} and L represent the minimum and maximum disparity in the range and the number of labels respectively. Here L is empirically set to 80 (resulting from our preliminary study which shows it achieves the best trade-off between the performance and the training time). Note that building the cost volume does not introduce any parameters to train.

3D aggregation For the 4D cost volume, five 3D convolution layers and two 3D transposed convolution layers are employed to aggregate the feature information along the disparity axis. The third 3D convolution layer is used as a down-sampling layer in order to facilitate the gains of receptive fields. The first 3D transposed convolution is

used to up-sample the features back to a higher resolution. The spatial kernel size of all layers are set to 3x3. The strides are set to 1, 1, 2, 1, 1, 2, 1 from the 1st convolution layer to the last transposed convolution layer, respectively.

Disparity Regression A differentiable soft argmin operator proposed by [89] is used in the aggregated cost volumes from all branches. Note that for *Branch 1*, the output cost volume is reshaped into $D \times H \times W \times 1$. The soft argmin operator converts the aggregated cost volume into the probability volume along the disparity dimension, and then calculates the predicted disparity $\widetilde{D}$ using the expectation of disparity distributions, as given in Eq. (5.4) [89]:

$$\widetilde{D} = \sum_{\hat{d}=d_{min}}^{d_{max}} \hat{d} * P(\hat{d}) \tag{5.4}$$

where $P(\hat{d})$ denotes the probability volume of pixels at disparity $\hat{d}$. .

5.5.3 Training Loss

The Mean Absolute Error (MAE) is used as the regression loss function, since it is less sensitive to outliers. The total loss is a weighted sum of the three losses from the three outputs, of which the weights are empirically set to 0.5, 0.5 and 1.0 for d_2, d_1 and d_0 (cf. Fig. 5.4), respectively.

5.5.4 Implementation Details

Training: The *MANet* is trained onto CVIA-HCI only for a fair comparison with the proposed *HFN* in Sec 5.4 and other state-of-the-art models. The d_{min} and d_{max} in Eq. (5.3) are set to the default -4 and 4 according to CVIA-HCI. We use patches of size 64x64 randomly cropped from the whole training dataset for the training and augment these patches as done in EPINET [68]. Besides, we exclude low-texture patches and extra reflection patches because of the ambiguous disparity estimation they may cause. We use the rmsprop optimizer [98] with the default parameter value ρ = 0.9, and a constant learning rate 1e-4. The model is trained on the CVIA-HCI for 150k iterations (almost two days and a half), and in each iteration the mini-batch size is set to 16.

Inference: The full-size image is directly used for predicting disparities, which is based on our observation that the inference time is sped up, without sacrificing accuracy.

5.5.5 Ablation Study

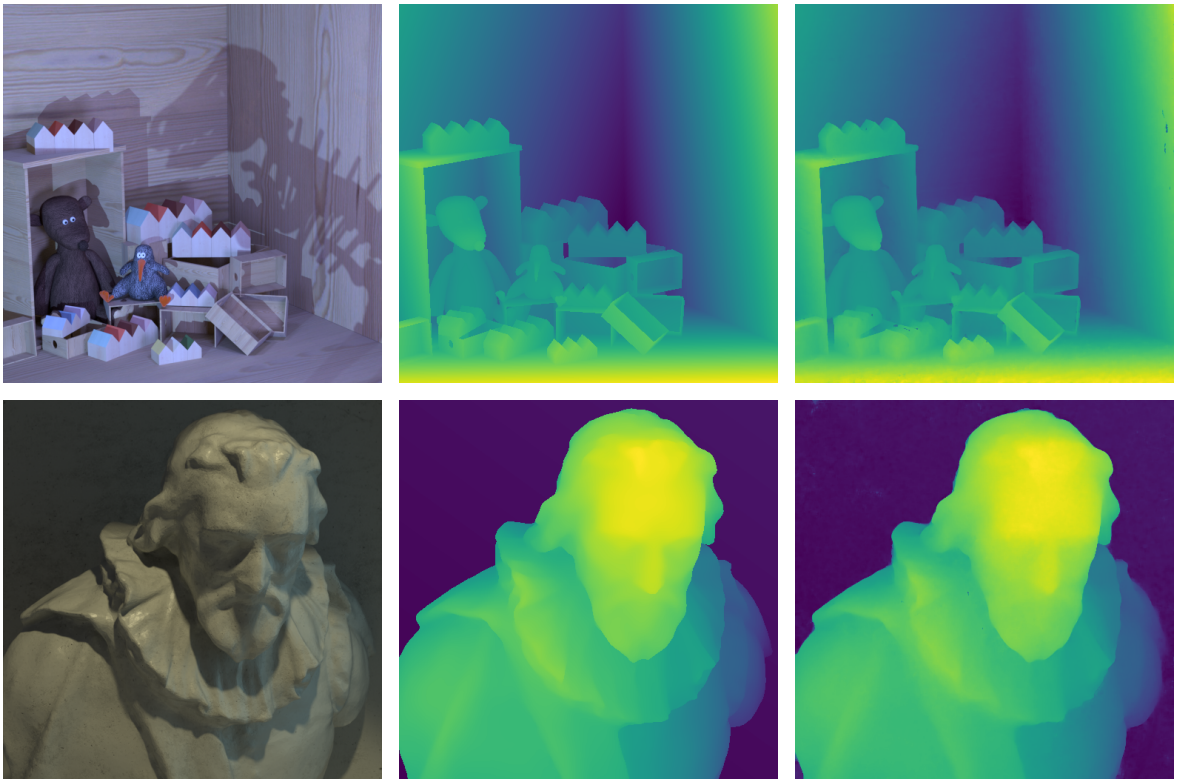
We perform the ablation study to demonstrate the effectiveness of the components designed in the proposed *MANet*. Table 5.3 shows the MSE performance of different variants. We learn that the aggregated network takes advantages from both the 2D aggregation module and the cost volume and 3D aggregation module, and brings in better performance than only using either of them.

Table 5.3: Ablation study: the module is ticked if it is used in training. The number of parameters is in million (M).

Network architecture			Parameters	CVIA-HCI
2D aggregation	Cost volume	3D aggregation	-	MSE
√	-	-	0.88M	2.96
-	√	√	0.94M	3.11
√	√	√	1.58M	2.12

5.6 Exemplar Results

Fig. 5.5 shows the exemplar results of the proposed method, which is tested on the 4D light field dataset with the narrow-baseline. We clearly see from this figure that the proposed network *MANet* produces high quality depth maps, explaining that this network learned the input-output relationship. Besides, these depth maps look almost same with the ground truth. We will conduct the more (comparative) experiments for verifying its effectiveness, and the detailed results are also shown in the Chapter 7.



Central view GT *MANet*

Figure 5.5: An example of depth estimation results of the Dino and Cotton scenes.

CHAPTER 6

DEPTH ESTIMATION FROM WIDE-BASELINE 4D LIGHT FIELDS

Existing traditional and CNN-based methods for light field depth estimation mainly work on the narrow-baseline scenario. This chapter explores the feasibility and capability of CNN to estimate depth in another promising scenario: wide-baseline light fields. Considering the practical goal for real-world applications, we design an end-to-end trained lightweight convolutional network to infer depths from light fields, called *LLF-Net*. The proposed *LLF-Net* is built by incorporating a cost volume which allows variable angular light field inputs and an attention module that enables to recover details at occlusion areas. Evaluations are made on the synthetic and real-world wide-baseline light fields, and experimental results show that the proposed network achieves the best performance when comparing to recent state-of-the-art methods. The proposed *LLF-Net* is also evaluated on the narrow-baseline datasets, and it consequently improves the performance of previous methods and is on par with the proposed methods in previous chapters. The dataset, code and models are available at https://github.com/YanWQ/LLF-Net. Besides, the training dataset is divided into two parts, and the two parts could be quickly found and downloaded from https://zenodo.org/record/3931237#.XwTSSxT7SaE and https://zenodo.org/record/3934712#.XwTTWRT7SaE respectively.

6.1 Introduction

The proposed CNNs in the previous chapter indeed have addressed the concerns existed in the proposed traditional method. However, the performance of the proposed CNNs in the wide-baseline light fields is limited since we found that learning the relation between the the EPI-line and the label or using 2D convolutions for inferring depth was not effective. Moreover, the angular resolution of the two proposed CNNs is fixed after the training, and when the test set has a different angular resolution, the CNN has to be retrained, which is inconvenient and time-consuming. In this chapter, we propose a new network architecture that is modified from the proposed *MANet* to tackle the limitations.

As mentioned before, the narrow-baseline light fields are typically captured by a

plenoptic camera, and the baseline between sub-aperture images is very narrow. To date, traditional [19, 54, 55, 58–61, 99–103] and CNN-based [64–68, 104, 105] methods have been well studied for high performance in narrow-baseline light fields, and achieved low percentage of errors, e.g. the proposed *HFN* and the proposed *MANet* in Chapter 5. For wide-baseline light fields, they are usually captured by a camera array or gantry (i.e., a conventional camera is placed onto a gantry, and then uniformly moved by a motor in a plane). The baseline between the recorded wide-baseline light-field images is large and the spatial resolution of images is usually high. To date, considerable efforts has been also made by traditional methods [27, 55, 60, 61, 101, 106, 107] to solve the problem of depth estimation in wide-baseline scenario. However, CNN-based approaches are rarely studied in this scenario due to the deficiency of training data. Our objective is to explore and apply CNNs into depth estimation for wide-baseline light fields.

With respect to the CNN model for wide-baseline scenario, Shi *et al.* [31] present a divide-and-train model with around 199 million parameters and Leistner *et al.* [70] present an end-to-end trained model with 36 million parameters. Both models are heavyweight that cannot satisfy our needs since we consider the more practical goals, e.g. applications in mobile devices. We made an attempt to resort to top-performing EPINET [68] in narrow-baseline scenario with only 5.1 million parameters to test and re-train (denoted as EPINET_T) the proposed *WLF* dataset, but the performances are too poor, which also fails to fulfill our goal. Thus, we propose a novel end-to-end trained lightweight network *LLF-Net* by taking knowledge from stereo-based CNN models. In our network, features are extracted for each view of cross-hair light fields, and then the cost volume is generated by shift-interpolation, cost calculation and fusion operations. With respect to the fusion, a divide-concatenate-sum operation is proposed, allowing flexible light field inputs and maintaining depth accuracy. An attention mechanism is introduced in the cost aggregation module, which enhances depth accuracy at occlusion regions. We make evaluations of the proposed network on *WLF* test sets and real-world datasets, and experimental results show that our network outperforms state-of-the-art methods in both quantitative and qualitative evaluations. Further, we validate and compare our model with state-of-the-art methods for narrow-baseline scenario, and our model achieves the best performance.

6.2 Methodology

An overview of the proposed network architecture is illustrated in Fig. 6.1 and detailed in Table 6.1. Given the full-shape light fields, a cross-hair of light field images are chosen and fed into the proposed network. To make image correspondence features distin-

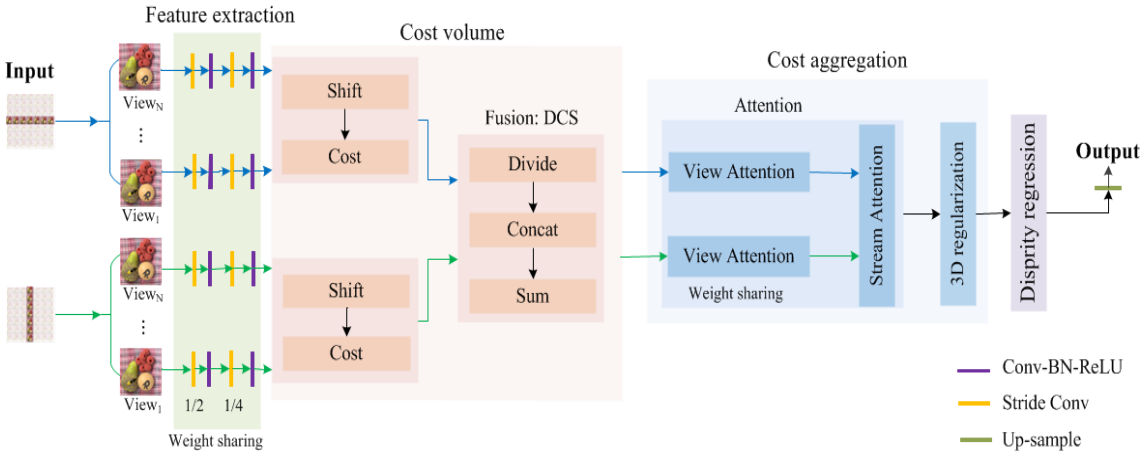


Figure 6.1: Overview of the proposed network architecture.

Table 6.1: The details of the proposed network architecture.

Layers	Output size	Input layer	Output layer
Feature extraction (for each X $\in$ $S_0 \cup S_{90}$)			
Conv_K2S2	$H/2 \times W/2 \times C$	X	C1_1
ConvBnR_K2S1	$H/2 \times W/2 \times C$	C1_1	C1_2
Conv_K2S2	$H/4 \times W/4 \times C$	C1_2	C2_1
ConvBnR_K2S1	$H/4 \times W/4 \times C$	C2_1	C2_2
Cost volume (C2_2S_0 = $\{C2\_2\}_1^N$, C2_2S_{90} = $\{C2\_2\}_1^N$)			
Shift_Cos	L/4×H/4×W/4× NC	C2_2S_0	SIC1
Shift_Cos	L/4×H/4×W/4× NC	C2_2S_{90}	SIC12
Div_Concat	L/4×H/4×W/4× $6C$	SIC1, SIC2	$\{DC\}_1^{3(N-1)/2}$
Sum	L/4×H/4×W/4× $6C$	$\{DC\}_1^{3(N-1)/2}$	CV
View and Stream attention			
View Attention	L/4 × H/4 × W/4 × 6C	CV_0	CV_v1
View Attention	L/4 × H/4 × W/4 × 6C	CV_{90}	CV_v2
Stream Attention	L/4 × H/4 × W/4 × 6C	CV_v1, CV_v2	CV_s
Cost regularization			
3DConvBnR_K3S1	L/4 × H/4 × W/4 × 2C	CV_s	3Cbr1
3DConvBnR_K3S1	L/4 ×H/4 × W/4 × 2C	3Cbr1	3Cbr2
3DConvBnR_K3S2	L/8 × H/8 × W/8 × 4C	3Cbr2	3Cbr3
3DeConvBnR_K3S2	L/4 × H/4 × W/4 × 2C	3Cbr3	3DCbr1
3DeConvBnR_K3S1	L/4 × H/4 × W/4 × 1	3DCbr1	3DCbr2
Upsampling	L × H × W × 1	3DCbr2	Up1
SoftArg	H × W × 1	Up1	$\widetilde{D}$

guished, deep feature descriptors are extracted for each view from cross-hair views in the *Feature Extraction* (Section 6.2.1). Next, the discriminative cost volume [54, 55] is constructed by operating all extracted features in the *Cost Volume Generation* (Section 6.2.2). Afterwards, an attention mechanism is introduced to remove disparity errors caused by occlusion, and 3D encoder-decoder network is applied to regularize the disparity space in the *Cost Aggregation* (Section 6.2.3). Finally, the disparity map is produced in *Disparity Regression* (Section 6.2.4), and a robust loss (Section 6.2.5) is used for training our network.

6.2.1 Feature Extraction

Our network takes as input horizontal and vertical streams of image views with the dimension $H \times W \times N$ from light fields, where H and W represent height and width of image (spatial resolution). We apply a 2D plane convolution network to extract distinguished features. It is firstly constructed by two Conv-Bn-Relu blocks (convolution layer followed by a batch normalization layer, and a ReLU unit), in which the stride of the former convolution layer is set to 2 for down-sampling inputs and the latter is set to 1. Then the blocks with the same structure are repeated to produce sub-scale features. Finally, the output of feature maps are downsized to quarter spatial resolution. The kernel of convolution filters is 2x2 for sub-disparity space. We adopt the shared 2D network on both streams of views since we found sharing parameters is better than non-sharing case in terms of disparity accuracy and efficiency.

6.2.2 Cost Volume Generation

Given two streams of feature maps, a sequence of operations, i.e., shift-interpolation, cost calculation and fusion are used to generate the cost volume. Note that building the cost volume does not introduce any parameters to train.

Shift-Interpolation

The feature maps of central view F_r are regarded as the reference, and the others along the stream are the target feature maps F_t. The feature maps of target views F_t are shifted toward the reference view by each hypothesis disparity $\hat{d}$ within disparity range (see Eq. 6.1). Then the bilinear interpolation is employed to calculate appropriate values for each pixel at sub-pixel position.

$$\hat{d} = d_{min} + n(d_{max} - d_{min})/L, (n \in \{0, 1, ..., L-1\}) \tag{6.1}$$

where the d_{min}, d_{max} and L represent the minimum and maximum disparity in the range and the number of labels respectively. The dimension of (warped) feature maps for each target view and the feature maps for reference view is herein ($L \times H^l \times W^l \times C^l$), where l denotes the scale level and C indicates the channels.

Cost Calculation

After obtaining the warped feature maps from streams of target views at the scale level l, we then calculate the matching cost by using the concatenation [89] between the reference and target feature maps to form a 4D cost volume , as shown in Eq. (6.2)

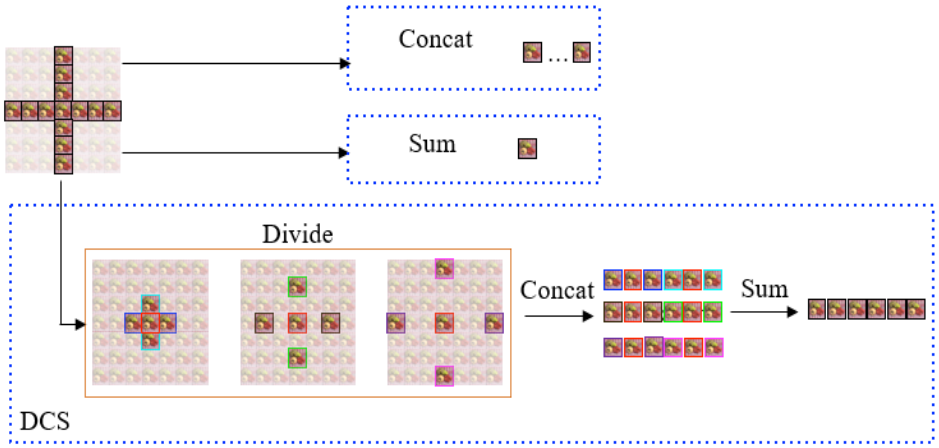


Figure 6.2: Variants for cost fusion (best viewed in color).

and Eq. (6.1):

$$C^l(\hat{d}, v, u, c) = \mathrm{C}\{F^l(v + \hat{d}(t^* - t), u + \hat{d}(s^* - s), c_i) \\ , (i = 1, ..., N)\} \tag{6.2}$$

Fusion

At this step, we make a fusion of calculated costs across views and streams such that neighboring views or streams enhance capabilities of solving ambiguity problems in correspondence matching. Actually, there exists different strategies to perform cost fusions. From the perspective of input sizes, strategies vary from fixed, to non-fixed or near-fixed inputs. Hereafter we discuss fusion variants in details, as are demonstrated in Fig. 6.2.

Concatenation is employed to stack all reference-target pairs of costs or horizontal and vertical groups of costs along the channel dimension, where the stacked size of the former is $L \times H^l \times W^l \times 4NC$, and the latter is $L \times H^l \times W^l \times 2NC$. Since the number of stacked feature channels is equal to that of convolution input filters, the networks then require the fixed inputs.

Sum computes the sum of all reference-target costs in which each cost is calculated by the absolute difference between the reference and target view. The sum fusion produces the fixed-length output $L \times H^l \times W^l \times C$ regardless of the input size.

Divide, Concat, Sum (DCS) is designed to fuse costs across multiple-baseline cost volumes. The cross-hair views are partitioned into $(N-1)/2$ divisions of a 3x3 cross shape. For each division, all feature maps are concatenated across the channel dimension. Finally, we take the sum over all divisions and the fused output has the dimension $L \times H^l \times W^l \times 6C$. DCS is flexible in the number of input angular views N. Note that when N is set to 3, DCS will be same with concatenation fusion.

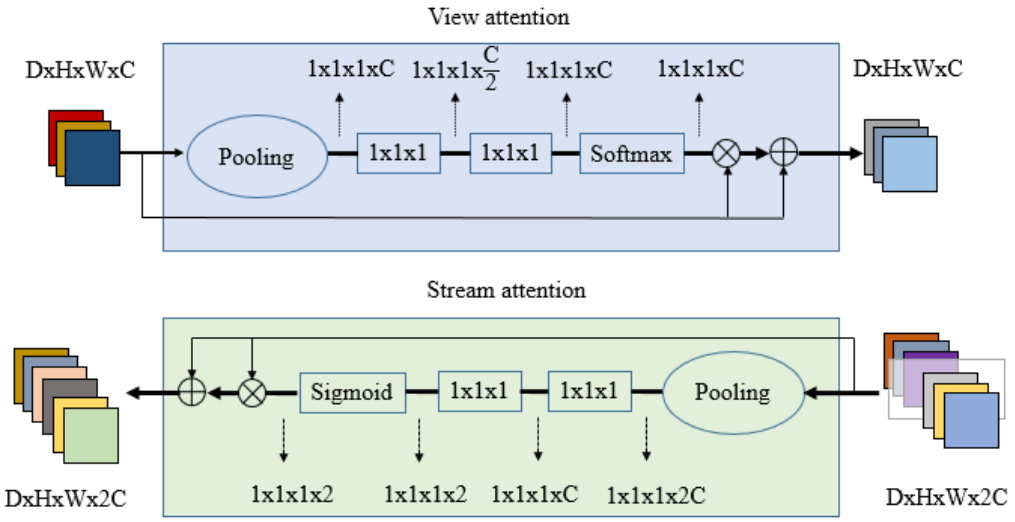


Figure 6.3: Epipolar view and stream residual attention. Global max-pooling is used in pooling to downsize inputs, and each first 1x1x1 convolution is followed by a ReLU activation.

6.2.3 Cost Aggregation

The cost aggregation is leveraged to refine the fused cost volume since we did not take into account the potential occlusion issues before. With respect to the occlusion, we know that the same 3D real-world point that is visible in the reference view might be occluded by foreground objects in the target view, which leads to difficulties in finding correspondences. This issue might be alleviated in our input cross-hair light fields since points might be visible in some angular views. Moreover, we are aware that points are heavily occluded in horizontal stream of views but might be less or not occluded in vertical streams of views, and vice versa (cf. Fig. 6.6). This is also true for views in any a stream. It causes troubles to find the correct disparity. To address this issue, we propose to apply the 3D attention network similar to the 2D attention network used in semantic segmentation [109] and super-resolution [110] tasks, where the view and stream residual attention network as shown in Fig. 6.3 will assign automatic weights to the views in a stream and these two streams. Specifically, at the view attention network, the pooling is firstly used to extract global features, and then two 1x1x1 3D convolutions are used for down-sampling and up-sampling the features for the non-linear characteristic. Afterwards, the softmax operation is used for normalizing the features into the weights, which is required to assign higher weights to the un-occluded views than occluded views during the training phase. Likewise, we adapt the multi-view attention to be the binary-stream network, where the softmax is replaced by the sigmoid operation.

Followed by the attention network, a 3D encoder-decoder network is used to regularize the output of the attention network across the disparity dimension. This naturally involves large context information, which enforces the smoothness at low texture re-

gions. This network is built by three 3D convolutions and two transposed convolutions. Last, the bilinear interpolation is used to resize back to the same spatial resolution of inputs, and the output has the dimension $L \times H \times W \times 1$.

6.2.4 Disparity Regression

The differential soft argmin operator proposed by [89] is employed to obtain the final disparity map. The soft argmin operator regresses continuous disparities $\widetilde{D}$ by calculating the expectation of weighted disparities, as given in Eq. 6.3,

$$\widetilde{D} = \sum_{\hat{d}=d_{min}}^{d_{max}} \hat{d} * P(\hat{d}) \tag{6.3}$$

where $P(\hat{d})$ is the weight probability of the pixel at disparity $\hat{d}$.

6.2.5 Training Loss

We use the smooth L1 loss for the training process, which is less sensitive to outliers and more possibly gets close to the minima due to the small gradient. The loss is computed between the predicted disparity $\hat{d}$ and the ground truth g in patch p as in Eq. 6.4 and Eq. 6.5,

$$L = \sum_{i \in p} Smooth_{L1}(\hat{d}_i - g_i) \tag{6.4}$$

$$Smooth_{L1}(x) = \begin{cases} 0.5x^2 & |x| \leq 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{6.5}$$

6.2.6 Implementation Details

We use randomly cropped patches of size 128x128 for wide-baseline training set *WLF* and a smaller size 64x64 for narrow-baseline training set CVIA-HCI (due to its smaller quantity). Color scaling, 90, 180 and 270 degree rotation, etc are used for increasing the number of the data samples to the order of millions. We use the rmsprop optimizer [98] , and start at the learning rate 1e-4, and then divide it by two after 80k iterations for *WLF* and after 150k iterations for CVIA-HCI. For each iteration the mini-batch size is 8 for *WLF* and 16 for CVIA-HCI respectively. The d_{min} and d_{max} in Eq. (6.1) are set to 0 and 50 for *WLF*, -4 and 4 for CVIA-HCI respectively. The number of labels L is set to 128.

Table 6.2: Comparisons of the bad-0.3, bad-0.6 and parameters for three fusions in *Cost volume generation*. The best performance is in **bold**.

Fusion	Parameters	Adaptive	Hand-designed	
			bad-0.3	bad-0.6
Concat	2.5M	✗	**7.00**	3.37
Sum	1.5M	✓	12.72	5.33
DCS	1.8M	✓	7.01	**3.04**

6.2.7 Ablation Study

To validate the effectiveness of two proposed components in the *LLF-Net*, i.e. the fusion in *Cost volume generation* and the attention in *Cost aggregation*, the ablation studies are conducted on the Hand-designed validation set that consists of 8 scenes split from the training set.

Fusion in *Cost volume generation*

Firstly, we make quantitative comparisons of different variants of fusions in *Cost volume generation* on aspects of depth accuracy and model size. Table 6.2 shows the evaluation results, and compares their adaptive ability of testing various angular resolutions without retraining the new angular resolution inputs. The proposed DCS fusion gets the best performance by bad-0.6 metric with considerable parameters.

Fig. 6.4 compares the performance results between two fusion ways (Sum and DCS) from testing variable angular resolution. The DCS fusion always produces more accurate depths than the Sum fusion. When limiting the angular resolution, the DCS achieves much better performance, which means that it is more adaptive to limited input views. Fig. 6.5 illustrates visual comparison results from these two fusions. The DCS fusion witnesses the degradation in performance, but this is much less than that from the Sum fusion where artifacts occur in the disparity map.

Attention networks in *Cost aggregation*

To test the necessity of the proposed attention networks, Table 6.3 compares the quantitative evaluation results with and without using them. We can find that using the attention networks considerably improves the quality of estimated disparity maps.

Fig. 6.6 shows a visual comparison of disparity estimation without and with using attention networks. For a pixel in the selected patch P (see Fig. 6.6 (b)), it is occluded in all horizontal views, but it is visible in all vertical views. With the attention networks for selecting more meaningful views, the disparities of pixels around occlusion regions

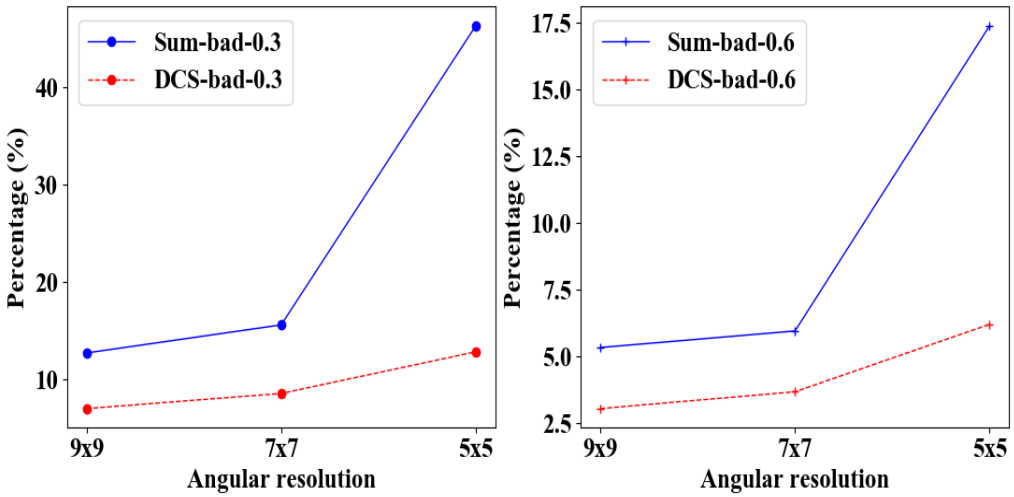


Figure 6.4: Comparisons of DCS fusion and Sum fusion on flexible angular inputs. The number in the vertical axis depicts the percentage of the bad pixels.

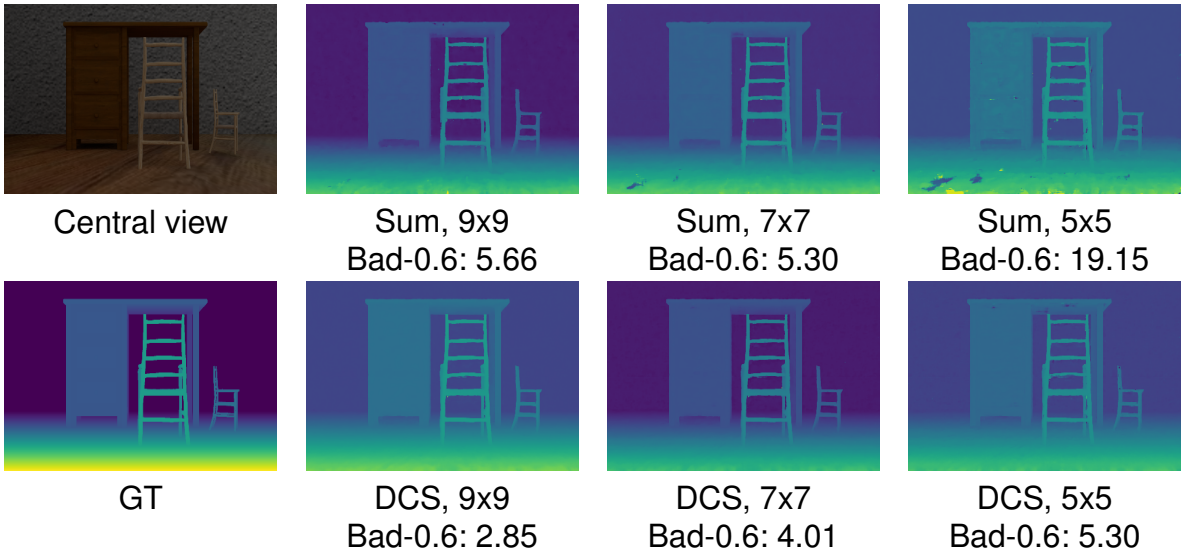


Figure 6.5: Visual comparisons of DCS fusion and Sum fusion on flexible angular inputs.

Table 6.3: Comparisons of the depth accuracy and parameters with and without attention block in *Cost aggregation*.

Module	Parameters	Hand-designed	
		bad-0.3	bad-0.6
W/o Attention	1.79 M	7.01	3.04
With Attention	1.82 M	**6.25**	**2.72**

Table 6.4: Training dataset scheduling.

Dataset	Hand-designed	
	bad-0.3	bad-0.6
Hand-designed	10.18	5.53
Hand-designed+Flying-objects	**6.25**	**2.72**

are better estimated and the sharp boundaries at depth discontinuities are better preserved, as shown in Fig. 6.6 (c-e).

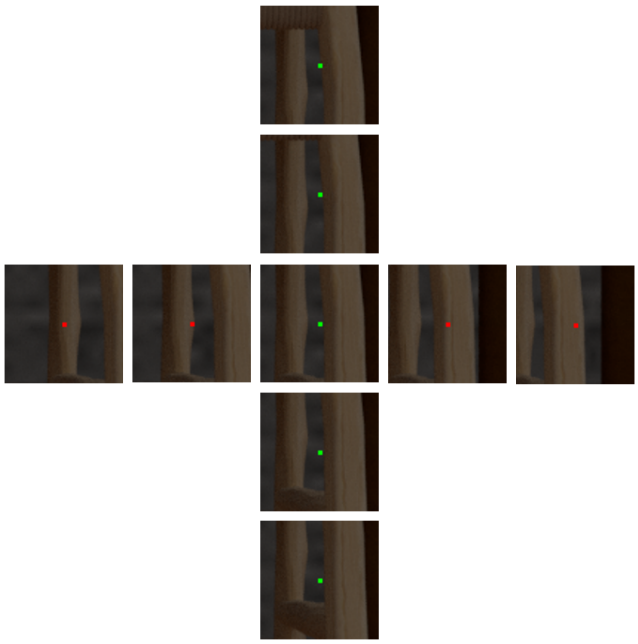
Dataset Scheduling

To check the necessity of the Flying-Object subset in the proposed *WLF* dataset, we performed ablation experiments under two different training set scheduling schemes. As is shown in Table 6.4, the qualitative performance with Flying-objects (with large-scale training frames) in training is improved by a large margin.

6.3 Exemplar Results

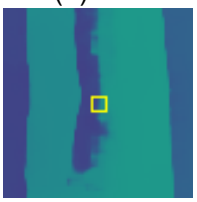
Fig. 6.7 demonstrates the exemplar results of the proposed method, which is tested on the 4D light field dataset with the wide-baseline and narrow-baseline respectively. We clearly see from this figure that the proposed network *LLF-Net* shows high capabilities in capturing the fine details at occlusion regions, and meanwhile have few noticeable artifacts in the foreground and even textureless regions. The more (comparative) experiments will be also carried on, and the detailed results will be also shown in the Chapter 7.

(a) Central view

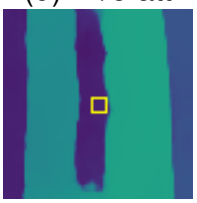


(b) Horizontal and vertical views

(c) GT



(d) W/o att



(e) With att

Figure 6.6: Visual comparisons of depth estimation results without and with attention block. (a) central view with a selected patch P in pink bounding box, (b) the patch P (the intersection) and the corresponding patches in the horizontal and vertical views, where the red point indicates the pixel in the central view is occluded in the current view, and the green point means visible in the current view, (c) the ground truth disparity of patch P, (d) the estimated disparity map without attention block and (e) the estimated disparity map with attention block (best viewed in color).

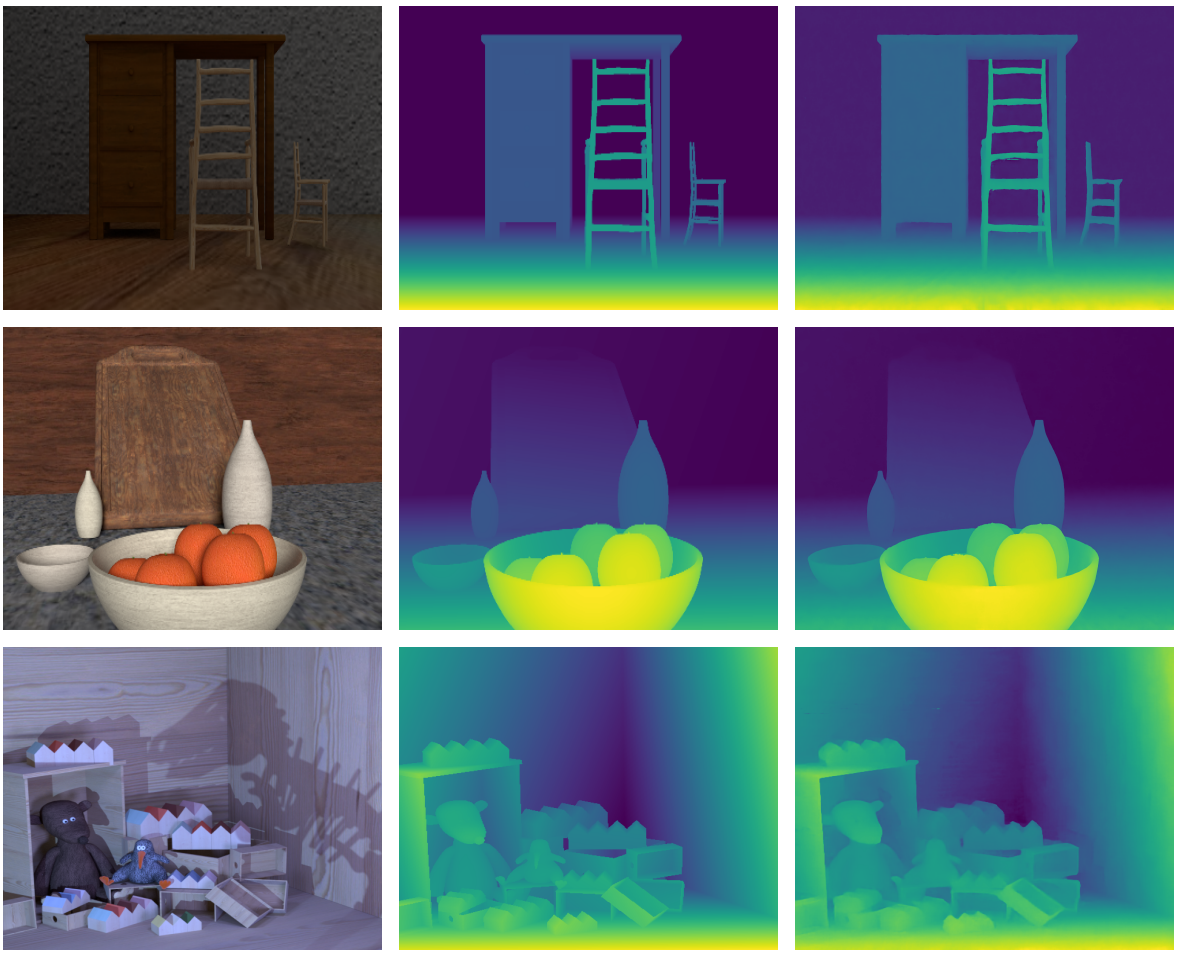


Figure 6.7: An example of depth estimation results on the Desk, KitchenTable and Dino scenes respectively.

CHAPTER 7

EXPERIMENTS

In this chapter, the evaluations are made on the 3D light field datasets and the 4D light field datasets respectively. For the latter, a common data set mentioned in Chapter 2 is used as our test set, and comparative experiments among the various proposed methods in the Part I and II, are made in a coherent way. For the assessment, two metrics are adopted: a quantitative metric (MSE and Bad pixel) for the synthetic datasets, and a qualitative metric for the real-world datasets.

7.1 Experimental Environment

The whole experiments are carried on a Windows PC equipped with an Intel i7 3.6Ghz CPU with 32GB memory. The proposed traditional methods *R3DE* and *S-R4DE*, which are described in Chapters 3 and 4 respectively, are both implemented in C++, and run on the CPU. The proposed CNN-based networks *HFNet*, *MANet* and *LLF-Net*, which are described in Chapters 5, 5 and 6) respectively, are all implemented in Tensorflow [111]. The training and the inference of the proposed neural networks are both run on a Nvidia GTX 1080Ti GPU with 11GB memory and the same CPU.

7.2 3D Light Fields

The evaluation is performed on the Disney dataset to verify the effectiveness of the proposed method *R3DE* targeting the 3D light fields. The proposed method is made comparisons with the state-of-the-art depth estimation methods FTC [19] and LAGC [47], which are also targeting the 3D light fields. Fig. 7.1 demonstrates the visual comparison results of the central view in the light fields. The *R3DE* shows less sensitive to homogeneous regions when compared with the FTC, while retaining the more sharp boundaries accuracy at occlusion regions in contrast with the LAGC. In general, the proposed *R3DE* is capable of to estimating high quality depth for the sparse light fields with the wide-baseline.

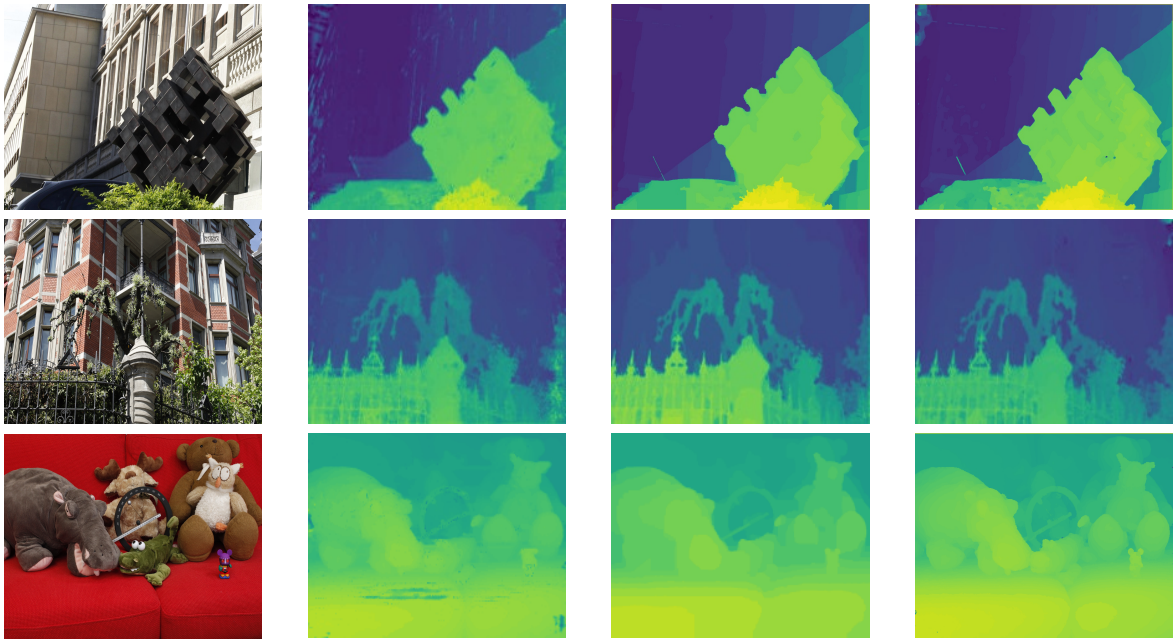


Figure 7.1: Visual comparisons of depth maps with state-of-the-art methods. The scene from the top to the bottom: Statue, Mansion, and Couch.

7.3 4D Light Fields

To verify the effectiveness of the proposed depth estimation methods for the 4D light fields, the evaluation is performed on the popular 4D light field datasets in the literature, being composed of the narrow-baseline datasets and wide-baseline datasets.

7.3.1 Performance on Narrow-baseline Datasets

The proposed methods for evaluations consist of one traditional method *S-R4DE*, and three CNN-based methods *HFNet*, *MANet*, and *LLF-Net*. We compare all the proposed methods with recent state-of-the-art depth estimation methods, comprising of traditional light field depth estimation methods (LF_OCC [55], LF [54] and RPRF [61]), and CNN-based methods (EPINET [68] and LBDE-E [31]).

Next we will show the performance comparisons, being comprised of the accuracy and runtime, the model performance, and qualitative (visual) comparisons.

Accuracy Comparison

Since the EPINET [68] sacrifices the 11-pixels length at each border of the image in the output resolution, we exclude these pixels in ground truth when evaluating its performance for an available/fair comparison.

Comparison of traditional methods: Table 7.1 and Table 7.2 illustrate that the proposed traditional method *S-R4DE* achieves the highest accuracy of depth maps on the 4D narrow-baseline testing dataset when compared with the traditional methods LF_OCC, LF and RPRF. The *S-R4DE* surpass 7.7% and 20.7% on the average of bad-0.1 and bad-0.07 error respectively, compared with the second RPRF.

Comparison of CNN-based methods: From Table. 7.1 and Table 7.2, we see that the proposed *MANet* and the proposed *LLF-Net* outperforms all of the other state-of-the-art methods in terms of average MSE and bad pixel percentages. The *MANet* achieves 34.7% MSE gain, and surpasses by 14.5% on the bad-0.1 error, compared with the EPINET. We also see that the proposed *LLF-Net* improves the mse and makes a decrease by a large percentage in bad-0.1, when compared with the EPINET. The *LLF-Net* is worse than the *MANet* in MSE but better in bad-0.1. Besides, the *LLF-Net* achieves similar accuracy with the EPINET in bad-0.07 metric.

Traditional methods vs CNN-based methods: From Table. 7.1 and Table 7.2, it is clear to notice that most of the CNN-based methods accordantly produce the lower depth errors than the traditional methods on this testing set. The proposed *MANet* and the proposed *LLF-Net* demonstrate a large margin of improvement in comparison with the proposed *S-R4DE* thanks to a large number of training patch samples.

Runtime Comparison

We further compare the computational efficiency of the proposed traditional method and the proposed CNN-based methods with the aforementioned methods in **Accuracy Comparison**. We give the results tested on the CVIA-HCI dataset in Table 7.3 and Table 7.4 for such comparisons. The reason why the runtime comparison is divided into two comparisons is that the traditional methods only report the elapsed time by CPU and the CNN-based methods usually report the GPU runtime, since the GPU is able to be used for accelerations in CNN-based methods but this is not always the case in traditional methods. Here, for methods [54, 55, 61, 68], we directly use the runtime values from the CVIA-HCI benchmark website and the others from the reported papers. Note that the traditional methods are all run on the CPU. As is given in Table 7.3, the proposed *S-R4DE* does not run slowly though none of speed optimizations is used in the implementation. Table 7.4 shows that the proposed *LLF-Net* runs the fastest (less than 0.5s per frame) among the CNN-based methods, about 3 times faster than the EPINET (also using a single Nvidia GTX 1080Ti GPU), which seems more practical in real applications. Further, the proposed *MANet* runs the second fastest among these methods.

Table 7.1: Comparison results of MSE on the CVIA-HCI and HCI test scenes. The lowest MSE (highest accuracy) is highlighted in bold for each line.

	Scenes	Traditional methods						CNN-based methods				
		LF_OCC	LF	MWBM	*S-R4DE*	RPRF	FDE	EPINET	LBDE-E	*HFNet*	*MANet*	*LLF-Net*
CVIA-HCI	Backgammon	22.78	13.01	-	6.50	5.58	10.35	3.63	14.48	**3.40**	4.24	8.06
	Boxes	9.59	18.84	-	9.43	8.55	12.10	6.24	10.30	**4.32**	5.21	8.02
	Cotton	1.07	9.19	-	5.00	0.81	0.65	**0.19**	0.72	0.31	0.32	0.55
	Dino	0.94	1.16	-	0.88	0.49	0.62	**0.17**	0.55	0.60	0.20	0.40
	Dots	3.19	5.68	-	25.44	21.21	4.05	**1.64**	23.07	4.37	4.44	6.16
	Pyramids	0.08	0.27	-	0.02	0.06	0.02	**0.01**	0.02	0.03	0.02	0.03
	Sideboard	2.07	5.09	-	1.63	1.34	1.85	0.80	1.05	1.22	**0.70**	1.17
	Stripes	7.94	17.45	-	4.19	7.90	1.37	**0.95**	3.41	1.58	1.56	2.39
HCI	Buddha	0.91	1.13	0.53	0.33	0.28	-	0.36	0.41	0.95	**0.27**	0.37
	Buddha2	1.18	0.45	0.55	0.50	0.75	-	6.64	**0.26**	1.36	1.37	0.41
	Horses	1.36	1.70	1.06	0.52	**0.50**	-	7.35	0.79	7.01	4.76	1.38
	Medieval	1.15	1.40	0.79	0.98	0.79	-	2.28	0.74	1.00	0.77	**0.64**
	MonasRoom	0.73	0.66	0.65	0.52	0.47	-	1.33	0.39	0.52	0.40	**0.34**
	Papillon	1.00	5.98	1.98	0.77	0.66	-	6.12	0.58	1.46	0.91	**0.54**
	StillLife	4.29	2.10	2.21	**1.06**	1.96	-	2.43	1.07	3.62	**1.06**	1.50
	Average	3.89	5.61	-	3.42	3.37	-	2.68	3.86	2.12	**1.75**	2.13
	Median	1.18	2.10	-	0.98	0.79	-	1.64	0.74	1.36	0.91	**0.64**

Note: The results of all the methods on CVIA-HCI are got from the CVIA-HCI benchmark website, except those of the LBDE-E which are provided by the authors. The results on HCI are reported in [61], except those of the EPINET which are obtained by running their public code. Those of the LBDE-E are provided by authors.

Table 7.2: Comparison results of average bad pixel percentage on the CVIA-HCI and HCI test scenes. The lowest bad pixel percentage value (highest accuracy) is highlighted in bold for each line.

Method	Traditional methods				CNN-based methods				
	LF_OCC	LF	*S-R4DE*	RPRF	EPINET	LBDE-E	*HFNet*	*MANet*	*LLF-Net*
bad-0.1	17.89	10.74	9.53	10.32	9.06	9.86	17.89	7.75	**6.60**
bad-0.07	30.16	16.20	12.63	15.93	10.54	13.61	23.58	**10.34**	10.66

Table 7.3: Comparison results of running time by traditional methods on CVIA-HCI test scenes.

Method	Traditional methods			
	LF_OCC	LF	S_R4DE	RPRF
Device	CPU	CPU	CPU	CPU
Time(s)	1.05e4	1.01e4	78	34.53

Table 7.4: Comparison results of running time by CNN-based methods on CVIA-HCI test scenes.

Method	CNN-based methods				
	EPINET	LBDE-E	*HFNet*	*MANet*	*LLF-Net*
Device	GPU	GPU	GPU	GPU	GPU
Time(s)	1.98	1.92	5.28	0.73	**0.46**

Model Comparison

At the same time, we compare the performance of CNN-based models, including the number of parameters, training time and training fashion. As shown in Table 7.5, all CNNs are end-to-end trained except LBDE-E, which might fall into the sub-optimal mimina during the training. We notice from the table that the proposed *MANet* has the fewest parameters, about 3 times fewer parameters than the EPINET, and 125 times fewer than LBDE-E. The proposed *LLF-Net* has the similar few parameters to that of *MANet*, but just require less than two days for training the same training set with the EPINET and the proposed *MANet*, which are trained more than five days and two days respectively.

Table 7.5: Performance comparison results on aspect of model parameters and training days.

Method	End-to-end trained	Parameters (M)	Training days
EPINET [68]	✓	5.1	5-6
LBDE-E [31]	✗	198.8	≈ 2
HFNet	✓	19.5	≈ 2
MANet	✓	**1.6**	≈ 2.5
LLF-Net	✓	**1.8**	$\approx$ **1.6**

Visual Comparison

Synthetic datasets Fig. 7.2 shows the proposed visual comparisons against the ground truth and the state-of-the-arts on the CVIA-HCI and HCI datasets. From this comparison, we clearly observe that the proposed methods produces closer depth maps to the ground truth. The proposed *S-R4DE*, *MANet* and *LLF-Net* perform much better at occlusion regions, preserving depth discontinuity (cf. the region between the buddha and the dice in "Buddha", the region between the ball and the raspberry in "StillLife", the grid in "Boxes" and the left-bottom corner in "Cotton".). We also notice that the proposed *MANet* is a bit better for recovering fine details around occlusion regions. Moreover, the depth maps estimated by *MANet* have few artifacts than the *S-R4DE* and *LLF-Net*.

Real-world datasets Fig. 7.3 shows the proposed visual comparisons against the state-of-the-arts on the EPFL-Lytro dataset. The depth maps from the proposed *MANet* and *LLF-Net* still achieve the higher quality of depth maps than those from the state-of-the-art methods, especially better in recovering the details around occlusion regions (e.g., the chain link fences or empty circles). In general, the proposed *LLF-Net* is capable of retaining the more sharp boundaries than the others, and recovering depth of the smooth surface with less noise. Whereas, the proposed *MANet* does not suffer from the few black holes that occur in the *LLF-Net*.

7.3.2 Performance on Wide-baseline Datasets

To verify the effectiveness of the proposed method on wide-baseline light field datasets, we conduct experiments on the synthetic *WLF* dataset and real-world Google [26] and ULB_Unicorn [21] datasets. We compare the proposed *S-R4DE* and *LLF-Net* with recent state-of-the-art depth estimation methods, comprising of traditional light field depth estimation methods LF_OCC [55] and RPRF [61], CNN-based methods EPINET [68], EPI-Shift [70] and LBDE-E [31]. Note that EPINET [68] is originally trained on narrow-baseline datasets and fails to infer depths on wide-baseline datasets, therefore we re-trained it using their public source code [1], and denote it as EPINET_T.

Next we will show the performance comparisons, being comprised of the accuracy, adaptive input ability and qualitative (visual) comparisons.

Accuracy Comparison

Table 7.6 shows the quantitative comparisons on the four exemplar scenes from Hand-designed test set of *WLF* dataset. When compared with state-of-the-art methods, the proposed *LLF-Net* achieves the lowest bad-0.3 and bad-0.6 errors in all four scenes.

1. https://github.com/chshin10/epinet

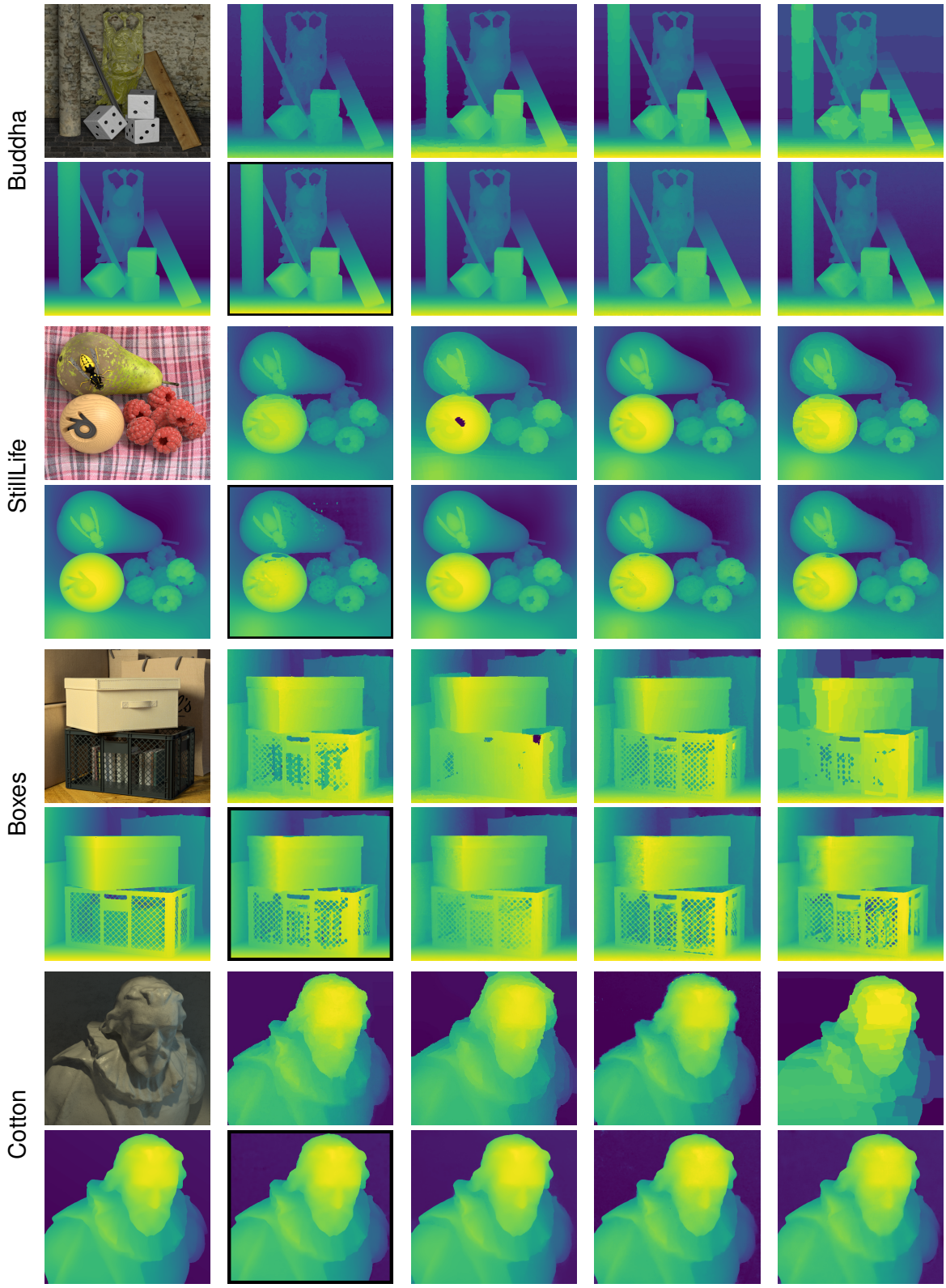


Figure 7.2: Visual comparisons of synthetic datasets. For each scene, the image from the left-top to the right-bottom corresponds to the Central view, LF_OCC, LF, *S-R4DE*, RPRF, GT, EPINET, LBDE-E, *MANet*, *LLF-Net* respectively.

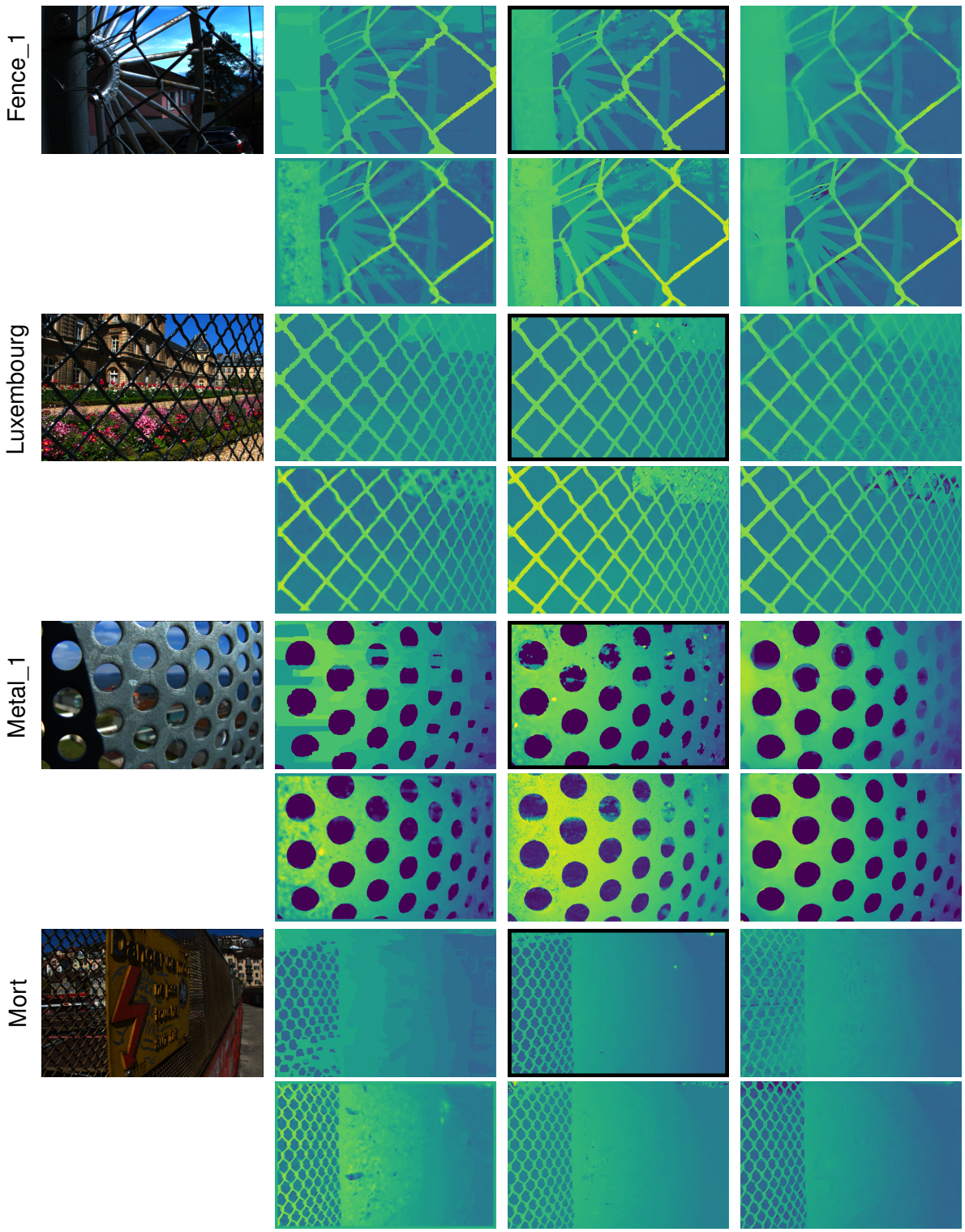


Figure 7.3: Visual comparisons of real-world datasets. For each scene, the image from the left-top to the right-bottom corresponds to the Central view, RPRF, EPINET, LFBE-E, *HFNet*, *MANet*, *LLF-Net*.

Besideds, the average of the MSE and Bad Pixel over the whole test set of *WLF* are calculated and reported in Table 7.7. It also turns out that the proposed end-to-end trained model (with the fewest parameters) far surpasses the state-of-the-arts, producing the lowest average errors in all metrics.

Table 7.6: Bad pixel error percentages of the four exemplar scenes of the *WLF* dataset against the ground truth.

Scene	Buddha2		Furniture2		Perikles		Sideboards	
	bad-0.3	bad-0.6	bad-0.3	bad-0.6	bad-0.3	bad-0.6	bad-0.3	bad-0.6
LF_OCC [55]	98.41	88.64	97.96	49.75	98.49	92.93	97.01	73.67
RPRF [61]	11.10	0.86	17.11	1.16	14.82	1.26	31.15	25.46
EPINET [68]	100	100	100	100	100	100	100	100
EPINET_T [68]	97.65	92.05	96.40	88.27	97.39	94.79	95.79	91.63
EPI-Shift [70]	22.22	4.51	22.09	6.57	48.81	9.36	50.67	38.16
LBDE-E [31]	14.01	7.72	16.92	8.18	46.20	32.78	45.89	39.23
LLF-Net	**1.44**	**0.74**	**1.93**	**1.01**	**3.30**	**0.58**	**21.17**	**14.82**

Table 7.7: Performance comparison results on the *WLF* test set. This test set comprises of the subset *Hand-designed*, containing 12 frames/scenes in total. The average errors of all frames are listed and the best performance is in bold. The quantity of parameters of CNN-based methods is in Million (M).

Method	Parameters	End-to-end trained	Hand-designed				
			mse	bad-0.15	bad-0.3	bad-0.6	bad-1
LF_OCC [55]	-	-	13.56	98.86	97.54	78.63	40.86
RPRF [61]	-	-	1.70	40.43	16.01	5.43	4.70
EPINET [68]	5.1	✓	458.13	100	100	100	100
EPINET_T [68]	5.1	✓	86.89	98.56	97.10	94.11	89.92
EPI-Shift [70]	31.6	✓	20.76	61.55	35.95	14.65	12.59
LBDE-E [31]	198.8	✗	11.12	36.86	29.02	20.86	16.09
LLF-Net	**1.8**	✓	**0.93**	**15.04**	**7.05**	**3.95**	**2.80**

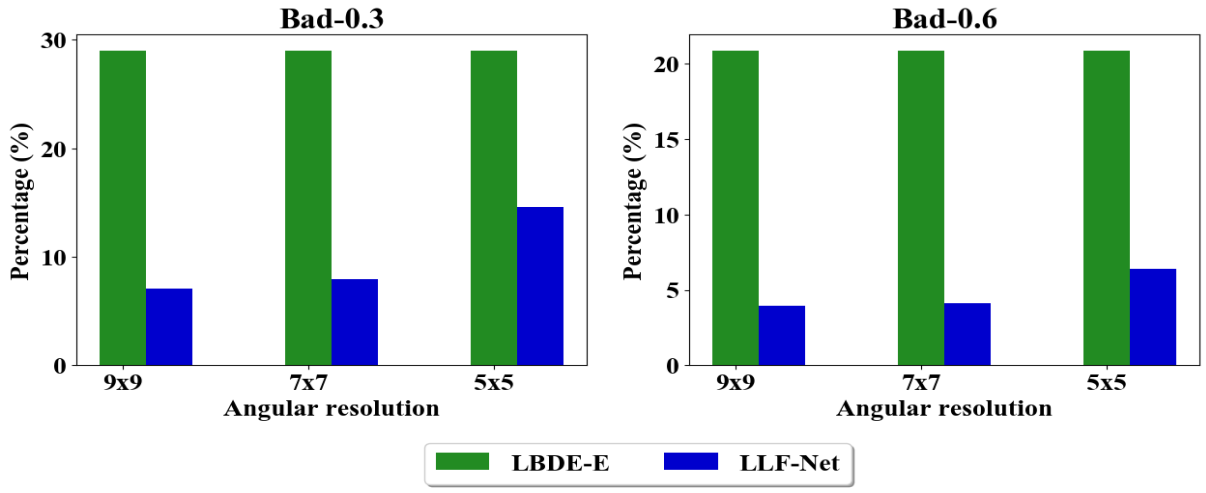


Figure 7.4: Performance comparisons results from testing the various angular light field inputs. The number in the vertical axis depicts the percentage of the bad pixels.

Adaptive Inputs Comparison

For the CNN-based light field depth estimation methods in the literature, the angular shape of inputs during inference are typically required to be same with that of the inputs used in the training stage. As with the proposed *LLF-Net*, it supports the various angular inputs thanks to the proposed cost volume using DCS fusion. We thus compare the proposed performance to the CNN-based method LBDE-E that also allows adaptive angular light field inputs (9x9, 7x7 and 5x5 light fields) during inference. Fig. 7.4 shows that when the angular resolution of light fields is lower, the performance of our model that is trained from 9x9 light field inputs gradually degrades but is still much better than LBDE-E [31].

Visual Comparison

Synthetic Dataset In Fig. 7.5, visual comparisons of the four above-mentioned scenes are given, in which each column for each scene displays the central view, the ground truth and the estimated depth maps. It is clear that our estimated depth maps are all closer to the ground truth, where the depth pixels at textureless regions and occlusion regions are recovered with the high fidelity. In contrast with the proposed *LLF-Net*, the estimated depth maps from LF_OCC are noisy, those from RPRF look over-smoothed and have quantification errors, both EPINET and EPINET_T fail to predict depths, EPI-Shift [70] and LBDE-E [31] both seem not able to handle the foreground well. Whereas, the proposed *LLF-Net* is not perfect and revealed its weak point in predicting the depth at the heavy occlusion regions (cf. the "Sideboards"), but this is similarly found in the occlusion-aware method LF_OCC, and more seriously found in all other methods.

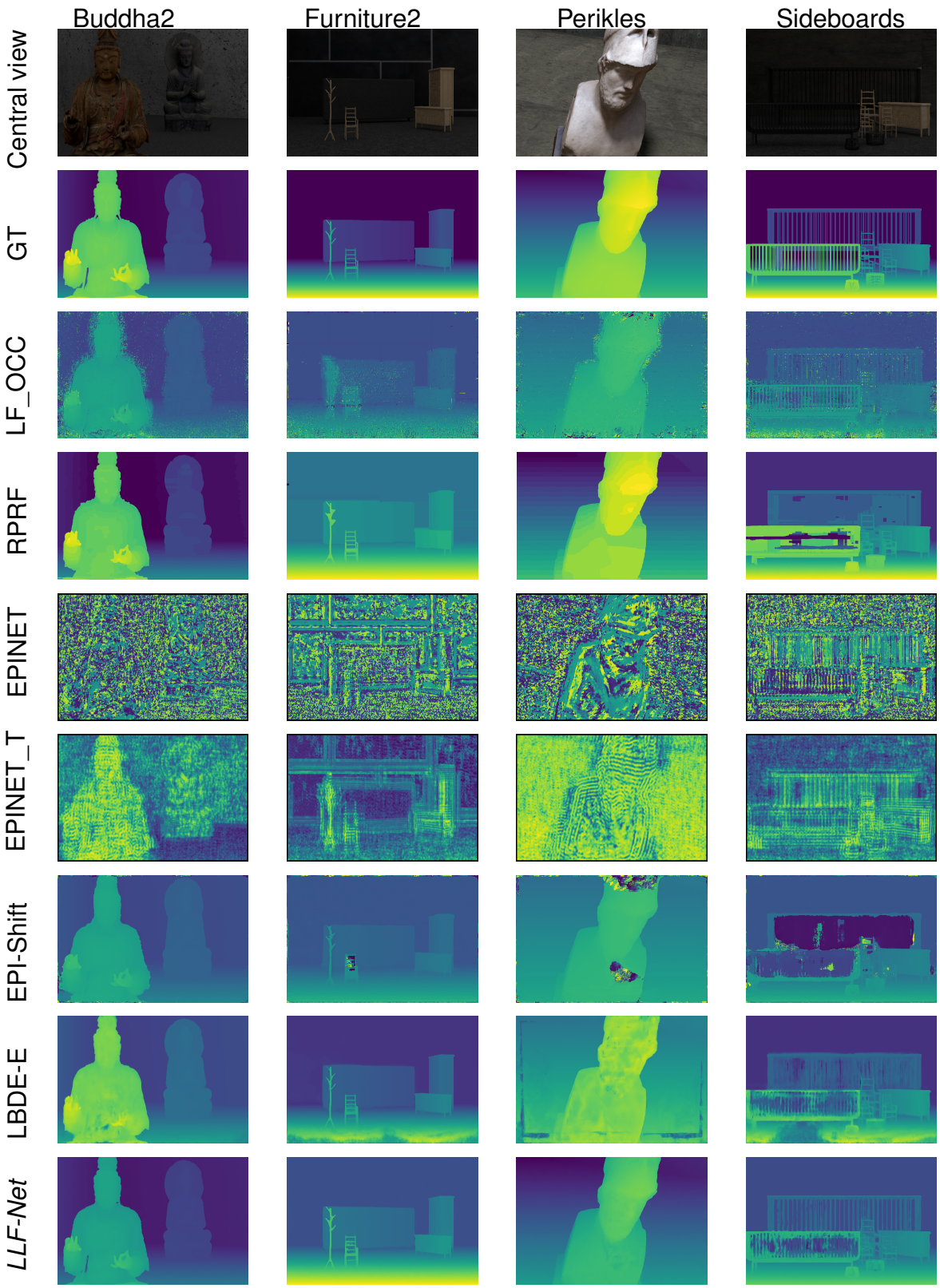


Figure 7.5: Visual comparison results of the scenes from the *WLF* dataset: the central view and ground truth disparity map are shown in the first and second row, and the other rows show the predicted depth maps from state-of-the-art respectively.

Real-world Dataset Fig. 7.6 demonstrates visual comparisons on the Google (5x5 light fields) and ULB_Unicorn (9x9 light fields) test scenes respectively. We exclude [70] for this comparison since the models that it provided only allowed 9x9 light field inputs. Though the proposed CNN *LLF-Net* is a lightweight CNN, it is capable of producing the more accurate depth maps in real-world scenes when comparing to the other CNN-based methods. Specifically, for both scenes, EPINET [68] is still not able to predict depths, similar to their results from synthetic datasets. Ours have few noticeable artifacts in the foreground than that in LBDE-E [31]. When we compare the proposed traditional method *S-R4DE* with the other traditional methods, the *S-R4DE* has fewer artifacts than LF_OCC [55] and have fewer over-smoothness issues at occlusion regions in RPRF [61]. When making comparisons between the proposed *S-R4DE* and *LLF-Net*, the performance seems similar, but the *LLF-Net* seems better in keeping sharp boundaries around occlusion regions. Besides we clearly see from the background of "Path" scene, the *LLF-Net* is able to capture more correct depths than the other methods, e.g., more depth are correctly recovered on persons.

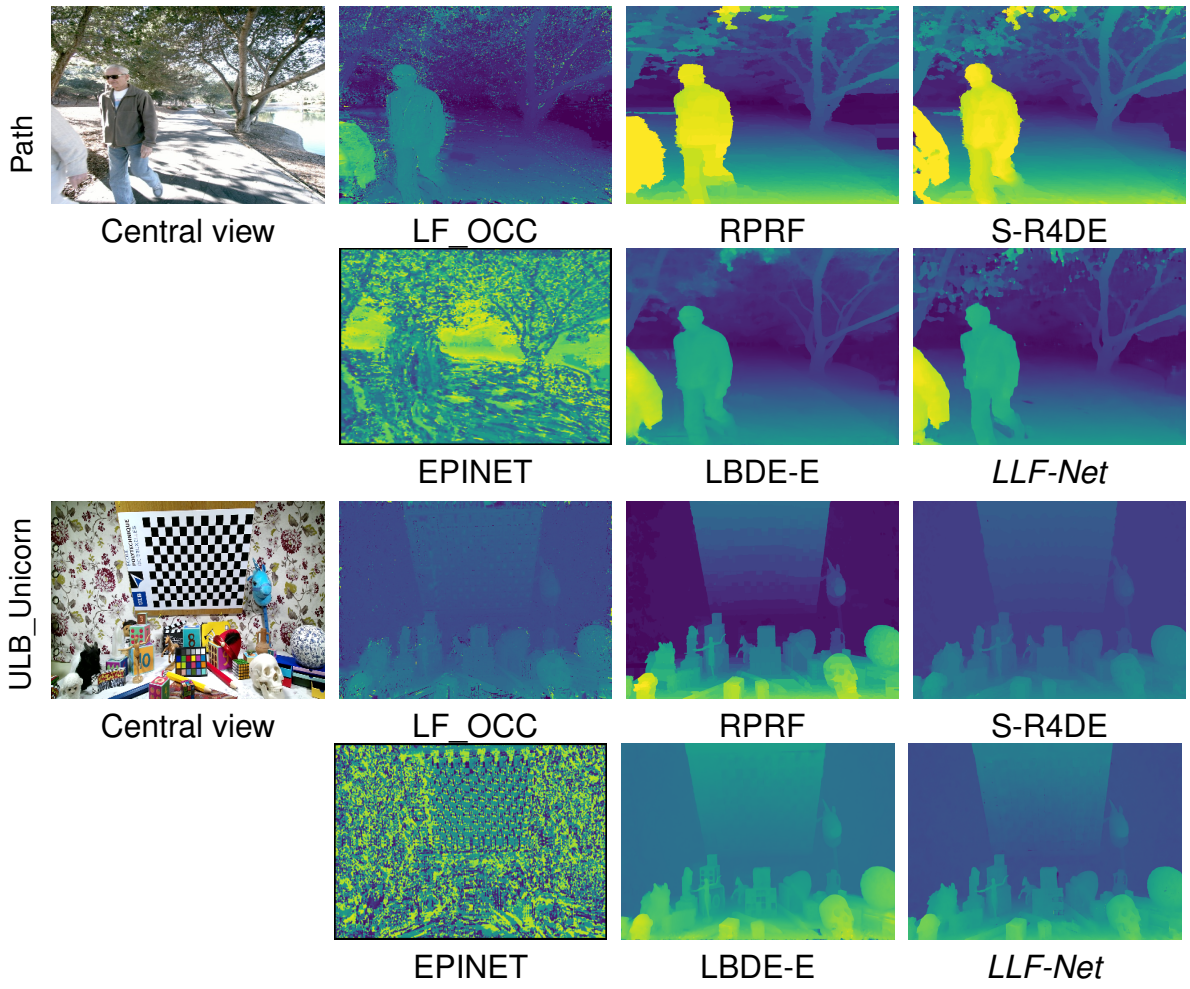
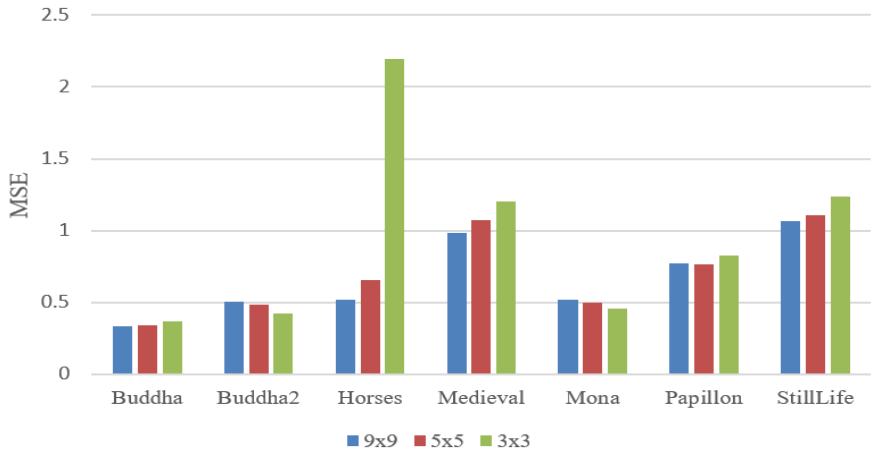


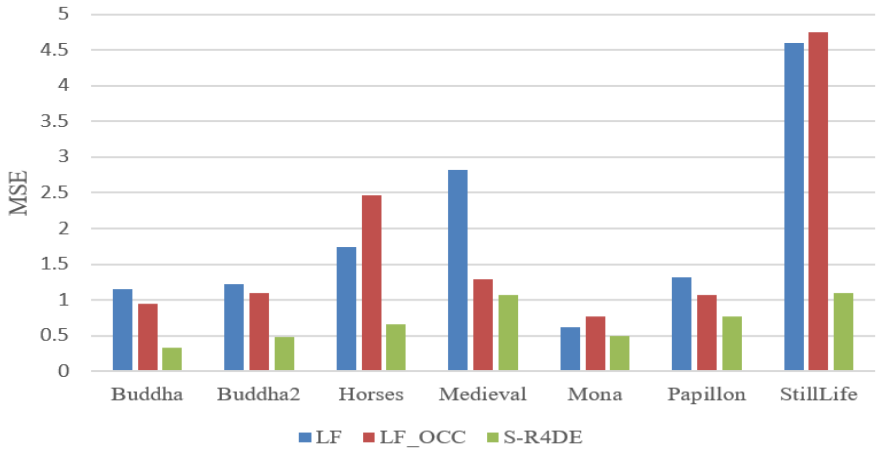
Figure 7.6: Visual comparison results of wide-baseline real-world datasets: the central view and colored disparity map are shown (best viewed in color).

7.3.3 Baseline

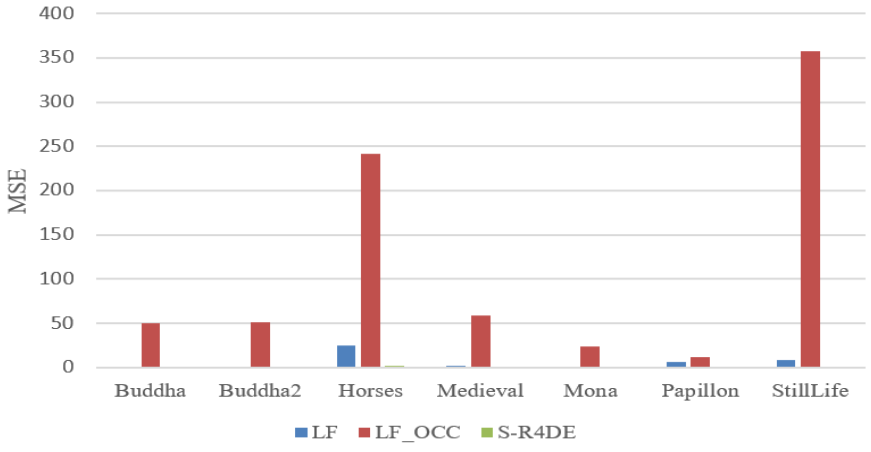
In this section, we evaluate the proposed *S-R4DE* since it was proposed being independent of the *baseline*. A part of scenes in the narrow-baseline dataset are utilized for experiments. The *density* and *baseline* are changed by skipping a multiply of 2 views from the 9x9 views in both angular directions (i.e., the 5x5 and 3x3 light field herein). The MSE is calculated for the 5x5 and 3x3 light fields and the proposed results are also compared with the state-of-the-art references (LF_OCC and LF), which are shown in Fig. 7.7. It describes that the proposed method mostly achieves the lowest errors and exhibits the robustness to the *density* or *baseline* of light fields. Fig. 7.8 illustrate the visual comparision results on the 'StillLife' scene respectively. We observe that the quality of the depth map from LF_OCC [55] degrades gradually with a smaller number of light field views. Whereas the LF [54] decreases a bit but more than that of the proposed method. Moreover, the proposed *S-R4DE* is not over-smoothed as LF [54]. Therefore the proposed *S-R4DE* is scalable to the density and baseline of the light fields.



Proposed *S-R4DE*



5x5



3x3

Figure 7.7: The MSEs of the proposed framework *S-R4DE* are compared with the state-of-the-art references on the 5x5 and 3x3 light field respectively. The lowest value means the highest accuracy.

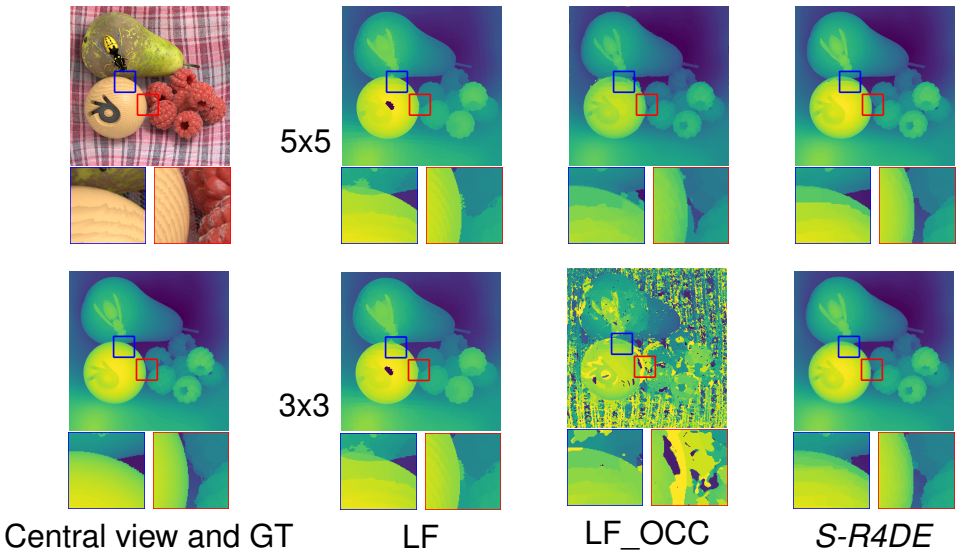


Figure 7.8: Depth estimation results on 'StillLife'. In contrast, our depth map is robust around the surface of the ball.

CHAPTER 8

CONCLUSION

We have presented new methods for depth estimation from the 3D light fields to 4D light fields, having achieved the state-of-the-art performance in a range of settings. The proposed methods have been made quantitative comparisons with previous works, and 1) experimental results on the 4D narrow-baseline datasets show that the *MANet* produces the lowest errors among all methods, achieving the 34.7% MSE gain, and surpasses by 14.5% on the bad-0.1 error, compared with the EPINET; 2) experimental results on the 4D wide-baseline datasets show that the *S-R4DE* and *LLF-Net* are capable of producing high quality depth maps. From the quantitative comparisons we learn that the *LLF-Net* far surpasses previous works on the average of the MSE and bad pixel over the whole test set of *WLF*, achieving the 45.3% MSE gain and the 56.0% bad-0.3 gain when comparing to the RPRF. Furthermore, the *MANet* and *LLF-Net* are the two most lightweight models and produce the depth map with the two lowest computational time (0.73s and 0.46s respectively on a consumer GPU) among the compared methods. Consequently, we suggest the potential readers to use the *MANet* and *LLF-Net* for the scenario that cares about the runtime, and use the *S-R4DE* and the *LLF-Net* for the scenario that the baseline of light field is uncertain or the high depth accuracy is in demand. Last but not least, we have presented a new large-scale synthetic 4D light field datasets with the wide-baseline, which can serve to the community for further training or comparing the deep learning-based models.

Next, we will make a summary of the thesis in Sec 8.1, and describe what the promising future works will be in Sec 8.2.

8.1 Summary

The thesis is focused on the depth estimation from light field images, which mainly predict the disparities (or depths) of the central view by searching the offset of corresponding points in other views. We have successively presented a new dataset, a traditional method for 3D light field depth estimation, a traditional method and three CNN-based methods for 4D light field depth estimation as follows:

In Chapter 2, a new large-scale synthetic 4D light field datasets with the wide-baseline is presented, aiming at training or comparing the potential depth estimation

methods for the light fields.

In Chapter 3, the traditional method is proposed for estimating depths from sparse sampled 3D light fields with the wide-baseline. The initial disparity map is firstly generated by the local cost calculation, and then the propagation with confidence metric and the optimization is applied to refine the initial prediction. For the initial disparity computation, a proper combination of the image decomposition into the edge and non-edge region, the relative gradient and bivariate kernel density function are utilized.

Considering the 4D light fields have the richer angular information against the textureless and occlusion issues, in Chapter 4, we extend the traditional method in Chapter 3 by taking full advantage of the more angular views in the 4D light fields. This extended method is built as an occlusion-aware scalable framework, where multiple edge cues are leveraged to improve the robustness of occlusion detection.

In Chapter 5, we put emphasis on CNN-based methods for the 4D light fields with the narrow-baseline. Two proposed CNN models are presented. The first proposed *HFNet* predicts the disparity by learning the hybrid feature representations from the Epipolar-Plane-Image and light field sub-aperture images. The second proposed *MANet* explores the multi-scale features from light field sub-aperture images, based on the idea that the high-scale features can keep more details and the low-scale features can bring in more context information.

In Chapter 6, the proposed network *LLF-Net* is motivated by the observation that existing deep-learning based networks perform well on the 4D narrow-baseline data, but not on the 4D wide-baseline data. The state-of-the-art methods are also heavyweight (with huge parameters), which are not practical. The proposed network *LLF-Net* is built by an incorporated cost volume and an attention module with very few parameters.

In Chapter 7, evaluations are firstly made on the 3D light field datasets, and the visual comparison shows that the superior depth estimation results of the proposed traditional method *R3DE* over state-of-the-art methods. Secondly, evaluations are made on the 4D narrow-baseline datasets, and the experimental results show that the proposed traditional method *S-R4DE* and CNN-based methods *HFNet*, *MANet* and *LLF-Net* achieve the state-of-the-art accuracy, in which the *MANet* and *LLF-Net* are the two most lightweight models and produce the depth map with the lowest computational overhead among the compared methods. Thirdly, the *S-R4DE* and *LLF-Net* are assessed on the 4D wide-baseline datasets, and experimental results demonstrate that the two proposed methods outperform previous works in depth accuracy. At last, the *S-R4DE* is proved to be scalable to the densities and baselines of light fields, which could attract more interests, especially the industrial applications that require small computational budgets for reconstructing accurate depths.

8.2 Future Work

Though our works have made progress toward a wider range of depth reconstruction applications from light fields, we believe that there are some promising future directions for further improvements, addressing the more challenging concerns and that are opened up from our explorations. Next, we list the promising future works in a preference order that we suggest.

Semantic cue: until now, a number of cues, including the defocus, focal stack, EPI, SCAM, boundary (or edge) cues, etc. are investigated to improve the quality of depth estimation from light fields. The so-large number of cues are indeed effective in recovering the depths at most of the everyday imaging objects or background regions. However, these cues might be invalid in the large textureless regions, e.g. the sky and the wall. The semantic cue/prior (separating the object and background) is suggested to be explored in traditional light field depth estimation (e.g., integrating this cue into the global optimization [112]) or embedding this into the deep CNNs (e.g. the proposed *MANet* or *LLF-Net*) [113, 114].

Occlusion: the light fields are advanced by the high potentials against the occlusion. The occlusion is explicitly studied in Chapters 3 and 4, and has also been studied for a long time in the traditional depth estimation methods. However, this issue is not considered in the deep learning-based methods except that the *LLF-Net* made an attempt by using the attention mechanism to handle the occlusion. The edge or boundary cues, or the specific features learned from the occlusion dataset [115] are possibly used in the future works.

Transfer learning: the CNNs proposed by the previous methods and the proposed methods are trained from scratch on the public (or proposed) light field datasets, which typically takes a long time for training (though we have reduced the training time in the proposed *LLF-Net* to the shortest time (1.6 days) among state-of-the-arts). Moreover, the scale of light field datasets is not large enough, which might pose a negative impact on the depth accuracy and the generality of the trained models. Actually, it is arduous and expensive to increase the light field data since the calibration and rectification for the real data or the 3D models for the synthetic data are extra required. Therefore, there might be a good attempt to use an advanced technique, i.e. transfer learning, as a complement for the shortcomings. This technique is capable of speeding up training on data with the similar domain, which has played an important role in many vision tasks. For the light fields, we firstly replace the low-level and/or the middle level image features with the features learned/pre-trained from a large amount of data samples, e.g. pre-trained from a basic task *image recognition*, and then train the model on the task of depth estimation from light fields.

Training CNNs from real-world datasets: existing CNN models train the network

mainly from synthetic light field datasets with the ground truth disparities, and infer the real-world datasets with the trained models. However, there might exist the gaps between the synthetic and real-world scenes even though the existing synthetic datasets are designed to imitate various challenging cases in real-world scenes. As a result, the gaps possibly influence the generalization accuracy. Training the deep network from real-world datasets directly is a very promising way, however, it is hard to obtain the ground truths for the real-world light field datasets. Thus, training on the real world datasets in an unsupervised manner or by a generative model is a potential future work.

Temporal information and consistency: most of light field depth estimation methods perform well on the light field image datasets, which only consider the spatial information. However, the light field video datasets are of importance and contain motions, but the temporal information is rarely investigated and the consistency among the depth maps of the adjacent frames is less paid attention to. Further research for adopting the temporal information and maintaining the consistency among the adjacent frames is a potential interesting field.

Metrics: in the manuscript and previous works, the objective metrics, including the mean square error (MSE) and bad pixel metrics, are often utilized for assessing the quality of depth maps. Actually, when the depth map has a very low MSE or bad pixel percentage, the quality is indeed very high with few/negligible artifacts, and the artifacts will be gone if either of them is much lower. However, there exists some scenes that are not so well estimated in all existing works that the MSE or bad pixels are not low. Though the value is high, what the artifacts are and what cause the artifacts can not be unknown from the reported value. For better assessment or recommendation of the algorithms, the metric taking into account the Human Visual System characteristics or the requirement of the potential application needs to be addressed in the future.

Real-time depth estimation from light fields: although the proposed CNNs for light field depth estimation achieve high efficiency in the literature, the existed methods still suffer from computational burdens, e.g., around two frames per second (by the proposed *LLF-Net*), which is far from reaching the real time processing. The reduction of the computational complexity is a promising research direction which could meet the needs of more realistic real-world applications.

APPENDICES

Graph cuts

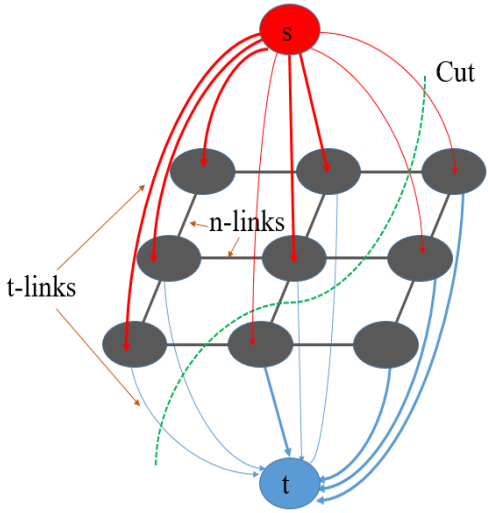


Figure 8.1: Example of a cut on a graph G

The Markov random field-based energy function in Eq. 4.6 is solved using graph cuts. With respect to the graph cuts, this technique is proposed by [116] for the first time to solve binary label optimization problem, in which the energy minimization is solved by computing the minimum cut or the maximum flow in a directed graph.

Let $G =< V, E >$ be a directed graph, which consists of two terminals (the source s and the sink t), a set of vertices V (pixels in the manuscript), and edges E between the two vertices. For edges, there exists two types of links: n-links and t-links, where the n-links (see black lines in Fig. 8.1) indicate the edges between the neighboring pixels and the t-links (see red and blue lines in Fig. 8.1) indicate the edges between the pixels and terminals (labels). The cost of a n-link represents the penalty between the neighboring pixels, corresponding to the smoothness term in the energy function (in our case, this is formulated as the weighted label difference, cf. Eq. 4.8 and Eq. 4.9). The cost of a t-link represents the penalty of one candidate label assigned to the pixel, corresponding to the data term (this is formulated as the aggregated local cost, cf. Eq. 4.1 and Eq. 4.2) in the energy function.

To minimize the energy is done by finding the cut with the smallest cost for the minimum cut on the graph. For a (s-t cut), the vertices will be cut into two disjoint sets

(S and T), as is shown in Fig. 8.1. The cost of the cut is computed as the sum of the weights of the edges that go from the source to the sink. Note that the minimum cut can be alternatively computed by the efficient maximum flow algorithm since it could be equivalent to the maximum flow according to the theorem Ford and Fulkerson [117].

Here, the alpha-expansion [118], one of the effective expansion move algorithms, is iteratively used for minimizing the multi-label (or disparity) energy Eq. 4.6. Though it is an approximate solution, a strong local minimum could be found. Specifically, for a candidate label (disparity) α in a fixed order, a single α-expansion moves from this label to another candidate label, and if the decrease of the energy occurs, then this label is assigned to the new candidate label; otherwise not. This similar step is repeated for all candidate labels, and if there is no α move that decreases the energy, the computation will be terminated.

ACRONYMS

BF	Bilateral Filtering.
BKDE	Bivariate kernel density estimation.
BN	Bilateral normalization.
CNN	Convolutional neural network.
EPI	Epipolar plane image.
GF	Guided Filtering.
GT	Ground Truth.
KDE	Kernel density estimation.
MAE	Mean Absolute Error.
MRF	Markov Random Field.
MSE	Mean Square Error.
MVS	Multi-view Stereo.
OBD	Occlusion Boundary Detection.
OPD	Occluded Pixel Detection.
	.
SCAM	Surface camera.
SURF	Speeded up robust features.

LIST OF FIGURES

List of Tables

LIST OF PUBLICATION

- Yan Li and Gauthier Lafruit, 2016, December. Convergent multi-view geometric error correction with pseudo-inverse projection homography. In 2016 International Conference on 3D Imaging (IC3D) (pp 1-8). IEEE.
- Yan Li and Gauthier Lafruit, 2017, June. Robust disparity estimation on sparse sampled light field images. In 2017 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON) (pp. 1-4). IEEE.
- Yan Li and Gauthier Lafruit, 2018. Scalable light field disparity estimation with occlusion detection. Journal of WSCG , 26 (2), pp.66-75.
- Yan Li, Lu Zhang, Qiong Wang and Gauthier Lafruit, 2020. MANet: Multi-scale aggregated network for light field depth estimation. In 2020 ICASSP. IEEE.
- Yan Li, Gauthier Lafruit, “View Synthesis on Compressed Big Buck Bunny,” MPEG 114th meeting, 2016
- Yule Sun, Yan Li, Qing Wang, Ang Lu, Lu Yu, Krzysztof Wegner, Gauthier Lafruit, “Software for SMV and FN sweeping subjective tests”, MPEG 114th meeting, 2016
- Lode Jorissen, Patrick Goorts, Yan Li, Gauthier Lafruit, “Soccer Light Field Interpolation Applied on Compressed Data”, MPEG 114th meeting, 2016
- Daniele Bonatto, Tim Lenertz, Ségolène Rogge, Yan Li, Arnaud Schenkel, Gauthier Lafruit, “ULB High Density 2D Camera Array data set, version 1”, MPEG 118th meeting, 2017.

BIBLIOGRAPHY

[1] Naokazu Yokoya, Takeshi Shakunaga, and Masayuki Kanbara. “Passive range sensing techniques: Depth from images”. In: *IEICE Transactions on Information and Systems* 82.3 1999, pp. 523–533.

[2] Brian Curless. “Overview of active vision techniques”. In: *Proceedings of the SIGGRAPH Course on 3D Photography* 1999.

[3] Martial Hebert. “Active and passive range sensing for robotics”. In: *IEEE International Conference on Robotics and Automation*. Vol. 1. IEEE. 2000, pp. 102–110.

[4] Sreenivasa Kumar Mada, Melvyn L Smith, Lyndon N Smith, and Prema Sagar Midha. “Overview of passive and active vision techniques for hand-held 3D data acquisition”. In: *Proceedings of the Optical Metrology, Imaging, and Machine Vision*. Vol. 4877. International Society for Optics and Photonics. 2003, pp. 16–27.

[5] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[6] Daniel Scharstein and Richard Szeliski. “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms”. In: *International Journal of Computer Vision* 47.1-3 2002, pp. 7–42.

[7] Yasutaka Furukawa, Carlos Hernández, et al. “Multi-view stereo: A tutorial”. In: *Foundations and Trends® in Computer Graphics and Vision* 9.1-2 2015, pp. 1–148.

[8] Sizhang Dai and Weibing Huang. “A-TVSNet: Aggregated Two-View Stereo Network for Multi-View Stereo Depth Estimation”. In: *CoRR* abs/2003.00711 2020.

[9] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. “MVSNet: Depth Inference for Unstructured Multi-view Stereo”. In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VIII*. 2018, pp. 785–801.

[10] Marc Levoy and Pat Hanrahan. “Light field rendering”. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM. 1996, pp. 31–42.

[11] Edward H Adelson, James R Bergen, et al. *The plenoptic function and the elements of early vision*. Vol. 2. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[12] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. "The lumigraph". In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 43–54.

[13] Sumit Shekhar, Shida Kunz Beigpour, Matthias Ziegler, Michał Chwesiuk, Dawid Paleń, Karol Myszkowski, Joachim Keinert, Radosław Mantiuk, and Piotr Didyk. "Light-field intrinsic dataset". In: *British Machine Vision Conference 2018 (BMVC)*. British Machine Vision Association. 2018.

[14] Gaochang Wu, Belen Masia, Adrian Jarabo, Yuchen Zhang, Liangyong Wang, Qionghai Dai, Tianyou Chai, and Yebin Liu. "Light field image processing: An overview". In: *IEEE Journal of Selected Topics in Signal Processing* 11.7 2017, pp. 926–954.

[15] Edward H Adelson and John Y. A. Wang. "Single lens stereo with a plenoptic camera". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2 1992, pp. 99–106.

[16] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, Pat Hanrahan, et al. "Light field photography with a hand-held plenoptic camera". In: *Computer Science Technical Report CSTR* 2.11 2005, pp. 1–11.

[17] Andrew Lumsdaine and Todor Georgiev. "The focused plenoptic camera". In: *Proceedings of the IEEE International Conference on Computational Photography*. IEEE. 2009, pp. 1–8.

[18] Christian Perwass and Lennart Wietzke. "Single lens 3D-camera with extended depth-of-field". In: *Human Vision and Electronic Imaging XVII*. Vol. 8291. International Society for Optics and Photonics. 2012, p. 829108.

[19] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus H Gross. "Scene reconstruction from high spatio-angular resolution light fields." In: *ACM Transactions on Graphics (TOG)* 32.4 2013, pp. 73–1.

[20] Matthias Ziegler, Ron op het Veld, Joachim Keinert, and Frederik Zilly. "Acquisition system for dense lightfield of large scenes". In: *IEEE 3DTV Conference: The True Vision-Capture, mission and Display of 3D Video*. IEEE. 2017, pp. 1–4.

[21] Daniele Bonatto, Arnaud Schenkel, Tim Lenertz, Yan Li, and Gauthier Lafruit. "ULB high density 2D/3D camera array data set version 2". In: *ISO/IEC JTC1/SC 29/WG11 MPEG* M41083 2017.

[22] Takeshi Naemura and Hiroshi Harashima. “Real-time video-based rendering for augmented spatial communication”. In: *Visual Communications and Image Processing’99*. Vol. 3653. International Society for Optics and Photonics. 1998, pp. 620–631.

[23] Bennett S Wilburn, Michal Smulski, Hsiao-Heng Kelin Lee, and Mark A Horowitz. “Light field video camera”. In: *Media Processors 2002*. Vol. 4674. International Society for Optics and Photonics. 2001, pp. 29–36.

[24] Jason C Yang, Matthew Everett, Chris Buehler, and Leonard McMillan. “A real-time distributed light field camera.” In: *Rendering Techniques* 2002 2002, pp. 77–86.

[25] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. “High performance imaging using large camera arrays”. In: 24.3 2005, pp. 765–776.

[26] Eric Penner and Li Zhang. “Soft 3D reconstruction for view synthesis”. In: *ACM Transactions on Graphics (TOG)* 36.6 2017, p. 235.

[27] Neus Sabater, Guillaume Boisson, Benoit Vandame, Paul Kerbiriou, Frederic Babon, Matthieu Hog, Remy Gendrot, Tristan Langlois, Olivier Bureller, Arno Schubert, et al. “Dataset and pipeline for multi-view light-field video”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 30–40.

[28] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 4040–4048.

[29] Sven Wanner, Stephan Meister, and Bastian Goldluecke. “Datasets and benchmarks for densely sampled 4d light fields.” In: *VMV*. Citeseer. 2013, pp. 225–226.

[30] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. “A dataset and evaluation methodology for depth estimation on 4d light fields”. In: *Asian Conference on Computer Vision*. Springer. 2016, pp. 19–34.

[31] Jinglei Shi, Xiaoran Jiang, and Christine 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.

[32] Martin Rerabek and Touradj Ebrahimi. “New light field image dataset”. In: *8th International Conference on Quality of Multimedia Experience (QoMEX)*. EPFL-CONF-218363. 2016.

[33] Xiaoran Jiang, Mikaël Le Pendu, Reuben A Farrugia, and Christine Guillemot. “Light field compression with homography-based low-rank approximation”. In: *IEEE Journal of Selected Topics in Signal Processing* 11.7 2017, pp. 1132–1145.

[34] Jae Woo Kim Howook Jang Do Hyung Kim Seong-Jun Bae Seongjin Park. “Camera Array based Windowed 6DoF Moving Picture Contents”. In: vol. 42542. 2018.

[35] Arnaud Schenkel, Daniele Bonatto, Sarah Fachada, Henry-Louis Guillaume, and Gauthier Lafruit. “Natural Scenes Datasets for exploration in 6DoF Navigation”. In: *International Conference on 3D Immersion (IC3D)*. IEEE. 2018, pp. 1–8.

[36] Xiao-Ran Jiang, Jing-Lei Shi, and Christine Guillemot. “A Learning Based Depth Estimation Framework for 4D Densely and Sparsely Sampled Light Fields”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing*. 2019, pp. 2257–2261.

[37] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. “FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 1647–1655.

[38] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. “Training deep networks with synthetic data: Bridging the reality gap by domain randomization”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2018, pp. 969–977.

[39] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. “Shapenet: An information-rich 3d model repository”. In: *arXiv preprint arXiv:1512.03012* 2015.

[40] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. “High-resolution stereo datasets with subpixel-accurate ground truth”. In: *German Conference on Pattern Recognition*. Springer. 2014, pp. 31–42.

[41] Moritz Menze and Andreas Geiger. “Object scene flow for autonomous vehicles”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 3061–3070.

[42] Sven Wanner and Bastian Goldluecke. “Variational Light Field Analysis for Disparity Estimation and Super-Resolution”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.3 2014, pp. 606–619.

[43] Haiting Lin, Can Chen, Sing Bing Kang, and Jingyi Yu. “Depth Recovery from Light Field Using Focal Stack Symmetry”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3451–3459.

[44] Vladimir Kolmogorov and Ramin Zabih. “Multi-camera scene reconstruction via graph cuts”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2002, pp. 82–96.

[45] Pedro F Felzenszwalb and Daniel P Huttenlocher. “Efficient belief propagation for early vision”. In: *International journal of computer vision* 70.1 2006, pp. 41–54.

[46] Thomas Pock, Daniel Cremers, Horst Bischof, and Antonin Chambolle. “Global solutions of variational models with convex regularization”. In: *SIAM Journal on Imaging Sciences* 3.4 2010, pp. 1122–1145.

[47] Zhan Yu, Xinqing Guo, Haibing Lin, Andrew Lumsdaine, and Jingyi Yu. “Line assisted light field triangulation and stereo matching”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 2792–2799.

[48] Huijin Lv, Kaiyu Gu, Yongbing Zhang, and Qionghai Dai. “Light field depth estimation exploiting linear structure in EPI”. In: *IEEE International Conference on Multimedia & Expo Workshops*. IEEE. 2015, pp. 1–6.

[49] Xiangsheng Huang, Ziling Huang, Ming Lu, Pengcheng Ma, and Weili Ding. “A semi-global matching method for large-scale light field images”. In: *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE. 2016, pp. 1646–1650.

[50] Lode Jorissen, Patrik Goorts, Gauthier Lafruit, and Philippe Bekaert. “Multi-view wide baseline depth estimation robust to sparse input sampling”. In: *3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video*. IEEE. 2016, pp. 1–4.

[51] Sven Wanner and Bastian Goldluecke. “Globally consistent depth labeling of 4D light fields”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2012, pp. 41–48.

[52] Michael W. Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. “Depth from Combining Defocus and Correspondence Using Light-Field Cameras”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 673–680.

[53] Can Chen, Haiting Lin, Zhan Yu, Sing Bing Kang, and Jingyi Yu. “Light field stereo matching using bilateral statistics of surface cameras”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1518–1525.

[54] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. “Accurate depth map estimation from a lenslet light field camera”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 1547–1555.

[55] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. “Occlusion-aware depth estimation using light-field cameras”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015, pp. 3487–3495.

[56] Shuo Zhang, Hao Sheng, Chao Li, Jun Zhang, and Zhang Xiong. “Robust depth estimation for light field via spinning parallelogram operator”. In: *Computer Vision and Image Understanding* 145 2016, pp. 148–159.

[57] Hao Zhu and Qing Wang. “An efficient anti-occlusion depth estimation using generalized EPI representation in light field”. In: *Optoelectronic Imaging and Multimedia Technology IV*. Vol. 10020. International Society for Optics and Photonics. 2016, p. 1002008.

[58] In Kyu Park and Kyoung Mu Lee. “Robust Light Field Depth Estimation Using Occlusion-Noise Aware Data Costs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.10 2018, pp. 2484–2497.

[59] Julia Navarro and Antoni Buades. “Robust and Dense Depth Estimation for Light Field Images”. In: *IEEE Transactions on Image Processing* 26.4 2017, pp. 1873–1886.

[60] Hao Zhu, Qing Wang, and Jingyi Yu. “Occlusion-Model Guided Antiocclusion Depth Estimation in Light Field”. In: *IEEE Journal of Selected Topics in Signal Processing* 11.7 2017, pp. 965–978.

[61] Chao-Tsung Huang. “Empirical Bayesian Light-Field Stereo Matching by Robust Pseudo Random Field Modeling”. In: vol. 41. 3. 2019, pp. 552–565.

[62] Kazu Mishiba. “Fast Depth Estimation for Light Field Cameras”. In: *IEEE Transactions on Image Processing* 29 2020, pp. 4232–4242.

[63] Josef Bigun. *Optimal orientation detection of linear symmetry*. 1987.

[64] Stefan Heber and Thomas Pock. “Convolutional networks for shape from light field”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 3746–3754.

[65] Stefan Heber, Wei Yu, and Thomas Pock. “Neural EPI-Volume Networks for Shape from Light Field”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 2271–2279.

[66] Yaoxiang Luo, Wenhui Zhou, Junpeng Fang, Linkai Liang, Hua Zhang, and Guojun Dai. “EPI-Patch Based Convolutional Neural Network for Depth Estimation on 4D Light Field”. In: *International Conference on Neural Information Processing*. Springer. 2017, pp. 642–652.

[67] Mingtao Feng, Yaonan Wang, Jian Liu, Liang Zhang, Hasan FM Zaki, and Ajmal Mian. “Benchmark Dataset and Method for Depth Estimation from Light Field Images”. In: *IEEE Transactions on Image Processing* 2018.

[68] Changha Shin, Hae-Gon Jeon, Youngjin Yoon, In So Kweon, and Seon Joo Kim. “EPINET: A Fully-Convolutional Neural Network Using Epipolar Geometry for Depth from Light Field Images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.

[69] Wenhui Zhou, Enci Zhou, Yuxiang Yan, Lili Lin, and Andrew Lumsdaine. “Learning Depth Cues from Focal Stack for Light Field Depth Estimation”. In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 1074–1078.

[70] Titus Leistner, Hendrik Schilling, Radek Mackowiak, Stefan Gumhold, and Carsten Rother. “Learning to Think Outside the Box: Wide-Baseline Light Field Depth Estimation with EPI-Shift”. In: *International Conference on 3D Vision*. IEEE. 2019, pp. 249–257.

[71] Kristian Bredies, Karl Kunisch, and Thomas Pock. “Total generalized variation”. In: *SIAM Journal on Imaging Sciences* 3.3 2010, pp. 492–526.

[72] Antonin Chambolle and Thomas Pock. “A first-order primal-dual algorithm for convex problems with applications to imaging”. In: *Journal of mathematical imaging and vision* 40.1 2011, pp. 120–145.

[73] Jonathan Long, Evan Shelhamer, and Trevor Darrell. “Fully convolutional networks for semantic segmentation”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.

[74] Lee-Kang Liu, Stanley H Chan, and Truong Q Nguyen. “Depth reconstruction from sparse samples: Representation, algorithm, and sampling”. In: *IEEE Transactions on Image Processing* 24.6 2015, pp. 1983–1996.

[75] Di Lin, Guangyong Chen, Daniel Cohen-Or, Pheng-Ann Heng, and Hui Huang. “Cascaded Feature Network for Semantic Segmentation of RGB-D Images”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1320–1328.

[76] Robert C Bolles, H Harlyn Baker, and David 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.

[77] Carlo Tomasi and Roberto Manduchi. “Bilateral filtering for gray and color images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 1998, pp. 839–846.

[78] Xiaozhou Zhou and Pierre Boulanger. “Radiometric invariant stereo matching based on relative gradients”. In: *IEEE International Conference on Image Processing*. IEEE. 2012, pp. 2989–2992.

[79] Zhengguo Li, Jinghong Zheng, Zijian Zhu, Wei Yao, and Shiqian Wu. “Weighted guided image filtering”. In: *IEEE Transactions on Image Processing* 24.1 2015, pp. 120–129.

[80] Kaiming He, Jian Sun, and Xiaoou Tang. “Guided image filtering”. In: *Proceedings of the European Conference on Computer Vision*. Springer. 2010, pp. 1–14.

[81] Asmaa Hosni, Christoph Rhemann, Michael Bleyer, Carsten Rother, and Margrit Gelautz. “Fast cost-volume filtering for visual correspondence and beyond”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.2 2013, pp. 504–511.

[82] Anat Levin, Dani Lischinski, and Yair Weiss. “A closed-form solution to natural image matting”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30.2 2008, pp. 228–242.

[83] Kaiming He, Jian Sun, and Xiaoou Tang. “Fast matting using large kernel matting laplacian matrices”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2010, pp. 2165–2172.

[84] Jianqiao Li, Minlong Lu, and Ze-Nian Li. “Continuous depth map reconstruction from light fields”. In: *IEEE Transactions on Image Processing* 24.11 2015, pp. 3257–3265.

[85] David Stutz, Alexander Hermans, and Bastian Leibe. “Superpixels: An evaluation of the state-of-the-art”. In: *Computer Vision and Image Understanding* 166 2018, pp. 1–27.

[86] John Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 8.6 1986, pp. 679–698.

[87] Ashutosh Saxena, Sung H. Chung, and Andrew Y. Ng. “Learning Depth from Single Monocular Images”. In: *Advances in Neural Information Processing Systems*. 2005, pp. 1161–1168.

[88] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention*. 2015, pp. 234–241.

[89] Alex Kendall, Hayk Martirosyan, Saumitro Dasgupta, Peter Henry, Ryan Kennedy, Abraham Bachrach, and Adam Bry. “End-To-End Learning of Geometry and Context for Deep Stereo Regression”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 66–75.

[90] Jia-Ren Chang and Yong-Sheng Chen. “Pyramid Stereo Matching Network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 5410–5418.

[91] Stepan Tulyakov, Anton Ivanov, and François Fleuret. “Practical Deep Stereo (PDS): Toward applications-friendly deep stereo matching”. In: *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada.* 2018, pp. 5875–5885.

[92] Xiaoyang Guo, Kai Yang, Wukui Yang, Xiaogang Wang, and Hongsheng Li. “Group-wise correlation stereo network”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3273–3282.

[93] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 770–778.

[94] Suyog Dutt Jain, Bo Xiong, and Kristen Grauman. “FusionSeg: Learning to Combine Motion and Appearance for Fully Automatic Segmentation of Generic Objects in Videos”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2117–2126.

[95] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. “Hidden two-stream convolutional networks for action recognition”. In: *arXiv preprint arXiv:1704.00389* 2017.

[96] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. “Fast End-to-End Trainable Guided Filter”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1838–1847.

[97] Sergey Ioffe and Christian Szegedy. “Batch normalization: accelerating deep network training by reducing internal covariate shift”. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning-Volume 37*. JMLR. org. 2015, pp. 448–456.

[98] Tijmen Tieleman and Geoffrey Hinton. “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural networks for machine learning* 4.2 2012, pp. 26–31.

[99] Lipeng Si and Qing Wang. “Dense Depth-Map Estimation and Geometry Inference from Light Fields via Global Optimization”. In: *13th Asian Conference on Computer Vision*. 2016, pp. 83–98.

[100] Michael Strecke, Anna Alperovich, and Bastian Goldluecke. “Accurate Depth and Normal Maps from Occlusion-Aware Focal Stack Symmetry”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 2529–2537.

[101] Yan Li and Gauthier Lafruit. “Scalable light field disparity estimation with occlusion detection”. In: *Journal of WSCG* 26.2 2018, pp. 66–75.

[102] Jie Chen, Junhui Hou, Yun Ni, and Lap-Pui Chau. “Accurate Light Field Depth Estimation With Superpixel Regularization Over Partially Occluded Regions”. In: *IEEE Transactions on Image Processing* 27.10 2018, pp. 4889–4900.

[103] Hendrik Schilling, Maximilian Diebold, Carsten Rother, and Bernd Jähne. “Trust Your Model: Light Field Depth Estimation With Inline Occlusion Handling”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 4530–4538.

[104] Wenhui Zhou, Linkai Liang, Hua Zhang, Andrew Lumsdaine, and Lili Lin. “Scale and Orientation Aware EPI-Patch Learning for Light Field Depth Estimation”. In: *IEEE International Conference on Pattern Recognition*. 2018, pp. 2362–2367.

[105] Wen-Hui Zhou, Enci Zhou, Yu-xiang Yan, and Li-li Lin. “Learning Depth Cues from Focal Stack for Light Field Depth Estimation”. In: *IEEE International Conference on Image Processing*. 2019, pp. 16–20.

[106] Łukasz Dąbała, Matthias Ziegler, Piotr Didyk, Frederik Zilly, Joachim Keinert, Karol Myszkowski, Przemyslaw Rokita, and Tobias Ritschel. “Efficient Multi-image Correspondences for On-line Light Field Video Processing”. In: *Computer Graphics Forum*. Vol. 35. 7. Wiley-Blackwell. 2016, pp. 401–410.

[107] Aleksandra Chuchvara, Attila Barsi, and Atanas Gotchev. “Fast and Accurate Depth Estimation from Sparse Light Fields”. In: *arXiv preprint arXiv:1812.06856* 2018.

[108] Xiaoran Jiang, Mikaël Le Pendu, and Christine Guillemot. “Depth Estimation with Occlusion Handling from a Sparse Set of Light Field Views”. In: *IEEE International Conference on Image Processing*. 2018, pp. 634–638.

[109] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. “Learning a discriminative feature network for semantic segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1857–1866.

[110] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. “Image super-resolution using very deep residual channel attention networks”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 286–301.

[111] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek Gordon Murray, Benoit Steiner, Paul A. Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. “TensorFlow: A System for Large-Scale Machine Learning”. In: *12th USENIX Symposium on Operating Systems Design and Implementation*. 2016, pp. 265–283.

[112] Fatma Guney and Andreas Geiger. “Displets: Resolving stereo ambiguities using object knowledge”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 4165–4175.

[113] Guorun Yang, Hengshuang Zhao, Jianping Shi, Zhidong Deng, and Jiaya Jia. “Segstereo: Exploiting semantic information for disparity estimation”. In: *Proceedings of the European Conference on Computer Vision*. 2018, pp. 636–651.

[114] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. “Semantic Stereo Matching with Pyramid Cost Volumes”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 7484–7493.

[115] Junhwa Hur and Stefan Roth. “Iterative residual refinement for joint optical flow and occlusion estimation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 5754–5763.

[116] Dorothy M Greig, Bruce T Porteous, and Allan H Seheult. “Exact maximum a posteriori estimation for binary images”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 51.2 1989, pp. 271–279.

[117] Lester Randolph Ford Jr and Delbert Ray Fulkerson. *Flows in networks*. Princeton university press, 2015.

[118] Yuri Boykov, Olga Veksler, and Ramin Zabih. “Fast approximate energy minimization via graph cuts”. In: *IEEE Transactions on pattern analysis and machine intelligence* 23.11 2001, pp. 1222–1239.