Universidad Zaragoza
1542

Master's Thesis

Single-View Depth from Focused Plenoptic Cameras

Profundidad Monocular con Cámara Plenóptica Enfocada

Author

Blanca Lasheras Hernández

Supervisors

Javier Civera Sancho

Klaus Strobl

ESCUELA DE INGENIERÍA Y ARQUITECTURA

2024

TRABAJOS DE FIN DE GRADO / FIN DE MÁSTER

Escuela de Ingeniería y Arquitectura Universidad Zaragoza

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./Dª. BLANCA LASHERAS HERNÁNDEZ,

con nº de DNI 73026898N en aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster en Robótica, Gráficos y Visión por Computador, (Título del Trabajo) Single-View Depth from Focused Plenoptic Cameras

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 26 de junio de 2024

Fdo: Firmado digitalmente por 5e88a4a3-e13f-42ac-94d4-e0d7a25eed04
Fecha: 2024.06.26 13:16:01 +02'00'

Abstract

In recent years, the research progress in computer vision has boosted the capabilities of machines for interpreting visual data, thereby expanding the complexity and range of tasks that robots could perform in fields such as autonomous driving, medicine, and industrial automation. A principal facet of computer vision is depth estimation, crucial for enabling robots to perceive, navigate, and interact with their environment in an effective and safe manner. Traditional setups, like stereo or multi-camera, face challenges such as calibration intricacies and computational and hardware complexity. Further, their accuracy is limited by the baseline between the cameras. Monocular depth estimation, thus using a single camera, offers a more compact alternative but is however limited by the unobservability of the scale.

Light field imaging technologies represent a promising solution to the above issues by capturing both the intensity and direction of light rays not only through the main lens, but also through a large number of microlenses placed within the camera. By these means, depth in front of the camera can be measured owing to depth-dependent refraction at the main lens. Despite their potential, there are limited studies exploring their application to single-view dense depth estimation. This scarcity can be attributed to several factors. The technology remains relatively costly and inaccessible for its widespread adoption, leading to a lack of datasets suitable for training deep neural networks. As a consequence, few projects have used light field imaging for depth estimation, and existing efforts often rely on outdated iterations of the technology. Furthermore, the lack of an open-source geometrical model impedes the development of model-based estimation.

This thesis explores the potential of focused plenoptic cameras for single-view depth estimation using learning-based methods. The proposed approach integrates techniques from image processing, deep learning, and scale alignment achieved through foundational models and robust statistics, to generate dense metric depth maps.

To support this approach, a novel real-world dataset of light field images with stereo depth labels was generated, addressing a current gap in existing resources. Experimental results demonstrate that the developed pipeline can reliably produce accurate metric depth predictions, setting a foundation for further research in this domain.

Acknowledgements

"If you have never wept bitter tears because a wonderful story has come to an end [...] life seems empty and meaningless" (Ende, 1979). This work has been possible thanks to all the people who walked by my side, supporting me at every step for almost two years while pursuing my master studies, and also before. So, of course, this will be extensive. I *do not* apologize in advance for it.

I want to thank *Javier* and *Klaus*, my supervisors, who have guided me excellently through the roller-coaster ride that a master thesis in research is. Thank you for your patience, dedication, and encouragement every single day. Without a doubt, you have become mentors and have inspired me to continue pursuing my dreams and working on what I love.

The best part of DLR is definitely the people I had the chance to work with: all the people from the Perception and Cognition Department, the Deep Learning group, and others from different teams, groups, and institutes. Exchanging ideas with you has been incredibly enriching.

Sergio, you have been like a second supervisor to me. Thank you for all your good advice, help with low-to-high-level doubts, and patience. This would not have been possible without you. *Laura*, I do not even know how to *label* you, but the first word that comes to my mind is *support*. You have not only been a supervisor at DLR, but you have also become a coffee-machine buddy, a furniture-moving professional, an excellent churros chef, and above all, a friend. It is crazy that I had to come so far from home to meet you. It was worth it, though.

Also, thanks to all the professors who transmitted their knowledge with passion, encouraging me to keep on learning forever and ever, from my school years to the present. Researchers, musicians, scientists, everybody. A special thank you to *José Luis Mongrell*, not only responsible for my love for math but also for cultivating my divergence and exemplifying the importance of being a good person.

Thanks for all the support received from my *colleagues from the MRGCV*, for all the shared knowledge and help, and the good moments outside the university. To my *friends from the Bachelor's degree*, always supporting me. My *friends from the world of Music*, who keep me balanced and fill my life in a different dimension. My *friends from Vienna*, who still make it feel as if the last time we saw each other was two days ago. My *friends from La Salle*, always there as if we never left the school.

Dani. Thanks for making me enjoy the (scary) world of computer science and supporting me in every facet during the last years. You have become a friend. I will never forget the crazy US trip, your always-present laugh, and your positiveness. *Marina*, thanks for always taking any train, flight, ship, carriage, or any necessary means to see each other, and for all those

conversations. It is always a pleasure to share with you. Thanks, *Dave*, for being the best big brother I could have ever asked for when I arrived in Germany. Thank you, *Pablo*, for always encouraging me to overcome any challenge that may arise, for all your care, and for supporting me in the journey of discovering myself.

Paco, Luchi, Pilar, Aparicio. Even if you do not take me to the Conservatory, or bring the *merienda* after school anymore, you have been and still are fundamental in my life. You have definitely shaped the person I am now, and I am very lucky to have you. Thanks to all the support my family always offered me (aunts, uncles, cousins...); I could not have a better one.

Mamá, Papá. I will not extend myself, but if I am where I am, it is because of you. Thank you for always supporting my decisions, being there for me in my changes of direction, and showing me that mistakes are not mistakes if you learn from them. Also, for coming to visit me — please come more often. *Daniel.* Thanks for dropping me messages here and there and being yourself. Please, do not change, *cruc.* Your older sister truly admires you. Also, thanks to *Kevin* and *Eustace* for bringing bright moments home.

A very special thank you to my friends from *Room 1108 et al.* Since the very first day till the last one, I have had the chance to work not only with brilliant professionals but also surrounded by friends, and I feel very lucky and grateful for that. All the memories will remain with me (well, and maybe at some photographic gallery...).

Last but not least, thank you *Michele* for believing in me, and making this journey *so viel einfacher.*

¡Gracias!

This work has been partially financed by the scholarship programme for Master Thesis (2023-2024) of the I3A (Instituto de Investigación en Ingeniería de Aragón).

.

Index of Contents

1. Introduction

Within the last decades, computer vision has become a fundamental domain for numerous applications such as facial recognition, autonomous vehicles, medical imaging, or industrial automation. This field allows machines to interpret and make decisions based on visual data, emulating the capabilities of human vision. Advances in computer vision have been leveraged by the development of new algorithms, large datasets, and increasing computational power. Despite these achievements, challenges related to complex scene understanding, variations in environmental conditions or real-time processing are still the focus of ongoing research.

Perception is key for robots to interact with their environment. While humans rely on sensory inputs like vision, hearing, and touch, robotic systems use a variety of sensors such as cameras, Light Detection and Ranging (LiDAR), microphones, force sensors, or ultrasonic sensors, among others. Sensor fusion, which integrates data from different sensors, allows robots to create a comprehensive depiction of their environment, and therefore understand it. Visual perception is particularly important in robotics for tasks such as navigation, object recognition, and interaction with the environment. And accurate depth estimation becomes critical for applications where safety and reliability are of greatest importance, such as autonomous driving [66], simultaneous localization and mapping (SLAM) [8], and robotic manipulation [27].

Metric depth from cameras is typically achieved by stereo vision, which captures images from two points of view, similar to human binocular vision. While effective, stereo systems require a precise alignment and calibration of cameras, and are limited for applications where, for example, occlusions or space constrains occur. Their accuracy is limited by the baseline between the cameras, which in turn is limited by the physical and mechanical properties of the capturing device. Monocular approaches offer a more compact and less complex alternative since they use a single camera, but also face challenges such as scale ambiguity, making it difficult to determine the absolute scale of objects without additional information.

Light field technology offers a solution by acquiring multiple views of a scene simultaneously through a single device. By capturing not only the intensity but also the direction of light rays, these devices result potentially applicable for metric depth inference without the monocular scale ambiguity, overcoming the limitations of traditional monocular and stereo systems.

1.1 Objective and Goals of the Project

This master thesis explores the capabilities of focused light field cameras for single-view depth estimation using learning-based methods. By leveraging the optic configuration in focused light field cameras, this work seeks to develop a robust pipeline that generates dense depth maps at

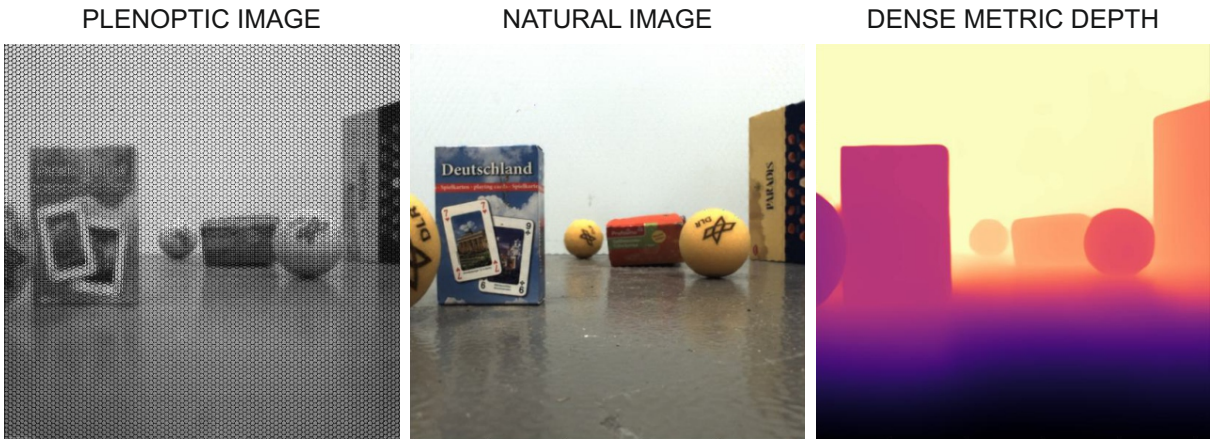

Figure 1.1: *Left:* Plenoptic image obtained from a light field camera, displaying the microlens pattern (details to be explained in subsequent chapters). *Center:* Corresponding natural image synthesized from the central viewpoint of the camera. *Right:* Dense metric depth map generated using the methodology proposed in this master thesis.

metric scale from plenoptic images (see Fig. 1.1), setting the groundwork for future developments in this field. To achieve this, the following specific objectives are established:

- Study the state of the art in light field imaging, with an emphasis on focused plenoptic cameras, and single-view depth estimation methods, particularly learning-based approaches.
- Search of existing datasets captured by a focused plenoptic camera and analysis of their applicability to the project.
- Design and capture of a real-world dataset, suitable for learning, using a focused plenoptic camera and a stereo system.
- Design and implementation of a methodology for predicting single-view depth from light field images.
- Evaluation of the model's performance with respect to similar approaches, and analysis of the experiments conducted during its design process.

This work is carried out at the Institute of Robotics and Mechatronics, part of the German Aerospace Center (DLR) in Munich, Germany. With 30 locations in Germany and offices in Brussels, Paris, Tokyo, and Washington D.C., the DLR is the national aeronautics and space research center of Germany, engaged in extensive research and development work in aeronautics, space, energy, transport, and security, and involved in national and international cooperative projects. The DLR significantly contributes to financing the European Space Agency (ESA), with 61% of its total space budget in 2020 [30].

The Institute of Robotics and Mechatronics focuses on developing a variety of robots designed to enable safe and efficient human interaction in different environments. The robots are designed

Stage	Process	Hours
Research	Study of light field imaging	43
	Study of plenoptic cameras	60
	Study of depth estimation	51
	Additional literature review	16
	Research on data availability	26
Practical work	Introduction to vision lab	12
	Lab work and dataset capture	83
Development	Image Processing Toolkit	90
	Micro Lens Depth Network	130
	Integration of Depth Anything	6
	Scale Alignment	8
	Additional studies	180
Writing	Writing of the manuscript	140
	TOTAL	**845**

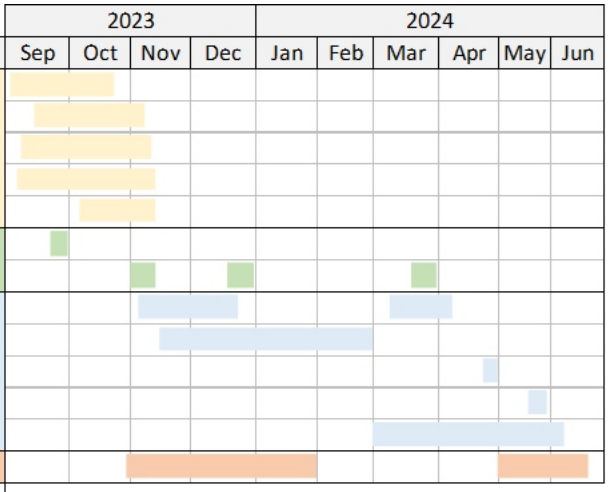

Figure 1.2: Gantt chart with the project's timeline.

to be used in inaccessible or dangerous surroundings, as well as to support humans in everyday tasks. They mimic and extend human manipulation and locomotion capabilities and perform tasks related to interaction with the environment, with a wide range of autonomy. The institute also emphasizes multimodal human-robot interaction to enhance the usability of robots.

1.2 Planning and Tools

The timeline for this work has been structured to align with the completion of the objectives outlined in Section 1.1. Fig. 1.2 shows the Gantt diagram detailing the schedule of different tasks. The tasks were organized iteratively and cumulatively, bulding on previous work. Recurrent tasks, such as manuscript writing, have been developed in parallel throughout the project. Also, certain tasks have been scheduled depending on limiting factors such as hardware availability. The work has been supervised regularly through periodic meetings with supervisors to discuss progress and plan subsequent steps.

This work has been developed using Python 3.9.16. For building, training, and testing the microlens depth model, the open-source framework PyTorch 2.0.1 has been utilized. The Image Pre-processing Toolkit leverages the computer vision library OpenCV 4.8.0.74. Data capture has been performed using DLR's internal software integration system *Cissy* (details in Chapter 5) and the light field camera manufacturer's software RxLive. Matlab was used for processing data from RxLive and transforming stereo disparities. Version control was managed with Git and GitHub[1], and Conda 23.1.0 was used for environment management.

Computations were carried out on a GPU cluster using Slurm as a workload manager. Specifically, this project was conducted on a server with a 18-core Xeon CPU with 2.3 GHz, 128 GB RAM, and a Quadro GV100 Volta GPU with 32 GB VRAM.

[1]All the code developed for this thesis is accessible upon request.

1.3 Structure of the Manuscript

The work developed in the master thesis is gathered in this document, organized as follows:

- Chapter 1 provides a motivation of the project, specifying its context, objectives, planning and tools used.
- Chapter 2 introduces the foundations of light field imaging, setting the theoretical basis for the project.
- Chapter 3 reviews the state-of-the-art methods for depth estimation and studies the availability of resources related to light field imaging.
- Chapter 4 delves into the fundamentals of a light field camera, including the camera model, its operation, and the capturing process.
- Chapter 5 covers the experimental procedure of data capture and processing towards the generation of a dataset, detailing design decisions, equipment, and software used.
- Chapter 6 explains the developed learning-based methodology for obtaining depth from light field information, providing details on its design and implementation.
- Chapter 7 presents and evaluates the obtained results, and discusses additional experiments.
- Chapter 8 summarizes the project contributions, performing an analysis of encountered limitations, and outlining potential future research directions.

2. Basic Concepts of Light Field Imaging

This section reviews the main principles of light field imaging. In detail, it provides an overview of the related concepts that are used throughout this work (Section 2.1), an explanation of the main representation methods (Section 2.2) and technologies for capture (Section 2.3), and finally summarizes several potential applications (Section 2.4).

2.1 What is a Light Field?

The term *light fields* stands for representations that collect the radiance of all the light rays that are emitted in a scene and hit a surface in the three-dimensional (3D) space. As a major difference against traditional images, they capture and model information not only about the intensity but also the direction of each light ray, which intrinsically provides a reconstruction of the geometry of the observed scene [128].

A light field can be formulated as a function that gathers the visual information and mapping between geometry of light rays and the light intensity components (i.e., red, green, and blue, namely RGB), frequently referred as the plenoptic function [128], thus providing information about the amount of light that flows in every direction through every point in the 3D space.

There are several approaches to model the plenoptic function. It has been frequently modelled as a five-dimensional (5D) function that defines each light ray as a three-coordinate position (x, y, z) and two angles that determine the orientation (θ, ϕ). However, more recent works gather

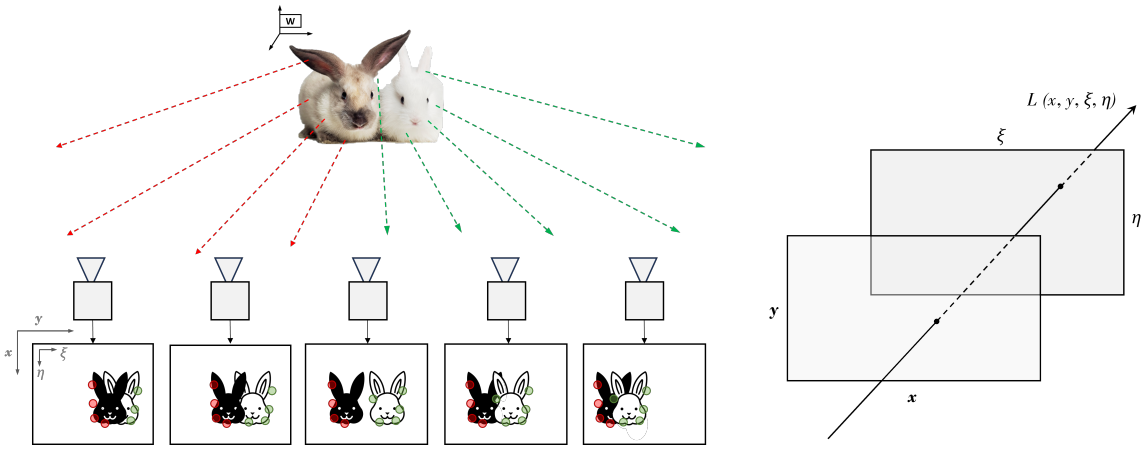

Figure 2.1: *Left:* Visual conceptualization of light field as a vector function that maps the geometry of light rays to the plenoptic attributes, providing information about the amount of light flowing in every direction through every point in space. *Right:* 4D light field representation. $L(x, y, \xi, \eta) \in \mathbb{R}^4$, where (x, y) represent the camera (lens) plane and (ξ, η) the image (sensor) plane.

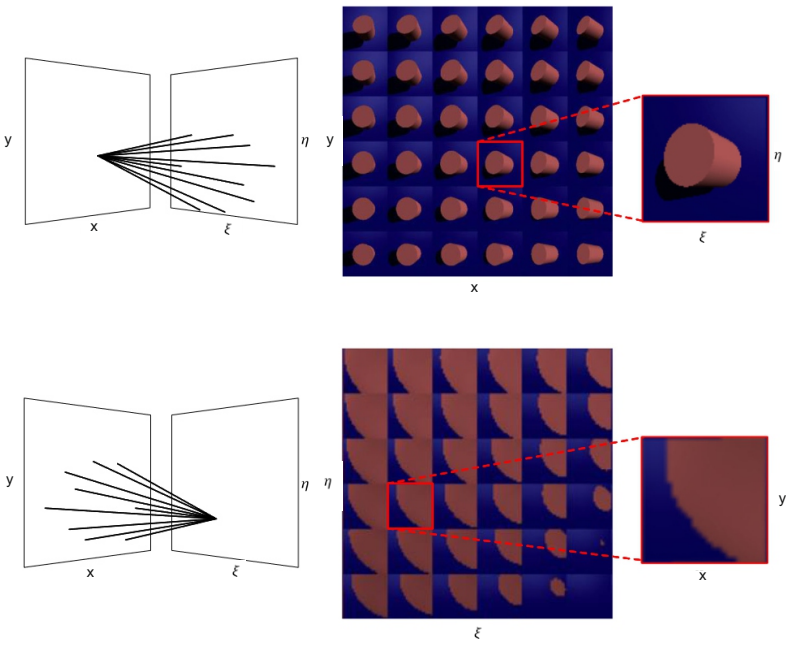

Figure 2.2: Two possible visualizations of a light field as a matrix of sub-aperture images can be generated by fixing either the aperture plane (x, y) or image plane coordinates (ξ, η). In the former *(top)*, each image represents all the light rays leaving the image plane that can pass the same point on the aperture plane. This generates a *direction-major* representation that provides a pinhole image for each point of view, arranged on a regular grid parallel to a common image plane. In the latter *(bottom)*, each image represents the light rays that pass through the same point on the image plane that reach different points on the aperture plane, thus generating a *position-major* representation.

the information of the scene in a four-dimensional (4D) plenoptic function $L(x, y, \xi, \eta)$. This method is known as the *two-plane parametrization*, and provides a description of each ray's intersection points with two parallel planes: the image plane ($\Omega \subseteq \mathbb{R}^2$) and the lens plane ($\Pi \subseteq \mathbb{R}^2$). This parametrization allows mapping the light field as:

$$L : \Omega \times \Pi \to \mathbb{R}, (\mathbf{p}, \mathbf{q})^{\mathrm{T}} \to L(\mathbf{p}, \mathbf{q}), \tag{2.1}$$

where $\mathbf{p} = (x, y)^{\mathrm{T}} \in \Omega$ and $\mathbf{q} = (\xi, \eta)^{\mathrm{T}} \in \Pi$ refer to the spatial and directional coordinates in the camera and sensor plane respectively (see Fig. 2.1).

2.2 Representation Methods

Since light fields are characterized by their complex and high-dimensional nature, they require dedicated representation methods that make them visually comprehensible. One conventional approach is to represent 4D light fields as a matrix of sub-aperture images which provides a more manageable 2D representation by fixing the spatial dimensions of one of the aforementioned camera and image planes [114, 128] (see Fig. 2.2).

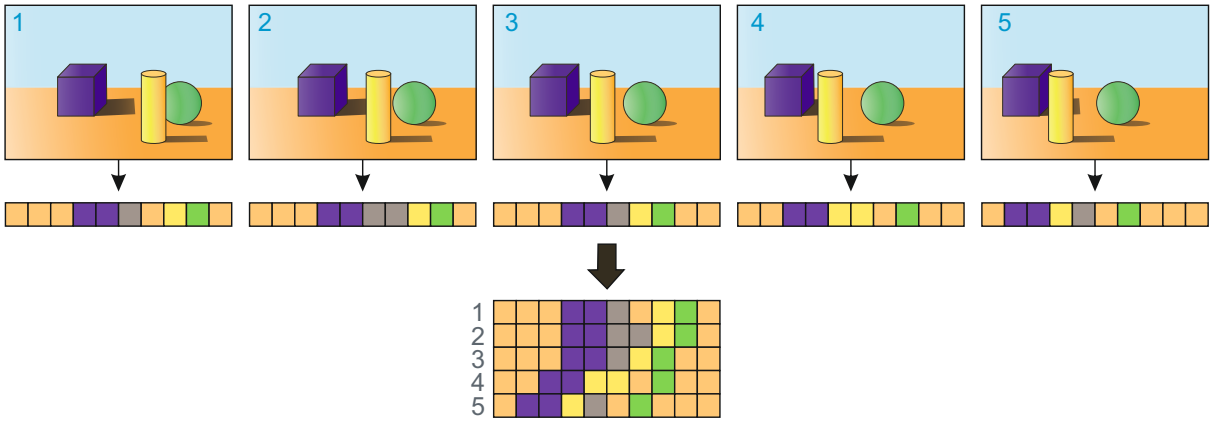

Figure 2.3: Representation of an Epipolar Plane Image (EPI), which is generated by executing a linear scanpath, typically in horizontal or vertical directions, across a scene. As the scanning progresses, images are integrated at each position, creating a slice that represents the distance from a specific point in the 3D space to the camera. This process provides a visual representation where each captured scene point corresponds to a distinct linear trace, and the resulting slope is indicative of the distance between a scene point and the camera, thus encapsulating spatial information of the scene, providing the distance relationships within the 3D environment.

Another common way of representing light fields is the Epipolar Plane Image (EPI), which is the 2D representation obtained by slicing the 4D light field holding constant one spatial and one angular dimension (i.e., one coordinate of each of the planes) [114]. In this manner, every point of the scene captured in one scan corresponds to a linear trace in the EPI, whose slope is related to the distance between the 3D point in the scene and the camera [63] (see Fig. 2.3).

The use of these representation techniques enhances the understanding of both the color and geometrical intricacies (i.e., spatial and angular dimensions) of captured scenes. However, these approaches come with certain limitations: They often involve decoding raw images taken by various capturing devices, necessitating the use of specifically designed light field image processing libraries, toolboxes, or proprietary software from manufacturers that may not always be open source or may cater to specific types of captured data or devices. Additionally, while synthesized representations serve as useful tools for comprehending light fields, they may introduce fidelity issues and artifacts. This reliance on handcrafted solutions brings up challenges such as resolution issues, sensitivity to generalization based on the device used for capturing the scene, and the need for careful calibration of capturing devices, among others.

2.3 Capture Technologies

In recent years, diverse technologies and devices have emerged for capturing light fields. A pioneering approach involves the use of *camera arrays*, where multiple cameras are spatially arranged to capture high-resolution light fields simultaneously (Fig. 2.4a). However, despite their efficacy, their bulkiness and cost have limited their widespread adoption. Another method, known as *time-sequential capture*, utilizes a single camera to capture a light field through multiple

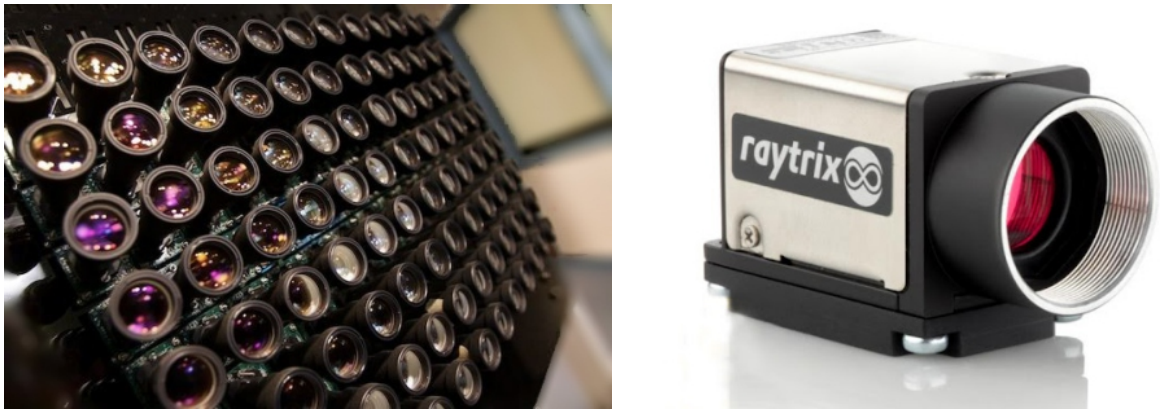

Figure 2.4: Overview of two of the most used light field capturing techniques. Stanford Multi-Camera Array [2] (*left*) and R5 plenoptic camera from Raytrix [93] (*right*).

exposures. However, this approach is time-consuming and imposes limitations on capturing scenes with dynamic elements.

An alternative technique, known as *multiplexed imaging*, involves encoding high-dimensional information into a simple 2D image. This approach has become the most popular method [102] since it is performed through the use of *plenoptic cameras*, also known as *light field cameras* (Fig.2.4b). These handheld devices offer a thorough view of a scene in a single shot. Their compact design enhances usability, since a microlens array (MLA) positioned in front of the sensor enables the capture of sub-aperture images arranged in a grid pattern [23].

Unlike other active light sensors, cameras employ passive technology, allowing the capture of light field images in outdoor scenes as easily as indoors. This capability to acquire light field images instantaneously in various environments adds to their versatility. The captured images offer multiple benefits, allowing visualization of scenes from slightly different viewpoints as well as post-shoot adjustments of the focal plane [78]. Also, their unique internal structure facilitates depth estimation even in uncontrolled settings, setting them apart from active sensing devices, which are restricted to controlled illumination and indoor scenes.

The emergence of plenoptic cameras has impacted photography, revolutionizing the way images are both captured and interpreted. However, these devices also present certain limitations. Their design entails a trade-off between angular, spatial, and sensor resolution. This challenge arises from the compact format of the camera, which inherently implies a narrow baseline between the microlenses, as well as the utilization of a conventional camera sensor.

2.4 Applications

From the early 2000s, light field imaging transitioned from research to practical applications in industry. This evolution led to the development of new devices such as the hand-held plenoptic camera [85]. In the 2010s, light field technology entered commercial markets [93, 1], offering a

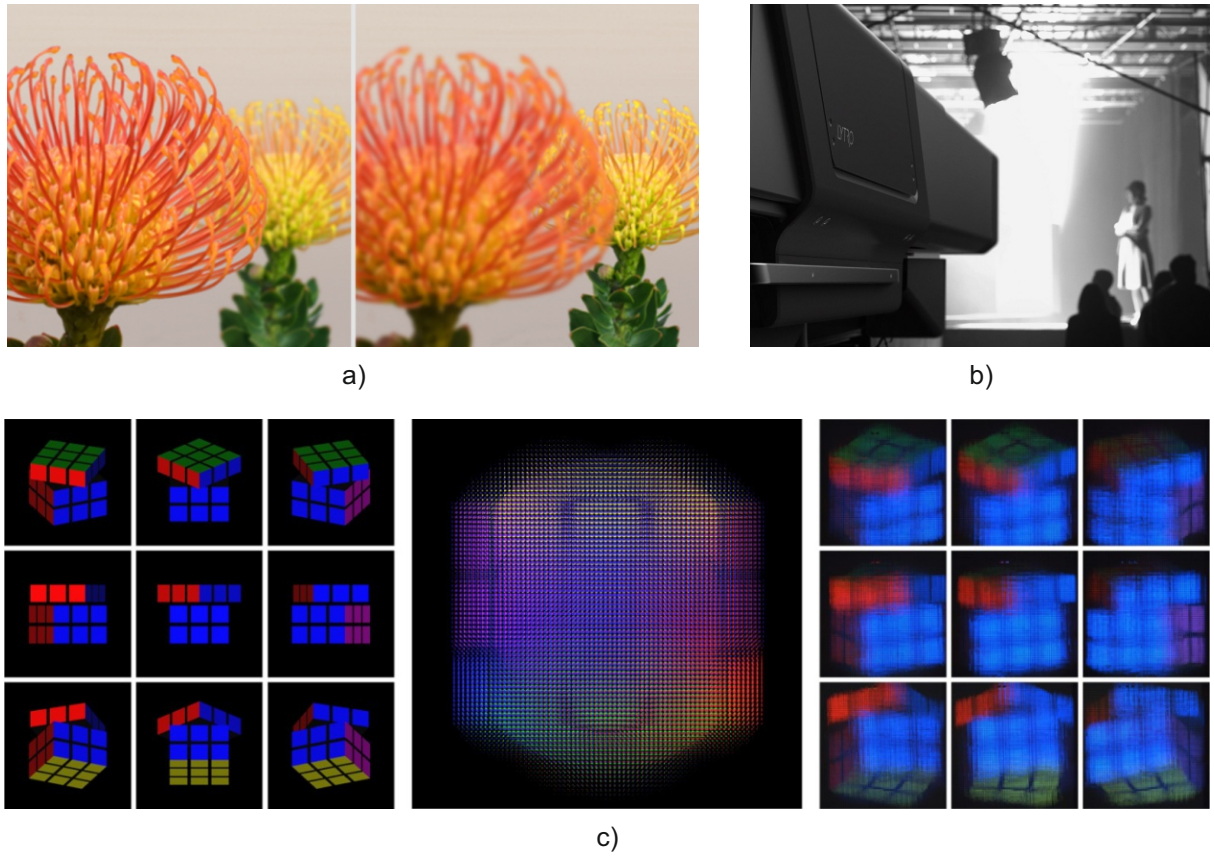

Figure 2.5: Examples of light field applications. a) Dynamic image refocusing [81] . b) Cinematography with a light field camera [21]. c) View synthesis of a Rubik's Cube's faces though light field data [51].

number of advantages over traditional cameras, notably the ability to refocus after capturing an image. These advances opened new possibilities of application in disciplines such as image editing, holography, perception, and augmented reality (see Section 2.3).

Furthermore, significant progress has been made in light field applications during the last decade (see Fig. 2.5): Areas such as light field editing [10], image quality enhancement [101], 3D reconstruction [23, 63], view synthesis [58], and the acquisition and display of light fields in industrial settings [77] have witnessed notable advancements. The capacity of light field technology in addressing diverse challenges and requirements makes it versatile to potentially be used across various domains. This work particularly focuses on the study of depth estimation (see Section 3.1).

3. Related Work

This section presents relevant past work regarding the estimation of depth, the generation of extended depth-of-field color images, and the availability of datasets and other resources in the area of light-field imaging.

3.1 Depth Estimation

The challenge of depth estimation has been a widely explored problem in the field of computer vision. Understanding the spatial relationships within a scene is a fundamental aspect of visual perception, which is essential for a broad number of applications ranging from robotics to augmented reality. The precision with which a system can measure[1] depth defines its ability to interpret the environment, which has a high impact in processes that involve decision-making.

This section delves into the state of the art of depth estimation. It focuses on three different approaches: First, on classical methods (Section 3.1.1); then, on cutting-edge learning-based techniques (Section 3.1.2); and finally, on methods applied to plenoptic imaging (Section 3.1.3).

3.1.1 Geometry-based Methods

As detailed in Section 2.3, light field capturing devices can be perceived as intricate derivatives of a stereo vision system [38]. Within this context, the principle of stereo triangulation still emerges as a fundamental concept, serving as a basis for a broad variety of depth estimation approaches.

Triangulation

In computer vision, the process of determining a point's 3D position from corresponding image projections and known camera positions is known as *triangulation* [110]. This foundational concept relies on the known distances between two separated vantage points observing the same point in the scene. By leveraging these parameters and relationships, the distance from the observing positions to the scene point can be precisely calculated. This fundamental idea finds a widely-known application in stereo vision systems, where depth information is extracted from a pair of images obtained from pre-calibrated cameras placed in two different points [84].

[1] *Measuring* differs from *inferring*; while machine learning can infer or deduce details like high resolution, depth, or refocus from training data, measuring lacks prior assumptions, reducing bias and enhancing accuracy and safety in unfamiliar environments.

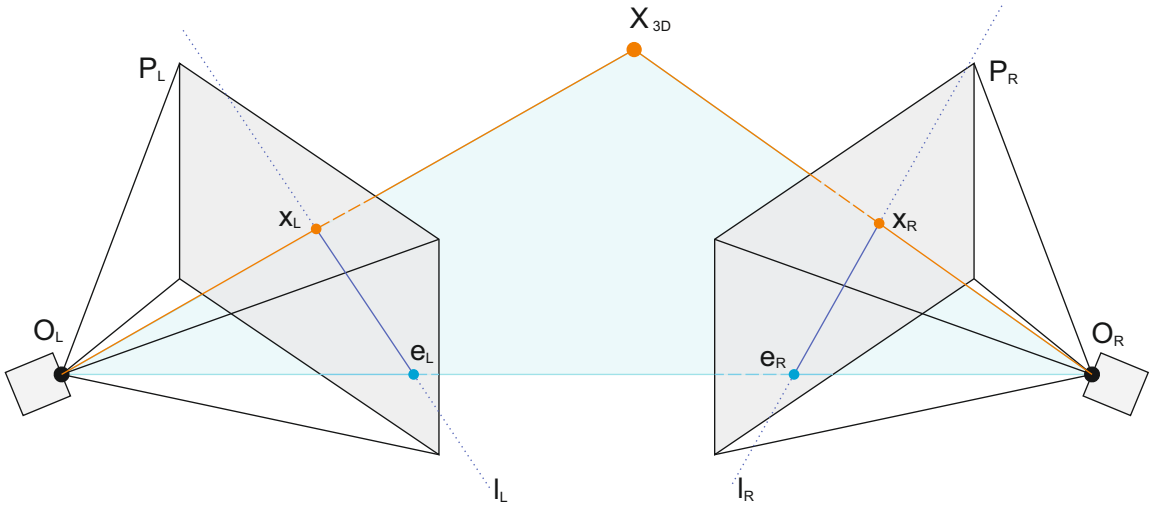

Figure 3.1: The 3D point stereo triangulation principle involves determining the point that best approximates the intersection of the 3D rays projected from the optical centers O_R, O_L of two cameras. In the picture, the 3D point $\mathbf{x}_{3D}$, belonging to the epipolar plane (*blue*), is found through the intersection of the rays that are projected from each camera towards the 2D projection of the point in each camera, x_L, x_R. The projections of the position of each camera in the other camera's field of view (e_L, e_R) are shown, as well as the respective epipolar lines, l_r, l_L, that, in this case, are formed between the projected point and the epipole.

Analogously, triangulation enables the transformation of 2D image data into precise 3D spatial information, forming the backbone of various depth estimation techniques [89].

To determine the 3D pose of a point in space, the use of two calibrated projective cameras with known poses, both observing the same point, is necessary. However, due to the presence of noises (from image processing) and inaccuracies (from calibration), the projected 3D camera rays for both cameras do not necessarily exactly cross.

A widely adopted approach to address this challenge is to identify the 3D point closest to the projected 3D rays with origin at the j^{th} camera optical center $\mathbf{c}_j$ and in a direction $\hat{\mathbf{v}}_j$, that correspond to each of the the 2D matching feature locations $\{x_j\}$ observed by the cameras $\{\mathbf{P}_j = \mathbf{K}_j[\mathbf{R}_j|\mathbf{t}_j]\}$, where $\mathbf{t}_j = -\mathbf{R}_j\mathbf{c}_j$ (see Fig. 3.1). This process aims to find the nearest 3D point by minimizing the distance between the two closest points along the traced rays. In other words, this method seeks to determine the spatial location that optimally aligns with the intersection of the camera rays by solving an optimization problem where the residual in the measurement equations is minimized. Therefore, the optimal value for the 3D point can be computed as a least squares problem.

Alternatively, it can be solved by minimizing the residual in the measurement equations, leading to non-linear least squares problem. This can be linearized and reformulated as a problem analogous to the Direct Linear Transformation (DLT) [40]. When using homogeneous coordinates, the resulting set of equations becomes homogeneous. Consequently, the problem can be tackled with the Singular Value Decomposition (SVD), where the objective is to identify the smallest singular eigenvector [110, 82].

Depth and Disparity

It is common to rely on disparity as a reference to the estimated depth, especially in stereo vision problems, since it measures the apparent shift in an object's position between the views obtained from a camera pair, which is naturally related to inverse depth. While disparity provides a measure of relative depth, it lacks the inherent real-world scale of the scene. To provide information in the real metric scale in the case of unknown scene objects, the baseline distance between both cameras (i.e., the separation between their optical centers) and the focal length have to be known and considered. These parameters are fundamental since they allow the conversion of depth information from disparity units to the real-world scale employing the triangulation principle.

However, it is noteworthy that certain methods in the literature (especially the monocular ones) operate under the constraint of scale ambiguity: In these cases, the baseline distance remains unknown, introducing an uncertainty on the estimated depth and leaving it as an up-to-scale parameter. One of the most widely used methods is Structure from Motion (SfM), which aims to recover the 3D structure as well as the poses of cameras from image correspondences, but it embraces the scale ambiguity issue [110]. Other methods, including learning-based approaches, may also encounter these restrictions [97]. In these cases, addressing scale ambiguity becomes an extended challenge, emphasizing the importance of careful calibration and exploring innovative techniques to enhance the accuracy of depth estimation for a broad variety of computer vision applications [106].

Multi-View Stereo Vision

In the reconstruction of a 3D scene from calibrated overlapping images taken from various viewpoints, multi-view stereo approaches prove to be an improvement for obtaining depth information [103]. Unlike stereo depth, where information comes from two viewpoints, multi-view stereo benefits from a higher number of data captured from multiple vantage points.

A prevalent approach involves utilizing the sum of squared differences (SSD), which assesses differences across all images concerning a reference image. These techniques involve challenges like the baseline constraint, which implies a balance between geometric accuracy and robustness to occlusions [28]. Another approach is the EPI analysis, involving stacking corresponding scanlines from all images to exploit the lateral movement of objects at varying depths [112]. Techniques like Kalman filtering or maximum likelihood inference deal with series of sequential observations, enhancing the robustness of the depth estimation process.

Whenever there is less availability of images, aggregation techniques such as sliding windows or global optimization become essential [60]. While computing a depth map from multiple inputs surpasses the accuracy of pairwise stereo matching, more significant improvements arise when

estimating multiple depth maps concurrently from each pair. This allows more accurate reasoning about occlusions, as regions obscured in one image may be visible in others. Algorithms like COLMAP leverage view selection and geometric consistency between multiple depth maps to filter matches [109, 59, 76, 123, 98].

Depth from Monocular Vision

Monocular (also known as single-view) depth estimation is an under-constrained problem, since in general it is geometrically impossible to determine the depth of an unknown scene from the information in the pixels of just one image [14]. Paralleling the way humans infer depth effectively with a single eye by leveraging cues such as perspective, scaling, and visual appearance through lighting and occlusion, there are computational methods that aim to compute depth with a single image [11, 128]. To address this challenge, learning-based approaches have emerged, capable of leveraging prior information to predict pixel-wise disparity or up-to-scale depth from RGB camera frames.

Monocular plenoptic cameras, however, are able to measure depth through a single camera lens. As discussed in Section 4.1, these devices enable the capture of light field images in a single exposure using one camera. This unique characteristic aligns with the principles of depth estimation from monocular vision, where a single device is employed to capture the scene.

3.1.2 Learning-based Methods

Over the past few decades, deep learning has been applied to a wide variety of domains, notably in computer vision and graphics, to extract valuable insights of 3D scenes from images using the prior knowledge learned from data, as well as the adequate architecture of the neural networks. One particularly notable challenge is inferring 3D information from monocular images, where conventional geometry-based approaches cannot be applied. Unlike scenarios with more than one point of view where multi-view stereo approaches can be applied, monocular images provide spatial information through a single viewpoint, making triangulation unfeasible. In this case, specific approaches tackle the problem of monocular depth estimation.

Monocular Depth Estimation

In recent years, significant progress has been achieved in learning-based monocular depth estimation methods particularly through the utilization of multi-scale deep networks. These networks predict coarse depth initially and then refine the predictions using both global and local networks [22]. While architectures like convolutional neural networks (CNNs) [115], transformer networks [20, 73, 9, 127], and those incorporating pixel-wise transformer layers [50, 92, 126] have

shown promising improvements, the challenge of slow convergence persists as these methods treat depth estimation as a regression task.

Some alternative approaches frame the problem as a classification task by discretizing depth into intervals [26, 13]. This yields higher model performance but often comes at the cost of visual and depth quality.

Furthermore, other works combine the aforementioned methods and adopt a hybrid classification and regression strategy. These approaches learn probabilistic representations for each pixel, predicting final depth values as a linear combination of probability distributions with discrete bins [7, 56, 69]. While these methods address issues in visual quality, they may introduce inductive biases, which can be limiting in tasks where different assumptions hold, or lose global information among other problems.

Additionally, some works in the literature explore strategies for enhancing monocular depth estimation. These include self-supervised learning [33], incorporating several auxiliary tasks like environment classification to perform multi-task training [57], using specific supervision losses, and employing relative estimation techniques [53].

Multi-View Stereo

A variety of works use deep neural networks to tackle the multi-view stereo problem. Initially, learning-based approaches focused on acquiring more robust feature representations to generate correspondence pairs which would be averaged for better matching features from multiple images [39, 122, 75], and consequently better depth estimation. However, learning a matching function using multiple images as an input [41] had an impact in how this problem was addressed. More recently, end-to-end learning methods have been introduced. These works combine both classical constraints such as feature matching [61, 55] or plane-sweep [49] approaches with learned high-level information to address the depth estimation problem.

More recent works integrate deep neural networks to compute matching or cost volumes and fuse them into disparity maps [49, 62]. They incorporate visibility and occlusion handling, confidence maps, geometric consistency checks, and efficient propagation schemes to achieve superior results [120, 125, 86].

Both single and multi-view stereo vision techniques find extensive applications in the fields of computer vision and robotics [118]. They are particularly prominent in visual Simultaneous Localization and Mapping (SLAM), where the fusion of multi-view constraints and depth information enhances maps accuracy as well as the camera pose estimation [15]. In the domain of autonomous driving, these techniques are crucial for urban reconstruction algorithms, which play a fundamental role in navigation. However, these algorithms must exhibit robustness to effectively tackle a number of challenges, including variations in lighting conditions, occlusions,

changes in appearance, high-resolution inputs, and the demand for large-scale outputs. The resulting 3D reconstructions have a significant role in applications such as static obstacle detection and precise localization for collision avoidance strategies [54].

3.1.3 Methods Applied to Plenoptic Imaging

As mentioned in Chapter 2, a light field contains the spatio-angular information of light rays, providing valuable cues, such as correspondence and defocus, binocular disparity, aerial perspective, and motion parallax. These cues allow performing depth estimation for a captured scene [128]. Additionally, the increase in the usage of light field cameras is related to their versatility and good capabilities handling occlusions, which contributes to enhance the robustness in depth estimations by solving a commonly encountered challenge [60].

Diverse methodologies have been proposed to tackle the task of depth estimation from light field images. Over the last two decades, the approaches have diversified into traditional geometry-based methods and more contemporary learning-based pipelines. According to the extensive review by Zhou et al. [128], depth estimation methods can be categorized as constraint-based when they use various constraints (i.e., combinations of depth cues) of the light field structure; EPI-based when the EPI representation is exploited to perform a dimensionality reduction; and CNN-based, employing convolutional neural networks leveraging prior data for obtaining a better balance between accuracy and computational cost.

Some works explore classical approaches, such as applying the triangulation principle to the MLA in a plenoptic camera [38]. However, these models are custom-made for a previous model of plenoptic cameras [85] (see Section 4.1.1), and addressing the more complex and updated models such as the one used in this master thesis is recognized as a future challenge. Remarkable efforts have been done for feature extraction for Structure from Motion in plenoptic cameras (P-SfM) [124] and the application of the Central Projection Stereo Focal Stack (CPSF), which performs a refocusing around the central projection for feature extraction [72]. Nevertheless, while these methods achieve high accuracy, their extensive computational requirements remain a significant drawback.

There are few works on machine learning techniques for light field analysis. Some models address challenges such as disparity reconstruction [42], object detection, and material recognition [117, 58]. Others implement model-free approaches for problems that are not confined to a specific statistical space, such as face reconstruction [23]. However, the scarcity of high-quality large-scale training data, crucial for training deep network architectures, makes them recur to classic methods as viable alternatives. These techniques rely on handcrafted solutions, including variational principles or EPI filtering [45], which can be combined with learning-based approaches, both supervised and unsupervised [19], such as CNNs [44, 43] or attention modules [111, 12].

3.2 Data and Resources

As detailed in Section 3.1.2, some works have addressed depth estimation problem from light field capturing devices, some of them with a focus on learning-based methodologies. However, when exploring the domain of learning-based approaches, a recurring obstacle emerges: the scarcity of available data. This shortage is caused by both the limited availability of light field capturing devices and, in the case of labeled, supervised 3D training, the difficulty in setting up an external ground-truth 3D measuring system for data registration. The utilization of a camera array, although effective, proves to be a cumbersome and expensive alternative. The requirements involve multiple cameras, considerable space, and processing units, which makes this approach impractical for a wide range of users and institutions. Additionally, the lack of mobility in such setups due to their considerable dimensions restricts the ability to capture diverse scenes from different locations. Therefore, the capture of light fields is constrained to a predominantly static point of view, often within a laboratory setting, where changes occur in the scene itself rather than in the pose and environment of the camera setup.

Considering plenoptic cameras, their compact and versatile design stands out as a notable advantage with respect to stereo cameras and structured-light solutions, positioning them as commercially viable and customer-oriented devices [1]. Moreover, the advanced technology incorporated into these cameras allows post-processing advantages, although it is conditioned by the dependence on the manufacturer's proprietary software for decoding raw data [94]. This reliance involves certain limitations, further influencing the accessibility and adaptability of the plenoptic camera. In addition, their rarity and the precision required in their manufacturing process contributes to their elevated cost, preventing widespread adoption in the consumer market [96], which in turn does not allow for the cost advantages of economies of scale.

These limitations have led to a lack of light field datasets captured with plenoptic cameras. Notably, the Stanford Light Field Archives stand out as significant contributions in this domain, since they gather information comprising images captured by a well-known camera array with high resolution [119], previous versions of plenoptic cameras [36], or multi-view camera systems [16]. Additionally, other datasets captured with similar setups exist [83, 95]. These datasets are particularly valuable for tasks involving decodings like EPIs and multi-view stereo. However, its utility in learning-based approaches is limited by a lack of sufficient data as well as a data format or version misalignment with respect to raw images obtained by a modern plenoptic camera 2.0.

Several toolboxes have been developed to tackle the decoding challenges associated with plenoptic cameras [17, 18, 35, 79, 80]. These tools enable the conversion of plenoptic images into natural images or other representation forms, such as EPIs, yet have been carried out for plenoptic cameras 1.0, leveraging its geometric model for the decoding process [74]. However, with the introduction of the focused plenoptic camera, or plenoptic 2.0 (see Section 4.1.1), significant

Table 3.1: Overview of the existing light field datasets that are publicly available [47]. Information about their public availability and other features that have been taken into account for this project. The suitability for learning is based on the number of images they have. The suitability for modern plenoptic cameras 2.0 depends on the device which has been used to capture the data.

Name	**Type**	**Suitable for learning**	**Suitable for modern plenoptic cameras 2.0**	**Ref.**
(Old) Stanford Light Field Dataset	Real-world	No	No - Camera array	[65]
New Stanford Light Field Dataset	Real-world	No	No - Camera array	[119]
Stanford Lytro Dataset	Real-world	No - 180 samples	No - Lytro 1.0	[1]
Stanford Multiview Dataset	Real-world	Yes - 7200 samples	No - Lytro 1.0	[16]
MIT Synthetic Light Field Archive	Synthetic	No	N.A. *	[87]
HCI 4D Light Field Dataset	Synthetic	No - 28 samples	N.A.	[48]
Lytro first generation dataset	Real-world	No - 30 samples	No - Lytro 1.0	[83]
EPFL Light-Field Image Dataset	Real-world	No - 118 samples	No - Lytro 1.0	[95]
Light field Saliency Dataset (LFSD)	Real-world	No - 100 samples	No - Lytro 1.0	[68]
LCAV-31 - A Dataset for Light Field Object Recognition	Real-world	No	No - Lytro 1.0	[31]
A 4D Light-Field Dataset for Material Recognition	Synthetic	No	N.A.	[117]
Occlusion-aware depth estimation using LF cameras	Synthetic	No	N.A.	[116]
DDFF 12-Scene 4.5D Lightfield-Depth Benchmark	Real-world	No - 720 samples	No - Lytro 1.0	[42]
University Rome, SMART Dataset	Synthetic	No	N.A.	[91]
MPI Light Field Intrinsics	Synthetic	No	N.A.	[100]
MPI Light Field Archive	Synthetic	No	N.A.	[4]
Matching Lytro and Raytrix Dataset	Real-world	No - 31 samples	Yes	[5]
CVIA Konstanz Specular Dataset	Synthetic	Yes	N.A.	[6]
V-SENSE Lytro Illum Dataset	Real-world	Yes	No - Lytro 1.0	[67]
Custom-built Plenoptic Camera Dataset	Real-world	No	No - Lytro 1.0	[37]
POV-Ray LF dataset	Synthetic	Yes - 900 samples	N.A.	[43]

**** N.A. stands for not applicable.***

internal modifications, including adjustments to the MLA position and other features within the camera lens setup have been introduced, rendering existing toolboxes and methodologies designed for plenoptic 1.0 to some extent obsolete for this new system.

As of now, there are no toolboxes available that are specifically designed for plenoptic cameras 2.0 [88]. This absence can be attributed to the non-trivial nature of the geometrical model and required computations associated with this device. Furthermore, a general lack of information arises due to the proprietary nature of these systems. Consequently, since detailed specifications (like the inner distance between the camera chip and the MLA) are not publicly available, decoding the obtained information becomes a challenging task. Moreover, the lack of transparency in such software may introduce potential errors into the decoding process.

Given these challenges, acquiring a real-world plenoptic dataset that encompasses both indoor and outdoor scenes becomes a difficult task. Particularly, assembling a dataset tailored for deep learning purposes presents complexities, mainly due to the substantial quantity of images required. Hence, up until this point, the majority of studies in the field of learnt depth in plenoptic imaging have relied on synthetic datasets crafted through modeling software [65, 87, 47, 91, 100, 4]. These synthetic datasets are meticulously designed, taking into account diverse parameters to enhance variability and robustness. However, the lack of real-world labeled data emphasizes the challenges related to obtaining realistic and extensive datasets for developing learning-based techniques in this area. An overview of the inspected datasets is provided in Table 3.1.

4. Operating Principles of a Light Field Camera

Throughout the evolution of light field technologies, different methodologies have emerged to encapsulate the intricacies of light fields, as described in Section 2.3. Among these, the *camera array* stands out: A grid of cameras strategically positioned to capture a scene from subtly varied perspectives, functioning as a multi-view stereo system, thus enabling the extraction of spatial information for points within the scene. Another avenue explores *time-sequential techniques*, and in this context, the focal sweep approach is noteworthy. This technique allows the correlation of different apertures to focusing distances, providing a mean to estimate the distance of a point with respect to the camera used for the capture. Finally, the *plenoptic camera* emerges as a user-friendly concept with several capabilities that make it a versatile tool with high potential. In this chapter, the plenoptic camera model is studied, making emphasis on its geometry, and delving into the core concepts that define its unique operation.

4.1 The Plenoptic Camera

While the fundamental concepts of 3D photography trace back to the early 20th century, particularly to integral photography works [52, 71], it was not until the end of the same century that the ground-breaking plenoptic camera concept was proposed [3].

The core design innovation consists of adding a MLA between the sensor and the main lens set. This arrangement allows sampling the 4D radiance at the microlenses. Therefore, these microlenses produce multiple sub-aperture images, which the sensor captures. Each sub-aperture image presents a slightly different perspective of the scene, similar to the approach of a camera array. This device does not only introduce a revolutionary method for capturing light fields with a single exposure, but also allows recording comprehensive spatio-angular information.

4.1.1 Camera Model

The first prototype of a plenoptic camera made its debut in 2005 [85]. This pioneering device marked a revolutionary shift in digital photography by introducing capabilities such as refocusing, noise reduction, and image sharpening. In addition, it operated in the same manner as an ordinary hand-held camera. This model is widely known as the *traditional*, *standard*, or *plenoptic camera 1.0*, and its optical design comprises a photographic main lens, a MLA and a photo-sensor array. In this traditional model, the main lens is focused at the microlens plane, and microlenses are focused at optical infinity – i.e., the main lens (see Fig. 4.2 a). Essentially, the main lens remains fixed at the microlenses' optical infinity, and the photo-sensor is glued

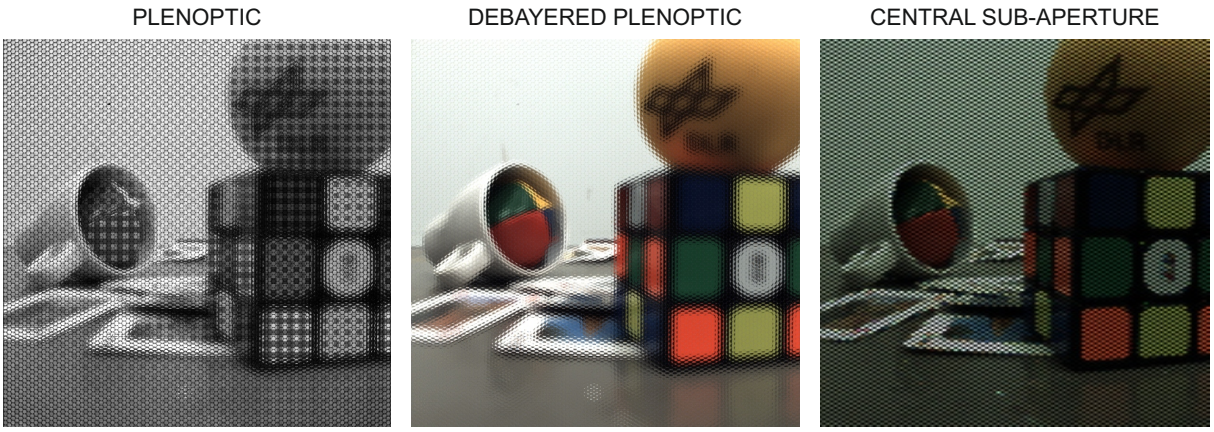

Figure 4.1: *Left*: The plenoptic light field image obtained from a light field camera allows obtaining the corresponding sub-aperture images by fixing two coordinates – in this case, the ones referred to the pixels inside each lenslet $\mathbf{p} = (x, y)^{\mathrm{T}}$ – thus generating a *direction-major* representation. *Center*: Corresponding debayered plenoptic image. *Right*: Central sub-aperture image (i.e., coordinates are fixed at the centroid of each microlens).

at their focal depth to enable the focusing of the microlenses [85]. This configuration enabled extracting light fields from plenoptic sub-aperture images according to the 4D parametrization explained in Section 2.2 (see Fig. 4.1 for further details).

However, in this model, due to the focusing condition at infinity, each microlens image is defocused with respect to the image created by the main camera lens and the scene object. This results in only a single pixel being rendered in the final image, leading to a loss of resolution, which is a limiting factor. Consequently, a new plenoptic camera model was developed in 2009 that interprets the MLA as an imaging system focused on the focal plane of the main camera lens [74]. This new setup involves a structural change, allowing the microlenses to focus on the image produced by the main lens (i.e., its focal plane) inside the camera rather than at infinity. This introduces a flexible trade-off between spatial and angular dimensions, enabling multiple pixels from each microlens to be rendered in the final image [29]. Widely known as the *focused plenoptic camera* or *plenoptic 2.0* (see Fig. 4.2 b), it is the model extensively used nowadays and in this work.

4.2 Light Field Software

Plenoptic cameras, despite attempts to introduce them to the market, are still rare devices and have not gained widespread acceptance among users. This may be attributed to the perceived complexity of the technology and their high price, restricting their usage mainly to research institutions and industries. In the context of this project, we use a camera device by Raytrix [1], a prominent manufacturer operating mainly in Europe.

The associated software, RxLive, plays a crucial role in handling data from the Raytrix light

[1]Raytrix R5 camera: `https://www.youtube.com/watch?v=1zBtKni9mRs&ab_channel=raytrix`

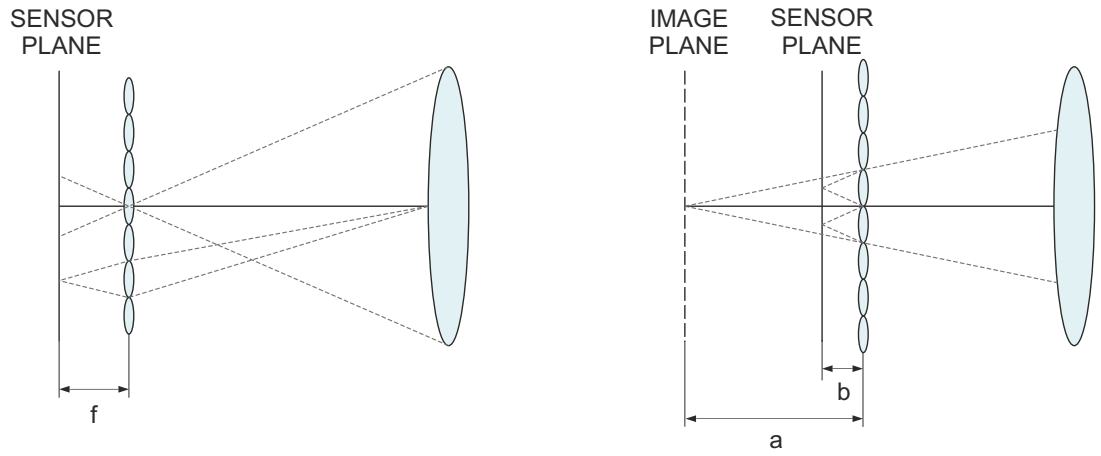

Figure 4.2: Illustration of (*left*) the traditional plenoptic camera 1.0 and (*right*) the plenoptic camera 2.0. In the plenoptic camera 1.0, the main lens is focused on the microlens plane, and the microlenses are focused at optical infinity, which corresponds to the main lens. In the plenoptic camera 2.0, the real image is focused in a virtual plane in front of or behind the MLA at a distance a, which corresponds to the imaging plane of the microlenses. The virtual image is then re-imaged onto the sensor [74].

field camera. RxLive facilitates data capture, processing, and calibration, through an intuitive and user-friendly visual interface. While the software enables the extraction of plenoptic images with the MLA, it also employs proprietary algorithms to reconstruct the natural image from the central point of view. Moreover, RxLive offers functionalities such as 3D reconstructions and data presentation in various formats like point clouds and color maps. Despite its versatility, RxLive is not an open-source solution, and the methods utilized for depth estimation are not publicly disclosed. This lack of transparency raises questions about the reliability of depth calculations and the potential presence of errors in the proprietary methods.

To address this concern, this project adopts a multi-faceted approach, dividing the problem into several sub-problems. The aim is to examine and evaluate the accuracy of the state-of-the-art proprietary software, such as RxLive, in comparison to a stereo system employing a semi-global matching algorithm. The latter has previously demonstrated sub-pixel accuracy [46]. The technical details about the devices used in this project can be found in Chapter 5.

4.3 Scene Capture with a Focused Plenoptic Camera

As shown in Section 4.1.1, the plenoptic camera 2.0 model can be conceived as a device to capture the scenes where the image focuses in a virtual plane that needs to be re-imaged onto the sensor (see Fig. 4.3). Therefore, the depth information obtained from the camera is provided in terms of virtual distances.

The inner parameters that relate to the configuration of the latest model of the camera remained undisclosed until 2016, when specific calibration software was developed to obtain values related to depth and brightness parameters [107]. Image-based calibration methods were

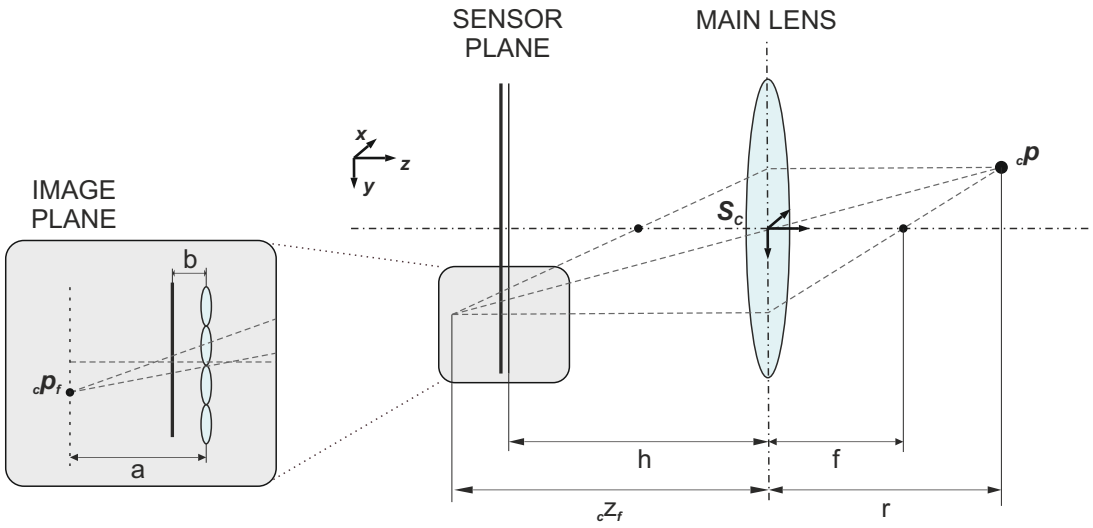

Figure 4.3: Representation of the plenoptic camera 2.0 model based on the thin lens camera model approach [107].

conceived to achieve this goal, enabling the deciphering of internal parameters of the camera configuration and the recovery of information about the scene not directly provided by the manufacturer. Obtaining a partial model of the camera plays a fundamental role in this problem, since it enables the calculation of metric depths from the virtual depths provided by the manufacturer – which magnitude is not provided originally.

Strobl et al. [107] refer to virtual depths, denoted as v, as a quotient of the focusing distance a of the MLA, and the distance b between the sensor and the MLA (see detail in Fig.4.3):

$$v = a/b, \tag{4.1}$$

where b is an unknown value associated with the camera's production. Virtual depths have limited utility due to their non-metric nature with respect to the corresponding depth ${}_c\mathrm{z}$ for a point ${}_c\mathbf{p}$ in the scene. However, the thin lens camera model allows expressing its corresponding focused projection ${}_c\mathbf{p}_f$ at the camera's virtual focusing plane as follows:

$${}_c\mathbf{p}_f = \begin{bmatrix} {}_c\mathrm{x}_f \\ {}_c\mathrm{y}_f \\ {}_c\mathrm{z}_f \\ 1 \end{bmatrix} \propto \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{bmatrix} \cdot \begin{bmatrix} {}_c\mathrm{x} \\ {}_c\mathrm{y} \\ {}_c\mathrm{z} \\ 1 \end{bmatrix}. \tag{4.2}$$

Therefore, internal metric projections ${}_c\mathbf{p}_f$ at the camera's virtual focusing plane can be obtained from virtual depths as follows:

$${}_c\mathrm{z}_f = a + h \equiv v \cdot b + h, \tag{4.3}$$

where h represents the distance between the MLA and the camera reference frame S_C (see Fig. 4.3). To perform such transformation, accurate estimations of h and b are required. For this purpose, the Plenoptic Camera Calibration Matlab Toolbox (*P-CalLab*) [108], developed at DLR, is utilized to obtain estimations of the camera parameters. Once obtained, and following the thin lens camera model assumption [107], the metric distance r of a point with respect to the camera can be calculated using:

$$\frac{1}{f} = \frac{1}{{}_c z_f} + \frac{1}{r}, \tag{4.4}$$

where f is the focal length of the main lens, ${}_c z_f$ is the depth of the virtual projection, and r is the metric depth in the direction of the principal axis of the camera (see Fig. 4.3).

Image Decodification Pipeline

In this project, RxLive is employed for capturing various scenes. Given the focus on depth estimation, the software's output is configured to provide the desired information: This includes the plenoptic image, representing the data gathered by the sensor through the microlenses (resulting in visible lenslets in the output). Additionally, the all-in-focus natural image is synthesized using proprietary algorithms. The plenoptic depth map is obtained, displaying virtual depth values and their confidence in a two-channel matrix. The plenoptic depth map translates this information into a continuous color map. The configurations for different scene codifications are fine-tuned using the RxLive user interface.

Pre-processing depth images obtained with RxLive is fundamental to provide the ground-truth plenoptic depth, serving as reference for training a network in subsequent stages of the project's pipeline.

5. Dataset Generation

As it has been previously described, there are limitations in applying learning-based approaches to address the depth estimation problem for plenoptic cameras, mainly due to the absence of a suitable dataset (see Section 3.2). Furthermore, the limited commercial availability of this kind of cameras, together with manufacturers' reluctance to disclose information for confidentiality reasons, hinders the utilization and interpretation of data obtained from these devices. As an alternative, it is possible to complement these cameras with secondary systems such as stereo cameras, infrared sensors, or LiDAR. These additional devices can provide valuable information about the scene that may not be directly available through the manufacturer's software. These cues can then be incorporated into a learning-based pipeline to infer information that has not been publicly provided.

This chapter focuses on the creation of a light field dataset that is suitable for deep learning-based approaches (see Section 5.1), such as the depth estimation problem that this project aims to tackle. It first reviews the available hardware and software, and their impact in an accurate and robust depth estimation (Section 5.2). Then, it induces a discussion of the design criteria that have been considered for its capture (Section 5.3), to finally conclude with the pre-processing pipeline for generating a dataset (Section 5.4).

5.1 Captured Data

The generated dataset is composed of images captured with a plenoptic camera and later processed with the software provided by its manufacturer (see Section 5.2). In addition, a stereo system is used to obtain natural depth information through the Semi Global Matching (SGM) algorithm [46] (see Fig. 5.1), that can be then re projected onto the light field camera pose.

The captured dataset consists of 24 sets of images of different scenes in laboratory conditions, in a static environment (i.e., constant lighting conditions, non-moving objects). Each set includes the following data of a single scene:

- Plenoptic image
- Plenoptic depth
- Natural image

As shown in Fig. 5.2, both plenoptic images show the microlens pattern that is projected over the sensor, whereas natural images represent the reconstructed natural image obtained at the

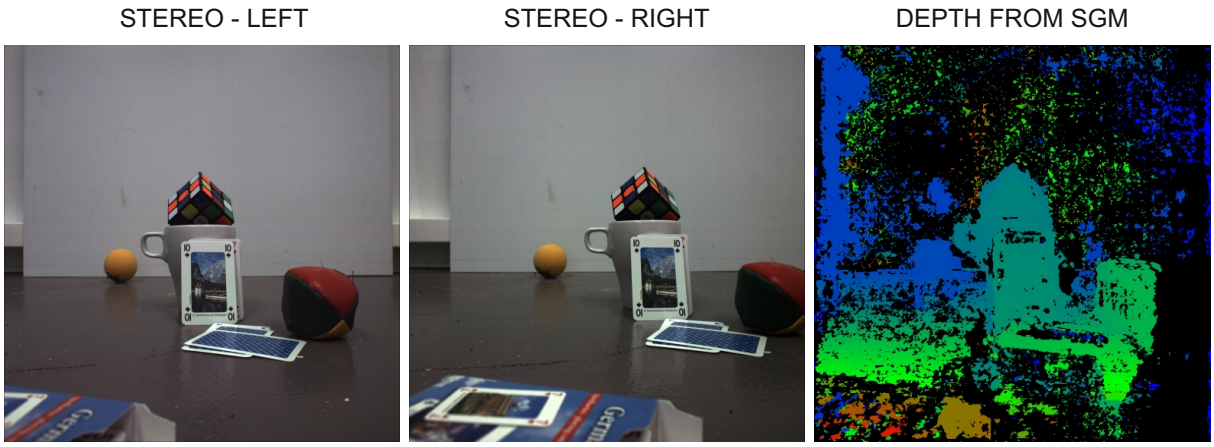

Figure 5.1: Images obtained from the stereo system employed in this work. From a stereo pair of images *(left and center)*, SGM allows the obtention of the natural, semi-dense, metric depth map *(right)*, projected onto the left camera's position.

central point of view of the camera – this is, the origin of its reference system. Both plenoptic depth and natural images are obtained through the manufacturer's reconstruction algorithms, which are not publicly available.

5.2 Experimental Setup

For the camera system to be suitable for data acquisition, the three cameras (i.e., the stereo pair and the plenoptic camera) need to have a representative percentage of co-visible areas of the same scene for it to be useful as a benchmark. This problem needs to be addressed in two complementary ways: First, through a setup that keeps the light field camera as close as possible to the stereo pair (i.e., one camera of the system) while keeping a rigid alignment and relative position during the whole capture, and also among different captures. It is also important that the configuration of the setup enables enough parallax between the stereo cameras so that depth can be correctly estimated. Second, it is highly important to do a proper choice of the set of lenses to use in each of the devices, since it is a decisive design choice for optimization purposes, as it produces the largest possible area of co-visibility for the three sensors (see Fig. 5.3).

In addition, the synchronization of the three devices for the scene capture is important since a temporal difference of milliseconds can be decisive in terms of accuracy for some depth estimation and image reconstruction algorithms. To ensure proper synchronization, a first approach is carried out by using a software-based synchronization and control tool that has been developed at DLR and integrated in the camera platform (see Section *Hardware*[1]). Then, a more fine-grained synchronization is carried out by hardware. For that purpose, a synchronization cable is specifically manufactured with the objective of triggering a signal between the light field camera

[1]The code is not publicly available due to the safety policy of German Aerospace Center. Please, contact the author of this report as well as their supervisors for further information.

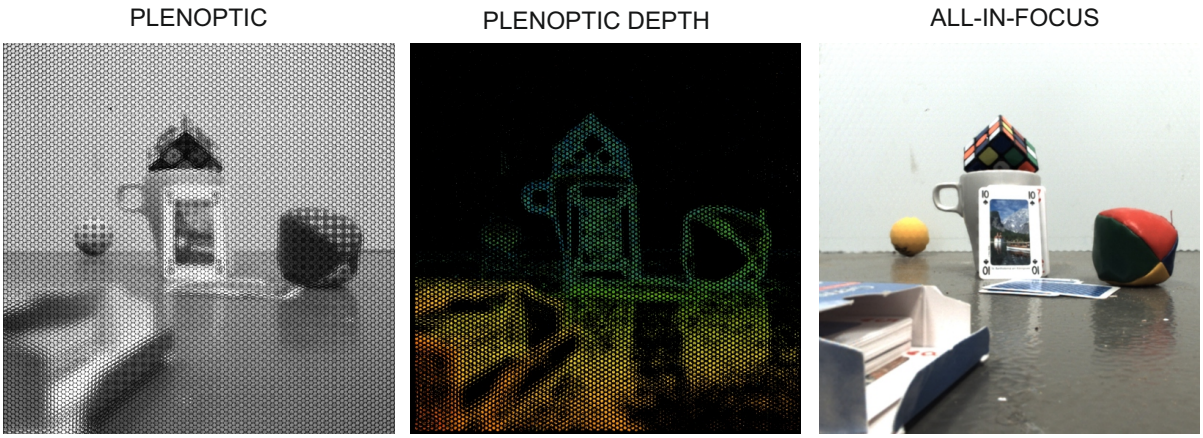

Figure 5.2: Images obtained after capturing a scene with R5 light field camera and RxLive software, both from Raytrix [93]. Plenoptic images show the projection of the array of microlenses over the sensor of the camera, whereas natural images are reconstructions of the natural images from the camera's central point of view.

and the stereo system, where one of the sensors (the plenoptic camera) would act as a master, and the others (the stereo system) would be slaves. In addition, the cable provides power and a common ground wire in case Power over Ethernet (PoE) is not available. This second synchronization option allows a significantly higher precision when carrying out the exposures, which has a high impact in the robustness and quality of the dataset.

5.2.1 Hardware

To establish a robust experimental setup to facilitate the capture, the development of a mechanical setup is carried out. This sturdy framework made of construction profiles has to be capable of maintaining the relative positions of all the integrated cameras within the system (see Fig. 5.3) to minimize the need of repeated extrinsic geometric calibration of the system components [105].

The cameras employed are two `Mako G-419C` cameras [113] and a `Raytrix R5` plenoptic camera [93]. The former device is a compact vision camera with a robust industrial cover at an appealing cost. The latter is a compact light field camera with up to 1 MP effective resolution and 4.2 Megarays light field resolution (i.e., the number of light rays that the sensor can capture). In addition, it has automatic and manual exposure, white balance and gain controls. All the cameras used have the same 12.7 mm CMOS CMV4000 image sensor at 4.2 MP resolution and 25 frames per second under the GigE vision standard.

5.2.2 Software

The whole software setup to configure, control and process the date of the aforementioned systems is deployed using the *Continuous Integration Software System (Cissy)* software, which

Figure 5.3: Camera setup with the stereo system and the light field camera, joined in a rigid structure to ensure the relative position during the whole capture time. It is important that the distance between the plenoptic camera and the reference camera of the stereo system (i.e., left camera) is minimized through this design, always taking the diameter of the lenses into account. In addition, the parallax between the stereo cameras has to be large enough in order to have an accurate triangulation at the desired focus plane.

is a package manager that provides a combination of several software engineering tools for smoothing the main development pipeline for software at the Robotics and Mechatronics Center at DLR. Additionally, the *Links and Nodes* software for system deployment is used to configure and launch the cameras, and to process their data, providing a clear overview of the running modules (i.e., viewers, image processing, camera configuration and control) using *SensorNet* during operation. In addition, it allows operating the hardware in order to capture the desired scenes, managing and storing the images, as well as saving real-time information about the devices being actively used.

Geometric camera calibration with DLR Calibration Detection Toolbox (DLR CalDe) and DLR Calibration Laboratory (DLR CalLab) [108] is carried out. The former allows the localization of corners of a chessboard 2D calibration panel with sub-pixel accuracy, whereas the latter uses the previously detected image features to obtain both intrinsic and extrinsic parameters of cameras, needed for further data processing (see Section 5.4).

Moreover, the software RxLive of the plenoptic camera manufacturer is used. It allows capturing of the plenoptic image as well as different visualizations (natural image, depth map, 3D reconstruction, point cloud, etc.) of the scene according to the proprietary reconstruction algorithms of Raytrix. In addition, metric stereo depth is reconstructed from the stereo pair using the SGM algorithm [46]. Finally, we developed some scripts for the extrinsic calibration

between cameras and to reproject images – either RGB or depth – into a different vantage point.

5.3 Scene Configuration

Setting up the capturing environment is a highly experimental and iterative process that has proven to have a significant impact in posterior computations. As far as the training process is concerned, it is a step that becomes crucial to allow the generation of a model with adequate learned inference and generalization capabilities. A number of aspects are taken into account for designing the captured scenes.

Shape and distance variability. The captured dataset encompasses variations in the shape and distance of the objects among scenes. If mostly planar surfaces are captured, a model would *overfit* to their specific depths, and predict a very similar depth score in consecutive areas. Therefore, the scenes have been designed introducing a variable number of objects, placed at different distances and in positions that avoid remaining planar with respect to the camera frame in most, or at least some, of the cases.

Texture. The use of objects with different textures would favor the generalization capabilities of any model trained on the dataset since it would learn to extract different features during the training process. Additionally, it is important to use different textures to enhance the robustness when combined with the stereo depth, which also has some limitations in specific kinds of surfaces. Objects with different nature have been used in the scene configuration, namely different rocks, foams, paper, carton, plastic, ceramics, or metals, all of them in objects with different roughness, color, or even text.

Maximum and minimum focusing distances. The depth of field is a function of focal length and aperture, as well as of the distance of the subject that is intended to appear focused. As in this dataset the scenes are static and the main lens used is fixed, it is important to define the range of distances that the plenoptic camera can cover in terms of focusing distance. Therefore, the *RxLive* software is used to assist the measuring, since it is able to provide the virtual distances that are possible to appear in focus in the scene (see Section 5.3 for further details).

Camera position. Optimizing the placement of the system is crucial to maximize the working area that is covered. To achieve this, it is positioned and oriented to exploit its entire depth of field based on the optics employed (see Section 5.2). Additionally, and prior to dataset capture, an experiment is conducted: A focal sweep across a scene is carried out, with items placed at various distances with respect to the camera, allowing for the determination of upper and lower limit working values. This method facilitates setting parameters up in the software to capture depth values within the physical limits by placing the camera at the furthest distance capable of being focused (100 cm), ensuring that depth information can still be effectively

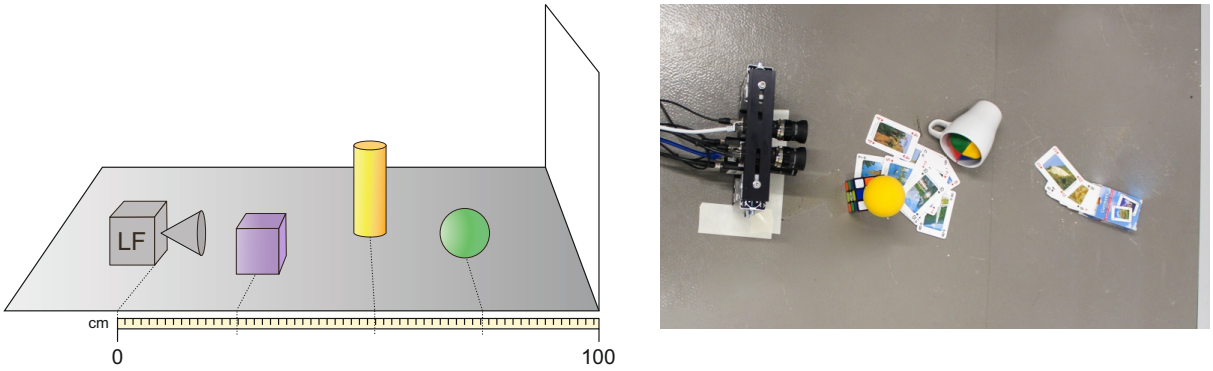

Figure 5.4: *Left:* Scene configuration scheme for the capture of the dataset. The camera is placed at a distance of 100 cm with respect to the background. Objects are placed in the range of 10-90 cm – the focusing limits – with respect to the camera center. *Right:* Picture of the top view of the laboratory setup.

obtained (see Fig. 5.4).

Up to date, as the dataset is taken in laboratory conditions, the illumination setting has not been changed to avoid significant changes in exposure times or the appearance of noise. However, this aspect should be explored in further steps of the process, since the versatility in different lighting conditions is one of the main advantages of a camera system.

5.4 Image Pre-processing Module

After the capture (see Section 5.1), the images obtained need to be processed to generate a dataset that is suitable for training a neural network. Hence, the development of the pre-processing module has the purpose of treating both plenoptic and stereo images before their input to the learning block of the pipeline of this project.

5.4.1 Plenoptic Images

The *Image Processing Toolkit* addresses the need for generating a dataset at the microlens scale from the plenoptic images captured. It consists of a parametric model to calculate and index the position of the projection of every microlens' centroid onto the sensor. This parametrization allows further operations such as disassembling an image into microlens cuts, or – the opposite process – reconstructing images by assembling lenslet images.

Hexagonal Grid Storage

In traditional n-dimensional square grids, associating elements with their indices is straightforward. However, the microlenses in the plenoptic camera follow an hexagonal packing arrangement [90] with an offset on even rows (see Fig. 5.5a), therefore vertically aligned, thus requiring

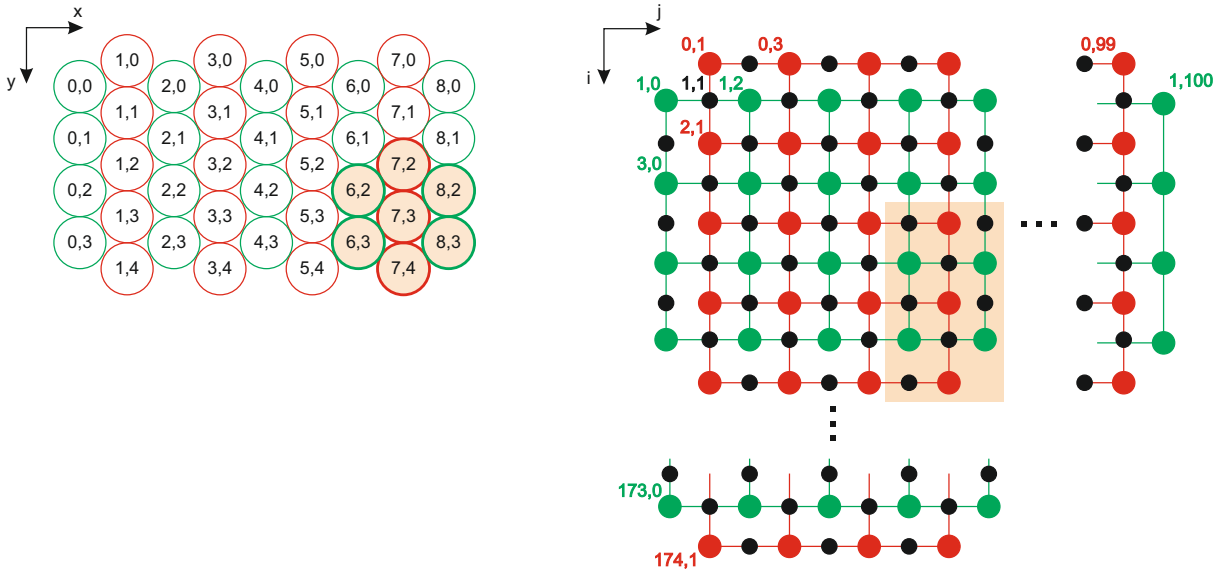

Figure 5.5: *Left*: Example of the hexagonal arrangement present in the used light field camera, with a vertical (i.e., column-wise) layout and offset in the even columns. *Right*: Resulting grid after merging two sub-grids, **A** (red) and **B** (green), following the checkerboard pattern, which doubles coordinates in rows and columns. The spots in black are the auxiliary cells that remain empty after such doubling. Note that coordinates are transposed in this arrangement as a convention for array indexing.

a different coordinate system for storage and usage.

Various coordinate systems exist for representing hexagonal arrangements [34]. In this work, the *doubled coordinate system* is chosen for its ease of implementation. This system, also known as *interlaced* or *checkerboard*, doubles either the horizontal or vertical step size (see Fig. 5.5a). While typically requiring an even number of columns and rows, the symmetric nature of the microlens array with respect to the camera main axis (z) in this work implies an odd number of columns and rows.

To accommodate this symmetry and for convenience in following steps, both horizontal and vertical components are duplicated, producing a checkerboard arrangement stored in a grid. This mesh represents a combination of two sub-grids $\mathbf{A}_{m\times n}$ and $\mathbf{B}_{p\times q}$, each containing every other set of rows (m and p) and columns (n and q) from the microlens array. These sub-grids are then merged into the aforementioned larger grid $\mathbf{G}_{(m+p)\times(n+q)}$ following the checkerboard pattern (see Fig. 5.5b). This approach facilitates indexing each pair of coordinates to a microlens based on its relative position on the grid, enabling its usage in further processing steps.

Parametric Centroid Calculation

To obtain the coordinates (x, y) of the centroid of each of the microlenses in the camera array, a parametric model is developed. This algorithm is designed to fit the grid based on seven parameters that define the lenslet arrangement: Namely, the pixel coordinates of the first two centroids a_{00}, a_{01} in the first row of grid **A**, and the first centroid of grid **B**; the horizontal (d_x)

and vertical (d_y) distances between consecutive *aligned* microlenses; and the number of rows (m, p) and columns (n, q) of both sub-grids.

The coordinates and grid shapes are obtained from a plenoptic image taken in laboratory conditions, applying a white diffuse filter to the camera lens. The resulting white plenoptic image can be easily used to identify the needed pixels due to the high resolution – 4 MP – of the camera sensor. The distances are obtained from calibration (see Section 4.3). Additionally, the skew angle α of the grid with respect to the camera reference can be provided as input if obtained from calibration. However, the model automatically calculates it if the aforementioned parameters are known.

With this information, the calculator generates both sub-grids with the coordinates of each of the centroids, and then merges them into the checkerboard grid such that the plenoptic image can be adequately stored and used in further steps.

Plenoptic Image Debayering

The original plenoptic image still contains the color information codified in 12 bits per pixel element, following the arrangement of photosensors of the color camera chip (Bayer pattern). A demosaicing algorithm is applied to transform – through bilinear interpolation – the GBRG Bayer pattern of the plenoptic image into a full-color image where each pixel has RGB values assigned.

Microlens Operations

The dataset generation implies the manipulation of the projections of microlenses in a plenoptic image to use them in the following steps of the pipeline. This is performed by two main operations:

Cropping. The microlens images are cropped according to their size in pixels and the previously calculated positions of each centroid in the array. The size of a microlens image needs to be previously known (again, it can be easily obtained from the plenoptic image after applying a white diffuse filter to the camera lens). For convenience, regarding their cropping, storage and posterior manipulation, the performed cut has the shape of a square, with a side length equal to the microlens diameter (in this case, 23 pixels). To avoid data-driven issues in following steps, the lenslets that are partially projected onto the sensor and, therefore, do not fully appear on the plenoptic image, are not considered in the computation. Thus, the borders of the image are either ignored, deleted or not saved.

Stacking. Due to the particular hexagonal arrangement of the grid, a microlens is surrounded by six additional lenses that follow the shape of a ring around the central one. This way, consecutive concentric rings can be traced with respect to the same microlens. By stacking

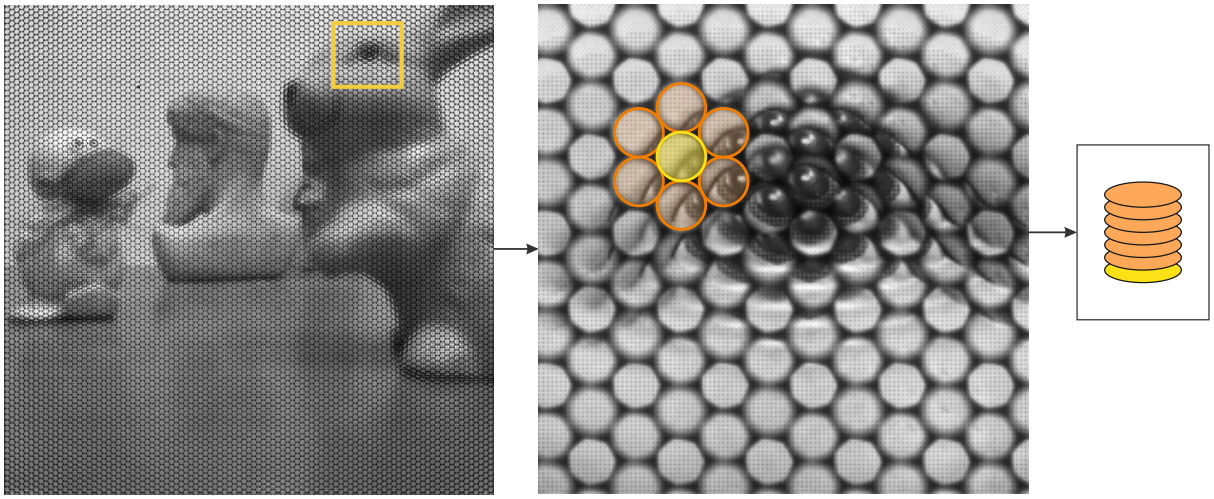

Figure 5.6: Processing of a plenoptic image to generate a dataset of *flower stacks*. Each stacked microlens is a debayered 3-channel RGB image.

the obtained cropped micro images following the ring arrangement along the channels dimension of the resulting tensor, a *flower stack* is generated and saved (see Fig. 5.6). This configuration can be replicated with the desired number of rings to be stacked and it proves to be very useful for the posterior learned feature extraction process (see Sec 6.2).

5.4.2 Plenoptic Depth

Additionally, the *plenoptic depth* information obtained through RxLive is processed since originally, due to the unavailability of the stereo system, it was used as an alternative way to obtain ground-truth depth.

These images are also processed using the *Image Processing Toolkit*, but it involves an additional tool for previously decoding the *plenoptic depth* information obtained from the manufacturer's software: Since it is unconventionally decoded in a 2-channel `.tiff` image format, the usage of Matlab becomes fundamental for an adequate reading of the values. In addition, the developed software needs to perform the transformation from virtual to metric depth making use of the thin lens camera model and the pinhole camera equations [107]. Moreover, the code provides plotting tools to visualize the transformed depth information in an overlying continuous color map over the *plenoptic* image (see Fig. 5.7). Finally, it is possible to store the obtained metric depth values for its posterior usage.

5.4.3 Depth from Stereo

The ground-truth depth used in this project is obtained through the stereo system, as it turned out to be more precise that the one obtained through the light field camera. This is due to the fact

Figure 5.7: Obtained maps showing metric plenoptic depths *(colored)* over their respective plenoptic image.

that the baseline in the stereo vision system is significantly larger than among the microlenses inside the light field camera. Additionally, the manufacturer's method for obtaining depth remains unknown. This lack of transparency makes it impossible to assess the uncertainties in their algorithm which is crucial for obtaining an accurate dataset.

The majority of the stereo processing pipeline (see Fig.5.8) is carried out by executing the aforementioned stereo processing packages fed by the image transport system *SensorNet*, configured with *Links and Nodes*, and managed by *Cissy*, as follows:

1. During capture, the images are debayered online and they are saved uncompressed in *Portable Pixel Map* (`.ppm`) format.

2. Each pair of images is rectified according to the previously performed geometric calibration with *DLR CalDe and DLR CalLab* [108]. The rectification parameters are stored for further steps.

3. The SGM stereo processing algorithm is executed for each pair of rectified stereo images, generating a disparity image (`.pfm, float32`) and its corresponding uncertainty map(`.pgm, mono16`) in the pose of the left camera.

4. The left rectified RGB image is re-projected onto the goal pose, which corresponds to the light field camera. The algorithm uses the computed disparities, the calibration files of both cameras with the poses needed for extrinsic calibration, and the rectification parameters with the intrinsic calibration and the baseline of the rectified stereo images.

5. As a result of the reprojection, both a color and a metric depth image are synthesized at the goal camera pose for each pair of rectified images.

The generated images have the appearance of natural images, although discontinuities can be noticed. This is caused by the framed scene, since the SGM algorithm might not be able to compute depth at every single pixel of an image. In addition, it is noticeable that there is a lack

of RGB and depth values in a specific area of the generated image: The differences in the areas framed by each system imply that only the co-visible area of the scene in both left stereo and light field cameras show generated image values.

Moreover, great noise can be observed in those areas from the obtained stereo depth where SGM is not robust (e.g., in planar diffuse surfaces such as the background or the floor). To reduce the ratio of outliers produced by noisy measurements, a texture filter is applied to the stereo depth. The texture filter applies a kernel of 10×10 pixels to locally compute the sum of gradients in the image, and then apply a threshold to remove textureless regions.

Still, after re-projection, the differences between a plenoptic and a natural projection do not make possible the direct translation of the depth values, but only those in the coordinates of the centroid at each microlens. In this case, the *Image Processing Toolkit* is used to take the metric depth values from the synthesized stereo depth at the centroids' coordinates (see Fig.5.8). To avoid the lack of information caused by the aforementioned discontinuities, a median filter is used. The obtained depths are stored for further stages in the general pipeline of the project.

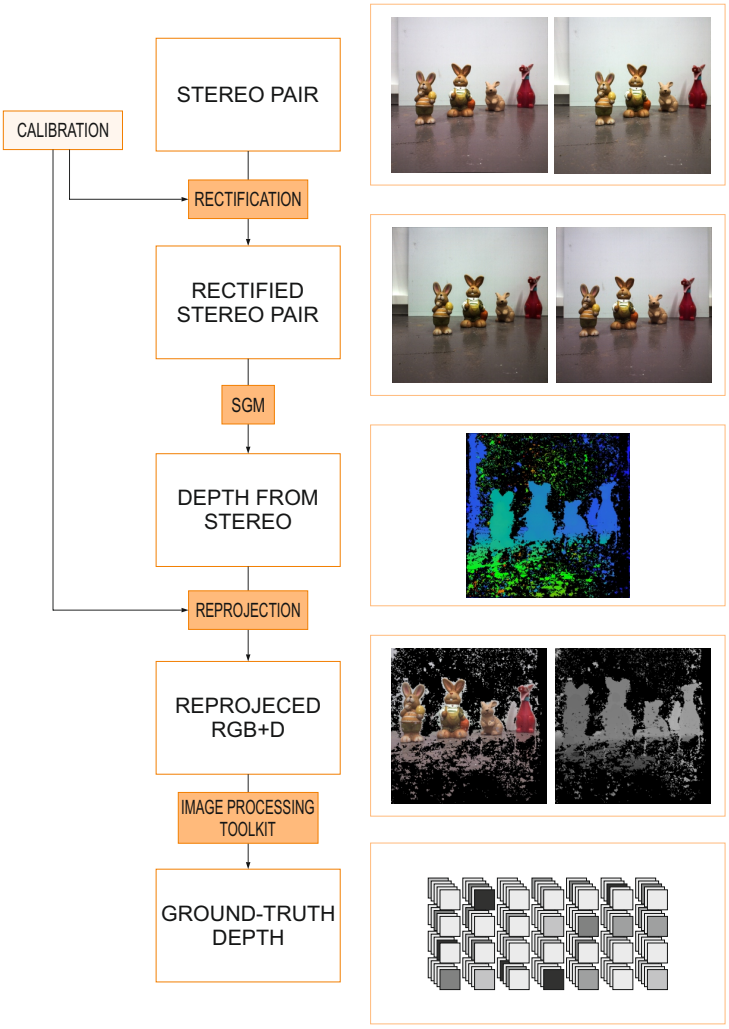

Figure 5.8: Processing pipeline and visual results to obtain suitable ground-truth depth values from stereo images.

6. Single-View Depth from Plenoptic Cameras

This chapter tackles the inference of depth from monocular light field cameras. Making use of the data and resources currently available, this project develops a modular pipeline to address this (Section 6.1). The high level of adaptability of the process makes it easily adjustable to similar problems that involve different elements from the ones used in this work (i.e., camera systems, optics, scenarios, or environment conditions among others), making it a versatile approach. This chapter explains the different modules that are used in the pipeline and address different parts of the problem after the generation of a dataset (Section 6.2). Last, details about implementation are provided (Section 6.3).

6.1 Overview

In this work, a complete pipeline is created with the objective of obtaining a dense representation that contains depth information inferred from a single raw light field image.

At an early stage of this project, an attempt of end-to-end learning-based approach was proposed. However, the difficulties encountered in the hardware (e.g., network switching, data transfer, online storage, synchronization, system mobility, capturing parameters, etc.) made it difficult to take a sufficient number of sequences that allowed to train a deep neural network using plenoptic images as input, and stereo depth as ground-truth data.

Nevertheless, the characteristic optical configuration of the camera can be exploited to generate an alternative set of data: A raw plenoptic image represents the projection of the scene captured through the array of microlenses placed in front of the camera sensor. This unique arrangement can be conceived as a multi-view stereo system at the microlens scale, with partial information of the scene gathered by each of these lenslets, thus allowing the extraction of depth information. This principle serves as inspiration for first generating a model that allows obtaining depth values for each projection of a microlens onto the sensor, to then perform the reconstruction of the scene by gathering the microlens information back [104].

The proposed alternative end-to-end pipeline consists of a succession of several computation steps that, in their combination, allow generating a continuous metric depth representation from captured light field data (see Fig. 6.1). This approach takes the aforementioned limitations into account and tackles them by concatenating modules for image processing, learning-based single-view depth estimation at the microlens scale, and a further densification and refinement of the resulting depth map. In this way, valid geometric model knowledge is introduced into the pipeline.

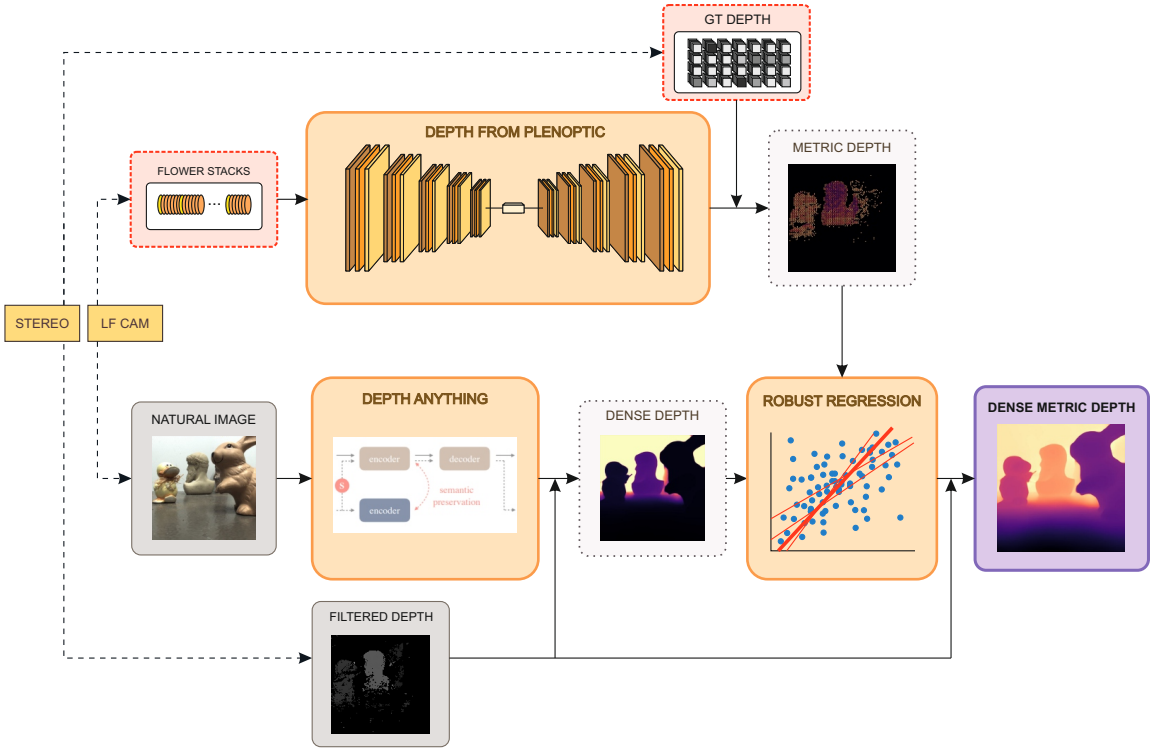

Figure 6.1: Overview of the pipeline for single-view depth estimation from light field images. Captured data needs to be pre-processed by the Image Processing Toolkit *(red)* to train the Microlens Depth Network *(orange)* for estimating sparse metric depth values that are then used to refine the estimates from Depth Anything *(orange)*. The result *(purple)* is a dense metric depth map in which darker colors represent objects closer to the camera, and lighter colors refer to elements placed further away. Filtered stereo depth *(gray)* is used as ground truth after both densification and scale alignment.

6.2 Depth Inference from Plenoptic Images

This section focuses on the development of a learning-based method able to infer metric depth information from a raw light field image provided as input.

6.2.1 Microlens Depth Network

An encoder-decoder architecture is developed to learn depth from light field (see Fig. 6.2). The input to the model is the aforementioned *flower stack*, which contains the RGB image crops of microlenses obtained from the plenoptic image (see Chapter 5). Flower stacks are provided to the encoder in the form of 4D tensors $X_i = (N, C, H, W)$ of batch size N, C channels, and $H \times W$ height and width of each microlens projection.

The proposed convolutional model infers a metric depth value for each stack of microlenses. Each output corresponds to a value prediction at the centroid's position of the main microlens (i.e., the central one) of each referred stack.

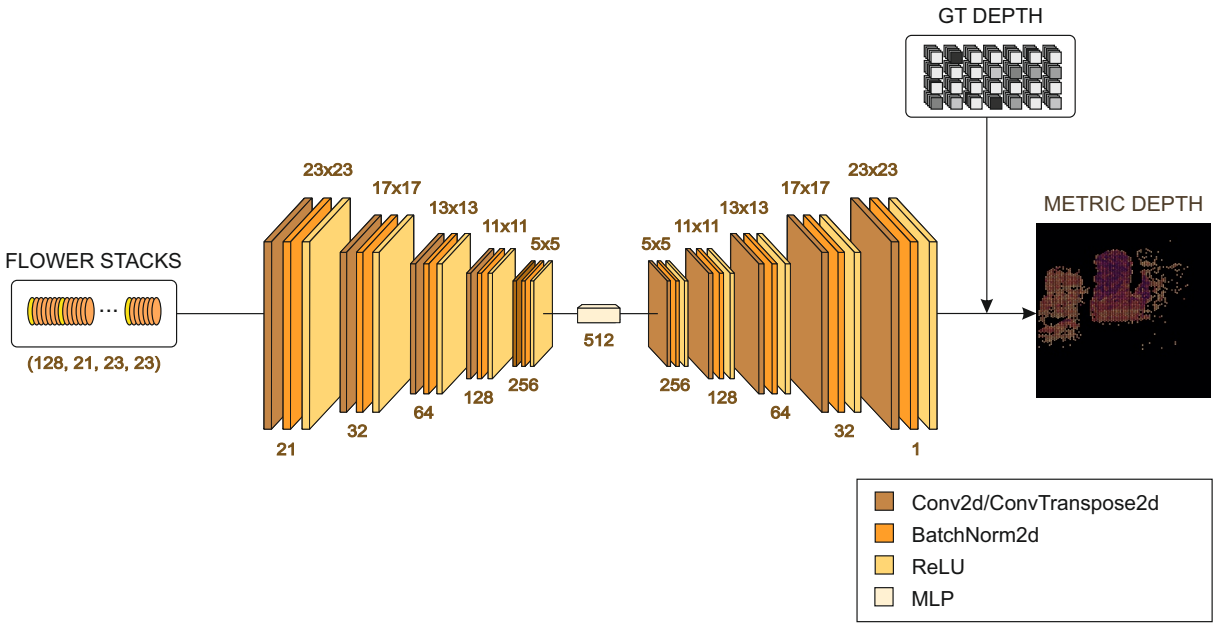

Figure 6.2: Architecture of the proposed encoder-decoder model based on convolutions, with MLP bottleneck, to predict depth from RGB flower stacks. Data shapes at each layer are specified

Convolutional Units

In this project, the network is formed by groups of layers that involve 2D-convolution operations, followed by *batch normalization*, which collects values over all spatial locations to compute the mean and variance and normalize the obtained value at each spatial location, and by an *activation function* that introduces non-linearities into the network, allowing it to learn complex correlations among data.

In the convolutional encoder, the network learns to extract features and then encode the correlations existing among the entire stack of micro images, depending on the relative position of each microlens. This method leverages the ability of CNNs to detect patterns and relationships within the data. By processing a central micro image together with its surrounding images, the network can learn to understand the spatial relationships for the features, and therefore interpret depth cues that are present in the scene.

Even if this setup introduces some redundancy in the information provided to the network, it can be beneficial as it enhances the network's robustness to occlusions. When certain areas are occluded in some views, the network can still infer depth information from the unoccluded views, therefore reducing the impact of occlusions. Therefore, by capturing and analyzing these redundant correlations among microlenses, the CNN could improve its performance in depth estimation even at the low resolution of the microlenses.

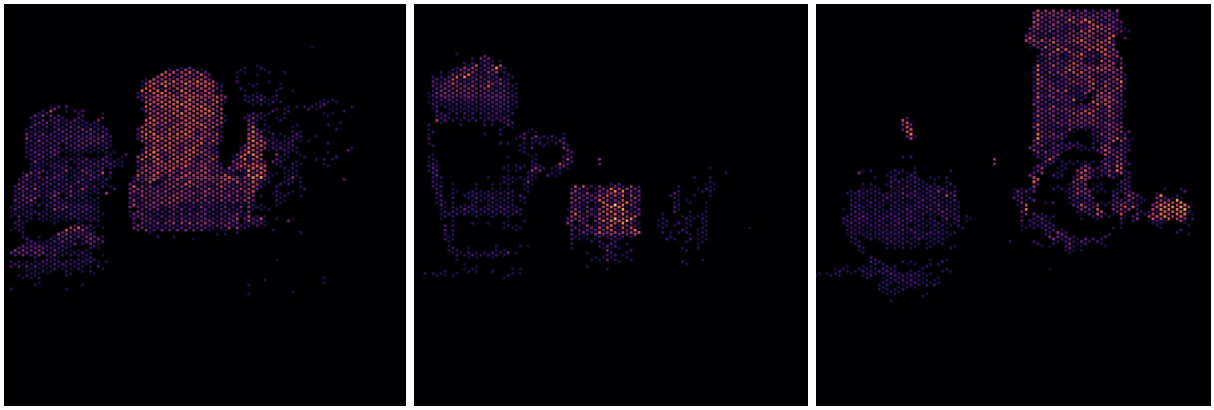

Figure 6.3: Reconstruction of the predicted depth values at the centroid of each microlens. The obtained result is a sparse depth map that shows the non-zero inferred values.

Depth Reconstruction

With the aforementioned estimations, a depth map related to the central sub-aperture image can be synthesized. This representation is equivalent to the natural image of the scene from the camera's central point of view, although at a low resolution, since only the predicted value at the centroid of each microlens is taken into consideration. Since the number of microlenses is smaller than the resolution of the sensor, the image obtained has a lower definition.

In particular, the camera used provides an array of 8,837 microlenses. Since the position of the centroids has been previously recorded onto a double coordinate grid system (see Section 5.4), it can be again used to generate the mentioned sparse depth map. This representation allows fitting the discrete values obtained to the desired output resolution. In this case, it is adjusted to the size of the synthesized natural image, with 1024×1024 pixels.

6.2.2 Scale Alignment with Plenoptic Sparse Depth

As seen in previous sections, the inferred depth is not dense, leading to a loss of information compared to a dense depth map of the scene. To address this issue, an additional module is introduced to densify the predictions while maintaining metric depth values. This module integrates a foundation model that generates a continuous, up-to-scale disparity map from monocular images.

To convert these disparity values into metric depth values, a robust regression method is employed. This method allows for the determination of a scale and offset that transform the dense estimation into the desired metric magnitude, therefore generating a more complete and accurate representation of the scene.

Depth Anything

Since there is not enough data to train a network that predicts dense depth from monocular images, a pretrained model is used. The chosen model must perform well with the kind of scenes that have been taken as part of the dataset. In this work, *Depth Anything* [121] is used to obtain disparity estimations from images captured through a monocular camera.

Depth Anything is a foundation model for high-quality monocular disparity estimation that is based on the DPT architecture, which leverages the Vision Transformer (ViT) as backbone for dense prediction tasks [92]. As a key aspect, this foundation model uses 1.5M labeled images from six public datasets to initially train a teacher model that assigns pseudo labels to nearly 62 million unlabeled images that have been collected from eight large-scale publicly available datasets through a data engine. This allows a student model to learn from the combination of both labeled and pseudo-labeled sets, which provide a wide variety of scenes and lighting conditions. Distinctively, the increased data coverage reduces the generalization error, which results in a good performance across a wide range of tasks and data types.

The model takes monocular images as input and provides disparity maps as output. The latter represent the information in a relative, up-to-scale manner. However, this result does not resolve the well-known scale issue of depth from monocular images. This is where the inferred metric information from the encoder-decoder CNN becomes crucial.

Robust Regression

To accurately align the scales between the dense depth predictions generated by Depth Anything and the ground-truth sparse depth values predicted by the Microlens Depth Network, we employ a robust regression approach.

In traditional regression methods such as ordinary least squares (OLS), the presence of outliers can excessively influence the model, leading to biased estimations. Robust regression techniques, however, alleviate this issue by reducing the influence of outliers and deviations from model assumptions, thus providing more reliable estimations in the presence of anomalies.

First, it is necessary to select the corresponding values available in both sets. To achieve this, the values from the dense prediction map are extracted at the coordinates where there is sparse depth information. This step isolates the relevant data points to be used for alignment. Additionally, we convert the sparse depth values into disparities. This transformation is crucial because the Depth Anything model operates in disparities, and aligning in disparities rather than depths helps to avoid propagating errors through the scaling process, making this approach more consistent. In addition, disparities are related to depth by the focal length and the baseline, which means that a linear transformation is now possible.

The Theil-Sen estimator [99] is used to perform the robust regression. This non-parametric technique is particularly robust to outliers, making it well-suited for applications where data might contain noise and anomalies. The Theil-Sen estimator fits a linear model by computing the median value m of the slopes of all lines through pairs of points, (x_i, y_i) and (x_j, y_j), in the dataset. To avoid indefinite slopes, the cases in which pairs of points share the same x values are skipped:

$$m = \text{median} \left\{ \frac{y_j - y_i}{x_j - x_i} \mid x_i \neq x_j \right\} \quad . \tag{6.1}$$

Once the Theil-Sen slope m is calculated, an intercept term b_i is calculated for each point:

$$b_i = y_i - m x_i \quad . \tag{6.2}$$

Then, the median value b of the intercept terms is taken:

$$b = \text{median}\{b_i\} \quad . \tag{6.3}$$

Thus, the linear model is represented as:

$$y = mx + b \quad . \tag{6.4}$$

This robust regression method ensures that outliers have minimal impact on the regressed line, resulting in a more accurate representation of the underlying relationship between variables, even when the data includes outliers or errors.

With this approach, the scale and offset are determined and applied to the dense predictions to obtain the appropriate scale. The refined predictions are then evaluated with respect to the previous ones through their conversion into the depth space.

6.3 Implementation Details

This section outlines the implementation details regarding the proposed depth estimation model, which uses an encoder-decoder backbone with a robust regression approach for scale alignment. The key components of the architecture, the training, and the evaluation processes are described below.

6.3.1 Architecture

The proposed model features an encoder-decoder architecture designed for depth estimation from flower stacks (see Fig. 6.2).

Encoder. It consists of five convolutional layers, each composed of a 2D-convolution operation, batch normalization, and a Rectified Linear Unit (ReLU) activation function. The encoder processes 4D tensors of shape $X_i = (N, C, H, W)$ as input, where the batch size is $N = 128$. The flower stacks are of size 23×23 pixels with seven RGB images each, resulting in $C = 21$ channels.

Bottleneck. After encoding the input, the information is passed to the bottleneck block, which consists of a multilayer perceptron (MLP). A flatten layer first converts the input tensor to 1D. Then, three hidden fully connected layers follow, each comprising a linear transformation and a ReLU activation function. These layers contain the compressed knowledge representations of the network. Finally, an output layer is added. Due to the lower-dimensional representation of the input data, the bottleneck can effectively capture and learn abstract feature representations. Therefore, using a MLP in the bottleneck can be convenient for capturing these high-level features, including finding correspondences and measuring disparities.

Decoder. The encoded representation from the bottleneck is then fed into the decoder, which reconstructs the compressed data back into the desired form – in this case, depth estimations – through five transposed convolution layers. These layers are symmetrical to the convolutional layers in the encoder and each includes a transposed convolution operation, batch normalization and a ReLU activation function. The only layer in the decoder that differs from its counterpart in the encoder is the last one, since it has been adapted to generate a single channel estimation corresponding to depth values.

6.3.2 Loss Function

With the main goal of the proposed network being to optimize itself (i.e., learn) to perform accurate depth estimates, the loss function plays a key role in the network's learning process. The loss function quantifies the difference between the network's predictions and the ground-truth data, computing residuals that indicate how much the predictions differ from the actual values. During training, the network performs a self-optimization by adjusting its weights to minimize the loss. This adjustment is achieved through backpropagation, where the gradients of the loss with respect to the network parameters are calculated and used to update these weights in a way that reduces the loss. This iterative process helps the network to progressively improve its performance, thus its ability to make more accurate predictions.

In this work, the loss function is calculated by computing the mean squared error (MSE),

which measures the pixel-wise error between the predicted values $\hat{y}$ and the ground-truth values y. However, this function is modified to avoid the backpropagation of non-measured ground-truth data. Some coordinates of the ground-truth depth have no measurements and these values are coded as zero (e.g., in areas where stereo matching couldn't find valid correspondences). To handle this, a mask M is applied to exclude these non-measured values from the loss calculation:

$$\text{MSE} = \frac{1}{\sum_{i=1}^{n} M_i} \sum_{i=1}^{n} M_i (\hat{y}_i - y_i)^2 \tag{6.5}$$

where M_i is a binary mask indicating whether the ground-truth value at pixel i is measured (1) or not (0). This ensures that only the measured ground-truth values contribute to the loss, improving the accuracy of the backpropagation process.

6.3.3 Training Details

The dataset used for training the model has been taken at DLR as a part of this work, as described in Chapter 5. It consists of 59 images of 8,837 microlenses each, resulting in 8,465 single-ringed flower stacks[1] of 7 RGB crops each, with a resolution of 21 × 21 pixels. The corresponding depth values are encoded with 16-bits per pixel. Since redundant microlenses may appear in different samples of stacks from one image, the train and test splits are done image-wise. Therefore, 49 images are selected as training data, while 10 images are held out for testing. This turns out to be a proportion of 83-17 % – close to an 80-20 % split, for convenience of having different types of scenes captured in both sets to maximize the generalization capabilities of the network, as well as to try the maximum variability of settings available during testing.

For training the model, the aforementioned samples are randomly gathered in batches of 128 stacks. The training process is carried out on a computing GPU cluster, providing various hardware configurations for this task. Specifically, the training has been conducted on a node with a Quadro FV100 Volta GPU with 32 GB of VRAM for 10.26 hours.

The training consisted of 125 epochs out of a default of 500 iterations. Early stopping was applied with a patience of 20 epochs for the validation loss using an callback based on the `EarlyStopping` class from the library PyTorch Ignite [25]. The learning rate was set to 0.001, and the Adam optimizer was utilized for optimization since it helps adjusting the learning rates for each parameter dynamically, promoting faster convergence and preventing oscillations during training [64].

[1] The generation of flower stacks is only carried out if all microlenses are within the image boundaries. Otherwise, the stack is discarded.

Table 6.1: Number of non-zero depth values per image, obtained from the Microlens Depth Network.

Image ID	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
# sparse values	1,900	1,700	2,983	1,805	1,627	614	918	1,905	2,088	1,781

6.3.4 Integration of Scale Alignment

To achieve dense depth inference using Depth Anything, 1024×1024 pixel RGB natural images are obtained from the camera manufacturer. For this step, a proprietary function [32] is utilized to generate a natural image from the light field camera's perspective, leveraging a model-based depth algorithm that is not open-source[2].

With the natural images, Depth Anything infers a continuous 518×518 pixel disparity map, which is then refined using the Theil-Sen method to robustly regress the scale and offset of the predicted disparities. For this process, predicted disparities from the foundation model that correspond to ground-truth depth values provided by the neural network are selected (see Table 6.1). Additionally, low disparity values are clipped to avoid extremely large depth values. Consequently, after clipping, 11,002 depth values from 10 images are used for alignment out of 17,321 sparse data points. The process was carried out using the same hardware mentioned above, with a mean execution time of 1.78 seconds per image.

[2]This step could be bypassed using additional hardware, which is discussed in Chapter 7. For convenience and due to the available resources at the moment of data capture, this alternative was not pursued as it was outside the primary focus of the thesis.

7. Evaluation

This chapter presents the results obtained from the implementation of the whole pipeline, using the captured data: It focuses on the metrics used to assess the model's performance; then, the results of the evaluation are presented; and finally, a study of additional experiments is conducted to analyze the contribution of different components of the model.

7.1 Metrics

To assess the performance of the proposed depth estimation model and refinement system, a number of commonly used metrics for monocular depth estimation are employed [22, 48, 42]. These metrics allow capturing different error characteristics, such as average errors (MSE, RMSE), relative errors (MARE, MSRE), and threshold-based accuracy metrics (Accuracy, BPR). The set of metrics chosen ensures that the model's performance is evaluated in detail from multiple perspectives.

To maintain consistency during the evaluation of different modules of the pipeline, a transformation is necessary after scale alignment. Depth Anything provides disparity values as output, and the regression is performed in this magnitude. However, for a more understandable comparison and to align with the previously obtained metrics related to the Microlens Depth Network, these disparities d are converted to metric depths Z using the intrinsic camera parameters obtained from metric calibration:

$$Z = \frac{Bf}{d}, \tag{7.1}$$

where f is the focal length and b represents the baseline.

Mean Squared Error (MSE). It is a standard metric that measures the average squared difference between the predicted $\hat{y}_i$ and the ground-truth depth values y_i. It is sensitive to large errors due to the squaring term, which makes it useful for penalizing significant deviations. In addition, if the error differences are Gaussian distributed, the minimization of this metric delivers zero bias, smallest variance, and consequently an optimal estimation of the model (maximum likelihood method).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad . \tag{7.2}$$

Root Mean Squared Error (RMSE). It measures the average difference between the model's predicted and actual values. It is the square root of the MSE, providing an error metric in the same units as the target variable (i.e., centimeters).

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad . \tag{7.3}$$

Mean Absolute Relative Error (MARE). It is calculated as the average of the absolute differences between predicted and true values, with respect to the latter. It provides insights of the average relative error, making it useful for understanding the proportion of error regardless of the actual depth values.

$$\text{MARE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{y}_i - y_i|}{y_i} \quad . \tag{7.4}$$

Mean Squared Relative Error (MSRE). It shares some similarity with MARE, but it squares the relative differences. This puts a stress into larger relative errors, thus highlighting their impact, which makes this metric specially interesting for obtaining precision in estimations.

$$\text{MSRE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2 \quad . \tag{7.5}$$

Accuracy. It measures the proportion of predicted depth values that are within a certain threshold δ of the ground-truth values:

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(\max\left(\frac{\hat{y}_i}{y_i}, \frac{y_i}{\hat{y}_i}\right) < \delta\right) \quad , \tag{7.6}$$

where $\mathbf{1}(\cdot)$ is an indicator function that returns a value depending on whether the condition is true (1) or false (0). This metric is useful for evaluating how well the predicted depths match the actual values with a set margin. It is often computed for different thresholds to obtain a detailed view on the behaviour and performance of the model. In this work, thresholds are set to common values in the literature (i.e., $\delta = 1.25$, $\delta^2 = 1.25^2$ and $\delta^3 = 1.25^3$).

Bad Pixel Ratio (BPR). It is a measurement of the proportion of pixels for which the absolute error exceeds a certain threshold τ. It is defined as:

$$\text{BPR} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\left(|\hat{y}_i - y_i| > \tau\right) \quad , \tag{7.7}$$

where $\mathbf{1}(\cdot)$ is an indicator function. This metric allows measuring the amount of significant prediction errors.

Table 7.1: Comparison of the proposed method (third and fifth rows) with additional baselines: A random generator (first), the manufacturer's depth calculation algorithm (second), and Depth Anything (fourth). Both sparse and dense modules developed in this thesis outperform the other baselines.

	MSE ↓	**RMSE** ↓	**MARE** ↓	**MSRE** ↓	**Accuracy** ($\delta = 1.25$)			**BPR** ↓
					δ ↑	δ^2 ↑	δ^3 ↑	
Random Depth	1551.39	39.27	1403.74	23.45	-	-	-	-
Raytrix Depth	136.47	10.94	65.43	2.70	**88.62**	**93.04**	**95.43**	**0.3043**
Ours Sparse	**124.68**	**5.55**	**52.75**	**2.63**	84.83	88.40	91.25	0.3949
Depth Anything	129.44	10.96	40.90	5.24	18.30	38.70	65.00	0.8623
Ours Dense	**83.21**	**8.50**	**37.30**	**3.94**	**46.40**	**74.60**	**90.00**	**0.4233**

7.2 Comparison against Baselines

The inference of sparse depth values took 10.54 minutes for a test set of 84,650 flower stacks, which corresponds to approximately 20% of the dataset (i.e., 10 images). The test images comprise different kinds of scenes designed with elements at various distances, in order to evaluate the network's ability to capture depth information across diverse scenes. Table 7.1 provides the quantitative results for the metrics explained in Section 7.1.

To address the lack of related works for comparison, and with the objective of validating the effectiveness of the developed methodology, a baseline model was first implemented. This approach involves the random generation of depth values as supervision, serving as an initial benchmark. The results are shown in the first row. Observe how the rest of the rows show significantly better results, showing the feasibility of learning metric depth with the baselines and our methods.

The second row displays the results obtained from using different sources of ground-truth depth in the Microlens Depth Network. The proposed method was also trained using ground-truth depth provided by the light field camera manufacturer. This comparison primarily involves differences in data treatment, including pre-processing of data, masking residuals for calculating the loss function, and slight modifications to the convolutional kernels in the decoding process. As observed in the third row, using stereo depth as ground truth improves the estimates of the proposed network, achieving a RMSE of 5 cm, which is considered a small value relative to the scale of scenes in the dataset, which is around 100 cm.

The fourth row presents results obtained from Depth Anything, which proves to be sensitive

to the unobservability of the scale. However, as shown in the fifth row, these metrics improve when our scale alignment is applied to the dense outcome of Depth Anything, using metric depth results from the Microlens Depth Network.

Both our sparse and dense solutions improve the results, compared to other baselines. As a consequence of the densification of predictions, a decline in metrics is noted in the latter two methods compared to the previously explained rows. Still, the developed method outperforms previous baselines and overcomes their limitations by providing a dense metric estimation.

Fig. 7.1 shows qualitatively the estimates obtained by the Microlens Depth Network, plotting the reconstruction of predicted sparse depth values. A threshold-based filter was applied to remove low depth value estimations, which are more abundant in the background, to in this way prevent error propagation in the densification module. Qualitative results of the scale alignment are shown in Fig. 7.2.

7.3 Further Analysis

In the previous section, the proposed model has been validated with respect to other baselines. For its development, several alternative or intermediate approaches have been explored to evaluate their performance and refine the final methodology. This section documents these approaches, highlighting their importance in the model development process. It is important to note that this section is not a strict ablation study but rather a narrative of the steps that have been taken alongside the process. The detailed results obtained from these models, as well as a discussion on the followed design criteria are explained below. Quantitative results are seen in Table 7.2.

Modifications in input data. Various input formats were explored alongside providing a flower stack as input to the network. The objective was to observe the network's behavior and its capability to extract features from differently treated data. Initially, *single micro images* were provided as input, leading to the exploration of three different architectures: Namely, utilizing a pre-trained model as an encoder (i.e., ResNet34 backbone without classification layers), the nominal encoder-decoder architecture, and a modified architecture with skip connections. Finally, *double-ringed flower stacks* were used in a similar network to the proposed one in order to determine if the architecture could effectively exploit the multi-view system with additional information.

The usage of a *single micro image* as input instead of a flower stack produces worse predictions, since the network is unable to learn relationships among different contiguous microlenses. In addition, three different architecture settings were tried with single images: Using a pre-trained model as an encoder does not seem to work well with these kind of images, since the features learned by the pre-trained model on natural images may not effectively transfer to the

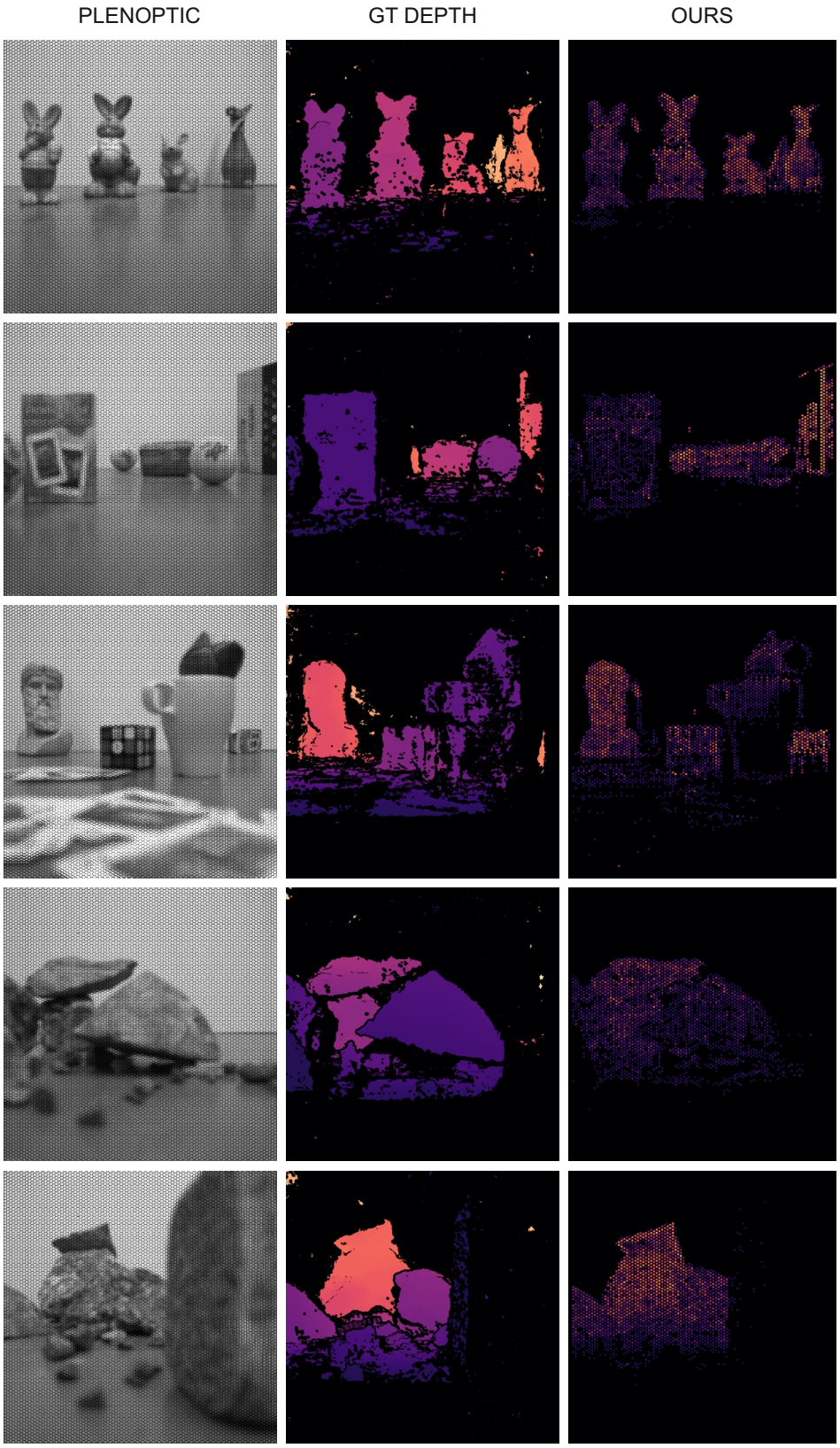

Figure 7.1: Comparison of results obtained from the Microlens Depth Network *(right)* with respect to the ground-truth depth *(center)*. After the reconstruction of the predicted sparse values on the coordinates of the microlens' centroid, low depth value estimations presented mainly in background surfaces are filtered to avoid the propagation of errors in subsequent modules of the pipeline. It can be observed that the depth values predicted by the developed model closely resemble the ground truth.

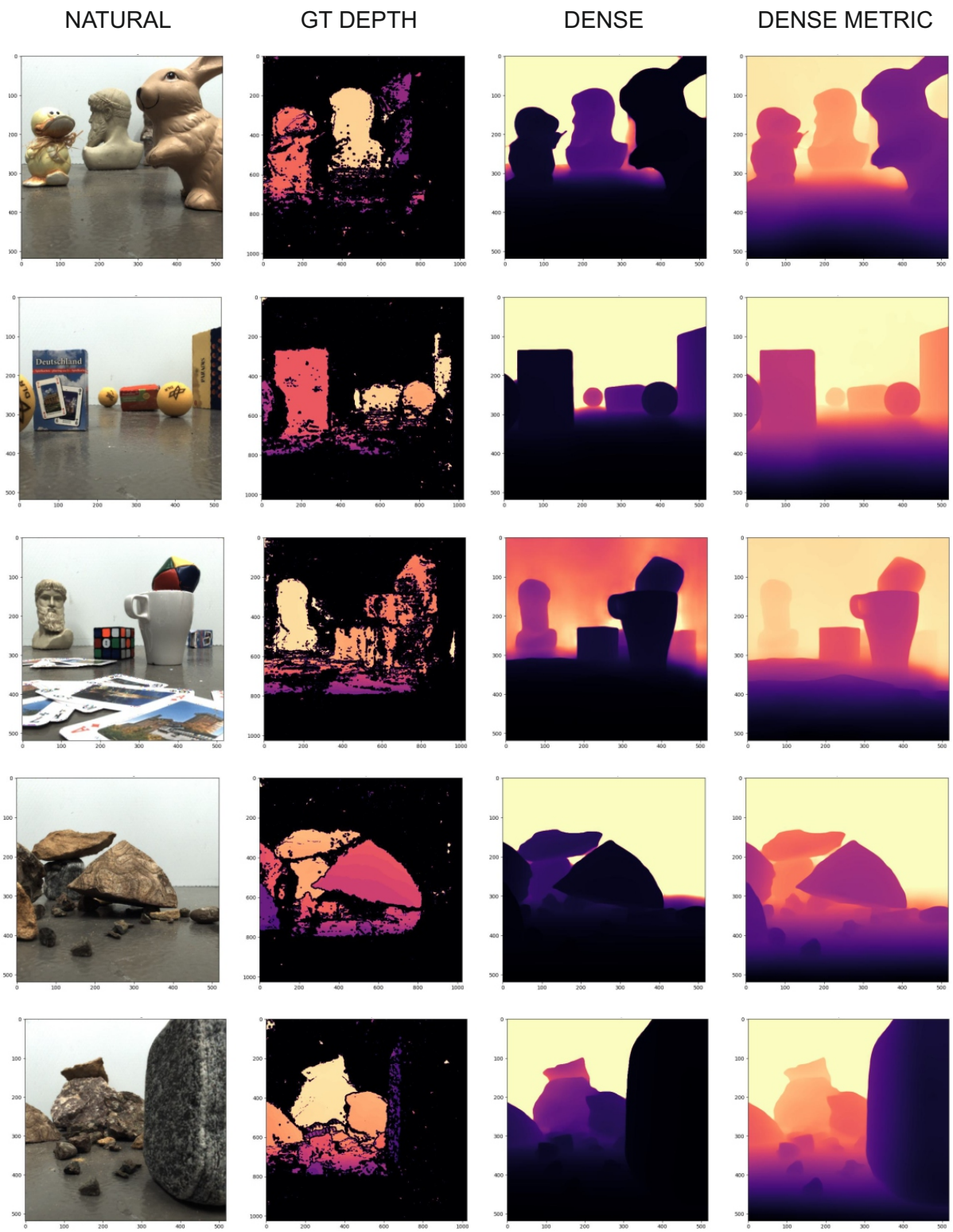

Figure 7.2: Obtained results after scale alignment. From left to right, the natural image of a scene *(first)* allows Depth Anything to perform a dense – although up-to-scale – prediction *(third)*. The scale alignment using metric values generates a dense metric depth map *(fourth)* of the scene. The color scale shows the improvement in the depth map after rectification through scale alignment, compared to the ground truth *(second)*.

Table 7.2: Error metrics for additional methods explored. The first and fourth rows present the refined methods. Each of these methods is then compared quantitatively to two other approaches. Both proposals outperform the alternative experiments carried out.

	MSE ↓	**RMSE** ↓	**MARE** ↓	**MSRE** ↓	**Accuracy** ($\delta = 1.25$)			**BPR** ↓
					δ ↑	δ^2 ↑	δ^3 ↑	
Ours Sparse	**124.68**	**5.55**	**52.75**	**2.63**	**84.83**	88.40	91.25	0.3949
Multihead Net	250.18	9.45	104.57	5.13	81.02	**89.74**	**93.03**	0.6872
Weighted Mask	171.57	12.30	80.82	3.64	83.43	88.32	93.01	**0.3802**
Ours Dense	**83.21**	**8.50**	37.30	3.94	**46.40**	**74.60**	**90.00**	**0.4233**
Huber	110.40	9.92	41.30	**2.96**	38.00	73.10	83.60	0.4758
SGD-Huber	110.75	9.94	**27.60**	3.12	43.40	73.60	87.90	0.4631

microlens data. In addition, implementing skip connections in the model can help to recover information in the reconstruction of the signal. However, while skip connections ease the flow of information and improve feature reuse, they do not fully compensate for the lack of multi-view context provided by the flower stack, resulting in non optimal depth estimations.

Moreover, the main optics utilized in the light field camera determine the maximum number of contiguous microlenses that contain co-visible features. In this work, the number amounts to two rings around a central microlens for the shorter measurable distances. An experiment is also carried out by feeding *double-ringed flower stacks* into the encoder-decoder architecture. However, its integration into the pipeline does not significantly contribute to the model's performance, while it expands its training and inference phases by five times. In addition, the storage of 19 microlenses per stack generates files with larger volumes, making it less efficient in terms of speed and storage space management (i.e., the dataset occupies more space, and there are many redundancies since one microlens appear in several stacks). Therefore, after trying different configurations, the single-ringed flower stack is chosen as input for the following models.

Modifications in the network's architecture. After setting the flower stack as the input format, and in order to explore the network's performance, different architecture designs have been explored. First, a fully-convolutional approach was proposed, involving an encoder-decoder backbone consisting of seven parallel encoders (see Fig. 7.3). Each unit processes a different microlens from the provided flower stack. Then, their outputs are aggregated into a vector in the latent space through concatenation. A convolution is applied in the bottleneck, which is then followed by a decodification into depth values. However, as shown in the first and second rows of Table 7.2, a single encoder seems to work better, since it allows the network to capture relatively more global context and relationships among the micro images of a flower stack, thus leading to an improvement of depth estimation accuracy.

Second, after the integration of fully connected layers in the bottleneck, and with the purpose of studying the network's ability to capture non-local information, the convolutional decoder is removed from the architecture. The purpose of this experiment is to evaluate whether a refined MLP could encode the complex relationships that were previously handled by the decoder.

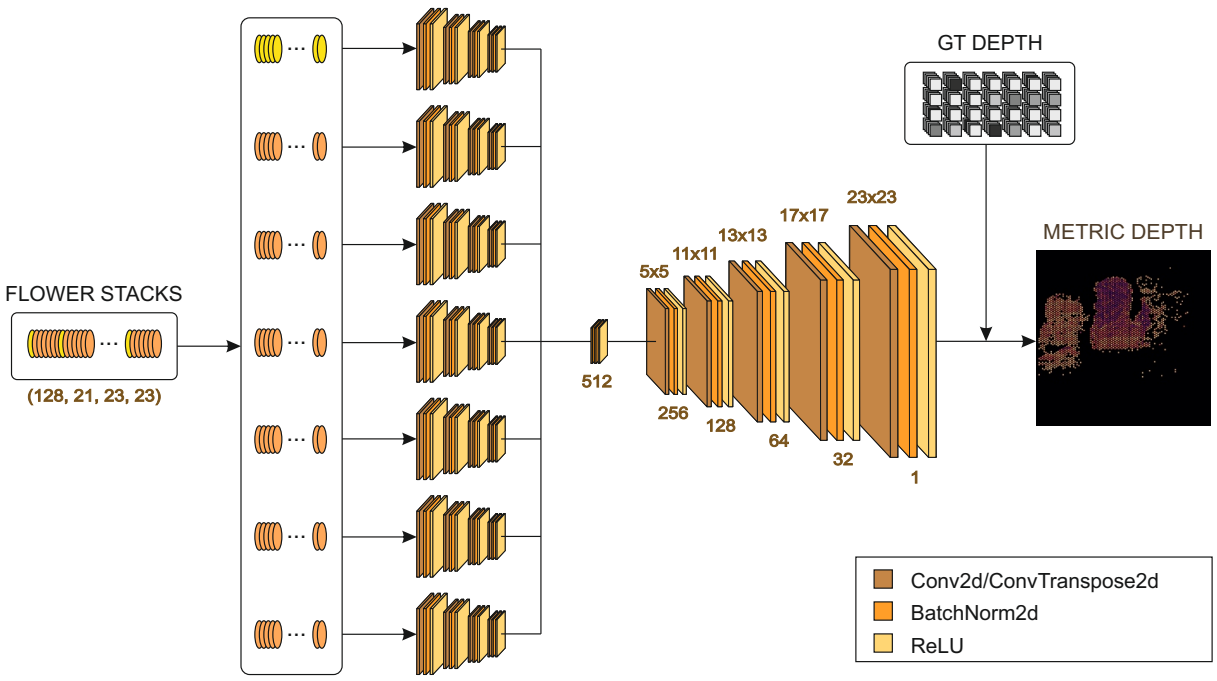

Figure 7.3: Nominal encoder-decoder architecture with seven encoding heads.

The substitution of the convolutional bottleneck by a fully-connected network seems to have a positive impact on the network's prediction capabilities. Introducing a MLP between the encoding and decoding modules aims to learn complex relationships that may be more general and not easily captured by convolutions, which are better in capturing local patterns. However, removing the decoder does not prove to be sufficient for the network to learn to codify the complex relationships that are demanded by the task. Therefore, a decoder is still necessary to reconstruct a signal similar to the one provided as an input.

Usage of different ground-truth data. The usage of ground-truth data from different sources is motivated by limitations in the availability of the hardware. Initially, ground-truth depth was obtained from the software provided by the light field camera manufacturer. However, this data is provided in virtual depths, requiring additional pre-processing steps and mathematical assumptions that lead to greater inaccuracies (see details in Section 5.4.2). Additionally, the geometric model used for depth estimation is closed-source, and errors in metric magnitude remained unknown.

This situation was turned into an opportunity to compare the differences between using depth data from a stereo system and from the manufacturer's software (see Table 7.1). The results when using stereo vision are better compared to the light field camera's virtual depths. This improvement could be caused by a superior performance of the semi-global matching (SGM) algorithm and the known error metrics associated with the stereo system. This may be related to the accurate geometric calibration of the stereo camera, allowing the generation of values in metric depth units, whereas the light field camera calibration delivered by the manufacturer

delivers unstable, inaccurate model parameters [70].

Modification of the loss function. The loss function was also tested without masking out 0 values during backpropagation. Depending on the format of the ground-truth data provided, the mask was either a single value or, as in the third row of Table 7.2, a circular mask that weighted the inside of the lenslet with respect to its outside in the crop. Additionally, the mean absolute error (MAE) was utilized as an alternative loss function.

It has been proven that using a loss function where void depth values are masked out results in better model predictions, as this approach prevents the backpropagation of those residuals. A variety of weighting schemes have been tested, and the most effective configuration has been found to be a binary mask when supervising with single depth values from the stereo system, or a custom weighted function when performing inference by reconstructing depths over the whole microlens (i.e., zero/void values are weighted less, and non-zero values are weighted more heavily). The implementation of this setting enables an optimal balance of the loss contributions from different parts of the data, which becomes reflected on the model performance.

Integration of Test-Time Refinement (TTR) in the pipeline. Inspired by the work of Izquierdo and Civera (2023) [53], the dense map estimation is developed by introducing their proposed method into the current pipeline. Their approach, which is compatible with any single-view depth network, has been adapted to integrate Depth Anything. The sparse point clouds originally obtained from SfM in the authors' method are replaced in this work by sparse depth values and coordinates obtained from the neural model. These values serve as a test-time self-supervisory signal: An alignment of the sparse data with the network's depth estimates is performed using random sample consensus (RANSAC) [24], with the objective of refining the network encoder and produce more accurate depth estimations.

Other Robust Regression Methods. In addition to RANSAC and Theil-Sen Regressions for scale alignment, other robust regression methods have been tested to evaluate their performance with the same data distribution. Specifically, two methods robust to outliers have been studied (see Table 7.2): the Huber Regressor and the Stochastic Gradient Descent (SGD) with a Huber loss function. The Huber Regressor optimizes the squared loss for inliers (i.e., samples below a threshold) and the absolute loss for outliers (i.e., samples above the threshold), while also optimizing the model's parameters, including the intercept and the scale. On the other hand, the SGD Regressor is a classical linear model fitted by minimizing a regularized empirical loss. It estimates the gradient of the loss one sample at a time and updates the model iteratively with a decreasing learning rate. The regularizer acts as a penalty added to the loss function to shrink the model coefficients towards zero. In this case, the L2 norm is used, adding a penalty equal to the sum of the squared coefficients.

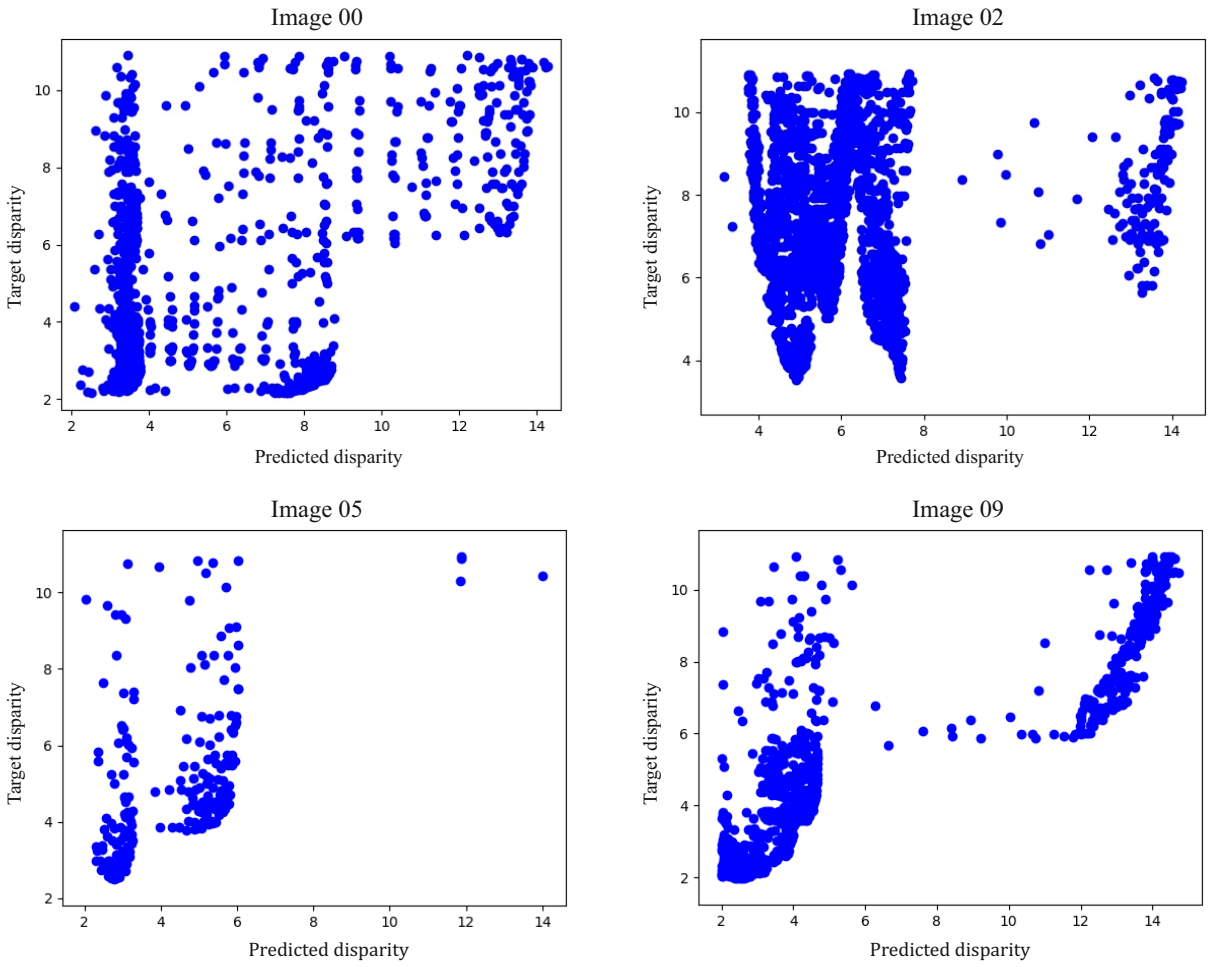

Figure 7.4: Analysis of the existing correlation between two sets of data: The obtained sparse disparity values (target) and the predictions by *Depth Anything*. The visualization provides auxiliary justification of the poor performance of the depth alignment method in this group of data, since significant amounts of noise can be observed.

7.3.1 Discussion on the Densification Module

The usage of SfM-TTR[1] has proven not to be the best alternative in this case, since the refinement does not provide an improvement to the predicted dense estimations. A further analysis has been carried out in order to understand the unsuccessful results obtained after the depth alignment. For that purpose, a study of the provided sparse depth information has been carried out.

Fig. 7.4 shows a scatter plot with sparse depth values and corresponding predicted values by Depth Anything in the same coordinates. This visualization shows that there is no linear correlation between both sets of data, as expected. In order to further understand the behavior of the obtained data, this same visualization is performed in the disparity domain, again proving no linear correlation. As a second step in the data exploration process, a detailed visualization of the sparse depths confirms that the values obtained exhibit a significant amount of noise, even if a texture filter is applied to them (see Fig. 7.5).

These conclusions lead to conduct further experiments on the ground truth used for training the network. Fig. 7.5 clarifies that, when plotting the correlations between stereo depth and the corresponding predicted values, even if a clearer linearity can be observed, there is an abundant amount of noise that is introduced in the training and evaluation process of the proposed neural network.

The reason behind this could be caused by the high amount of textureless regions – specially challenging for SGM – that are found in the pictures of the dataset: Even if the captured images are real-world, the environment is still limited to laboratory conditions. This lack of textures supposes a challenge because it generates highly empty regions in the image, and a notable amount of noise caused by wrong measurements. Texture filters have been applied to the sparse depth predictions to obtain denoised ground truth, but still the computation of RANSAC for the depth alignment generates a very biased and noisy prediction.

As an alternative, robust-to-outliers regression methods are tested. As seen in Table 7.2, Theil-Sen method outperforms the other proposals, therefore obtaining an improvement with respect to the initial estimation.

Even if it seems that the last step of the pipeline produces deterioration of the metrics with respect to the values obtained from the Microlens Depth Network, this result is still inside the expected behavior of the whole pipeline: The densification process generates values for the whole image, which could significantly influence metrics. Then, obtained results experience an improvement after the refinement.

[1]Some pretrained built-in models were also tested, but they yielded less satisfactory results. These models, trained on smaller or more specific datasets (e.g., KITTI, which focuses on road scenes), did not generalize as well to different environments and conditions. Therefore, Depth Anything was chosen for its robustness and broad applicability.

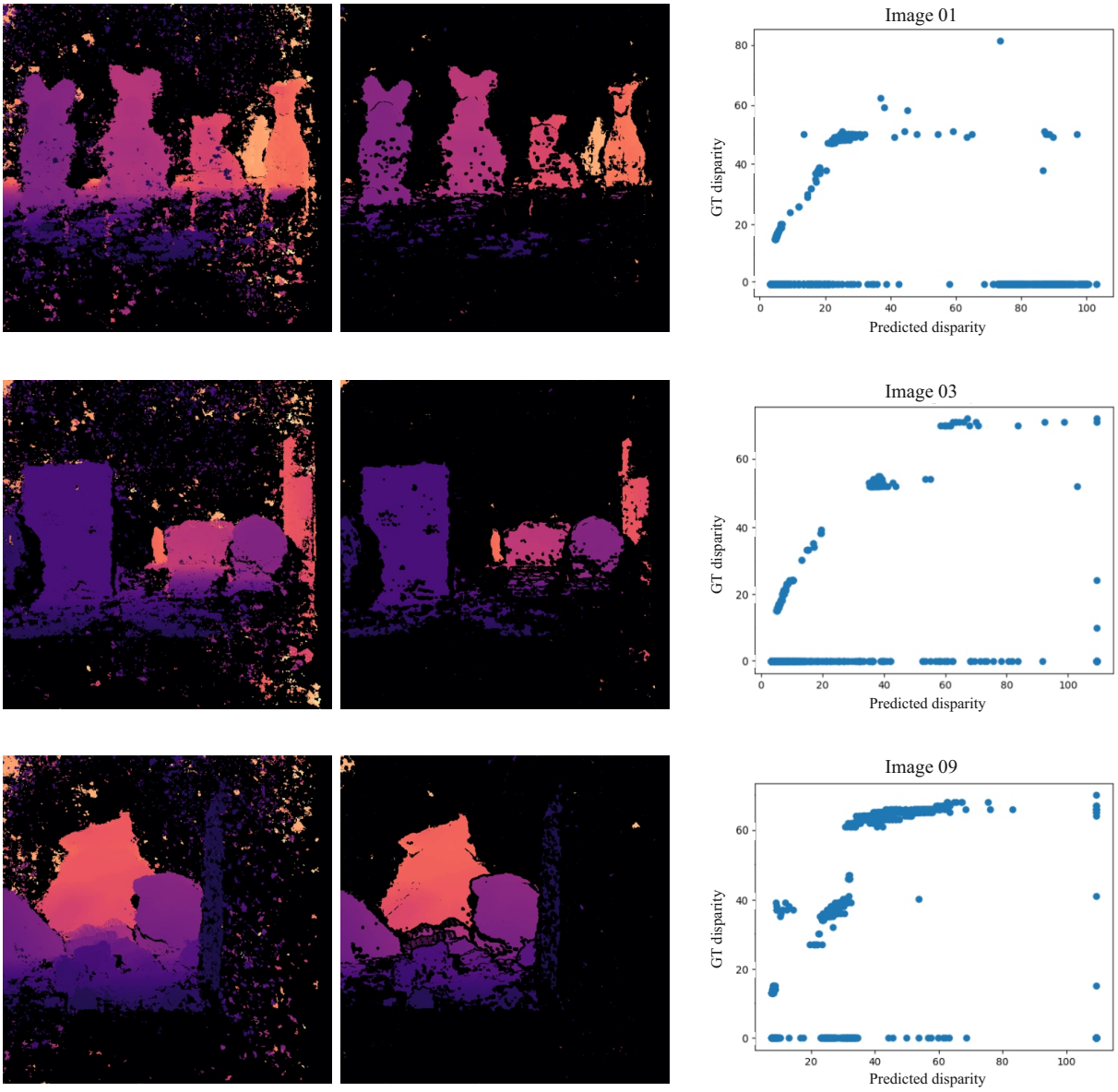

Figure 7.5: Visualization of the ground-truth disparity map before (*left*) and after (*center*) applying a texture filter to eliminate the existing noise. The large noise rate makes the filtering still not sufficient to obtain a refined sparse set of values, as it can be observed in the plotted correlation between the filtered ground-truth disparity and the prediction of Depth Anything (*right*).

8. Conclusions

This chapter summarizes the work carried out, highlighting the key contributions and noting the limitations encountered. Additionally, it outlines potential directions for future research.

8.1 Summary

This master thesis has involved a comprehensive study of the state of the art in light field imaging, with a special focus on the single-view depth estimation problem. A learning-based method has been proposed to address this challenge, specifically adapted to plenoptic cameras 2.0. These cameras leverage their internal configuration of microlenses in an array setting to obtain depth cues, which are learned by a convolutional neural network with an encoder-decoder backbone. A geometry-based approach was also considered. However, the camera model is not publicly available. The proposed approach generates sparse metric depth values that are then densified and refined by an additional learning-based module. This method allows the generation of dense metric depth maps from raw plenoptic images in an end-to-end manner.

8.2 Contributions

This work has involved the following contributions:

- The design and development of a learning-based Microlens Depth Network.
- A method for obtaining dense metric depth maps from single-view images.
- A new dataset which contains plenoptic images from a state-of-the-art focused light field camera, and metric depth labels obtained from a stereo system.
- The development of an auxiliary pre-processing methodology to make the captured data suitable for learning-based applications, and to assist as a toolbox for plenoptic imaging.

Given the novelty of the proposed method and the uniqueness of the studied problem and its related challenges, it results challenging to find comparable methods that allow evaluating the performance of the proposed alternative. As a result, an analysis of different proposed architectures, datasets, and modifications was carried out to establish both a course of development and a baseline for future research in the field.

The results obtained demonstrate that metric depth can be inferred from raw plenoptic images captured by a single-view device. This capability can offer significant advantages for various application fields.

8.3 Limitations

The work carried out aims to provide a starting point in exploiting the capabilities of the focused plenoptic camera for single-view depth calculation in a learning-based manner. This methodology is still at an initial development stage; therefore, it provides significant room for improvement and further research.

While the proposed end-to-end pipeline shows promising results for the studied device, it is not yet fully optimized for variable conditions such as images obtained from a different camera model (i.e., a different configuration of the microlens array), or with different optics. A future refinement of each stage of the pipeline could provide better performance and robustness to the method.

In addition, addressing the problem at the microlens scale makes the dataset suitable for learning, as well as a versatile method that can be adjusted to any plenoptic camera. However, this feature makes the pipeline highly dependent on the system's metric calibration, specially in the pre-processing module for dataset generation after image capture. The creation of an extensive dataset that includes large sets of full-frame raw plenoptic images and their respective labels from a synchronized stereo system could help to overcome this limitation and train a model that learns at the macro scale instead.

Moreover, it has been observed that the method's performance is highly influenced by environmental factors that influence the captured images, such as surface textures, object shapes, occlusions, lighting conditions, and depth ranges. These constraints need to be taken into account to obtain lower noise levels when capturing a dataset, which could result in a better performance of the model.

Even if the majority of the pipeline is self-standing, the generated natural image used to feed to Depth Anything still relies on the manufacturer's software. This issue was not tackled due to time constraints. However, it could be addressed by closely attaching a monocular camera to the body of the light field camera, computing the dense depth map for its pose and then reprojecting the obtained values into the light field camera pose. By reducing this dependency, a more versatile approach could be generated.

As far as the hardware is concerned, an encountered limitation is the requirement of specialized hardware and, especially, light field cameras, which might not be affordable for every institution. This constraint affects both the practicality of its research and the widespread

adoption of any developed method that requires this device. Also, the availability and the synchronization of hardware have been limiting factors throughout this work. Issues related to this topic have constrained the scope and scale of the data collecting process. For that purpose, if available, further research needs to be made in deploying a robust system that allows an efficient and precise capture.

A main technical advantage of the technology behind the light field camera lies in its ability to capture a greater amount of photons by projecting the image in a different plane to the sensor's. This enables the possibility of reducing shutter time and increasing the speed of movement of the camera. However, this advantage is counteracted by the loss of the full-frame RGB representation, as well as by an increase in the computational complexity for processing data.

8.4 Future Work

This work provides a first approach to single-view depth estimation from light field data, addresses the initially proposed objectives, and proposes alternative solutions to the limitations encountered during its development process. However, throughout the project, a number of potential ideas of possible further improvements and contributions have emerged:

It has been observed that a further refinement of the neural network could provide an improvement on the obtained results, as well as the implementation of different architectures that could exploit the particular arrangement shown in these cameras. In addition, it would be a good practice to review both pre-processing pipelines, with special attention into the stereo depth, including an accurate fine tuning of SGM hyperparameters, a study of the reprojection and computation of metric depths, and a capture of data in more optimal conditions (e.g., using outdoor environments, densely annotated images, etc.) to avoid noise introduction in the pipeline (*garbage in, garbage out*).

Exploring the integration of full-frame images within the pipeline could leverage the usage of microlens data by providing a broader understanding of the context. This could potentially lead to a subsequent development of a more sophisticated neural architecture that gathered more complex data patterns, thus enhancing its generalization capabilities and accuracy levels.

Despite the encountered limitations, the light field camera still can have a significant potential in several application domains. For instance, its integration into robotic exploration could revolutionize vision systems in rovers and other devices. The inclusion of these devices could offer greater flexibility and versatility in the way images are captured, allowing handling occlusions and other challenges constrained by the baseline of a stereo system (e.g., looking through narrow gaps). In addition, this technology could offer an alternative to reduce the reliance on conventional stereo systems by having lower space requirements. By using a single device, issues

related to the synchronization of multiple devices are avoided, and the limitations related to malfunctioning in one camera of a stereo system can be overcome.

Bibliography

[1] Raj Shah Abhilash Sunder Raj, Michael Lowney and Gordon Wetzstein. Stanford lytro light field archive. `http://lightfields.stanford.edu/LF2016.html`. [Last access 2023-08-29].

[2] Andrew Adams. The stanford multi-camera array. `https://graphics.stanford.edu/projects/array/`. [Last access 2023-08-28].

[3] Edward H Adelson and John YA Wang. Single lens stereo with a plenoptic camera. *IEEE transactions on pattern analysis and machine intelligence*, 14(2):99–106, 1992.

[4] Vamsi Kiran Adhikarla, Marek Vinkler, Denis Sumin, Rafał Mantiuk, Karol Myszkowski, Hans-Peter Seidel, and Piotr Didyk. Towards a quality metric for dense light fields. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[5] Waqas Ahmad, Luca Palmieri, Reinhard Koch, and Märten Sjöström. Matching light field datasets from plenoptic cameras 1.0 and 2.0. In *2018-3DTV-Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4. IEEE, 2018.

[6] Anna Alperovich, Ole Johannsen, Michael Strecke, and Bastian Goldluecke. Light field intrinsics with a deep encoder-decoder network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9145–9154, 2018.

[7] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4009–4018, 2021.

[8] Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[10] Billy Chen, Daniel Horn, Gernot Ziegler, and Hendrik PA Lensch. Interactive light field editing and compositing.

[11] Chunsheng Chen, Yongli He, Huiwu Mao, Li Zhu, Xiangjing Wang, Ying Zhu, Yixin Zhu, Yi Shi, Changjin Wan, and Qing Wan. A photoelectric spiking neuron for visual depth perception. *Advanced Materials*, 34(20):2201895, 2022.

[12] Jiaxin Chen, Shuo Zhang, and Youfang Lin. Attention-based multi-level fusion network for light field depth estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1009–1017, 2021.

[13] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[14] Bryan Chiang and Jeannette Bohg. Monocular depth estimation and feature tracking. *Stanford Education*, 2022.

[15] Alejo Concha and Javier Civera. Rgbdtam: A cost-effective and accurate rgb-d tracking and mapping system. In *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 6756–6763. IEEE, 2017.

[16] Donald G. Dansereau, Bernd Girod, and Gordon Wetzstein. LiFF: Light field features in scale and depth. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019.

[17] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1027–1034, 2013.

[18] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams. Linear volumetric focus for light field cameras. *ACM Trans. Graph.*, 34(2):15–1, 2015.

[19] S Tejaswi Digumarti, Joseph Daniel, Ahalya Ravendran, Ryan Griffiths, and Donald G Dansereau. Unsupervised learning of depth estimation and visual odometry for sparse light field cameras. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 278–285. IEEE, 2021.

[20] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[21] AWN Staff Editor. Lytro cinema brings revolutionary light field technology to film and tv production, 2016. Accessed: 2023-11-22.

[22] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in neural information processing systems*, 27, 2014.

[23] Mingtao Feng, Syed Zulqarnain Gilani, Yaonan Wang, and Ajmal Mian. 3d face reconstruction from light field images: A model-free approach. In *Proceedings of the European conference on computer vision (ECCV)*, pages 501–518, 2018.

[24] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[25] V. Fomin, J. Anmol, S. Desroziers, J. Kriss, and A. Tejani. High-level library to help with training neural networks in pytorch. `https://github.com/pytorch/ignite`, 2020.

[26] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2002–2011, 2018.

[27] Stefan Fuchs, Sami Haddadin, Maik Keller, Sven Parusel, Andreas Kolb, and Michael Suppa. Cooperative bin-picking with time-of-flight camera and impedance controlled dlr lightweight robot iii. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4862–4867. IEEE, 2010.

[28] David Gallup, Jan-Michael Frahm, Philippos Mordohai, and Marc Pollefeys. Variable baseline/resolution stereo. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[29] Todor Georgiev and Andrew Lumsdaine. Superresolution with plenoptic 2.0 cameras. In *Signal recovery and synthesis*, page STuA6. Optica Publishing Group, 2009.

[30] German Aerospace Center (DLR). About us. `https://www.dlr.de/en/dlr/about-us`, 2024. Accessed: 2024-06-13.

[31] Alireza Ghasemi, Nelly Afonso, and Martin Vetterli. Lcav-31: a dataset for light field object recognition. In *Computational imaging XII*, volume 9020, pages 283–290. SPIE, 2014.

[32] Raytrix GmbH. *Raytrix Light Field API Computation - Detailed*, 2015. Accessed: 2024-02-05.

[33] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3828–3838, 2019.

[34] Paul Hadley. Hcp structure. Accessed: 2023-08-13.

[35] Christopher Hahne and Amar Aggoun. Plenoptisign: An optical design tool for plenoptic imaging. *SoftwareX*, 10:100259, Jul 2019.

[36] Christopher Hahne and Amar Aggoun. Plenopticam v1. 0: A light-field imaging framework. *IEEE Transactions on Image Processing*, 30:6757–6771, 2021.

[37] Christopher Hahne, Amar Aggoun, Vladan Velisavljevic, Susanne Fiebig, and Matthias Pesch. Refocusing distance of a standard plenoptic camera. *Optics Express*, 24(19):21521–21540, 2016.

[38] Christopher Hahne, Amar Aggoun, Vladan Velisavljevic, Susanne Fiebig, and Matthias Pesch. Baseline and triangulation geometry in a standard plenoptic camera. *International Journal of Computer Vision*, 126:21–35, 2018.

[39] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. Matchnet: Unifying feature and metric learning for patch-based matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3279–3286, 2015.

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

[41] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *Proceedings of the IEEE international conference on computer vision*, pages 1586–1594, 2017.

[42] Caner Hazirbas, Sebastian Georg Soyer, Maximilian Christian Staab, Laura Leal-Taixé, and Daniel Cremers. Deep depth from focus. In *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*, pages 525–541. Springer, 2019.

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

[44] Stefan Heber, Wei Yu, and Thomas Pock. U-shaped networks for shape from light field. In *BMVC*, volume 3, page 5, 2016.

[45] 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*, pages 2252–2260, 2017.

[46] Heiko Hirschmüller. Semi-global matching-motivation, developments and applications. *Photogrammetric Week 11*, pages 173–184, 2011.

[47] 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.

[48] Katrin Honauer, Ole Johannsen, Daniel Kondermann, and Bastian Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part III 13*, pages 19–34. Springer, 2017.

[49] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2821–2830, 2018.

[50] Lam Huynh, Phong Nguyen-Ha, Jiri Matas, Esa Rahtu, and Janne Heikkilä. Guiding monocular depth estimation using depth-attention volume. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 581–597. Springer, 2020.

[51] Fraunhofer IOSB. Light field technology, 2023. Accessed: 2024-06-05.

[52] Frederic E Ives. Parallax stereogram and process of making same., April 14 1903. US Patent 725,567.

[53] Sergio Izquierdo and Javier Civera. Sfm-ttr: Using structure from motion for test-time refinement of single-view depth networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21466–21476, 2023.

[54] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020.

[55] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE international conference on computer vision*, pages 2307–2315, 2017.

[56] Adrian Johnston and Gustavo Carneiro. Self-supervised monocular trained depth estimation using self-attention and discrete disparity volume. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 4756–4765, 2020.

[57] Hyunyoung Jung, Eunhyeok Park, and Sungjoo Yoo. Fine-grained semantics-aware representation enhancement for self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12642–12652, 2021.

[58] Nima Khademi Kalantari, Ting-Chun Wang, and Ravi Ramamoorthi. Learning-based view synthesis for light field cameras. *ACM Transactions on Graphics (TOG)*, 35(6):1–10, 2016.

[59] Sing Bing Kang and Richard Szeliski. Extracting view-dependent depth maps from a collection of images. *International Journal of Computer Vision*, 58:139–163, 2004.

[60] Sing Bing Kang, Richard Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[61] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. *Advances in neural information processing systems*, 30, 2017.

[62] 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*, pages 66–75, 2017.

[63] Changil Kim, Henning Zimmer, Yael Pritch, Alexander Sorkine-Hornung, and Markus H Gross. Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph.*, 32(4):73–1, 2013.

[64] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[65] Stanford Computer Graphics Laboratory. The (old) stanford light fields archive. `https://www.theverge.com/2018/3/27/17166038/lytro-light-field-camera-company-shuts-down-google-hiring`, 1996. [Last access 2023-07-22].

[66] Blanca Lasheras-Hernandez, Belen Masia, and Daniel Martin. Drivernn : Predicting drivers' attention with deep recurrent networks. In *Spanish Computer Graphics Conference (CEIG)*, 2022.

[67] Mikael Le Pendu, Xiaoran Jiang, and Christine Guillemot. Light field inpainting propagation via low rank matrix completion. *IEEE Transactions on Image Processing*, 27(4):1981–1993, 2018.

[68] Nianyi Li, Jinwei Ye, Yu Ji, Haibin Ling, and Jingyi Yu. Saliency detection on light field. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[69] Zhenyu Li, Xuyang Wang, Xianming Liu, and Junjun Jiang. Binsformer: Revisiting adaptive bins for monocular depth estimation. *arXiv preprint arXiv:2204.00987*, 2022.

[70] Martin Lingenauber, Dominic Seyfert, Christian Nissler, and Klaus H. Strobl. Plenoptic cameras as depth sensors for the precise arm positioning of planetary exploration rovers. In *71. Deutschen Luft- und Raumfahrtkongress 2022 (DLRK 2022)*, September 2022.

[71] Gabriel Lippmann. Epreuves reversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1):821–825, 1908.

[72] Qingsong Liu, Xiaofang Xie, Xuanzhe Zhang, Yu Tian, Yan Wang, and Xiaojun Xu. Feature detection of focused plenoptic camera based on central projection stereo focal stack. *Applied Sciences*, 10(21):7632, 2020.

[73] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[74] Andrew Lumsdaine and Todor Georgiev. The focused plenoptic camera. In *2009 IEEE International Conference on Computational Photography (ICCP)*, pages 1–8. IEEE, 2009.

[75] Wenjie Luo, Alexander G Schwing, and Raquel Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5695–5703, 2016.

[76] Matthieu Maitre, Yoshihisa Shinagawa, and Minh N Do. Symmetric multi-view stereo reconstruction from planar camera arrays. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[77] Manuel Martínez-Corral and Bahram Javidi. Fundamentals of 3d imaging and displays: a tutorial on integral imaging, light-field, and plenoptic systems. *Advances in Optics and Photonics*, 10(3):512–566, 2018.

[78] Belen Masia. *Course in Computational Imaging: Lecture Notes.* Universidad de Zaragoza, 2022.

[79] Pierre Matysiak, Mairéad Grogan, Mikaël Le Pendu, Martin Alain, and Aljosa Smolic. A pipeline for lenslet light field quality enhancement. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 639–643. IEEE, 2018.

[80] Pierre Matysiak, Mairéead Grogan, Mikaël Le Pendu, Martin Alain, Emin Zerman, and Aljosa Smolic. High quality light field extraction and post-processing for raw plenoptic data. *IEEE Transactions on Image Processing*, 29:4188–4203, 2020.

[81] M. Miller. Understanding light field photography. *Make*, 2014.

[82] José María M Montiel. *Course in Computer Vision: Lecture Notes.* Universidad de Zaragoza, 2022.

[83] Antoine Mousnier, Elif Vural, and Christine Guillemot. Partial light field tomographic reconstruction from a fixed-camera focal stack. *arXiv preprint arXiv:1503.01903*, 2015.

[84] Vishvjit S Nalwa. *A guided tour of computer vision.* Addison-Wesley Longman Publishing Co., Inc., 1994.

[85] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. *Light field photography with a hand-held plenoptic camera.* PhD thesis, Stanford university, 2005.

[86] Michael Oechsle, Songyou Peng, and Andreas Geiger. Unisurf: Unifying neural implicit surfaces and radiance fields for multi-view reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5589–5599, 2021.

[87] Massachusetts Institute of Technology. Synthetic light field archive. `https://web.media.mit.edu/~gordonw/SyntheticLightFields/index.php`, 2013. [Last access 2023-07-23].

[88] Luca Palmieri, Reinhard Koch, and Ron Op Het Veld. The plenoptic 2.0 toolbox: Benchmarking of depth estimation methods for mla-based focused plenoptic cameras. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 649–653. IEEE, 2018.

[89] Ramviyas Parasuraman. Mobility enhancement for elderly. *arXiv preprint arXiv:1410.5600*, 2014.

[90] Amit Patel. Hexagonal grids, 2015. Accessed: 2023-08-13.

[91] P. Paudyal, R. Olsson, M. Sjöström, Federica Battisti, and Marco Carli. Smart: A light field image quality dataset. In *Proceedings of the 7th International Conference on Multimedia Systems, MMSys 2016*, pages 374–379, 2016. cited By 17.

[92] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 12179–12188, 2021.

[93] Raytrix. 3D light-field vision technology made in Germany. `https://raytrix.de/products/`. [Last access 2023-08-28].

[94] Raytrix. Rxlive. `https://raytrix.de/experience-the-new-rxlive-5-0-download-try-buy/`. [Last access 2023-12-28].

[95] Martin Rerabek and Touradj Ebrahimi. New light field image dataset. In *8th International Conference on Quality of Multimedia Experience (QoMEX)*, number CONF in I, 2016.

[96] Adi Robertson. VR camera maker Lytro is shutting down, and former employees are going to Google. `https://www.theverge.com/2018/3/27/17166038/lytro-light-field-camera-company-shuts-down-google-hiring`, 2018. [Last access 2023-12-28].

[97] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016.

[98] Johannes L Schönberger, Enliang Zheng, Jan-Michael Frahm, and Marc Pollefeys. Pixelwise view selection for unstructured multi-view stereo. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 501–518. Springer, 2016.

[99] Pranab Kumar Sen. Estimates of the regression coefficient based on kendall's tau. *Journal of the American Statistical Association*, 63(324):1379–1389, 1968.

[100] Sumit Shekhar, Shida 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 2018, Northumbria University, Newcastle, UK, September 3-6, 2018*, page 120, 2018.

[101] Likun Shi, Wei Zhou, Zhibo Chen, and Jinglin Zhang. No-reference light field image quality assessment based on spatial-angular measurement. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):4114–4128, 2019.

[102] 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*, pages 4748–4757, 2018.

[103] Sudipta N. Sinha. *Multiview Stereo*, page 516–522. Springer US, Boston, MA, 2014.

[104] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *ACM siggraph 2006 papers*, pages 835–846. 2006.

[105] Klaus H. Strobl and Gerd Hirzinger. Optimal Hand-Eye Calibration. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4647–4653, Beijing, China, October 2006.

[106] Klaus H. Strobl and Gerd Hirzinger. More Accurate Pinhole Camera Calibration with Imperfect Planar Target. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV), 1st IEEE Workshop on Challenges and Opportunities in Robot Perception*, pages 1068–1075, Barcelona, Spain, November 2011.

[107] Klaus H. Strobl and Martin Lingenauber. Stepwise calibration of focused plenoptic cameras. *Computer Vision and Image Understanding*, 145:140–147, 2016.

[108] Klaus H. Strobl, Wolfgang Sepp, Stefan Fuchs, Cristian Paredes, Michal Smíšek, and Klaus Arbter. DLR CalDe and DLR CalLab. Institute of Robotics and Mechatronics, German Aerospace Center (DLR). Oberpfaffenhofen, Germany. `http://www.robotic.dlr.de/callab/`, 2005. [Last access 2024-03-11].

[109] Richard Szeliski. A multi-view approach to motion and stereo. In *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, volume 1, pages 157–163. IEEE, 1999.

[110] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[111] Yu-Ju Tsai, Yu-Lun Liu, Ming Ouhyoung, and Yung-Yu Chuang. Attention-based view selection networks for light-field disparity estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12095–12103, 2020.

[112] Brown University. Computational photography lab: Light fields and camera arrays. `https://cs.brown.edu/courses/csci1290/labs/lab_lightfields/`, 2022. [Last access 2024-03-11].

[113] Allied Vision. Mako G-419. Small size, powerful performance. https://www.alliedvision.com/en/camera-selector/detail/mako/g-419/. [Last access 2023-08-26].

[114] Abrar Wafa, Mahsa T Pourazad, and Panos Nasiopoulos. A deep learning based spatial super-resolution approach for light field content. *IEEE Access*, 9:2080–2092, 2020.

[115] Lijun Wang, Jianming Zhang, Yifan Wang, Huchuan Lu, and Xiang Ruan. Cliffnet for monocular depth estimation with hierarchical embedding loss. In *European Conference on Computer Vision*, pages 316–331. Springer, 2020.

[116] 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*, pages 3487–3495, 2015.

[117] Ting-Chun Wang, Jun-Yan Zhu, Ebi Hiroaki, Manmohan Chandraker, Alexei A Efros, and Ravi Ramamoorthi. A 4d light-field dataset and cnn architectures for material recognition. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part III 14*, pages 121–138. Springer, 2016.

[118] Chao Wen. The stanford multi-camera array. https://github.com/walsvid/Awesome-MVS. [Last access 2023-12-27].

[119] Bennett Wilburn. *High-performance imaging using arrays of inexpensive cameras*. Stanford University, 2005.

[120] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018.

[121] Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024.

[122] Jure Zbontar, Yann LeCun, et al. Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.*, 17(1):2287–2318, 2016.

[123] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Recovering consistent video depth maps via bundle optimization. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[124] Yingliang Zhang, Peihong Yu, Wei Yang, Yuanxi Ma, and Jingyi Yu. Ray space features for plenoptic structure-from-motion. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4631–4639, 2017.

[125] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[126] Jiawei Zhao, Ke Yan, Yifan Zhao, Xiaowei Guo, Feiyue Huang, and Jia Li. Transformer-based dual relation graph for multi-label image recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 163–172, 2021.

[127] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[128] Shuyao Zhou, Tianqian Zhu, Kanle Shi, Yazi Li, Wen Zheng, and Junhai Yong. Review of light field technologies. *Visual Computing for Industry, Biomedicine, and Art*, 4(1):29, 2021.

List of Tables

List of Figures

List of Acronyms

BPR	*Bad Pixel Ratio*
Cissy	*Continuous Integration Software System*
CNN	*Convolutional Neural Network*
CPSF	*Central Projection Stereo Focal Stack*
DLR	*Deutsches Zentrum für Luft- und Raumfahrt (in English, German Aerospace Center)*
DLR CalDe	*DLR Calibration Detection Toolbox*
DLR CalLab	*DLR Calibration Laboratory*
DLT	*Direct Linear Transformation*
EPI	*Epipolar Plane Image*
ESA	*European Space Agency*
LiDAR	*Light Detection and Ranging*
MARE	*Mean Absolute Relative Error*
MLA	*Microlens Array*
MLP	*Multilayer Perceptron*
MSE	*Mean Squared Error*
MSRE	*Mean Squared Relative Error*
OLS	*Ordinary Least Squares*
PoE	*Power over Ethernet*
P-CalLab	*Plenoptic Camera Calibration Matlab Toolbox*
P-SfM	*Structure from Motion in Plenoptic Camera*
RANSAC	*Random Sample Consensus*
ReLU	*Rectified Linear Unit*
RMSE	*Root Mean Squared Error*
RMSLE	*Root Mean Squared Logarithmic Error*
SAI	*Sub-aperture Image*
SfM	*Structure from Motion*
SGD	*Stochastic Gradient Descent*
SGM	*Semi Global Matching*
SLAM	*Simultaneous Localization and Mapping*

SSD	*Sum of Squared Differences*
SVD	*Singular Value Decomposition*
TTR	*Test-Time Refinement*
ViT	*Vision Transformer*