University of Padua

Department of Information Engineering

Master Thesis in ICT for Internet and Multimedia

Recovering Occlusion Aware Depth and Image using Rotating Point Spread Function

Master Candidate:
Elia Tezze

Supervisors:
Muhammad Siddiqui
Sony Europe B.V.

Pietro Zanuttigh
University of Padua

July 2023
Academic Year 2022/2023

Abstract

Depth estimation is a key challenge in Computer Vision. In particular, when we consider the limitations of traditional techniques. In the monocular depth estimation field, the presence of occlusion boundaries is one of the most critical issues, we can find them as depth discontinuity along the edges of the objects, where the object in the foreground occludes the objects in the background. This results in incomplete or inaccurate depth prediction, making it difficult to extract accurarte geometry information from the scene.

To this end, recent studies have shown how coded aperture-based methods using phase and/or amplitude masks can encode strong depth cues within 2D images using depth-dependent point spread functions (PSFs). In this thesis, we propose a new approach to address the problem of occlusion boundaries with the aim of improving the result for depth estimation. In our case, the depth dependency is achieved using a phase mask that is jointly optimized with the weights of a convolutional neural network in an end-to-end manner. A fully-working camera model is used to simulate the imaging system that can reliably estimate the depth map starting from a single RGB image. Compared to the most common methods used to solve occlusion boundaries in monocular depth estimation problems, in our final pipeline we propose a preconditioning step that aims at reducing the total effort required from the neural network, reducing the total time required to train the network, and achieving better results in terms of accuracy over the final estimate. In this preconditioning step is already computed a raw estimate of the depth, using a well-known deblurring strategy that reconstruct the details of the image in the region that correspond to a specific level of depth. In this way a layered image is already processed and the final neural network performed only an association operation between the various layers.

Moreover, to address the problem of image quality degradation due to the PSF-Blurring effects, our network can recover the all-in-sharp image along with the depth estimate, starting from the output of the preconditioning step.

Abstract

La stima della profondità rappresenta una sfida in qualsiasi ambito relativo alle applicazioni di Computer Vision. In particolare, se si considerano le limitazioni delle tecniche tradizionali in fotografia. Nel campo della stima della profondità monoculare, la presenza degli occlusion boundaries è uno dei problemi più critici; essi possono essere individuati come discontinuità di profondità lungo i bordi degli oggetti, dove l'oggetto in primo piano copre gli oggetti sullo sfondo. Ciò comporta una stima della profondità incompleta o inaccurata, rendendo difficile estrarre informazioni significative dalle scene in esame. A tal fine, recenti studi hanno dimostrato come i metodi basati sulla coded aperture, utilizzando maschere di fase e/o di ampiezza, possano codificare segnali di profondità all'interno di immagini 2D utilizzando le Point Spread Functions (PSF) dipendenti dalla profondità. In questa tesi, proponiamo un nuovo approccio per affrontare il problema degli occlusion boundaries con l'obiettivo di migliorare il risultato finale per la stima della profondità. Nel nostro caso, la dipendenza dalla profondità è introdotta da una maschera di fase che viene ottimizzata assieme ai pesi di una rete neurale con approccio end-to-end. In fase di sviluppo viene utilizzato un modello di camera che permette di simulare l'intero sistema presente in una fotocamera reale che può stimare affidabilmente la mappa di profondità a partire da una singola immagine RGB. Rispetto ai metodi più comuni utilizzati per risolvere i problemi degli occlusion boundaries nell'ambito della monocular depth estimation, la nostra pipeline finale utilizza uno step di pre-condizionamento, inserito con lo scopo di ridurre lo sforzo totale richiesto dalla rete, riducendo così il tempo totale necessario per addestrare la rete e ottenendo risultati migliori in termini di accuratezza sulla stima finale. In questo step di precondizionamento viene calcolata una stima rozza della profondità, modificando un noto algoritmo di riduzione del rumore che ricostruisce i dettagli dell'imagine nelle regioni che corrispondo a specifici livelli di profondità. In questo modo otteniamo una imagine stratificata e la rete neurale finale dovrà eseguire solo una associazione tra i vari livelli.

Inoltre, per affrontare il problema della degradazione nella qualità dell'immagine dovuta agli effetti di sfocatura PSF, la nostra rete può recuperare l'immagine completamente a fuoco insieme alla stima di profondità, a partire dal risultato dello step di pre-condizionamento.

Contents

Abstract iii

List of Figures xii

List of Tables xiii

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Image Formation and Point Spread Function (PSF) . . . 3
  - 1.2.1 Optical Model . . . 3
  - 1.2.2 PSF and Rotating-PSF (RPSF) . . . 8
- 1.3 Pupil Phase Engineered Camera . . . 10
  - 1.3.1 Gaussian-Laguerre Modes . . . 11
- 1.4 Occlusion Boundaries Problem . . . 14
- 1.5 Contribution and Thesis outline . . . 16

2 Related Work 19
- 2.1 Depth Estimation . . . 19
  - 2.1.1 Classical Methods . . . 19
  - 2.1.2 Coded Aperture methods . . . 21
  - 2.1.3 End-to-end Learning . . . 22
- 2.2 Image Deblurring . . . 22
- 2.3 Occlusion Aware Model . . . 24

3 Depth Estimation using RPSF 27
- 3.1 Phase Mask Design . . . 27
- 3.2 Initial Architetture . . . 30
  - 3.2.1 Linear Image Formation Model . . . 30
  - 3.2.2 Deep Learning Architecture . . . 32
  - 3.2.3 Image Deblurring Module . . . 34
- 3.3 RPSF Prototype . . . 36

4 Occlusion Handling **39**
4.1 Pre-Processing Approach . . . 39
4.1.1 Non-Linear Formation Model . . . 39
4.1.2 Approximate Inverse Step . . . 41
4.2 Post-Processing Approach . . . 44
4.2.1 Displacement Field . . . 44
4.3 Proposed Approach . . . 45

5 Results and Discussions **49**
5.1 FlyingThings3D Dataset . . . 49
5.2 NYUV2 Dataset . . . 50
5.3 Results . . . 50
5.3.1 Phase Mask Design . . . 51
5.3.2 Displacement Field . . . 54
5.3.3 Non-Linear Image Formation Model . . . 57
5.3.4 Approximate Inverse Step . . . 61

6 Conclusions and Future Works **67**
6.1 Conclusions . . . 67
6.2 Future Work . . . 67

Bibliography **69**

List of Figures

1.1 A thin lens model where is shown the rays path from the object to image plane. . . . 4
1.2 Example of Chromatical aberration. . . . 5
1.3 Example of Color Filter Array with Bayern pattern. . . . 6
1.4 Example of sensor chip with possible noise injection. . . . 7
1.5 Normalized FI of the rotating and standar PSFs as they go through defocus. Image take from [1]. . . . 10
1.6 Comparing PSF profile for different PPE camera approach: from left to right we increase the value of defocus. (a) Clear aperture PSF. (b) Single-Helix phase mask from [2] with L= 7 fresnel zones partitioning. (c) Double-Helix phase mask composed by superposition of (1,1)(3,5)(5,9)(7,13)(9,17) GL modes [3]. Image is taken from Kumar et al. [4]. . . . 12
1.7 Examples of GL modes: (a) intensity, (b) phase. Image taken from Piestun et al. [5]. . . . 13
1.8 Result for the Gauss-Laguerre modal plane. Each mode that composes the wave is represented by a point within grid. Image take from [1]. . . . 13
1.9 In this figure we show an example of augmentation for an RGB image from NYUv2 with a virtual Stanford rabbit using different depth maps for occlusion aware integration. This image is taken from [6] and [7]. . . . 14
1.10 Samples of NYUv2-OC++ dataset taken from [8]. The selected highlighted regions in red rectangles emphasize the high-quality and fine-grained annotations. . . . 16

3.1 Schematic diagram of a specific Fresnel zone with its spiral phase retardation, taken from [9]. . . . 27
3.2 Generated N-order helix PSF with different number of rotating lobes $[L = 5, \epsilon = 0.9]$. . . . 29
3.3 Generated RPSFs with different values of ϵ with $[N = 2, L = 5]$. . . . 29

3.4 Differentiable phase mask approximation with 2D tanh functions. . . 30
3.5 Representation that shows how a single layer for the coded blurred image is computed, starting from the all-in-focus image and applying first a convolution with the $RPSF_d$ and then multiplying by the corresponding mask. 31
3.6 Stage 2 of the pipeline proposed by Mel in [10]. It used to simulate a realistic image formation model. 32
3.7 Representation of the standard U-Net used in [10] and in [11]. 33
3.8 Graphical representation of Deep Wiener deconvolution network. First extracts useful feature information from the blurry input image and then conducts an explicit Wiener deconvolution in the (deep) feature space. A multi-scale cascade encoder-decoder network progressively restores clear images, with fewer artifacts and finer detail. Image taken from [12]. 34
3.9 Modal plane that represents the possible combination for GL modes that produce a rotating beam. Green (SH), Black (DH) Image taken from [13]. 36
3.10 Examples of SH and DH phase masks, on top there are the phase representation of GL modes that follow the combination described in Fig.3.9 can produce the phase mask report in the middle row. On the bottom is reported the corresponding RPSF where we can clearly distinguish the different configurations of the mask and also the depth dependency. 37

4.1 Comparing image formation models that simulate defocus blur from an RGB image (top left) and a depth map (top center left) with an accurate simulation (top center right). We compare a simple linear image formation model (bottom left), the variants of the linear model proposed by Wu et al. [11](bottom center left) and Chang et al. [36] (bottom center right), and non-linear image formation model (bottom right). 40
4.2 Results for modified DWDN network proposed in [12], in the first row from left to right is display the Ground Truth (GT) depth, the sharp all-in-focus image and the coded image obtained as output of the camera model. In the four columns below there are samples for the 3D layer representation with their corresponding GT layer. . . . 43

4.3 The full architecture for the proposed solution. As in the work of Mel [10] in the first stage the height map of the phase mask is jointly optimized with the weights of a U-Net trained on noise-free synthetic images for the monocular depth estimation task. In the second stage the phase mask is incorporated within the optical layer of the camera model to reproduce a realistic image formation model and to obtain the coded image. (In this stage the nonlinear image formation model is applied to take into account the depth cues for the occlusion boundaries). In the following stage a modified version of DWDN [12] is used to perform a preconditioning step that creates a 3D layered image in which there are sharp details in the corresponding ground truth depth layer. At final stage a CNNs is trained to get the final all-in-focus image along with the predicted depth map. 46

5.1 FlyingThings3D samples of resolution 278 x 278. 50
5.2 The generated RPSFs for the experiments and its height maps. Notice that the GL modes phase mask is not differentiable so it must be considered as a fix phase mask. 52
5.3 Sample of depth map prediction for the four ablation studies with the FlyingThings3D. 53
5.4 Initial pipeline proposed by Mel in [10] within the Displacement Field module (red rectangle). 54
5.5 Qualitative simulation results for Displacement Field module on FlyingThings3D dataset, the red circles are used to enphasize the key point to observe. 55
5.6 Learned RPSF as function of defocus for NYUv2 depth dataset. . . . 56
5.7 Qualitative simulation results for Displacement Field module on NYUv2 Depth dataset. The red circles are used to enphasize the key point to observe. 57
5.8 Comparison between the Linear Image Formation Model (b) and the non-linear(c). (a) is the sharp all-in-focus image that provide the scene under examination, and (d) is the plot of the differences. 58
5.9 Qualitative comparison results on image formation model from the test set of FlyingThings3D. (a) Sharp all-in-focus image. (b) Coded image with Linear image formation model. (c) Coded image with Non-Linear image formation mode. (d) Estimate depth map from b, (e) Estimate depth map from c. (f) Ground truth depth map from the dataset. 59

5.10 Qualitative comparison results on image formation model from the NYUv2 Depth dataset. (a) Sharp all-in-focus image. (b) Ground truth depth map, in dark blue the invalid pixels. (c) Predicted depth map with Linear approach. (d) Results with non-linear image formation model. 60
5.11 Third stage's pipeline for the starting approach (a) and to the final proposed pipeline (b). 61
5.12 Example of "perfect input". Notice how moving through the image, that represent the depth layer, different details are in focus, while the rest of the images remains blurry. 62
5.13 Results for one sample of the pre-conditioning step. 63
5.14 Qualitative comparison for the FlyingThings3D dataset. (a) Ground Truth depth image. (b) Noisy estimate disparity from Mel [10]. (c) Proposed Non-Linear IFM. (d) Proposed Non-Linear IFM with Approximate inverse step. (e) Recover sharp all-in-focus image. 64

List of Tables

5.1 Quantitative results for the four conducted experiments on the subset of FlyingThings3D. FR means Fresnel Zone instead GL is the Gauss Laguerre mode. 51
5.2 Quantitative comparison over different approach that perform monocular depth estimation, +DF means that Displacement Field module is present (↓: Lower is better; ↑: Higher is better). First rows of the table is taken from Ikoma in [14]. 55
5.3 Quantitative comparison with all competing methods for monocular depth estimation on NYUV2 Eigen test set [15]. 61
5.4 Quantitative result for the ablation study with the perfect input. . . . 62
5.5 Quantitative comparison over different approaches that perform monocular depth estimation, (↓: Lower is better; ↑: Higher is better). Firsts rows of the table is taken from Ikoma in [14]. 63

1 Introduction

1.1 Motivation

Depth estimation is one of the most interesting and important challenges in the field of artificial vision. It involves determining the distance between the camera and objects in the scene. There are several methods to estimate depth and it depends on the target applications and the set of physical constraints. One example of an application could be autonomous driving or obstacle detection where we exploit stereo vision settings, it consists of two or more cameras set in the same environment and they are used to capture the same scene from different angles. The disparity information that we decode from the image, together with the information coming from the set-up, can be used to calculate depth.

Another solution is to use reliable active ranging sensors such as Time of Flight (ToF) cameras or LiDAR sensors, the first measure the time it takes light to travel to an object and back to the sensor to calculate the distance, the second is based on laser technology and it uses the laser to map the object around the sensor by observing the reflection. Both sensors are useful for fast real-time depth data acquisition and find applications in motion object detection and autonomous vehicle navigation.

In recent years, the interest in the depth estimation task has intensified, thanks to the important role it is playing in the latest studies related to 3D reconstruction. NeRF and NeuS are just two examples of those technologies that allow to create a rendering of objects or scenes, starting from a series of images. This data can be obtained by scanning real-world scenes or by creating 3D models on the computer. The 3D rendering process requires knowledge of the depth of the objects present in the scene. Nowadays, this information is obtained through stereoscopy, or the use of multiple cameras to obtain 3D data information. However, in recent years, monocular depth estimation has proved to be an equally valuable and cost-effective solution for estimating depth from a single image.

Monocular depth estimation has several advantages over other depth estimation methods. First, it does not require the use of multiple cameras or sensors but can be performed with a single image. This makes it cheaper and easier to deploy in a wide

range of applications while meeting more stringent power consumption and space requirements. In return, some drawbacks to using monocular depth estimation may be related to the accuracy of the final estimate and may be affected by shadows, bright lights, blurred backgrounds, or partial occlusion. One of the problems that affect all the methods adopted to perform the depth estimation is the well-known occlusion problem. In particular, the thesis focus on finding a method that solve the occlusion boundaries problems in a monocular case. An Occlusion Boundary occurs along the edges of the object when a foreground object covers a background object in the scene. This results in an inaccurate or incomplete depth estimate and the error could spread to the final application where, for example, in an Augmented Reality work the insertion of an object does not result good.

To address this problem, in the thesis a computational camera will be explored wherein a phase mask is inserted in the pupil plane of the camera to encode depth information in the 2D captured images. The phase mask encodes the depth information by altering the Point Spread Functions (PSF) in a way that it rotates with a depth dependence. The main advantage of this approach is that we can encode meaningful depth cues in single RGB images making it easier for post-processing algorithms to produce accurate depth maps. Starting from the capture image coded by the PSF, in the thesis, we will explore two different approaches to get the depth map and recover the all-in-focus image. The first is a post-processing method, called Displacement Field, where at the end of the proposed pipeline we attach a Convolutional Neural Network (CNN) that acts as "edge enhancer". In this way, the network can learn where the possible problems of the estimated image are and through a resampling of the value of the pixels in these areas can return sharper edges.

The second and also the most delved method has two main differences concerning the previous proposed works, here a Non-Linear Image formation model is inserted into the camera model. With respect to the most common Linear one this new approach is able to model in a better way the defocus blur introduced by the PSF at depth discontinuities. Moreover, we will include an effective preconditioning approach to our pipeline, this approximate inverse makes it significantly easier for the Monocular Depth Estimation (MDE) framework to infer the prediction of the depth map from the coded image. To conclude, about the problem of image quality deterioration, a deep learning module base on Wiener Deconvolution is used to solve it.

1.2 Image Formation and Point Spread Function (PSF)

Image formation is a fundamental process in the capture and representation of the visible world. Understanding how images are created by cameras is essential for a wide range of applications, including digital photography, computer vision studies, and depth estimation task. In the context of a common RGB camera, the main objective is to capture images that accurately represent the real scene in terms of color, brightness, and details. The image formed by the RGB camera is the result of a series of steps, including optical capture, conversion of light energy into electrical signals, and digital imaging. It is important to note that the image formed by the RGB camera is an approximate representation of the actual scene. There are several challenges and complexities involved in the image formation process, such as lens distortion, noise compensation, variable lighting, and other environmental factors. All these inconvenients introduced by the optical system are also known as Optical Aberrations.

In order to proceed, it is crucial to have a clear understanding of the image formation process and how it works, as our project revolves around simulating an optical model and subsequently adding or modifying components on that. In this section, an introduction for the image formation model based on simple and linear optical principals is given. We provide also an overview over the different camera components as the configuration of lens, the image sensor and the noise source in the camera.

1.2.1 Optical Model

To have a complete and detailed description of the optical model used you can refer to Figure 1.1 where a classic "thin lens model" is shown. That is a simplification of the more complex optical model used that is usually combined with the paraxial approximation (a small angle approximation used in Gaussian optics), to produce a fully working simulation for the camera model. The thin lens model is udes to describe the behavior of light rays that pass through the lens. Basically, a ray that passes through the lens can follow three simple rules. If a ray enters parallel to the axis on one side of the lens proceeds towards the focal point F on the other side, if it arrives at the lens after passing through the focal point on the front side, comes out parallel to the axis on the other side or, if it passes through the center of lens, it will not change its directions.

$$\frac{1}{Z} + \frac{1}{Z'} = \frac{1}{f_{tl}} \tag{1.1}$$

The thin lens equation (1.1) gives the relationship between the object distance Z , the image distance Z' and how they are relate to the focal length F_{tl} . A point of the scene at distance Z is focused on a specific distance depending on Z and F_{tl}, if the resulting distance is different from Z' the point result in a circle of confusion. The image quality level is also relate to the circle of confusion. In fact, if the circle is smaller than a pixel size the quality is acceptable, otherwise the image results blurry. Another important aspect in a thin lens model is the control of Depth of Field (DOF), that refers to the range of distances in which the image result in focus. To this scope, the aperture size plays a significant role in determining the DOF. A larger aperture results in a shallower DOF, meaning that only a narrow range of distances around the point of focus will be in focus. On the other hand, a smaller aperture increases the DOF, leading to a greater depth of focus. However, also the focal length affects the DOF. In fact, a lens with a shorter focal length will have a larger DOF compared to a lens with a longer focal length when shooting at the same aperture and distance.

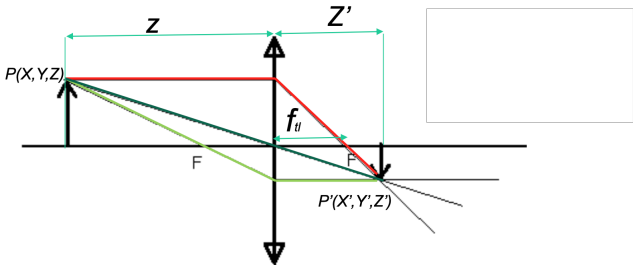

Figure 1.1: A thin lens model where is shown the rays path from the object to image plane.

An important prerequisite for the thin lens model is the property of not considering aberrations. An aberration is a lens property that can affect the way of light reaches the image plane. Usually, a real camera has much more complex lens systems that are used as a correction for different types of aberrations. The most common aberrations that we can encounter when we use a lens are the Chromatic Aberration and the Radial Distortion. The Chromatic Aberration (Figure 1.2) is caused by the dispersion of white light into different colors component. Light dispersion occurs because the refractive index of optical materials varies slightly depending on the wavelength of the light. At the end, the resulting image can show colored halos or blurred edges around objects, especially at the edges of the image where the effect is higher. These halos may appear as purple or green stripes around bright objects on a dark background. Instead, Radial Distortion is a type of optical aberration that causes a deformation of straight lines in the image, where lines that should be straight appear curves radially relative to the center of the image. This aberration occurs mainly in wide-angle lenses and lenses with a wide field of view. It can be

affected by factors such as the shape of lenses, the alignment of the lenses, and the arrangement of the optical elements inside the lens. Both those aberrations and many others can be resolved using corrective optical elements or post-processing correction software. In our work we exploit the concept of in-focus and defocus objects, this could be seen as a defect of the lens. This kind of aberration is linked to the circle of confusion that we have described before.

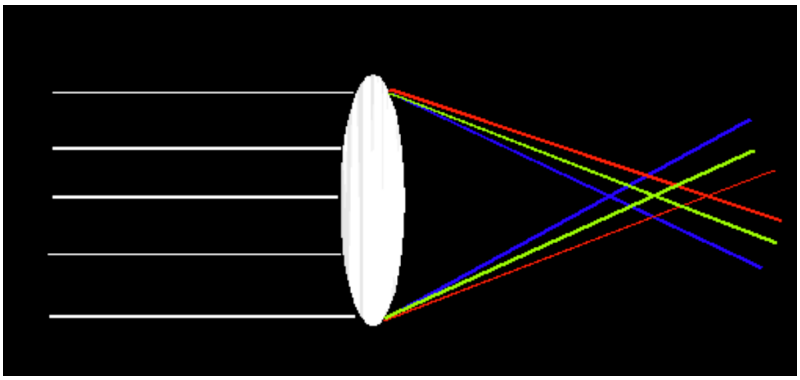

Figure 1.2: Example of Chromatical aberration.

In order to give a total overview on the tools used during the work we still have to see some features about the digital camera. To perform a proper simulation of a realistic camera we have to go deeper over how handle noise in the camera model.

A digital camera is usually composed of three parts, the optics is the first part that we encounter, and we have talked a lot about it in the previous section. Another part is the sensor chip, where we usually found the color sensor and the ADC (Analog to Digital Conversion). The color sensor is made with a CCD (Charge Coupled Device) or with a CMOS (Complementary Metal Oxide Semiconductor). A CCD sensor consists of a photodiode matrix, in which each photodiode represents one pixel. When light hits the photodiodes, it is converted into electric charge proportional to the light intensity. These charges are then transferred to a charge accumulation structure called the accumulation register, which allows you to read the charge values of each pixel sequentially. CCD sensors tend to have higher light sensitivity and better image quality but may require more power and generate some thermal noise. A CMOS sensor, on the other hand, uses an integrated photodiode array with transistors for each pixel. Here, each pixel is capable of converting light into an electric charge, which is amplified by the transistor and then converted into a digital signal. CMOS sensors are generally more compact, cheaper and require less power than CCD sensors. However, they tend to have lower light sensitivity and slightly lower image quality than the other. Those two sensors are able to measure only the intensity of light, but not its wavelength, which determines the color. To this purpose a Color Filter Array (CFA) allows the separation of light

into several primary color components and capture color information via a single sensor. Figure 1.3 shows a well know example of CFA with the Bayer pattern [16] with Red, Green and Blue color combination, where the green filter was doubled to take into consideration the peak sensitivity of the human visual system. The sensor chip consists of many other components such as micro lenses and an amplifier circuit that allows to control the sensitivity of the sensor to light, i.e. to control the ISO level of the camera. The last component is made by the ADC which performs a quantization of the raw analog input signal with a resolution of 8 or 16 bits.

After the sensor chip, we have the last part to complete the process of image formation composed by the Image Signal Processor (ISP). This processor is responsible for processing the raw signal from the image sensor and turning it into a final image that is optimized and ready for subsequent processing or even for visualization. The principal task for the ISP is to perform the demosaicing, so takes the output of the CFA and interpolates its result to create three fully populated color planes. Can perform also noise reduction by filtering the data to reduce the noise level introduce from the image sensor. At the end it adjusts contrast, brightness, sharpness, and balance the level of white. The ISP can also perform the compression to get the final image in the correspondent format.

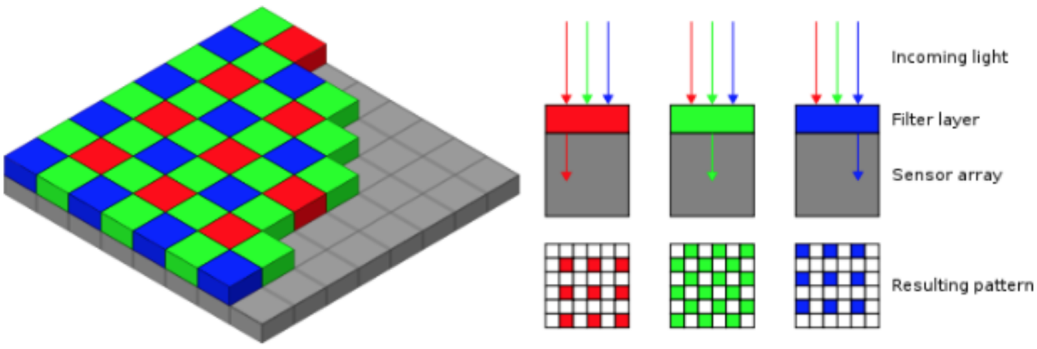

Figure 1.3: Example of Color Filter Array with Bayern pattern.

Since we want to simulate a realistic model for a camera, we have to model the noise. There exist several different types of noise in a camera and can be caused by various factors. The most known are thermal noise, reading noise, noise in electric signals, and quantization noise. These results in modifications, artifacts or random variations in the image. In the remaining part we present and discuss only the relevant noise sources that are important in our work and we leave the reader to refer to Healey et al. [17] and Tsin et al. [18] works that provides a full overview about the existing kind of noise in a common camera model.

Figure 1.4 is used to show where are the possible noise injection in our simulation. We can split it into two groups, the sensor noise that is still sub-split into read and shot noise and the quantization noise. The read noise (1.2) can be modeled as an

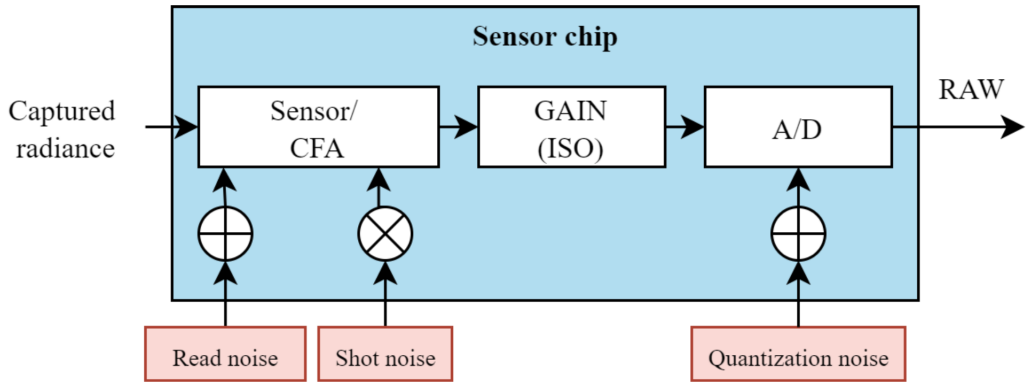

Figure 1.4: Example of sensor chip with possible noise injection.

additive Gaussian noise with zero mean and a fixed standard variation. It could appear during the reading process due to manufacturing imprecision, this noise can be most noticeable as small fluctuations in areas of the image with low brightness or when using high ISO settings.

$$Out(x, y) = In(x, y) + N(\mu, \sigma^2) \tag{1.2}$$

where In e Out are the two 2D function that represent the image and $N(\mu, \sigma^2)$ is a random process with mean μ and standard deviation σ.

Shot noise also known as Poisson noise (1.3), is a type of intrinsic noise due to the nature of electric light. It is caused by the photons that make up light. When light hits the camera sensor, the number of photons reaching each pixel is subject to random fluctuations. It can be approximated using a Poisson probability distribution, which describes the uncertainty in photon counting.

$$P(n = K) = \frac{e^{-\lambda t}(\lambda t)^2}{k!} \tag{1.3}$$

where λ is the expected number of photons per unit time, N is the number of measured photons over a time interval t.

As the last, Quantization noise is a type of noise that occurs when we convert an analog signal, such as a continuous intensity, into a digital signal represented by a discrete set of values or also when we perform a final compression. In those cases, the intensity level of each pixel is converted into a numeric value. However, since the total amount of available values is fixed by the number of bits or shall be reduced, an inevitable approximation occurs.

1.2.2 PSF and Rotating-PSF (RPSF)

Although the thin lens model is an abstraction of the optical behavior of a lens, it may not be enough accurate to describe in details the formation of image in certain complex situations. In these cases, we need to consider a more sophisticated model with emphasis on wave optics and field propagation in frequency domain. To this end we give an introduction over the Point Spread Function and also over a modified version of it, the Rotating-PSF, that will be useful in the following section to resolve the task of the monocular depth estimation. The PSF represents the response of an optical system to a point light source. Its knowledge is essential in many applications, such as image processing, deconvolution, optical correction for aberration, and as system performance evaluation. With the PSF, we can simulate or correct unwanted optical effects and improve the quality of the resulting image. One possible way to analyze the response of an optical model may be to analyze its Optical Transfer Function (OTF). The OTF shows how the optical system attenuates or amplifies the different spatial frequencies in the image and provides information about the resolution, sharpness and accuracy of them. In our work we deal with the PSF with the defocus dependencies. To this purpose the ambiguity function (AF) has proved to be a useful tool for characterizing an incoherent optical system's performance in the presence of defocus. Brenner et al. [19] have shown that the AF of a pupil function simultaneously displays the OTF for all values of defocus. In the equation (1.4) and (1.5) there are defition for the ambiugity function $A(u_1, u_2, y_1, y_2)$ with an arbitrary two-dimensional pupil where the variables y_1 and y_2 have the units of space, and a formula that show its relation with the OTF given defocus W_{20} and wavelenght λ.

$$A(u_1, u_2, y_1, y_2) = \iint_{-\infty}^{+\infty} P(v_1 + \frac{u_1}{2}, v_2 + \frac{u_2}{2}) \times P^*(v_1 - \frac{u_1}{2}, v_2 - \frac{u_2}{2}) \times exp[j2\pi(v_1 y_1 + v_2 y_2)]dv_1 dv_2 \quad (1.4)$$

$$H(u_1, u_2) = A\left(u_1, u_2, \frac{2W_{20}}{\lambda}u_1, \frac{2W_{20}}{\lambda}u_2\right) \quad (1.5)$$

where the ambiguty function is computed for the pupil function $P(u_1, u_2)$ calculated over the normalized spatial frequences $u_i = 2x_i/D$ and W_{20} is the defocus level.

The way to compute the PSF starting from the description of an optical system is proposed from Dowsky et al. [20], where he relates the Optical Transfer Function (OTF) to the Fourier transform of the PSF.

$$H(u_1, u_2) = \iint_{-\infty}^{\infty} h(x_1, x_2) \times exp[-j2\pi(x_1 v_1 + x_2 v_2)] dx_1 dx_2 \tag{1.6}$$

$$= \mathcal{F}[h(x_1, x_2)] \tag{1.7}$$

where $H(u_1, u_2)$ represent the OTF for the spatial frequency variables u_1, u_2. $\mathcal{F}$ is the Fourier Transform and $h(x_1, x_2)$ is the point spread function.

We leave the reader to Dowsky et al work [20] to have further specification on the derivation for the final formulas and also to have some examples over different pupil function results. It is important to note that for our work the pupil function is defocused by a sum of effects. Starting from the general formula (1.8) for the pupil function $P(x_1, x_2)$ where $C(x_1, x_2)$ is the plain clear aperture pupil function and $\phi(x_1, x_2)$ is the phase shifting, we can split the phase shift into two parts. One that refer to the defocus ϕ^{DF} introduced by the optical system and the other ϕ^M that relates to the phase mask

$$P(u_1, u_2) = C(u_1, u_2) exp(j\phi(u_1, u_2)) \tag{1.8}$$

$$\phi(u_1, u_2) = \phi^{DF} + \phi^M \tag{1.9}$$

For the specific task of monocular depth estimation, the PSF of such systems has not been optimized. In general, optical images transmit three-dimensional information (3D) through the level of blur present in each region of the image, farther object from the in-focus plane appears with more blur. This principle is used in techniques known as "depth from defocus"(DFD). Piestun et al. [1] present for the first time an engineered version of the PSF that exploits the spatial rotation and it was named the Rotating Point Spread Function (RPSF). Piestun is able to show how the RPSF provides a faster rate of change with depth than the normal PSFs computed from a clear pupil. This because, the more dissimilar the PSF is at different levels of defocus, the easier is to distinguish between different depth planes. To quantitatively compare the standard PSF with the RPSF (Figure 1.5) they evaluate the Fisher Information (FI) calculated with respect to defocus. Higher level of Fisher Information implies a potential increase of the accuracy over the estimate of the defocus parameter, and thus the depth level of an object.

Another problem that we have with the standard PSF, is the fact that with increasing defocus the PSF broadens rapidly, instead, the RPSF maintains its shape and size. This results in a lower constraint for the signal-to-noise ratio when we pass through the deblurring process. In the next section, we will see which are the possible different strategies to get an RPSF by exploiting the pupil phase engineered camera.

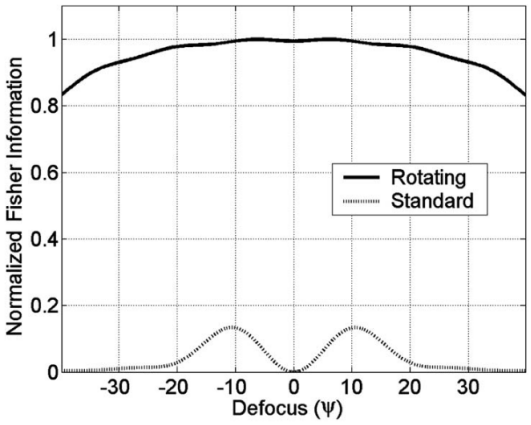

Figure 1.5: Normalized FI of the rotating and standar PSFs as they go through defocus. Image take from [1].

1.3 Pupil Phase Engineered Camera

In section 1.2.2 we had seen which are the main steps to get the PSF starting from a description of an optical system, through the math we had shown how this is possible if we know the pupil function of the system. But we haven't seen where this pupil function comes out and how we can encode that function to solve our task. To this end, a particular focus will be given to the Pupil Phase Engineered (PPE) camera model which adds to a normal camera model an optical element, named as phase mask, in the entrance pupil. Such mask introduces a phase delay on wavefronts and is able to engineer the final PSF by imposing a specific diffraction pattern. Under the assumption of incoherent scene illumination, we can consider a thin lens model with thick profile h located in the pupil plane. Define h with respect to its spatial coordinate x and y as:

$$h(x, y) = h_0 - \frac{x^2 + y^2}{2f(n-1)} \tag{1.10}$$

where h_0 is the thickest section, $n = 1.5$ is the usual refractive index in the case we use a convex lens made of glass and f the focal length of the optical system. When a light wavefront passes throught a lens it's delayed by an amount of phase that is proportional to the thickness of the lens. Thus, once we know $h(x, y)$ we can compute the phase shift ϕ^M, that we use in (1.9) as:

$$\phi^M(x, y) = \frac{2\pi(n-1)}{\lambda} h(x, y) \tag{1.11}$$

Note that the ϕ^M can suffer from the Chromatic aberration due to the wavelength dependency of the equation.

In the case of Out of Focus (OOF) imaging, also the optical system can introduce

a phase shift ϕ^{DF}, that is usually expressed as the quadratic phase term of the defocus aberration in the captured image.

$$\phi^{DF}(x,y) = \frac{\pi}{\lambda}(x^2+y^2)(\frac{1}{z} - \frac{1}{z_i}) \tag{1.12}$$

The final formula for the pupil function becomes,

$$P_{\lambda,W_m}(x,y) = C(x,y)e^{j\left(K_\lambda \Delta_n h(x,y) + K_\lambda W_m r(x,y)^2\right)} \tag{1.13}$$

where P_{λ,W_m} used to highlight the dependency from those variables, $C(x,y)$ is the clear aperture and the other unknown terms are computed as follow:

$$K_\lambda = \frac{2\pi}{\lambda} \qquad \Delta_n = (n-1) \qquad r(x,y) = \left(\frac{\sqrt{x^2+y^2}}{R}\right) \qquad W_m = \frac{R^2}{2}\left(\frac{1}{z} - \frac{1}{z_i}\right)$$

We have seen how by inserting an optical element in the entrance of the pupil we can encode the PSF. Usually, when we talk about the PPE, the optical element is always referred to as a phase mask, but exists some approaches where amplitude mask is used to modulate the amplitude component of the incident wave. From the formula the height map $h(x,y)$ can be manipulated to generate specific patterns for the PSF. Pavani [3] presents how the Gauss-Laguerre (GL) modes can be exploited for this task, and also shows how to obtain a high-efficiency RPSF for the monocular depth estimation. Prasad et al. [9] and [2] proposed a different approach for generating RPSF that uses Fresnel zones in the entrance pupil of the system. In Figure 1.6 we show a comparison of the resulting PSF. Notice how the standard PSF spreads rapidly with increasing defocus level. Meanwhile, the Single Helix proposed by Prasad and the Double Helix from Pavani preserve their shape and size while rotating with defocus.

1.3.1 Gaussian-Laguerre Modes

In Chapter 1.3 we present an approach that exploit the Gauss-Laguerre modes to creates an incoherent PSF that rotates at a uniform rate with changing defocus while maintaining its shape and form approximately.

In this paragraph we go in deeper to this argument by analyzing all the peculiarity and the possible application that they have. If we consider the function (1.14) that represent the propagations of scalar waves present by Piestun et al. [5].

$$\mathcal{U}(r,t) = u(\boldsymbol{r})exp[j(kz - wt)] \tag{1.14}$$

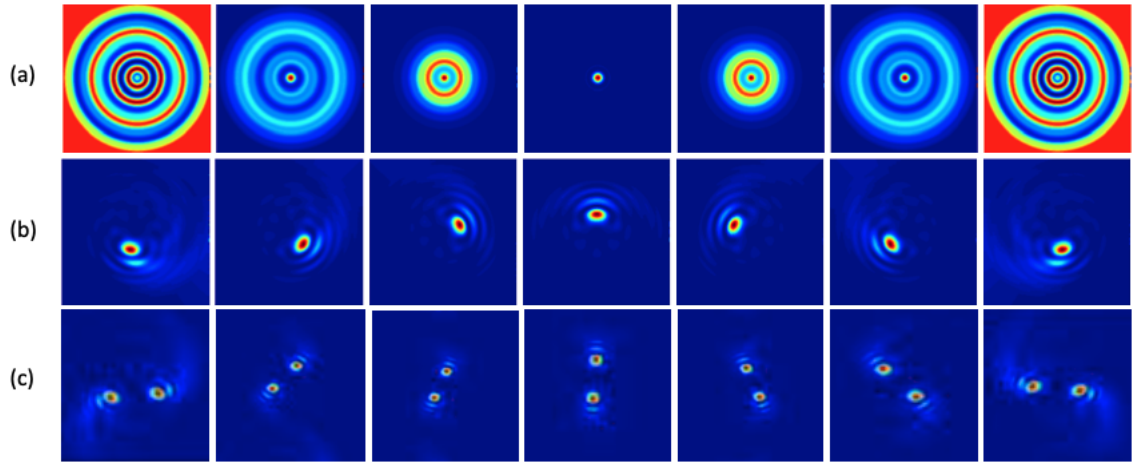

Figure 1.6: Comparing PSF profile for different PPE camera approach: from left to right we increase the value of defocus. (a) Clear aperture PSF. (b) Single-Helix phase mask from [2] with L= 7 fresnel zones partitioning. (c) Double-Helix phase mask composed by superposition of (1,1)(3,5)(5,9)(7,13)(9,17) GL modes [3]. Image is taken from Kumar et al. [4].

The Gaussian Laguerre modes are a family of functions that solves the paraxial wave equations. These modes form an orthogonal basis for the two-dimensional complex function that compose $u(\boldsymbol{r})$ and each element of the basis is controlled by the two integer m and n. Increasing the value of n, the effective width of the beams grows. Instead, m determines the number of 2π phase jumps from the center along the transverse plane.

$$m = \pm(n - 2k) \tag{1.15}$$

While n can be any positive integer, m is limited from the formula (1.15) where k is a value between $[0, n/2]$. Figure 1.6 shows the intensities and phase profiles of various GL modes. Itis important to note that the angular spread increases monotonically with n and thus the accuracy of the paraxial approximation becomes poorer as n grows. In addition, the phase discontinuities only occur when the amplitude is zero.

These family of solutions are interesting because every paraxial wave field can be expressed as a combination of the basis functions. Performing a modal decomposition of the wave front, Piestun proposes to represent the information in a modal plane. The result is shown in Figure 1.7 and this representation is also useful to visualize important properties of the wave field.

To our goal, any distribution that owns the property of continuous rotation with propagation is formed by the superposition of modes that below to the same line, where the rate of rotation depends on the slope. If we define z_0 the Rayleigh range

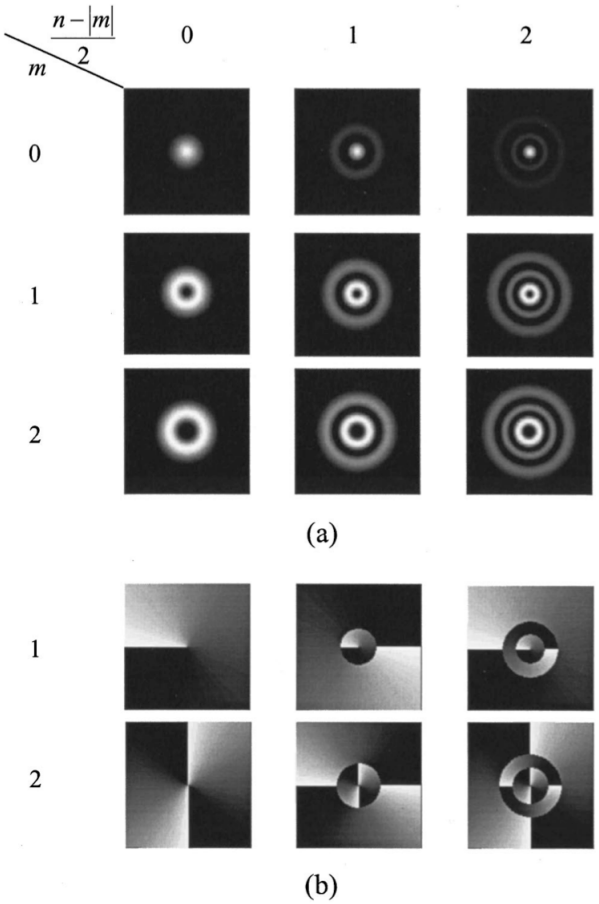

Figure 1.7: Examples of GL modes: (a) intensity, (b) phase. Image taken from Piestun et al. [5].

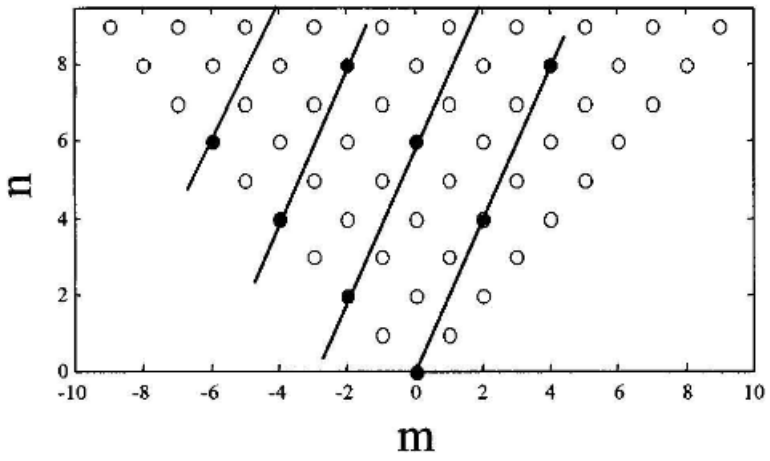

Figure 1.8: Result for the Gauss-Laguerre modal plane. Each mode that composes the wave is represented by a point within grid. Image take from [1].

as follow:

$$z_0 = \frac{\pi(w_0)^2}{\lambda} \tag{1.16}$$

where w_0 is the waist size that controls the transverse scaling of the distribution.

We can compute the angle of rotation at some propagation distance z as:

$$\theta = c\, tan^{-1} \frac{z}{z_0} \tag{1.17}$$

where c is the slope of the line.

In the next chapters we will see how we exploit the superposition of GL modes to build a Single-Helix and a Double-Helix phase mask and, by exploiting the RPSF that are able to produce, how we can code depth cue of the scene in the captured images.

1.4 Occlusion Boundaries Problem

A correct estimate for depth in a general depth estimation framework is a challenging task, due to the two-dimensional nature of the image taken by the camera. Usually, the camera only captures a two-dimensional projection of the surrounding environment, therefore depth estimation algorithms must rely on visual information, such as the shape and size of objects in the image, to infer depth. However, when an object partially or completely covers another object, the camera does not have enough information to estimate the depth of the occluded object, since its point of view is blocked by the foreground object. This makes it difficult to estimate the depth of the occluded object and can cause problems in final applications such as robotics and Augmented Reality task. In addition, the problem of occlusion can be aggravated by the complexity of the objects in the image and the presence of transparent or reflective objects, which can alter the shadows and shapes of objects in the image and make difficult to estimate depth.

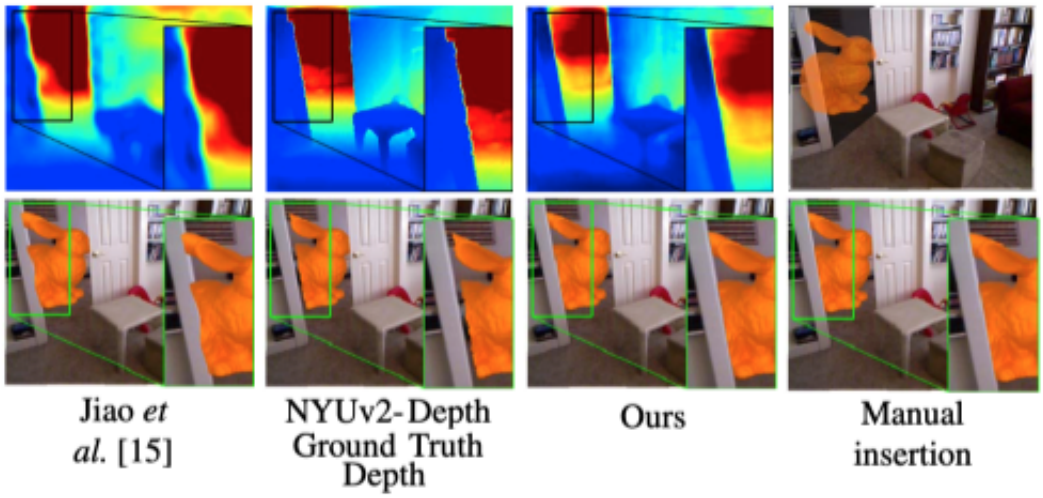

Figure 1.9: In this figure we show an example of augmentation for an RGB image from NYUv2 with a virtual Stanford rabbit using different depth maps for occlusion aware integration. This image is taken from [6] and [7].

There are many different types of problems related to occlusion, in our work we take into account the problem of occlusion boundaries and we will look for possible methods to solve it. In particular, the problem of occlusion boundaries occurs when a foreground object covers a background object in the scene. In this case, the foreground object blocks the view of the object in the background, this results in a poor or inaccurate estimate for the edges, due to the ambiguity introduced in the estimate of the surrounding region.

At the moment, current methods used for predicting depth maps from monocular images tend to generate smooth and inaccurate occlusion contours. This is problematic because occlusion contours are key clues for object recognition and can potentially facilitate the discovery of new objects through scene reconstruction. To improve the accuracy of predicted depth maps, modern methodologies use various filtering techniques or provide an additional residual depth map to improve the initial estimate. For the sake of completeness we mention two works that could be seen as alternatives to solve the occlusion boundaries problem and refine depth maps prediction in monocular case. A first approach is the popular Conditional Random Fields (CRFs). These works [21] typically define a pixel-wise and pair-wise loss term between close pixels using geometric features or reliability maps as guidance. With this element the initial predicted depth map is refined by inference. While most of these methods help improving the initial depth predictions still under-perform state-of-the-art MDE methods. Another idea to refine depth prediction and handle the occlusion boundaries problem is to use enhancement methods. Even if these methods do not explicitly target for occlusion boundaries, they can be a potential solution for it. Speaking about image enhancement, the current state-of-the-art method is the bilateral filtering, in particular as edge-preserving denoising method. Although, historically it was limited as a post-processing approach, due to its high computational complexity, recent methods have shown as bilateral filtering should be useful in downsampling /upsampling schemes. The drawback for this kind of solution is that sometimes it produces false depth gradient artifacts.

In order to give some visual examples we can see what Ramamonnjisoa published in [8]. He presents for the first time a dataset named NYUv2-OC++ where manually annotated the occlusion boundaries in all the images of the test set, splitting from popular NYUv2-Depth dataset [22].

The NYUv2-Depth dataset is a popular MDE benchmark that provides such depth ground truth. Several other datasets related to image contours or occlusion boundaries already exist. However, those datasets often lack annotation between two regions separateed by the boundaries. For this purpose, the NYUv2-OC++ with manual annotation is more reliable than automatic annotation, that could be derived from depth maps that are not perfect.

Figure 1.10: Samples of NYUv2-OC++ dataset taken from [8]. The selected highlighted regions in red rectangles emphasize the high-quality and fine-grained annotations.

1.5 Contribution and Thesis outline

In this thesis will be presented three different approaches to solve the occlusion boundaries problems in the monocular depth estimation environment. Starting from an end-to-end learning pipeline where exploiting a pupil engineer phase mask we are able to encode cues of depth in the original image and recover an initial estimate for the depth [10]. We explore a Post Processing framework called "Displacement Field" that performs a sort of edge enhancer. And two Pre-Processing steps, the first concern in a new Non-Linear Image Formation model, which has the specific task to encode the depth cue in a more accurate way. The second precondition step named "Approximate Inverse" that reduces the total effort required from the Neural Network by computing a layered image in an early stage, wherein each layer has just sharp details in the region that correspond to the ground truth layer. This work is presented as a full experimental study on a computational camera model with the idea of using this simulation to produce a fully working prototype. Outline for the thesis is organized as follow:

- Chapter 2 is dedicated to a brief literature review of the most common approaches for the task of monocular depth estimation, exploring which are the classical methods as well as those based on code aperture. In addition, a review of image deblurring methods and occlusion aware model is presented with particular attention on how occlusion boundaries are handled.

- Chapter 3 shows which is our starting point. We present the initial architecture and all the required steps that allow to recovery the all-in-sharp image and the depth estimate exploiting the RPSF. We focus on the phase mask design part and on the explanation for the Linear Image Formation Model to conclude

with a part related to the Deep Learning Architecture and one for the Image Deblurring.

- Chapter 4 describes the three different approaches adapt to solve the occlusion boundaries problem, along with the full description of the different simulation pipelines with training details for the monocular depth estimation and the image deblurring process.

- Chapter 5 presents all the results that we get, with a brief introduction over the Dataset that we used in the simulation. A comparison between the two technologies used for the phase mask computation is present, together with all the results obtained by the various approaches used.

- Chapter 6 concludes this work presenting the conclusions along with some possible future works.

2 | Related Work

This chapter aims to provide a comprehensive overview of the methods used for monocular depth estimation, with a focus on the three main categories discussed: the classical methods, the coded aperture methods and the end-to-end learning methods. Through a detailed analysis of these categories, the theoretical foundations, specific techniques and performance achieved in monocular depth estimation will be explored. The chapter continue with an analysis of existing methods for image deblurring problems analyzing the difference between blind and non-blind approaches, presenting which methods are useful in this work.

At the end, a brief overview of the works in the literature related to Occlusion Aware Models will allow our analysis to fully understand the current landscape of monocular depth estimation and offer important insights for the development of new approaches.

2.1 Depth Estimation

Accurate depth estimation represents a major challenge in the field of computer vision and has received considerable attention in the scientific literature. In recent years, the application of deep learning approaches to monocular depth estimation has led to significant progress and promising results. This trend has opened new perspectives to address the problem of depth estimation using deep neural networks, which can learn directly from the data and provide more accurate and consistent estimates.

2.1.1 Classical Methods

Classical methods for depth estimation can generally be divided into two categories: active methods (active) and passive methods (passive). This distinction is based on the manner in which depth information is acquired from the scene. However, it is important to note that is not a strictly division and there might be approaches that combine features from both categories or use hybrid strategies to achieve more accurate and complete depth estimates. Active methods generally exploit the projection

of structured light patterns or the emission of light pulses to obtain additional depth information. These include Holography [23] methods, mainly used when coherent light source, as a laser wave, is available. They exploit the wave interference of phase light intensity to records depth cues. This group of methods is severely limited by the need for a coherent light source and a precise optical interference setting. With a more accessible incoherent light source several methods as the structured light [24] and time-of-flight (ToF) cameras [25] became popular and made their ways to commercial products, such as the Microsoft Kinect V1 [26]. These approaches provide greater robustness, as the use of incoherent light can reduce the effect of environmental disturbances, such as specular reflections, improving the robustness of depth estimation algorithms, and offer less complexity, which makes them more versatile as they can be used in a wide range of scenarios, including those with reflective surfaces, such as glass or mirrors.

In contrast, passive methods for depth estimation exploit the information present in the images without requiring active interaction with the scene. Using these techniques, depth estimates can be obtained only from the visual information in the captured images. One of the most widely used is the stereo camera system [27] that takes advantage of the information in images captured by two or more cameras located in different positions, using the disparity between the corresponding points to infer depth information. Depth From Focus (DFF) [28] and Depth From Defocus (DFD) [29] are methods that exploit variations of focus and blurring in images to estimate the depth of objects. They are very good solutions as they do not require additional sensors and can therefore be implemented using a single camera. On the other hand, they suffer more the image quality degradation, the lighting conditions and both methods require a series of image acquisitions with different settings which can lead to increase the acquisition time and limits the real-time applications.

More recently, deep learning methods based on single image have been presented. For example, Eigen et al. [15] shown as two stack deep neural networks, where one predicts the coarse depth map and the other refine it, can reach good level of accuracy. Or as Cao et al. [30] that used a residual neural network and refine the final result using a CRF. Common to all these approaches is the use of depth cues and since the availability of them in a regular RGB image is limited, these approaches require large architectures with significant regularization. To overcome these limitations, a new group of methods based on the coding aperture finds its way into the literature. These approaches can be used to encode depth information in the captured image, where a mask placed in front of the pupil lens is able to encode the PSF of the camera with a specific diffraction pattern.

2.1.2 Coded Aperture methods

Imaging methods that use coded aperture techniques became more common in the last decades. In Chapter 1.3 we have seen how with the introduction of a diffracted optical element (DOE) in the optical model it was possible to modify the PSF in a way that it was able to encode the captured image by inserting information useful for the estimation of the depth. One of the first works presenting this innovation was proposed by Levin et al. [31] where an amplitude mask is inserted in the pupil of the camera to encode depth information in the captured image by diffracting the pattern of the PSF. Levin's idea is then carried forward in Zhou's work in [32] where he presents a criterion for evaluating a pair of aperture masks with respect to the precision of the depth recovery. In [1] it is shown how the use of amplitude masks, which block most of the light reaching the image sensor, or the amplitude modulation of the pupil function can lead to a low light throughput and can reduce the efficiency of the system resulting in a low PSNR value for the capture image. This leaves a low margin for post processing methods that will be used to reconstruct the all-in-focus image or to refine the depth map. To go above those issue, Piestun et al. [5] and Quirin [33] show how it is possible to construct phase masks by exploiting a superposition of GL modes. They presented both a Single-Helix (SH) and a Double-Helix (DH) phase mask that is able to engineer the PSF make it rotate with distance i.e., defocus. Such property as shown in [1] can increase the Fisher information along the depth dimension and can be used as reliable depth indicator in passive ranging task. The RPSFs GL-modes can be further optimized to ensure high light throughput as is show in [3] by Pavani. Where he introduces a new type of PSF named high-efficiency RPSF (HER-PSF) that its transfer function efficiency is over 30 times higher than the plain GL modes base RPSF design. This is possible because the HER-PSF present a phase-only coherent transfer function and so can be implemented with non-absorbing masks. Another work that exploit phase-only mask to encode depth dependency in the PSF is presented in [11], where the author optimized a free-form mask to the task of monocular depth estimation producing a depth dependent PSF whit invariant shape but that rotates with defocus.

Prasad et al. [9] proposed a different approach to generate RPSF that used Fresnel Zones with successive carrying spiral phase profile on larger quantum number. In addition, due to the single-lobe feature in the PSF, thus a SH configuration, the extraction of the variation between defocus levels should be less challenging than the case with two or more lobes. Prasad in [2]presents a follow up over his work, describe a computational-imaging approach that overcame the limitation of PSF in terms of low light throughput and low Modulation Transfer Function (MTF). To this end he divided a circular pupil aperture into L Fresnel zone, where the lth zones

has a radius proportional to $\sqrt{l}$ and impresses a spiral phase of $l\theta$ on the light wave. In [34] and [4] the authors generalize Prasad's earlier work by considering a phase function that allows the generalization to a multi-order-helix RPSF environment, introducing new parameters such as the number of rotating lobes and the inner and outer radius of each zone. While they use a purely empirical approach to determine the value of the parameters, in our work we optimized it together with the weights of a neural network making them optimal for the case of monocular depth estimation.

2.1.3 End-to-end Learning

In recent years, a new trend is catching on and new methods, where deep learning techniques have been used as a tool for end-to-end optimization, are being presented. The key idea of these models is to exploit deep neural networks, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), to construct the physical design of phase masks by starting from a circular aperture and exploiting the parameterization. Or, as in more general approaches, where they models the layers of the optical image formation model as the layers of a neural network and using back-propagation to train the network on large dataset to update and improve the optical construction parameters. A first example is Haim et al. [35] where he presents a novel deep learning framework for the design of phase-coded aperture mask jointly with a Fully Convolutional Network (FCN) modelled ad hoc for the task of singular-image depth estimation. That mask is designed to increase sensitivity related to small defocus changes and introduce a depth-dependent chromatic aberrations over the three primary color. Chang in [36] and Wu in [11], almost at the same time, they both present in two separate works a similar approach that optimized a freeform lens phase mask based on superposition of Zernike polynomials together with a CNN-based depth estimation. An improvement to Wu's work is proposed by Mel in [10] where he shows as the camera model used was unrealistic, because it only consider the additive Gaussian noise. In addition, the design of parametric masks without restrictions and with high degrees of freedom can lead to local minima obtaining sub-optimal performance. That is why he proposed a phase mask parameterized just with respect three parameter and optimized in an end-to-end manner with the weights of a U-Net [37].

2.2 Image Deblurring

The use of RPSF and the consequent introduction of different levels of blur in the captured image, combine with our desire to obtain sharp images and sometimes even all-in-focus image. It forces us to deal with a very popular topic in literature, the

Image Deblurring. Image Deblurring aims to restore the original image appearance by eliminating or reducing the blurring effect introduced by RPSF, this results in sharper images with greater clarity and detail, and can also be useful as refinement methods for depth maps. There are several approaches to handle the problem of Image Deblurring. One of these is the "blind" approach, which refers to methods where there is no direct knowledge about the PSF. In those cases, the goal is to estimate both the deblurred image and the PSF from the degraded image only. This requires sophisticated algorithms that can accurately infer the PSF and restore the original image. On the other hand, and more in line with our case, there are also "non-blind" approaches that assume to have predefined knowledge about the PSF. These approaches exploit the information about PSF to directly apply deconvolution over the degraded image. Due to the fact that the problem of Image Deblurring is considered ill-posed because it requires the recovery of information that has been degraded or lost due to the blurring process, it is necessary to introduce prior knowledge and regularization techniques to accurately recover the sharp image with all its details.

In the formula (2.1) we will show which is the relation between blurred noisy measure image y and its ideal sharp counterpart x, where k and n are the blur kernel and the noise term respectively,

$$y = k * x + n \tag{2.1}$$

Between the deblurring methods that can be used to restore images degraded due to blurring we want to mention the Wiener deconvolution filter [38] that uses an estimate of the original signal power spectrum and the one of the blur to deconvolute the image. Richardson-Lucy deconvolution [39], is an iterative algorithm used for image deconvolution that starts from an initial estimate of the original image and iteratively refines the image using the known blur and the degraded image as input. As last we mention the Tikhonov Regularization used by Ikoma in [14]. This method tries to minimize a cost function that acts as trade-off between data fidelity and the spatial regularity of the deblurred image. However, all this approaches suffers from the common problem of noise sensitivity, slow convergence and sometimes may result in deconvolution artifact such as ringing near the edge of the objects.

Even there, the introduction of neural networks has found space for application. Deep learning-based techniques are now used in various image processing tasks, such as deblurring and denoising of images and for super-resolution imaging. A first work is present by Xu et al. [40], he use the SVD (singular value decomposition) over the pseudo-inverse kernel to initialize the parameter of a neural network. Zhang et al. [41] use a fully convolutional neural network to learn the gradients of the

featurescoming from the deconvolution perform in the image space. This method is able to achieve a good image quality level but the requirement of designing these two different frameworks will make this approach non optimal for the task of image deblurring. At the end, another deconvolution approach was introduce by Dong and Roth in [42] where they proposed to perform an explicit deconvolution in the feature space and integrating them with the classical Wiener deconvolution framework base on deep learned features, and then a multi-scale feature refinement stage predicts the deblurred image from the deconvolved features.

2.3 Occlusion Aware Model

In the Monocular depth estimation the occlusion is a fundamental concept to take in consideration. Despite recent improvements in depth estimation using deep neural networks, the estimation of occlusion boundaries remains difficult and many times inaccurate while it still remain a key feature for such tasks, like object recognition and augmented reality. The possible cause for this problem might be result from the poor quality of depth annotation along the occlusion boundaries. Especially if they are obtained with stereo reconstruction methods or a structured light camera as it is for the NYUv2-Depth dataset. This is because in those cases the occlusion boundaries are computed starting from two images of the same scene where almost surely one or both sides are not visible. Moreover, the occlusion boundaries represent a small part of the image so many approaches do not pay much attention to this detail as it may not affect so much the final result.

In this context, a variety of solutions have already been proposed as first Wang et al. [21] present the SURGE method that improve the scene reconstruction on planar and edge regions by learning to jointly predict depth and normal maps. Lepetit in. [6] present the SharpNet. Inspired by recent methods, they show how it is possible to learn and reconstruct occluding contours by adding a simple term in the loss function. Specifically, they train a first NN to predict depths, normals and occlusion boundaries from a single image, then by minimizing a loss function with specific constrain between depth and normals is able to demonstrate that can improve the depth output. Ramamonjisoa in. [8] introduces a simple method to overcome smooth occlusion boundaries by exploiting a 2D displacement field that can re-sample pixels around the boundaries resulting in a sharper reconstructions.

As last work we mention what Ikoma do in. [14], inspired by depth from defocus and the emerging PSF engineering approaches, he start from an analysis that shows as the works [11] [35] [36] do not take full advantage of the available monocular depth cues. Continue by showing as the linear optical image formation models do not model in accurate way the defocus blur at occlusion boundaries. For this

purpose, he first proposed a nonlinear occlusion-aware optical image formation that encode the defocus blur at depth discontinuity in a more accurate way. Secondly, similar to our proposed approach, he derives a preconditioning step that applies an approximate inverse of optical image formation model to the sensor measurements. This preconditioning step makes considerably easier for the CNN network to deduce the depth map from the captured sensor image.

3 | Depth Estimation using RPSF

As shown in Chapter 2, the introduction of Deep Neural Networks and the use of aperture-based coded methods has lead to the development of several useful approaches to solve the problem of Monocular depth estimation. In this chapter we will go through an analysis and mathematics of the tools used in our project. In this work the depth-dependent RPSF is obtained from a phase mask using Fresnel zones, this approach was first present by Prasad in [2] and then used by Mel in [10]. Following what Mel did in [43], we will then go on to show which elements were used to simulate the Linear Image formation model and for the Deblurring module.

3.1 Phase Mask Design

The design of Fresnel zones phase mask proposed by Prasad is inspired in some measure by the spiral phase plat (Figure 3.1) that has been used in [44] where they demonstrate how convert a laser beam into a pure Orbital Angular Momentum (OAM). When a light wave passes through a Fresnel zone, the phase difference between adjacent regions due to the increased thickness of the spiral plate, causes a change in the shape of the wave, which can be interpreted as a screwing or twisting effect. This phenomenon is often referred as "optical vortex". The light wave takes a spiral shape, with the wave front winding around the propagation axis.

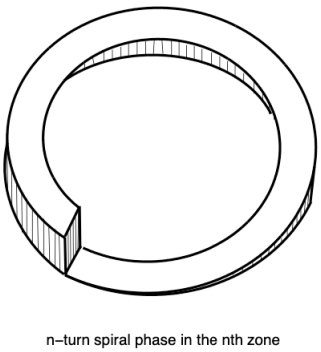

Figure 3.1: Schematic diagram of a specific Fresnel zone with its spiral phase retardation, taken from [9].

To create a Fresnel phase mask we can starting to consider a circular pupil of radius R which is subdivided into L different contiguous Fresnel zones where each zone is bounded above from an outer radius $R_{l_{th}}$ computed as follow:

$$R_{l_{th}} = R\sqrt{\frac{l}{L}} \tag{3.1}$$

where each l_{th} zones is equipped with a spiral phase profile of form $l\phi$, the l_{th} zone completes l phase cycles as the azimuthal angle ϕ completes a single rotation around the optical axis. With these arrangements the phase delay introduced by the mask defined above has the form

$$\psi(u,\phi) = \left\{ l\phi | \sqrt{\frac{l-1}{L}} \leq u \leq \sqrt{\frac{l}{L}}, l = 1, ..., L \right\} \tag{3.2}$$

Here the function $\psi(u,\phi)$ denotes the pupil phase, and it is a function of normalized radial coordinates $u = |\vec{u}|$ with respect to the pupil phase position $\vec{u}$ and the azimuthal angle ϕ.

The intensity distribution computed form that phases function return in a single helix PSF that rotates as function of defocus. This phase function can be further generalized to a multi-order-helix RPSF. In order to achieve such flexibility Kumar et al. [4] and Berlich et al. [34] has extend the formula as follow:

$$\psi(\rho,\phi) = \begin{cases} \phi & 0 \leq \rho < (\frac{1}{L})^{\epsilon} \\ [(l-1)N+1]\,\phi & (\frac{l-1}{L})^{\epsilon} \leq \rho < (\frac{l}{L})^{\epsilon} \\ [(L-1)N+1]\,\phi & (\frac{L-1}{L})^{\epsilon} \leq \rho < 1 \end{cases} \tag{3.3}$$

In this case the function is defined in terms of polar position, that means ρ is normalized by the pupil radius R, $\vec{u} = \vec{\rho}/R$.

Knowing that the phenomenon of the RPSF in function of defocus can be inferred from the expression $h_c(s,\phi;\Psi)$ of the coherent PSF defined by Prasad in [9]:

$$h_c(s,\phi;\Psi) = 2\sqrt{\pi}e^{-i\Psi/2L}\frac{sin(\Psi/2L)}{\Psi}\sum_{l=1}^{l=L} i^l e^{-il(\phi-\Psi/L)} J_l(2\pi\sqrt{l/L\rho}) \tag{3.4}$$

where J_l is the Bessel factor, and Ψ is the defocus parameter defined in Eq 1.13, notice that the bound for the summation must be $[(l-1)N+1]$ is written as simply l in the formula. At this point, we have reached a definition for the PSF that can be controlled by basically 3 parameters: the number of Fresnel zone L and the design parameters N and ϵ.

Mel in his work [43] showed how the control of each of those parameters can affect the final shape of PSF. N controls the number of lobes of the generated PSF, for example if we want to produce SH-PSF or DH-PSF the parameter must be set respectively equal to 1 or 2, (in Fig.3.2 we show a set of N-order helix PSF). Consider that increasing the number of peaks N, reduces the depth range of the RPSF as an ambiguity is introduced whenever the rotation exceeds $\frac{\pi}{N}$.

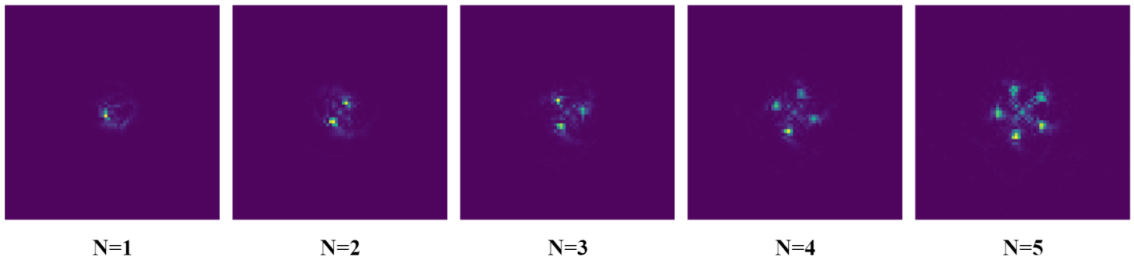

Figure 3.2: Generated N-order helix PSF with different number of rotating lobes $[L = 5, \epsilon = 0.9]$.

The third design parameter that appear in the formula is ϵ, as it was empirically theorized by Berlich and also as we can see from Figure 3.3, that parameter controls the peaks separation as well as the confinement of each peak.

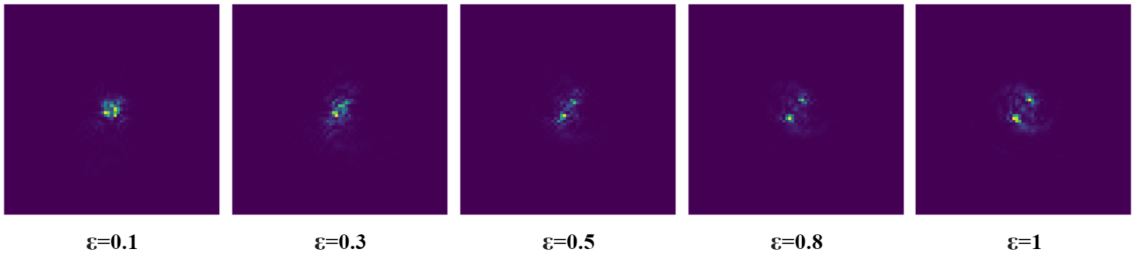

Figure 3.3: Generated RPSFs with different values of ϵ with $[N = 2, L = 5]$.

In summary, we understood how $[N, L, \epsilon]$ can be jointly optimized to adapt the resulting PSF depending on the task, changing its shape and the depth range thus making it an excellent tool to encode depth information in the captured image.

Furthermore, the fact that we want to optimize the design parameter of phase mask with a neural network, collides with the problem that Mel solved in [10]. The phase function proposed in Eq (3.3) is not differentiable with respect to the design parameter N and ϵ. This can be solved by approximation with a 2D *tanh* function defined in polar coordinates that goes to simulate the ring masks defining the boundaries of each Fresnel zone. In Figure 3.4 it is illustrated a scheme that shows how the *tanh* function modeled the different Fresnel zone and how each phase array is multiplied by the corresponding mask. The resulting regions are summed

up together to obtain the final phase profile of the Fresnel phase mask, which will then be optimized for the task Monocular Depth Estimation. Later in Chapter 5 we will see which are the results for the PSF obtained from the optimization of this definition, and we will compare them with the phase masks obtained in Chapter 3.5.

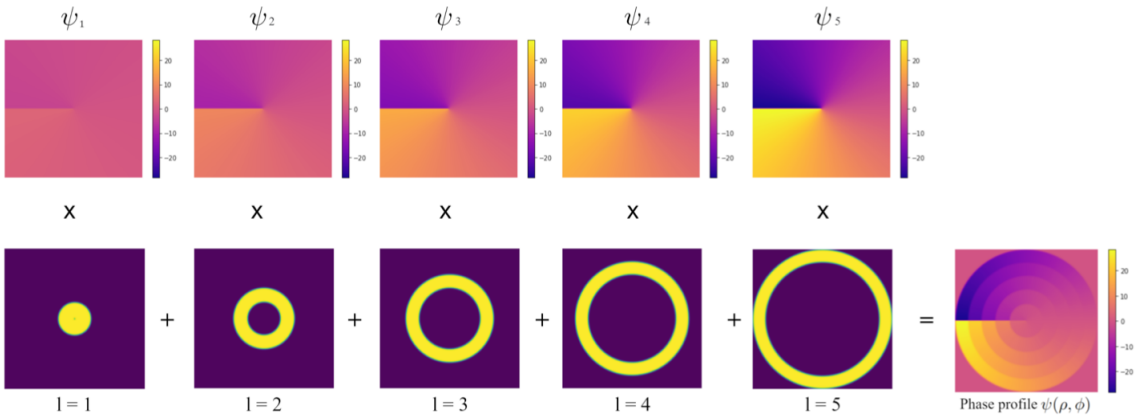

Figure 3.4: Differentiable phase mask approximation with 2D tanh functions.

3.2 Initial Architetture

Our proposed solution is a continuation of what is described by Mel in his work [10] [43]. To this end, as previously done for the description of the phase mask we now see which his proposals are to handle the image formation model and the image deblurring part. To have an overview of the entire pipeline we can divide that into three main stages. In the first stage, following the description in Section 3.1, the height map of the phase mask is specifically optimized along with the weights of a DNN. The second stage is responsible for taking the height map from the previous stage and incorporate it into the optics module as a part of simulation for a full digital image pipeline. In this stage also different kinds of noise are added to the captured and encoded images. At the final stage, the noisy output is used to fine-tune the network meanwhile a dedicate network recovers the all-in-focus image using a non-blind and non-uniform image deblurring module.

3.2.1 Linear Image Formation Model

The image formation model considers several factors that influence the creation of an image, including lighting, scene geometry and the optical properties of the camera or sensor used to capture the image. As the others that use the same principle, for simulation purposes Mel decided to approximate the input image with a layer depth

model where a finite number of planes are used to create the encoded blur image by convolution with the corresponding depth dependent PSF. The model can be visualized in Figure 3.5 and described by the following equation:

$$I^B = \sum_{d=1}^{d=D} (I^S * RPSF_d) \circ M_d \tag{3.5}$$

where I^B is the coded blur image, I^S is the sharp all-in-focus image, the $RPSF_d$ is the RPSF at the depth layer d, and M_d is the corresponding mask relate to the depth plane d and $\circ$ is the element-wise multiplication. Please, notice that the sum of all mask respect to d is equal to 1, as we can see from Eq.(3.6).

$$\sum_{d=1}^{d=D} M_d = 1 \tag{3.6}$$

This linear approach does not model the occlusion; thus, the render image is not accurate near the depth discontinuity. In chapter 4 we present a new non-linear image formation that models defocus blur at occlusion boundaries in a more accurate way.

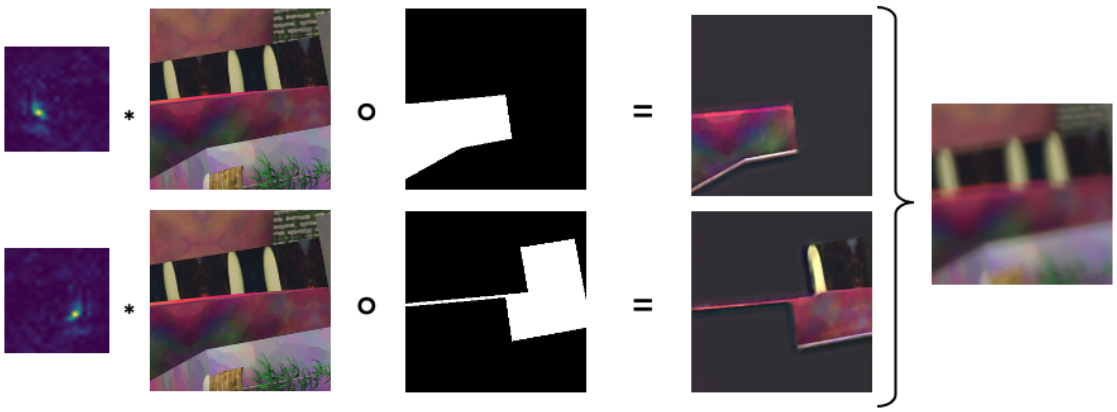

Figure 3.5: Representation that shows how a single layer for the coded blurred image is computed, starting from the all-in-focus image and applying first a convolution with the $RPSF_d$ and then multiplying by the corresponding mask.

Lighting plays a key role in determining the appearance and detail of the image. The amount of light in the scene, its direction and spectral distribution influence the color and brilliance of the final image. Take in consideration this effect Mel decided to use two tricks in the simulation of the camera model. First he assumes that the phase mask is mounted on the back side of the aperture of the imaging system. In this way, the mask would act directly on the light distribution so each point is blurred with the corresponding RPSF. Second, in order to obtain the phase

mask, he trains the first network using noise-free image data and the camera noise will be add later in the camera model. Fig.3.6 represents the second stage of the pipeline where Mel applied different stages of camera pipeline.

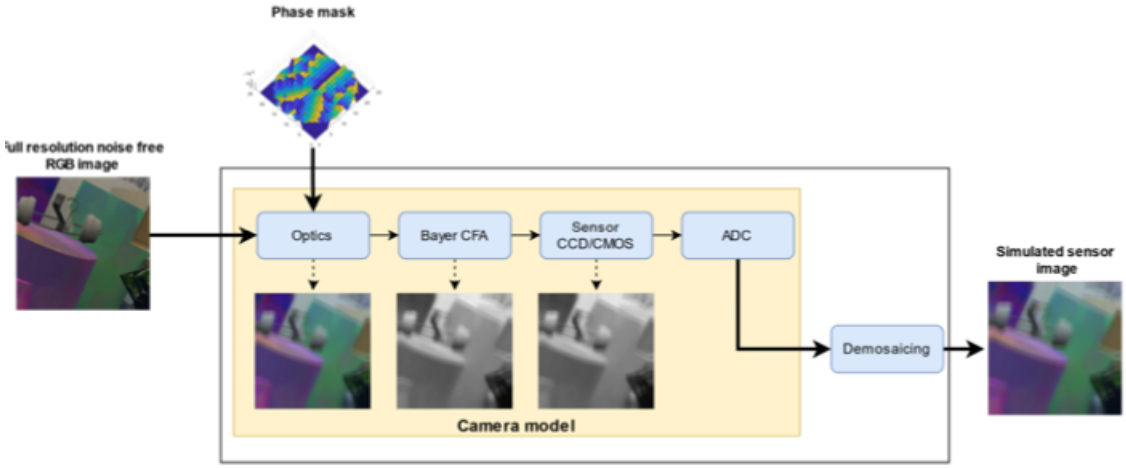

Figure 3.6: Stage 2 of the pipeline proposed by Mel in [10]. It used to simulate a realistic image formation model.

As we can understand from the figure, once we obtain the full resolution blurred image as output of the optics module, the Bayer CFA downsampled the input into a color pattern of the form RGGB and read and shot noise are added to simulate the sensor chip. Finally, as the last part of the camera model the image is quantized by an ADC unit and a high-quality linear interpolation demosaicing is used to recover the full color channels to compute the final input.

3.2.2 Deep Learning Architecture

As we set out in Chapter 2, the introduction of neural networks has revolutionized the field of image processing and, in particular, brought important innovations to address the problem of depth estimation. Before, traditional approaches were based on rules, geometric models or statistics, requiring the manual extraction of features and the design of complex algorithms to derive an accurate estimate. In particular, CNNs have demonstrated their ability to automatically learn relevant features from the input data and they are perfectly adaptable in our application.

The CNN used to jointly optimize the design parameter of the phase mask is a simple U-Net as one presented by Ronnerberger et al [37]. The network is divided in two stages, the down-sampling part is composed of 5 layers, where every layer in this part presents the following step:

$$(\{Conv - BN - ReLU\} \times 2 \rightarrow MaxPool2 \times 2) \tag{3.7}$$

where BN stand for Batch Normalization, and in the up-sampling part where $Conv^T$ is the transpose convolution:

$$(Conv^T + Concat \rightarrow \{Conv - BN - ReLU\} \times 2) \tag{3.8}$$

The network was implemented with skip connection to avoid problems, such as the gradient vanishing, during the training. This network takes in input a batch of coded blurred image and return in the output the predicted depth maps with the same resolution as the input.

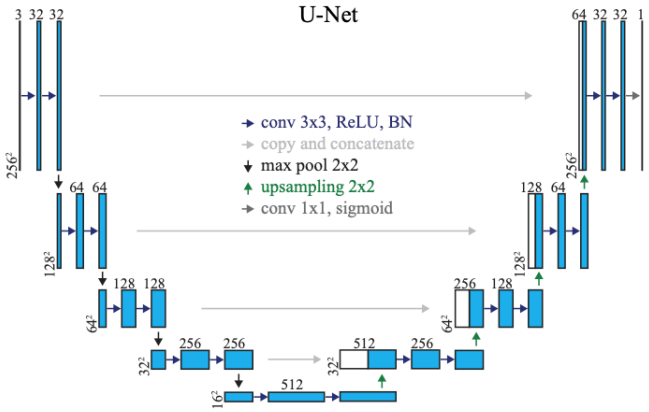

Figure 3.7: Representation of the standard U-Net used in [10] and in [11].

For what concern the network parameter used during the training process, we found two different learning rates, $Lr_{optical} = 0.1$ which refers to the parameter of the phase mask, and the $Lr_{digital} = 1e-4$ which was related to depth estimation network. The network is trained for 150k iterations for the first stage and for another 50k to fine-tune the phase mask in the third stage. Both times Mel uses a batch size of 20 images in order to regularize as much as possible the result, and the well know Adam optimizer [45] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ as value for first and second momentum. As also Wu do in [11], a combination between Root Mean Square Error (RMSE) and gradient loss is used to force the network for predicting good depth maps

$$L_{depth} = L_{rmse} + L_{grad} \tag{3.9}$$

where the RMSE loss $Lrmse$ is expressed as:

$$L_{rmse}(\theta, \theta^*) = \sqrt{\frac{1}{|T|}\sum ||\theta - \theta^*||^2} \tag{3.10}$$

and the same for the gradient loss $Lgrad$:

$$L_{grad}(\theta, \theta^*) = L_{rmse}(\frac{\partial\theta}{\partial x}, \frac{\partial\theta^*}{\partial x}) + L_{rmse}(\frac{\partial\theta}{\partial y}, \frac{\partial\theta^*}{\partial y}) \tag{3.11}$$

know that θ and θ^* are the prediction and the ground truth, $|T|$ is the number of samples in each batch and (x, y) are the coordinates of the images.

In Chapter 5 where we show the results, there is a fair comparison between the different proposals.

3.2.3 Image Deblurring Module

The use of coded aperture method to solve depth estimation problems is faced with a common problem. The addition of blur resulting from convolution with the depth-dependent RPSF cannot be avoided and it is increasingly complicated to find valid methods to recover the all-in-focus image.

To mitigate this problem, several techniques and algorithms have been developed for deconvolution of images acquired with coded apertures methods. These algorithms attempt to reverse the blurring effect introduced by the phase mask, restoring sharpness and detail to the image. One of these solutions is first presented al later improved by Dong et al. [42] and [12].

Starting from their solution, which we will describe shortly, we obtain a non-blind and non-uniform image deblurring approach that fits perfectly to our problem.

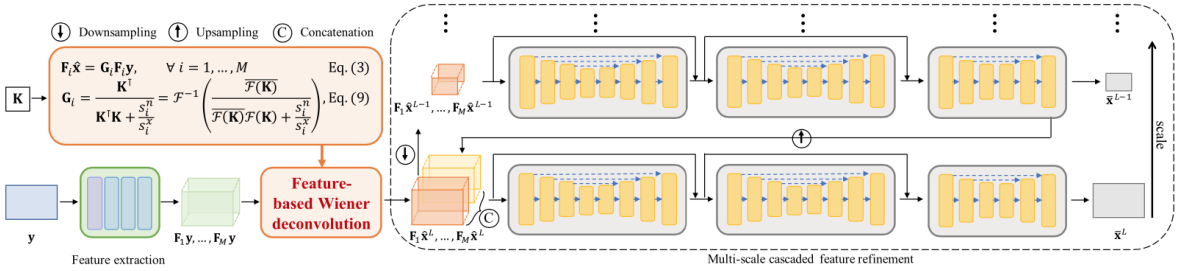

Figure 3.8: Graphical representation of Deep Wiener deconvolution network. First extracts useful feature information from the blurry input image and then conducts an explicit Wiener deconvolution in the (deep) feature space. A multi-scale cascade encoder-decoder network progressively restores clear images, with fewer artifacts and finer detail. Image taken from [12].

Looking at Fig. 3.8, recalling the relation defined in Eq (2.1) and define $\{f_i\}_{i=1}^{M}$ as the set of M linear filter applied in the feature extractor stage, we can write the output of that stage as:

$$F_i y = K F_i x + F_i n \qquad \forall i = 1, \ldots, M, \tag{3.12}$$

where F_i, K, y, x, and n denote the matrix/vector forms of f_i, k, y, x, and n.

The goal of feature-based Wiener deconvolution module is to explicitly decon-

volve the blurry features $\{F_i y\}$ from Eq (3.12) to obtain the latent features as

$$F_i \hat{x} = G_i F_i y \qquad \forall i = 1, \ldots, M, \tag{3.13}$$

where $\hat{x}$ is the latent clear image, by finding a set of feature-based Wiener deconvolution operators $\{Gi\}$. To get the latent features as close as possible to the clear features we can define and minimize the MSE.

$$e_i = \mathbb{E}(|F_i x - F_i \hat{x}|^2) \tag{3.14}$$

Assuming independent and zero mean noise from the latent clear image, we can derive that

$$e_i = (1 - G_i K)(1 - G_i K)^T \mathbb{E}(|F_i x|^2) + G_i G_i^T \mathbb{E}(|F_i n|^2) \tag{3.15}$$

Computing the derivate with respect to G_i and set it to zero we obtain the feature-based Wiener deconvolution operators as:

$$G_i = \frac{K^T}{K^T K + \frac{\mathbb{E}(|F_i n|^2)}{\mathbb{E}(|F_i x|^2)}} = \mathcal{F}^{-1}\left(\frac{\overline{\mathcal{F}(K)}}{\overline{\mathcal{F}(K)}\mathcal{F}(K) + \frac{\mathbb{E}(|F_i n|^2)}{\mathbb{E}(|F_i x|^2)}}\right) \tag{3.16}$$

Where $\mathcal{F}$ denotes the Fourier transform and $\overline{\mathcal{F}}$ is its complex conjugate.

The deconvolved feature maps are refined in a two-stage module with residual-based auto-encoders. The first stage produces a deblurred image at half of the initial resolution, using a deconvolved feature map. In the second stage, the full-resolution feature maps and up-sampled features from the previous stage are combined to produce an all-in-focus sharp image at the desired resolution.

This implementation is further improved in [12] where more attention is placed to the multi-scale feature refinement by implementing multiple encoders and decoders in the early stages of the cascade to capture extensive contextual information and later less information but more focus on details. In his work Mel [10] modified this network to allow it to work with non-uniform kernels due to the fact that the RPSF is spatially variant.

For reasons that we will describe later in the thesis, we also have modified the network presented in [12] to obtain an intermediate result, where starting from the coded blur image, iterating the process for all the depth layer, just some details in the corresponding region of the image is recovered, leaving the others invariant.

3.3 RPSF Prototype

One of the initial aims of the project included the production and testing of a fully working prototype based on what was done by Mel. To this end, and once received the news about the problems that arose in the laboratory during the construction of the phase mask, we thought of testing the optical model used by Mel with a different technology to build the phase mask. In Chapter 1.3 we introduced the GL-modes as one of the possible solutions for the paraxial wave equations (Eq. 1.14), and we can see how it can be exploited for the construction of a Diffracted Optical Element (DOE) that is able to encode depth information in the PSF.

In this section we will analyze which are the necessary step to build a phase mask with the GLmodes and how we compare the two implementations.

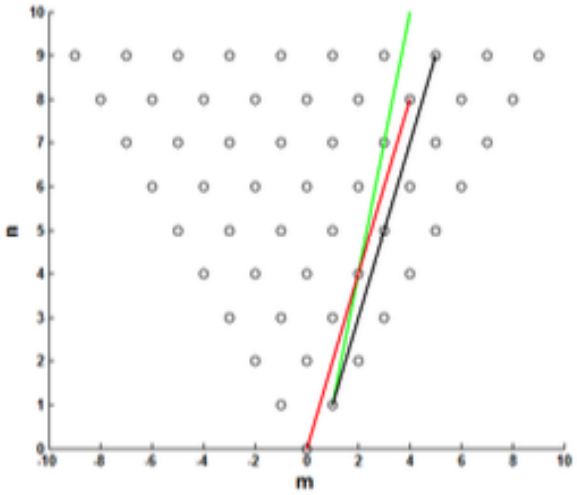

Figure 3.9: Modal plane that represents the possible combination for GL modes that produce a rotating beam. Green (SH), Black (DH) Image taken from [13].

Following what Arsalan et al. [13] did, where they showed that in order to obtain a rotational beam to engineer the PSF, we must choose a combination of GL modes lying on the same line, as it shows in Figure 3.9. Every paraxial wave-field is composed as a weighted combination of GL basic functions.

If we start considering a scalar wave-field represented as:

$$\mathcal{U}(r,t) = U_{nm}(r)exp(i(kz - wt)) \tag{3.17}$$

where $r = (\rho, \phi, z)$ are the cylindrical coordinates and t and w are the time and the angular frequency. The reduced wave field, $U_{nm}(r)$ in paraxial approximation can be represented in terms of GL modes as follows:

$$U_{nm}(r) = G(\rho, \hat{z})R_{nm}(\rho)\Phi_m(\phi)Z_n(\hat{z}) \tag{3.18}$$

From (3.18) we can recover all the GLmodes $G(\rho, \hat{z})$, once we know that $R_{mn}(\rho)$ is a rotational operator, and $\Phi_m(\theta)$ and $Z_n(\hat{z})$ are two exponential term that controls phase and amplitude with respect to m and n the order of modal plane. A complete description of all the mathematics behind those formulas are provide by Arsalan in Chapter 4 of his work [13].

In Fig. 3.10 we report some examples of phase masks with correspondent RPSF that we can obtain by superpositions of GL modes, by following what is shown in Fig.3.9.

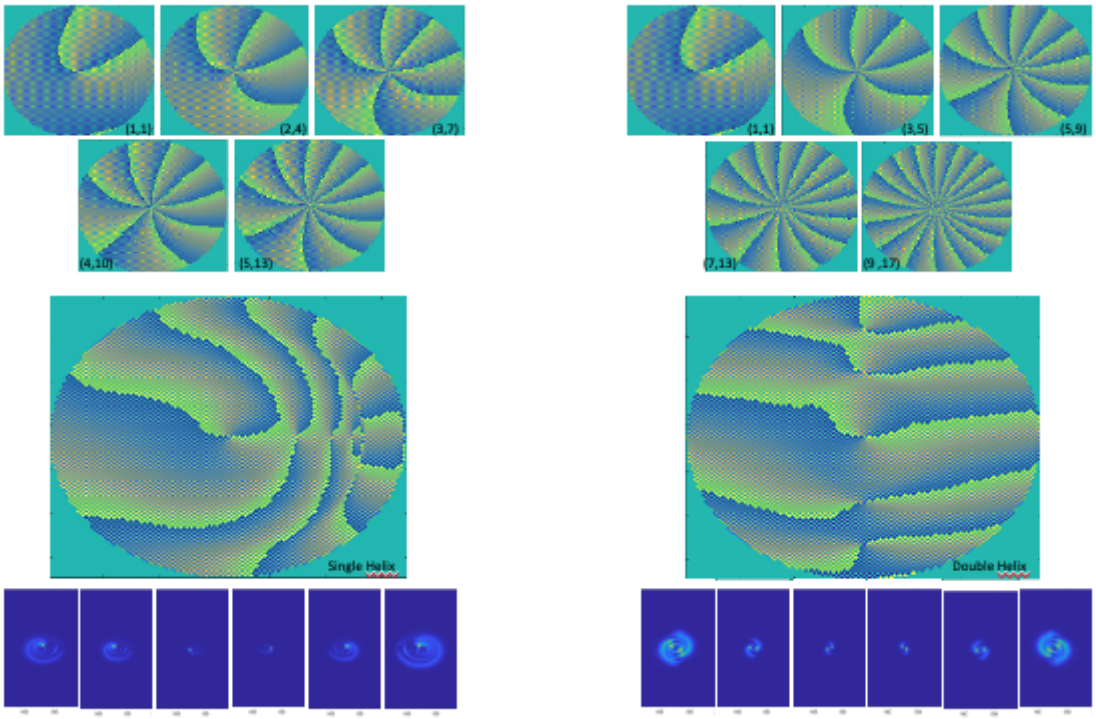

Figure 3.10: Examples of SH and DH phase masks, on top there are the phase representation of GL modes that follow the combination described in Fig.3.9 can produce the phase mask report in the middle row. On the bottom is reported the corresponding RPSF where we can clearly distinguish the different configurations of the mask and also the depth dependency.

4 | Occlusion Handling

The occlusion problem in depth estimation field is one of the main challenges for computer vision algorithms. Despite recent improvements gained using neural networks, the problem of occlusion, and in particular those related to occlusion boundaries, still remains difficult to solve. As we have already discussed in Chapter 2, there are a number of methods that attempt to overcome these problems, but sometimes they result in inaccurate estimates for the depth maps.

In particular, this chapter is used to give a detailed analysis of our approaches used to improve the current state of the art in terms of monocular depth estimation. Specifically, we will divide the proposed approaches into two categories, which can be associate with pre-processing and post-processing methods.

4.1 Pre-Processing Approach

Pre-processing approaches include all those methods that perform operations on the image before the algorithm is applied, or in our case, before it is inserted into the neural network. The main objective of pre-processing approaches is to improve the quality of the image and make the information more easily to detect.

In the field of depth estimation, the most common operations are those related to filtering applied to reduce noise in the image or to eliminate some undesirable components. There are also normalization techniques used to regularize the illumination in the images, or some detection techniques where, for example, are used to highlight edges or important details.

4.1.1 Non-Linear Formation Model

In Chapter 2 we have shown an example of an E2E method that exploits a linear optical image formation model. Meanwhile this model works well for images with locally constant depth region, it fails to model defocus blur at occlusion boundaries. It's well known in the computer vision community that the defocus blur provides a stronger depth cue respect to pictorial cues. To this end, we integrate in the pipeline

a non-linear image formation model based on alpha compositing[1], which should be able to model defocus blur at depth discontinuities in a more accurate way.

Starting from a variant of simple linear image formation model with the form

$$b(\lambda) = \sum_{k=0}^{K-1} RPSF_k(\lambda) * l_k(\lambda) + n \tag{4.1}$$

where $*$ is the operator for convolution, $b(\lambda)$ is a single wavelength component of the coded image and n is the noise. In this case the input RGBD image is subdivided into K depth layer l_k, where for $K = 0$ we refer to the farthest layer. We can express the non-linear image formation model as follow:

$$b(\lambda) = \sum_{k=0}^{K-1} \tilde{l_k} \prod_{k'=k+1}^{K-1} (1 - \tilde{\alpha_{k'}}) + n \tag{4.2}$$

where items not yet mentioned are defined as:

$$\tilde{l_k} := (RPSF_k(\lambda) * l_k)/E_k(\lambda) \qquad \tilde{\alpha_k} := (RPSF_k(\lambda) * \alpha_k(\lambda))/E_k(\lambda) \tag{4.3}$$

where α_k are the binary mask, and $E_k(\lambda)$ is a normalization factor to take into account the energy at the transition of depth layers.

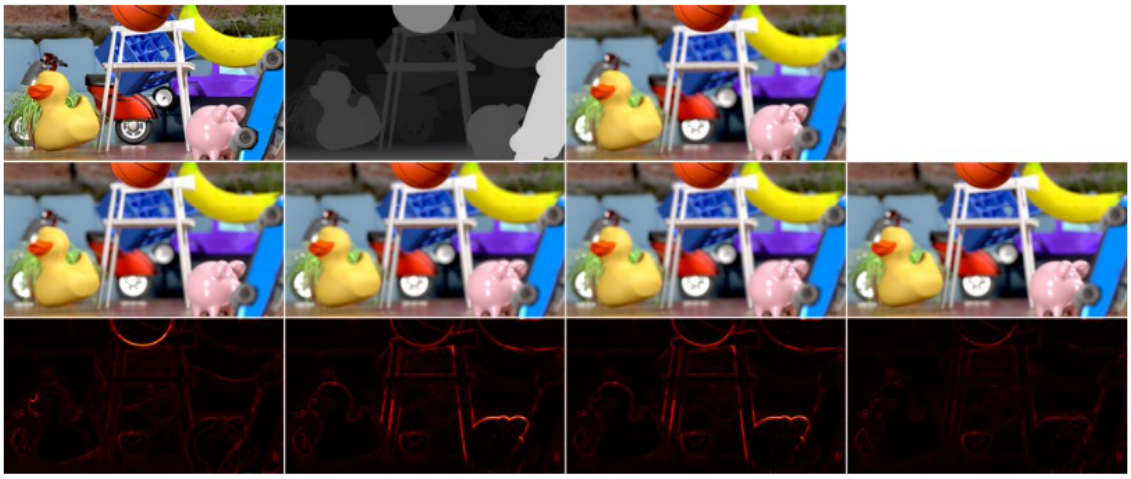

Figure 4.1: Comparing image formation models that simulate defocus blur from an RGB image (top left) and a depth map (top center left) with an accurate simulation (top center right). We compare a simple linear image formation model (bottom left), the variants of the linear model proposed by Wu et al. [11](bottom center left) and Chang et al. [36] (bottom center right), and non-linear image formation model (bottom right).

[1]Alpha compositing is a technique used in computer graphics to combine several overlapping images to create transparency and blending effects between visual elements.

A visual result for the comparison between different approaches find in the literature and the implementation of [14] for non-linear image formation model we refer to Fig.4.1.

From the figure we can clearly see as the non-linear image formation model produces a more realistic defocus image from the all-in-focus image provided as input. With respect to all the other models it produces more accurate defocus blur around texture and occlusion boundaries. The error maps shown in Fig.4.1 are computed with respect to the ray-traced[2] ground truth sensor image.

4.1.2 Approximate Inverse Step

Many times, the use of artificial neural networks, including deep learning models, may encounter various problems related to their capacity learning and performance. In our specific case proposed in Chapter 3, once the coded image was obtained, a fine-tuning over the previously trained network was performed to optimize it on noisy data for the task of the monocular depth estimation. Then to retrieve the all-in-focus image, a dedicated network was used which perform non-blind and non-uniform image deblurring. To this end we first present an approximation inverse step proposed by Ikoma et al. [46] and then we will see which are modifications required to integrate it into our pipeline.

Ikoma noticed that although the linear image formation model, (refer to Eq (4.1)), is not accurate at the occlusion boundaries, it can be exploited to project a framework to work as preconditioning step before the final stage. Specifically he formulates an inverse problem where the goal is to find a multiplane representation $l^{est} \in \mathcal{R}^{M \times N \times K}$ from the coded image. To do so he reformulates the problem as a Tikhonov-regularized least squares problem (Eq. 4.4). The Tikhonov-regularization is a general technique used in machine learning to reduce overfitting and to improve the generalization of models.

$$l^{(est)} = \underset{l \in \mathcal{R}^{M \times N \times K}}{\operatorname{argmin}} \left|\left| b - \sum_{k=0}^{K-1} RPSF_k * l_k \right|\right|^2 + \gamma ||l||^2 \tag{4.4}$$

where γ is the regularization parameter and all the other parameters are defined as before. We further reformulate the problem in the DFT domains (Eq. 4.5) to make it separable and allows us to solve it for each spatial frequency f_x, f_y.

$$\widehat{l}^{(est)} = \underset{\hat{l} \in \mathcal{R}^{M \times N \times K}}{\operatorname{argmin}} \left|\left| \widehat{b} - \sum_{k=0}^{K-1} \widehat{RPSF_k} \circ \hat{l_k} \right|\right|^2 + \gamma ||\widehat{l}||^2 \tag{4.5}$$

[2]Ray tracing is a rendering technique used to create realistic images by tracing virtual light rays. This technique simulates the behavior of light in the real world, allowing for very accurate lighting, reflection and refraction effects.

where $l \in \mathcal{R}^{M \times N \times K}$ is the multiplane image with K layers, $\widehat{.}$ denote the DFT of the variables and $\circ$ is the element-wise multiplication. Equation (4.6) to (4.9) are all the step requred, presented in [46], to reach a closed form solution.

$$\widehat{l}^{(est)}[f_x, f_y, 1:K] = \underset{\hat{l}[f_x,f_y,k] \in \mathcal{C}^K}{\operatorname{argmin}} \left|\left| \widehat{b}[f_x, f_y] - \sum_{k=0}^{K-1} \widehat{RPSF}_k[f_x, f_y, k] \hat{l_k}[f_x, f_y, k] \right|\right|^2 + \\ + \gamma ||\widehat{l}[f_x, f_y]||^2 \tag{4.6}$$

$$= \underset{\widehat{l} \in \mathcal{C}^K}{\operatorname{argmin}} \left|\left| \widehat{\boldsymbol{b}} - \boldsymbol{P}\widehat{\boldsymbol{l}} \right|\right|^2 + \gamma ||\widehat{\boldsymbol{l}}||^2 \tag{4.7}$$

$$= \left(\boldsymbol{P}^H \boldsymbol{P} - \gamma \boldsymbol{I}\right)^{-1} \boldsymbol{P}^H \widehat{\boldsymbol{b}} \tag{4.8}$$

$$= \frac{1}{\gamma} \left(\boldsymbol{I} - \frac{1}{\gamma \boldsymbol{P}\boldsymbol{P}^H} \boldsymbol{P}^T \boldsymbol{P} \right) \boldsymbol{P}^H \hat{\boldsymbol{b}} \tag{4.9}$$

Here, $\widehat{l}[fx, fy, 1:K] = \widehat{l} \in C^K$ is a column vector with the values of a single spatial frequency of the multiplane image across all layers $1...K$, $P \in C^{1K}$ is a complex-valued matrix with just a single row but K columns, each corresponding to the value of the DFT of the RPSF, $\widehat{RPSF}$ (i.e., the optical transfer function) at fx, fy at layers $k = 1...K$, and $I \in R^{K \times K}$ is the identity matrix.

Due to the implementation with the FFT, and edge-tampering is used to reduce the ringing artifacts. To summarize, the idea behind this preconditioning approach is to maps the 2D captured image into a 3D layered representation that has the sharpest details on the layer that corresponds to the ground truth depth. Starting from the output of this step it's intuitive that the CNN will have an easier work to do, in fact it must find the layer with the sharpest details or the highest gradient and associate it to the corresponding depth.

For our work, starting from the idea provided by Ikoma [46], we sought which method was the most compatible with the characteristics of our pipeline. Compared to their work, in our case the Tikhonov regularization could not be effective as the total level of blurring introduced into the image, by the optical model, is considerably higher, this due to the different type of masks used to encode the PSF. To address this problem, several known methods of deblurring were tested in the experimental phase, trying to obtain an intermediate result that approaches the idea of structure defined above.

The best results are obtained by applying the already described Deep Wiener Deconvolution Network (DWDN) [12] where we modified the network to take out the desired results. More precisely, with respect to what Mel did in his work, for us it is no longer necessary to perform the crop operation with the corresponding depth mask after the deconvolution step. This because in our case is sufficient to get the 3D layer representation. Thus, we perform a separate deconvolution for every depth

layer with a different kernel provided by the depth dependent RPSF. In this case, every deconvolution should be able to reconstruct the details in the image which had previously been coded with the same kernel. In the end, the final output results as K images where, for each image, different details result sharper than the others, allowing the neural network to distinguish between different layers of depth. In Fig.4.2 we show some samples of results getting from our implementation, reported along with the corresponding depth layer ground truth.

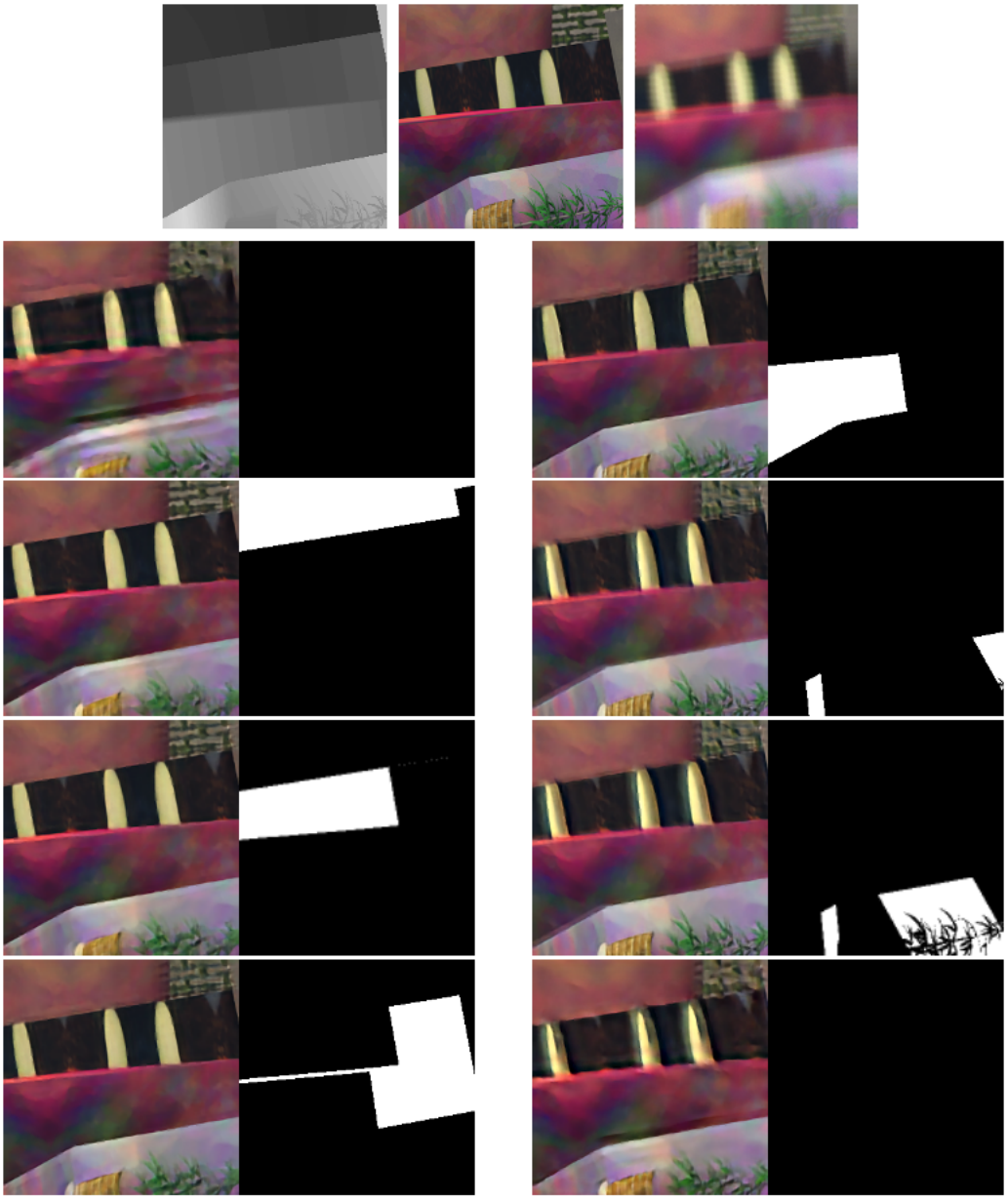

Figure 4.2: Results for modified DWDN network proposed in [12], in the first row from left to right is display the Ground Truth (GT) depth, the sharp all-in-focus image and the coded image obtained as output of the camera model. In the four columns below there are samples for the 3D layer representation with their corresponding GT layer.

From the figure we can point out as the predicted result has been achieved, in fact we can see how in between different layers different parts of the image are clear while the rest of the image present some undesirable artifacts.

4.2 Post-Processing Approach

As counterpart of the pre-processing methods, we found those of post-processing. In this case, we grouped all the methods that, performed some operations after obtaining the depth map from the algorithm, or over the output got from the neural network. The goal of this kind of approach is to improve the quality of the depth map making the estimation more consistent and accurate. One of the most common techniques is the spatial filters that are commonly used to reduce the noise in the final estimate, improving the correlation between adjacent pixels.

4.2.1 Displacement Field

For what concern the post-processing approach in our project we implement a refinement occlusion boundaries algorithm that was first proposed by Ramamonjisoa et al. [8]. This method can be applied to all depth estimation and the fact that is also fully differentiable, enabling the end-to-end training. Usually, recent methods to improve predicted depth maps, rely on various forms of filtering or they try to predict additive residual maps to refine the first estimate. In this case, Ramamonjisoa proposed to learn a 2D displacement field able to re-sample the pixels around the depth discontinuity region into sharper details. This kind of work sometimes can improve the estimate for the occlusion boundaries that are important cues to recognize object and may lead to discover new objects from the scene reconstruction.

Straining from the fact that occlusion boundaries correspond to regions where depth exhibits large and sharp variations, Ramamonjisoa showed that the resampling can be formalized as:

$$D(p) \leftarrow D(p + \delta p(p)) \quad \forall p \in \Omega \tag{4.10}$$

where D is the depth map predicted for a common MDE framework, p denotes an image location in the domain Ω, and $\delta p(p)$ represent the 2D displacement that depends on p. In short, in order to verify the assumption, the aim is to predict the optimal displacement. Taking a usual dataset that provide also the ground truth, the displacement field can be predicted training a CNNs.

$$\delta p^* = \underset{\delta p : p + \delta p \in \mathcal{N}(p)}{\operatorname{argmin}} (D(p) - \widehat{D}(p + \delta p))^2 \quad \forall p \in \Omega \tag{4.11}$$

To report an example, we can think about a 1D signal D with strong discontinuities that must be recovered as piecewise continuos functions. Starting by convolving the signal D with randomized Gaussian blurring we can obtain a smooth version $\widehat{D}$ and our training set $\mathcal{T}$ can be composed from the pairs $(d, \widehat{D})$. We can use this, training set to train a network function $f(.;\Theta_f)$ to predict the displacement field as:

$$\min_{\Theta_f} \sum_{(\widehat{D},D)\in\mathcal{T}} \sum_{p} L\left(D(p) - \widehat{D}\left(p + f(\widehat{D};\Theta_f)(p)\right)\right) \tag{4.12}$$

where $L(.)$ could be any loss functions. Once we known the optimal displacement fields we are able to re-sample the initial depth prediction to refine the depth maps.

As is present in [8], the common metrics to evaluate the monocular depth prediction. Here below will find the definition of those metrics that we also use to evaluate our results in Chapter 5.

$$RMSE = \sqrt{\frac{\sum(gt - pred)^2}{N}} \tag{4.13}$$

$$Rel = \frac{1}{N}\left(\frac{|gt - pred|}{gt}\right) \tag{4.14}$$

$$\log 10 = \frac{1}{N}\left(\sum(|\log_{10}(gt) - \log_{10}(pred)|\right) \tag{4.15}$$

$$RMSE_{log} = \sqrt{\frac{\sum(\log_{10}(gt) - \log_{10}(pred))^2}{N}} \tag{4.16}$$

$$AUT \quad \sigma_i = \max\left(\frac{gt}{pred}, \frac{pred}{gt}\right) \qquad with : \sigma_i < 1.25^i, i = 1, ..., 3 \tag{4.17}$$

where AUT stand for Accuracy Under Threshold and $gt, pred$ are the Ground Truth and the prediction of the depth map respectively.

4.3 Proposed Approach

The first stage consists of the design of the phase mask. To this end, exploiting the network described in Section 3.2.2 we are able to estimate the design parameters $[N, \epsilon]$ together with the weights of the CNN. The number of Fresnel zones treated as a hyperparameter is first set equal to 7. The network is trained on a subset of synthetic images with two different learning rates, one is specific for the optical layer and the other is suitable selected for the weights of the neural network. In this first stage the network is trained on two datasets, which we will present later in Chapter 5. For the FlyingThings3D we train the network for 150k iterations with batch size equal to 20 and Adam optimizer, the same number of iterations and the same settings parameters are also used for the second dataset, the NYUv2. In this

case a bilinear up-sampling is used to increase the resolution of the input and output images from a 240×320 to 480×640 this is done for evaluation purposes.

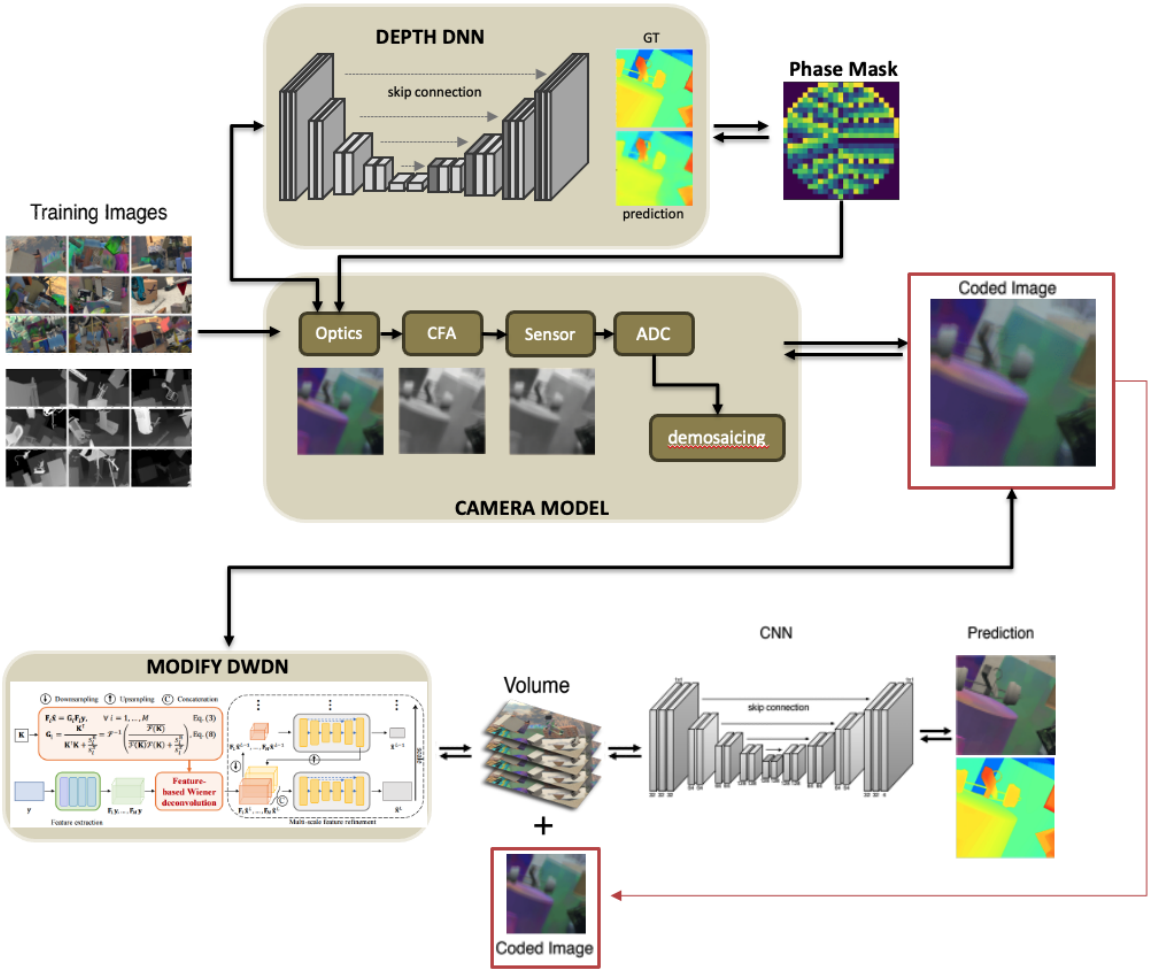

Figure 4.3: The full architecture for the proposed solution. As in the work of Mel [10] in the first stage the height map of the phase mask is jointly optimized with the weights of a U-Net trained on noise-free synthetic images for the monocular depth estimation task. In the second stage the phase mask is incorporated within the optical layer of the camera model to reproduce a realistic image formation model and to obtain the coded image. (In this stage the nonlinear image formation model is applied to take into account the depth cues for the occlusion boundaries). In the following stage a modified version of DWDN [12] is used to perform a preconditioning step that creates a 3D layered image in which there are sharp details in the corresponding ground truth depth layer. At final stage a CNNs is trained to get the final all-in-focus image along with the predicted depth map.

Once the phase mask has been obtained from the first stage and the noise-free images have been encoded with the non-linear image formation model described in Section 4.1.1, camera noise must be added to simulate a realistic image formation pipeline. From Fig. 4.3 we see as a Bayer Color Function Array (CFA) receives the blurred images from the optics module and outputs a downsampled Bayer color

pattern of RGGB form.

To simulate the sensor chip, read and shot noise described in Section 1.2.1 are added to the Bayer output and the resulting coded noisy image is quantized in range value between 0 and 255 by the ADC with 8-bit resolution. Last step for the simulation of the camera model is the demosaicing part where a high quality linear interpolation is applied to recover the full color image and to produce the final output.

The following step consists of training the modified version of DWDN adapted for the task of non-blind and non-uniform image deblurring. The training was done on a subset of the FlyingThings3D with 500 epochs, Adam optimizer and exponential decay rates of first $\beta = 0.9$ and second $\beta = 0.999$ momentum with learning rate Lr_{image} that starts from 1e-4 and it is halved after 250 epochs. As specified in [42] the numbers of auto-encoders in multiscale-refinement modules is equal to 2 with $M = 16$ filters applied at the first stage and the network is trained with a batch size equal to 8. For the loss function, it is experimentally shown that L1-loss (Eq.4.18) leads to better results compared to MSE loss.

$$\mathcal{L}_{image}(\theta, \theta^*) = \frac{1}{|T|} \sum_{\theta \in T} |\theta - \theta^*| \tag{4.18}$$

where θ ad θ^* are the reconstructed image and the ground truth sharp image.

The last step has the task of taking as input the 3D layered image produced by the DWDN, the coded image produced by the camera model and return in output a pair of images formed by the all-in focus image and the depth map. For this purpose, the definition of a U-Net such as those presented by Ronnerberger et al [37], is still useful. In the network we find a first convolutional layer that transform the input into a 32-channel feature map. The network implements skip-connection with four consecutive down samplings and up samplings. Each layer presents two consecutive convolution over the output features. All the convolutional layers are then followed by a batch normalization step with ReLU activation functions and the down/up sampling are performed with maxpooling layer 2x2 and bilateral interpolation. The other network parameters remain the same used in the first stage. Thus, 150k iterations with Adam optimizer and learning rate equal to 1e-4.

5 | Results and Discussions

In this chapter we are going to present all the results obtained in our studies. The aim of the chapter is to compare the performance of our algorithm with those that exist to solve monocular depth estimation problem, analyzing precision and accuracy metrics on a large test dataset. We will start first presenting which datasets were used and why this choice was made. Then in the results section we will show the achieved performance, with a comparison of the different types of phase mask and the different configurations network parameters. We will finish by showing the results related to the different methods for occlusion handling and how these affect the final result.

5.1 FlyingThings3D Dataset

The FlyingThings3D dataset is a popular dataset used in artificial vision for depth estimation and scene analysis. It was introduced by Mayer et al. [47] as part of the largest Scene Flow Datasets. It is a synthetic dataset for optical flow, disparity and scene flow estimation. It consist of a random object flying along randomized 3D trajectories. More than 25000 stereo frames with ground truth data are provided. The dataset has become a benchmark in the field of depth estimation and 3D scene understanding. It has been used extensively to evaluate depth estimation algorithms, compare different techniques and facilitate the development of new approaches for scene analysis. In our work a subset of this dataset is used, by following what Wu do in [11]. To accurately generate a final depth map of 256 x 256 pixels, working with RPSF of size 23 x 23, we need to get the all-in-focus images of size 278 x 278. We are able to obtain the final size by cropping the original image whose resolution was 960 x 540. At the end the subset was divided in training, validation, and test sets contain respectively 5078, 555, and 420 images. Fig. 5.1 shows some examples of the final resolution samples.

Figure 5.1: FlyingThings3D samples of resolution 278 x 278.

5.2 NYUV2 Dataset

Willing to compare the proposed approach with the competitor and SoTA for the monocular depth estimation, the entire project is also trained on the NYUv2 depth dataset [15]. The ground truth depth maps are acquired with the Microsoft Kinect V1 with a maximum depth value of 10 meters. Unfortunately, the Kinect V1 produces invalid pixels around the edge of objects especially which is located in the background, this is due to the structured light methos used in this kind of sensor.

To overcome this drawback, in the evaluation, the invalid pixels are not taken into consideration by applying the mask that is also provided within the dataset. In those experiments where the NYUV2 is used, the depth plane that we consider is 10 and the ground truth depth map are rounded-up into the range values of [0,10]. In this case the input RGB image have size 240 x 320 that is half of the original resolution. The same crop used by Eigen's work [48] of shape (45, 471, 41, 601) is applied to avoid invalid border regions in the ground truth and the predicted depth maps. At the end of the pipeline the outputs are up-sampled with a bilinear interpolation to reach the original size 480 x 640. With this dataset we are not dealing with noise free image samples, so the input data is directly convolved with the RPSF obtained from the phase masks.

For training and testing of the displacement field approach described in Section 4.2.1 we use the provided dataset NYUv2-OC++ [8] that contains manual annotation for the occlusion boundaries, made by the authors on top the NYUv2 Depth dataset for all the 654 test images. Refer to Fig. 1.10 to have some visual examples.

5.3 Results

Once we get an overview of which datasets we used in the project, and developing the methods and applications presented in the previous chapters, we will show in the remaining part what results have been obtained for each method. Please note that the results, related to the displacement field part, have been obtained with the simple implementation of the method as it is described and presented in [8], while

to the other sections, some changes were necessary to allow them to be integrated into our proposed pipeline in Fig. 4.3.

5.3.1 Phase Mask Design

Taking into consideration the first stage of the proposed pipeline, a number of experiments were carried out as an initial ablation study to evaluate different configurations for the pupil functions. This was deemed necessary as different approximations of the same formula (Eq. 1.13) were found in the literature, and morover, we wanted to test the optical model adopted in the simulation of the RPSF prototype. Among the various experiments, we included a comparison with the implementation of phase mask based on superposition of GL modes. In experiments of the ablation study, the network architecture and the training settings remain the same. The quantitative and qualitative results are reported in Fig. 5.2 and Tab. 5.1, in the table the second column represents the phase parameter K that is specified in equation (5.1).

$$P(x,y) = e^{j\left[\left(\frac{2\pi}{\lambda} * \delta_n * H\right) + \left((x^2+y^2) * K * \psi\right)\right]} \tag{5.1}$$

The term K is used to control the phase in the pupil function, and variation on this term leads to variations in the amount of rotation in the resulting PSF.

Experiments	**Mode**	**Parameter K**	**RMSE ↓**
1	FR	$\frac{\pi}{\lambda}$	1.907
2	FR	$\frac{\pi * \lambda_g}{\lambda}$	1.389
3	FR	$\frac{\lambda_g}{\lambda}$	**0.801**
4	GL	$\frac{\pi * \lambda_g}{\lambda}$	1.312

Table 5.1: Quantitative results for the four conducted experiments on the subset of FlyingThings3D. FR means Fresnel Zone instead GL is the Gauss Laguerre mode.

These experiments were carried out using the subset of the FlyingThings3D dataset, in all cases the input image has size $278 \times 278 \times 3$ where the third dimension was related to the color channels. The total number of depth layers used for the representation of the RPSF is set to 21, as we can infer from Fig. 5.2. Consequently, the RPSF has a shape of $21 \times 23 \times 23 \times 3$, where 23×23 is also the dimension of the phase mask. The convolution operation between the input all-in-focus image and the learned RPSF return an output image of size $256 \times 256 \times 3$ as described in Chapter 4.

As evaluation metrics for the performance of the phase mask we use the RMSE as defined in Eq.(4.13), and computed over the test set. Some examples of results for the conducted experiments are shown in Fig. 5.3, where we reported the predicted depth maps and the corresponding ground truth.

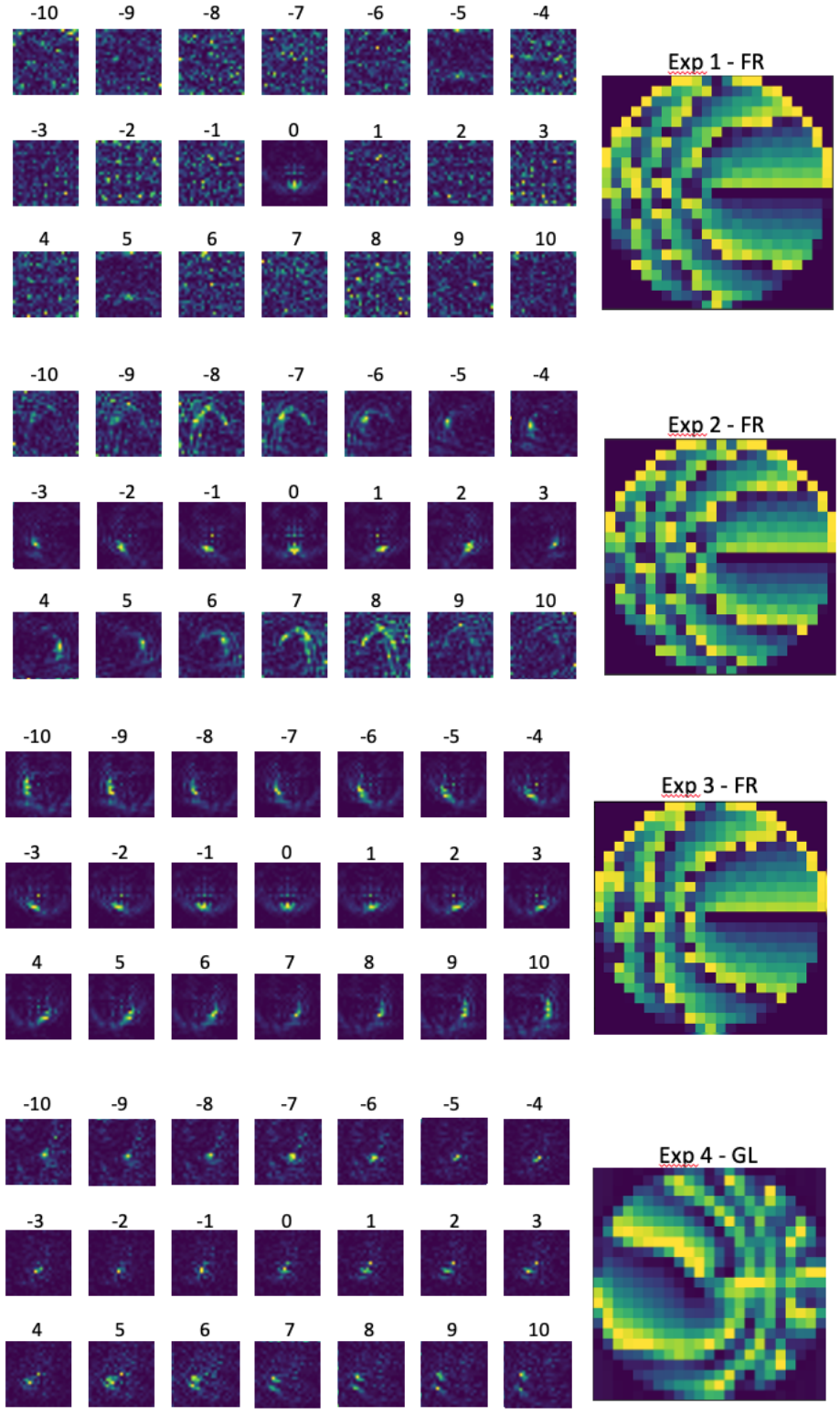

Figure 5.2: The generated RPSFs for the experiments and its height maps. Notice that the GL modes phase mask is not differentiable so it must be considered as a fix phase mask.

Looking at Figure 5.2 we can see that all four phase masks have almost a similar shape, and in particular that they are very similar to the one computed as

superposition of GL modes, which we mentioned in Section 3.3. This mask is not differentiable and therefore in the end-to-end learning within the CNNs part, relating to the design of the mask, remains fixed and only the weights of the neural network are optimized. These results lead us to think that the implemented optical model in [10] is indeed a correct implementation of the studied optical model, as by exploiting different technologies, these converge in the same direction, obtaining very similar results. In particular, experiment 3 is the same implementation as Mel presents in his work and therefore will be our starting point for start dealing with occlusion problems.

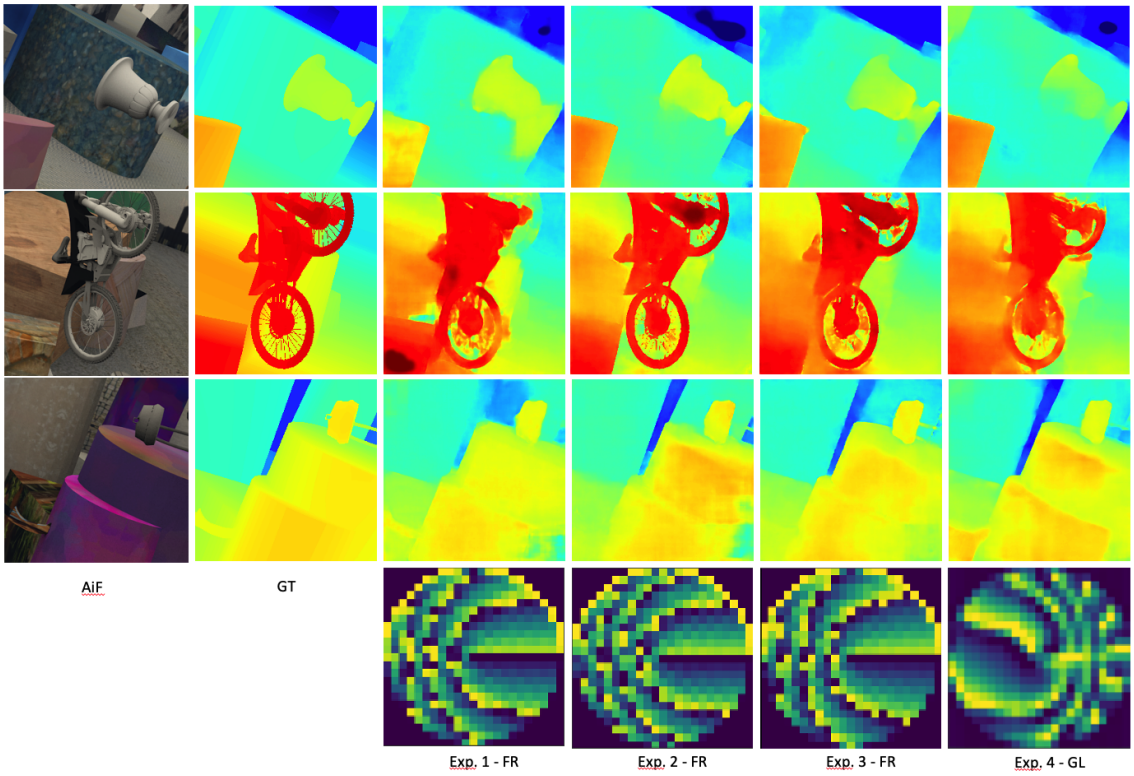

Figure 5.3: Sample of depth map prediction for the four ablation studies with the FlyingThings3D.

Referring to Fig. 5.3 and to the RMSE's column in Tab.1 we get that the best results are carried by experiment 3. Exp. 1 is not able to distinguish the object in the background, and this means that the RPSF produced with that pupil function is too blurry in those levels and thus the NN cannot be able to reach a good RMSE value. The results of exp. 2/3/4 are quite similar. Exp. 2 leads to a better reconstruction for the edge of the object but completely misclassified the planar region (that composed the large part of the images). Experiment 3 is a good trade-off between classification of planar region and sharpness around the edges. In Exp. 4 we are more interested in the RPSF produced and we used that result as a comparison for the phase mask based on Fresnel zones.

5.3.2 Displacement Field

The first method for handling occlusions in which we analyze the results is the one presented by Ramamonjisoa in [8] and which we have discussed in previous chapters. In Section 4.2.1 we present the post-processing approach based on the displacement field. Through the prediction of this displacement the module should be able to re-sample the occlusion boundaries returning a depth map with sharper edges at depth discontinuity.

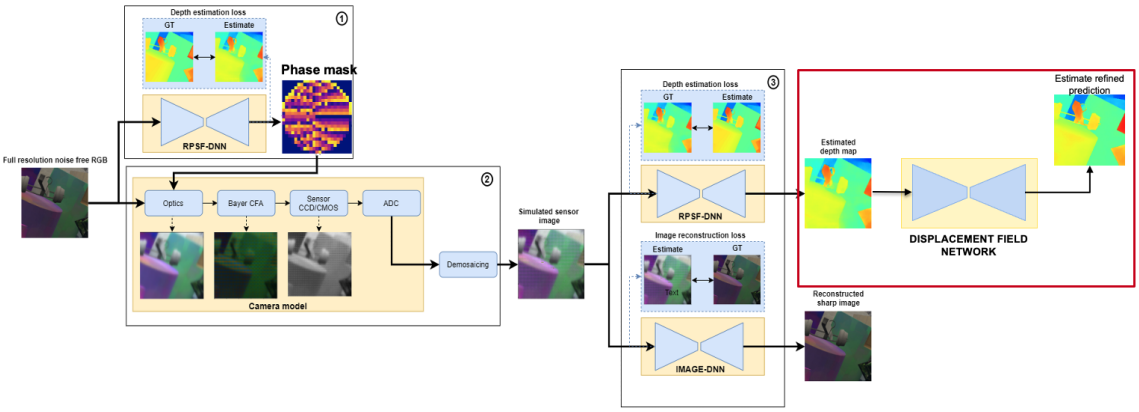

Figure 5.4: Initial pipeline proposed by Mel in [10] within the Displacement Field module (red rectangle).

In the text was described as this fully differentiable method can be applied to the output of any depth estimation method. In Figure 5.4 we present the proposed pipeline where we integrate the module at the end of the initial pipeline proposed by Mel. To summarize the steps, the phase mask's trainable parameters are initialized to $[N = 1, \epsilon = 0.1]$, and the number of Fresnel zones is set equal to 7. Using the FlyingThings3D dataset the network is first trained over noise-free images to learn the phase mask height map and the corresponding RPSF. The final configuration of the design parameter is reached for $N = 1$ and $\epsilon = 0.97$. (To visualize the height map and the RPSF refer to Figure 5.2 exp.3). Once we obtained the RPSF we have to apply the camera model in other to add the noise effect in the captured image and in the third stage we fine-tune the network weights to the task of monocular depth estimation. Finally, we get the predicted depth map and, we feed the displacement field module with our initial prediction and the network returns the refined version of it. Tab. 5.2 and Fig. 5.5 are used to report quantitative and qualitative results for this proposed solution. Notice that the pure implementation of the approach with the pre-trained weights of the network does not improve the results, from the figure, looking the red circles, we can see how the approach works. In these regions, the edges are clearly sharper than the initial prediction but the method is not able to recover a fair estimate with respect to GT and so it misclassified the large part of the

pixels level. In this example, we can see the limit of the proposed solution. Firstly, as the network is pre-trained on a special dataset where manual annotations have been added to highlight occlusion boundaries, the network is not able to generalize over a larger set of data. Secondly, the fact that it acts as an estimation refiner means that it never finds new edges because if the initial estimate does not present any ambiguity, the method does not take any action.

Model	MAE ↓	RMSE ↓	Log_{10} ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Haim et al. [35]	0.297	0.635	0.109	0.803	0.879	0.923
Wu et al. [11]	0.207	0.521	0.090	0.865	0.918	0.945
Chan et al. [36]	0.205	0.490	0.077	0.888	0.945	0.968
Ikoma et al. [14]	0.089	0.191	**0.034**	**0.941**	0.970	0.981
Exp. 3	**0.036**	**0.054**	**0.034**	0.905	**0.975**	**0.992**
Exp. 3 + DF	**0.036**	0.055	0.035	0.907	**0.975**	**0.992**

Table 5.2: Quantitative comparison over different approach that perform monocular depth estimation, +DF means that Displacement Field module is present (↓: Lower is better; ↑: Higher is better). First rows of the table is taken from Ikoma in [14].

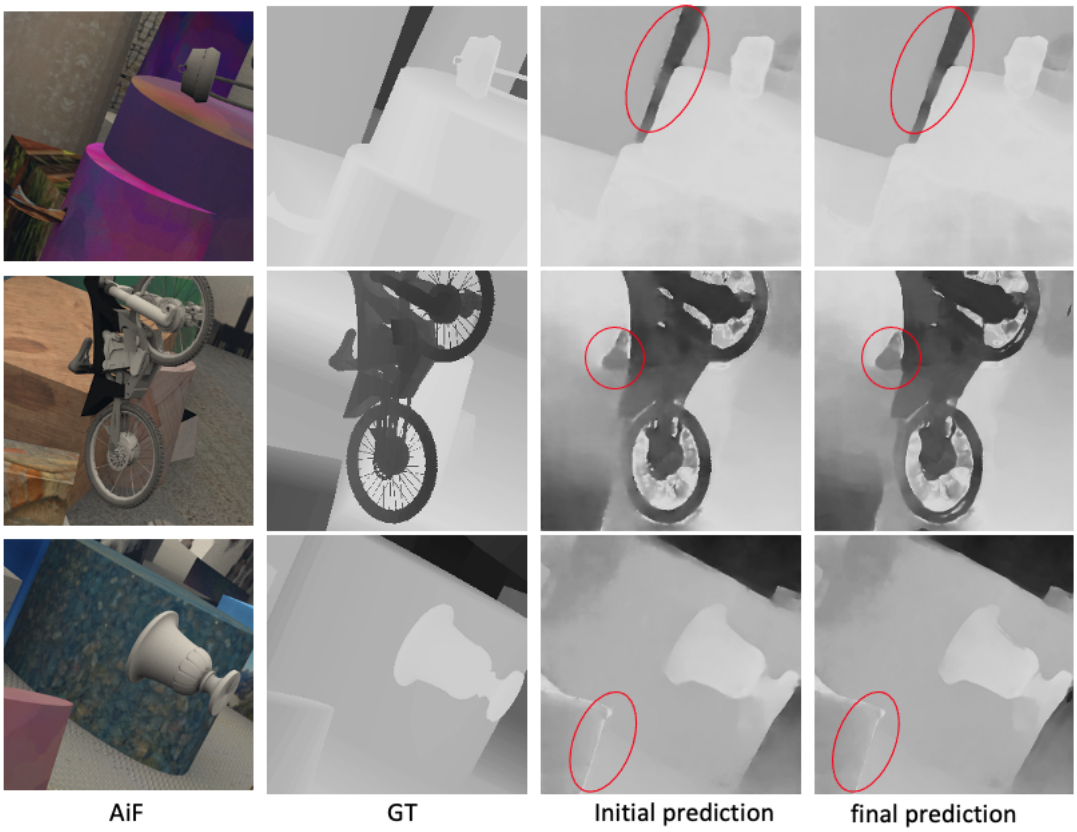

Figure 5.5: Qualitative simulation results for Displacement Field module on FlyingThings3D dataset, the red circles are used to enphasize the key point to observe.

To evaluate this method over a more coherent dataset we train the entire pipeline over the NYUV2 dataset. For this dataset, even if we used the same initialization, the learned RPSF converges into a different configuration. As we know from Section 5.2 this time, we used just 10 depth planes. Fig 5.6 shows the learned phase mask and the corresponding RPSF for this case.

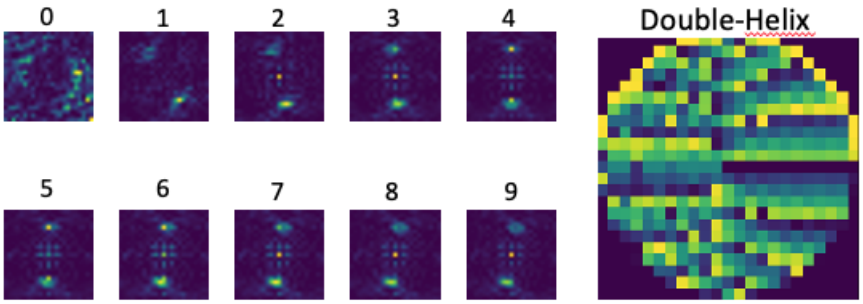

Figure 5.6: Learned RPSF as function of defocus for NYUv2 depth dataset.

The final design parameters are $N = 2$ and $\epsilon = 0.99$, thus resulting in a RPSF with a double-helix configuration, so two main lobes that rotate with defocus and the high value for the high confinement parameter produce the spread out of peaks.

Fig. 5.7 shows some visual results for the Displacement field approach over the NYUv2 Depth dataset. Notice that the pre-trained network of the Displacement Field module is trained over the NYUv2-OC++, which is a subset of the main dataset where manual annotations for the occlusion boundaries has been added. In the figure, the first two columns refer to the sharp images and the ground truth depth maps. Moreover, this time the images already present the noise introduced by the camera model and the Kinect V1 sensor produces invalid pixels at depth discontinuity. We put some markers to visualize the saddened points. In the first row, the bed's image, the module recognized the texture of the sheet as edges, and it tried to highlight this region, resulting in a worst depth map. In the third row, the approach is able to improve the edges of the pillows above the sofa. In the initial prediction, those edges are very confusing, in the end they appear clearer and sharper. However, the overall image quality does not seem real, and in fact it presents a sort of cartoon effect.

From the implementation of this approach, we realized how necessary it is to have a good initial raw estimate. So, in those cases, the post-processing methods can improve the quality and increase the accuracy. Future works based on this approach could focus on trying to join the two modules in an end-to-end training. In this way, the network could learn step-by-step how to reproduce clearer edges and this could help also in finding new edges.

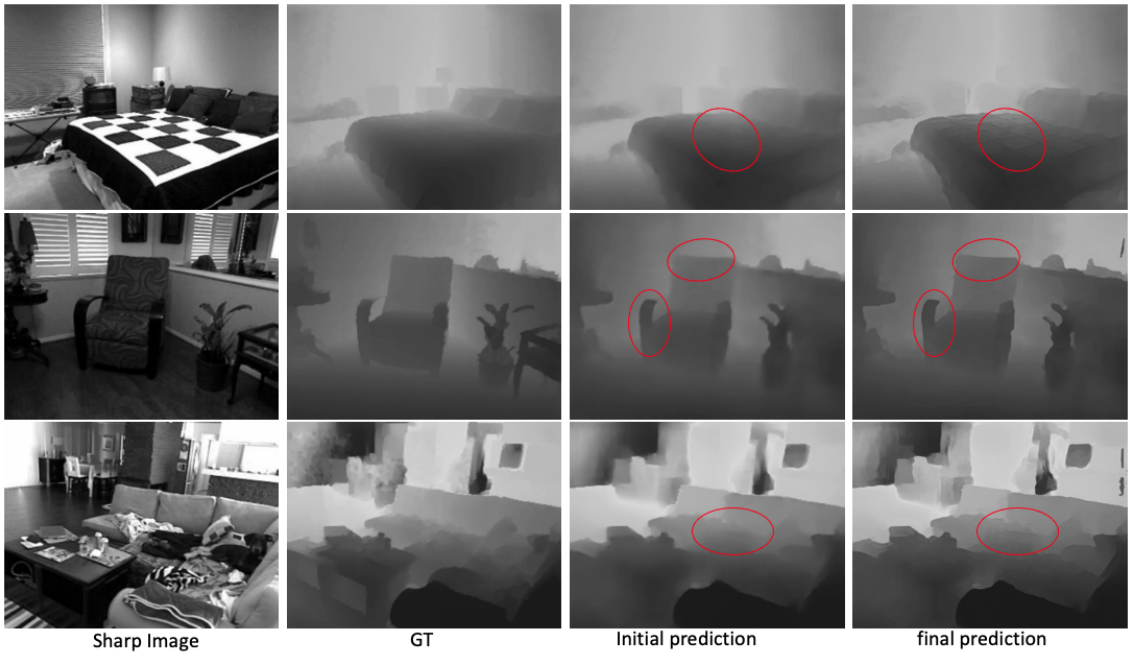

Figure 5.7: Qualitative simulation results for Displacement Field module on NYUv2 Depth dataset. The red circles are used to enphasize the key point to observe.

5.3.3 Non-Linear Image Formation Model

A large part of our project involved the implementation of the new image formation model into our pipeline. In Chapter 4 we pointed out how a linear approach, to combine the different depth layers obtained from the convolution of the plain image with the depth dependent RPSF obtained from the phase mask, did not lead to a clear definition of the occlusion boundaries, introducing ambiguity in the surrounding regions. This ambiguity does not allow the neural network to reconstruct in detail the object edges, especially in the presence of depth discontinuities, resulting in an inaccurate depth map with evident problems in revealing the occlusion boundaries.

For this purpose, the new non-linear image formation model exploiting the alpha compositing technique allows to define a more natural transition between the depth discontinuities, resulting in many cases as a good approach to obtain better clarification in the definition of occlusion boundaries. In Figure 5.8 is reported a comparison between the implementation of the two optical models (columns (b) and (c)) with the plot of their difference in the last column, we can actually visualize where the improvements are made. Comparing the last three columns it is evident that the non-linear case exhibits a clear transition between depth levels (look at the background of the images), and also we can distinguish new details as it is for the

object over the cylinder on the right, where its contour is detailed.

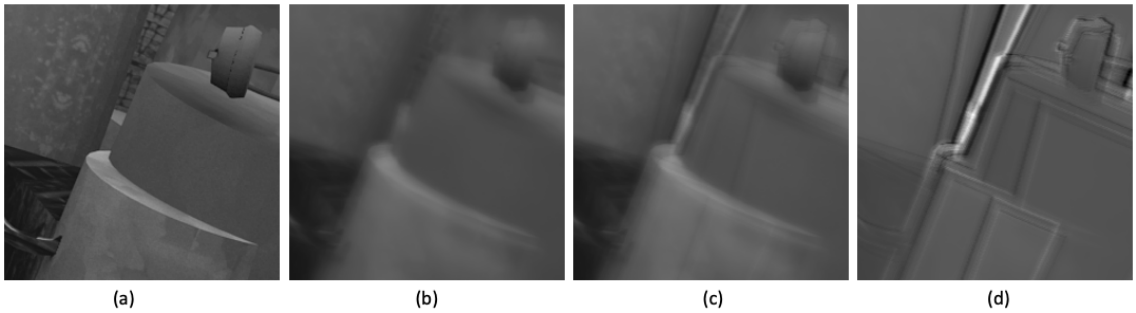

Figure 5.8: Comparison between the Linear Image Formation Model (b) and the non-linear(c). (a) is the sharp all-in-focus image that provide the scene under examination, and (d) is the plot of the differences.

Figure 5.9 shows the results of the two different image formation models following what is described in Section 4.1.1. The image formation model is incorporated in the end-to-end learning and acts in the way the image is constructed by exploiting the convolution with the depth dependent RPSF. The second and third columns, as in Figure 5.8, show the noisy RPSF-coded image with the corresponding image formation model. In the non-linear case it is possible to note how the alpha compositing reduces the ambiguity around the depth discontinuity. Looking at the predicted depth maps in columns (d) and (e), with respect to ground truth (f), the improvement is clear. Some edges such as the blue triangle on the second row, the seat in the third, or the stripes in the gym weight on the fifth row can now be clearly distinguished. Or as in the first and last row where the classification level of the planar region is more similar to the ground truth. To support those visual results, we report also the numerical result in Tab. 5.5 in the next session where we summarize all the results obtained for this subset of the FlyingThings3D dataset concerning the other competitors.

The same analysis can be done also over the NYUv2 dataset, and the quantitative and qualitative results are reported in Figure 5.10 and Table 5.3. Looking at the figure where we compare what Mel et al. [10] gets with his linear approach and our results. Both approaches get the same phase mask starting the end-to-end learning with design parameter set to $[N = 1, \epsilon = 0.1]$ and reach the equilibrium point to a double helix configuration and spread out peaks with $[N = 2, \epsilon = 0.99]$. The learned phase mask is the one reported in Figure 5.6. Upon visual inspection on the last two columns, it can be easily seen that our new implementation produces more accurate and realistic depth maps respect to the ground truth. Due to the inconsistency of model the defocus blur at occlusion boundaries, Mel works tend to predict blur transition and equalize some small regions where there will be holes or hollows. As in row three between the chairs or in row five between the bed and the wardrobe.

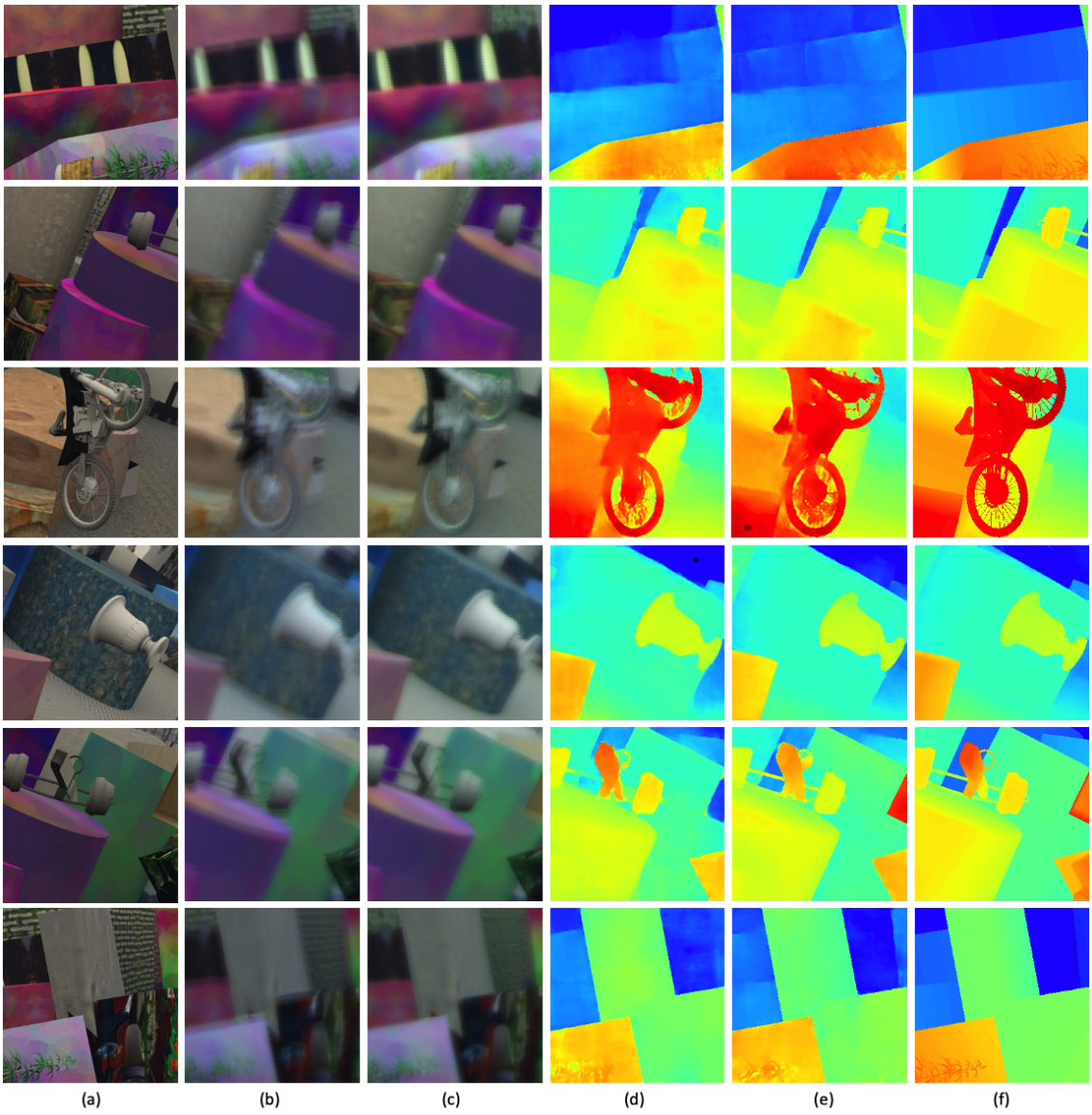

Figure 5.9: Qualitative comparison results on image formation model from the test set of FlyingThings3D. (a) Sharp all-in-focus image. (b) Coded image with Linear image formation model. (c) Coded image with Non-Linear image formation mode. (d) Estimate depth map from b, (e) Estimate depth map from c. (f) Ground truth depth map from the dataset.

However, one drawback of the proposed method, compared to other works in the literature as Adabins [49], the network still fails to learn sharp boundaries due to the blurring effects introduced by the RPSF within the input coded image, although the overall result is greatly improved. In another aspect, the proposed model is suitable for real-time applications without compromising depth estimation accuracy.

Tab. 5.3 reports a quantitative comparison with the state-of-the-art that used an approach based on coded aperture PPE. Our approach together with Adabins et al. [49] are the only that use exactly the same subset of 50k training images as specified by Eigen in [15], all the other competitors used the complete dataset that

contains 120k training images. Moreover, our network is based on a simple U-Net with at most 7 millions of trainable parameters. In contrast, other approaches used a more complex network that increase the training time and slower the inference. Even if our approach is lighter and requires fewer training samples, it outperforms the more computationally demanding and advanced approaches. This is mainly due to the way we encode the coded image by exploiting the depth dependent RPSF, which makes it easier for the network to predict the final depth maps.

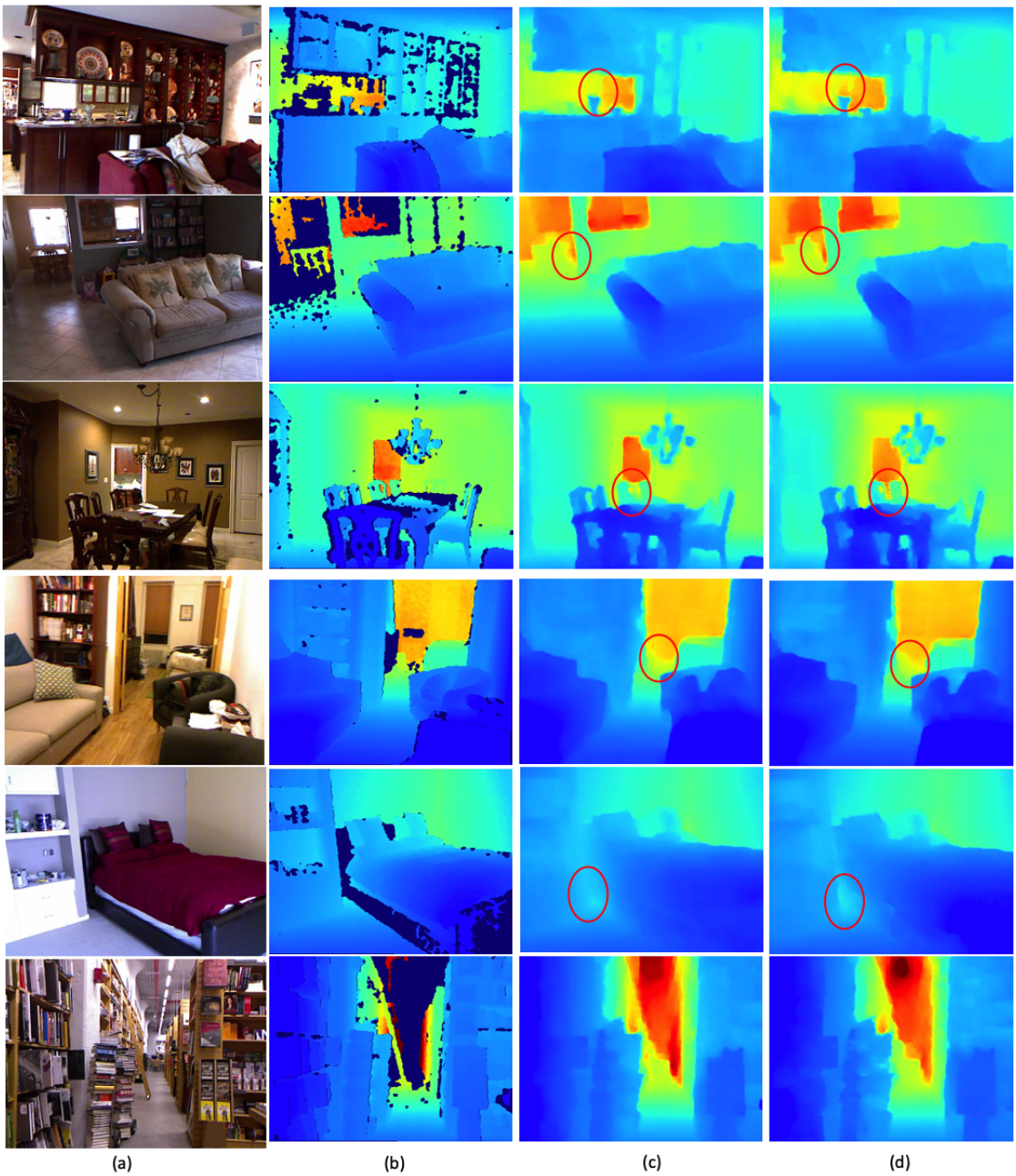

Figure 5.10: Qualitative comparison results on image formation model from the NYUv2 Depth dataset. (a) Sharp all-in-focus image. (b) Ground truth depth map, in dark blue the invalid pixels. (c) Predicted depth map with Linear approach. (d) Results with non-linear image formation model.

Model	RMSE ↓	Rel ↓	Log_{10} ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Eigen et al. [15]	0.641	0.158	-	0.769	0.950	0.988
Hao et al. [50]	0.555	0.127	0.053	0.841	0.966	0.991
Qi et al. [51]	0.569	0.128	0.057	0.834	0.960	0.990
Alhashim et al. [52]	0.382	0.093	0.050	0.932	0.989	0.997
AdaBins et al. [49]	0.364	0.103	0.044	0.903	0.984	0.997
Mel et al. [10]	**0.274**	**0.075**	**0.029**	0.950	0.986	**0.997**
Proposed	0.275	0.076	**0.029**	**0.952**	**0.989**	**0.997**

Table 5.3: Quantitative comparison with all competing methods for monocular depth estimation on NYUV2 Eigen test set [15].

5.3.4 Approximate Inverse Step

The third step in the pipeline proposed by Mel in [10] consists of the training of two neural networks, one to recover the predicted depth maps by finetuning the pre-trained network for the task of monocular depth estimation. Then, to recover the sharp all-in-focus image starting from the coded blurred image a second network that uses a non-blind and non-uniform image deblurring module is trained on the subset of the FlyingThings3D dataset. As shown in Figure 5.11, our approach exploits the pre-conditioning step based on Approximate Inverse of image formation model to simplify the approach and get in output the pairs of images made of the depth maps and the all-in-focus image.

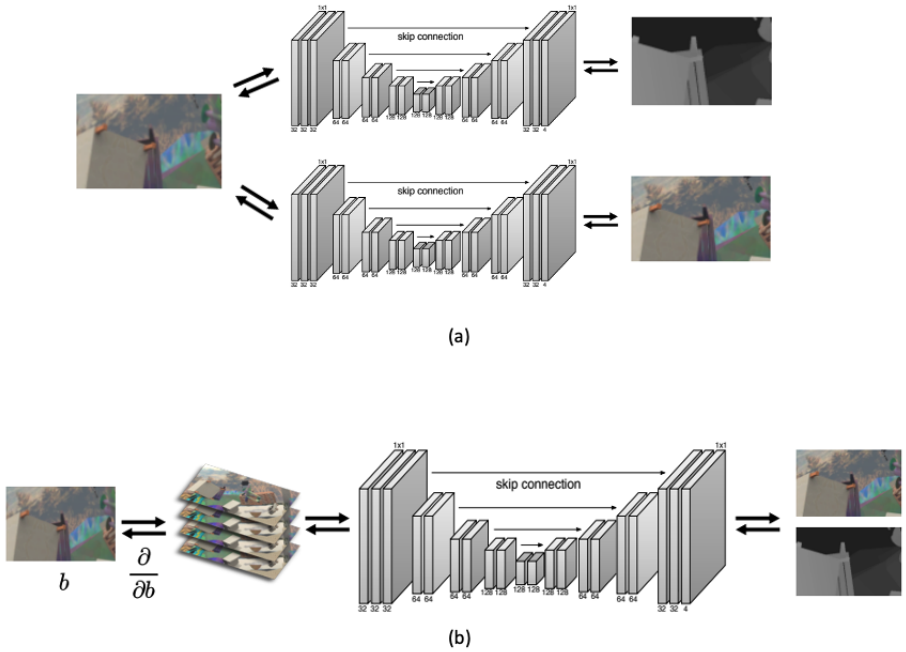

Figure 5.11: Third stage's pipeline for the starting approach (a) and to the final proposed pipeline (b).

The idea behind this pre-conditioning step is trying to find a deblurring method that starting from the 2D captured image is able to return a 3D layered representation in which there are sharps details in the corresponding ground truth layer. In this way, the simple neural network based on the U-Net model [37] should be able to easily obtain an accurate prediction for the depth maps and reconstruct simultaneously the sharp all-in-focus image. To support this, in Tab. 5.4 we report the results of an ablation study. Using the same U-Net proposed in the previous steps, and with a specific input (refer as "perfect input") defined as described above, it is possible to improve the previous result. An example of “perfect input” is present in Fig. 5.12.

Model	RMSE ↓	Rel ↓	Log_{10} ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Mel et al. [10]	0.274	0.075	0.029	0.950	0.986	0.997
Perfect Input	0.180	0.039	0.015	0.996	0.999	0.999

Table 5.4: Quantitative result for the ablation study with the perfect input.

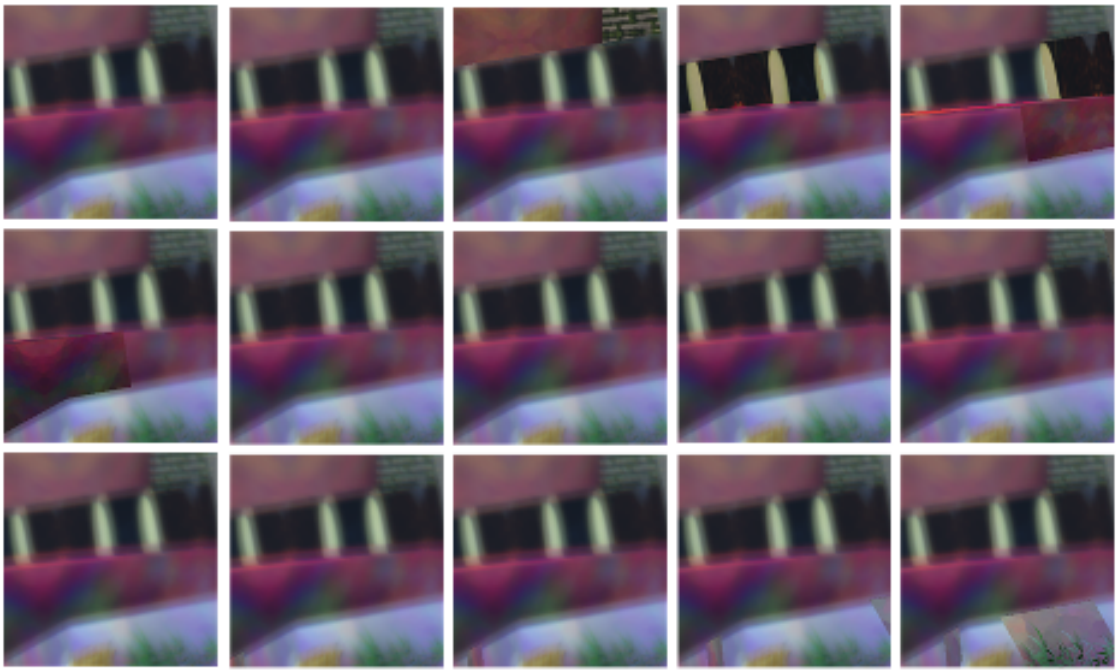

Figure 5.12: Example of "perfect input". Notice how moving through the image, that represent the depth layer, different details are in focus, while the rest of the images remains blurry.

Once verified the ablation study that with the "perfect input" this idea could achieve the desired results. We modified the DWDN [42] in order to obtain a method that can perform partial deblurring in specific region of the image. The modified version of the network that was earlier used as a non-blind and non-uniform image

deblurring model becomes very useful to this purpose, as our network is now able to perform a partial deblurring of the image. Remember that our image formation model, present in the second the stage of the pipeline (Fig. 4.3), encodes different parts of the image using different kernels, which are provided by the depth dependent RPSF obtained with the design of the phase mask.

Fig. 5.13 shows the results of pre-conditioning step for one sample. Notice as moving between layers, different part of the image results sharp, while the rest presents some artifacts or shadows.

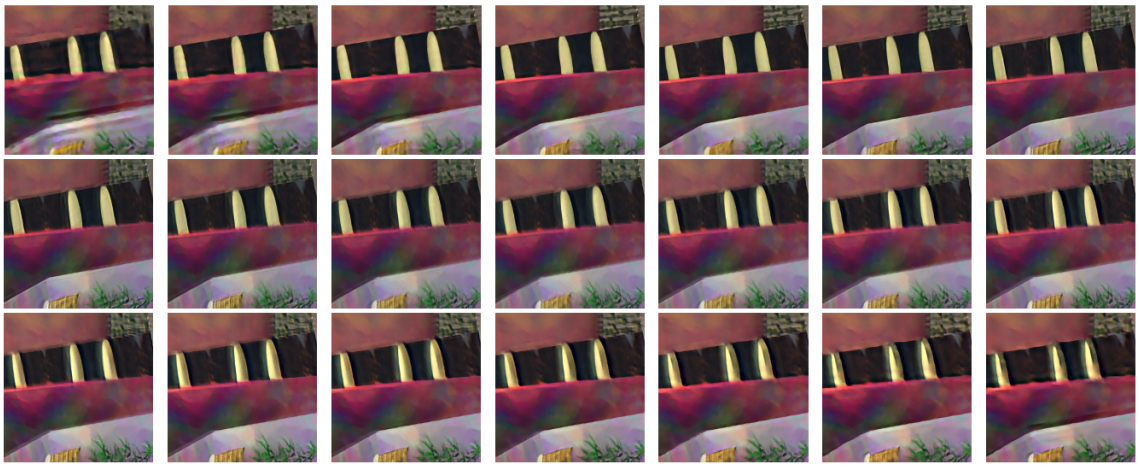

Figure 5.13: Results for one sample of the pre-conditioning step.

At the end, we finally summarize the obtained results in Table 5.5 and Figure 5.14. Looking at the table the best result is obtained with the simple Non-Linear Image Formation Model, where we are able to overperform all the other methods for the task of monocular depth estimation with PPE camera. In the table are present also the results from the competitor approaches presented in Chapter 2. Our method is able to get the best value for the accuracy metrics that, as explained in Section 4.2.1, are those related to the estimate for the occlusion boundaries.

Model	MAE ↓	RMSE ↓	Log_{10} ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Haim et al. [35]	0.297	0.635	0.109	0.803	0.879	0.923
Wu et al. [11]	0.207	0.521	0.090	0.865	0.918	0.945
Chan et al. [36]	0.205	0.490	0.077	0.888	0.945	0.968
Ikoma et al. [14]	0.089	0.191	0.034	0.941	0.970	0.981
Mel et al. [10]	0.044	0.087	0.034	0.956	0.990	0.996
Non-Lin. IFM	**0.026**	0.072	-	**0.959**	**0.992**	**0.997**
Non-Lin. IFM + Pinv	0.063	0.117	-	0.932	0.984	0.994

Table 5.5: Quantitative comparison over different approaches that perform monocular depth estimation, (↓: Lower is better; ↑: Higher is better). Firsts rows of the table is taken from Ikoma in [14].

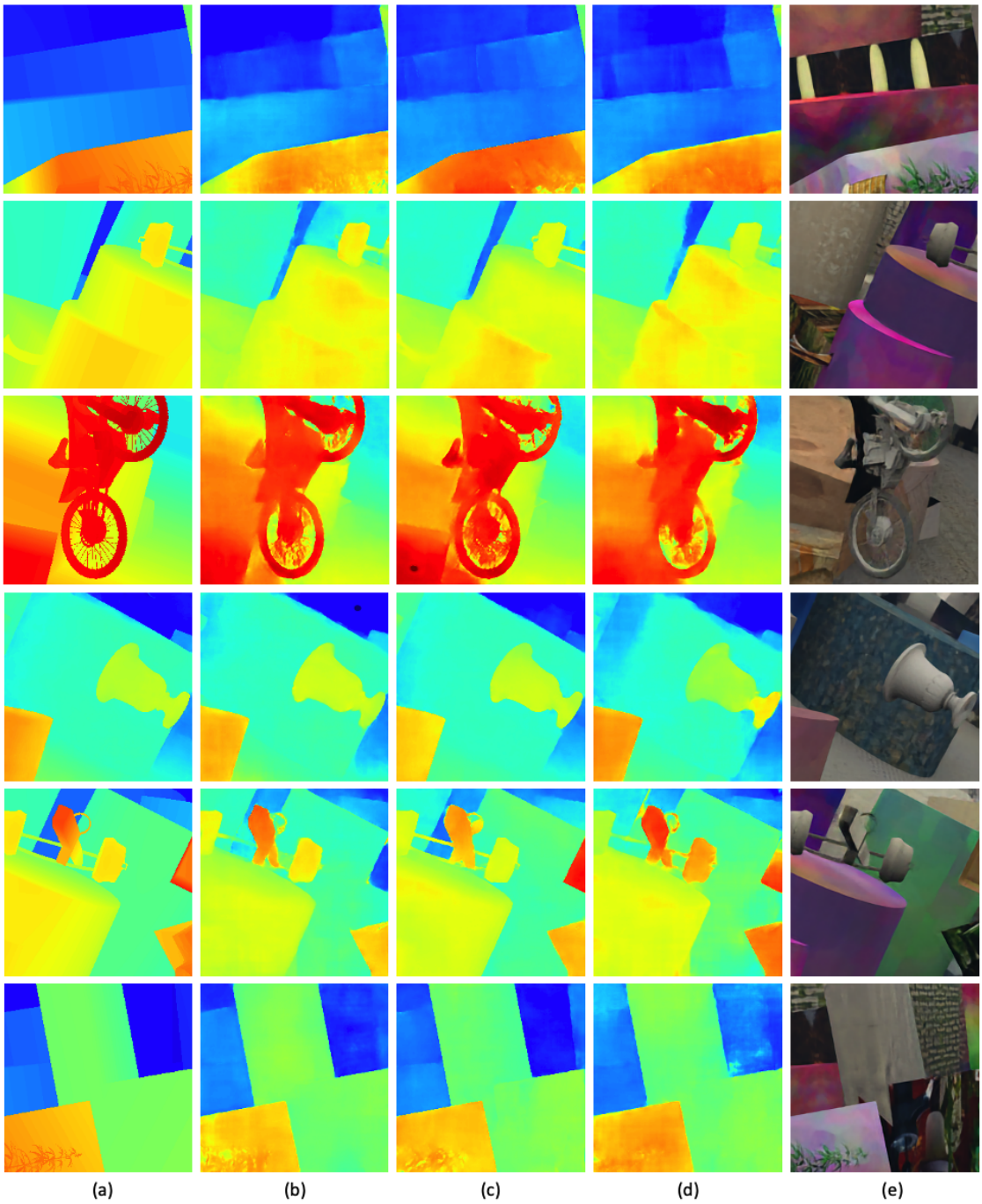

Figure 5.14: Qualitative comparison for the FlyingThings3D dataset. (a) Ground Truth depth image. (b) Noisy estimate disparity from Mel [10]. (c) Proposed Non-Linear IFM. (d) Proposed Non-Linear IFM with Approximate inverse step. (e) Recover sharp all-in-focus image.

Instead, for qualitative results we refer to Figure 5.14. Here we report the comparison in terms of predicted depth map between our two proposals and the work of Mel [10], which we consider as the baseline for the Thesis. Mel struggles to predict good occlusion boundaries especially when they are located in the extremes in terms of depth. This could be a drawback that comes from the shape of the RPSF

which becomes blurry for high level of defocus. Sometimes, the network produces erroneous depth predictions for objects in the scene, as could be in the second row of column (b). On the other hand, our proposed approach based on the Non-Linear image formation model respects the quantitively results shown in Tab. 5.5 and returns an overall result which best reflect the ground truth (look at the second and fifth rows).

Column (d) represents the results for the approximate inverse of the image formation model. In this case, the results do not meet expectations. Looking at the overall result looks worse, but if we focus on certain points in the images, we see that these are considerably better than all the other approaches as in the top right corner in the second row or for the couple of objects on the right in the fifth row. In our idea, and also looking at the results in Figure 5.14, our preconditioning step is not able to clearly differentiate two layers encoded with a very similar kernel. The DWDN network is able to recover the details not only from images encoded with the corresponding kernel but also with kernels that are very similar to it. In order to address this issue, a possible future work would exploit what we discovered in the ablation study for the phase mask design and create a new phase mask that better satisfy the requirements for the DWDN, getting a RPSF that differs more between consecutives layers.

For what concern the image deblurring, last column of Figure 5.14 represents the results obtained from the Approximate inverse approach. As we expected the network is able to reconstruct in a perfect manner all the details of the image, making it indistinguishable from the sharp all-in-focus image.

6 | Conclusions and Future Works

6.1 Conclusions

This thesis is presented as a continuation of an earlier work where a comprehensive simulation of a novel computational camera model was presented. In the first part we tried to verify the correct operation of the optical model, proposing a comparison between phase masks of different concepts. We show as in both cases the PPE camera is able to generate a Rotational-PSF where we encode the information about the depth dependency as function of defocus. In the second part of the thesis, we try to implement alternative methods to handle the occlusion boundaries, which result less clear in this as in other competitive methods. A new non-linear image formation model has been included in the pipeline that can better encode depth discontinuity within the image, thus improving previous methods in terms of RMSE and other metrics. In addition, we present an idea for the implementation of a pre-conditioning step based on approximation inverse functions. It should facilitate the work of the final network and improve its performance. The final estimate for the depth maps seems improving in some areas but the quantitative result does not support this statement.

6.2 Future Work

Starting from this project some potential future work can address several areas. One could focus on finding a better PSF shape. In Chapter 5 we saw how small differences in PSF shape can drastically affect the final result and it could b e interesting to investigate what is the best trade-off between having an accurate final depth estimate and recovering the image in high quality. A second future project could focus on the preconditioning step. In the simulations we verified how getting a good result from the preconditioning step can lead to much better results. I would recommend focusing on alternative methods with respect to DWDN to perform partial deblurring or a more careful implementation of loss functions. For examples, a loss function that penalizes the reconstruction of the details in the undesired

areas. In order to evaluate the performance of the proposed approach in other outdoor environments, one could consider training the model on the examples in the KITTI dataset [53], to enlarge the results obtained with the indoor benchmarks. In this way, it would be also possible to confirm the effectiveness of the approach in environments with different characteristics as lighting conditions, terrain variations and the presence of moving objects such as vehicles and pedestrians. Finally, a work that is a natural consequence of this would be to produce a fully working camera prototype and through the capture and process of images try to optimize the project parameters for the task of monocular depth estimation.

Bibliography

[1] Y. Y. S. Greengard, Adam and R. Piestun.

[2] S. Prasad, "Rotating point spread function via pupil-phase engineering."

[3] S. R. P. Pavani and R. Piestun., "High-efficiency rotating point spread functions."

[4] K. Rakesh, "Three-Dimensional Imaging using a Novel Rotating Point Spread Function Imager."

[5] Y. Y. Schechner, J. Shamir, and R. Piestun., "Propagation-invariant wave fields with finite energy."

[6] V. L. Ramamonjisoa Michael, "Sharpnet: Fast and accurate recovery of occluding contours in monocular depth estimation." 2019.

[7] Y. a. J.Jiao, Y.Cao, "LookDeeperinto Depth: Monocular Depth Estimation with Semantic Booster and Attention-Driven Loss." 2018.

[8] V. L. Ramamonjisoa Michael, Yuming Du, "Predicting Sharp and Accurate Occlusion Boundaries in Monocular Depth Estimation Using Displacement Fields," 2020.

[9] R. Kumar and S. Prasad, "PSF rotation with changing defocus and applications to 3D imaging for space situational awareness."

[10] M. S. Mazen Mel and P. Zanuttigh, "End-to-end Learning for Joint Depth and Image Reconstruction from Diffracted Rotation." 2022. [Online]. Available: arXiv:2204.07076v1[eess.IV]

[11] H. C. A. S. Y. Wu, V. Boominathan and A. Veeraraghavan, "Phasecam3d—learning phase masks for passive single view depth estimation," 2019.

[12] J. Dong, S. Roth, and B. Schiele, "Dwdn: deep wiener deconvolution network for non-blind image deblurring," pp. 9960–9976, 2021.

[13] M. Arsalan, "Single shot depth and image using engineered point spread function (psf)," 2013.

[14] C. A. M. Hayato Ikoma, Cindy M. Nguyen, "Depth from Defocus with Learned Optics for Imaging and Occlusion-aware Depth Estimation."

[15] C. P. D. Eigen and R. Fergus, "Depth map prediction from a single image using a multi-scale deep network."

[16] B. E. Bayer, "Color imaging array," 1976.

[17] G. E. Healey and R. Kondepudy, "Radiometric ccd camera calibration and noise estimation."

[18] V. R. Y. Tsin and T. Kanade, "Statistical calibration of ccd imaging process."

[19] A. L. K. Brenner and J. O.-C. eda, "The ambiguity function as a polar display of the OTF."

[20] E. R. D. FitzGerrell, Alan R. and W. T. Cathey., "Defocus transfer function for circularly symmetric pupils."

[21] B. R. S. C. B. P. P. Wang, X. Shen and A. L. Yuille., "SURGE: Surface Regularized Geometry Estimation from a Single Image." 2016.

[22] P. K. N. Silberman, D. Hoiem and R. Fergus., " Indoor Segmentation and Support Inference from RGBD Images." 2012.

[23] D. Gabor, "A new microscopic principle."

[24] J. Geng, "Structured-light 3d surface imaging: a tutorial."

[25] H. Y. S. B. Gokturk and C. Bamji, "A time-of-flight depth sensor-system descrip- tion, issues and solutions," 2004.

[26] D. L. H. Sarbolandi and A. Kolb, "Kinect range sensing: Structured-light versus time-of-flight kinect."

[27] H. H. Baker, "Depth from edge and intensity based stereo."

[28] P. Grossmann, "Depth from focus."

[29] M. Subbarao and G. Surya, "Depth from defocus: A spatial domain approach."

[30] Z. W. Y. Cao and C. Shen, "Estimating depth from monocular images as classification using deep fully convolutional residual networks."

[31] F. D. A. Levin, R. Fergus and W. T. Freeman, “mage and depth from a conventional camera with a coded aperture.”

[32] S. L. C. Zhou and S. K. Nayar, “Coded aperture pairs for depth from defocus and defocus deblurring.”

[33] S. Quirin, “Quantitative optical imaging and sensing by joint design of point spread functions and estimation algorithms. .”

[34] R. Berlich and S. Stallinga, “High-order-helix point spread functions for monocular three- dimensional imaging with superior aberration robustness.”

[35] S. E. H. Haim and E. Marom, “Depth estimation from a single image using deep learned phase coded mask,” 2018.

[36] J. Chang and G. Wetzstein, “Deep optics for monocular depth estimation and 3d object detection.”

[37] P. F. O. Ronneberger and T. Brox, “U-net: Convolutional networks for biomedical image segmentation.”

[38] N. Wiener, “Extrapolation, interpolation, and smoothing of stationary time series: with en- gineering applications.”

[39] W. H. Richardson, “Bayesian-based iterative method of image restoration.”

[40] C. L. L. Xu, J. S. Ren and J. Jia., “Deep convolutional neural network for image deconvolution.”

[41] W. S. L. R. L. J. Zhang, J. Pan and M. H. Yang., “Learning fully convolutional networks for iterative non-blind deconvolution.”

[42] S. R. J. Dong and B. Schiele, “Deep wiener deconvolution: Wiener meets deep learning for image deblurring.”

[43] M. Mel, “Deep Learning Based Depth And Image Reconstruction Using Rotating Point Spread Functions,” 2021.

[44] M. Beijersbergen, R. Coerwinkel, M. Kristensen, and J. Woerdman, “Helical-wavefront laser beams produced with a spiral phaseplate,” pp. 321–327, 1994.

[45] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.

[46] H. Ikoma, C. M. Nguyen, C. A. Metzler, Y. Peng, and G. Wetzstein, “Depth from defocus with learned optics for imaging and occlusion-aware depth estimation,” IEEE, pp. 1–12, 2021.

[47] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," pp. 4040–4048, 2016.

[48] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," pp. 2650–2658, 2015.

[49] S. F. Bhat, I. Alhashim, and P. Wonka, "Adabins: Depth estimation using adaptive bins," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4009–4018.

[50] Z. Hao, Y. Li, S. You, and F. Lu, "Detail preserving depth estimation from a single image using attention guided networks," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 304–313.

[51] X. Qi, R. Liao, Z. Liu, R. Urtasun, and J. Jia, "Geonet: Geometric neural network for joint depth and surface normal estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 283–291.

[52] I. Alhashim and P. Wonka, "High quality monocular depth estimation via transfer learning," *arXiv preprint arXiv:1812.11941*, 2018.

[53] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.