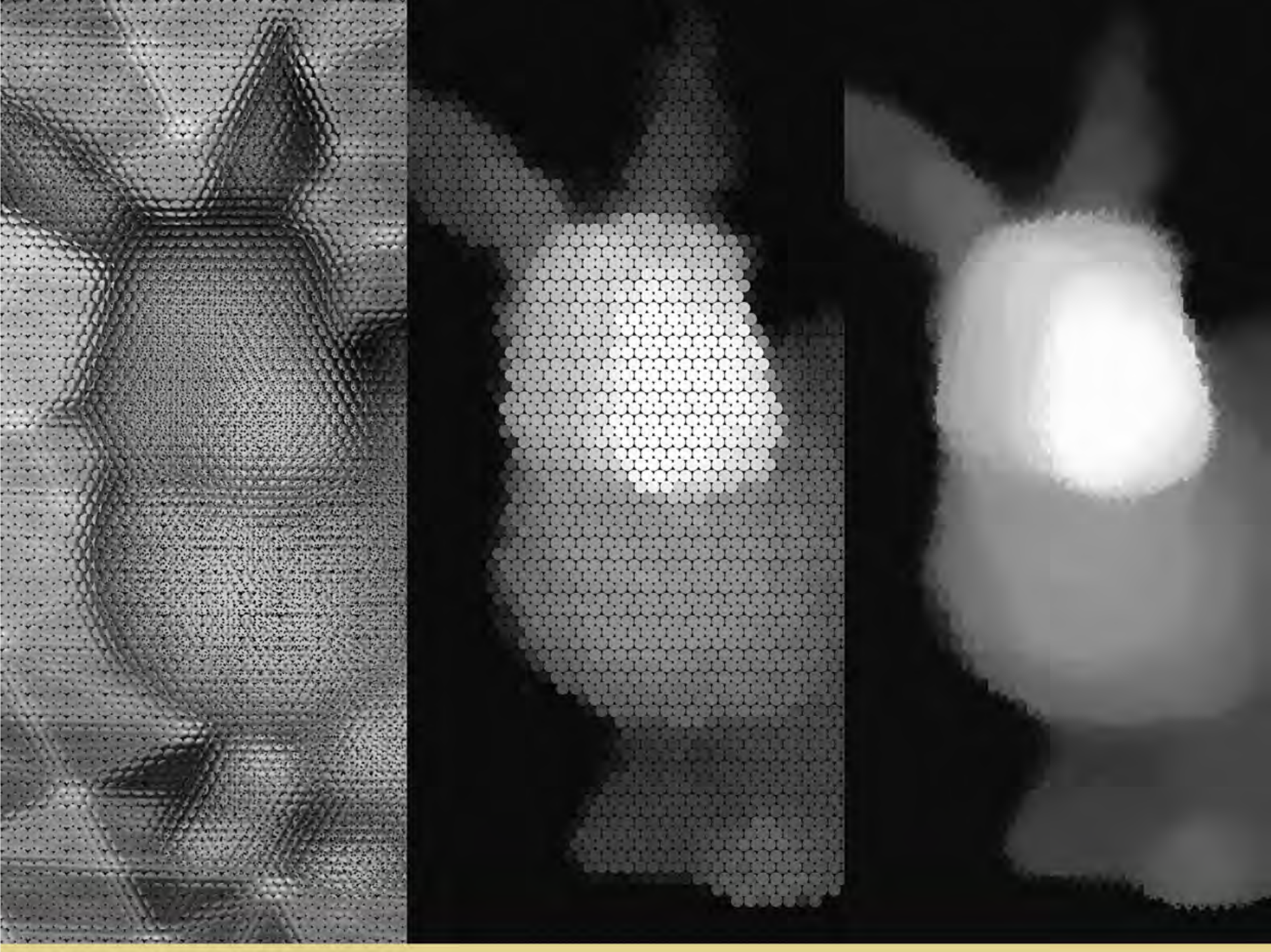


Rodrigo Miguel Belo Leal Toste Ferreira

A fully automatic depth estimation algorithm for multi-focus plenoptic cameras: coarse and dense approaches

February 2016

· U C ·

Universidade de Coimbra

Department of Electrical and Computer Engineering,
Faculty of Sciences and Technology, University of Coimbra,
3030-290 COIMBRA, PORTUGAL.

A Dissertation for Graduate Study in MSc Program
Master of Science in Electrical and Computer Engineering

A fully automatic depth estimation algorithm for multi-focus plenoptic cameras: coarse and dense approaches

Rodrigo Miguel Belo Leal Toste Ferreira

Supervisor:
Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves

Jury:
Prof. Doutor Jorge Manuel Moreira de Campos Pereira Batista
Prof. Doutor João Pedro de Almeida Barreto
Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves

February 2016

Agradecimentos

Quero começar por agradecer aos meus pais António e Luísa e irmãos Rui e Raquel pelo apoio incondicional, paciência e aconselhamento que me deram durante todo o meu percurso. A eles devo tudo o que sou hoje.

Quero também agradecer ao meu orientador Professor Doutor Nuno Gonçalves pela excelente orientação. Sempre presente, disposto a discutir ideias e, acima de tudo por me incentivar a ser crítico.

Um agradecimento ao Laboratório de Visão do Instituto de Sistemas e Robótica, embora reduzido em número, forte em presença, apoio e entreajuda.

Por último, um agradecimento geral a todos os meus amigos. Por toda a força e apoio, pelos momentos de descontração, pelos momentos de discussão mas principalmente por permanecerem.

Abstract

Light field cameras capture a scene's multi-directional light field with one image, allowing the estimation of depth of the captured scene and focus the image after it has been taken.

In this thesis, we introduce a fully automatic method for depth estimation from a single plenoptic image running a RANSAC-like algorithm for feature matching. We filter the estimated depth points on a global and fine scale, allowing a more accurate depth estimation.

The novelty about our approach is the global method to back project correspondences found using photometric similarity to obtain a 3D virtual point cloud. We use a smart mixture of lenses with different focal-lengths in a multiple depth map refining phase, generating a dense depth map. This depth map is then used to generate very high quality all-in-focus renders. We also introduce a new method for detection and correction of highly blurred areas, which greatly improves the depth estimation of the scene and subsequently the all-in-focus as well.

As far as the author knows, our algorithm is the first fully automatic (zero intervention) method to process multi-focus plenoptic images.

On the previous work a plenoptic data simulator was introduced which allows us to create plenoptic datasets with specific parameters. Knowing the depth ground truth of these datasets we are able to test and improve our algorithm and provide guidelines for future work.

Tests with simulated datasets and real images are presented and show very good accuracy of the method presented. We also compare our results with other methods, being able to achieve comparable results to the state of the art with substantial less processing time.

A short paper was submitted and accepted to Eurographics 2016, the 37th Annual Conference of the European Association for Computer Graphics and a full paper was also submitted to ICCP 2016, an International Conference on Computer Photography.

Keywords: Plenoptic cameras, light field, depth estimation, all in focus, synthetic plenoptic data, Raytrix, Lytro.

Resumo

As câmaras de campo de luz são capazes de capturar o campo de luz multidirecional de uma cena, permitindo a estimação da profundidade da cena capturada e foco da imagem depois de ser tirada.

Nesta tese apresentamos um método algorítmico automático para estimação de profundidade para uma imagem plenóptica, dando uso a um algoritmo do estilo RANSAC para deteção de correspondências para pontos salientes na imagem. Filtramos a nuvem de pontos de profundidade estimada a uma escala global e fina, de forma a obter uma estimação mais precisa.

A novidade do nosso método surge da reprojeção de correspondências encontradas através de semelhanças fotométricas para obter uma nuvem 3D de pontos virtuais. Usamos uma mistura inteligente de micro-lentes de tipos diferentes numa fase de refinamento por múltiplos mapas de profundidade, gerando uma mapa de profundidades denso. Este mapa de profundidades é então usado para processar o "all-in-focus" da imagem. Também apresentamos um novo método para deteção de áreas desfocadas, que melhora a estimação de profundidade da cena e subsequentemente o "all-in-focus".

O nosso algorítmo é o primeiro método completamente automático para processar imagens plenópticas multifocais.

No trabalho anterior foi apresentado um gerador de dados simulados que nos permite criar "datasets" plenópticos com parâmetros específicos. Sabendo a profundidade real destes "datasets" é-nos possível testar e melhorar o nosso algoritmo.

São apresentados testes com "datasets" simulados e com "datasets" reais, mostrando boa precisão por parte do método apresentado. Também comparamos o nosso método com outros métodos, sendo que obtivemos resultados comparáveis com os resultados dos métodos padrão usando menos tempo computacional.

Foi submetido e aceite um artigo (short paper) para Eurographics 2016, 37^{a} Conferência Anual da Associação Europeia de Computação Gráfica, e foi submetido um artigo para ICCP

2016, uma Conferância Internacional de Fotografia Computacional (em análise).

Palavras-chave: Câmeras Plenópticas, campo de luz, estimação de profundidade, all in focus, dados plenópticos simulados, Raytrix, Lytro.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A conventional camera is able to capture a single image representing a scene on a time instant. The camera has to be set with the desired focus before taking the image or else another image has to be taken. With a plenoptic camera, the image can be focused after it has been taken. This is possible due to the fact that plenoptic cameras capture a limited light field of the scene. Light field is thus the amount of light flowing in every direction through every point in space.

This type of cameras can capture the light field of the scene due to its setup. While a conventional camera is composed of a main lens and image sensor, on a plenoptic camera a micro-lens array is introduced between the main lens and the image sensor. This micro-lens array projects multiple views of the captured scene onto the image plane. Additionally this allows the depth estimation of the scene.

In 1908 Lippmann [13] introduced the concept of plenoptic cameras where he places a small lenses in front of the film. Lippmann's concept was latter refined by Ives [10].

This concept could no be fully explored at the time due to hardware limitations. Nowadays with the fast growth of computational power, higher quality and resolution of image sensors and lens manufacture refinement, it is possible to create high resolution plenoptic camera sensors and process all its captured light field.

Plenoptic cameras thus open new possibilities for several fields such as photography, production line inspection, 3D reconstruction, face recognition, SLAM (simultaneous localization and mapping) [20], cellphones, endoscopy [9] and many other fields related with computer vision, robotics and computer graphics.

There are some concepts used throughout this thesis that need to be previously explained, such as coarse depth map, dense depth map and point set. Starting by the simplest, a point set is a group of 3D points that, in this case, are 3D representations of depth information. We refer to point set (or global point set) as all the depth points estimated through our algorithm for the plenoptic image. Local point sets consist of subgroups of 3D points projected into each micro-lens. These projected 3D points are originally chosen from the global point set. A coarse depth map is a depth map based on the micro-lens structure, where each micro-lens contains a single value of depth information. The dense depth map is a depth map image where each pixel contains a color value representing the captured scene's depth.

1.1 Motivation

There has been a fast growing on 3D content for both personal or industrial use. Nowadays industrial inspection is often made via 3D reconstruction and, on a consumer base, more and more applications use augmented reality as a form of entertainment. Google for example, with the promising project ”Tango” intend to deliver 3D augmented reality, SLAM (simultaneous location and mapping) and many other 3D reconstructing applications all on a single hand held device.

Plenoptic cameras are an alternative to the conventional stereo setup, being able to capture a scene's light field with only one 2D image. Extracting the scene's depth information accurately is our main topic, hoping to develop new and faster methods.

Since Raytrix's algorithm is not publicly available, our work becomes more challenging. We aim to obtain even better results than Raytrix and develop alternative techniques for the depth estimation.

1.2 Objective

This thesis is the continuation of two previous master thesis, first by Custódio's [5], followed by Cunha's [4] master thesis.

Our objective is to improve the depth estimation while maintaining its fully autonomous algorithm. In Cunha [4] a coarse depth map is composed, from which a dense depth map is synthesize. One of our objectives is to test coarse depth maps with multi-depths per micro-lens and draw a conclusion on its accuracy for the coarse depth estimation. We also intend to compare our improved depth estimation with the state of the art Fleischmann and Koch [7] which is fully replicable, alongside with a comparison with Raytrix's results whose algorithm is closed.

1.3 Contributions

In this thesis we present several contributions. These contributions made the depth estimation improvement possible for both the coarse depth map and the dense depth map.

- Depth estimation from improved point set. An outlier removal was introduced in the previous work removing grain outliers. Additionally we applied a standard deviation filter to each micro-lens, for a more robust filtering. This filtering is applied to the point set of each micro-lens, removing outliers from the local depth point clouds.

- Merging of multiple depth maps to produce a more accurate depth estimation. Cunha's point set estimation is an estimation from lens of the same type. We do a smart mixture of several depth maps estimated though lenses of different types. Since the lens type used for the point set estimation depends on the scene initial depth estimation, the final point set is more accurate, using more adequate lens types for different depths of the scene.

- Detection and correction of highly blurred areas. By analyzing different point set estimations, for different acceptance values of the RANSAC like algorithm, we are able to identify and label highly blurred areas in the image. If the area is considered as blurred, a fixed depth is assigned to that area, based on the point set of non-blurred area.

- Coarse depth map with multiple depth on each lens. As stated before, Cunha composes a coarse depth map with one depth per micro-lens and uses it to reconstruct a dense depth map. We implemented two types of coarse depth map with multiple depths on each lens. One with two depths per lens, based on a dual-clusterization of each micro-lens point set. Each lens is sectioned based on the cluster's position relative to the micro-lens center. The second coarse depth map with multiple depth on each lens is a merge of Cunha's single depth per lens and a 2D interpolation of each lens point set.

- Fully automatic algorithm. All of the introduced improvements are automatic and do not require any parameter from the user. The code has been modularized and reorganized into functions, facilitating its understanding for future developers.

- Improved performance. With a dynamic memory usage and optimized processes, we where able to improve the overall performance of the algorithm.

1.4 State of the art

As stated before, the concept behind plenoptic cameras was first addressed in 1908 by Lippmann [13] and later refined by Ives [10] in 1930. Nowadays with digital image sensors we are able to process the plenoptic images and extract information from them. This technology has several possible applications such as robotics, face recognition, photography and filmography, augmented reality, depth reconstruction, industrial inspection and more.

1.4.1 Depth estimation

We are able to achieve the scene depth with only one raw image, which is also essential for image rendering.

In 2004 Dansearau and Bruton [6] proposed a method for depth estimation using 2D gradient operations. Using a two plane parametrization *(s, u)* and *(t, v)* they were able to define the light field direction and thus the depth of the corresponding elements within the light field. The areas where the depth could not be estimated were filled by applying region growing.

Since plenoptic cameras are not immune to spatial aliasing, which can result on depth estimation errors, in 2009 Bishop and Favaro [3] applied a different approach to compensate the present aliasing, allowing them to recover the depth map from the multiple views provided by the 4D light field. Their method produced good results for both real and synthetic data.

Wanner and Goldluecke [19] presented in 2012 a technique for depth estimations for 4D light fields, using dominant directions on epipolar plane images. By assuming that the 4D light field can be sliced into 2D dimensions they started to locally estimate the depth of the epipolar plane images and then labeled the local estimations, integrating them on the global depth maps by imposing spatial constraints.

Most recently, Fleischmann and Koch [7] approached the depth estimation paradigm with disparity between neighbor lenses. Their method requires a very dense sampling of the light field. The depth maps, for each micro-lens, are fused using a regularization process using a semi-global strategy. They further incorporate a semi-global coarse regularization for insufficiently textured scenes. Their results for per-lens dense depth map are well suited for volumetric surface reconstruction techniques and the algorithm is well suited for parallel processing.

1.4.2 Image rendering

As for the image rendering, it consists in converting the plenoptic image into a focused image the same way as a conventional camera would see the world.

There are three main approaches within the image rendering, one of them introduced by Ng et al. [16] in 2005 where each micro-lens contributes with only one pixel for the rendered image. The final image will have the same amount of pixels as the number of micro-lenses, allowing a fast processing without the usage of depth. This is good for hand held cameras with integrated rendering capabilities.

A different approach was presented by Lumsdaine and Georgiev [14] where each lens contributes with a small patch, the same way a puzzle works. This approach can be used with high resolution plenoptic cameras and requires a small computational power. However the final results will show many artifacts in the patch borders, requiring a filter to solve the output of the image patching.

Another approach and the one that achieves the best results is proposed by Perwass and Wietzke [17]. Having a scene dense depth map it is possible to back trace each pixel into the image plane. This method allows the render of a high resolution image with few artifacts which can be achieved with a multi-focus plenoptic camera. The major drawback is the high computational power required to process the dense depth map and the image rendering.

1.5 Outline of the thesis

- Chapter 2, Light Field Cameras. It presents the theory behind light fields and plenoptic cameras. A short explanation of optics geometry follows, for both standard cameras and plenoptic cameras. This chapter serves as background for the understanding of the rest of the thesis.

- Chapter 3, Depth Estimation. First, it is explained the concept of virtual depth, followed by the algorithm for depth estimation of both Cunha [4] and our improved work. The main topics of the depth estimation are the estimation of the global point set, estimation of the coarse depth maps and dense depth map and finally a review of Fleischmann and Koch [7].
- Chapter 4, Experiments and Results. In this chapter we present the obtained experimental results for the produced tests on simulated data and real data.
- Chapter 5, Conclusions and Future Work. It is presented the final conclusions from the work and suggested future work.

Chapter 2

Light Field Cameras

In this section we present the theory of light fields and plenoptic cameras. We begin with light field's formalization in order to introduce the plenoptic camera's concept and how they work. This chapter serves as background information for a better understanding of the rest of the thesis.

2.1 Light Fields

A conventional camera consists of a light sensor behind a main lens and is able to capture the light's wavelength (normally defined as λ) that passes through the main lens, thus forming a color photograph. This camera captures light rays for a single instant of time and a single point of view, forming a light field. The light field can be formalized as $l(\theta, \phi, \lambda)$ in polar coordinates or $l(x, y, \lambda)$ in Cartesian coordinates, where (θ, ϕ) and (x, y) are light rays directions in the corresponding coordinates. Figure 2.1 represent both coordinate configurations.

For a single lens conventional camera, it is only possible to capture light's wavelength from a single point of view for a time instant. For a video recording we add the time variable to the light field function l. With this addition we are able to record the light field over time from a single point of view. To fully describe light's wavelength over all space and all time we finally added the point of view to the plenoptic function, being V_x, V_y and V_z the observer position. Equation (2.1) describes this plenoptic function that represents all rays of light traveling through time and space which depend on all previous variables. Those variables are the wavelength, time, observer position and orientation. The light field concept was formalized by Adelson and Bergen [1] but it is too complex to be worked with as some variables can be simplified.

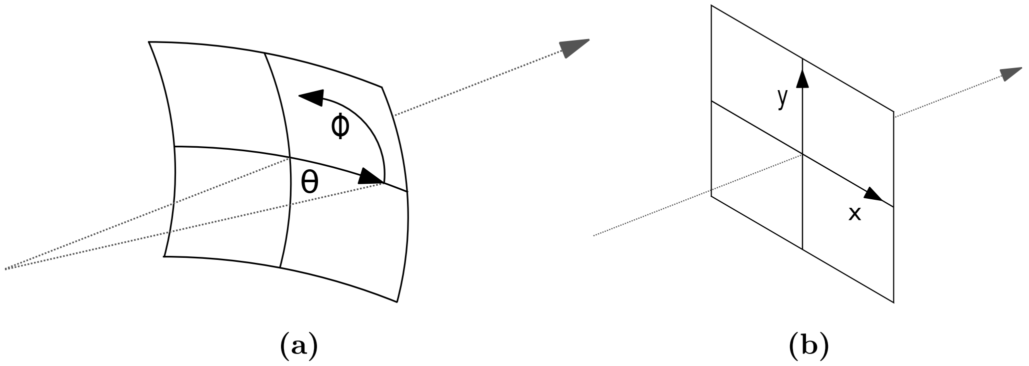


Fig. 2.1: A light field's light ray can be defined in polar coordinates or Cartesian coordinates. Figure 2.1a illustrates polar coordinates from one perspective where a direction is defined using two angles *(θ, ϕ).* The same perspective in the case of Cartesian coordinates is illustrated in figure 2.1b, where a direction is defined with *(x, y).*

$$l = l(\theta, \phi, \lambda, t, V_x, V_y, V_z) \tag{2.1}$$

The 7D light field plenoptic function of equation (2.1) can be reduced to a 4D function, given by equation (2.2) by using a two-plane parameterization (Levoy and Hanrahan [12], Gortler *et al.*[8]). This parameterization defines a light ray with its intersection with two non coincident parallel planes. A generic two-plane parameterization is illustrated in figure 2.2, demonstrating a 4D parameterization in space.

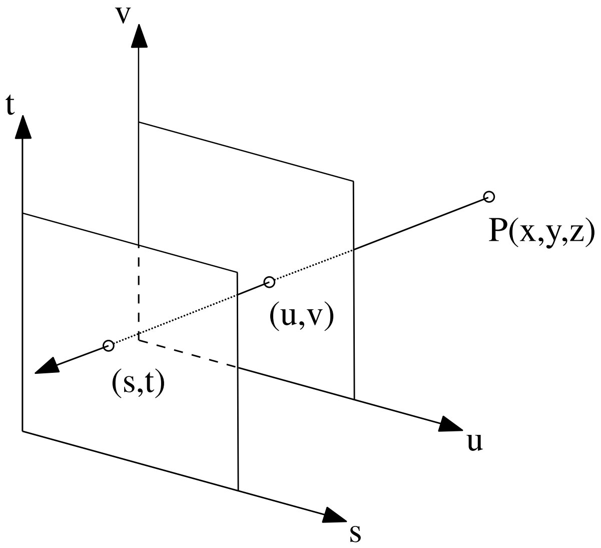


Fig. 2.2: Two-plane parameterization for a generic light ray. The light ray emitted by P is defined by the intersection with each plane.

By removing time and wavelength from the light field we obtain the 4D plenoptic function of equation (2.2). Wavelength is removed because it is used gray-scale plenoptic images, being each pixel represented by a light intensity value. Since we are dealing with photography, the time variable is also removed.

$$l = l(s, t, u, v) \tag{2.2}$$

2.2 Plenoptic Cameras

The concept behind plenoptic cameras has been studied over the last decade and a half, resulting in the development of some devices to acquire the light field of a scene such as camera arrays, multiplexing cameras and plenoptic cameras.

What differs a plenoptic camera from a conventional camera is the placing of a micro-lens array between the image sensor and the camera's main lens. This allows the capture of the light field from various points of view, forming a 4D light field with a 2D image, thus it allows us to estimate the scene's depth and a possible all-in-focus image render.

2.2.1 Micro-lens arrays

A plenoptic camera captures a 2D image of a 4D light field. For the two-plane parameterization we consider the image sensor and the micro-lens array. Starting from its source, a light ray intersects the micro-lens array and then it is projected into the image sensor, giving several perspectives of the same source. The first concept of a plenoptic camera proposed by Lippmann [13] is illustrated by figure 2.3.

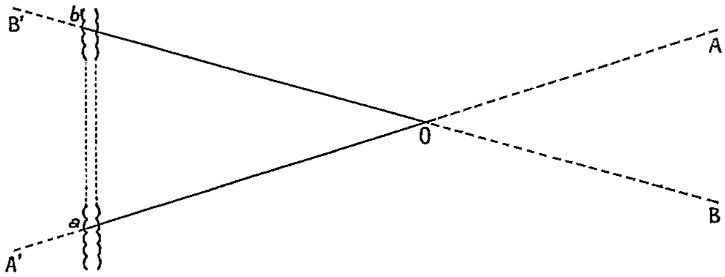


Fig. 2.3: First concept of a plenoptic camera proposed by Lippmann [13]. This side view shows two light rays A and B of the same point O being projected through the micro-lens array, obtaining A' and B'

There are several different configurations for the micro-lens array, being the two most common represented on figure 2.4. These are the orthogonal configuration (figure 2.4a) and the hexagonal configuration (figure 2.4b). In our work we use datasets provided by Raytrix and they use the hexagonal configuration so we are more interested on that particular configuration. The advantage of the hexagonal configuration over the orthogonal one is the better coverage of the image plane, since the gaps between lenses (which do not contain useful information) are smaller on the hexagonal configuration, maximizing the information captured by the image sensor.

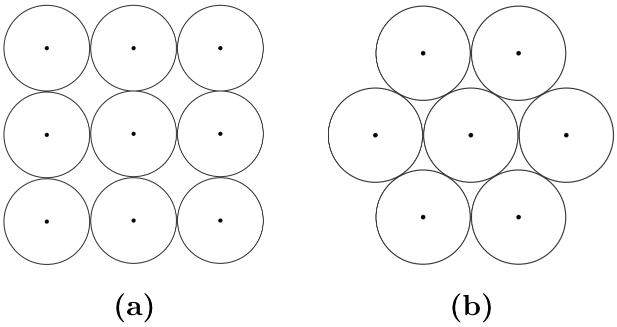


Fig. 2.4: Two most common micro-lens configurations. (a) Orthogonal. (b) Hexagonal.

2.2.2 Optics geometry

On a conventional camera the main lens generates an image of a 3D point by gathering the whole cone of light that emanates from that point, refocusing all the cone of light rays into a single point on the image plane. This behavior is ideal and does not always happen.

The main lens can lead to defocus if not placed at the correct focal distance from the image plane (or vice-versa). In this case, as shown in figure 2.5 the cone's light rays do not converge at a single point at the image sensor, producing a blurred image described by its circle of confusion.

For a sufficiently thin lens we can use the thin lens equation, equation (2.3), to calculate the distance between the lens and the image plane. In equation (2.3), a is the distance between the real-world point and the lens, b the distance between the lens and focal plane and f the focal length. We can guarantee that a projected point is in focus if equation (2.3) is satisfied.

$$\frac{1}{f} = \frac{1}{a} + \frac{1}{b} \tag{2.3}$$

- f - focal distance.

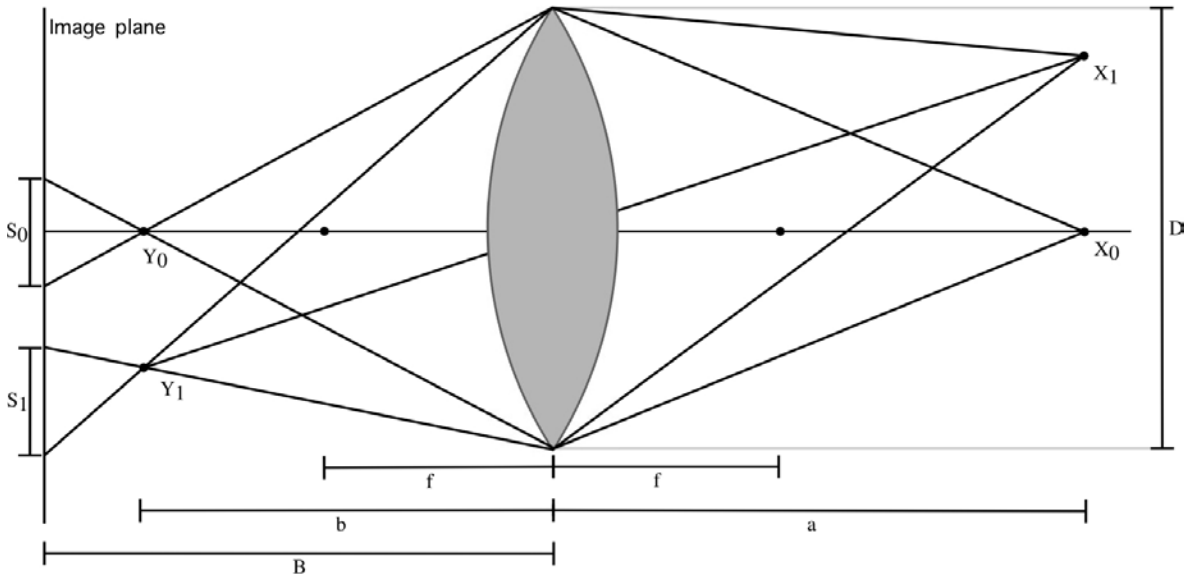


Fig. 2.5: Image formation on a standard camera with a thin lens model.

- D - lens aperture.
- a - distance from the lens plane to object.
- b - distance from the lens plane to projected image through the lens (parallel to the optical axis).
- B - distance from the image plane to the lens plane.
- X_0 and X_1 - real-world object points.
- Y_0 and Y_1 - images formed by projecting X_0 and X_1 through the lens.
- S_0 and S_1 - blur diameter (also known as circle of confusion) from the projected images Y_0 and Y_1 in the image plane.

On a plenoptic camera, an object is projected into a virtual object through the main lens and then projected by the micro-lenses into the image plane. As explained before, this allows the capture of several perspectives of the same virtual point, thus capturing a 2D image containing the 4D light field. This is illustrated in figure 2.6.

For plenoptic cameras, we can assume the thin lens equation (2.3) to establish a relation between the real-world point and the point projected by the micro-lenses. To express the relation between the main lens and micro-lenses we define the f-number. The f-number defines the ratio between the focal-length of a lens and its diameter (this diameter is variable and is called, in photography, aperture).

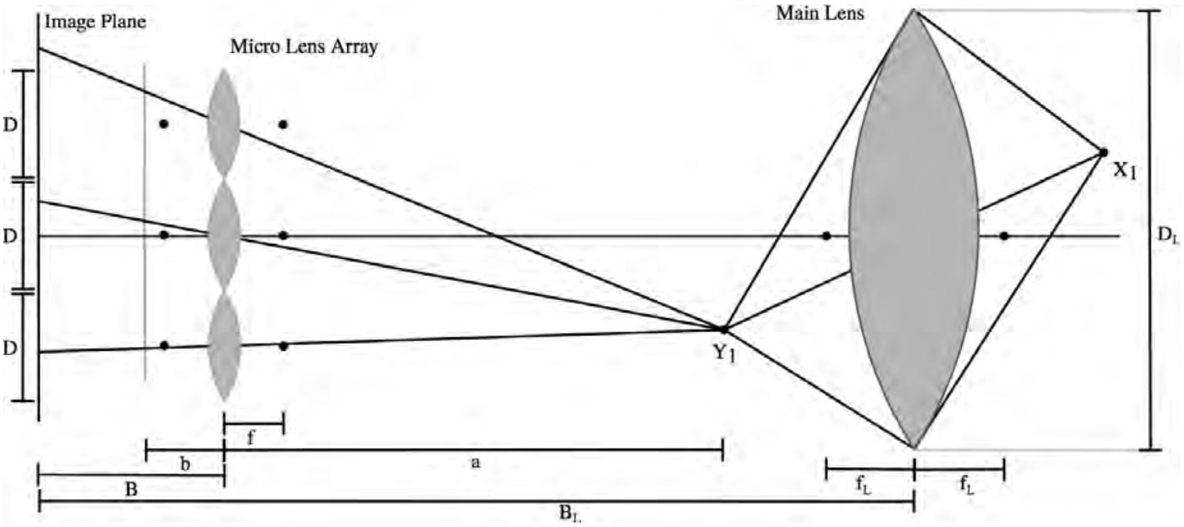


Fig. 2.6: Image formation on a plenoptic camera.

- f - micro-lens focal distance.
- D - micro-lens aperture.
- a - distance from the micro-lens plane to virtual image plane.
- b - distance from the micro-lens plane to projected virtual image through the micro-lenss.
- B - distance from the image plane to the micro-lens plane.
- f_L - main lens focal length.
- D_L - main lens aperture.
- B_L - distance from the main lens to the image plane.

In order to maximize the usage of the image sensor, the f-number of the micro-lenses should match the f-number of the main lens, equation (2.4). This avoids overlapping images or oversized gaps of the micro-lenses on the image plane, meaning that the micro-lenses images will touch each other but will not overlap.

$$\frac{B}{D} \approx \frac{B_L}{D_L} \tag{2.4}$$

2.2.3 Standard plenoptic cameras

As proposed by Ng [16], the focal length of the micro-lenses on a standard plenoptic camera is equal to the distance from the image sensor to the micro-lens array, $f = B$. All micro-lens has the same focal length. On this case, each lens contributes with only one pixel value in the final image, being the final image resolution equal to the number of micro-lenses. This drastically reduces the images resolution. On the other hand it reduces the computational power needed to process the image, being adequate for compact hand-held cameras with built in rendering capabilities. Since each micro-lens contributes with only one pixel value for the final image, there is no need for micro-lenses projections with high resolution. The Original Lytro, illustrated in figure 2.7 was the first commercially available standard plenoptic camera.

Fig. 2.7: Original Lytro plenoptic camera.

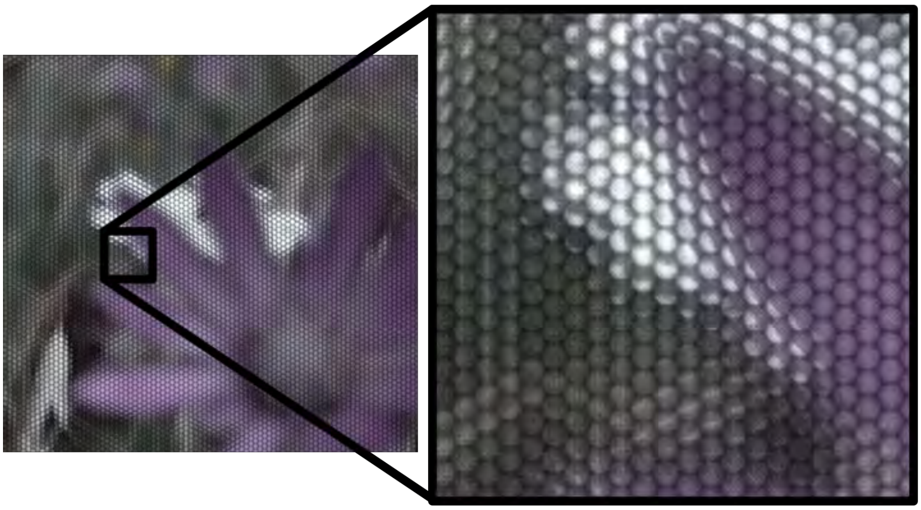


Fig. 2.8: Original Lytro raw plenoptic image.

2.2.4 Multi-focus plenoptic cameras

Multi-focus plenoptic cameras or Plenoptic 2.0 [14] solve the lack of resolution of the standard plenoptic cameras as the full micro-lens image contributes to the final image rather than one pixel per micro-lens. Figure 4.11 shows an example of Raytrix's multi-focus plenoptic camera.

Fig. 2.9: Raytrix's R8 multi-focus plenoptic camera.

The main difference from the standard plenoptic camera and the multi-focus plenoptic camera is on the micro-lens array. On multi-focus plenoptic cameras there are at least three types of micro-lens, each type with different focal lengths. Their most common structure is shown in figure 2.10 where we have an hexagonal configuration and the numbers represent the lenses type, being each lens surrounded by lenses with different type from its own.

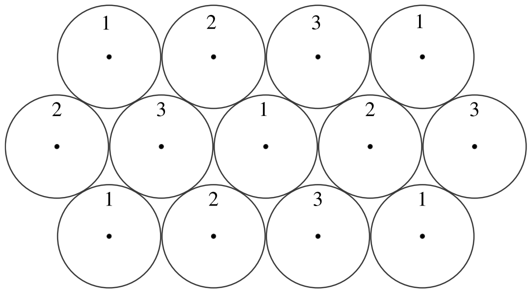


Fig. 2.10: Hexagonal micro-lens configuration with micro-lens type marked by numbers.

Having different focal lengths among the micro-lens allows to obtain a larger depth of field over the scene. Each micro-lens is in focus within a depth range thus, for a given depth of the scene, there is at least one micro-lens type in focus. For a far plane one micro-lens type will be in focus but will be out of focus on other micro-lenses. Figure 2.11 show this behavior and identify different micro-lens types by the blur they produce.

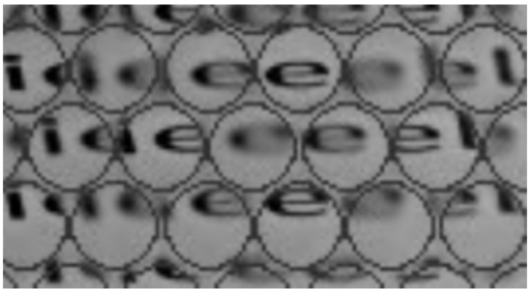


Fig. 2.11: Different blurs produced by different focal length micro-lenses.

When a virtual point is out of focus, the cone of light rays that emanate from that point will not converge into a single point in the image plane (as shown in figure 2.5). This will produce a blurred image where the blur can be defined by the diameter of the cone of light rays (S_0 and S_1 in figure 2.5). The blur diameter is given by equation (2.5), where f is the focal length of the lens which projects the virtual point, d is the depth of the virtual point and A is the lens aperture (in this case the lens diameter).

$$S = \frac{fA}{d} \tag{2.5}$$

We can see in figure 2.12 a representation of a camera projecting different points at different distances to the lens plane.

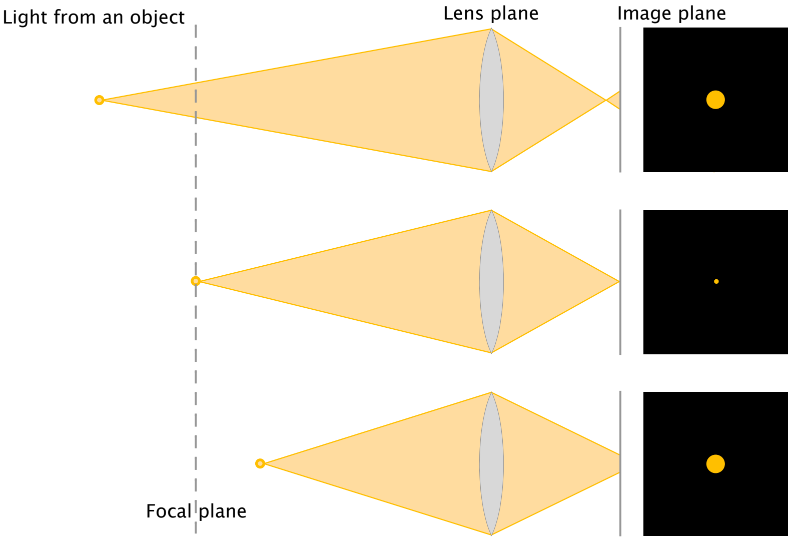


Fig. 2.12: Optics geometry for different points with different distances to the lens plane.

Chapter 3

Depth Estimation

In this chapter we focus on the depth estimation and its improvement as it is the main objective of this thesis. We start by explaining what virtual depth is, followed by both previous and improved depth estimation algorithm. Cunha's feature detection [4] will be explained followed by the micro-lens pattern, a depth refining to the previous depth estimation, improved and new approaches to the estimation of the coarse depth map and finally the reconstruction of the dense depth map. At the end, we explain Fleischmann and Koch's algorithm [7] for a better understanding of their approach, since it's a complex algorithm and we present a direct comparison to our algorithm.

3.1 Virtual Depth

Virtual depth is the ratio of the distance between the micro-lens array and the virtual point and the distance between the micro-lens array and the image plane, a and B respectively. The variable a is positive if X_1 is a main lens real image and negative if X_1 is a main lens virtual image. As illustrated by figure 3.1, a virtual image projected by the main lens is given by X_1 and is projected by a micro-lens into Y_1. The virtual depth of X_1 is given by the equation 3.1.

$$v = a/B \tag{3.1}$$

In other words, virtual depth is the number of B times that the virtual point is from the micro-lens array. Notice that for a virtual point with virtual depth of $v = 2$ is at least imaged by two micro-lenses, meaning that a virtual point with virtual depth $v = 3$ is imaged by at least 3 micro-lenses and so on (Perwass and Wietzke [17]).

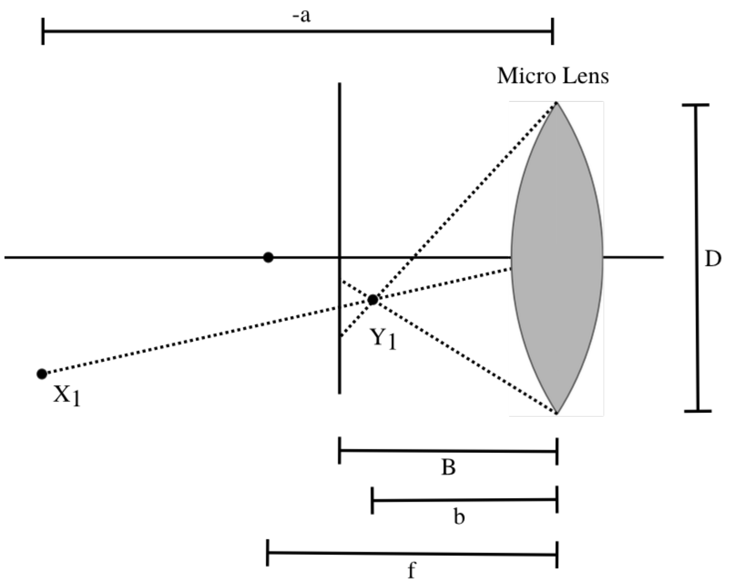


Fig. 3.1: Projection of a virtual point already projected by the main lens.

3.2 Pipeline of the Algorithm

Starting with an overview, our automatic algorithm follows the pipeline of figure 3.2, where gray squares represent processes and gray balloons represent inputs and outputs. The algorithm is modular, making it easy to improve each module separately and introduce new modules with new features.

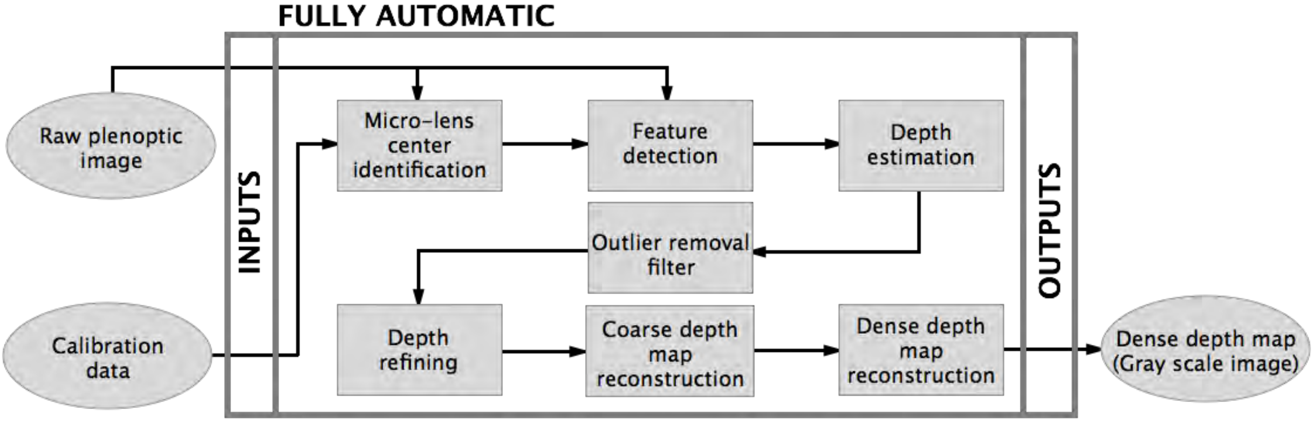


Fig. 3.2: Pipeline diagram for our full algorithm.

The algorithm has two inputs, the raw plenoptic image and the camera's calibration data. First, knowing the position of the central micro-lens of the micro-lens array (given by the calibration data), the algorithm identifies all the micro-lenses in the image. Then we preform a feature detection based on a SIFT descriptor and we estimate the depth of the detected features (with the method described at the beginning of section 3.3). Thus, we obtain a global point set. This point set has some outliers and so we filter it (section 3.3.2) and we proceed to a depth refining (explained in section 3.3.3), which improves the depth estimation of the point set. On the next step, we reconstruct a coarse map (section 3.4) which is the basis for reconstructing the dense depth map. The algorithm outputs a gray scale image for the depth estimation, where the pixel intensity value represents the depth of the captured scene.

3.3 Feature Detection and Depth Estimation

Our algorithm to estimate a dense depth map is based on texture detail matching (photometric similarity) between pairs of micro-lens images. As stated before, we use SIFT descriptor to search for salient points. This method allows us to obtain the most significant points like corners, edges and contrast points only by adjusting threshold parameters. Salient points are then searched for in neighboring lenses to obtain correspondences, by relying on stereo epipolar geometry. Since we are provided a big number of salient points and their correspondences, we apply a RANSAC-like method to obtain the best 3D point cloud. Our algorithm then back projects the pairs of correspondences, since the distance from the micro-lens array and the image plane is provided by the camera manufacturer (calibration data), allowing to obtain a sparse 3D point cloud. Our method is based on the back projection model presented by Perwass and Wietzke [17]. We summarize our method as follows:

- Step 1 - **Selection of an epipolar band** - For each correspondence, a subset of three lines is considered. The central epipolar line is defined by the salient point and the test correspondence. For each epipolar line two adjacent parallel lines are incorporated in the model, representing a one pixel error tolerance. This step is illustrated in figure 3.3.

- Step 2 - **Estimation of the 3D virtual points** - The previous defined lines are grouped two by two and for each pair it is computed the 3D point that minimizes the distance between them. The final 3D point has the median of their coordinates.

- Step 3 - **Testing the model** - Having an hypothetical 3D point obtained in the previous step, we now need to test the hypothesis for this virtual point. The chosen error measurement is the distance of the virtual candidate point to all the correspondence lines obtained in the previous step.

- Step 4 - **Assessment of the model** - A threshold is defined so we can distinguish the good from the bad estimations. This allows to assume which lines are suited to add to the model (labeled as inliers). If there is more than one outlier, the model is discarded and we go back to the first step. If not, we advance to Step 5.

- Step 5 - **Re-estimations of the 3D virtual point** - This step is similar to Step 2. We re-estimate the 3D virtual point using only the inliers. These lines are again grouped two by two and the 3D point for every combination is the point that minimizes the distance between them. The final 3D point is the median coordinates of all points generated by every line combination.

- Step 6 - **Error metrics** In this step we evaluate the model in terms of error. It is a mean error from the inliers's distances obtained in Step 3.

- Step 7 - **Repeat steps 1-6 for every correspondence**

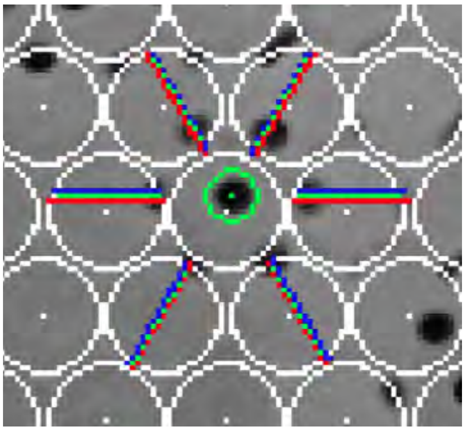


Fig. 3.3: Model tested by the RANSAC-like algorithm. The green circle is the salient point and the red, green and blue lines are the epipolar band where we search for correspondences. The green epipolar line is the main test line while the red and blue lines represent the ±1 pixel tolerance.

The output of the previous algorithm is a 3D point cloud of virtual points as projected by the main lens of the camera to their virtual image (between the main lens and the array of micro lenses). Having the 3D point cloud, a coarse regularization method will reproject the 3D points of the cloud to the micro-lens images and, thus, attribute an average depth value for every micro-lens (section 3.4.1). The final dense depth map is built by weighting the 3D point cloud depth and the coarse depth map. The main contribution of this algorithm is a smart mixture of neighboring micro-lenses of different type that, although with different blurs, are able to improve the depth estimation of the sparse point cloud and of the dense map.

3.3.1 Micro-lens patterns

As for the lens pattern used in Step 1 (where neighbor lenses are searched for replications of a given salient point) we use different combinations of lenses. Knowing that for a multi-focus plenoptic camera there are lenses with different types, we define lens groups based on the lens type and the distance to the central lens. Figure 3.4 shows these configurations. We do a smart mixture of lens groups that, even mixing different blurs, is able to optimize the depth estimated, considering different depth ranges. Notice that the depth accuracy depends on the stereo baseline, which is smaller for higher scene depths. Our smart adaptive mixture of micro-lens is able to adjust baseline and range.

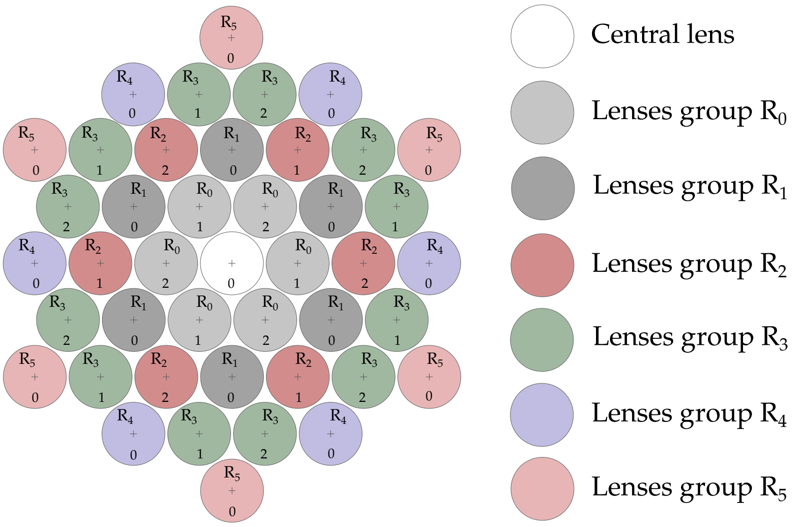


Fig. 3.4: Illustration of the lens neighborhood, with every group labeled from R_0 to R_5, and lens type from 0 to 2. The lower value in the micro-lens illustration is the lens type.

The neighborhood is limited to R_5 because there is no major correspondences above this distance to the central lens (about $3.5D$). Table 3.1 summarizes all lens patterns studied in our work, where D is the diameter of each lens. Another lens configuration usage will be explained on section 3.3.3.

Lenses Patterns	# Of Lenses	Lenses Types	Distance to central micro-lens
R_0	6	1, 2	D
R_1	6	0	$\sqrt{3}\times$ D
R_2	6	1, 2	2×D
R_3	12	1, 2	$\sqrt{7}\times$ D
R_4	6	0	3×D
R_5	6	0	$2\sqrt{3}\times$D

Table 3.1: Table summarizing the lens pattern parameters. For each lens pattern it is presented the number of lenses, lens types and distance to central lens (distance to central lens is presented as multiple of the micro-lens diameter D). It assumes that the central lens type is 0, without loss of generality.

3.3.2 Outlier elimination

Due to the presence of outliers in the 3D point cloud estimated, a filter to remove outliers was applied by Cunha [4], making the final result more robust and immune to high variations in the depth estimation. The filter is applied after the depth estimation algorithm so that we can have a reliable point cloud as basis for the creation of the dense depth map. This filter was presented by Ruse *et al.*[18]. To detect outliers in the point cloud we compute all distances of pairs of points and establish a threshold based on the distribution of distances in a given vicinity. Every point that falls outside of the threshold defined by the average distance to the neighbors (μ) and standard deviation (σ) is considered an outlier. All the outliers are removed from the point cloud.

Additionally, as we project the virtual points to each micro-lens for the coarse depth estimation of section 3.4, we applied a fine filter for each group of virtual points projected through each micro-lens (local point cloud). The presence of outliers on a local micro-lens scale justifies the use of this filter. This filter is based on a median and standard deviation of each local point cloud, reducing local outliers. Our fine filter follows equation 3.2 for a local median $\widetilde{p}$ and standard deviation σ_p of P(n) (local point set with n points).

$$P_{filtered} = \{P(n) : P(n) \in [\widetilde{p} - \sigma_p, \widetilde{p} + \sigma_p] | n \in \Omega_p\}, \qquad \text{where } \Omega_p \text{ is the point set domain} \quad (3.2)$$

3.3.3 Depth refining

Assume z as the virtual depth of a generic point of the captured scene. As stated by Perwass and Wietzke [17], the maximum radius (R_{max}) that determines the number of micro-lenses that replicate a certain feature is given by equation (3.3), where B is the distance between the micro-lens plane and the image plane and D is the micro-lens diameter (in pixels). With that we know that the closer a point is to the camera, the more lenses will replicate it. Figure 3.5 is an example for both close and far depth feature replication on a raw plenoptic image. When using the R_0 lens pattern (see figure 3.4) it searches adjacent lenses with different types for feature matching, being adequate for farther depth ranges. On the other hand, the R_1 configuration is adequate for closer depth ranges since it searches lenses of the same type (same focal length). Notice that, although the number of correspondences obtained using the R_5 configuration is much lower, their baseline is higher and, therefore, the back projection of its correspondences is more stable. Our algorithm then presents an adaptive mixture of micro-lens patterns, by using as many information as possible, and selecting the most stable configurations when available.

$$R_{max} = \frac{\mid z \mid \times D}{2 \times B} \tag{3.3}$$

3.3.3.1 Multiple depth map merging

Our work focuses on the usage of R_0, R_1 and R_5 lens configuration since they produce the majority and most consistent results of all configurations (as studied by Cunha [4]). We propose the aggregation of R_0, R_1 and R_5 point sets by quartile sectioning and weight attribution. Since correspondences in R_5 are not always available (only for very close points) and their blur is similar to the blur of correspondences in R_1 (since these lenses are of the same type), we mention the union of both configurations as $R_1 + R_5$. For the fusion of R_0 and $R_1 + R_5$ depth maps we consider a linear combination of their estimated depths, given by equation (3.4), where α is the weight parameter, varying between 0 and 1.

We divided the depth map range (from the complete point cloud) into quartiles so that the first quartile represents the closer depth points and the fourth quartile the farther depth points relative to the camera position. For the first quartile (closer points) the depth points are extracted from the $R_1 + R_5$ depth map, being $z = z_{R_1+R_5}$. The same applies for the fourth quartile, being $z = z_{R_0}$. For the second and third quartile we use a linear weight of both depth maps.

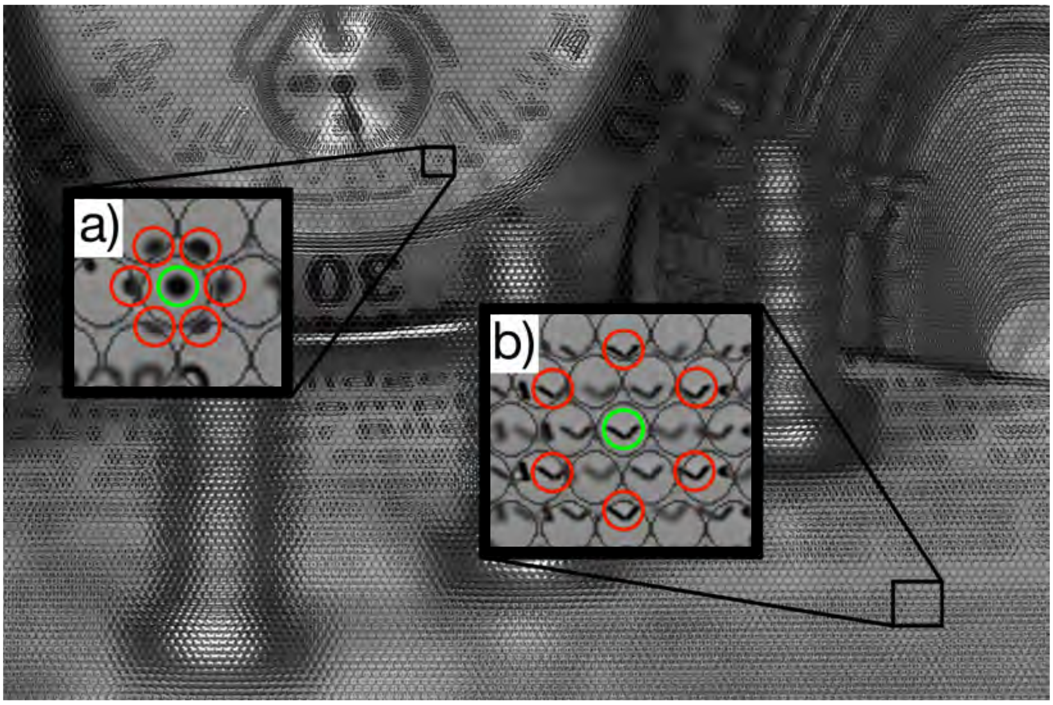


Fig. 3.5: Salient point replication on neighbor lenses on a raw plenotptic image. a) is an example of feature replication for a high z value sample. b) is an example of feature replication for a low z value sample

$$z = \begin{cases} z_{R_0}, & \text{if } \ \hat{z} \in Q4 \\ (1-\alpha)z_{R_0} + \alpha z_{R_1+R_5}, & \text{if } \ \hat{z} \in Q2 \cup Q3 \\ z_{R_1+R_5}, & \text{if } \ \hat{z} \in Q1 \end{cases} \tag{3.4}$$

where $\hat{z} = \frac{z_{R_0}+z_{R_1+R_5}}{2}$.

With this multiple depth map merging we use only our best estimations from each estimated depth point cloud, based on the lenses type used for the estimation of each one.

3.3.3.2 Adaptive depth for highly blurred areas

For this section we consider a minimal solution of the RANSAC-like algorithm a solution obtained from the model with 2 or more correspondences (Step 2 of the algorithm presented in section 3.3). A non-minimal solution is considered a solution obtained with 5 or more correspondences. A minimal solution produces a volumetric point set but is more vulnerable to error. On the other hand, the non-minimal solution produces a less volumetric point set (comparing to the minimal solution) but is less vulnerable to error, being more accurate then the minimal solution.

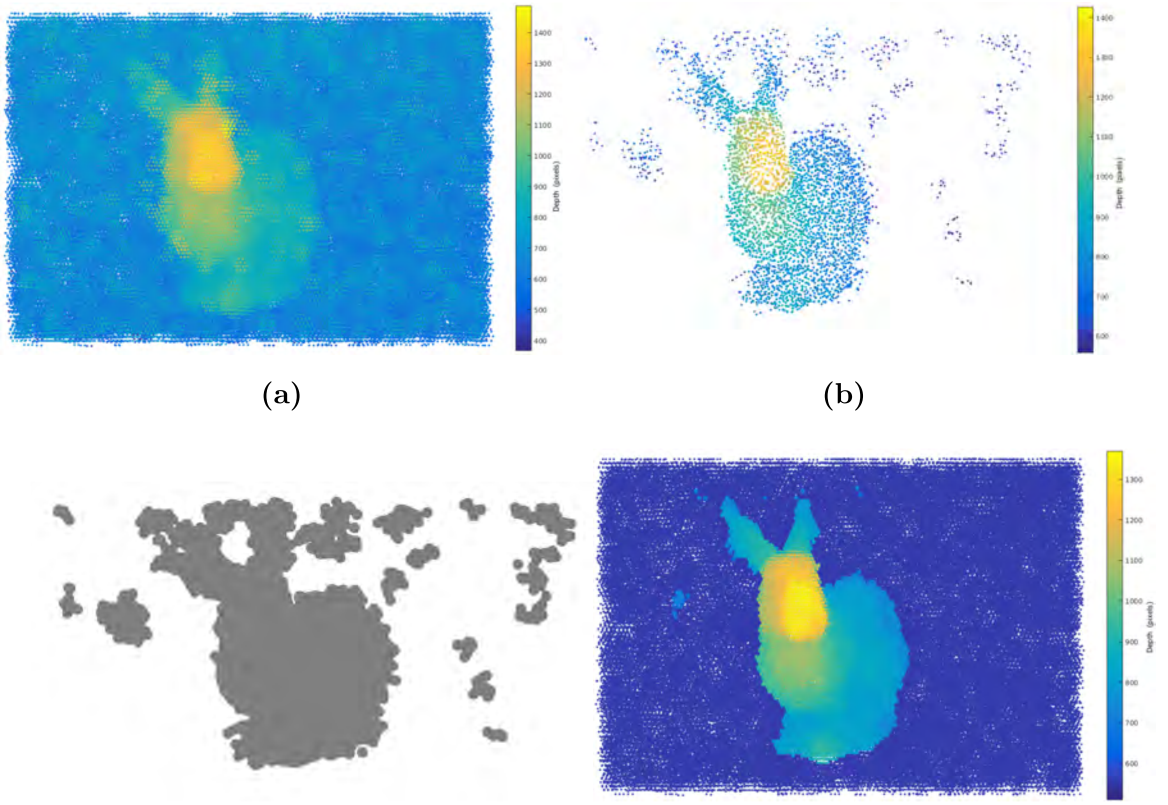
Some plenoptic images might contain sections where none of the lens types can focus. The texture in these farther planes is blurred and it is hard to find salient points since they highly depend on texture detail. Then, the algorithm might detect salient points and correspondences for a minimal solution (when relaxing the correspondences threshold) of the RANSAC-like algorithm, which will result in a less accurate depth map. For these cases the estimated point set is not consistent and assume a noisy representation (this is a classical overfitting problem). For a non-minimal solution (when more correspondences are used) the point set is not dense enough for a sufficiently dense reconstruction. In these cases it is more favorable to assume $z = z_{R_1+R_5}$ rather than a multiple depth map merging for the final point set estimation because $z = z_{R_0}$ produces substantial noise due to the inaccurate photometric similarities detected by the algorithm for these blurred areas. Consequently, for the problem of highly blurred areas we then generate a minimal solution point set and we cross it with a generated non-minimal solution point set of the same plenoptic image. The following steps describes our approach:

- Step 1 - **Section labeling of the depth map with stable estimations - non minimal solution.** We create a label from the non-minimal solutions (that use five or more correspondences and yet are more accurate) to serve as reference. We call it inlier label.

- Step 2 - **Reject all correspondences outside the inlier label area.** The rejected points from the point cloud are labeled as outliers and might represent a highly blurred area.

- Step 3 - **Analyze the rejected points depth.** By establishing a threshold and comparing the outliers's depth mean ($\mu_{outlier}$) with the depth range interval ($[z_{min}, z_{max}]$) we can determine the presence of a defocused scene area. If the outlier's depth mean is lower than the threshold, a farther scene is detected and we assume all the non-minimal outliers depth as z_{min}. On the other hand, if the outlier's depth mean is higher than the threshold, we assume they are not outliers and have useful depth information. Thus, the multiple depth map merging of section 3.3.3.1 is assumed to be a reasonable solution.

Figure 3.6 shows a real example of detected blurred area and its correction.



(a) **(b)**

(c) **(d)**

Fig. 3.6: Example of defocused area detection and correction. (a) R_0 point set with at least 2 correspondences in the RANSAC-like algorithm. (b) R_0 point set with at least 5 correspondences in the RANSAC-like algorithm. (c) Generated label map with gray label for the inliers and white label for outliers. (d) Final corrected and filtered point set estimation for the $R_1 + R_5$ depth map.

3.4 Coarse Depth Map

Having filtered the point set estimated, we synthesize a coarse depth map with one or more depths per micro-lens. This coarse depth map is used to reconstruct the dense depth map as explained in section 3.5. We use three methods for the coarse depth map reconstruction, a depth map with one depth per micro-lens, another with two depths per micro-lens and finally a depth map with multi depth for each micro-lens.

Regardless of the method, we have to identify which features of the point set estimation are projected through each micro-lens. Even though we do not have the focal length value for each micro-lens, we project every feature within the cone centered on every micro-lens and with radius R_{max} (this is of key importance since even without calibration of the lenses we are able to reconstruct depth). We estimate the micro-lens R_{max} from equation (3.3) by averaging the depth value of the point set projected into its R radius. The R_{max} projection is illustrated in figure 3.7.

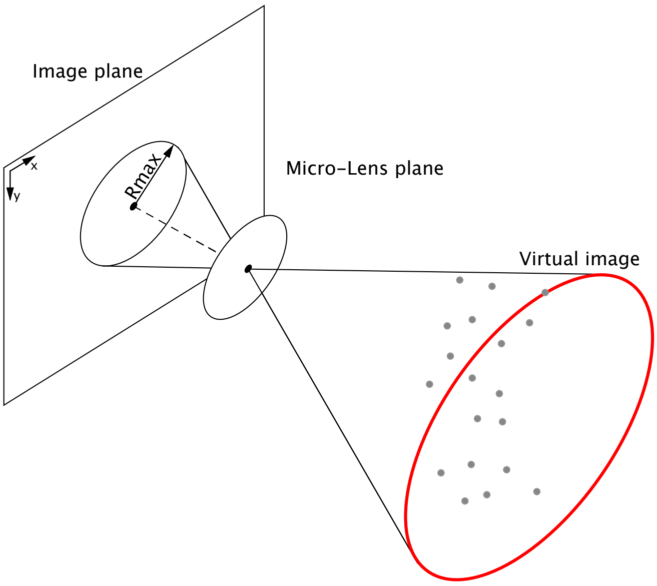


Fig. 3.7: Generic illustration of R_{max} radius projection cone for one micro-lens and features that fall inside it.

3.4.1 Single depth per lens

In Cunha [4], a coarse depth map with a single depth per lens is generated and used as a basis for dense depth map estimation. Instead of working within the pixel dimension, a zoomed out approach was chosen thus working on a micro-lens scale, having one depth per micro-lens.

This algorithm involves some steps related with the tracing from the source virtual object into the image plane. Having the depth estimation of the filtered point set from the merging of the multiple lens type images, we reproject each point into the image plane through the micro-lens array. First we determine which points fall inside the projection cone with R_{max} radius (figure 3.7). Then we project those point into the image plane through the micro-lens assigning a color intensity to each projected point with the same value as its virtual depth. Notice that a point can be projected through several micro-lens depending on its virtual depth v as explained previously on section 3.1.

For each group of points projected into each micro-lens a fine filter is applied. The basis of the filter is explained in section 3.3.2. This filter allows a more robust estimation for the depth of each micro-lens, being this depth the averaging of every point's color intensity that follows equation (3.2).

As part of the previous work, we make use of a propagation algorithm to densely fill every micro-lens without depth information by propagating its neighbor lens's depth value. The propagated depth is an averaging of the neighbor lenses depth, as it is assumed a robust propagation only if there are three or more neighbor lenses with depth information.

3.4.2 Micro-lens sectioning

One of our approaches to improve the depth estimation is the sectioning of the micro-lens into two depths. For this we use a clusterization algorithm called k-means [15].

The k-means algorithm is a self-learning (loop) algorithm based on vector quantization. This method classifies a $N-$dimension point set through k number of clusters. The main objective is to find k centroids, each one representing the center of a group of points.

K-means aims at minimizing the square error function described by equation (3.5) where $x_i^{(j)}$ is the data point and c_j the cluster center.

$$E = \sum_{j=1}^{k} \sum_{i=1}^{N} \left\| x_i^{(j)} - c_j \right\|^2 \tag{3.5}$$

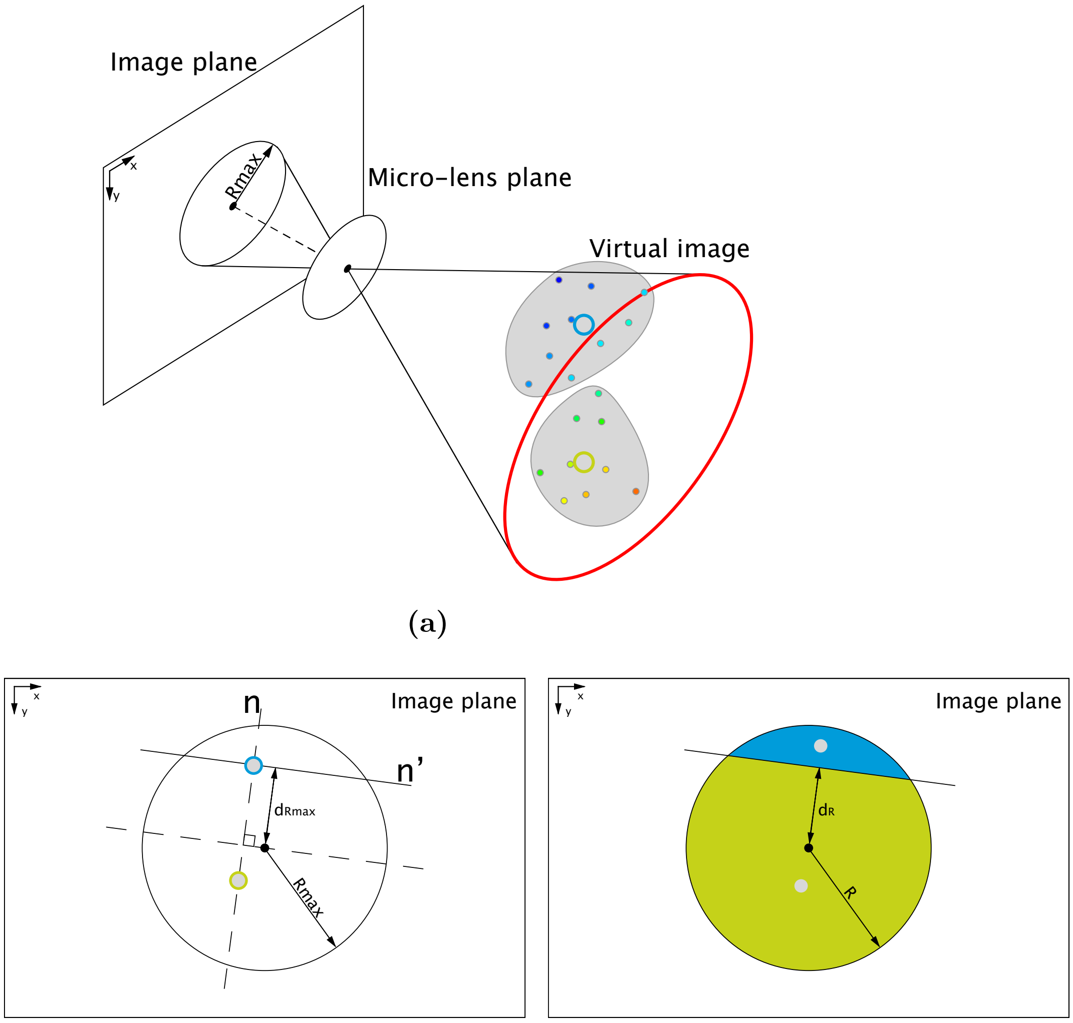
For the first centroid placement, since different locations causes different solutions, they are placed far away from each other. This simple algorithm is described by the following steps:

- Step 1 - **Choose k points**. These are the initial group of centroids;
- Step 2 - **Assign each point of the $N-$dimension point set to one centroid**. The point is assigned to the centroid whose distance to is minimum. This step is processed for every point in the point set.
- Step 3 - **Recalculate the position of the k centroids**.
- Step 4 - **Repeat Steps 2 and 3**. This only applies if the centroids position moves, otherwise the algorithm stops.

This method allows us to separate the point set into groups. Since the algorithm is sensitive to the initial randomly selected cluster centers, it can be triggered several times to reduce the error effect of the random initial conditions.

The C++ library OpenCV (computer vision) already has an implementation of this algorithm that is optimized for CPU (central processing unit) parallel processing (multithreading). Given a point set and the number of clusters desired, the function returns the clusters's centers and labeled point set according to the cluster it has been assigned.

As for the micro-lens sectioning, similarly to the single depth per lens approach, we identify which points fall inside the projection cone with a radius R_{max} for each micro-lens. Having a



(a)

(b)

(c)

Fig. 3.8: (a) R_{max} projection cone, features that fall inside it (dense colored points) and cluster's center (border colored circles). (b) Clusters and R_{max} projection on the image plane. Normal line (continuous line) passing through the farthest cluster relative to the micro-lens projected center. (c) Sectioned micro-lens with assigned depths equal to the clusters virtual depth.

local point set for each micro-lens, without projecting these points, we group them into 2 clusters with the k-means OpenCV function and extract their centers. This is illustrated in figure 3.8a.

Following, the clusters centers are projected into the image plane through the micro-lens center into the image plane, assigning them a color intensity value of their respective virtual depth. As seen in figure 3.8b, we then calculate the 2D linear equation that contains both projected cluster centers (designated as n in the image). Following, we calculate its normal line and intersect it with the cluster that maximizes the distance to the projected center of the micro-lens (designated as n' in the image). This normal line's equation is then normalized and scaled to the R radius of the micro-lens. The assigned depth for both partitions is equal to the color intensity of the assigned cluster (figure 3.8c).

The micro-lens sectioning only occurs if the two clusters present a significant depth difference. Otherwise, we assume a single depth per micro-lens.

3.4.3 Second order fitting

For a multiple depth per lens approach we chose to estimate a surface for each micro-lens based on a 3D least square approximation. This method is simple and its accuracy increases with the size of the point set.

As stated by Ambrosius [2], given a set of n points $x_1 \dots x_n, y_1 \dots y_n$, with corresponding $z_1 \dots z_n$ and degree p, it is possible to find a polynomial of degree p that fits the data with a minimum error in the least squares sense.

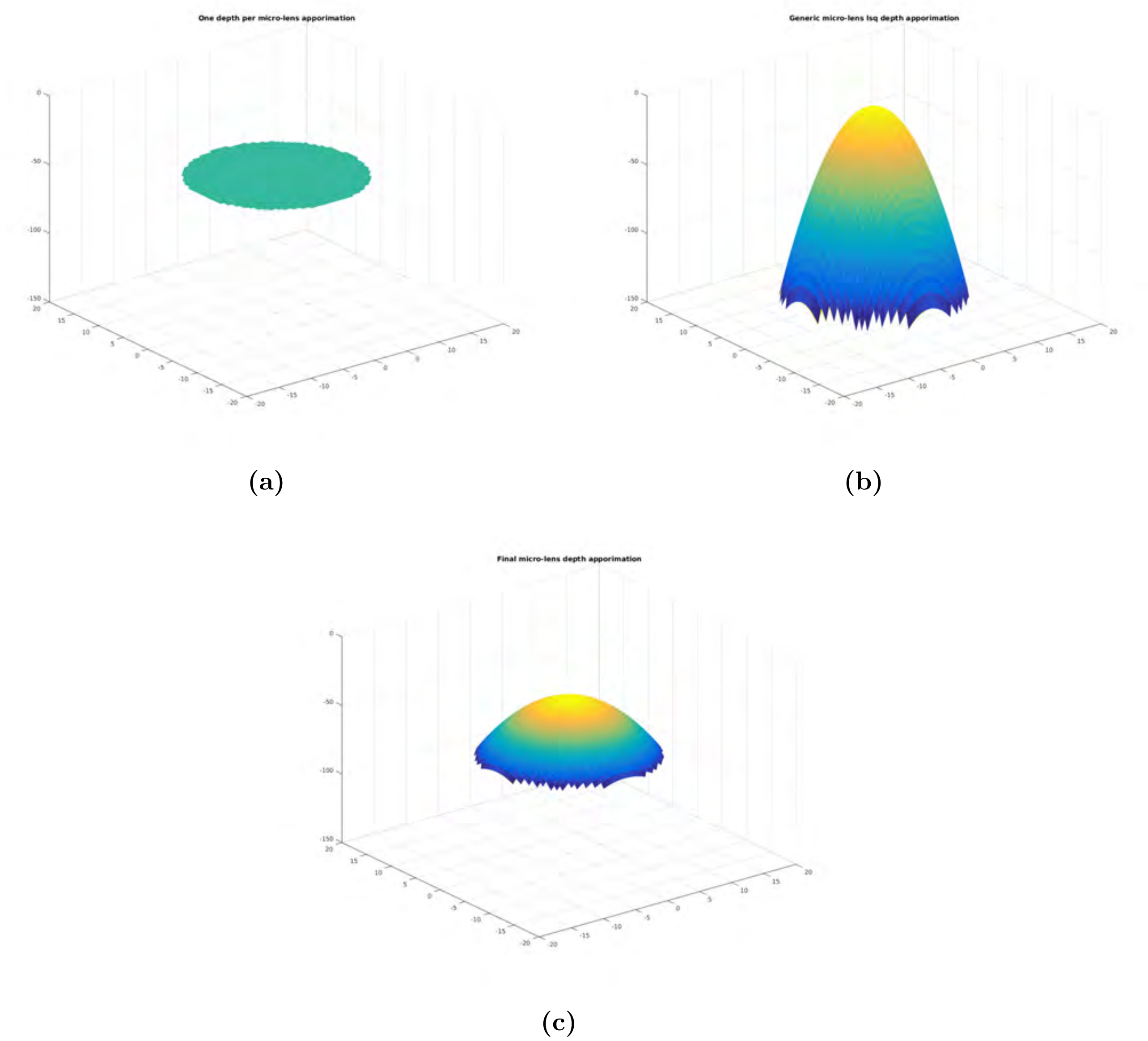
The generic polynomial is given by equation (3.6). We can write this equation in matrix notation as shown in equation (3.7). The left most matrix is called the Vandermonde matrix.

$$z = a_1 + a_2 x + a_3 y + a_4 x^2 y + a_5 x y^2 + a_6 x^2 y^2 + \dots + a_{(2p+2)} x^p y^p \tag{3.6}$$

$$\begin{bmatrix} 1 & x_1 & y_1 & x_1y_1 & x_1^2y_1 & x_1y_1^2 & x_1^2y_1^2 & \cdots x_1^py_1^p & \\ 1 & x_2 & y_2 & x_2y_2 & x_2^2y_2 & x_2y_2^2 & x_2^2y_2^2 & \cdots x_2^py_2^p & \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & y_n & x_ny_n & x_n^2y_n & x_ny_n^2 & x_n^2y_n^2 & \cdots x_n^py_n^p & \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_2p+2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{bmatrix} \Rightarrow Va = z \tag{3.7}$$

Knowing each point's x, y and z coordinates and seeking a polynomial of degree $p = 2$ we can easily determine the coefficients a by inverting the Vandermonde matrix V. By densely resampling a micro-lens, it is possible to reconstruct its surface with the obtained coefficients that describe the second order surface fitting.

The least squares approximation is a good approach when the local micro-lens point set is dense, otherwise the approximation might generate less accurate data for these less dense point sets. Generally, these badly estimated surfaces have a really fast varying slope, inaccurately estimating the micro-lens depth. Having these cases in mind we integrate the single depth per



(a)

(b)

(c)

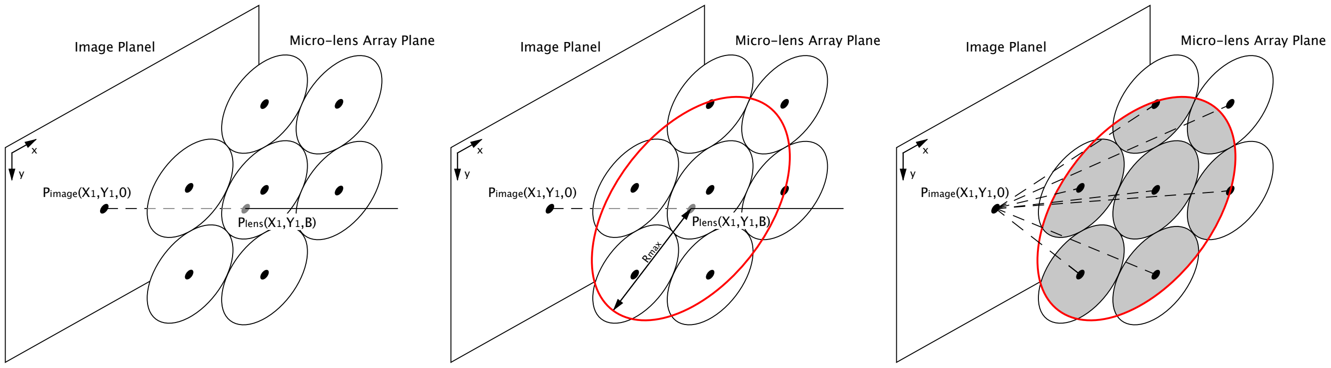
Fig. 3.9: Generic example depth estimation of a micro-lens. (a) Single depth per micro-lens estimation for the local micro-lens point set. (b) Least squares approximation surface for the local micro-lens point set. (c) Fused single depth and least squares approximation for the local micro-lens point set.

micro-lens with the multiple-depth per micro-lens generated with the least squares method. In figure 3.9 we can se a generic example of a surface estimated with a relatively small point set for a single micro-lens. After filtering the point set, we estimate both the least squares surface and the single depth for the micro-lens of section 3.4.1 (figure 3.9b and 3.9a respectively). By doing a weighted average of both we are able to attenuate the fast varying slope of the least square surface slope and correct its mean value, as shown in figure 3.9c.

This method is computationally fast and produces a good estimation for non constant depth surfaces (the general case). Since the least squares method is highly vulnerable to noise, for constant depth surfaces it might generate bad results. Even so, the noise effect on the estimated surface is attenuated by our multiple and the single depth merging.

3.5 Dense Depth Map

For the dense depth map synthesization implemented by Cunha [4] it is used the coarse depth map with one depth per lens. The creation of the dense depth map follows a group of steps. The final depth map is reconstructed on the image plane, resulting on an image with a depth value per pixel.



(a) (b) (c)

Fig. 3.10: Illustration of a few steps of the synthesization algorithm. The first step illustrated by (a) determines the central micro-lens to which point P_{image} belongs. Second step, illustrated by (b), is the R_{max} reprojection to determine which lenses project point P_{image}. Finally for the fourth step, figure (c) illustrates the projection of P_{image}'s depth value intensity into the image plane corresponding to averaging of the lenses depth value within the R_{max} radius.

The method can be described by the following steps:

- Step 1 - **Determine the central lens.** We back project an image point P_{image} onto the micro-lens array plane and determine to which lens it belongs. That lens will be the central lens. This is illustrated in figure 3.10a.

- Step 2 - **Determine which lenses belong to the radius** R_{max} given by equation (3.3), that determines which neighbor lenses project point P_{image} (illustrated in figure 3.10b). R_{max} is calculated using the depth value of the central micro-lens.

- Step 3 - **Estimate P_{image}'s depth value by averaging the depth values of all the lenses contained within** R_{max}. This is illustrated in figure 3.10c.

Every image point is processed by these three steps and the depth value attributed to each is the average of all projected points value.

To improve the dense depth map synthesization we altered step 3 of the algorithm. Similarly to the fine filter of the outlier removal, we calculate the median and standard deviation for all the lenses within R_{max} radius. The final P_{image} depth value is the average of the depth values of all lenses within the R_{max} radius that follow equation (3.2). With this filtering we can achieve a more accurate and sharp dense depth map, as will be shown in section 4.1.3.

3.6 Review of Fleschmann and Koch Disparity Map

Since Raytrix's results come from a closed algorithm, we can't directly compare algorithms. Thus we replicated Fleischmann and Koch [7] for a direct comparison, which is actually the state of the art for multi-focus plenoptic cameras. This will give us better perception of the robustness of our algorithm and a better view for future work.

Their algorithm is rather complex and is based on ”image-space multi-view stereo depth estimation algorithms which estimates a depth map *per view*” [7], generating a disparity map per micro-lens image. A cost volume is calculated from the sum of absolute differences (SAD) for different disparity values and this cost volume is minimized and thus extracted the desired disparity for each pixel on a lens region. Before extracting the disparity map, a fine regularization is performed followed by a coarse regularization. These regularizations increase the fine and global consistency of the results.

In this section we will explain the algorithm of Fleischmann and Koch [7] for a better understanding of the differences between their approach and ours.

3.6.1 Disparity estimation

Given a main lens a and n target lenses $a_1, \dots, a_n$ with respective centers c_a and $c_{a_1}, \dots, c_{a_n}$ and radius r, they estimate the virtual depth for each pixel x within the main lens. Following the classic stereo paradigm, for each lens pair $(a, a_i), i \in \{1, \dots, n\}$ they measure the photometric similarity of a local neighborhood $\Omega(x)$ in a reference image I_a (pixels contained in a target micro-lens) and a local neighborhood $\Omega(x - d_j\upsilon)$ in the target image I_{a_i}, where the epipolar line for a point x in the reference image is $L_i = \{x + t\upsilon : t \in\}$ with $\upsilon = (c_{a_i} - c_a)/2r$. A range of disparities d are tested where $d_j \in [0, d_{max}], d_{max} < 2r$ and δr denotes a lens border rejection in pixels. The sum of absolute differences is calculated as equation 3.8.

$$SAD(x, d_j; a, a_i) = \frac{1}{A(x, \upsilon, d_j)} \sum_{u \in \Omega(x)} |I_a(u) - I_{a_i}(u - d_j\upsilon)| \mathbb{1}(u - d_j\upsilon) \tag{3.8}$$

with

$$A(x, \upsilon, d_j) = \sum_{u \in \Omega(x)} \mathbb{1}(u - d_j\upsilon) \qquad \mathbb{1}(x) = \begin{cases} 1 & \text{if } \|x\| < r - \delta r \\ 0 & \text{else} \end{cases}$$

Equation (3.8) is a similarity measure among the chosen disparity $d_j \in D$. It gives the cost volume for each point of the reference lens image I_a relative to the target image I_{a_i} for all chosen disparities along the epipolar line on each neighbor target lenses a_i. If the target point $x - d_j$ falls outside of the image radius, a maximum cost volume C_{max} is attributed. Equation (3.9) gives this calculated cost volume.

$$C_i(x, d_j; a) = \begin{cases} SAD(x, d_j; a, a_i) & \text{if } \|x - d_j\upsilon\| \leqslant r - \delta r \\ C_{max} & \text{else} \end{cases} \tag{3.9}$$

3.6.2 Fine Disparity Map

Having obtained the n cost volumes $C_1, \dots, C_n$, the method average them into a single fine grained cost volume, given by equation (3.10), which will serve as a normalization constant respectively to the number of views that see the scene corresponding point x for a given disparity

d_j.

$$C^f(x, d_j; a) = \begin{cases} C_{max} & \text{if } \forall i : C_i(x, d_j) = C_{max} \\ \frac{1}{A^f(x,d_j)} \sum_i^n C_i(x, d_j) \mathbb{1}^f(C_i(x, d_j)) & \text{else} \end{cases} \tag{3.10}$$

$$A^f(x, d_j) = \sum_{i=1}^n \mathbb{1}^f(C_i(x, d_j; a)) \qquad \mathbb{1}^f(x) = \begin{cases} 1 & \text{if } x < C_{max} \\ 0 & \text{else} \end{cases}$$

Then the fine grained cost volume per lens are regularized using a semi-global strategy for each single lens, denoted as a in equation (3.11). The choice of the semi-global strategy is mainly due to its speed and simplicity.

$$C^f_{ref}(x, d_j; a) = \sum_{\omega \in W^f} C^f_\omega(x, d_j; a) \tag{3.11}$$

$$\begin{aligned} C^f_\omega(x, d_j; a) = c^f(x, d_j; a) + min\{ & C^f(x - \omega, d_j; a), C^f(x - \omega, d_{j+1}; a) + p^f_1, \\ & C^f(x - \omega, d_{j-1}; a) + p^f_1, \\ & \min_d C^d(x - \omega, d; a) + p^f_2 \} \end{aligned} \tag{3.12}$$

where W_f are the directional deviations induced by the chosen pixel neighborhood, p^f_1 is a constant penalty for deviations of one disparity and p^f_2 a constant penalty for deviations of more than one disparity.

By minimizing the regularized fine grained cost volume for each pixel of each micro-lens we extract the desired disparity for the fine grained disparity map, given by equation (3.13)

$$\hat{d}^f(x; a) = \operatorname*{argmin}_d C^f_{reg}(x, d; a) \tag{3.13}$$

3.6.3 Coarse Disparity Map

An optimal coarse regularization is used. For scenes with insufficient texture, a per lens fine grained regularization might not be enough to obtain a dense depth map for the complete micro-lens grid. The coarse cost volume for a single cost per micro-lens is calculated and regularized with the same semi-global strategy of the fine grained cost volume of equation (3.11).

First, the fine cost volume $C_f(x, d; a)$ is averaged, resulting in a single coarse cost slice per micro-lens. This averaging follows equation (3.14)

$$C^c(a, d_j) = \frac{1}{A^c(a, d_j)} \sum_{x \in dom(I_a)} C^f(x, d_j; a) \mathbb{1}^c(C^f(x, d_j; a)) \tag{3.14}$$

$$A^c(x, d_j) = \sum_{x \in dom(I_a)} \mathbb{1}^c(C_f(x, d_j; a)) \qquad \mathbb{1}^c(x) = \begin{cases} 1 & \text{if } x < C_{max} \\ 0 & \text{else} \end{cases}$$

Only then, as explained before, the averaged coarse cost volume is regularized using the directions W^c representing the chosen neighbor micro-lenses instead of neighbor pixels, with respect to the micro-lens hexagonal configuration and coarse penalizing cost constants p_1^c and p_c^c. For the final coarse disparity map, similar to the fine disparity map, the cost volume is minimized for every disparity deviation on each micro-lens, following equation (3.15)

$$\hat{d}^c(a) = \operatorname*{argmin}_d C_{reg}^c(a, d) \tag{3.15}$$

3.6.4 Fusion of Fine and Coarse Disparity Map

On their work, Fleischmann and Koch [7] merge the coarse and fine disparity maps for the final disparity map. This merging is weighted by a constant factor λ, affecting the overall influence of the coarse estimation, and a standard deviation exponential factor, to reduce the influence of coarse estimation for micro-lenses with more structure.

$$C^{c,f}(x, d_j; a) = C^f(x, d_j; a) + \lambda |\hat{d}^c(a) - d_j| e^{(-\sigma(I_a)^2/\sigma_{struct}^2)} \tag{3.16}$$

Equation (3.16) give the merged fine and coarse cost volume, being $\sigma(I_a)$ the standard deviation of I_a. The constant σ_{struct} controls how fast the influence of the coarse estimation will decay based on its structure. The exponential weighting will tend to zero if the micro-lens has considerable structure, meaning the image content allows a good disparity estimation (has enough texture). If the micro-lens has little structure, the coarse estimation will have a bigger weighting.

Afterwards the cost volume $C^{c,f}$ is regularized similarly to equation (3.11), being the final disparity map obtained from $C_{reg}^{c,f}$ acording to equation (3.13).

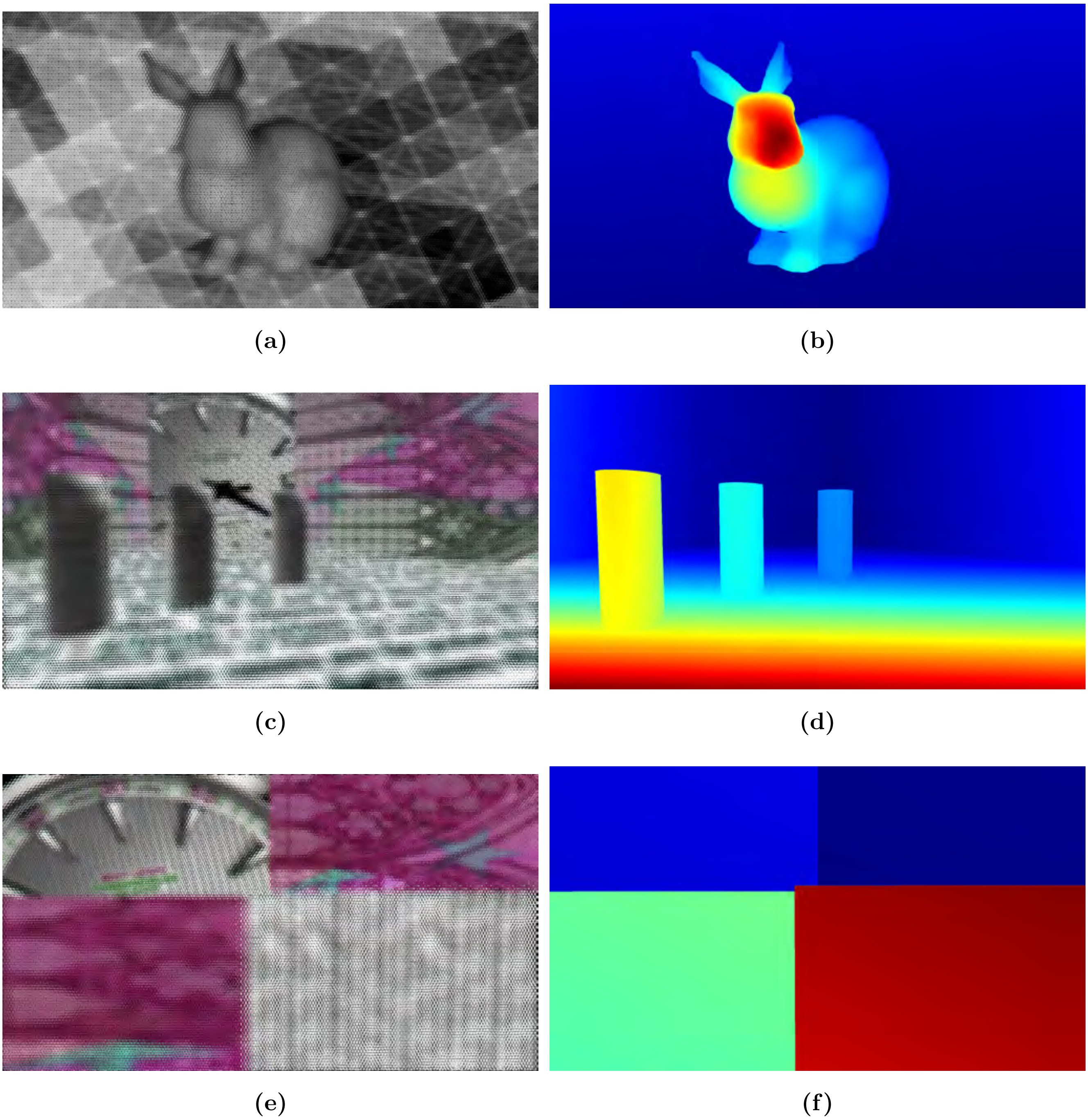
Chapter 4

Experiments and Results

In this chapter we present the obtained results from our algorithm. First we present our results for the approached reconstruction of the coarse depth maps and the ones obtained through the replication of Fleischmann and Koch's algorithm with simulated data. We perform a direct comparison of both algorithms results (ours and Fleischmann and Koch's). Then, we compare our dense depth map estimation with Cunha's [4] and Raytrix's estimation for simulated and real data.

4.1 Synthetic Datasets Results

For the synthetic data, we have three plenoptic datasets, which are computer generated. These datasets where produced by Cunha [4]. We have the "Bunny" dataset, "Bolt" dataset and "4planes" dataset (figure 4.1). The "Bunny" dataset contains the Stanford Bunny with a background plane. This dataset is a good representation of a silhouette with high detail. The "Bolt" dataset is a replica of Raytrix's "Watch" dataset and contains three bolts with four background planes. The "4planes" dataset is a simple four plane representation where each plane has one depth, different from each other. These datasets contain the plenoptic image, calibration data and depth ground truth for the captured scene. With the real depth values provided by the ground truth we can measure the error of the estimation.

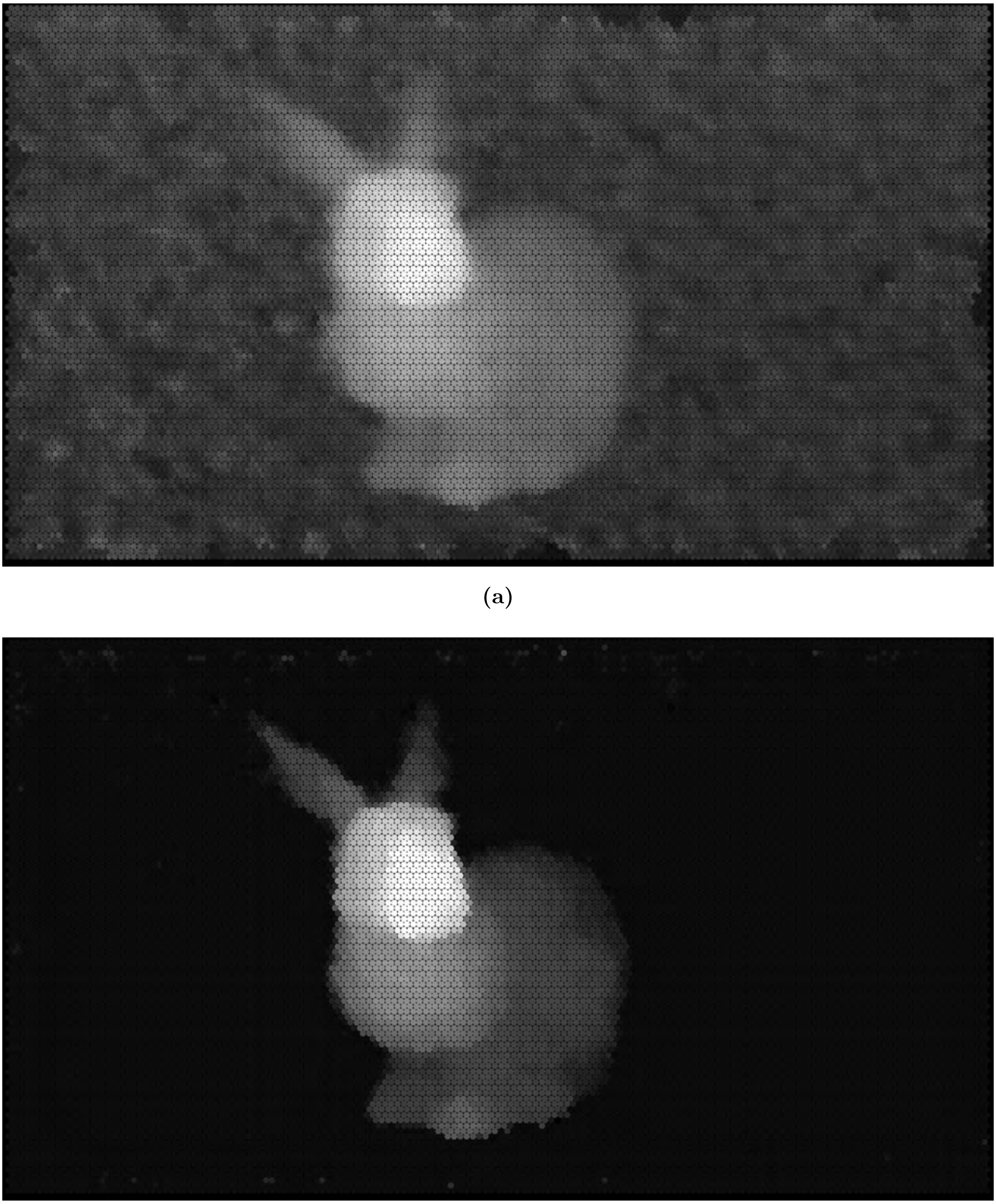


(a) (b) (c) (d) (e) (f)

Fig. 4.1: Plenoptic images and depth ground truth for the synthetic data. (a) and (b) ”Bunny” dataset. (c) and (d) ”Bolt” dataset. (e) and (f) ”4planes” dataset.

4.1.1 Coarse depth map

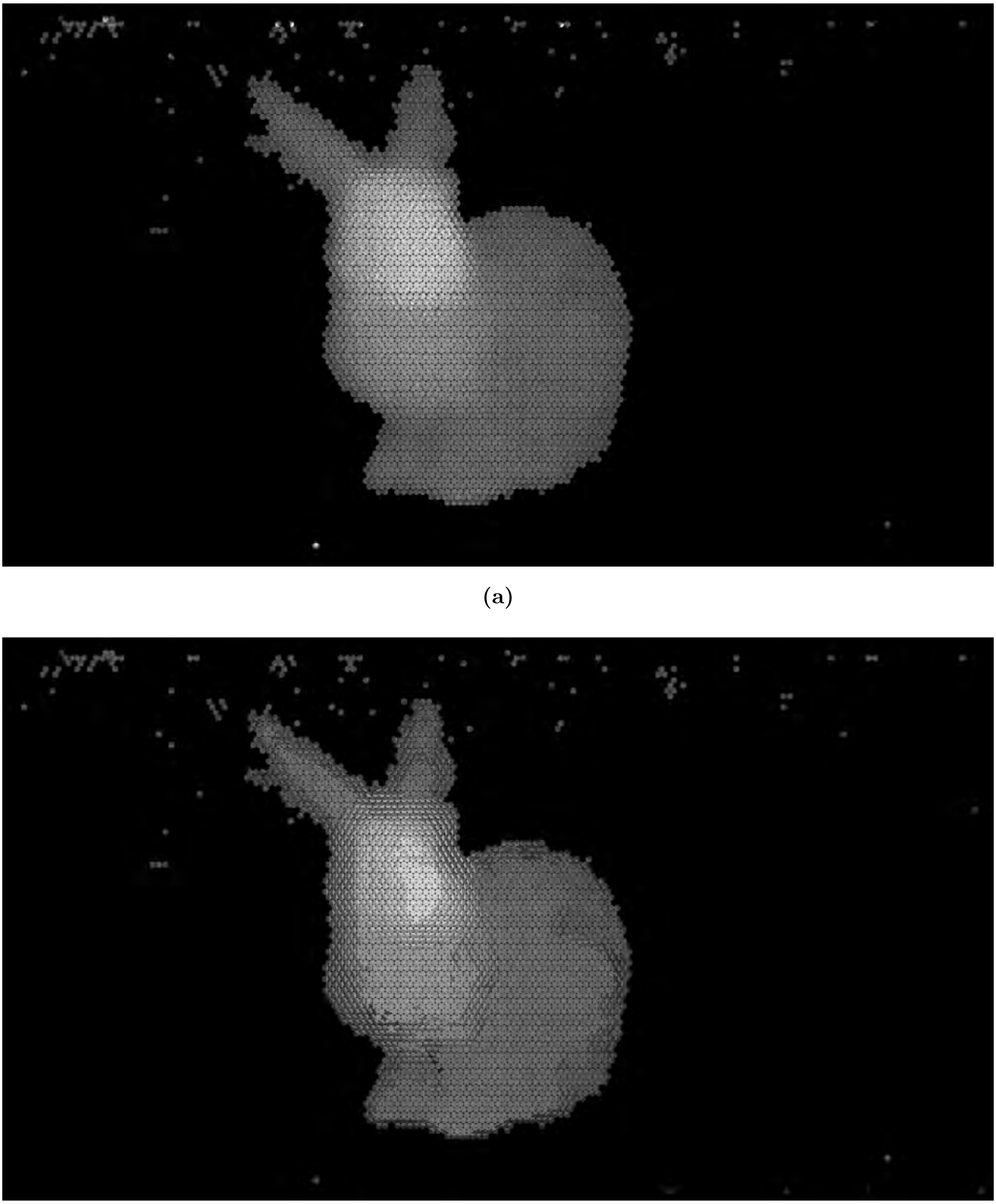
For each synthetic dataset we reconstructed the three coarse depth maps studied on this thesis. In figures 4.2 and 4.3 we can see our results for the three coarse maps for the ”Bunny” dataset and Cunha’s coarse estimation.



(a)

(b)

Fig. 4.2: Coarse depth map for ”Bunny” dataset. (a) Cunha’s one depth for each micro-lens, (b) our one depth for each micro-lens.



(a)

(b)

Fig. 4.3: Coarse depth map for "Bunny" dataset. (a) our two depths for each micro-lens, (b) our multiple depth for each micro-lens.

On our coarse map of one depth for each micro-lens (figure 4.2b) we can see a clear overall improvement compared to Cunha's estimation for the same map (figure 4.2a) compared to the depth ground truth (figure 4.1b). The depth refining algorithm identified and corrected the blurred background on our results. We can see that in Cunha's estimation, this blurred area produced substantial noise.

For a better understanding of the effects of the two developed coarse maps (two and multiple depths for each micro-lens), in figure 4.4 we can see a close up of both maps for the "4planes" dataset.

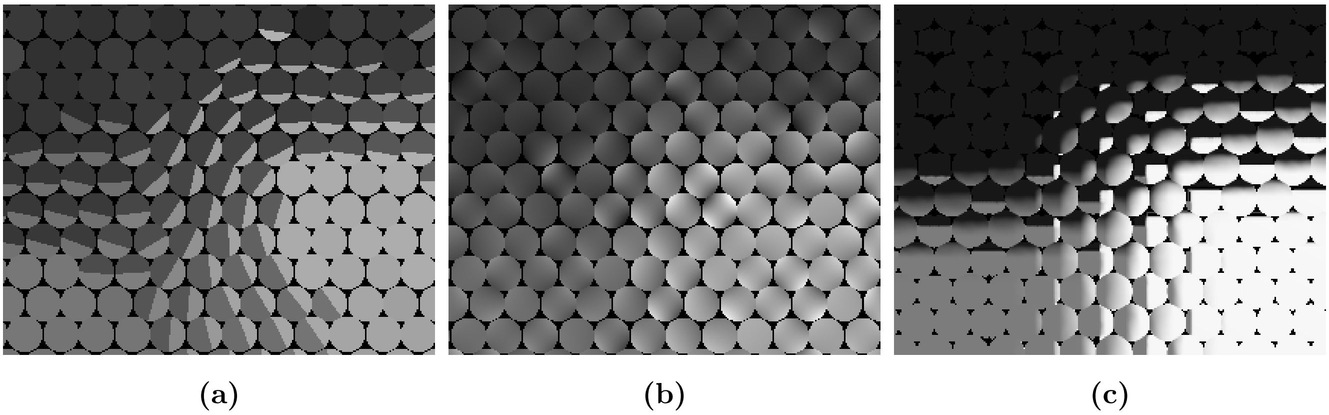


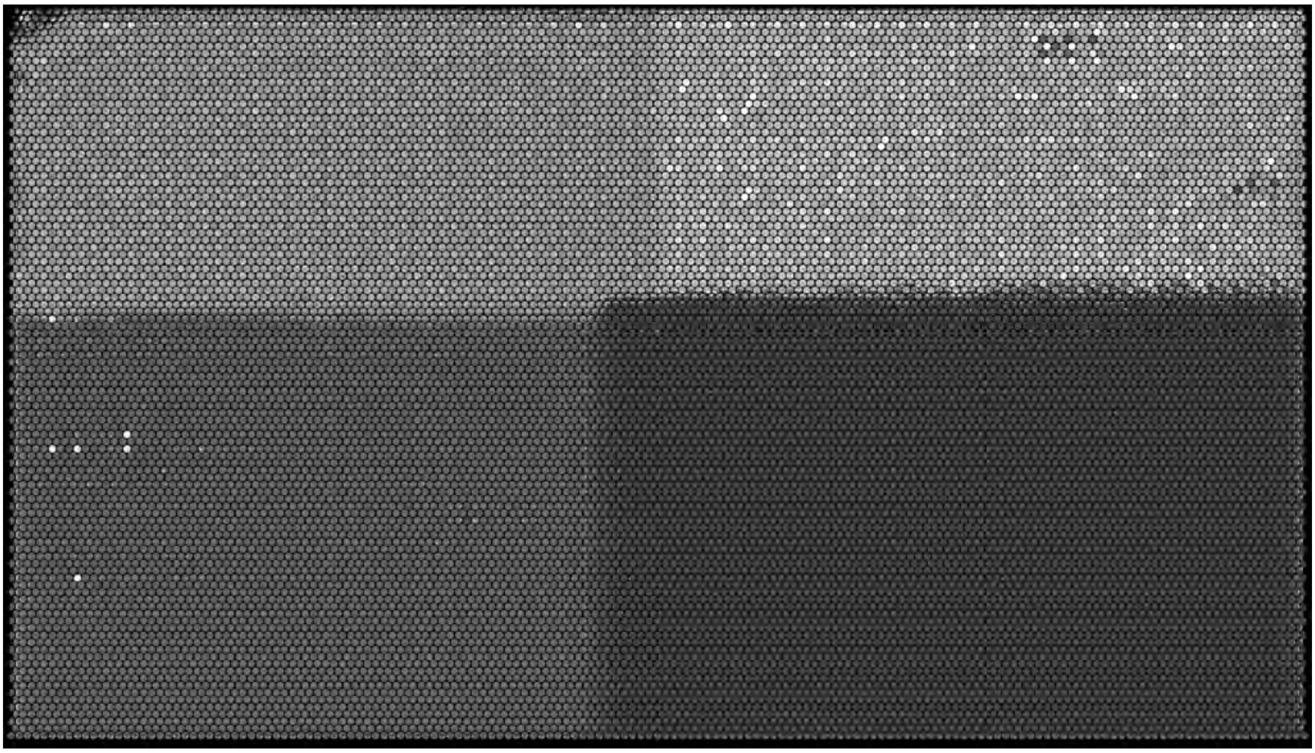
Fig. 4.4: Close up of the same section for the "4plane" dataset coarse estimation for: (a) two depths for each micro-lens, (b) multiple depths for each micro-lens, (c) depth ground truth projected through the micro-lenses.

As for the coarse maps with two and multiple depths for each micro-lens (figure 4.3), it is unclear if there are improvements only by visual inspection. Notice that the results are presented in gray scale images, meaning that the depth values were normalized before being converted into gray scale. If there is a high (or low) value outlier, the 8-bit color values will have to represent all map values, thus resizing its scale. In section 4.1.2 we conduct an error measurement for all obtained results, comparing the disparity error from our results with the disparity error from Cunha and Fleischmann and Koch results.

4.1.2 Disparity map error comparison

For the replication of Fleischmann and Koch [7] we had to estimate some parameter for the disparity calculation. We used the following algorithmic parameters: $p_1^f = p_1^c = 0.01$, $p_2^f = p_f^c = 0.03$, $\lambda = 2$, the lens border $\delta = 1px$, the local neighborhood Ω was chosen as a $3 * 3$ pixel neighborhood and $D = \{\triangle d, \ldots, k\triangle d\}, \triangle d = 1/4, k = r/\triangle d$. We used a fixed lens neighborhood strategy, using the complete R_0, R_1 and R_5 rings.

We can see the disparity map for the "4planes" dataset produced with Fleischmann and Koch in figure 4.5. As stated before, this algorithm is based on photometric similarities, depending on textures for the disparity estimation. Pattern textures or lack of texture might produce bad disparity estimations.



(a)

Fig. 4.5: Disparity map for the "4planes" dataset.

Our estimations are for depth values, so we have to convert our estimated depth maps into disparity maps. We achieve this with equation (4.1), where d is the disparity value, f_a is the focal length of the micro-lens, z the virtual depth value and $2r$ the lens aperture.

$$d = \frac{f_a 2r}{z} \tag{4.1}$$

We perform error measurements for all the obtained coarse estimations. For the error metrics we calculated the mean absolute error (MAE) (equation 4.2), where y_i is the measured value and $\hat{y_i}$ is the real value.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.2}$$

Table 4.1 shows a comparison of the calculated disparity error for all the studied methods tested on all simulated datasets. The methods we tested are (left to right on table 4.1): Fleischmann and Koch, Cunha's one depth for each micro-lens, our one depth for each micro-lens, our two depths for each micro-lens and our multiple depths for each micro-lens.

		Methods					
		Fleischmann and Koch	Cunha		Our one depth	Our two depths	Our multiple depths
Datasets	Bunny	**0.195574**	0.659667		0.469724	**0.384297**	0.388338
	Bolt	**0.174741**	0.498349		0.271392	**0.190443**	0.197552
	4planes	**0.178315**	0.352118		0.230346	**0.217686**	0.231478

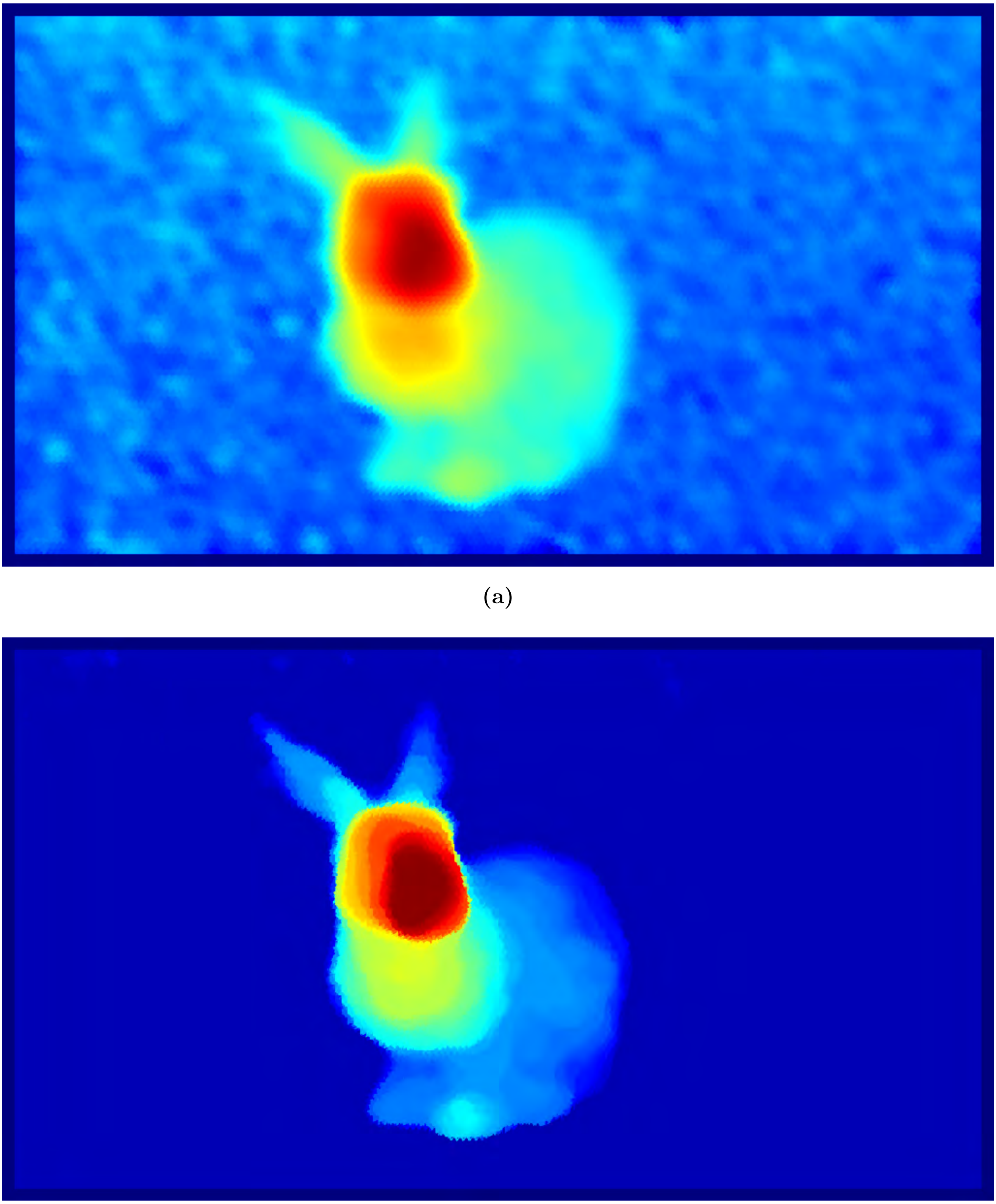
Table 4.1: Mean absolute disparity error for all studied coarse maps. Disparity error values are presented in pixels.

Our methods show a clear improvement compared to Cunha's work. Overall, both our two and multiple depths for each micro-lens maps show improvements compared to our single depth for each micro-lens approach. The coarse map with two depths for each micro-lens present the best results for the error measurements, adapting to each scene's characteristics.

Fleischmann and Koch's results are better that our results but, comparing both algorithm's computational time, their algorithm takes 122 minutes to reconstruct a disparity map for the "Bunny" dataset and our algorithm takes 37 minutes to estimate the dense depth map for the same dataset, roughly three times less then Fleischmann and Koch's algorithm (the "Bunny" dataset, from all the available datasets, is the most time consuming to process though our algorithm). Since it is a considerable computational difference, the small error difference from our method to Fleischmann and Koch is acceptable and we can conclude that our approach produces good results in less computational time (compared to Fleschmann and Koch).

4.1.3 Dense depth map

We can see in figure 4.6 Cunha's and our estimated dense depth map for the "Bunny" dataset.



(a)

(b)

Fig. 4.6: Dense depth map. (a) Cunha's estimation. (b) Our estimation.

Table 4.2 presents an error comparison for Cunha's estimation for the dense depth map and our improved estimation for the dense depth map. To measure this error we use the root mean squared error (RMSE) for the error metric. This is a robust error metric and is given by equation (4.3), where y_i is the measured value and $\hat{y}_i$ is the real value.

		Methods	
		Cunha's dense	Our dense
Datasets	Bunny	9.5739%	**3.4599%**
	Bolt	6.6127%	**2.9692%**
	4planes	5.6639%	**2.5509%**

Table 4.2: Root mean squared error comparison of Cunha's estimation and our estimation.

$$RMSE(\%) = 100 * \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2(\%)} \tag{4.3}$$

By analyzing table 4.2, we can see that there is a significant difference between the tested methods. Our method presents good improvements. From figure 4.6 we can see that the scene is sharper due to the improved method for estimating the dense depth map.

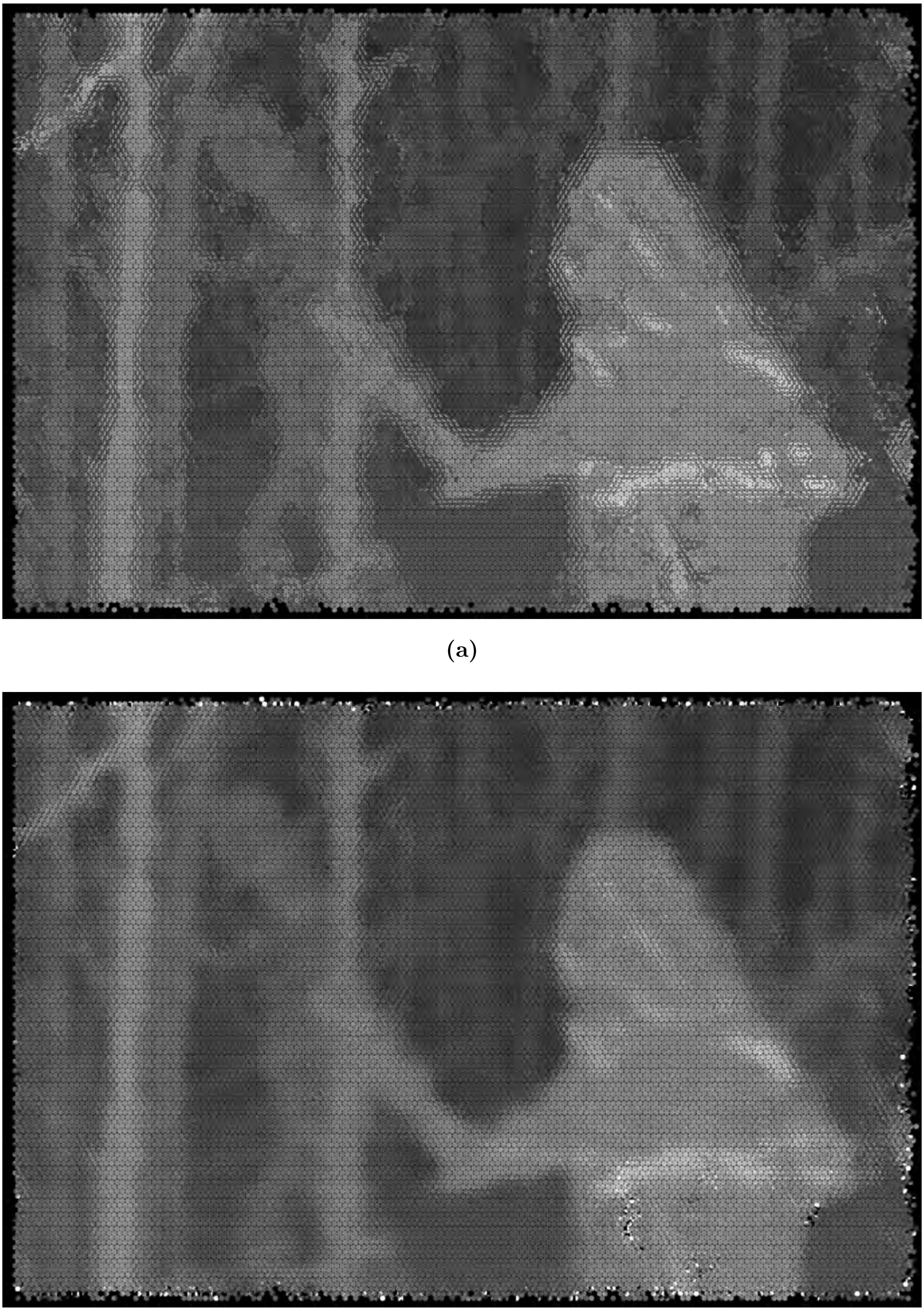
4.2 Real Images Results

As for the results with real images, first we analyze the results for the estimated coarse maps. We can see in figure 4.7 the coarse estimation with two and multiple depths for each micro-lens for Raytrix's "Forests" dataset. Both methods adapt to the complex scene and are able to produce its coarse map.

Comparing Cunha's results with our results for the coarse estimation of one depth for each micro-lens (Raytrix's "Andrea" dataset), we can see a clear improvement on behalf of our method (figures 4.8a and 4.8b). The merging of multiple depth maps improved the estimation of the scene's depth.

As for the two and multiple depths for each micro-lens, we can see in figures 4.8c, 4.8d, 4.9a and 4.9b the obtained results for Raytrix's "Andrea" and "Watch" datasets. For the dense map estimation, figure 4.10 shows both Cunha's results with our results for Raytrix's "Watch" dataset. As stated for the synthetic data, the scene is sharper due to our improved method.

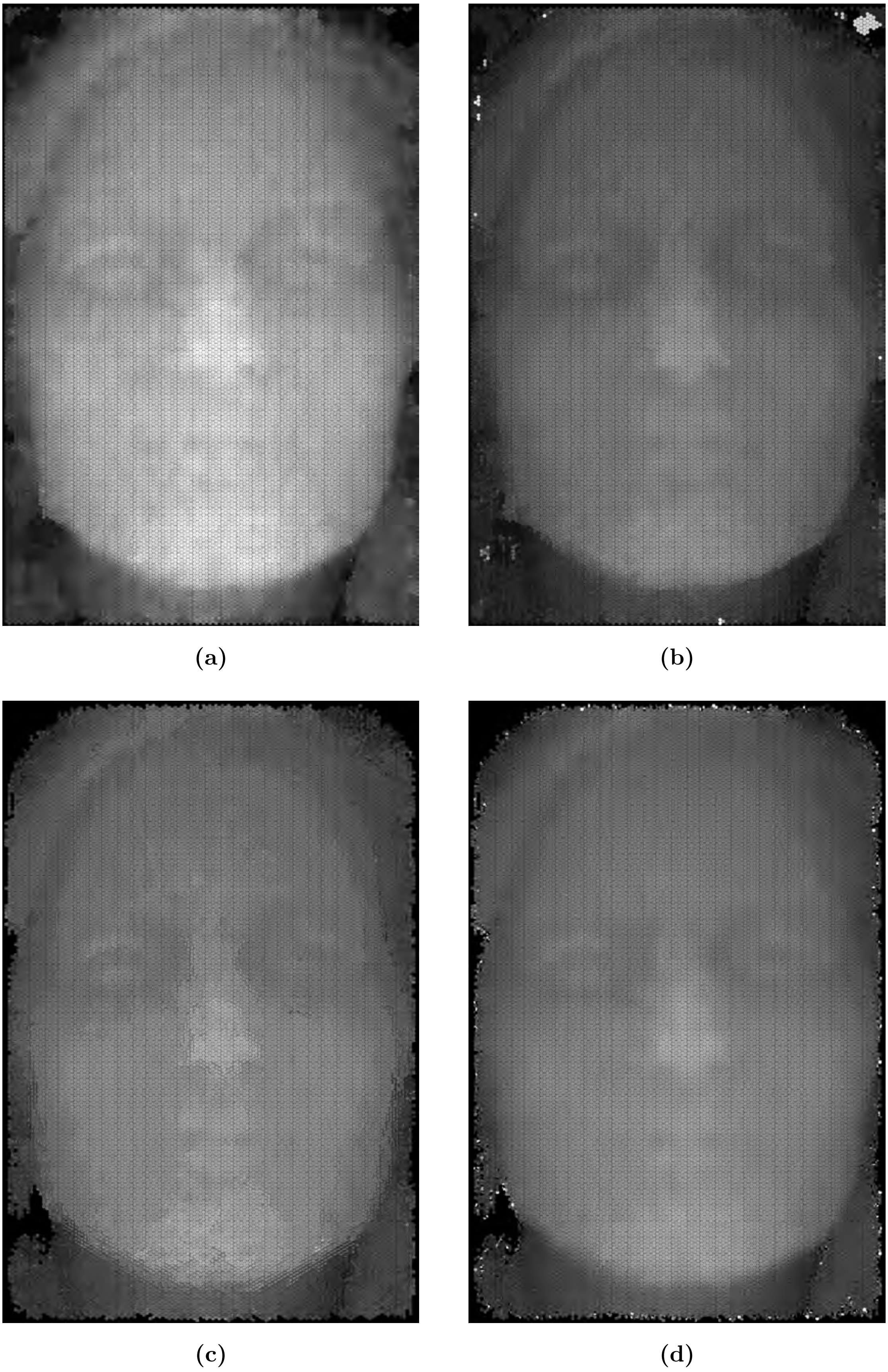
Figure 4.11 shows our and Raytrix's results for the dense depth map for Raytrix's "Andrea" dataset and Raytrix's results for the dense depth map for their "Watch" dataset.



(a)

(b)

Fig. 4.7: Coarse depth map for Raytrix's "Forest" dataset (a) our two depths for each micro-lens, (b) our multiple depths for each micro-lens.



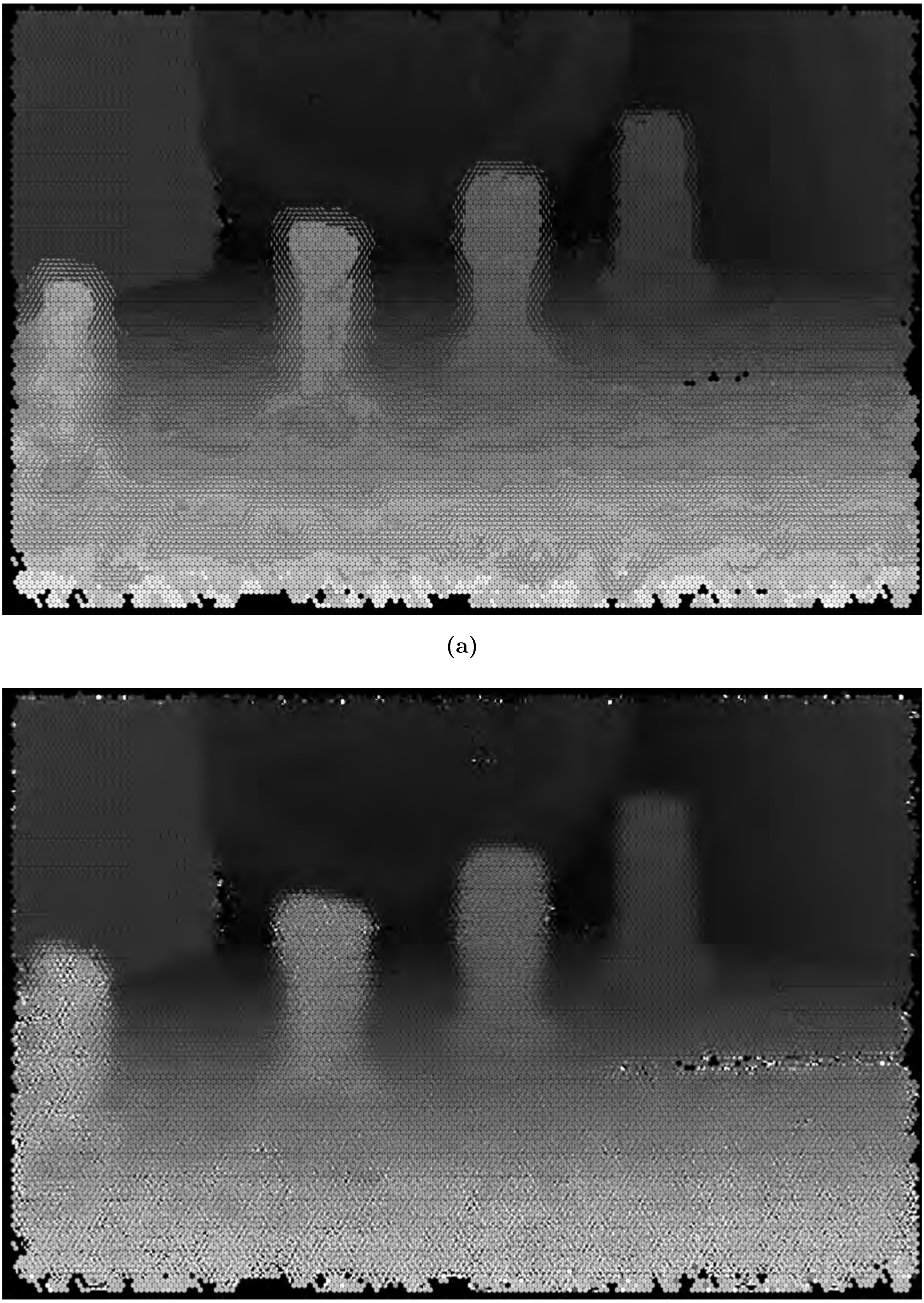
(a)

(b)

(c)

(d)

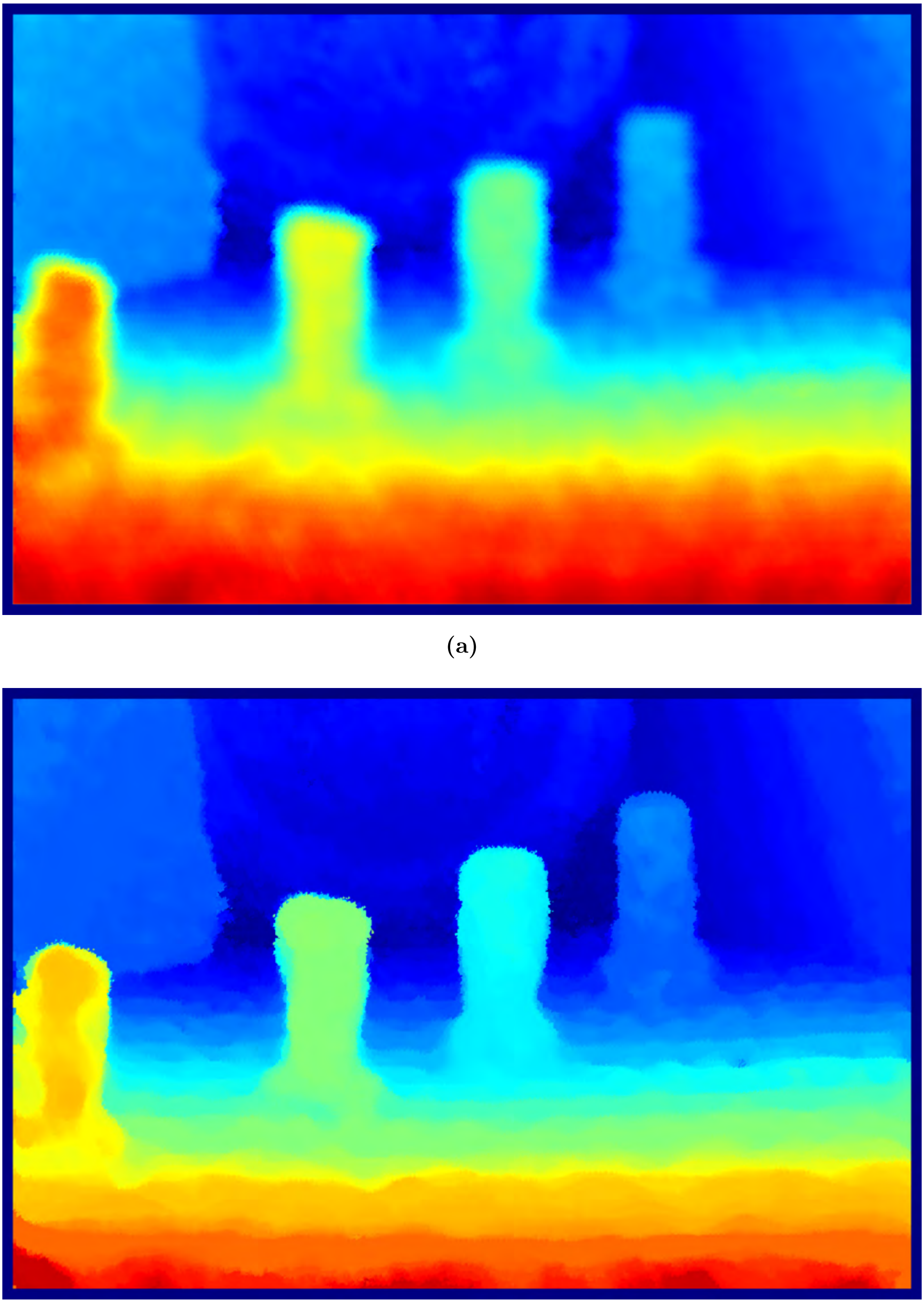
Fig. 4.8: Coarse depth map for Raytrix's "Andrea" dataset (a) Cunha's one depth for each micro-lens, (b) our one depth for each micro-lens, (c) our two depths for each micro-lens, (d) our multiple depths for each micro-lens.



(a)

(b)

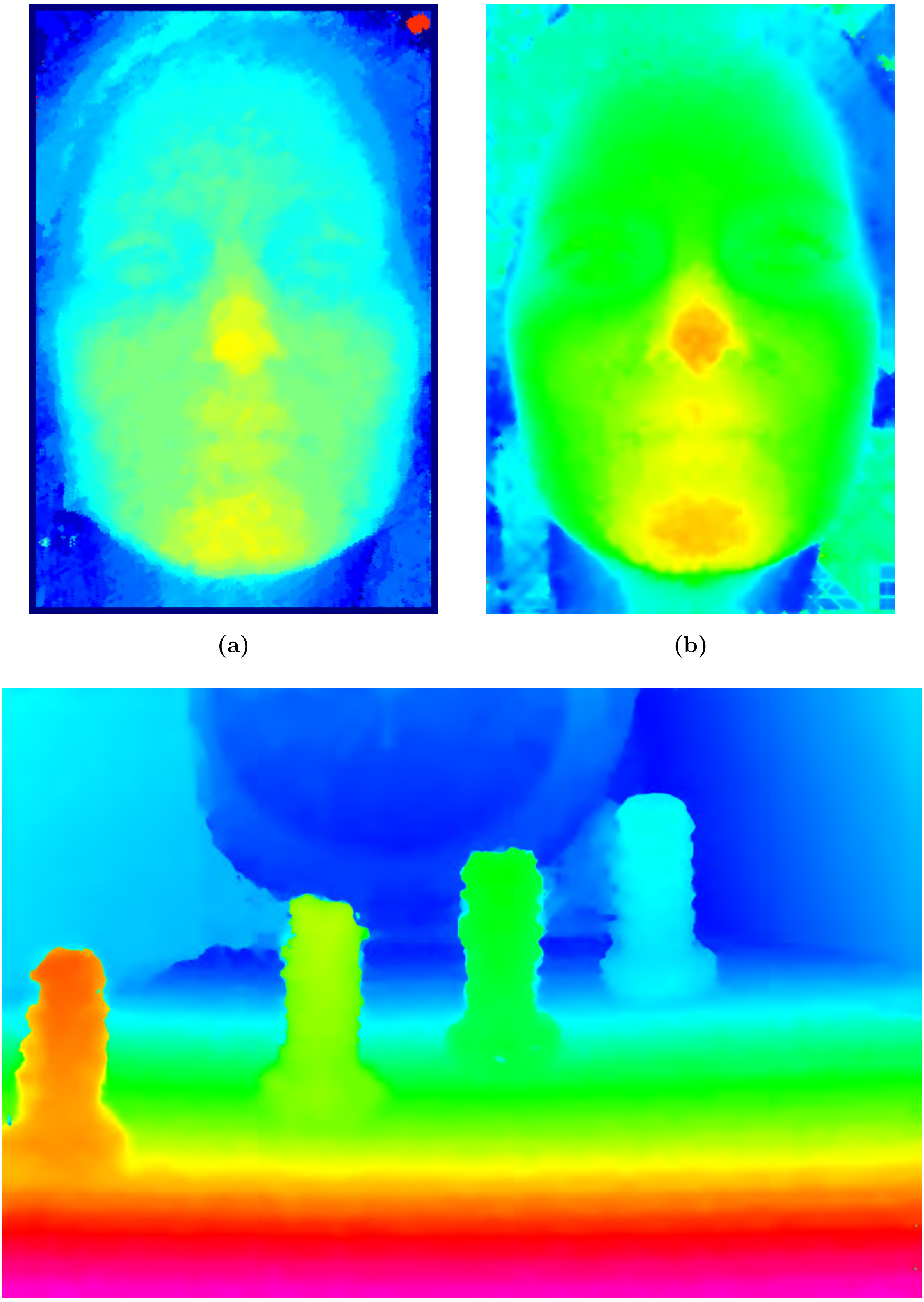
Fig. 4.9: Coarse depth map for Raytrix's "Forest" dataset (a) our two depths for each micro-lens, (b) our multiple depths for each micro-lens.



(a)

(b)

Fig. 4.10: Dense depth map for Raytrix's "Watch" dataset. (a) Cunha's depth estimation for the dense map, (b) our depth estimation for the dense map.



(a)

(b)

(c)

Fig. 4.11: Dense depth map for Raytrix's "Andrea" dataset (a) our depth estimation, (b) Raytrix's depth estimation. Dense depth map for Raytrix's "Watch" dataset (c) Raytrix's depth estimation.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

Plenoptic cameras capture a scene's light field and it is possible, through computation, to achieve the scenes' depth with the measured light field. Furthermore, with the scene's depth, a fully focused image can be rendered.

This thesis approach the depth estimation with two new methods for the coarse estimation and an improved algorithm for both coarse and dense map estimations. The algorithm is fully automatic where the only inputs needed are the plenoptic image and the camera's calibration data. This algorithm is modular, enable easy and fast future improvements.

We also replicated Fleischmann and Koch, which is an algorithm based on photometric similarities just like our algorithm. We do a direct comparison of both algorithms to better understand our algorithms performance compared to other similar algorithms.

In summary we were able to improve and achieved good results for the depth estimation. The tests performed with the available synthetic data allowed to better understand the behavior of the algorithm and test its robustness for different scenarios. Error measurements show that our method produces good results, both compared to the previous achieved results and other methods results.

5.2 Future Work

As future work, there are several improvements that can be done and new methods to be tested. Some are listed bellow.

- Improve the new methods to estimate the coarse depth map. Some parameters can be further studied and improved.

- Dense depth map for the new methods to estimate the coarse depth map. Since these coarse maps show good results, the rendering of the dense depth map can be modified for the purpose.

- Estimate the micro-lenses calibration parameters. Some parameters can be estimated, such as the focal-length of each micro-lens.

- Correct the micro-lenses distortion. With the micro-lens calibration parameters, it is possible to calculate and correct the micro-lens distortion and improve the depth estimation [11].

References

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

[2] Frank Ambrosius. Interpolation of 3d surfaces for contact modeling. Technical report, University of Twente, March 2005.

[3] Tom E Bishop, Sara Zanetti, and Paolo Favaro. Light field superresolution. *ICCP, IEEE International Conference on Computational Photography*, pages 1–9, 2009.

[4] Joel António Teixeira Cunha. Improved depth estimation algorithm using plenoptic cameras. Master's thesis, Department of Electrical and Computer Engineering, Faculty of Sciences and Technology, University of Coimbra, September 2015.

[5] João Custódio. Depth estimation using light-field cameras. Master's thesis, Department of Electrical and Computer Engineering, Faculty of Sciences and Technology, University of Coimbra, September 2014.

[6] Don Dansearau and Len Bruton. Gradient-based depth estimation from 4d light field. *International Symposium on Circuits and Systems*, 3:III – 549–52, 2004.

[7] Oliver Fleischmann and Reinhard Koch. Lens-based depth estimation for multi-focus plenoptic cameras. *36th German Conference on Pattern Recognition*, 8753:410–420, October 2014.

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

[9] Amir Hassanfiroozi, Yi-Pai Huang, Bahram Javidi, and Han-Ping D.Shieh. Hexagonal liquid crystal lens array for 3d endoscopy. *Optical Society of America (OSA)*, 2(23):971–981, 2015.

[10] Herbert E. Ives. Optical properties of lippman lenticulated sheet. *Journal of the Optical Society of America*, 21:171, 1930.

[11] O. Johannsen, C. Heinze, B. Goldluecke, and C. Perwass. On the calibration of focused plenoptic cameras. In *German Conference on Pattern Recognition Workshop on Imaging New Modalities*, volume 8200, pages 302–317, 2013.

[12] Marc Levoy and Pat Hanrahn. Light field rendering. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42, 1996.

[13] Gabriel Lippmann. Épreuves réversibles. photographies intégrals. *Comptes-Rendus Academie des Sciences*, 146:446–451, 1908.

[14] Andrew Lumsdaine and Todor Georgiev. Full resolution lightfield rendering. *Indiana University and Adobe Systems, Tech. Rep*, 2008.

[15] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. University of California Press*, pages 281–297, 1967.

[16] Ren Ng. *Digital light field photography.* PhD thesis, stanford university, 2006.

[17] Christian Perwass and Lennart Wietzke. Single lens 3d-camera with extended depht-of-field. *SPIE Human Vision and Electronic Imaging*, 2012.

[18] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[19] Sven Wanner and Bastian Goldlueck. Globally consistent depth labeling of 4d light fields. *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, pages 41–48, 2012.

[20] N. Zeller, F. Quinta, and U. Stilla. Narrow field-of-view visual odometry based on a focused plenoptic camera. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):285–292, 2015.