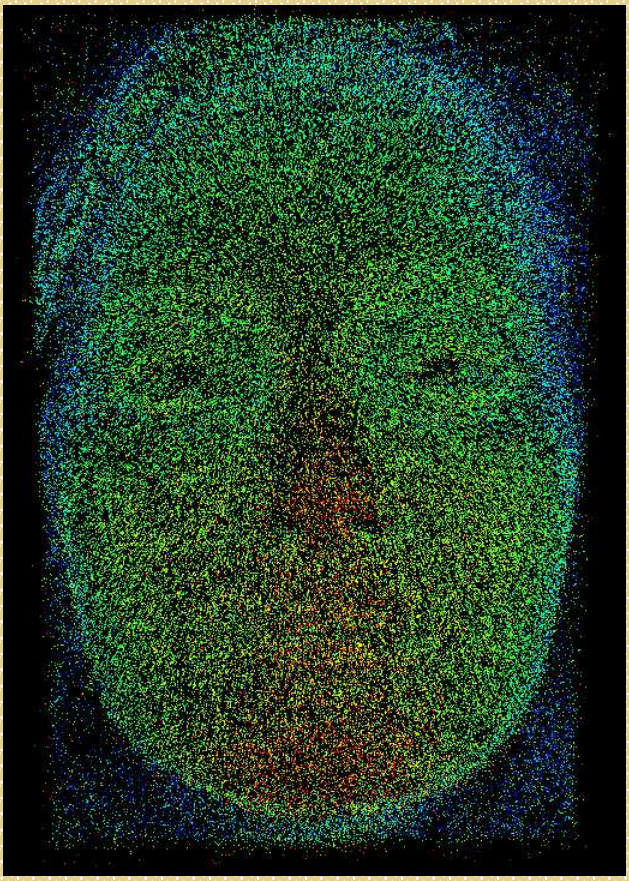


João Pedro Barata Lourenço Custódio

Depth Estimation using Light-Field Cameras

Master Dissertation

September 2014

· U C ·

Universidade de Coimbra

Departamento de Engenharia Electrotécnica e de Computadores

Faculdade de Ciências e Tecnologia

Universidade de Coimbra

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

Depth Estimation using Light-Field Cameras

João Pedro Barata Lourenço Custódio

Research Developed Under Supervision of
Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves

Jury
Prof. Doutor João Pedro de Almeida Barreto and
Prof. Doutor Nuno Miguel Mendonça da Silva Gonçalves and
Prof. Doutor Paulo José Monteiro Peixoto

September 2014

Work developed in the Institute of Systems and Robotics of the University of Coimbra.

Acknowledgments

"Our greatest glory is not in never falling, but in rising every time we fall"
(Confucius)

Quero começar por agradecer aos meus pais José e Fátima e ao meu irmão Sérgio, pelo apoio incondicional, pelo amor e por toda a ajuda ao longo destes anos, pois sem eles teria sido muito mais complicado atingir esta meta. São as pedras basilares do meu crescimento e a eles lhes devo tudo o que hoje sou. Um agradecimento também muito especial à Joana Rita, pela força e coragem que em mim despoleta, pelo amor que demonstra, pela confiança que deposita e, sobretudo, pela paciência inesgotável nas alturas menos boas.

Não posso deixar de agradecer ao meu orientador, o Professor Doutor Nuno Gonçalves. Foi também peça fundamental nesta dissertação. A constante ajuda e disponibilidade, mesmo nas alturas em que o tempo escasseava, a brilhante orientação e os seus vastos conhecimentos na área da Visão por Computador foram fatores determinantes para o sucesso deste projeto.

Um agradecimento ao Laboratório de Visão do Instituto de Sistemas e Robótica da Universidade de Coimbra, pelo excelente acolhimento, pelos momentos de lazer e pela ajuda prestada.

Um agradecimento geral aos amigos. Aos de sempre, porque permanecem e dão força, e aos que a vida me foi dando, e que certamente permanecerão. Todos eles foram também um fator chave para o meu sucesso académico. E esperando que os demais me perdoem, destaco o Tiago Dias e o João Costa. As noites em branco, as discussões que fortaleceram a amizade, a ajuda constante e o apoio que sempre surgia nas fases de maior desmotivação foram fundamentais para o alcançar desta meta.

A todos os referidos, o meu sincero obrigado.

Abstract

Plenoptic cameras or light field cameras are a recent type of imaging devices that are starting to regain some popularity. These cameras are able to acquire the plenoptic function (4D light field) and, consequently, able to output the depth of a scene, by making use of the redundancy created by the multi-view geometry, where a single 3D point is imaged several times.

Despite the attention given in the literature to standard plenoptic cameras, like Lytro, due to their simplicity and lower price, we did our work based on results obtained from a multi-focus plenoptic camera (Raytrix, in our case), due to their quality and higher resolution images.

In this master thesis, we present an automatic method to estimate the virtual depth of a scene. Since the capture is done using a multi-focus plenoptic camera, we are working with multi-view geometry and lens with different focal lengths, and we can use that to back trace the rays in order to obtain the depth.

We start by finding salient points and their respective correspondences using a scaled SAD (sum of absolute differences) method. In order to perform the referred back trace, obtaining robust results, we developed a RANSAC-like method, which we call COMSAC (Complete Sample Consensus). It is an iterative method that back trace the ligth rays in order to estimate the depth, eliminating the outliers.

Finally, and since the depth map obtained is sparse, we developed a way to make it dense, by random growing.

Since we used a publicly available dataset from Raytrix, comparisons between our results and the manufacturers' ones are also presented.

A paper was also submitted to 3DV 2014 (International Conference on 3D Vision), a conference on three-dimensional vision.

Keywords: Plenoptic cameras, light-field, Raytrix, Lytro, RANSAC, COMSAC, depth estimation, virtual depth

Resumo

Câmaras plenópticas ou câmaras de campo de luz são dispositivos que estão a voltar a ter alguma popularidade. Este tipo de câmaras são capazes de adquirir a função plenóptica (campo de luz 4D) e, consequentemente, capazes de estimar a profundidade de uma cena, usando a redundância criada pela geometria "multi-view", onde um ponto 3D é projetado na imagem diversas vezes.

Apesar de ter sido dada uma maior atenção na literatura às câmaras plenópticas standard, como são exemplo as Lytro, visto que são mais simples e com um preço muito mais interessante, o nosso trabalho foi desenvolvido tendo por base os resultados obtidos por uma câmara plenóptica "multi-focus" (uma Raytrix, no nosso caso), visto que a qualidade é superior e consegue obter imagens com maior resolução.

Nesta tese de mestrado, apresentamos um método automático para estimar a profundidade de uma cena. Visto que a captura da cena é efetuada com uma câmara plenóptica "multi-focus", estamos a trabalhar com geometria "multi-view" e com lentes que têm distâncias focais diferentes, podendo ser usado esse facto para efetuar traço de raios de forma a obter a profundidade.

Começamos por encontrar pontos salientes e as suas respetivas correspondências usando um valor escalado de SAD (soma das diferenças absolutas). Por forma a efetuar o traço de raios, obtendo resultados mais robustos, desenvolvemos um método estilo RANSAC, ao qual chamámos COMSAC (Complete Sample Consensus). É um método iterativo que efetua o traço de raios de forma a estimar a profundidade, eliminando resultados indesejados.

Finalmente, e visto que o mapa de profundidade obtido é esparso, desenvolvemos um método para o tornar denso, com recurso a um método de crescimento aleatório.

Por forma a testar os algoritmos desenvolvidos, utilizámos um conjunto de dados abertos ao público da Raytrix e apresentamos também a comparação entre os nossos resultados e os resultados obtidos pela Raytrix.

Foi também submetida um artigo para a 3DV (Conferência Internacional em Visão 3D), uma conferência em visão tri-dimensional.

Palavras-chave: Câmaras Plenópticas, campo de luz, Raytrix, Lytro, RANSAC, COMSAC, estimação de profundidade, profundidade virtual

Contents

List of Figures

Chapter 1

Introduction

Plenoptic cameras or light field cameras are a recent type of imaging devices that are starting to regain some popularity. While 2D conventional cameras project a 3D scene into a 2D image, plenoptic cameras work in a different way, since they acquire the plenoptic function, which not only contains information about the intensities of each pixel but also the amount of light that travels in every direction.

The construction of these cameras is slightly different from the standard ones. As standard cameras have only one lens, plenoptic cameras have a micro lens array between the main lens and the CCD sensor. This second level of lenses causes a single 3D point to be imaged several times, which leads to the formation of smaller images that compose the 4D light field.

Plenoptic cameras were formalized in the early 20th century, but since the resolution of the micro images are limited by the resolution of the sensors, they only started being produced in the last decade. The computational power back then was also a setback to the development of this cameras, as well as the quality of the micro lenses arrays. Nowadays, the existence of graphic cards with high processing power and the existence of higher resolution sensors surpasses the difficulties previously found.

We can distinguish two types of plenoptic cameras. The first one developed is called standard plenoptic camera (or plenoptics 1.0), being Lytro™ the main example. Then, Raytrix™ developed a different type of camera, called multi-focus plenoptic camera (or plenoptics 2.0). The main difference between them is the position of the focal plane. In the standard plenoptic camera, the focal plane is in the same place as the micro lenses array (the main lens is focused to match the micro lenses array), however, in the multi-focus camera, this does not happen. Also, Raytrix

camera has micro lenses with different focal lengths, which extends the depth-of-field (DOF) and leads to higher resolution images and better quality.

Basically, these cameras can obtain 3D information from a single shot, making them excellent devices for several applications like computer graphics, photography, microscopy, 3D shape recovery, depth map estimation and many more.

1.1 Motivation

We are facing a brutal growing on 3D content, whether it is on a consumer or commercial level. On a consumer level, entertainment (e.g. 3D televisions and 3D cinema) is one of the main examples of this growing, as the technology fascinates the users. The same happens in a commercial level, with areas like computer vision (e.g. virtual reality) working in order to develop and improve technologies able to use or create 3D content.

Projecting a 3D scene into a 2D scene results in a loss of the depth information. However, this depth information is vital to 3D reconstruction, for instance. There have been developed ways to recover the depth from a scene, from stereo camera systems to devices like Kinect, as well as plenoptic cameras. Despite the common goal, the way to achieve it differs. Kinect, on oposition to the micro lens array used by plenoptic cameras, use a time-of-flight technology by combining an IR projector with a CMOS sensor.

It is important to be able to estimate the depth from a scene, due to the numerous applications that can make use of it. It is then the main problem we try to solve on this dissertation. However, and despite being formalized in an article [14], Raytrix's algorithms to depth estimation are not publicly available, making our work harder and almost a process of reverse engineering.

1.2 Objectives

The objective of this dissertation is clear and it has to do with depth estimation. Our goal is to develop a fully-automatic algorithm able to estimate the depth of a scene captured with a multi-focus plenoptic camera and, if possible, obtain better results than those presented by the manufacturers.

1.3 State of the art

Despite being formalized in 1908 by M. G. Lippmann [11] and developed by several other authors during that time, plenoptic or light field cameras are a recent theme. Due to the lack of high computational power (at least at a reasonable cost), to low resolution image sensors and other issues, it was only in 2005 that Ng [13] built a plenoptic camera and, since then, some companies started to build and commercialize their own plenoptic cameras, like Lytro™[12, 13], Raytrix™[14] or Pelican Imaging [18]™.

We have nowadays several configurations for optical systems. On plenoptic cameras, the configuration used by Raytrix and Lytro consists of a micro lens array placed behind the main lens and in front of the CCD sensor.

However, there has been some research on this topic and there are now configurations of optical systems that use, in front of the main lens, spherical mirrors [2, 16]. Agrawal et al. [1] and Reddy et al. [15] present a similar configuration where they use, in front of the main lens, an array of refractive devices.

Plenoptic cameras have a wide range of applications. We can point depth map estimation, 3D shape recovery [3, 19, 21] and photography as the main ones. In this last one, we can detach the super resolution, the all-in-focus images [4] and the ability of performing a digital refocusing of the image after it has been taken.

Robotics or microscopy are also topics that found ways to integrate plenoptic cameras on their field of study. Robotics are starting to make use of plenoptic cameras on robotic navigation [6] and microscopy used them on the conception of a three-dimensional microscope [10].

In terms of depth estimation, several approaches have been tried. Dansearau [5], in 2004, proposed a method of depth estimation using 2D gradients applied to subsets of the light field data. The application of these gradients allowed him to found the orientation of the planes and thus, the depth of the scene. By using three color channels instead of a gray scale, he created redundancy on his depth estimations, resolved using a weighted sum based on the confidence (magnitude of the corresponding gradient vector) of each estimation. Areas where the depth cannot be estimated were filled by applying region growing. His method turned out to be effective even in the presence of occlusions.

Recently, there has been some development of this techniques, like multi-view stereo ap-

proaches with Bishop and Favaro [3], where the depth estimation is done with some calibrated images from a scene at different viewpoints. They start to extract images from the light field data, in order to have suitable data for a multiview geometry. This extraction allows them to obtain low-resolution views of the scene and, therefore, estimate their depth. However, plenoptic cameras are not immune to spatial aliasing, which can result on depth estimation errors. The method proposed by Bishop and Favaro tries to compensate the present aliasing, allowing them to recover the depth map from the multiple views extracted from the 4D light field. Their method produces good results, both on real or synthetic data.

Epipolar Plane Images (EPIs) are another approach to this depth estimation topic and can be described as 2D slices of constant angular and spatial direction in the 4D light field data. Wanner and Goldluecke [19] used the referred approach, starting to locally estimate the depth of EPI images. Then, labeling the local estimations, they integrate them on global depth maps, and, imposing spatial constraints, they are able to recover depth estimations that satisfy global visibility constraints. They claim to obtain results that surpasses the ones from traditional stereo-based methods and even the results from Raytrix, as they also test their method on Raytrix datasets.

1.4 List of main contributions

The main contribution of this work is a novel approach used to achieve a virtual depth estimation, where we can use the redundancy created by the fact that a single 3D point is imaged several times by the main lens. This redundancy allows us to backtrace the light rays in order to estimate the depth.

The main contribution is followed by a series of minor contributions that helps to reach the goal of this work, as follows:

- Finding correspondences of the image points

  Before we backtrace the light rays, we need to find correspondences of the image points, which we do using the SAD method, but scaled in a way to obtain better results.

- Implementation of a COMSAC (Complete Sample Consensus) method to perform the sparse depth map

We developed an iterative method to perform the referred backtrace, where we estimate the depth of each point. This iterative method allows us to eliminate bad results (outliers) and, consequently, produce robust results.

- Random growing of the sparse depth map

  The final contribution is the random growing of the sparse depth map, where we analyze a pixel without color information and use the intensity of neighboring pixels to fill the referred pixel, transforming the sparse depth map into a complete dense depth map.

1.5 Structure of the thesis

In chapter 2, we present an overview on light fields and plenoptic cameras, like the image formation model on a plenoptic camera or the difference between different types of these cameras, providing the essential information to better understand these subjects. Chapter 3 has all the information about the developed algorithms. We explain in detail our approach in order to obtain the depth estimation. The obtained results and their discussion are presented in chapter 4. Our work is concluded on chapter 5.

Chapter 2

Light Fields and Plenoptic Cameras

In this chapter, we present the theory about light fields and plenoptic cameras. We start with a theoretical background on light fields in order to introduce the concept of plenoptic cameras and how they work. This chapter has all the vital information needed to understand both light fields and plenoptic cameras, and the way they are related.

2.1 Light Fields

A light field can be described as a vector field that spans all the rays of light that travels in all directions for every point in space.

It was mathematically formalized as a 7D plenoptic function, described as

$$l(u, v, x, y, z, \lambda, t) \tag{2.1}$$

where x, y and z represents the coordinates of the 3D point, u and v represents the direction of the light rays, λ represents the light wave length and t the time.

It is however very hard to measure the referred plenoptic function in a 3D scene, for every locations at all times. Having that in mind, the plenoptic function was reformulated to a 4D one [8, 9], where the observer stays in free space.

In terms of representing light fields, there are several configurations. The most common one is the two-plane parameterization (see Fig. 2.1).

In this case, the 4D light field can be viewed as a set of pinhole images from different viewpoints that are parallel to an ordinary image plane. Figure 2.2 represents the geometry to this

parameterization of the 4D light field.

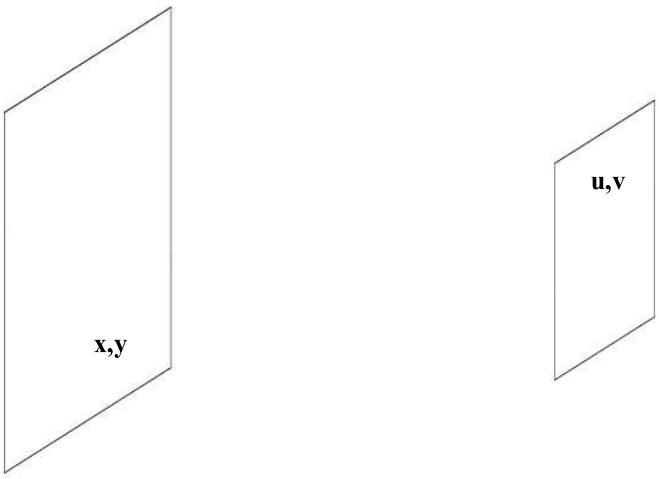


Figure 2.1: Two-plane parameterization

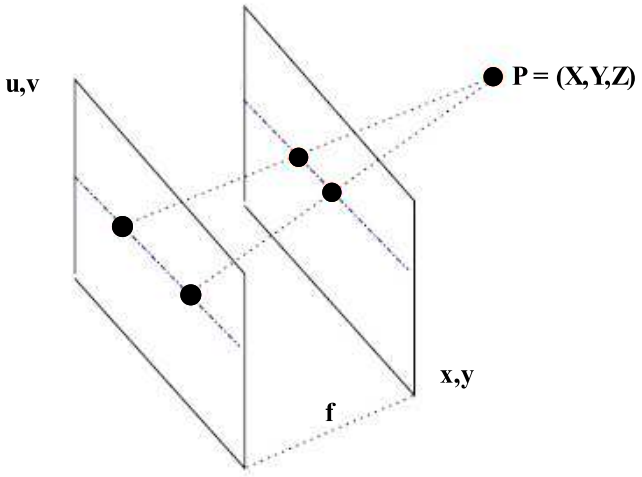


Figure 2.2: 4D geometry

2.2 Plenoptic Cameras

Over the past 100 years, it has been studied ways to acquire the light field from real world scenes. Devices like camera arrays, multi-plexing cameras and plenoptic cameras have been developed in order to perform that acquisition.

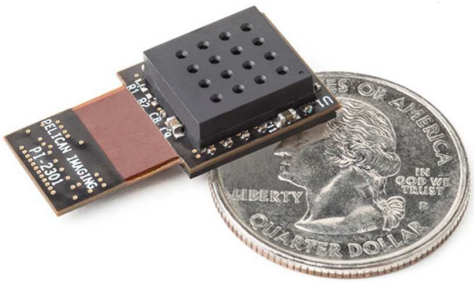
As stated before, in 1908, Lippmann [11] formalized the concept of plenoptic cameras, which was improved during time. As these cameras are one of the main topics of this dissertation, we need to understand them and how they work.

Plenoptic cameras are, as previously said, cameras able to capture the 4D light field of a scene. This capture is done by placing a micro lens array between the main lens and the image sensor. This micro lens array placement turns a standard camera into a 3D camera, with many advantages and a wide range of applications, which we are going to present later in this chapter.

This topic has been improved in the last decade and we start now to see the first cameras commercially available. Lytro™ and Raytrix™ are the most significant manufacturers, but other companies are starting to join this list, like Pelican Imaging™, who was recently bought by Nokia™, and has been developing plenoptic cameras small enough to be incorporated on smartphones. Recently this year, Lytro presented Illum, a plenoptic camera for photography enthusiasts.

(a) Raytrix R11 Camera

(b) Lytro Camera



(c) Pelican Imaging Array camera

(d) Lytro Illum

Figure 2.3: Different types of existing plenoptic cameras

Despite Lytro's popularity, mainly to its smaller price and simplicity, Raytrix cameras can achieve a higher resolution due to their particular construction and are called multi-focus plenoptic cameras or plenoptic cameras 2.0. The first ones are denoted as standard plenoptic cameras or plenoptic cameras 1.0.

In this dissertation, we give more attention to Raytrix cameras since they produce images with better quality and higher resolution. Also, we have a Raytrix camera (at the Vision Lab of Institute of Systems and Robotics of University of Coimbra) that we intend to use in the future, to test our method.

2.2.1 Standard Camera Image Formation

Before we actually start to explain the image formation on plenoptic cameras, it is important to revise the image formation on standard cameras.

On a standard camera, and ideally, the main lens generate an image of a point by gathering the whole cone of light that emanate from that point and refocusing all the light rays gathered to a single point on the image plane.

However, using lenses can lead to defocus. If the image plane (I in figure 2.4) is not at the correct focal distance, as it happens in figure 2.4, the lens will not be able to focus the entire cone of rays that passes through it, producing a blurred image of the point.

If a lens is sufficiently thin, we can calculate the distance between lens and the image plane using the thins lens equation, expressed as follows:

$$\frac{1}{a} + \frac{1}{b} = \frac{1}{f} \tag{2.2}$$

In the previous equation, a represents the distance from the "real-world" point to the lens, b is the distance between the lens and the focal plane and f denotes the lens focal length. This distances are measured parallel to the optical axis. If this equation is satisfied, we guarantee that the projected point is in focus.

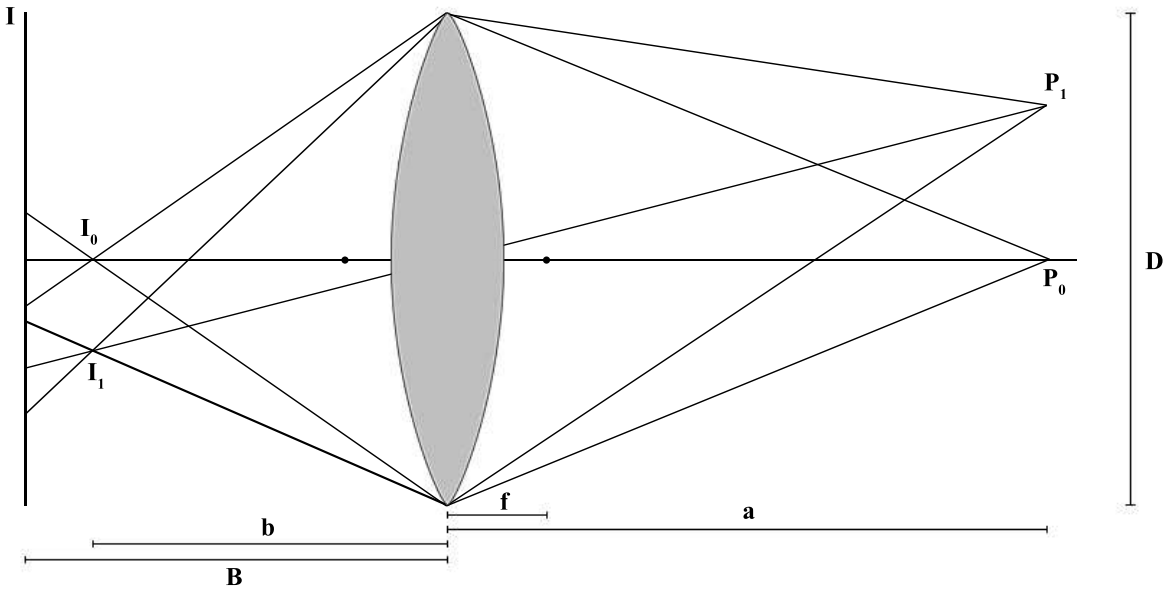


Figure 2.4: Simplified model of image formation on a standard camera

2.2.2 Multi-Focus Plenoptic Cameras

As mentioned early in this chapter, Raytrix cameras have a different construction than Lytro's ones. They still have a CCD sensor and a main lens with a micro lenses array between them, but with an improvement, since this array is composed by micro lenses with three different focal distances, extending then the depth of field (DOF). This is why they are called multi-focus plenoptic cameras.

In a light field camera, the main lens generate an intermediate image somewhere behind it. It can be a real main lens image or a virtual one, in the sense that the formation of the image happens behind the image plane. Let us denote a 3D point in the scene as ${}^{(\mathbf{3D})}\mathbf{p}$ and the virtual point projected by the main lens as ${}^{(\nu)}\mathbf{p}$. We can assume, without loss of generality, and for a simple model, that the virtual point is projected between the main lens and the micro lens array. Perwaß and Wietzke [14] mentioned that the main lens focal plane depends on the main lens itself and its focal properties. For virtual images formed behind the CCD sensor, and accordingly to authors, the only consequence is the negative value of virtual depth.

As we are able to see in figure 2.5, the exact placement of the optical devices is important to understand the image formation. So, the system parameters are:

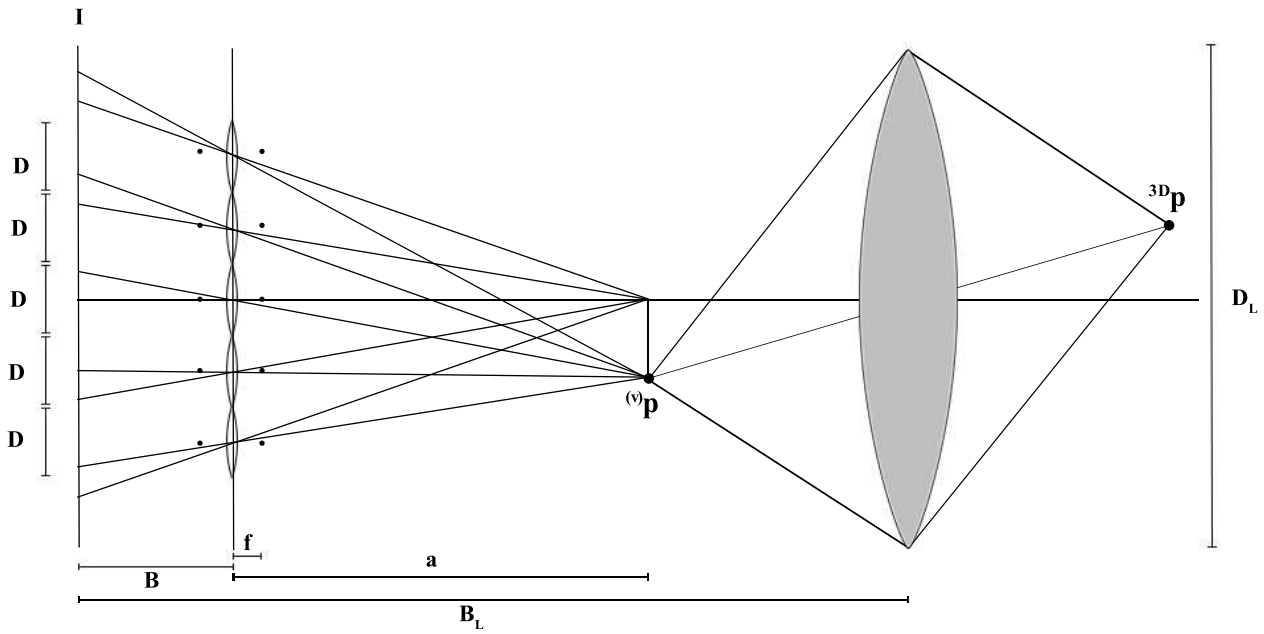


Figure 2.5: Image formation on a plenoptic camera

- D_L - aperture of the main lens
- D - aperture of the micro lenses
- B_L - distance from the main lens and the image plane
- B - distance from the array of micro lenses to the image plane (CCD sensor)
- f - focal length of the micro lenses

We can establish a relation between the "real world" point and the virtual one (projected by the main lens), which is given by the thin lens equation (2.2). However, as this relation is straightforward, our focus is on estimating the virtual point and hence in developing a computational method to estimate the virtual depth.

In order to design the relation between the main lens and the micro lenses (in terms of optical properties), we need to first define the f-number. The f-number of an optical system is thus the ratio of the focal length of a lens to the diameter of that same lens (it should be noticed that this diameter is variable and it is called, in photography, aperture).

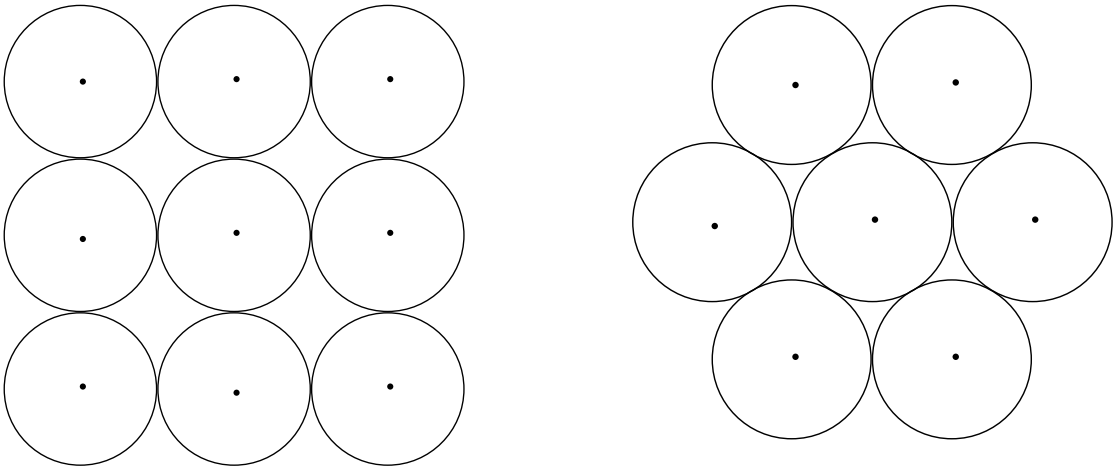
So, in order to use the image sensor as much as possible, the f-number of micro lenses should match the f-number of the main lens to avoid that micro image overlaps or the existence of gaps between neighboring images [14].

So, this f-number matching can be expressed as:

$$\frac{B}{D} \approx \frac{B_L}{D_L} \tag{2.3}$$

This relation implies that the design of the main lens is fixed by the design of micro lenses. It also means that the micro images will touch each other and not overlap, which is what we want.

Since we have a two-dimensional array of micro lenses, its design needs to be defined as the micro lenses can be placed in an orthogonal or hexagonal grid. As we can see in Figure 2.6, the hexagonal grid is the one that allows to achieve the maximum usage of the sensor, since it minimizes the gaps between the micro lenses. Thus, Raytrix chooses the hexagonal grid in the construction of their cameras.



(a) Orthogonal grid (b) Hexagonal grid

Figure 2.6: Micro lens projection cones in a 2D configuration

The location of the micro lens array will also determine the number of micro lenses that will project a virtual point onto the image plane. Perwass and Wietzke [14] showed that, when the image plane dists B from the micro lenses array, every virtual point is projected at least one time to the image. This plane is therefore called TCP (Total Covering Plane). Similarly, when

this distance increase to $2B$, every virtual point is imaged by at least two micro lenses, in which case the plane is called Double Covering Plane. Since depth estimation is only possible at that double covering plane (or at higher distances), the complete design of a plenoptic camera places the micro lens array at exactly $2B$ distance from the CCD sensor.

We now have a multi-focus plenoptic camera fully designed and configured. So, the projection of a virtual point ${}^{(\nu)}\mathbf{p}$ to its image points ${}^{(ik)}\mathbf{p}$ is made by all three types of micro lens, each type with different depth of field, and focusing different planes.

2.2.3 Virtual Depth

Since our main goal is to estimate the virtual depth of a scene, we need to formally define this concept. Like Perwaß and Wietzke stated in [14], virtual depth is the ratio of the distance between the micro lens array and the virtual point (a in figure 2.5) and the distance between the micro lens array and the image plane (B).

We thus have $\nu = a/B$.

For a better understanding of this concept, virtual depth is the number of B times that the virtual point is far from the micro lens array. Keep in mind that the double covering plane, mentioned in the previous section, is the locus of points with $\nu = 2$, which means that every point in that plane is imaged at least by two micro lenses.

Thus, points with virtual depth equal to 3 are imaged at least by 3 micro lenses and so forth.

2.2.4 Differences Between Standard and Multi-Focus Plenoptic Cameras

As stated before in this dissertation, there are standard plenoptic cameras and there are multi-focus plenoptic cameras, being Lytro and Raytrix the main examples, respectively.

On a standard plenoptic camera, like the one proposed by Ng et al. [13], the distance between the micro lens array and the image plane equals the focal length of micro lenses ($f = B$). This case has one major limitation like the low resolution of the images. So, to increase the image resolution, the number of micro lenses need to be increased too in the same amount, which not always is viable. However, it is simpler to obtain an all-in-focus image in standard plenoptic

cameras than in multi-focus ones. To synthesize an image, only one pixel from each micro image at the same relative position to the respective micro lens center is taken into account.

Regarding the limitation on image resolutions, Lumsdaine and Georgiev, in 2006, introduced a new design of a plenoptic camera, a focused plenoptic camera or plenoptics 2.0, like the ones Raytrix produces.

In this new design, the focal plane is not the same as micro lens array ($f \neq B$), which gives a larger depth in which the micro images appear focused. Combining this new design with micro lenses of different focal lengths, it is better since it allows a better compromise between a higher effective resolution and the size of depth of field, resulting in higher resolution images.

2.2.5 Applications, Advantages and Disadvantages

A plenoptic camera can be viewed as a single 3D camera that allows the user to perform some post-processing.

They have a wide range of applications like stereo [20], video [17], depth map estimation or 3D shape recovery [3, 19, 21].

Robotics are making use of these cameras on robotic navigation [6] and recently, plenoptic cameras were also applied to microscopy in the conception of a 3D microscope [10].

Depth map estimation is obviously an application to these cameras, which leads to other ones, like quality inspection.

However, photography is the area with the most popular applications. We can obtain the depth of a scene, all-in-focus images, super resolution and some advanced post-processing, where the user is able to digitally change the focus and the point of view after the picture has been taken.

The astonishing growing on digital photography and the referred applications for plenoptic cameras lead to a consumer market that is ready and anxious to embrace these systems, however, the prices of these optical systems are the main setback. To the commercial market, this is still a disadvantage. Nevertheless, the technology behind plenoptic cameras needs to be improved in order to reduce the prices of these systems.

On the other hand, plenoptic cameras are robust (when compared with stereo pairs of cameras, for instance), since we only have one camera and one main lens. Another advantage is the fact

that we do not need to synchronize multiple cameras in order to obtain a plenoptic function.

These are great accomplishments that can change photography and computer vision the way we know them.

Chapter 3

Algorithms

In this chapter, it will be presented the developed algorithms in order to obtain the depth estimation. It should be noticed that computational efficiency wasn't one of our main concerns. The developed methods are applied to a raw image present in a publicly available data set from Raytrix™[1] and we can divide them in 3 major categories: pre-processing, point set virtual depth estimation and depth recovery.

The work was developed on a Linux environment, using C and C++ languages and OpenCV.

3.1 Pre-Processing

In this section, we will explain the necessary steps to perform before we actually start estimating the depth, which is to find all the micro lenses centers and their types and find a suitable point cloud to analyze.

3.1.1 Finding micro lens centers and their types

The first thing to do, in this step, is to find the center coordinates of the central type 1 micro lens, which we are able to do by analyzing the calibration data file generated after taking a picture using a Ratrix™ camera. In our case, since we are using public datasets available, we just analyze

[1]The dataset contains all files generated after the capture of an image using a Raytrix camera, such as raw images, depth images (grayscale and color), all-in-focus image, an image containing a 3D view and calibration data. It is available in: http://www.raytrix.de/index.php/Research.html

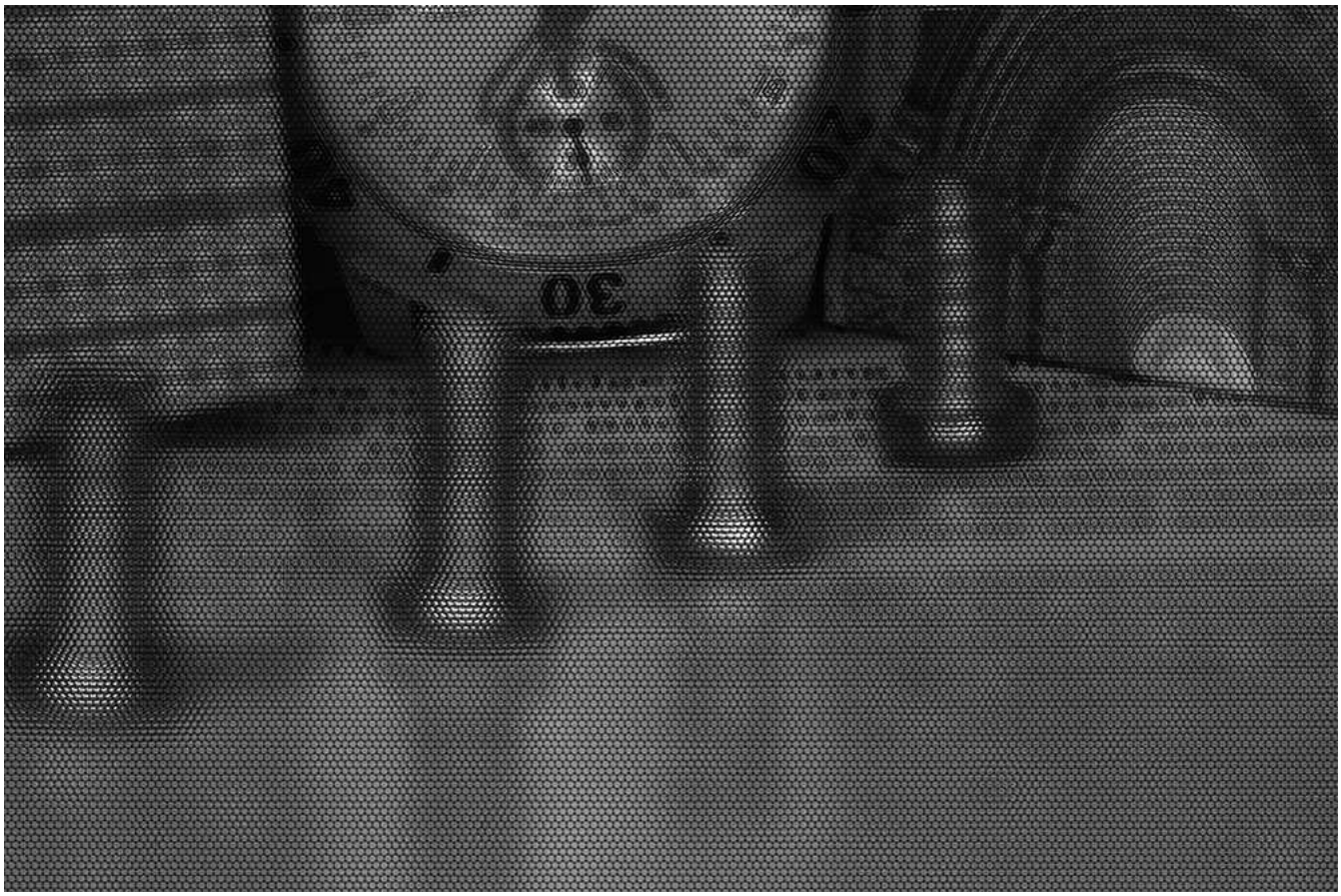


Figure 3.1: Raw image from the Watch dataset provided by Raytrix

the data they provide.

The referred central micro lens center is obtained by adding a vector present in the referred calibration data to the image center. Then, since we know the micro lens diameter, we are able to find the first micro lens in the same line as the central one. We need to refer that this method works when the rotation of the image is zero (or approximately). When the rotation is different than zero, we need to previously perform some axis alignment in order to use it. Using another vector present in the calibration data, we can obtain the first micro lens center in the image. To make this process simpler, we only consider complete micro lenses.

Having all the complete micro lens centers, we created an associative array to help us to identify the micro lens, which will be necessary later. Each one of the micro lens has an identification number, but has too an associative code, which allows us to better identify them. This array is filled based on the following formula:

$$Associative\,Code = (y\,coordinate\,of\,center\ *\ image\,width)\ +\ x\,coordinate\,of\,center \quad (3.1)$$

The use of an associative array to identify a micro lens center is faster, and therefore advan-

tageous, than other techniques (e.g. calculating the distance of a point to all micro lens centers) because it allows us to identify a micro lens center by only comparing two associative keys, which means we only perform a single sweep of the associative array.

3.1.2 Obtaining a 2D Point set

In order to estimate the depth map, we need a point set to analyze. We need points that are visually recognized (with sufficient texture detail), which we denote as salient points. It is more difficult to achieve a depth estimation on points with less texture and that's why this is a key step on the algorithm. It is obvious that we want as many points as possible, but we need to avoid wrong correspondences that can produce bad results and therefore, a bad depth estimation.

The point set is obtained using the SIFT descriptor. This method allows us to obtain the most significant points like corners, edges and contrast points, only by adjusting some parameters, like edge and contrast thresholds.

We choose this method because it is fast and produces good results, comparing to other methods available.

3.2 Point set depth estimation

Since we now have a suitable point set, we can start to process it in order to estimate the depth.

This estimation can be divided in 3 minor steps, which are the identification of the micro lens where the point resides, finding the correspondences of that point and, finally, the actual virtual depth estimation, where we developed a COMSAC method to eliminate points that leads to bad estimations.

3.2.1 Finding in which micro lens the point is

To identify to which micro lens the point belongs, the first step to do is a projection of the point to the line of centers below the point (see Fig. 3.2). Then, we calculate the associative value of the projected point (using equation 3.1) so we can search the associative array of micro lenses to find out the values right after and before of the projected point, and, therefore, the respective micro lenses center.

This method allow us to achieve the identification of the micro lens to which the point belongs without computing many distances that are computationally expensive.

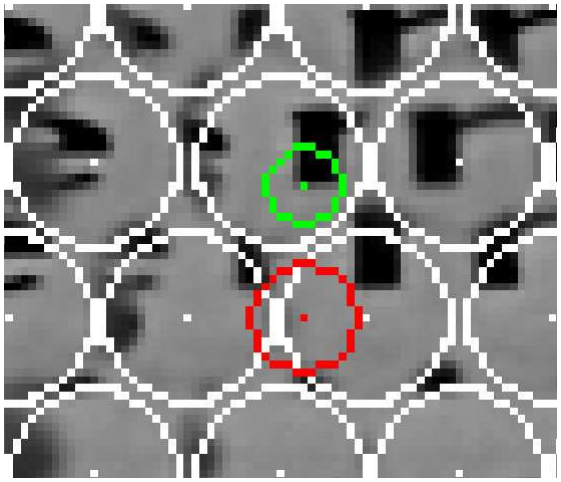


Figure 3.2: Projection of the point

The next step is to find out the micro lens center in the line above those two we just discovered. Since we know how many horizontal micro lens we have in a single line, it is easy to find the referred micro lens center.

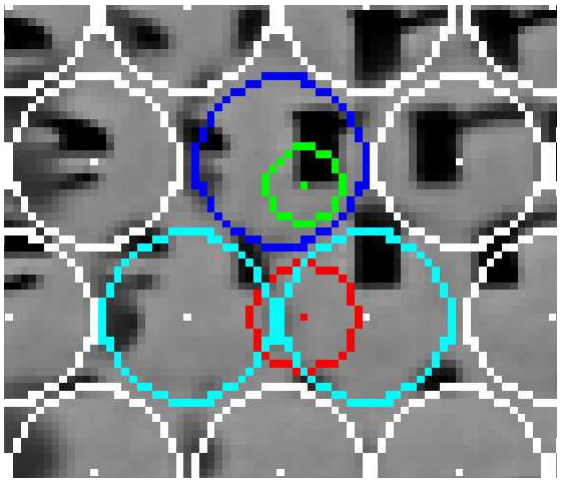
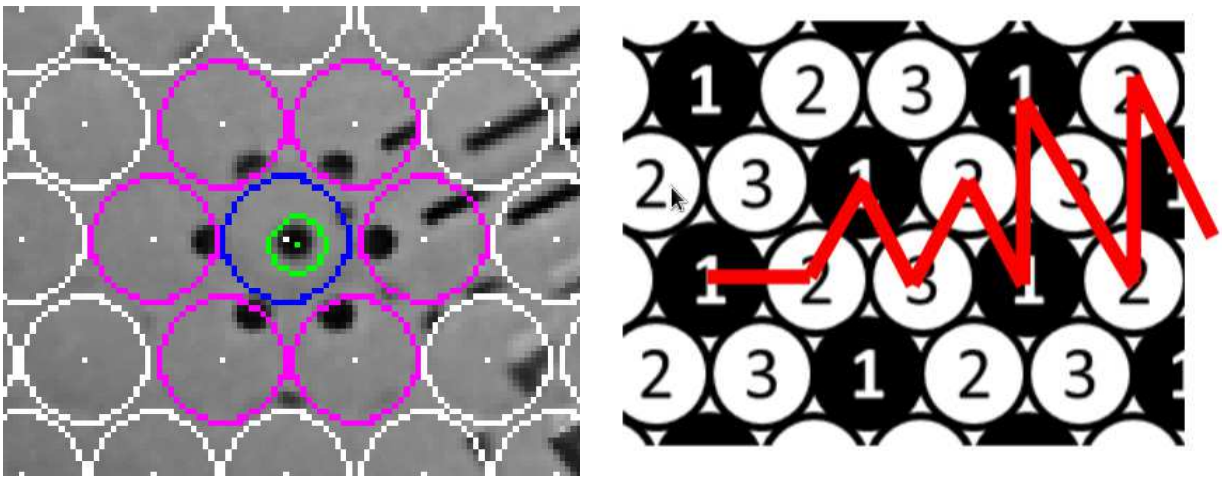


Figure 3.3: The three possible micro lenses and the detected one on blue

By now, we have 3 different micro lenses (and their respective centers) and we need to identify which one of them is the one that contains the point. This can be achieved by simply calculating the distance of the point obtained by the SIFT method to each one of the 3 micro lenses centers. The minimum distance tells us which micro lens contains the point.

3.2.2 Finding correspondences

The step of finding correspondences is automatic and consists on searching for the salient point in the neighboring micro lens images. It should be noticed that our searching profile is different from Raytrix. As we can see on Fig. 3.4, we use the surrounding micro images instead of a sequence of 10 micro lens to the right of the salient point.



(a) Our search profile (b) Raytrix search profile

Figure 3.4: Searching profiles for correspondences of a given salient point.

Perwaß and Wietzke stated, in [14], that a point is imaged more and more times as the virtual depth increases. This leads to the existence of multiple searching profiles for finding correspondences. Our profile, as seen in figure 3.4(a), highlights the surrounding micro images from the micro lenses detected. However, we intend to test other profiles, as future work. For instance, we intend to use a profile where we search for correspondences on microlens that dists two diameters from the detected micro lens.

To find a correspondence we need to select the micro lens where the salient point is and another one from the searching profile. Then, given those two micro lenses, we use the sum of absolute differences (SAD) of a shifting window along the epipolar lines of the selected pair of lenses. Since the micro lenses are in a fronto-parallel configuration, the epipolar lines are parallel to the baseline, which is the line that intersects both micro lens centers, assuming to be the principal points.

However, when we talk about subpixel resolution, a small deviation can cause the missing of a correct correspondence. To overcome this situation, we search for correspondences in a band around the epipolar lines, as illustrated in Fig. 3.5.

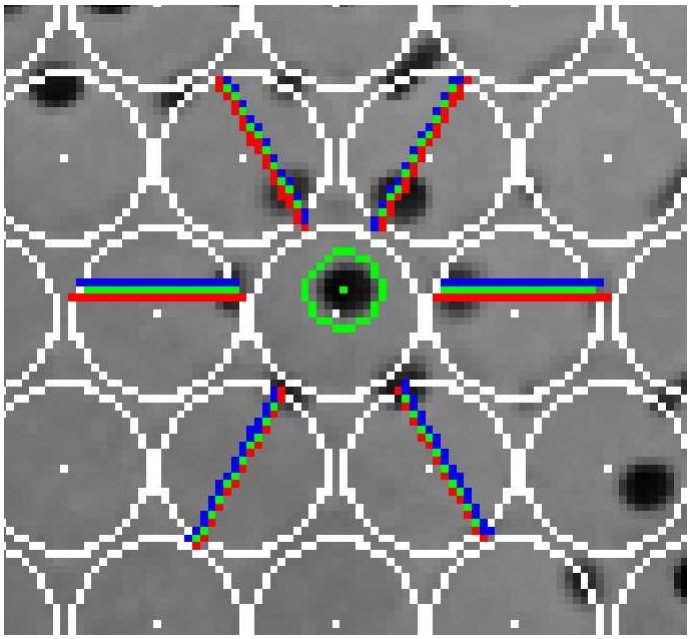


Figure 3.5: The epipolar band where we search for correspondences

In order to obtain robust results, we introduced another improvement in which we scaled the SAD by the number of pixels that lie inside the shifting window. We did it because it happens very often that the correct correspondence is very close to the micro lens border. So, to avoid rejecting this points and use only the part of the window that really contributes to the SAD, we must scale it by the number of pixels that actually belong to both window and micro lens.

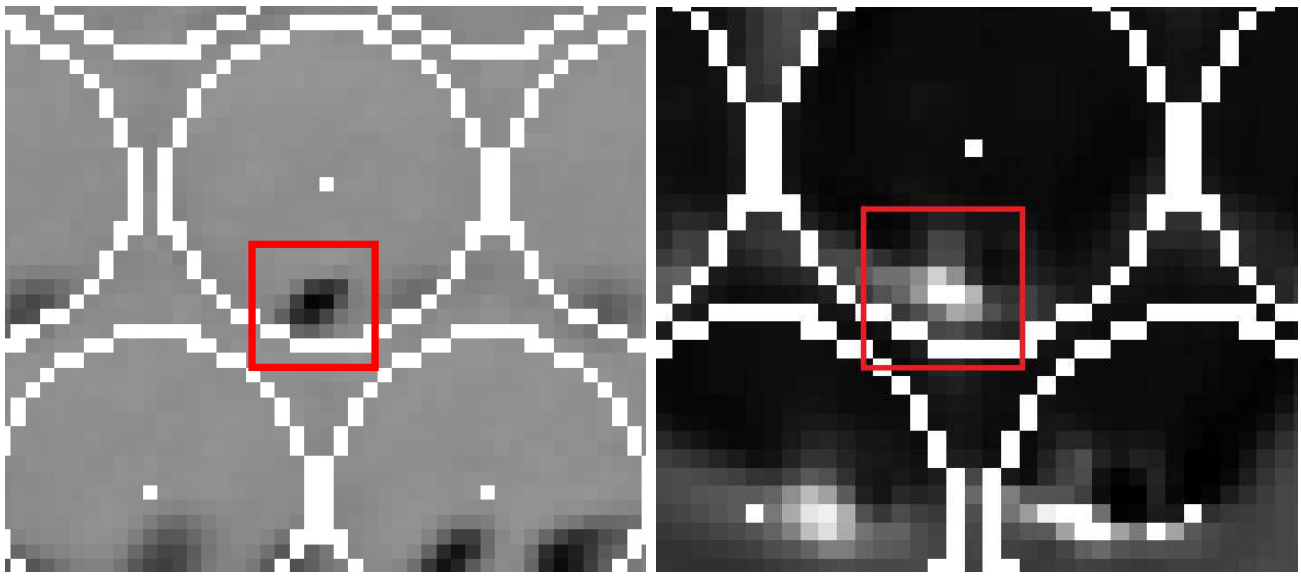
We denote this new value as mean absolute difference (MAD), expressed as follows:

$$MAD = \frac{SAD}{number\ of\ pixels\ inside\ image} \tag{3.2}$$

Our main goal, as stated, is to achieve a fully automatic algorithm, so we need to be aware of every situation that could happen on an image. Having that in mind, we also need to fulfill salient points with different scales of intensity. The best example that can illustrate this situation is the case of a dark point on a bright zone, or even the total opposite.

The need of a threshold to reject automatically some of the correspondences is vital and we need to be able to tune it for the entire image. So, we defined another measurement by doing another scaling, this time to the MAD. In this case, we divide it by the difference between the value of the central pixel of the SAD window and the mean value of the micro lens where the salient point is. The referred scaling factor is denoted by μ, while the final value is called MRD (Minimum Relative Difference).

$$MRD = \frac{MAD}{\mu} \tag{3.3}$$



(a) Dark point on bright zone (b) Bright point on dark zone

Figure 3.6: Salient points with different surrounding intensities

After searching the whole band, we can establish the correspondence point as the one with the smallest value of MRD, ensuring that it is below a threshold.

Using only the SAD method without any scaling did not produce the best results. This happened because, sometimes, we had to search for correspondences on areas without considerable different levels of intensity, which easily leads to bad correspondences. The fact that we are searching for correspondences on a micro lens, which is circular, did not help either, since the shifting window used to search for them is square and we could be using information from outside of the micro lens. So, regarding this limitations, we had to perform the explained scaling to the SAD in order to obtain better results, which we did.

3.2.3 Virtual Depth Estimation

Now that we have the correspondences for each point of the point set, we can estimate the virtual depth using the redundancy present in the image. We decided to recover the depth by back tracing the rays that comes from micro images and intersecting them to obtain the virtual point. The profile we use to find correspondences may obtain up to 6 correspondences for a single salient point, resulting on an intersection of seven back traced rays in space.

To eliminate points that could lead to bad results, we developed a COMSAC method (inspired in RANSAC) in order to robustly obtain the virtual depth. Points with a small number of

correspondences are more error prune, so that's why we only consider points to apply this method with two or more correspondences.

This method consists on a series of seven steps, which we present in detail, for a better understanding.

- **Step 1 - Selection of a subset of three lines**

  This step is really simple, as we only select the first three lines of the model to estimate the 3D virtual point.

- **Step 2 - Estimation of the 3D virtual point**

  The main goal of this step is to estimate the 3D virtual point. In order to obtain the referred point, we group the three lines considered in the previous step two by two, and we search for the point that minimizes the distance between those lines [7]. Since we are talking about lines in space, we cannot do a simple intersection of them, as they can be skewed lines. Selecting three lines, we are able to make three different combinations with them, which leads to three 3D points. In order to estimate the final 3D point, we use 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 point to all the correspondence lines obtained in the previous step.

- **Step 4 - Assessment of the model**

  In this step, we establish a threshold so we can distinguish the good from the bad results, allowing us to assume which lines are suitable to add to the model. If the distances of the 3D point to every single one of the correspondence lines is below the referred threshold, we can label that line as an inlier, since it passes close enough to the 3D point. If not, it is considered and outlier and it is discarded from the model.

  We are able now to classify the chosen model, based on the number of inliers and outliers. If there is more than one outlier in the model, the model is discarded and we go back to **Step 1** and test another combination of lines. If not, we advance to the next step, since we can consider it an usable model.

- **Step 5 - Re-estimation of the 3D virtual point**

  The idea on this step is similar to the one in **Step 2**, since we do another estimation of the 3D virtual point. But there is a slightly difference to **Step 2**. In that step, we considered three random lines. But now, the lines considered are the ones labeled as inliers in the previous step. So, in order to estimate the new 3D virtual point, we group lines two by two and repeat the process of finding the point that minimizes the distance between them as many times as possible combinations of inliers, which will lead to several 3D points. Finally, the new 3D virtual point is obtained by computing the median of the coordinates of the previously referred 3D points.

- **Step 6 - Error metrics**

  In this step, we evaluate the model in terms of error. It is a mean error, and it is the sum of the inliers distances obtained in **Step 3** divided by the number of inliers.

- **Step 7 - Repeat steps 1-6**

  Now that we have the error of the model, we start over the process and repeat steps 1 to 6 as many times as possible combinations between the correspondence lines.

This COMSAC method is done for each one of the salient points, and we analyze every error model. In order to obtain robust results, we need to define another quality threshold for the error model. The 3D point considered to the virtual depth is thus the one returned by this method with minimum error, regarding that it is below the referred threshold.

For instance, a salient point with 6 correspondences can lead up to 35 models, which means 35 different 3D virtual points. Since this method returns the model with the minimum error, we can assume that the respective 3D virtual point is the correct one.

After analyzing the entire point set, we are able to obtain each respective 3D point and, consequently, the virtual depth map.

3.3 Depth Recovery

In this section, we will present the developed method to densify the sparse depth map and create the final depth image. It is the process of transforming a sparse 3D point cloud into a dense

2D map. We developed this method using a random approach, where we pick a random point without depth (intensity) information and estimate its depth considering the surrounding points.

The densification of sparse point clouds is a topic that is intensively studied, due to its importance on 3D reconstructions or depth estimations. In the majority of the cases, the depth maps obtained are sparse and it is vital that they become dense, in order to completely estimate the depth of a scene, for instance. There are several methods that have been developed and improved in order to perform this densification. We can highlight region growing methods, an iterative process that begins to select seed points and then analyze the neighboring pixels in order to decide their inclusion on the selected region. However, due to the lack of time to explore solutions of densification sparse maps, we developed an heuristic method of random growing that produced good results.

For a better understanding of the developed method, we can divide it in two major steps. The first one is the processing of the sparse data, where we make the adequate adjustments to use it on the second step, which is the random growing of the point cloud.

Since it is hard to perform a random growing of an image with the original raw size (it would lead to an heavy computational time), we start the processing step by creating an image downsampled 8 times from the raw image. Then, we scale the x and y coordinates of the virtual points from the sparse depth map to the new image. This leads to information loss, as some points can overlap, however, it is not a major concern (as proved by our experiments).

The depth values (z coordinate) from the sparse point cloud obtained in the previous section are also mapped, but this time they are mapped to values between 123 and 255, representing the grayscale intensity of the mapped points. Filling the image with the mapped information, we obtain a representation of the 2D sparse depth map (see figure 4.7), where we can already perceive some of the depth.

Finally, we create a list of the pixel coordinates from the partially filled image without depth information.

To perform the random growing, we pick a random point without depth information previously stored. Then, and defining a 3 by 3 window, we analyze the 8 surrounding pixels. To fill the random point, we must assure that half of the window has depth information. If this condition is satisfied, we compute the median of those values, which becomes the intensity of the randomly picked point. If not, we randomly pick another point to analyze.

If a point is filled with depth information, we remove it from the list of points without depth information and we pick another point from it to analyze.

This process is repeated until the whole image has depth (color) information, which leads to a totally filled gray level depth image - a dense depth map (Figure 4.8).

We need to reinforce the idea that computational efficiency was not our main concern, which means that there may exist methods with better efficiency, but this random growing method produced good results, for a first approach.

Chapter 4

Results

In this chapter, we present the obtained results of our depth estimation method as well as a brief discussion of them.

In order to test the method, we used a dataset from Raytrix that is publicly available. The dataset has all the files generated after the capture of an image using a Raytrix camera. Using the dataset, we have access to the raw image, depth images and also to the calibration data, which means we have the necessary parameters to develop our work, such as the micro lens diameter, the distance between micro lenses of the same type, and so many others.

Using a dataset from Raytrix is important since it allows us to compare our results with the ones they provide.

We developed our work using the Watch dataset. It presents an image with a watch in the background, between two heavily textured planes and also a sloped plane in the ground where we can see four upside-down screws at different distances from the observer. It is an interesting image to depth estimation purposes, since the objects in the image are at different distances, which is good to depth perception. We can also notice the loss of texture and blur increasing on the ground plane as it comes closer to the observer.

We will present the results in the same way we presented the developed algorithms, dividing them in three categories, so we get a better understanding and also a better organization.

4.1 Pre-Processing

The first relevant result to be presented is the salient point set obtained using the SIFT method. Figure 4.1 shows the 2D point set obtained marked in the raw image.

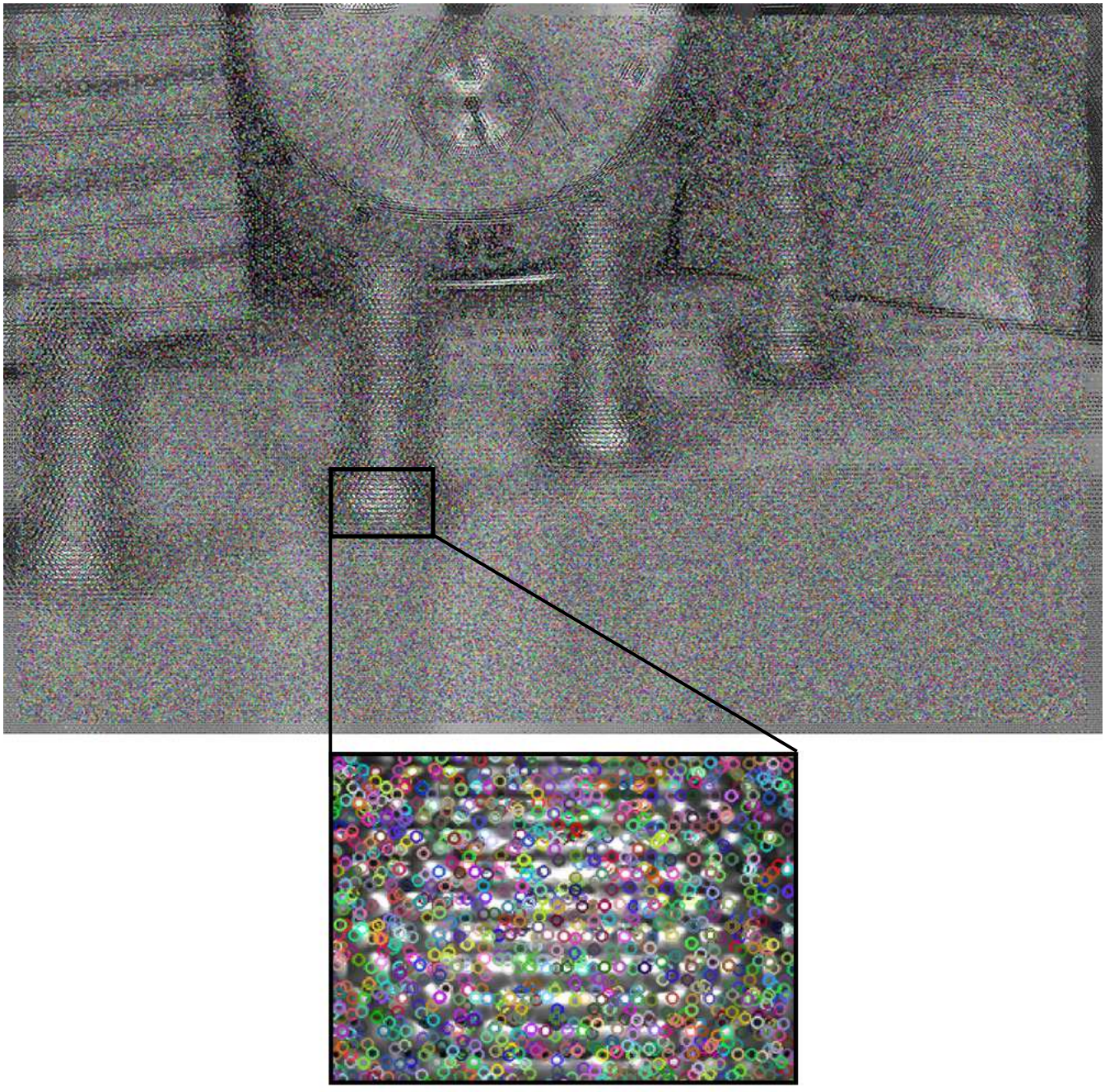


Figure 4.1: 2D point set obtained with the SIFT method

In order to obtain a different amount of points, SIFT parameters need to be tuned. As we can see in the figure 4.1, we manipulated them so we could find a trade-off between the amount of points and their location, only to cover the majority of the image, which we did.

The SIFT method returned a large amount of points (almost 400000), which turned to be enough to obtain the sparse depth map.

Analyzing the obtained results, they were satisfactory, since we have a great amount of points at lower computational time. Thus, we can say that SIFT method is fast and it has almost no impact on the performance of the developed algorithm.

4.2 Sparse Depth Map

Having now the point set to analyze, it is time to analyze the results about the sparse depth map estimation.

Before we present the depth results, it is important to show some results obtained during the method, like the process of finding correspondences.

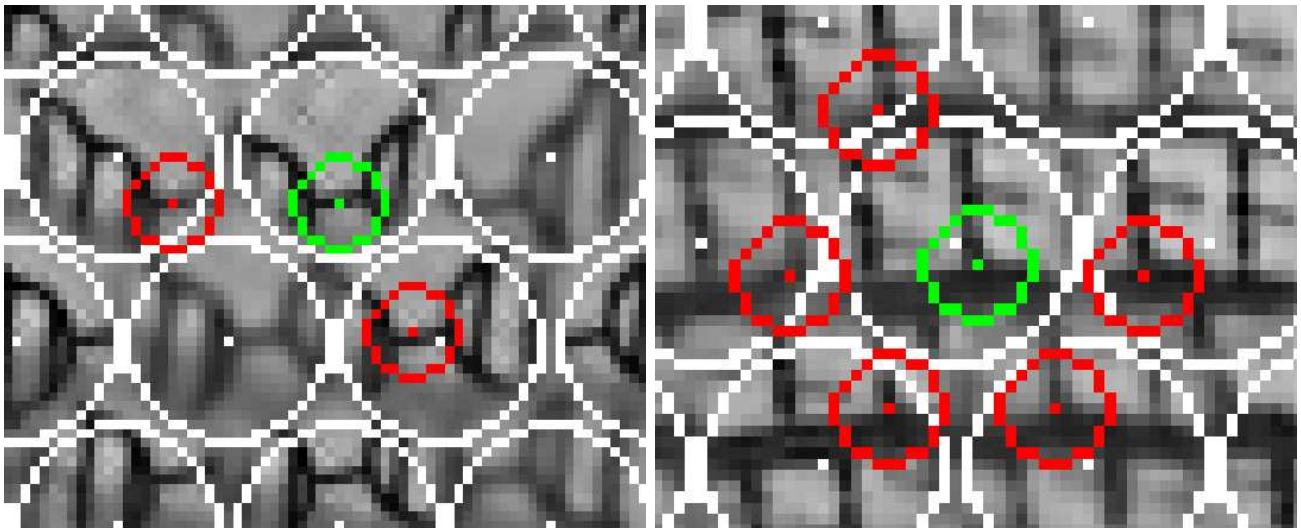
Figure 4.2 shows some correspondences found and how we print them in the image (marked in red). The presented correspondences are examples of good ones, but there are points where the task of finding correspondences is harder. However, and following what has been said in section 3.2.2, all the scaling done to the SAD helps us in this process.

As stated before, we choose only to process points with two or more correspondences, as we get the sufficient redundancy to process and apply the COMSAC method.

The presented result in figure 4.3 is the representation of all correspondences found for each salient point detected. Doing a brief analysis to some points, like we did in figure 4.2 it is the best way to analyze the results, however, even with the overlap of data, our results appear to be reasonable.

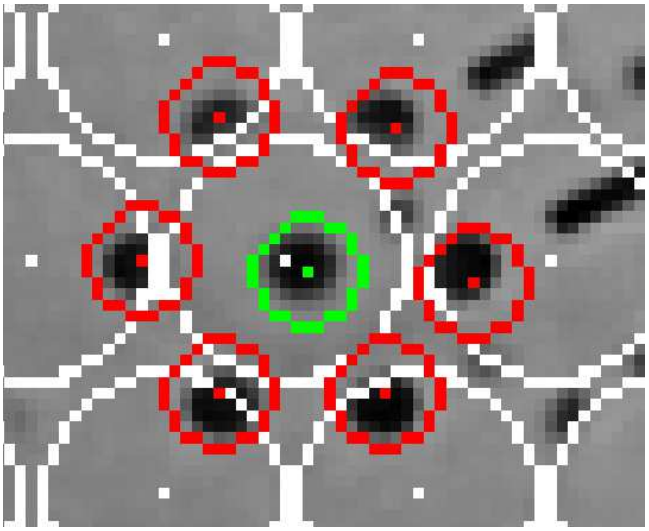
Despite the good results produced by the developed method of finding correspondences, the method is not immune to errors. Bad correspondences can lead to bad depth estimations, which is why we need to apply the COMSAC method, in order to eliminate them and obtain robust results.

The results obtained after applying that method are presented in figures 4.4, 4.5 and 4.6. They are the representation of the results at different views, allowing us to do a better analysis.



(a) Two correspondences

(b) Five correspondences



(c) Six correspondences

Figure 4.2: Examples of correspondences found for some points

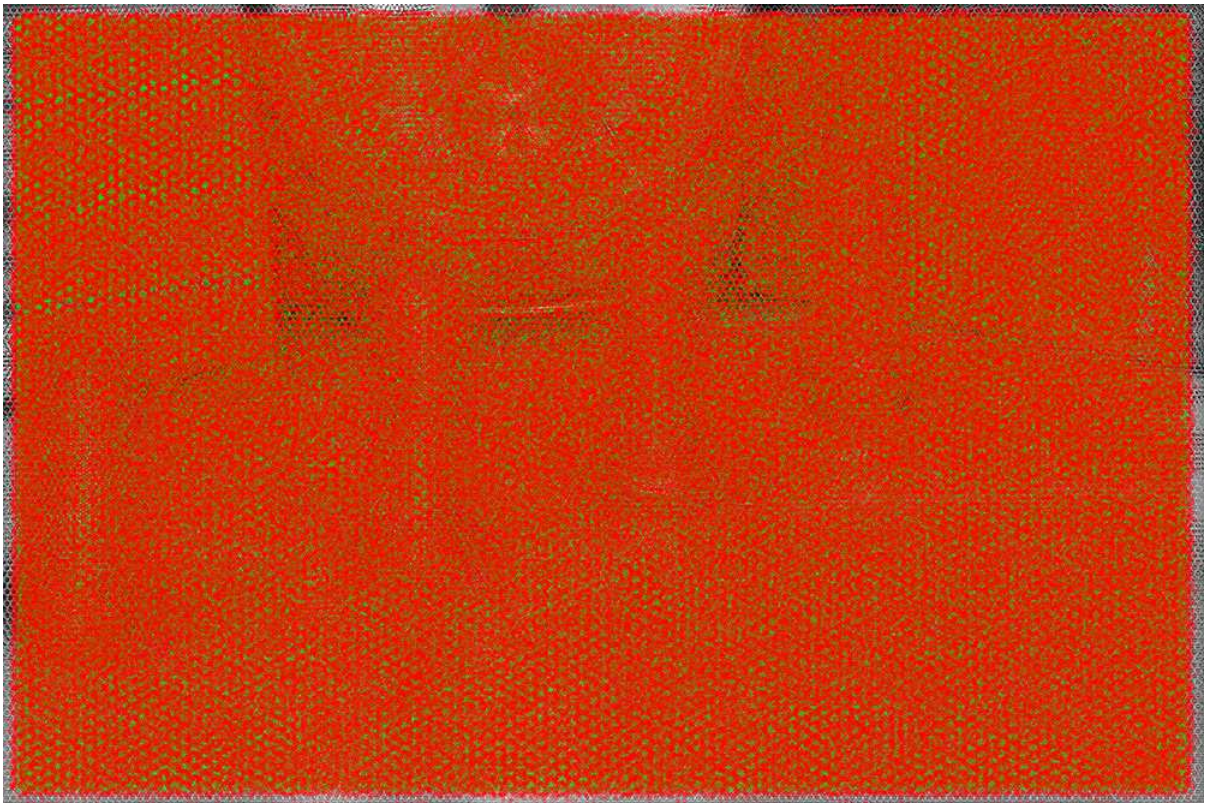


Figure 4.3: All correspondences found

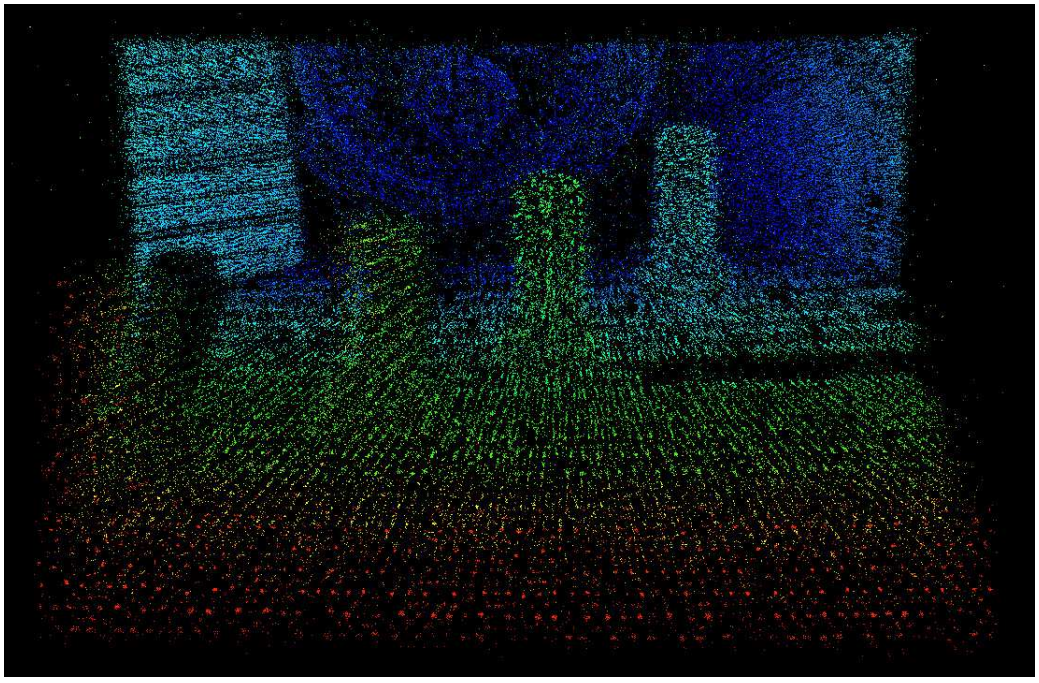


Figure 4.4: Frontal view

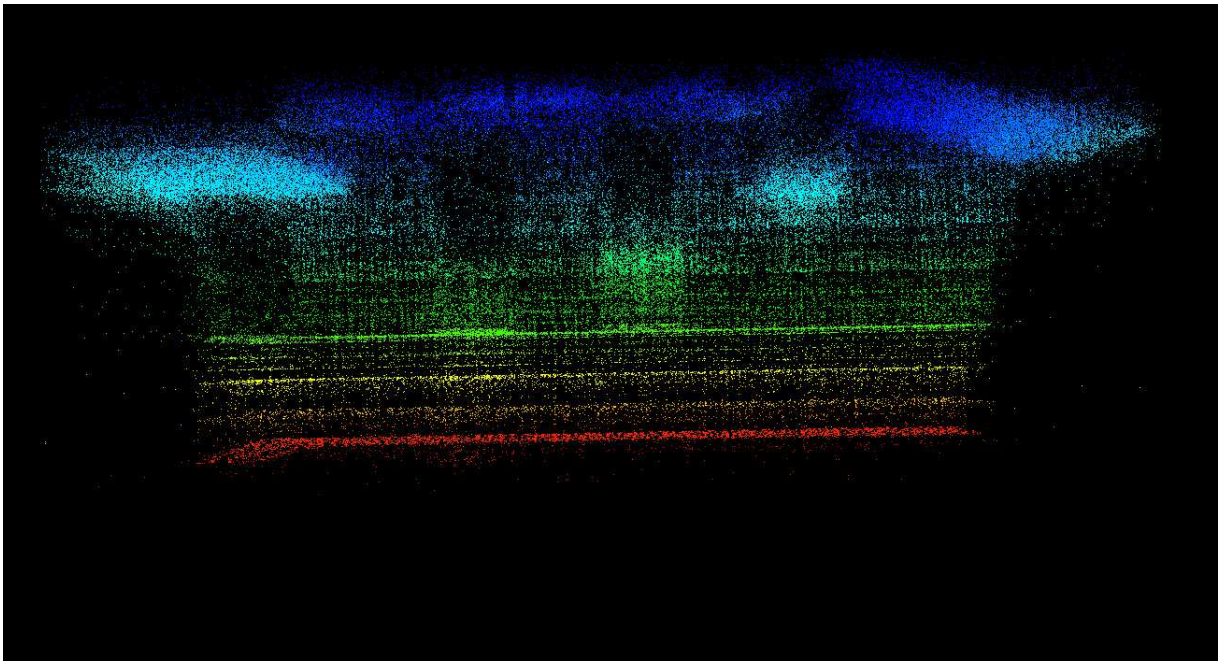


Figure 4.5: Top view

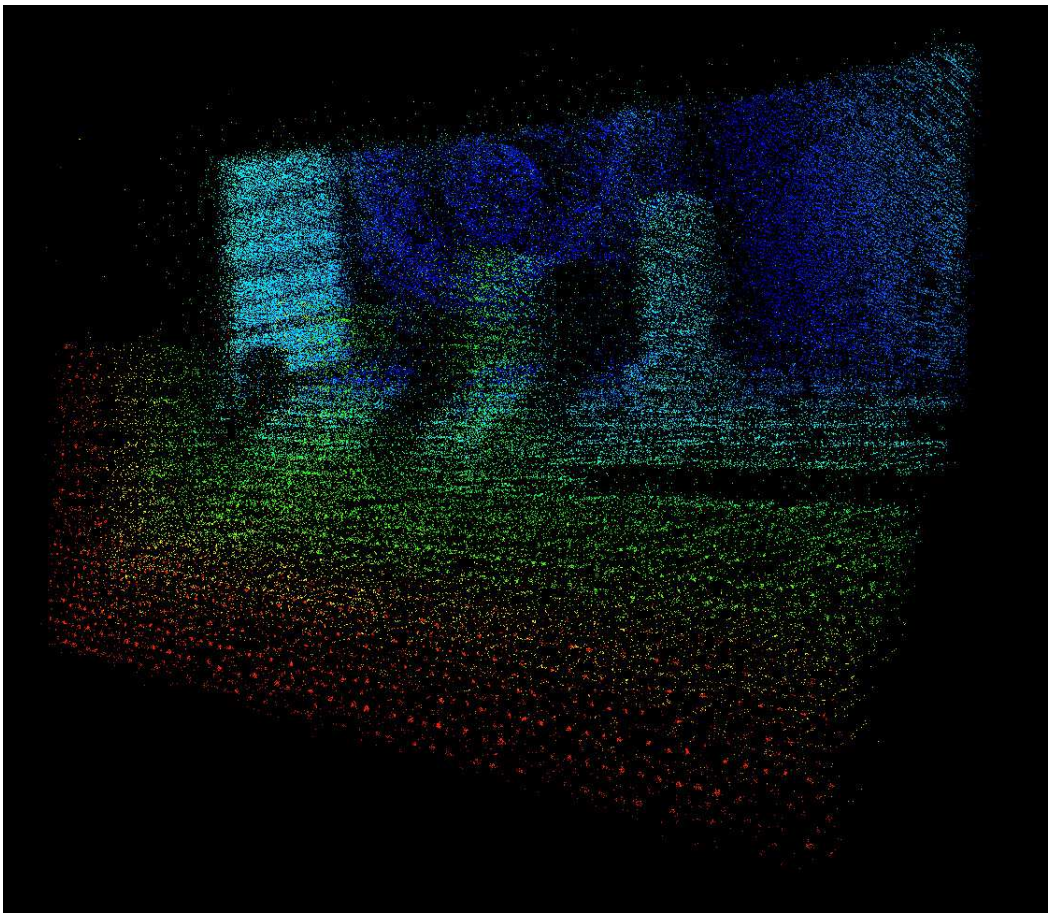


Figure 4.6: Lateral oblique view

To analyze the obtained results, we will consider the frontal view, as it is the best one. The depth is clearly visible in the figure, with the color map helping us to have a perception of the depth. Comparing the point cloud depth estimation with figure 4.1, it is possible to spot some similarities. The most visible one is the depth differences between the four screws, but there are some other examples, like the watch, which we know that it is the object with the most distance from the observer, with figure 4.1 corroborating that fact.

Despite we cannot guarantee the correctness of Raytrix's results, as they are also estimated from the raw image, we still consider them as the reference results. So, in order to measure the difference between the results, we compared our depth values with their values. Their depth values were obtained from the depth image available in the dataset. Since the values are 16 bit values, we need to divide the pixel value by 65535, resulting in a normalized value. Then, we compute the virtual depth based on the following formula:

$$\nu = \frac{1}{1-p} \tag{4.1}$$

where ν is the virtual depth value and p the normalized pixel value.

Since 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, we were also able to estimate our virtual depth values.

We obtained a mean error of 11%, which we consider good and allows us to believe that the deviation between our depth and the reference depth is not high.

4.3 Dense Depth Map

The final part of the developed method is to transform the sparse depth map into a dense depth map. Figures 4.7 to 4.12 illustrate the process.

Figure 4.11 shows the color depth map. Doing an analysis of the obtained results by comparing them to Raytrix ones (see Figures 4.10 and 4.12), we conclued that our method of region growing produces good results, despite the presence of discrepancies. However, the greatest limitation of this method is the computational time. For instance, to build an image with a 400x300 resolution, it take approximately 6 minutes. The problem is that, for bigger images, the time it takes to fill the image increases dramatically. If we try to fill a 800x600 image, the method takes, at least, a couple of hours before ending. This happens because the random picking of points becomes less efficient over time. However, we highlight the fact that the efficiency of our algorithms is not our main concern.

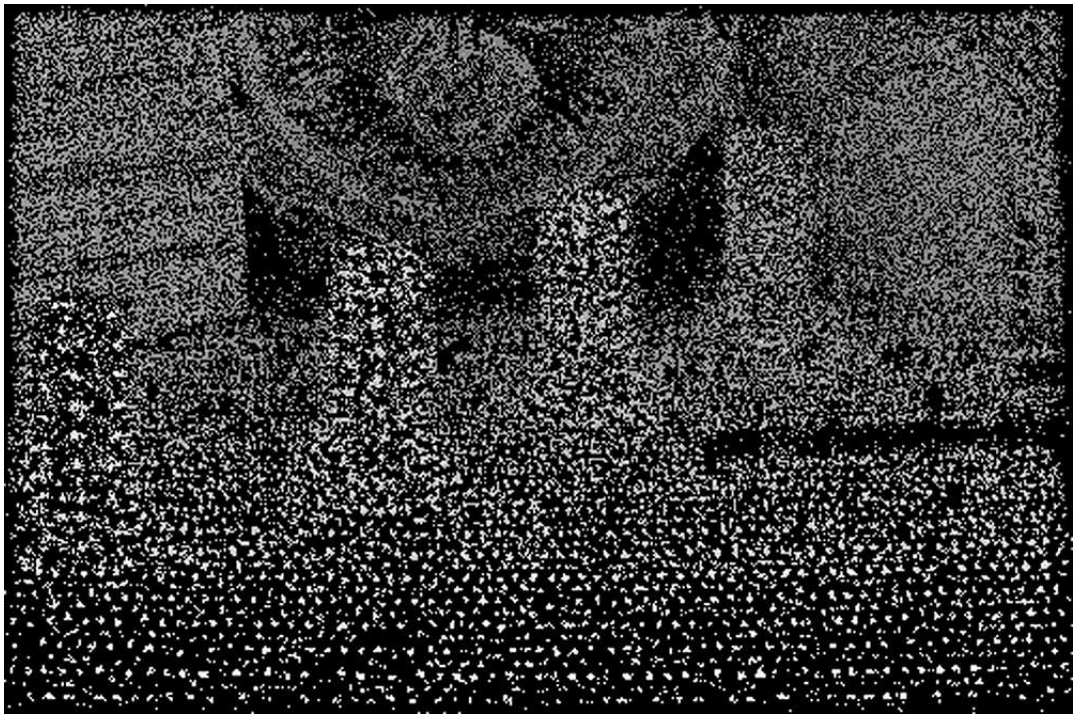


Figure 4.7: Sparse virtual depth estimation

Figure 4.8: Dense virtual depth estimation

Figure 4.9: Dense virtual depth estimation with a median filter

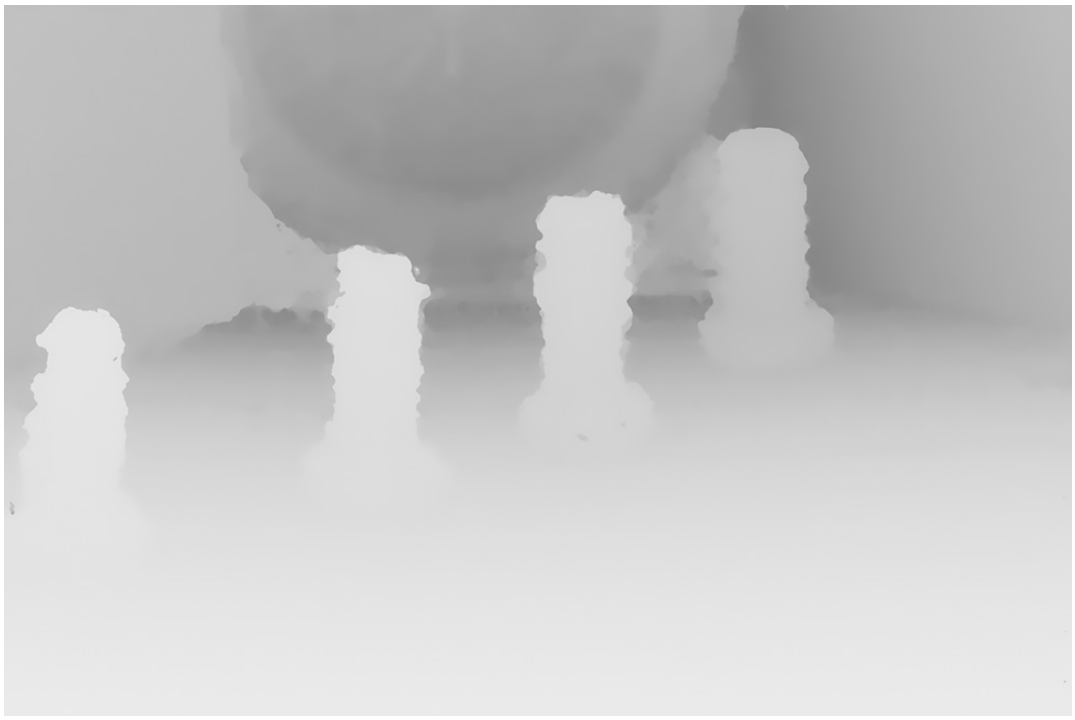


Figure 4.10: Raytrix depth estimation

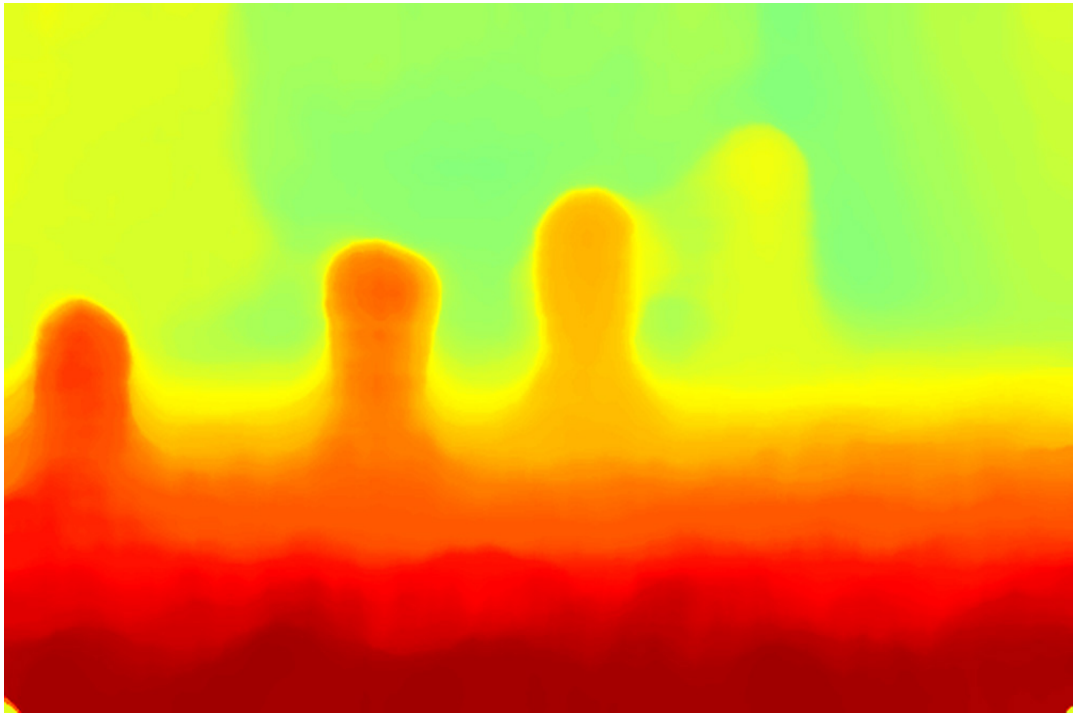


Figure 4.11: Dense virtual depth estimation with a color map

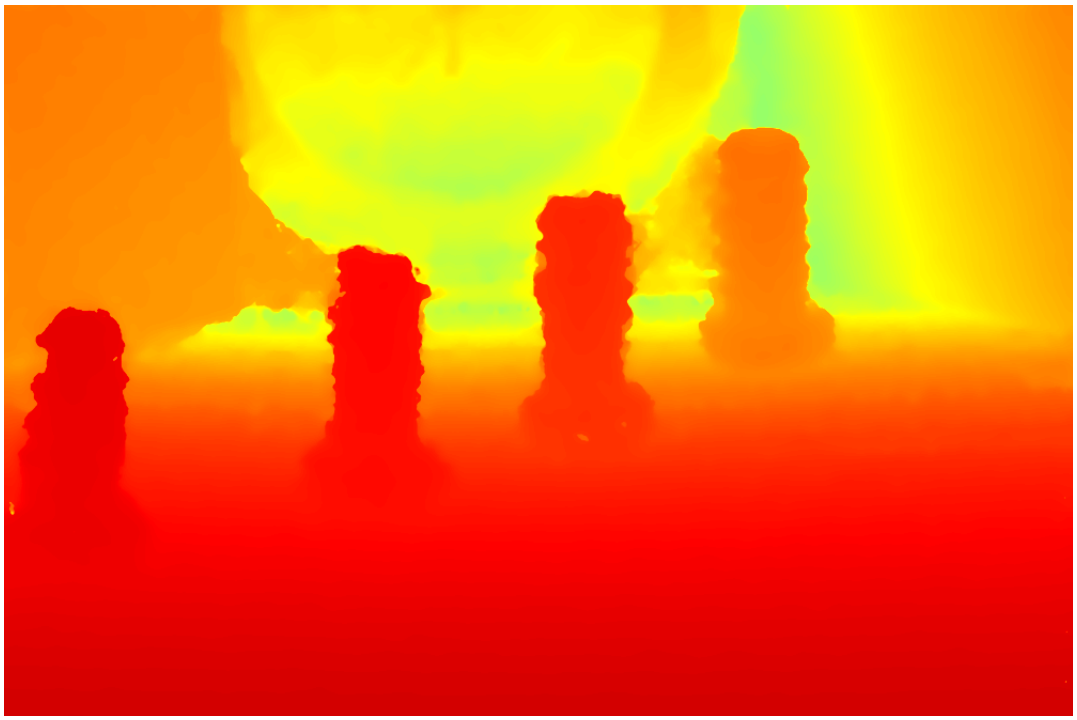


Figure 4.12: Raytrix depth estimation with the same color map

4.4 Other Datasets

To understand how well our depth estimation method works and it limitations, we applied the developed method to another dataset. This time, we choose the Andrea dataset, which is composed by a face capture. This image is harder to process due to the lack of texture on the skin, being a great challenge to our method.

Figure 4.13: Raw image from the Andrea dataset

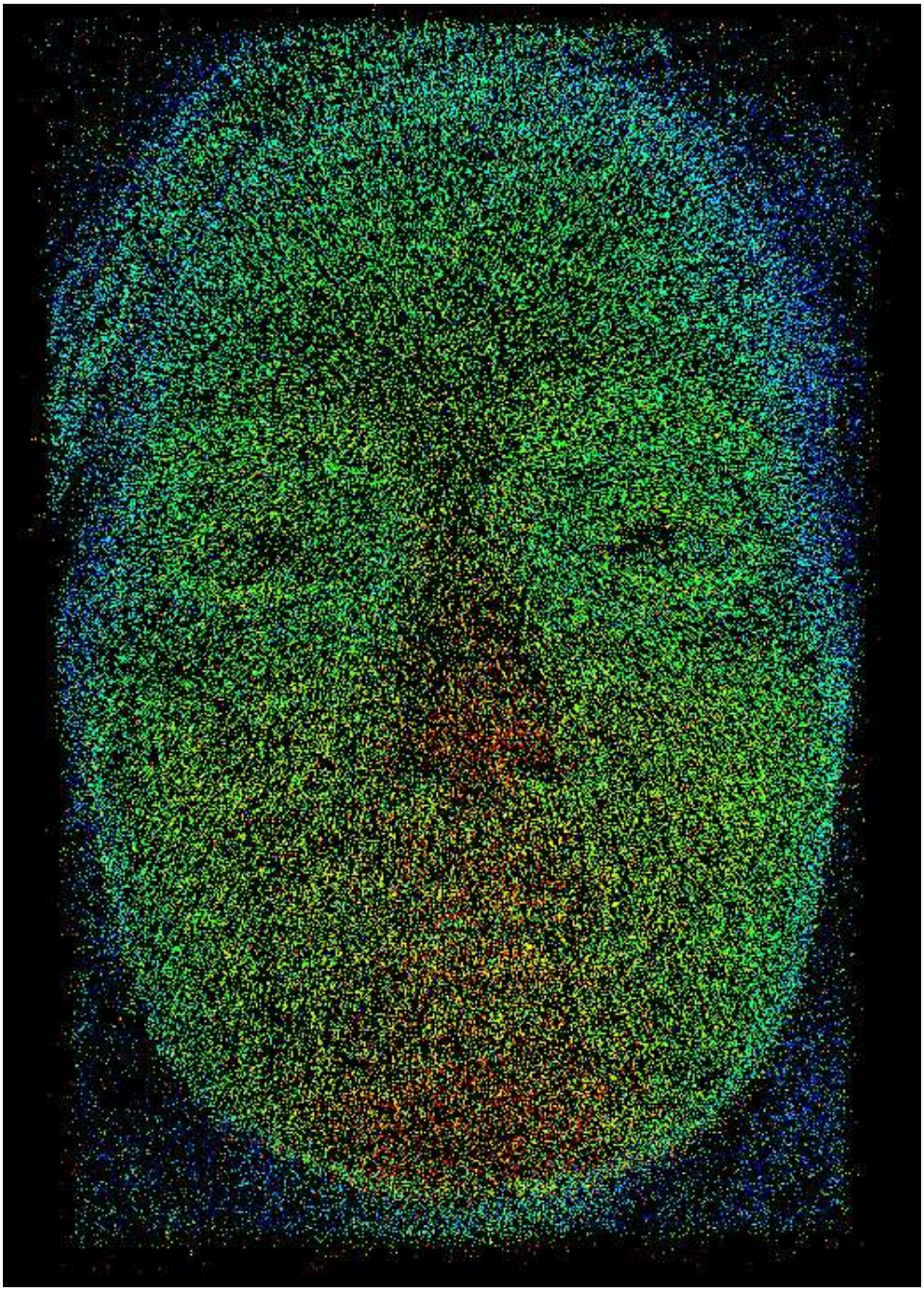


Figure 4.14: Frontal view

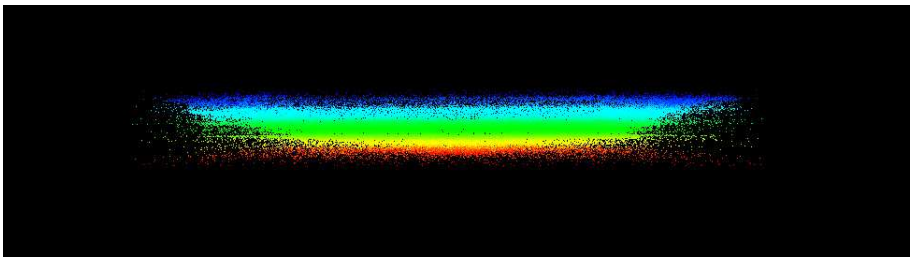


Figure 4.15: Top view

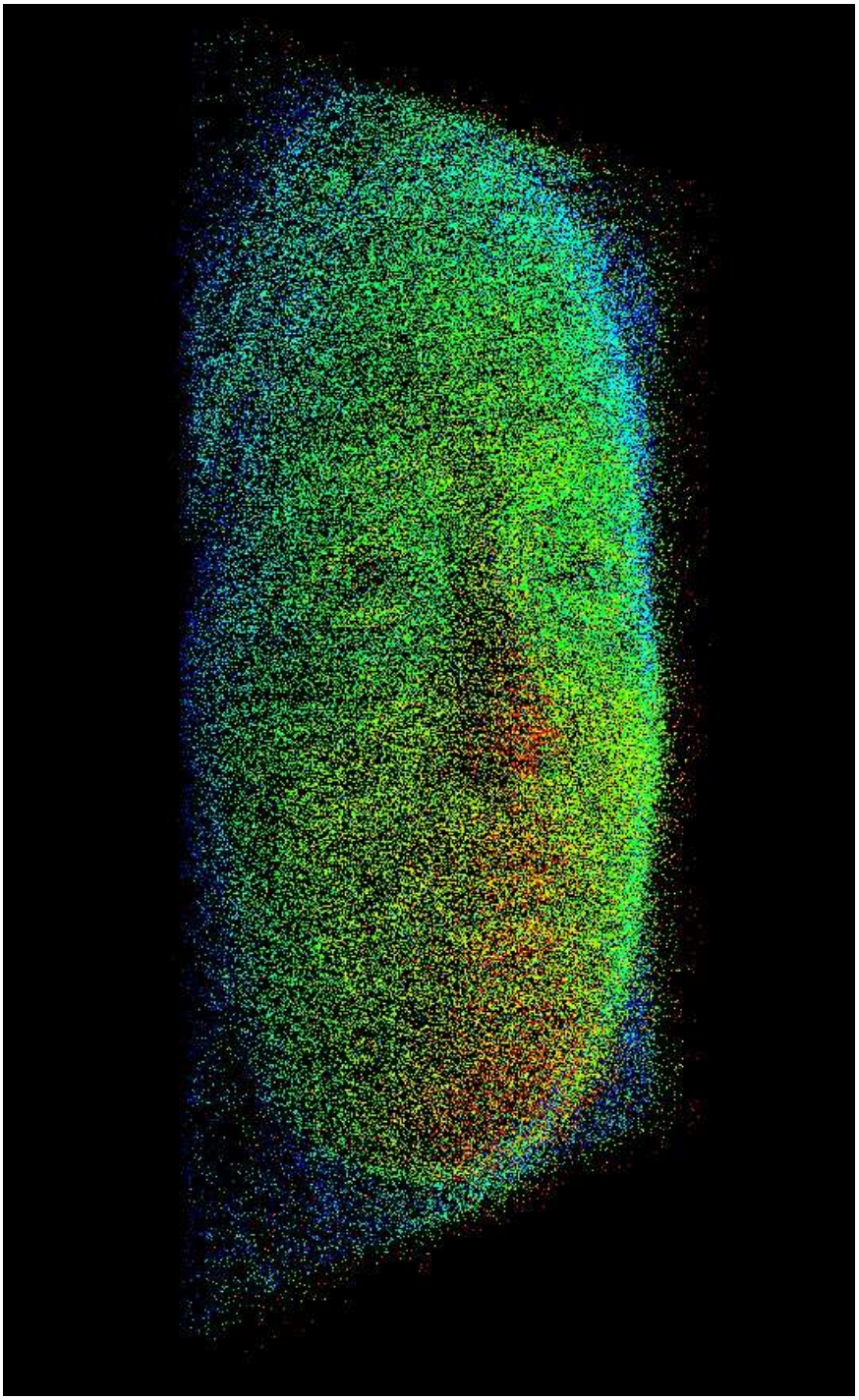
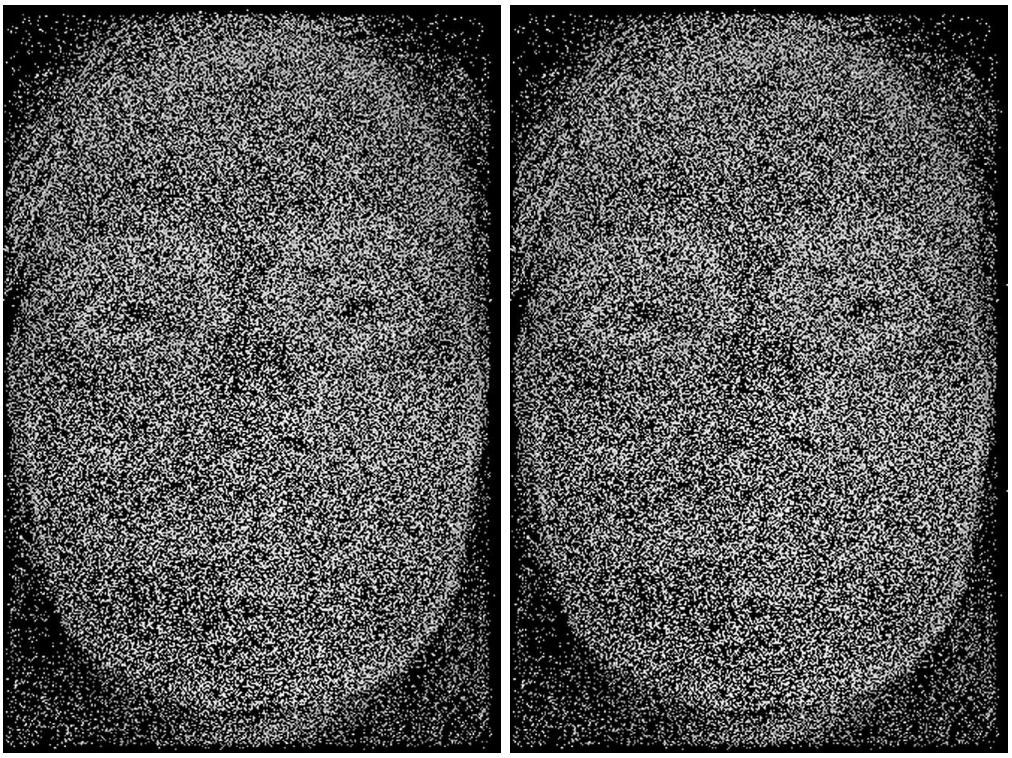
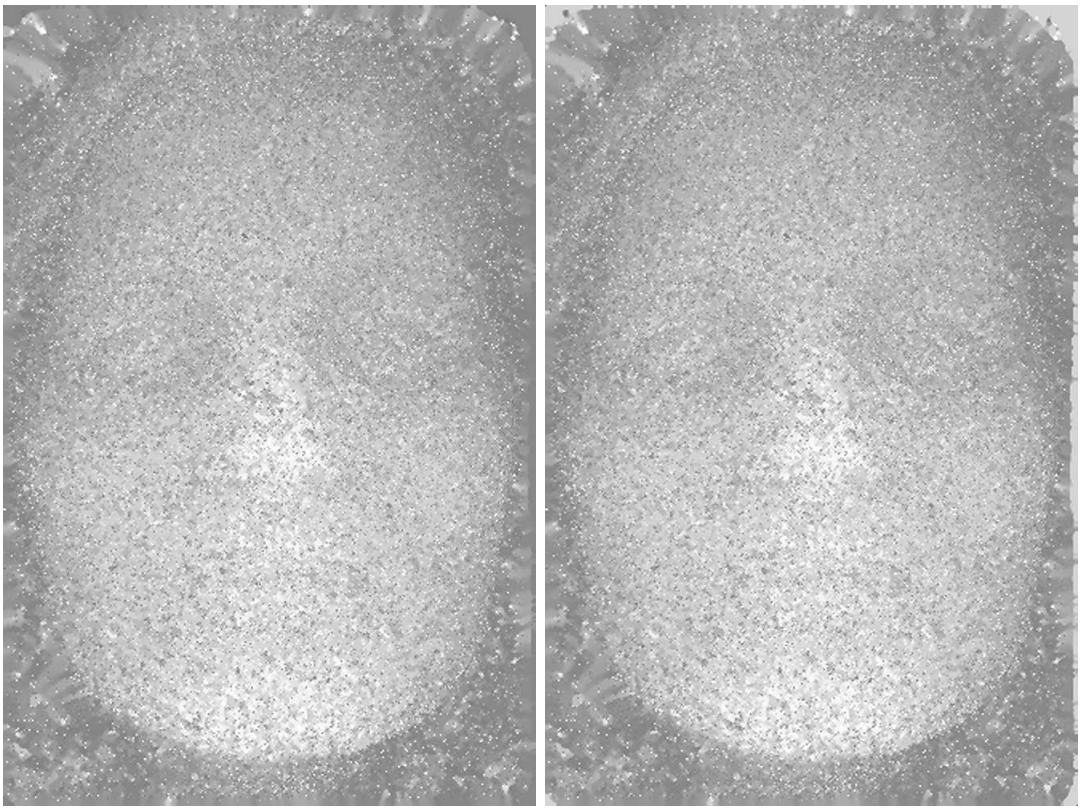


Figure 4.16: Lateral oblique view



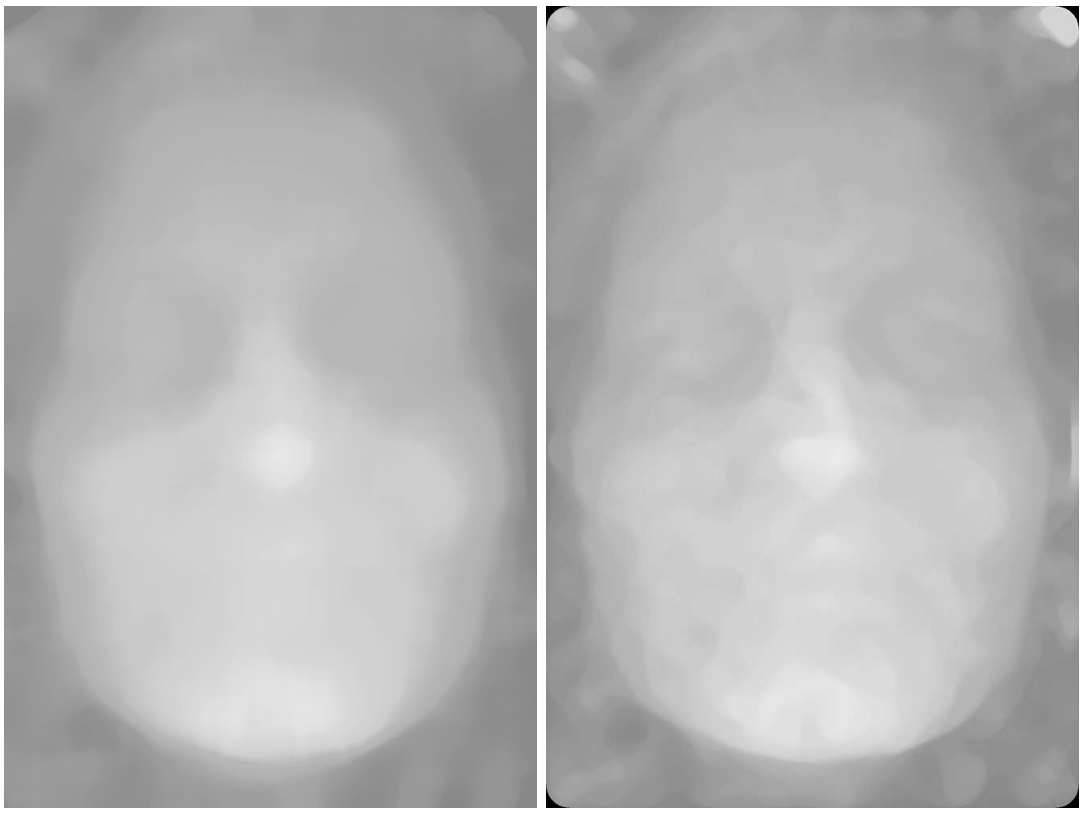
(a) 8 bits (b) 16 bits

Figure 4.17: Sparse virtual depth estimation



(a) 8 bits (b) 16 bits

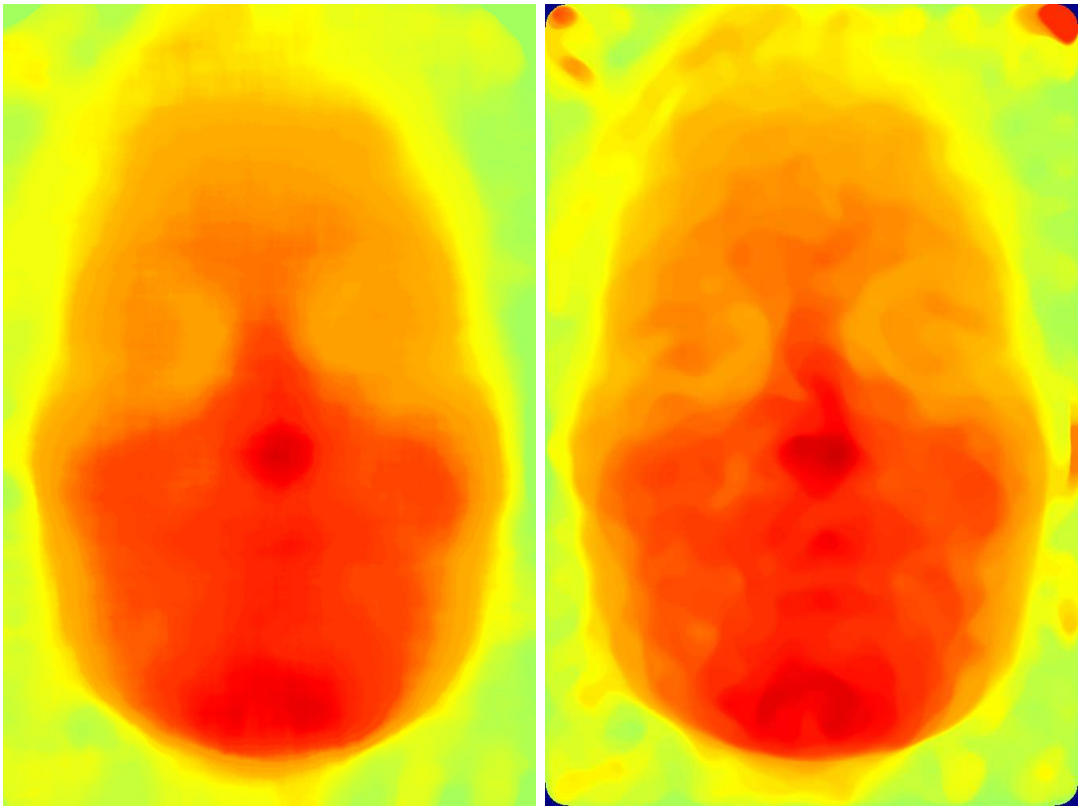
Figure 4.18: Dense virtual depth estimation



(a) 8 bits (b) 16 bits

Figure 4.19: Dense virtual depth estimation with a median filter

Figure 4.20: Raytrix depth estimation



(a) 8 bits (b) 16 bits

Figure 4.21: Dense virtual depth estimation with a color map

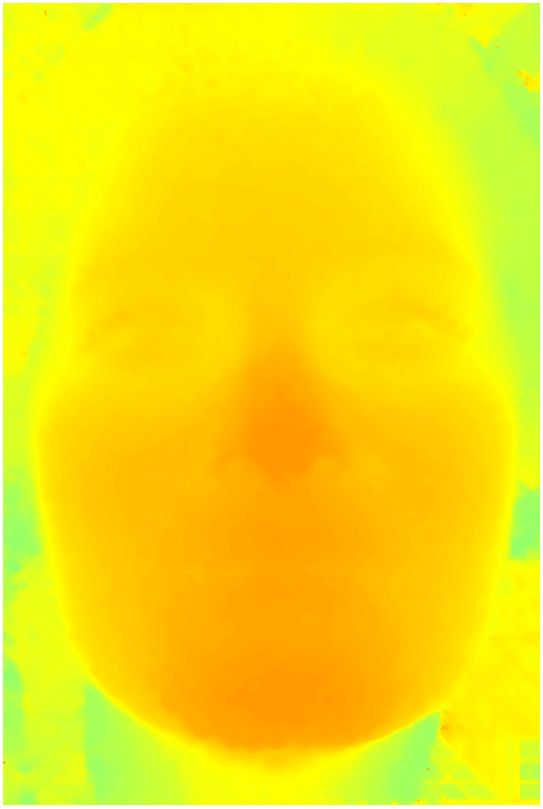


Figure 4.22: Raytrix depth estimation with the same color map

As stated before, the lack of texture poses as one of the biggest challenges to our method. The results of our method applied to the Andrea dataset are presented from figure 4.14 to 4.22.

The application of the Andrea dataset to the developed method turned out to be an excellent way to test our method, due to the results produced. It is presented, in figure 4.14, the frontal view of the sparse depth map. We obtained a significant mesh of points - more than 159.000 - which was vital to obtain good results. Analyzing the referred figure, we can perceive the difference of depths, but not as much as in the Watch dataset. The chin and the nose are clearly closer than the eyes or the forehead, however, in areas like hair, where it is lacking information, the perception of depth is not as good as that.

In a general way, the obtained results to the sparse depth map are good, in which we obtained a relative error of 8%, comparing the difference between the virtual depth obtained and the corresponding virtual depth in the Raytrix image.

The process of transforming the sparse depth map into a dense depth map also produced satisfactory results. Figure 4.21 is the representation of the dense depth map and the different depths obtained are consistent with the theoretical assumptions. For instance, the nose and the chin are closer than the forehead, which makes sense.

We can see the similarities between the obtained color depth map and the one provided by Raytrix (see Figure 4.22), being the chin and the nose the most perceptive cases of similarity. The remaining zones are similar too, but the lack of textured points leads to bigger areas with the same intensity, which is something that we can observe on both figures.

However, analyzing the whole color depth map from Raytrix, we are able to perceive some artifacts in the image, especially outside of the face zone, which makes us to believe that they also had difficulties to estimate the depth on that areas, due to the referred lack of salient points to process. It is also important to mention that the color maps used in both images are different, despite the similar tonalities. However, we can clearly see the similarities between our results and Raytrix results.

In a brief review, the good results obtained in both datasets (Andrea and Watch) makes us believe that we developed a quality and consistent method.

Chapter 5

Conclusions

Plenoptic cameras are definitively a theme to be followed. The fact that these cameras are able to acquire the plenoptic function and then use the redundancy created by a single 3D point being imaged several times makes them useful cameras to several applications like photography, robotics or microscopy. Depth estimation or the ability of refocusing an image after it has been taken are probably the most appreciated features of these type of cameras.

In this dissertation, we presented a novel approach to perform a depth estimation from a raw image captured by a multi-focus plenoptic camera. It is not a fully automatic algorithm, since one threshold cannot be automatically tuned, however, it is able to fully automatically search for salient points and find their correspondences. They allow the algorithm to back trace the light rays that come from the micro lens image and intersect them in the virtual space, which is where a "real world" 3D point is imaged by the main lens of a plenoptic camera. However, as some found correspondences are affected by noise, producing bad results, we developed a COMSAC method to obtain robust results, eliminating the outliers and producing a sparse depth map. Since the density of the salient point cloud is high enough, the algorithm is also able to transform the sparse depth map into a complete dense depth map, by performing a random growing.

The developed algorithm was tested using a dataset from Raytrix that is publicly available, and the obtained results were good, as different objects and their depth can be identified. We were able to find similarities on the results, when compared to Raytrix ones, which makes us believe that our method is good and consistent, as we also tested our method on a different and challenging dataset. As stated before, the algorithm to depth estimation from Raytrix is not publicly available, which lead to a process of reverse engineering in order to develop our own

method.

However, and despite what has been said about the concern on the efficiency of the algorithms, the computational time is the most important and noticeable limitation. With the increased computational power and all the parallel processing techniques, we want good results but as fast as possible. Real time, if possible. Graphic cards are an excellent example that may help to overcome the time limitation, as they already have their own processing units, called GPUs, specialized in graphics processing, being CUDA, from Nvidia, the main example on this parallel processing techniques. However, the first improvement it has to be done at the implementation level. During the development of the depth estimation method, the efficiency of the algorithm was taking in consideration (despite not being the main concern), as we were always trying to produce code in the most efficient way possible.

In summary, the developed method can and should be improved. Not only at a programming level (parallel programming, for instance) but also on a processing level, with the use of CUDA devices being one of the most used methods to improve the speed of algorithms.

As future work, and obviously considering the improvement of the algorithm as it, we intend to make this a fully automatic method. We also intend to enhance our algorithm by augmenting the number of searching profiles, as different profiles produces different results. Additionally, we want to use our depth map in order to synthesize an all-in-focus image.

Finally, we want to port our method to standard plenoptic cameras, like the Lytro ones. This requires some modifications on the algorithm, but it is a work that we think that has do be done.

References

[1] A. Agrawal and S. Ramalingam. Single image calibration of multi-axial imaging systems. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1399–1406. IEEE, 2013. 1.3

[2] A. Agrawal, Y. Taguchi, and S. Ramalingam. Beyond alhazen's problem: Analytical projection model for non-central catadioptric cameras with quadric mirrors. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2993–3000. IEEE, 2011. 1.3

[3] T. E. Bishop and P. Favaro. Plenoptic depth estimation from multiple aliased views. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1622–1629. IEEE, 2009. 1.3, 2.2.5

[4] T. E. Bishop and P. Favaro. The light field camera: Extended depth of field, aliasing, and superresolution. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(5):972–986, 2012. 1.3

[5] L. B. Don Dansereau. Gradient-based depth estimation from 4d light fields. *Dept. of Electrical and Computer Engineering, University of Calgary, Alberta, Canada*, 2004. 1.3

[6] F. Dong, S.-H. Ieng, X. Savatier, R. Etienne-Cummings, and R. Benosman. Plenoptic cameras in real-time robotics. *The International Journal of Robotics Research*, 32(2):206–217, 2013. 1.3, 2.2.5

[7] S. Friedberg, A. Insel, and L. Spence. Linear algebra. *Third ed., Prentice-Hall*, 1997. 3.2.3

[8] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '96, pages 43–54, New York, NY, USA, 1996. ACM. 2.1

[9] M. Levoy and P. Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996. 2.1

[10] M. Levoy, R. Ng, A. Adams, M. Footer, and M. Horowitz. Light field microscopy. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 924–934. ACM, 2006. 1.3, 2.2.5

[11] G. Lippmann. Epreuves reversibles, photographies integrales. *Academie des sciences, 446451*, 1908. 1.3, 2.2

[12] R. Ng. *Digital Light Field Photography.* PhD thesis, Stanford University, 2006. 1.3

[13] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light Field Photography with a Hand-held Plenoptic Camera. *Computer Science Technical Report CSTR. 2.11 (2005) - Stanford University*, 2005. 1.3, 2.2.4

[14] C. Perwaß and L. Wietzke. Single lens 3d-camera with extended depth-of-field. In *IS&T/SPIE Electronic Imaging*, pages 829108–829108. International Society for Optics and Photonics, 2012. 1.1, 1.3, 2.2.2, 2.2.2, 2.2.2, 2.2.3, 3.2.2

[15] D. Reddy, J. Bai, and R. Ramamoorthi. External mask based depth and light field camera. In *Computer Vision Workshops (ICCVW), 2013 IEEE International Conference on*, pages 37–44. IEEE, 2013. 1.3

[16] Y. Taguchi, A. Agrawal, A. Veeraraghavan, S. Ramalingam, and R. Raskar. Axial-cones: modeling spherical catadioptric cameras for wide-angle light field rendering. *ACM Transactions on Graphics-TOG*, 29(6):172, 2010. 1.3

[17] S. Tambe, A. Veeraraghavan, and A. Agrawal. Towards motion aware light field video for dynamic scenes. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1009–1016. IEEE, 2013. 2.2.5

[18] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar. Picam: An ultra-thin high performance monolithic camera array. *ACM Trans. Graph.*, 32(6):166:1–166:13, Nov. 2013. 1.3

[19] S. Wanner and B. Goldluecke. Globally consistent depth labeling of 4d light fields. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 41–48. IEEE, 2012. 1.3, 2.2.5

[20] Z. Yu, X. Guo, H. Ling, A. Lumsdaine, and J. Yu. Line assisted light field triangulation and stereo matching. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2792–2799. IEEE, 2013. 2.2.5

[21] Z. Yu, J. Yu, A. Lumsdaine, and T. Georgiev. Plenoptic depth map in the case of occlusions. In *IS&T/SPIE Electronic Imaging*, pages 86671S–86671S. International Society for Optics and Photonics, 2013. 1.3, 2.2.5