POLITECNICO DI TORINO

Facoltà di Ingegneria

Master Degree Course in Mechatronic Engeneering

Master Thesis

Analysis and characterization of a plenoptic camera for industrial applications

Advisor

Prof. Bartolomeo Montrucchio

Co-Advisor

Dott. Luca Ulrich

Candidate

Edoardo Maria Ferrero

April 2021

Contents

Introduction

Conventional cameras are instruments to get images of real objects. The photography is gotten collecting on a film, through a lent, the light spread by the objects of the scene. In the digital cameras, that last is substituted by an array of sensors (CCD) which allow to measure the amount of light for each pixel.

In order to understand the physics behind the plenoptic camera, we have to define: the focus and the depth of field.

In geometrical optics, a focus, also called an image point, is the point where light rays originating from a point on the object converge; the depth of field (DOF) is defined as the distance between the nearest and the farthest objects that are in acceptably sharp focus in an image.

In the conventional cameras in order to increase the depth of field, we need to reduce the diaphragm, however in this way we reduce the resolution of the image.

The evolution of those digital cameras is the plenoptic camera. It allows to increase the depth of field without any depletion of the diaphragm. With one shot, we are able also to refocus the gotten imagine, or increase the depth of field, or change the point of view without reducing the resolution.

So this camera is able to re-capture the light distribution entering from the world, that in the conventional cameras is lost: so the capability of this camera is to measure not just our 2D photographs of the total amount of the light at each point on the photosensor, but also the 4D light field through the computation of the amount of light traveling along each ray that intersects the sensor.

The aim of capturing the additional two dimensions of data allows us to apply ray-tracing techniques to compute synthetic photographs flexibly from the acquire light. The overall concept is to re-sort the rays of light to where they would have the terminated if the camera had been configured as desired.
Plenoptic camera, or light field camera,in order to acquire 4D light field of a scene, is provided with an array of individual lenses. The arrangement of the lens means that the multiple light rays are linked to each sensor pixel and synthetic cameras can compute the information. The physics linked to the plenoptic camera is explained in details in the following chapters.

The aim of the work is to understand the most important physical principles linked to plenoptic camera and the estimation of the accuracy in practical applications using a 3D chessboard designed throught Solidworks software and printed throught 3D printer.

Chapter 1

Plenoptic Camera

1.1 Light Field Representation

The light field, as defined by Gershun [1] in 1936, describes the radiance travelling in every direction through every point in space. Mathematically it can be described by a 7D function which is called *plenoptic function*.

To obtain such a function, we must measure the light rays at every possible location (x, y, z), from every possible angle (θ, ϕ), at every wave length γ and at every time t.

As said before, the plenoptic function is then a 7D function denoted as $L(x, y, z, \theta, \phi, \gamma, t)$ (see Fig. 1.1). However, such high dimensional data is difficult to record and handle in practice. Thus, the light field model has been simplified twice for practical usage. In the first simplification, the measured function is assumed to be monochromatic and time-invariant. The wavelength γ of each light ray is record independently in different colour channels and for a dynamic light field the time sequence t can be recorded in different frames.

The second simplification was made by Levoy and Hanrahan [2] and Gortler et al [3], who realized that the 5D representation still contained some redundancy and could be reduced to 4D by assuming that the light field was measured in free space.

In such cases, light ray radiance remains constant along a straight line, making one dimension redundant in the 5D plenoptic function.

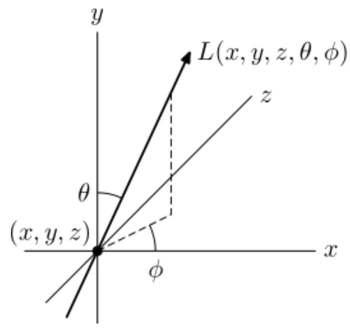


Figure 1.1: Plenoptic function in 5-dimensions of light.

When parameterising a 4D light field [2], there are three key issues: computational efficiency, control over the set of rays, and uniform sampling of the light field space. Based on the issues, the most common solution to the representation of a 4D light field is to parameterise the light rays by the coordinates of their intersections with two planes placed at arbitrary positions. The coordinate system is represented by $(u,\ v)$ for the first plane and $(s,\ t)$ for the second plane. The defined system first intersect the uv plane at coordinate $(u,\ v)$ and then intersect the st plane at coordinate $(s,\ t)$, and is thus denoted by $L(u,\ v,\ s,\ t)$. Thus, the planoptic function that describe a light field is reduced from 7 to 4 dimensions, and parameterised by four coordinates $(u,\ v,\ s,\ t)$. This type of representation is named *light slab* (see Figure 1.2).

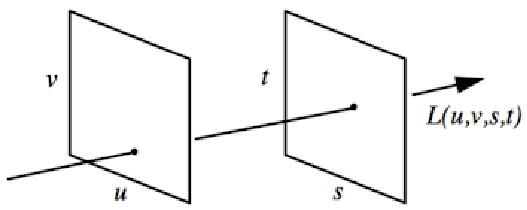


Figure 1.2: Light slab representation.

In this representation one of the planes may be placed at infinity. This is useful since the lines may be parameterised through a point and a direction. This is convenient for constructing light fields either from orthographic images or images with a fixed field of view. The efficiency of geometric calculations is the big advantage of this representation. Mapping from (u, v) to points on the plane involves only linear algebra (multiplying by a 3x3 matrix).

1.2 History of Plenoptic Camera

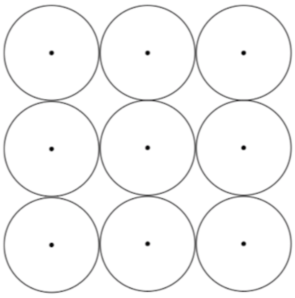
The introduction of the integral photography happened in the early 20^{th} Century by Lippman [4], who proposed the use of a microlens array to capture the full radiance of a scene. This design was approved, then, by Ives [5] in 1928 who added, in order to improve the sharpness of the image, an objective lens .
Many new opportunities become available to investigate integral photography, with the invention of digital photography, such as Adelson and Wang [6], who in 1990 placed on the focal plane of the cameras main lens a lenticular array in order to estimate the depth of the scene from a single image. They propose that using a plenoptic camera because, compared to a binocular stereo system, it improved the reliability of the depth estimation. The reason is because informations about both vertical and horizontal parallax are available and greater number of views are recorded.
Then, in 2005 Ng [7] improved this design, introducing new digital processing techniques such as digital refocusing and the extension the depth of field.
Despite being the basis of most current plenoptic camera, this design was faulty, as the special resolution was dependent on the number of micro images, this is typically too small for most applications.
The spatial resolution was improved by Georgive and Lumsdaine [8] with the introduction of the focused plenoptic camera. It has been improved by creating a relay

system between the microlens array and the objective lens in order to decouple the dependency of a resolution from the number of microimages. The focused plenoptic camera has been developed further iin order to comprise the microlens with varying focal lengths interwoven into each other in order to have a better depth resolution. Plenoptic image formation and manipulation require substantial computation, and through the developments of algorithms many improvements have been made.
The first algorithms was developed by Levoy et al. for light field rendering [2]. Further Isaksen developed the light field image rendering method in order to extend their utility by advancing the image based rendering algorithms through the use of a new parameterisation method [9]. Digital refocusing was improved by Ng by deriving algorithms to process in the Fourier space [10]. Also the spatial resolution was improved by developing new super-resolution algorithms [8, 11, 12] . Into image blending works have been performed in order to create a more natural blur of out of focus planes in a rendered image [13]. In addition efforts have been performed in order to reduce the artefacts in a plenoptic camera [14]. Through the manipulation of the colour demosaicing process, spacial resolution has also been developed [15]. The full resolution was developed by Favaro and Bishop by using multiple aliased views [16].

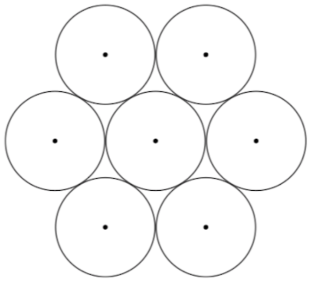
1.3 Micro Lens Array

Light field or plenoptic cameras allow to capture the 4D light field with 2D image. As it said before to capture a light ray, we need to planes and considering the two-plane parameterisation, they can be applied in practice with an image sensor and a micro lens array.
The light ray starts from the source and it goes into the micro lens array. The light ray is projected into the image plane through the lens, giving different perspectives from the same source.

The micro lens array can have different configurations in term of lens organisation, two of the most common are: the orthogonal configuration and the hexagonal configuration (see Fig. 1.3).



(a) *Orthogonal configuration*



(b) *Hexagonal configuration*

Figure 1.3: Micro lens configurations.

We are more interested in the hexagonal arrangement because it is the configuration used by Raytrix, and in our work we uses plenoptic image provided by them.
The main advantage of the hexagonal configuration is a better coverage from the image plane. Instead, the orthogonal configuration have bigger gaps between lenses.

1.4 f-Numbers and lens characteristic

The directional resolution relies not just on the clarity of the images under each micro lens, but also on their size. The micro lens should cover as many photosensor pixels as possible.
The relative sizes of the main lens and micro lens apertures should be chosen so that the images result as large as possible without overlapping. This occurs when the two f-numbers are equal, as shown in the ray diagram below (see Fig. 1.1).

If the main lens' f-number is higher (i.e. the aperture is smaller with respect to its

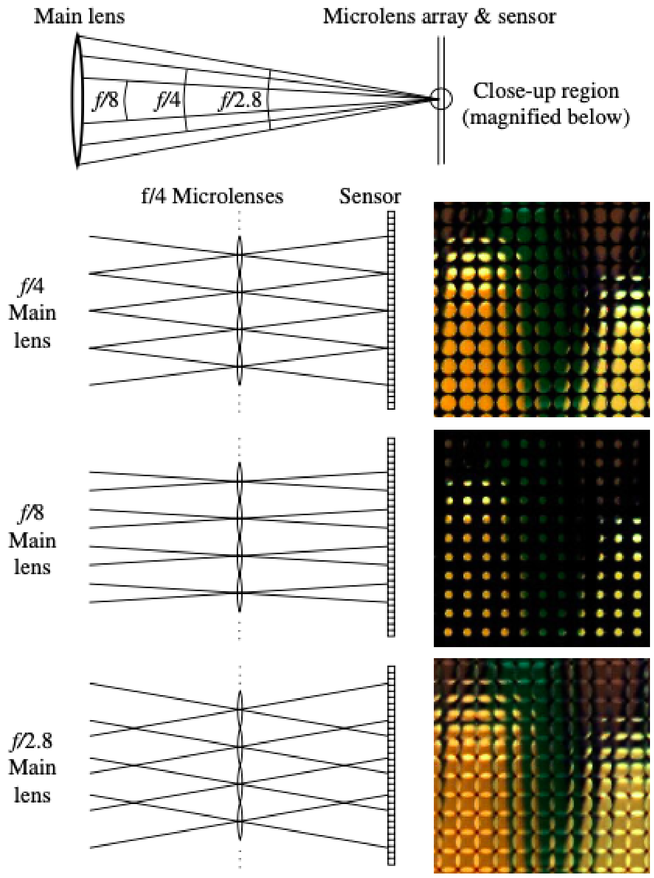


Figure 1.4: The different matches between main lens and micro lens f-numbers.

focal length), then the images below each micro lens are cropped, this means that many pixels are black, and resolution is wasted. On the contrary, when the main lens' f-number is lower (i.e. the aperture is larger), the images under each micro lens overlap contaminating each other's signal through "cross-talk".

It can be calculated from the following equation, where F is the f-number, f is the focal length of the lens and D is the diameter of the lens.

$$F = \frac{f}{D} \tag{1.1}$$

Figure 1.5 shows the projection of an object at distance a_L in front of a thin lens and the focused image formed at distance b_L behind the lens.

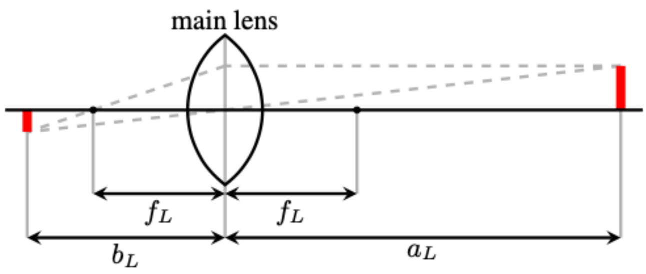


Figure 1.5: Optical path representation of a thin lens.

As given in eq (1.2), *the thin lens equation* defines the relationship between the object distance a_L and the image distance b_L.

$$\frac{1}{f_L} = \frac{1}{a_L} + \frac{1}{b_L} \tag{1.2}$$

In eq (1.2) f_L (that is f of the eq (1.1)) is the focal length of the main lens.

1.5 Synthetic photography equation

To understand the importance of the acquired light field to compute photographs, it is important to understand how an image is formed inside a conventional camera. The camera is modelled, for semplicity, in just four parameters: the aperture size and location, the depth of the lens (which are parallels) and the sensors plane.

Now the synthetic light field L' concept can be introduced. It is parametrised by the synthetic $u'v'$ and $s't'$ planes shown in Figure 1.6, such that $L'(u', v', s', t')$ represents the light travelling between the synthetic aperture plane (u', v') and the

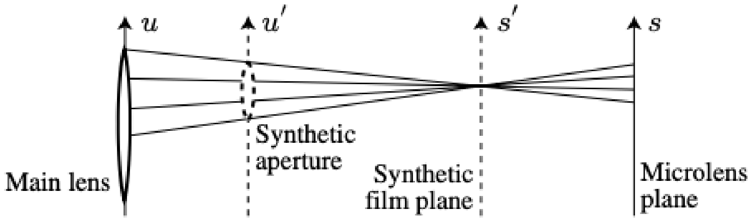


Figure 1.6: Synthetic photography conceptual model. The u and s planes figure the physical surfaces in the light field camera. u' represents a virtual plane containing the synthetic aperture in dotted line while s' symbolizes the synthetic film plane. Together they are a synthetic camera.

synthetic film plane (s', t').

With this definition, as the physics literature demonstrates [22], the irradiance image value which appeares on the synthetic film plane is given by:

$$E(s', t') = \frac{1}{D^2} \int \int L'(u', v', s', t') A(u', v')\ cos^4\theta\ du\ dv \tag{1.3}$$

In the equation above, D represents the separation between the film and the aperture, A is an aperture function (e.g. the sensor is loaded within the opening and zero outside it), and θ is the incidence angle that ray (u', v', s', t') forms with the film plane.

In order to eliminate the $cos^4\theta$ term, it is invoked a partial approximation, and further simplification are made in the equation, as ignoring the constant $1/D^2$. The final result is shown in eq. (1.4).

$$E(s', t') = \int \int L'(u', v', s', t') A(u', v')\ du\ dv \tag{1.4}$$

The equation above can be express also in terms of the acquired light field, $L(u, v, s, t)$. The diagram below illustrates the relationship between L and L'.

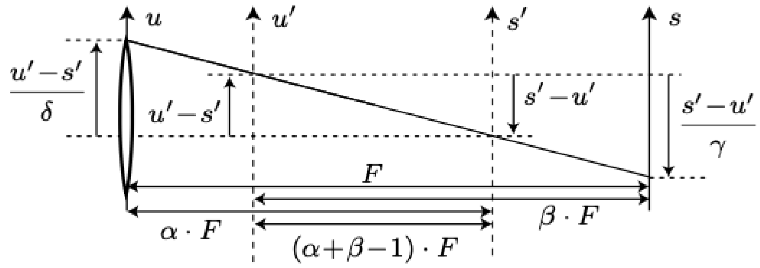


Figure 1.7: Relationship between L' and L.

From [23], for notational convenience, we are able to define two parameters:

$$\gamma = \frac{\alpha + \beta - 1}{\alpha} \quad \text{and} \quad \delta = \frac{\alpha + \beta - 1}{\beta} \tag{1.5}$$

Figure 1.7 shows that the ray does not intersects only u' and s' but it intersects also the u plane at $s' + (u' - s')/\delta$ and the s plane at $u' + (s' - u')/\gamma$. Consequently,

$$L'(u', v', s', t') = L(s' + \frac{u' - s'}{\delta}, t' + \frac{v' - t'}{\delta}, u' + \frac{s' - u'}{\gamma}, v' + \frac{t' - v'}{\gamma}) \tag{1.6}$$

Applying eq (1.6) to eq (1.4), it gives as result the Synthetic Photography Equation. It is used as the basis of image formation:

$$\bar{E}(s', t') = \int\int L(s' + \frac{u' - s'}{\delta}, t' + \frac{v' - t'}{\delta}, u' + \frac{s' - u'}{\gamma}, v' + \frac{t' - v'}{\gamma})A(u', v')\ du\ dv \tag{1.7}$$

1.6 Optical transformation

Following the developments in Refs. [24] and [25], the radiance at a given plane perpendicular to the optical axis is represented by $r(q, p)$, where q and p are the position and the direction in ray space. In it, for this reason, any coordinate can be represented by $x = (q, p)^T$ in the ray space.
Using the notation just introduced, the light field is the radiance as a function of ray space, $r(x)$. If we consider now an arbitrary ray transfer matrix A, each ray becomes:

$$x' = Ax \tag{1.8}$$

The refraction due to the lens and travel of rays in free space are described by the two matrices L and T:

$$L = \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \text{ and } T = \begin{bmatrix} 1 & t \\ 0 & 1 \end{bmatrix} \tag{1.9}$$

If in the equation (1.8) A represents the optical transformation, considering the transfrmation of $r(x)$ to $r'(x)$, since in the optical transfer matrices the $det(A) = 1$ and assuming that the entire optical system follows the convertion property, $r'(x') = r(x)$.
Applying the last concept in equation (1.8), then $r'(Ax) = r(x)$. Considering a ray in the form $y = Ax$, we obtain $r'(y) = r(A^{-1}y)$ [26], but since y is an arbitrary ray, the equation representing the radiance transformation turns out to be as:

$$r'(x) = r(A^{-1}x) \tag{1.10}$$

For simplification, the light field function $L(u, v, s, t)$ can be represented as radiance with spatial component q and angular component p so that:

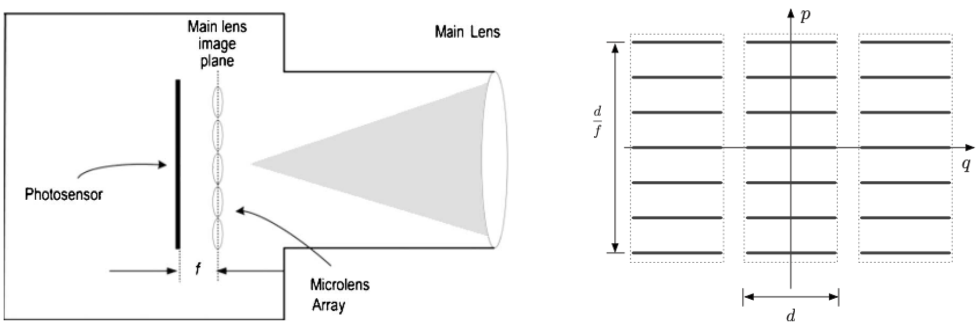
$$L(u, v, s, t) = r(q, p) \tag{1.11}$$

In the sensor each pixel receives light from all directions. At a given spatial point the intensity of an image, indicated as $I(q)$, is the integral of the radiance over all the rays incident at that point, [26]

$$I(q) = \int r(q, p)\ dp \tag{1.12}$$

1.7 Plenoptic camera 1.0

The traditional plenoptic camera is composed by an array of microlenses at the image plane of the main camera lens, where the sensor are placed behind the microlenses at a certain focal length (fig 1.8(a)). In front of the microlenses the radiance is sampled by the camera with a kernel as shown in Fig. 1.8(b).



(a) *Conventional plenoptic camera* (b) *Sampling of the radiance $r(q, p)$ in 2D (q, p) plane*

Figure 1.8: Plenoptic camera 1.0

Each microlens image is formed by vertical stack of samples in the (q, p) plane,

which capture at the image plane the angular distribution of the radiance.
For each pixel, the energy of rays that come at an angle specific is measured by the microlens, and in front of it, at one focal length, the rays pass through a plane [26]. In order to see it, the matrix A and A^{-1} representing the incident rays on a plane at one focal length in front of a given microlens are identified [26].

$$A = \begin{bmatrix} 1 & f \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 1 \end{bmatrix} \begin{bmatrix} 1 & f \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & f \\ -\frac{1}{f} & 0 \end{bmatrix}; \quad A^{-1} = \begin{bmatrix} 0 & -f \\ \frac{1}{f} & 0 \end{bmatrix} \tag{1.13}$$

By considering eq (1.13), on the sensor a pixel replied approximately with equally rays for all the angles. Therefore, in ray space the kernel is showed as a vertical line as thick as the pixel in Figure 1.9.

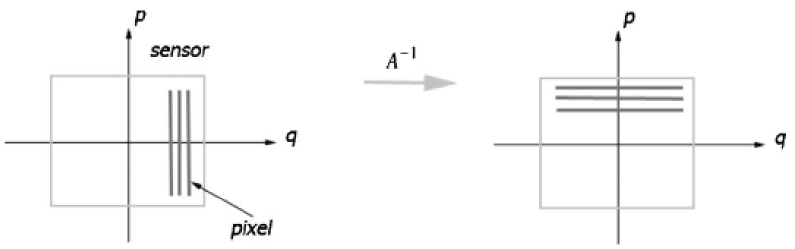


Figure 1.9: Sampling pattern of one microlens in Plenoptic 1.0 camera.

The Matrix A^{-1} transforms the vertical line to an horizontal one because an input p does not influence an output p, due to the bottom right zero matrix element. Morover the diameter of the microlens limits the spatial size of that horizontal line. At a specific spatial point, integrating all the angular samples, the images captured by the traditional plenoptic camera are rendered from the radiance. However the single microlens samples each spatial point, consequentially the integration is necessary in the rendering for all the pixels in each microimage. As described, from the traditional plenoptic camera rendering produces only 1 pixel per microlens, consequently the rendered image will result with a very low resolution [26].

1.8 Plenoptic camera 2.0

The focused plenoptic camera, or plenoptic camera 2.0, bases its functioning on an array of microlenses focused on the image plane of the main lens, as shown in Fig. 1.10 [26].

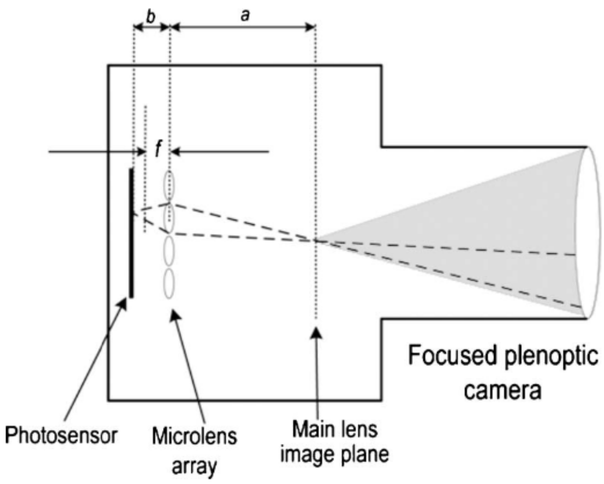


Figure 1.10: Focused plenoptic camera.

A portion of the image is captured by the microlens and this is formed on the main lens. We can think of the sensor like moved back from the main lens, so that to allow the formation of the image at some distance a in front of the microlenses. The microlenses works as an array of real cameras, they are able to reimage parts of that image into the sensor.
In thius way, a relay imaging is formed with the main camera lens by every microlens.
Considering the following lens equation $1/f_L = 1/a_L + 1/b_L$, b is greater than f [26] in the setup of the camera.

In the (q,p) plane samples of each microlens image is stack, so that we are able to capture both positional and angualr distribution of the radiance at the image plane.
The total transfer matrix starting from the plane to the sensor is

$$A = \begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -\frac{1}{f} & 6 \end{bmatrix} \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -\frac{b}{a} & 0 \\ -\frac{1}{f} & -\frac{a}{b} \end{bmatrix} \tag{1.14}$$

The last equality represents the refocusing. Computing the inverse

$$A^{-1} = \begin{bmatrix} -\frac{a}{b} & 0 \\ \frac{1}{f} & -\frac{b}{a} \end{bmatrix} \tag{1.15}$$

Here we have to consider that due to the zero top right element, after the inverse mapping, the sampling kernel for each pixel remains vertical in optical phase space. Consequently, a dense set of thin vertical kernels are the results of the sampling, and it is decoupled from microlens size (see Fig. 1.11) [26].

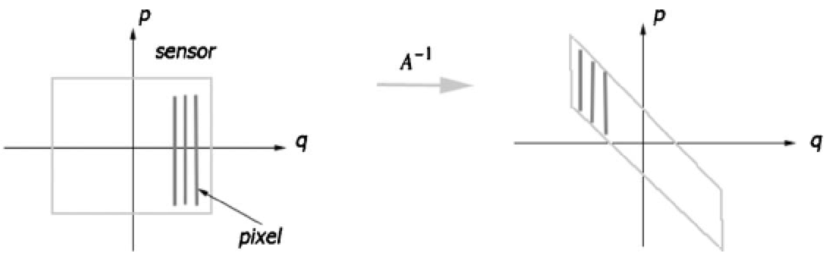


Figure 1.11: Sampling pattern of one microlens.

The most important result of the consideration done before is that for the focused plenoptic camera the spatioangular trade-off for is not fixed by the number of microlenses. In fact the optical geometry (a and b) is the one which determines the spatioangular trade-off.

Like in the traditional plenoptic camera, integrating the angular samples at each space point, we obtain a rendered image as result of the radiance captured (see Fig. 1.12).

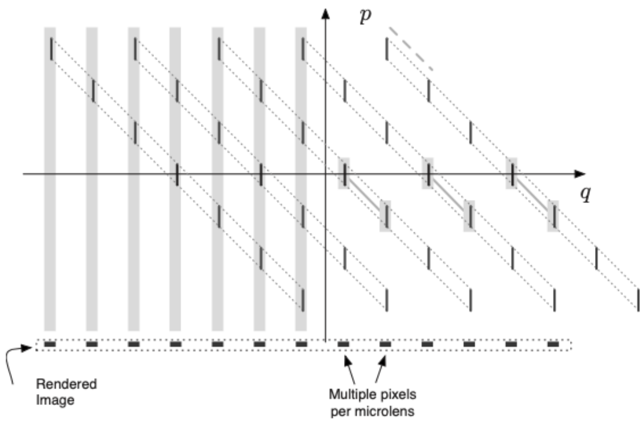


Figure 1.12: Image rendering through focused plenoptic camera. The left half shows the rendering towards all directions considering a given position; the right half shows the rendering towards a single direction at each position.

In contrast with the traditional plenoptic camera, different microlens samples the given spatial points. In addition, rendering data coming from this type of plenoptic camera includes integrating across microlens images rather than within microlens images [26].

1.9 Focusing in plenoptic camera 2.0

In general the attainable resolution of an image rendered from focused plenoptic camera depends on the depth of the scene. As derived in [25], the spatial resolution of a rendered image is equal to b/a times the spatial resolution of the sensor. Image planes closer to the microlens plane raises the resolution, or equivalentely, planes closer to the main lens encreases it.

As with the conventional plenoptic camera, the focuses plenoptic camera renders an image according to eq.(1.12). By evaluating $r(q, p)$ at some particular value of

$p = p_0$, i.e., let $I_0 = r(q, p_0)$, we are able to implement single viewpoint rendering of eq.(1.12). In order to render the final image at each microlens we consider a range of spatial sample corresponding to a range direction (see Fig. 1.12, where M samples is 2) instead of selecing spatial sample which corresponds to a single value of p.

Selecting a contiguous set of pixels (a patch) from every microimage and tiling all such patches together into a final image, we are able to render the corresponding image of a given view. The pixel size of the patch to select is the meaningful parameter to consider.

In the image showed in Fig. 1.13, where μ represents the distance among microlenses (the pitch of the microlens array), the main lens image plane is divided into μ x μ sections such that each of them is mapped in portion, called patches, of size equal to M x M. Putting together those M x M patches, we are able to recontruct the main lens image [26].

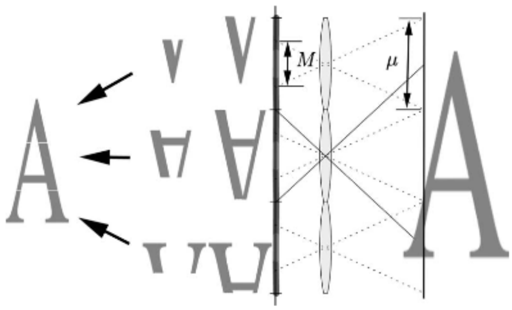


Figure 1.13: Capturing geometry.

However, there is an alternative interpretation about the way in which an image is captured [26]. For a rendering pitch, there is an image plane in front of the microlens placed a certain distance s that will satisfy the relation $\mu = M(a/b)$. In that case the plane results "in focus" only with the patch with a certain size. In other words, the algorithm operates as is showed in Fig 1.14. In particular the pitch and the squares of each microlens image are selected. By tiling the selected

squares all together, we get the rendered final image.
In this way selecting one pitch rather than an other, we put different world planes "in focus".

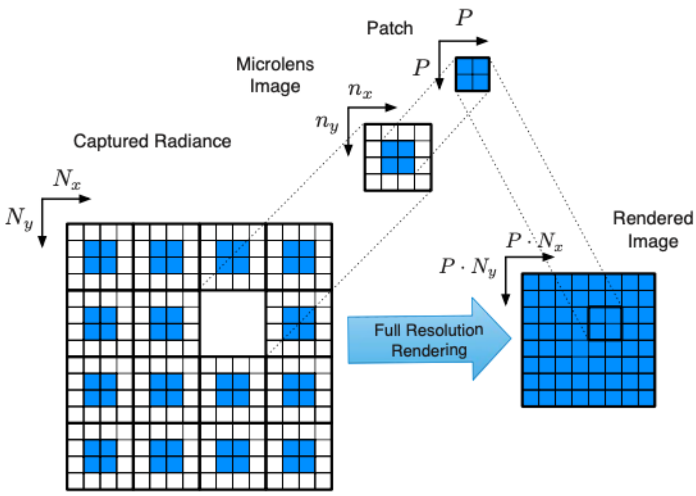


Figure 1.14: Scheme of the algorithm.

1.10 Refocusing in plenoptic camera 2.0

As said in Paragraph 1.1, there are two additional dimensions of a 4D light field compared to a conventional 2D image. These allows us to produce such effects like the refocusing even after a photo has been taken, as long as the depth image has been computed.
In particular, the plenoptic camera 2.0 requires an accurate depth estimation in order to reconstruct in a mieaningful way an image from the captured light field. The depth reconstruction is possible thanks to partial redundant information coming from the data stored by the camera. In particular, an object tends to have similar appearance if obsterved in slightly different viewing angles. If this redundancy can be used to compare the data coming from the camera, it has exploited in order reconstruct the very spars data [27].

The classical parameterization (u, v, s, t) uses distinctive planes aligned with the optical axis z: $[u, v]^T$ represents the coordinates on the plane at the main lens (ML) z_{UV} and $[s, t]^T$ are the coordinates place on the focal plane of the main lens. Since the points on the focal plane are mapped on the image plane, the coordinates $[s, t]^T$ represent also the coordinates on the image plane at z_{ST}.

The light field can be used to calculate radiance for a new plane (q, r) places at distance z_{QR}, parallel to the (s, t) plane. The mapping to the original coordinates requires only the computation of the intersection of the ray, originating at $[q, r, z_{QR}]^T$ with direction $[u, v, z_{UV}]^T$ - $[q, r, z_{QR}]^T$ with the (s, t) plane at z_{ST}. The intersection point of the $[s, t]^T$ coordinates can be determined in two step: first, a scaling, α, of $[q, r]^T$ depending on the positions of the three planes (s, t), (q, r) and (u, v) is computed: $\alpha = \frac{z_{UV} - z_{ST}}{z_{UV} - z_{QR}}$.

Second a traslation equal to $\Delta(u, v) = -\beta \cdot [u, v]^T$ with $\beta = \frac{z_{QR} - z_{ST}}{z_{UV} - z_{QR}}$ brings us to the final coordinates in the (s, t) plane: $[s, t]^T = \alpha \cdot [q, r]^T + \Delta(u, v)$, see Fig.1.15.

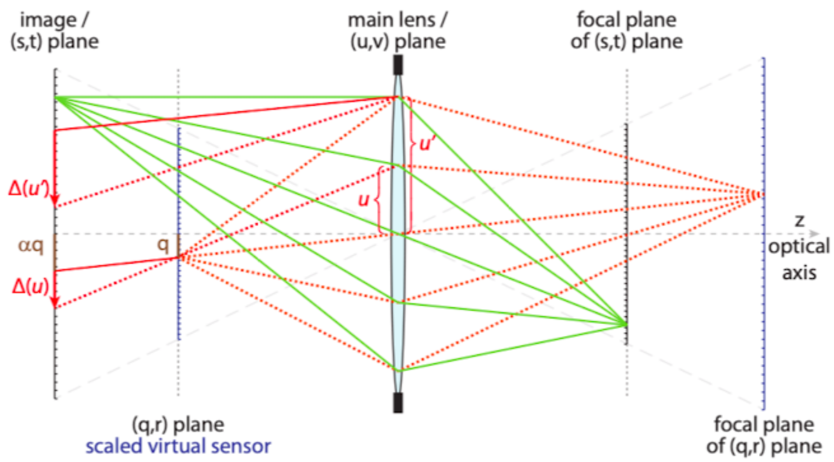


Figure 1.15: Refocus representation.

While a pinhole camera could have an infinitesimal aperture, the plenoptic camera would not produce any image because of the lack of light. So, cameras need to have a large aperture and they use lens to refocus the acquired rays. All the points inside the focal plane are projected to an exact location on the sensor; on the contrary, outside the focal plane the points can be projected to several locations. For what concerns the adjustment of the focal plane, this is a major challenge. We can imagine the light rays which moves away from the camera, all the rays of a given pixels will encounter on the focal plane. Retracing the rays in the opposite direction, it would mean integrated at that specific sensor pixel. We are able to perform this integration in a post-capture process through the 4D light fields [27].

In a common camera with a senor placed at the image plane (IP), the integration is performed over all directions, so, the (u, v) plane. Approximately, for a plenoptic camera 1.0, all the pixels under a microlens are summarize as: $L(s, t) = \sum_u \sum_v L(u, v, s, t)$. The focal plane placement is dependent to the distance of the IP to the ML. Assuming the thin lens model approximation, the original focal plane is placed at a distance equal to $d_{org} = (1/f - 1/(z_{UV} - z_{ST}))^{-1}$ from the main lens, where f represents the focal length of the main lens. If we consider a virtual movment of the image plane to an other plane, (q, r), this has the effect of the change of the focal plane. In particular, the new focal plane will be placed at a distance $d_{refocus} = (1/f - 1/(z_{UV} - z_{QR}))^{-1}$.

The most difficult aspect in the refocuse is that it requires an elevated number of different view points in order to reach a smooth out-of-focus blur.

In photography, the effect for which the camera renders an area out-of-focus is called *Bokeh*. This effect, for small and bright out-of-focus-light, is used often as stylisation method. The shape of the lens aperture determines, inderectly, the Bokeh effect. In a standard camera the lens aperture cannot be changed, for this reason photographers often add in front of the lens an additional aperture. The adding of it blocks the incoming light in certain directions. We are able to simulate this behaviour through the 4D ligh-field: $L'(u, v, s, t) = b(u, v)L(u, v, s, t)$ where b is a function which reproduces the aperture shape.

In practice, the pixels of the plenoptic camera do not coincide exactly to the rays. On the contrary, a pixel is able to record an incident ray within a small cone of directions. Every point related to the microlens coincides to an image taken with a lens with a limited aperture. These views shows a depth-of-field, consequently the depth-of-field range forced by the optics, limits each refocusing operation. Likely, with large apertures the captured light-field might not be enough and so the border pixels might be missing.

Chapter 2

Plenoptic depth map

Since we have limited informations provided about the internal configuration of the camera, it is treated as a balck-box system. In addition we are not able to access to the RAW image data of the camera, and the RAW images are processed using the Raytrix software (RxLive 4.0). Therefore, predictions have to be made about the algorithms used to produce the virtual depth.
Raytrix uses an hexagonal grid microlens array made up of three different focal length lenses interlaced into the array, which can reach a much greater depth of field (dependent on the focal lengths) reducing only the effective resolution ratio in each dimension to the half [18].

2.1 Reconstruction from multiple images

The depth algorithms are calculated via triangulation [18]. This requieres that the pixels need to be incident in at least two microimages if they belong to the same image point. Therefore, only in area with sufficient local contrast triangulation is possible to perform.
The recontruction of a scene for depth in a plenoptic imaging is made through

multiple images. In this section, we will explore how reconstruction is formed mathematically, by using a scene made entirely of points for mathematical simplicity and only two images.

In this case, we consider a set of correspondences $x_i \leftrightarrow x'_i$ in two images. Taking a set of three dimensional points, X_i, and two camera matrices, P and P', the correspondance image will be $PX_i = x_i$ and $P'X_i = x'_i$. Then the two data points are projected to the point X_i. However, since the point X_i or the projection matrices P and P' are unknown, they have to be determined.

In plenoptic imaging, often the depth is outputted in the form of virtual depth. We have to underline that it is impossible to determine the positions of points with precition without knowing anything about the calibration of the camera [20]. This ambiguity does not depend on the number of images given, since the informations about absolute orientation, scale or position of the imaged object are unknown. This ambiguity, of lack of informations about camera calibration, is represented as a $projective\ transformation$.

The projective transformation, H, can be applied to each point X_i and to the right of the camera matrix, P_j, without having projected image points altered, such that:

$$P_j X_i = (P_j H^{-1})(H X_i) \tag{2.1}$$

In these circumstances, the choice about the projective transformation is arbitrary as there is no reason to select one set of camera matrices and points over another. This reconstruction, therefore, having projective ambiguity, it will have something known as projective reconstruction. For this reconstruction, we must have seven points at least which must not lie in one of the various critical configurations [20]. The tool used to compute the points from the two images based its working on the use of the fundamental matrix, which can be thought of as if the constraint on image points x and x' would be imaged in 3D space. This is true because of the image points themselves, the coplanarity of the centers of the cameras on both views, and the space point. Given a fundamental matrix, F, and pair of matching points $x_i \leftrightarrow x'_i$ it must follow the equation (2.2) where F is a 3×3 matrix of rank 2 [20]:

$$x_i^{'T} F x_i = 0 \tag{2.2}$$

If F is unknown, it can be computed from a set of point in which we have correspondences where the equations are linear in the entries into matrix F. The fundamental matrix F can be entirely defined from a pair of camera matrices, P and P'. Alternatively, the pair of camera matrices, P and P', can help to determine the matrix F up to a 3D projective ambiguity. Accordingly, the 3D projective transformation does not change the fundamental matrix F, it keeps the full projective geometry of the camera matrices, P and P'.
One method through which the imaged scene is recontructed using the fundamental matrix, F, is described:

1. Using some correspondences between points so that $x_i \leftrightarrow x'_i$ from two views, then there are the composition of the linear equations in the entries of F from the coplanarity equation $x^{'T}\mathrm{F}x = 0$;

2. Compute the linear equations in order to find a solution to F;

3. Estimate a pair of camera matrices, P and P', from F;

4. Through the pair of camera matrices, P and P', and the corresponding image point pairs, $x_i \leftrightarrow x'_i$, we are able to find the point X_i in three dimensional space.

The method just described for solving a point in 3D space, X, is known as *triangulation*. As a plenoptic raw image can be considered as much two view representations, depth can be calculated through the application of this method to neighbouring microimages.

2.2 Calculating the Depth Map

As we can see from Figure 2.1, the virtual main lens image, which would happen behind the sensor, is projected on the sensor by each of the three middle micro lenses.

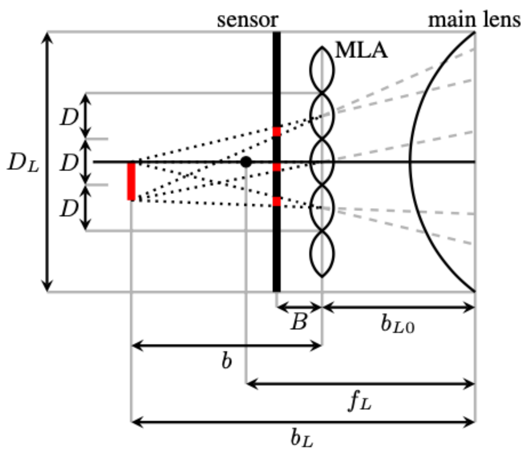


Figure 2.1: Raytrix camera optical path representation.

The virtual main lens image is shown from a lightly different prospective by each micro image, that is the image of a micro lens formed on the sensor. Considering the focused image of a point presents in two or more micro images, the distance between the virtual main lens image b and the MLA can be calculated by triangulation.
As Figure 2.2 shows, the distance b can be calculated by considering its projection in the two micro images.
In the figure below, p_{xi} (for $i \in \{1,2\}$) represents the distance of the points in the

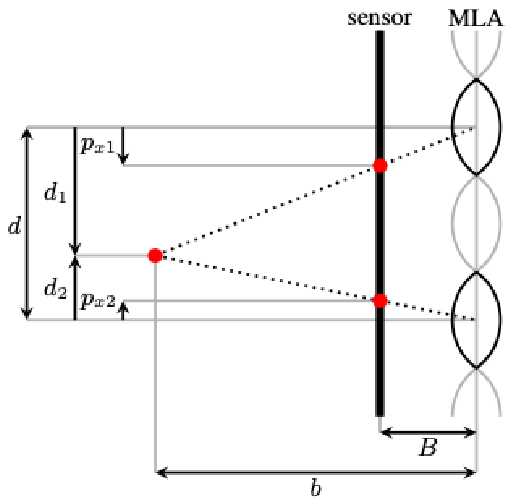


Figure 2.2: Optical path of a thin lens.

micro images to the principal point of the micro image correspondent. Moreover, d_i (for $i \epsilon$ {1,2}) determines the distance of the respective principal point with respect to the orthogonal projection of the virtual image point to the MLA.

Consequently, distances with an upwards pointing arrow, shown in Figure 2.2, are treated as positive values and those with an downwards pointing arrow are considered as negative values.

Triangles with equal angles are similar so we can consider the following relations:

$$\frac{p_{xi}}{B} = \frac{d_i}{b} \quad \rightarrow \quad p_{xi} = \frac{d_i B}{b} \; for \; i \; \epsilon \; \{1,2\} \tag{2.3}$$

Furthermore, the base line distance between the two micro lenses can be computed:

$$d = d_2 - d_1 \tag{2.4}$$

If we define the parallax of the virtual image point p_x like the difference between p_{x1} and p_{x2} from eq.(2.3) and (2.4), the definition in eq.(2.5) is reached.

$$p_x = p_{x2} - p_{x1} = \frac{(d_2 - d_1)B}{b} = \frac{d \cdot B}{b} \tag{2.5}$$

After readjusting eq.(2.5), the distance b between the MLA and a virtual image point can be characterize as a function of the base line length d, the distance between the sensor B and the distance between MLA, and the estimated parallax p_x , as given in eq.(2.6).

$$b = \frac{d \cdot B}{p_x} \tag{2.6}$$

According to the distance of its virtual image to the MLA, a point takes place in different micro images. In consequence, dependending on the distance, the length of the baseline d changes during the computation of the triangolation. If two neighbored micro images would be used during it, the baseline would match with the micro lens aperture ($d = D$).

Since Raytrix camera has two neighbored micro lenses with different focal lengths, they never focus the same point on the sensor. Thus, the baseline results to be always greater than the microlens aperture ($d > D$).

In addiction, since the distance B between MLA and sensor is not known precisely, the depth map, supplied by the plenoptic camera, is equal to the distance b divided by B. This relative depth value is called *virtual depth* and is represented by v.

From eq. (2.6), the virtual depth v is a function of the base line distance d and the estimated parallax p_x, as given in eq. (2.7).

$$v = \frac{b}{B} = \frac{d}{p_x} \tag{2.7}$$

The virtual depth, as said before, can be computed for a point only which is focused in at least two micro images. Because of the arrangement of the MLA (hexagonal) with three different focal lengths, like in a Raytrix camera, the minimum of the measurable virtual depth is equal to $v_{min} = 2$ [21].

2.3 Depth Accuracy

By considering the rules coming from the theory of propagation of uncertainty, we can understand how the depth accuracy is affected by an error of the estimated parallax. From the derivative of v with respect to the derivative of the measured parallax p_x, the standard deviation of the virtual depth v is equal to (see (2.8)) :

$$\sigma_v \approx \left|\frac{\partial v}{\partial p_x}\right| \sigma_{p_x} = \frac{d}{p_x^2}\sigma_{p_x} = \frac{v^2}{d}\sigma_{p_x} \tag{2.8}$$

From eq. (2.8) we are able to say that the proportionality of the virtual depth is equal to v^2 [21].

The baseline distance d is a discontinuous function of the virtual depth v, Since a maximum baseline length of a microlenses sees a point changes according to the virtual depth of that point, we can assert that the baseline distance d is a discontinuous function of the virtual depth v . This result conducts us to understand that the discontinuous dependency of the depth accuracy is a function of the object distance a_L.

The mathematical relationship between the virtual depth v and the image distance b_L is represented by the linear function given in eq. (2.9).

$$b_L = b + b_{L0} = v \cdot B + b_{L0} \tag{2.9}$$

Here the variable b_{L0} is unknown but it is a constant distance between MLA and main lens.

Through the thin lens equation (1.2), the object distance a_L can be expressed as a function of the virtual depth v.
If the derivative of a_L is computed with respect to b_L, the standard deviation of the object distance a_L can result as given in eq. (2.10)

$$\begin{aligned} \sigma_{a_L} &\approx \left|\frac{\partial a_L}{\partial b_L}\right| \sigma_{b_L} = \frac{f_L^2}{(b_L - f_L)^2}\,\sigma_{b_L} \\ &= \frac{(a_L - f_L)^2}{f_L^2}\,\sigma_{b_L} = \frac{(a_L - f_L)^2}{f_L^2}\,B\,\sigma_v \end{aligned} \tag{2.10}$$

We can further simplify eq. (2.10) when the object distances is much higher that the focal length of the main lens f_L. From eq. (2.11) we can see that in presence of a constant object distance a_L, the depth accuracy increases to f_L^2.

$$\sigma_{a_L} = \frac{(a_L - f_L)^2}{f_L^2}\,B\,\sigma_v \approx \frac{a_L^2}{f_L^2}\,B\,\sigma_v \quad for\ a_L \gg f_L \tag{2.11}$$

The depth accuracy decreases proportional to a_L when the focal length of the main lens is constant. Though, the depth accuracy as a function of the object distance a_L is a better than the given in eq. (2.11) because when the object distances are large, they are equivalent to small virtual depth.

2.4 Projection model

After the theoretical background for what concern the depth estimation, it is necessary to introduce the algorithms necessary to perform a robust automated camera calibration.
Figure 2.3 gives an overview about the projectional mechanism from the virtual depth behind the MLA to metric depth values in front of the camera.

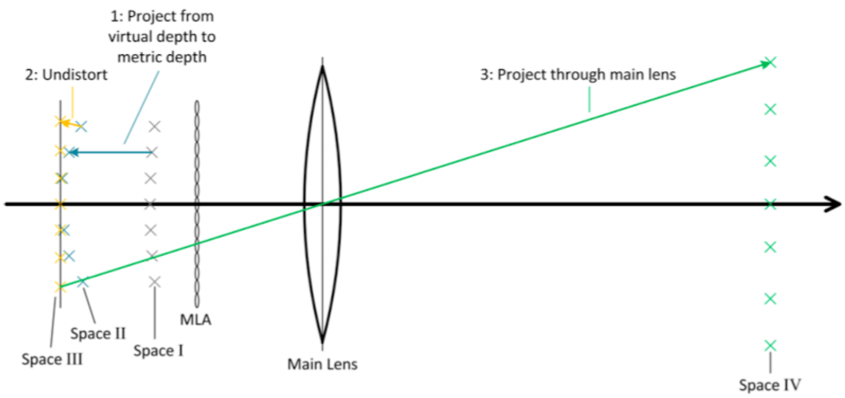


Figure 2.3: Representation from virtual to metric depth.

Points placed in space I are expressed in virtual depth units and lateral pixel positions on the sensor. Space II contains the points projected from space I and converted into metric coordinates. The points placed in space III are undistorted and are in metric coordinates depending on the principal plane of the main lens. Finally, on the other side of the main lens space IV is located. It contains the points in the object space. The coordinates relative to the sensor center of these points are metric [28].

The first step is the transformation from virtual depth values in space I z_I to metric depth values in space II z_{II}. In order to calculate this distance, we need to the definition of virtual depth:

$$v = \frac{b}{B} \tag{2.12}$$

Solving the equation through the metric distance b, the variables are replaced:

$$z_{II} = z_I \cdot B_i \tag{2.13}$$

Now every detected point in the target image, has a 3D position, which is compared to the main lens.

The effect of the lens tilt, in a plenoptic camera, affects the 3D image which results tilted. The effect is known as the Scheimpflug Principle [29], as illustrated in figure 2.4.

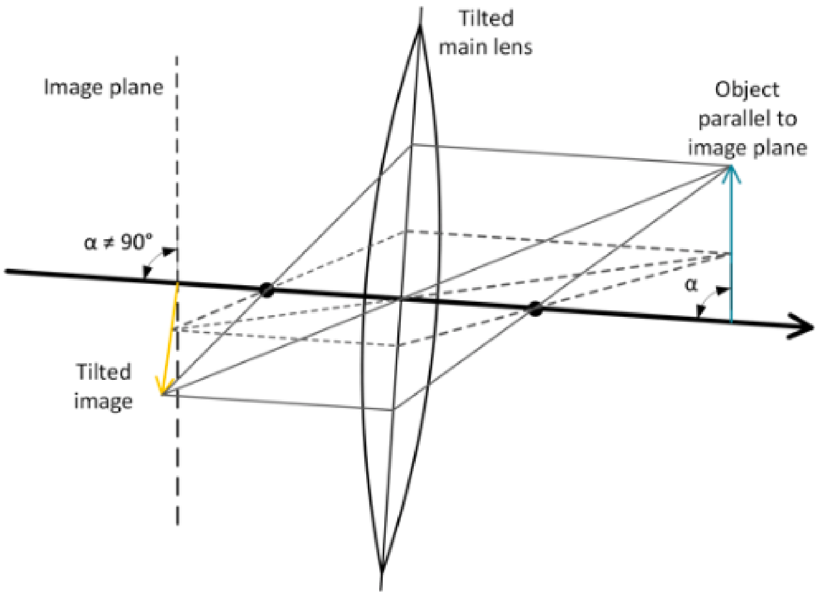


Figure 2.4: Tilt effect.

Through the 3D pose of the main lens, the algorithm model is able to model the tilt and shift the main lens in 3D. The paramerters θ_L, σ_L are introduced in order to controll the direction of the optical axis of the tilted main lens, along with the introduction of parameters X_L, Y_L in order to perform the shift of the main lens relative to the sensor center. The image distance B_L represents the distance between the main lens Z_L and the sensor [28].

The lateral undistortion model is applied after the removal of the tilt. The distortion coefficients k_1 and k_2 are able to controll the amount of distortion. The radius r, used in all of these computations, represents the lateral euclidean distance relative to the distortion center.

Around the optical axis, the distortion of the lens is considered radially symetric, so through the parameters X_L, Y_L we able to know the center of the distortion (see figure 2.5).

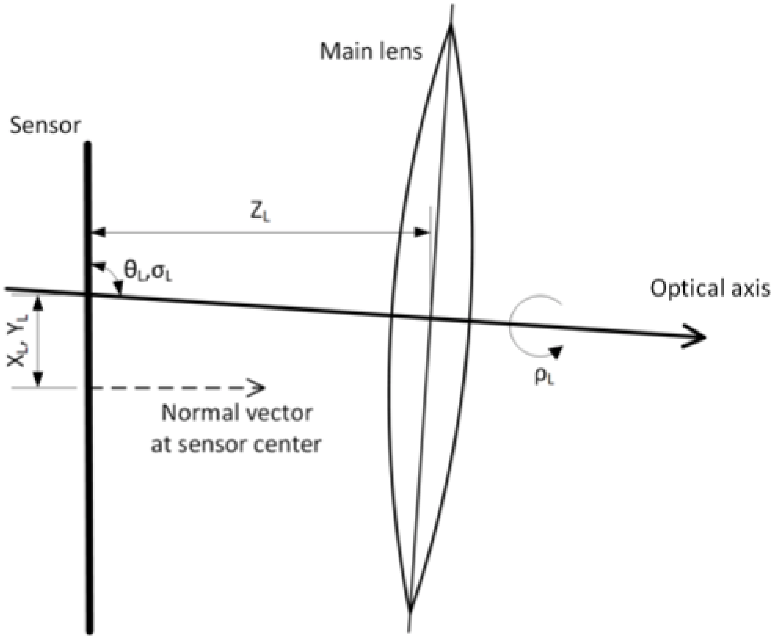


Figure 2.5: Tilted/shifted main lens representation.

By applying the method described by Brown, through the two following equation we are able to shift the lateral position of the points in order to count the radial distortion [30]:

$$\begin{aligned} x_{III} &= x_{II} \cdot (1 + k_1 r^2 + k_2 r^4), \\ y_{III} &= y_{II} \cdot (1 + k_1 r^2 + k_2 r^4), \end{aligned} \tag{2.14}$$

Following what said before, the radial depth undistortion method is applied. In the equation (2.15) the coefficients d_1 and d_2 mold the distortion according to the radius r, while the coefficient d_d tunes a linear relationship between the virtual depth of a point and the distortion strength [31]:

$$z'_{III} = z_{II} + (1 + d_d z_d) \cdot (d_1 r^2 + d_2 r^4) \tag{2.15}$$

Once that the points have been undistorted in order to perform the projection on the main lens, we need to calculate the image distance of each point. In order to perform it, we need to compute the image distance B_L of the main lens through the current values of the focal length f_L and the focus distance T_L :

$$B_L = \frac{T_L}{2}\left(1 - \sqrt{(1 - 4\frac{f_L}{T_L})}\right) \tag{2.16}$$

The image distance is measured from the total covering plane (i.e. TCP that is the plane on which the main lens has to be focused) of the plenoptic camera to the principal plane of the main lens. Thus, before adding the metric depth of the points to the image distance, we have to subtract the distance between the MLA and the TCP:

$$z_{III} = (z'_{II} - 2B_i) + B_L \tag{2.17}$$

From the plenoptic camera design theory, we know that the distance from the sensor to the MLA B_i is equivalent to the distance from the TCP to the sensor. So the distance from the TCP to the MLA is equal to $2B_i$. By subtracting the latter value, we are able to compute the distance from each point z_{III} with respect to the principal plane of the main lens [28].

Now through the main lens, the projection can be applied. The thin lens model is applied in order to calculate the distance z_{IV} to which a point is projected from the principal plane of the main lens:

$$\frac{1}{f_L} = \frac{1}{G_L} + \frac{1}{B_L} \tag{2.18}$$

2.5 Metric Calibration

Standard camera calibration techniques are performed with the aid of checkerboard targets with known geometry [32]. In the plenoptic cameras using this type of target, some problems can rise because of self-similar structures in the image. The computation of the depth estimation is no certain when an epipolar line of the microlenses is made parallel to an edge of the checkerboard. Through the use of circular targets, this can be avoided (see figure 2.6).

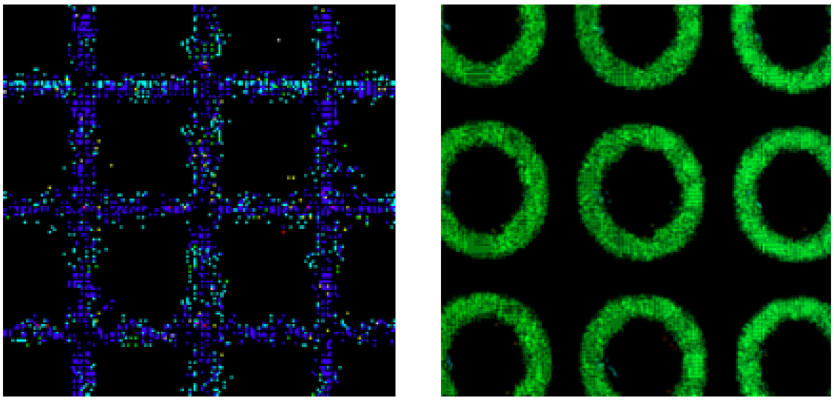


Figure 2.6: Linear or circular targets.

In the calculated total focus image, the circular features are detected from the light field by using OpenCV's MSER implementation. At this point, for each 2D position point lying on the target, the virtual depth is computed (compare figure 2.6 right). After this, a custom algorithm, using as seeds the 2D pixel positions of the circular, associates the correct metric distances given the pitch of the target and aligns the characteristic on a rectangular grid. This algorithm, described in greater detail in

[33], is designed with robustness.

At this point we have two sets of point clouds. The first cloud represents the points extracted from the 2D image, with a virtual depth values associated. The second cloud carries the points (called model points) aligned on the rectangular grid, which have a known constant distance.

Between the two sets of point clouds there is a one-to-one correspondance from a point on the rectangular target model to a point detected in the image. Now the target points are projected with initial parameters as described in paragraph 2.4. This projection happens in space IV. The points, with appropriate initial parameters, should lay close to their true position.

Since the model points is placed on a flat target, the sensor plane, on which they are, is $z_{IV} = 0$. Now in order to shift the projected target points into the sensor plane, an extrinsic pose is applied.

In this position, we are able to calculate the error function of an optimizing algorithm. Seven parameters are added for the extrinsic pose, for every image of the target with a different placement. tThe calibration is robust by adding the additional input data because this is a 3D-to-3D calibration. The stability of the optimization algorithm could be decreased by a recent addition to the lens model of a full lens pose [28].

Chapter 3

Experiment

In this chapter, we are going to introduce and explain the whole procedure adopted and that lead us to the results present in the last chapter of this work.
The chapter follows a sequential order, so any section corresponds to a step that brought us to the final results.

3.1 The 3D board

Since the final objective of this work is to test the plenoptic camera, we need of an object that includs the whole wanted aspects.
Mainly, we want to try out the accuracy of a single image taken by the camera with respect to the correspondent real value and the metric deviation among the whole images taken by the camera. The object used for all the tests is a board design by us.
The software used for the design of it is *Solidworks 2020 SP03* and the 3D printer model used to print physically the board it is *Prusa MK3S*.
The first step in order to obtain the final design was to decide the dimension of the board: since the print plate has a fixed maximum dimension available for printing, that is 210 mm, we adopted a quadratic shape with length 210 mm.

Another aspect that we have to take into account is that the 3D printer proceeds in layers in order to print the object, adding material till to reach the stablished measure coming from the file containing the board project.
The problem that could rise is that since the printer has a minimum layer height equal to 2 mm, if the wanted measure does not reach this threshold, the corresponding layer is not printed. The consequence of that is a different value of length between the theoretical board and the real one. So all the measures are decided also in order to satisfy this constraint.
In order to avoid the bending of the panel, an extrusion of 3 mm is been adopted for the entire board.
As said before, since we want to test the camera on different depth metric condition, we design the board with small parallelepipeds with side of 28 mm but different height. Between two parallelepipeds there is a distance of 1 mm.
The board has seven rows and seven column as it can be seen in Fig. 3.1.

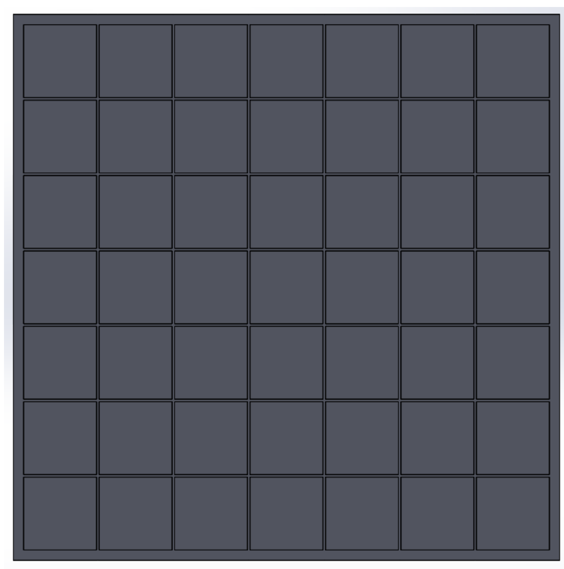


Figure 3.1: The board seen on the top.

Starting from the bottom right parallelepiped, it has an height of 3 mm (this measure is considered counting from the top of the extrusion) and it has a distance from the edge of the board of 4 mm. The other parallelepipeds on the same row are increased of 2 mm until the 4^{th} parallelepiped of the row that reach the maximum value for the row of 9 mm, after that the height decreases until the 7^{th} parallelepiped of the row that has the same height of the first, as shown in Fig.3.2.

Figure 3.2: Different height of the parellelepipeds (horizontal view)

As regards the parallelepipeds on the same column, starting from the bottom right parallelepiped, the parallelepipeds increase of multiplier 3 until the 4^{th}, then they decreases until the 7^{th} parallelepiped of the same column.
The final shape is shown in Fig.3.3

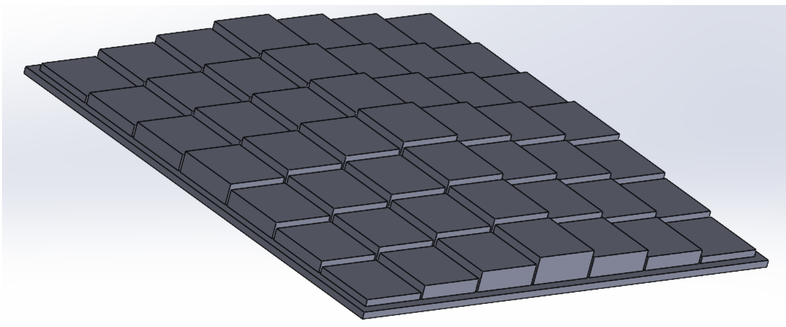


Figure 3.3: The board.

3.2 Lighting system

The lighting is the most crucial aspect of our experiment. As any camera needs light in order to acquired a good image, because of its working principles for the recontruction of a 3D image, plenoptic camera requres a well enlightened scene too, otherwise the image and the acquired data are not reliable.
In addition to this, our aim is to evaluate the accuracy of the camera under different lighting working conditions.
So in order to reach both of the goals, we built a particular enlightening system capable of well lighting the scene but also able to change the intensity of it in order to study the data under different conditions.
In particular the system is compound of five 3W LED with variable luminosity intensity between 160 and 240 LM, linked one by one to five different potentiometer.
Every potentiometer is connected to a single LED beacuse in this way we are able to controll all of them indipendently from the others.
The LED are supplied by a voltage generator of 5 V.
The electrical scheme of each led is shown in Fig. 3.4.

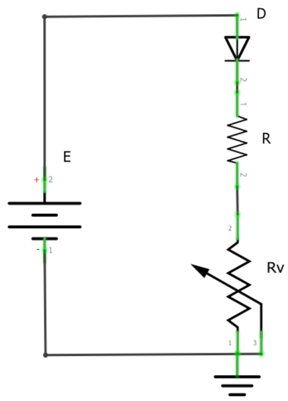


Figure 3.4: Electrical scheme.

As Fig. 3.4 shows, the circuit has two resistances in series: the resistance R, which

has a fixed impedance equal to 2.5 Ω, and the resistance R_v, that has a variable impedance between 0 and 5 Ω. The aim of the fixed impedence is to prevent the short circuit condition. This is verified when the value of the variable impedence is equal to 0 Ω.
Considering the characteristic caming from the datasheet of the LED, the forward and the breakdown conditons are verified for the following values of current and voltage:

Forward condition:

- $I_F = 750$ mA
- $V_F = 3.2$ V

Breakdown condition:

- $I_{BR} \leq 50$ μA
- $V_{BR} \geq 5$ V

In addition to that, we have studied the power of the led (P_{led}) values depending on the potentiometer position. Starting with potentiometer in the maximum configuration (i.e. variable impedance equal to 0 Ω), P_{led} is equal to 2.4 W. The procedure has been replicated increasing the variable impedence of 0.25 Ω until reaching its maximum value of 5 Ω. By increasing the variable resistance value, the power of the led decreases because of the reduction of current passing through the diode which is hold by the variable resistance. When the potenziometer is in the minimum position (i.e. $R_v = 5$ Ω), $P_{led} = 0.778$ W.
Contemporary to the computation of P_{led}, we have measured the power of the variable resistance considering the current coming from the diode. In order to do that, considering the power low $P_{R_v} = V \cdot i_{LED}$ and the Ohm's Law $V = R_{R_v} \cdot i_{LED}$, we

apply the following formula:

$$P_{R_v} = R_{R_v} \cdot i_{LED}^2 \tag{3.1}$$

From the computations, we get the following plot:

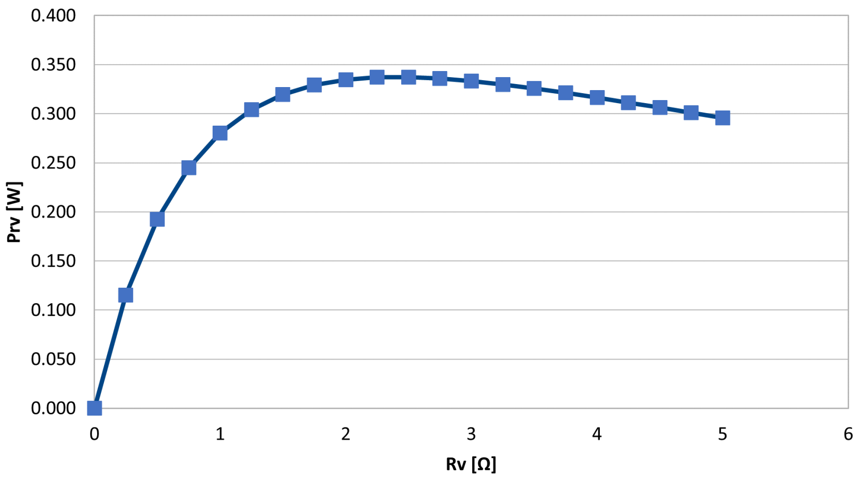


Figure 3.5: P_{R_v} varying R_v.

As we can see, P_{R_v} increses from 0 W, in which the corresponding variable resistance value is 0 Ω, to a maximum value equal to 0.337 W, where $R_v = 2.25\ \Omega$. After this value, P_{R_v} decreases slightly until the maximum value of the variable resistance is reached. At the maximum value of R_v, $P_{R_v} = 0.296$ W.
Initially P_{R_v} increases because R_v increases. Contemporary considering the decresing of the current, at a certain value of it, the tendency of P_{R_v} is inverted because the proportionality of the power with the current became greater with respect to the one with the resistance, being it equals to the square according to the power law.
The following figure shows the complete lighting system mounted in order to light up the board. As we can see the LED are mounted on a circular structure that

through the central hole does not interfere with the camera.

Figure 3.6: Rear view of the lighting system.

3.3 Camera calibration

As said in the previous chapter, before any experiment with a Raytrix camera, we must to calibrate it in order to increase the metric accuracy. The software given by Raytrix with the camera is Raytrix 4.0. This software contains two type of calibration: metric calibration and MLA calibration. More recent Raytrix softwares have additional calibration tools.

Both of them must be done, as the software highlight, and the MLA calibration must be run before the metric calibration.

The aim of the MLA calibration is to align the micro lens image with the calibration grid in order to have a correct 3D calculation and 2D refocusing. In order to achieve this aim, the calibration requires a homogeneous grey image. This is solved by mounting on the lens the calibration filter given by Raytrix.
After that, once the software automatically aligns the micro lens to the calibration grid, the calibration target can be removed in order to perform the metric calibration which aim and functioning is described in section 2.5.
More the metric calibration result is accurate, more the metric accuracy of the camera will be higher.
The calibration result coming from the Raytrix software gives as results some parameters in order to immediately understand the level of accuracy: mean euclidean distance, max euclidean distance and standard deviation.
The mean euclidean distance is, in average, how each model point is away from its corresponding measured point. The max euclidean distance parameters represents the maximum value of euclidean distance registered during the calibration and in order to have a good calibration, this parameter has to be less than 10 times the mean euclidean distance. Standard deviation indicates how the points are distributed on a great range of values. More this value is small, more the points are in a restricted range of value and so the calibration is accurate and reliable.
The values of our calibration used to take all the data are the following:

- Mean: 0.264
- Std. Dev.: 0.349
- Max Distance: 2.002

With these parameters, the calibration is considered by the software a good calibration. As the data shows, the value of the maximum euclidean distance is less than 10 times the mean euclidean distance.

3.4 The algorithms

In this section we are going to explain in detail the two algorithms designed to analyse the acquired data.

In order to make more simple the understanding, below a flowchart resumes the most important steps made.

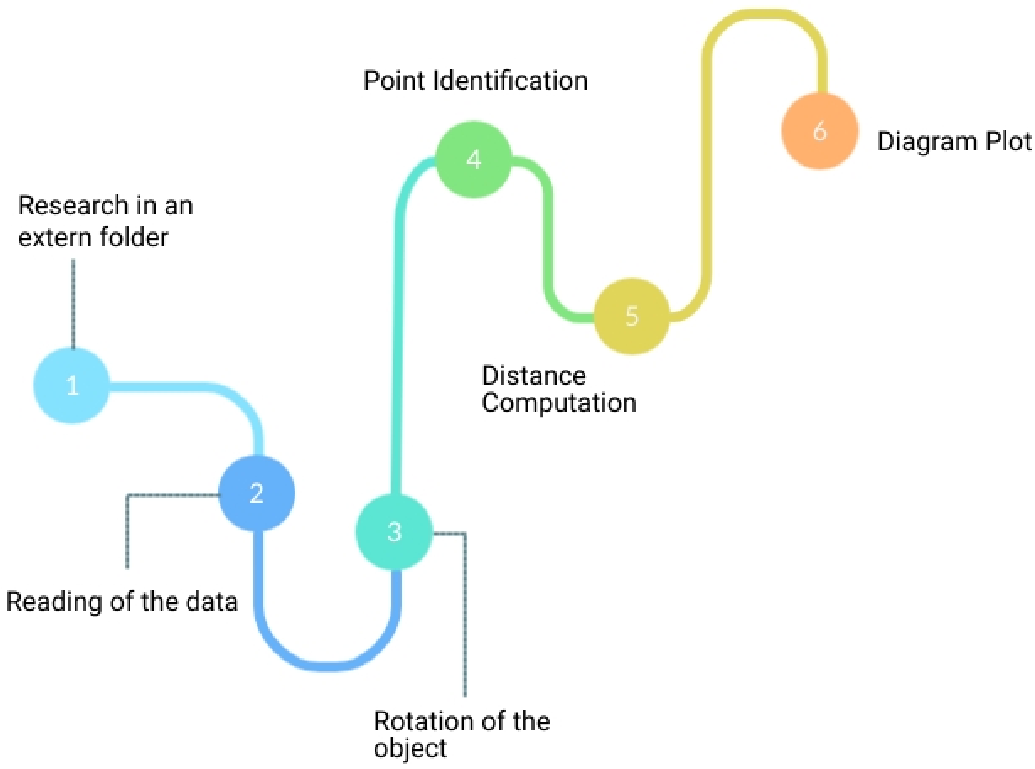


Figure 3.7: Flowchart of the algorithm.

Before starting with the explanation, it is important to say that the Raytrix software allows to export the acquired data of the camera in different exportable file formats: PLY, STL, XYZ, PCD. In order to export them, we used the PCD format, this means that all the informations are stored as cloud point. The choice fell on this type of file format because, during the researches for the designing of the algorithm, the best transformations for our porpouses are possible in an easy way through the extension .PCD.

The software used to acquire and process the data through the designed algorithm is *Matlab 2020b* by *MathWorks*.

The first step of the two algorithms is the acquisition of the whole amount of data coming from the folder in which are stored them. Since the extension, as said before, is .PCD, we use the command *dir*, as shown in fig.3.8, in order to search in the folder all the files with the extension .PCD.

All the files are stored in a structure array named *files*.

```
files=dir('C:\Users\Dodo-\OneDrive\Documenti\Università\16 dicembre\*.pcd');
```

Figure 3.8: *dir* command.

After having imported all the files, the algorithms read the data carried by them through the *pcread*, command specific for the extension .PCD (see Fig.3.9). The data are stored in a cell array named *ptCloud*.

Since the files stored in the folder are many, we introduce a *for* cycle in order to read contemporary all the data in the folder. The length of the latter corresponds to the length of the cycle, as shown in Fig. 3.9.

```
for i = 1:(length(files))
   ptCloud{i}=pcread((files(i).name))
```

Figure 3.9: *for* cycle and *pcread* command.

At this point, the idea is to rotate the image depicting the board in order to place it perpendicular with respect to the position of camera at the moment of the acquisition. By doing so, we have the depth values all equal for the whole cloud point and it is easier to compare the real measurements of the board with the ones coming from as results of the algorithm. The difference between the two algorithms is the

way in which the equation of the plane to rotate is computed.
In order to calculate the slopes of the board with respect to the three cartesian orientation, first of all, the first algorithm calculate the plane on which the board lies, using the mathematic equation of the plane passing through three points that in the equation below they are represented by the subscripts A, B and C.

$$\begin{vmatrix} x - x_A & y - y_A & z - z_A \\ x_B - x_A & y_B - y_A & z_B - z_A \\ x_C - x_A & y_C - y_A & z_C - z_A \end{vmatrix} = 0 \tag{3.2}$$

Since the position of the points is unknown until the entire cloud is plotted, in order to know the coordinates of the three points, we take them directly from the output of the *plot* command manually, through the *data tips* from the toolbar option.

```
figure(1)
title('Originale')
hold on
pcshow(ptCloud{1,1});
```

Figure 3.10: Plot of the cloud of points.

As it is shown in Fig.3.10, in order to plot the board, we use the command *pcshow*. Since the cell array, named *ptcloud*, contains all the point cloud of the different instants in which the board is acquired by the camera, in order to calculate the plane we consider only the first acquisition as reference for the identification of the three points. This assumption is made for semplicity and with the idea to select, for the identification of the plane, the points in an area well enlightened of the board in ordero to increase the accuracy of the coordinates.

However, the rotations transformations are performed on all the acquired data.
Once known the coordinates and the plane equation, we consider the plane passing through the x, y and z planes with the purpose to measure the slope of the board with respect to the three cartesian orientations.
In particular for the measure of the slopes, we procede measuring the slope of the board considering the plane yz and then depending on the resulted one, we rotate the board of an angle θ in a way to reach the parallelism between it and the y plane. Then considering the updated coordinates, we do the same procedure for the xy plane, in which the rotation is performed around z in a way to reach the parallelism between the board and the x plane, and for the xz plane in which we perform the rotation around y in order to have the parallelism between the board and the x plane.
For what concern the second algorithm, the plane equation is calculated in a different way in order to compare the accuracy of the depth map.
In particular this algorithm calculate the plane equation through four points that represent the mean value of the value of the coordinates belonging to four surfaces of the board. These selected four faces have the characteristic to be at the same height. So in order to do that the camera must be positioned in a way to capture at least four surfaces of the board with the same height.
The algorithm select the wanted surface through a command specific for the cloud of point called *findNeighborsInRadius*.

```
points_a=[15.0084,16.6967, 5.7888];
radius= 20;
[indices_a,dists_a]=findNeighborsInRadius(ptCloudOut3{1,i},points_a,radius);
pt_a{i}=select(ptCloudOut3{1,i},indices_a);
```

Figure 3.11: *findNeighborsInRadius* command.

This command select the surface in a circular way but selecting an apropriate radius, we are able to select a few points that does not belong to the surface of interest. The center of the circonference is chosen in order to select as center the center of the square surface. The command gives as result the indices and the distances of that points that lie within the specified radius.
The *select* command gets the point cloud data of radial neighbors and the informations are stored in a cell array.
At this point, we are able to calculate the mean value of that surface in the three cartisian directions *x, y, z*. The procedure is repeated for the other three surfaces. At the end of it, the four coordinates are stored in three different vectors: one for the x coordinates, one for the y coordinates and one for the z coordinates.
In order to calculate the plane we adopted the Matlab's *curve fitting* toolbox. This tool is able to perform the computation of a line or a surface starting from a certain number of points.
The software uses the method of the *linear least squares* in order to fit the data, even for the fitting of a surface. This is done through the assumption that the z coordinate is dependent on the x and y in the form:

$$z = a_0 + a_1 x_i + a_2 y_i \tag{3.3}$$

Given a set of data (x_n, y_n, z_n), the best fitting curve has the least square error:

$$S = \sum_{i=1}^{n} [z - f(x_i, y_i)]^2 = \sum_{i=1}^{n} [z_i - (a_0 + a_1 x_i + a_2 y_i)]^2 = min \tag{3.4}$$

In order to obtain the least square error, the coefficient a_0, a_1, a_2 first derivatives must yield to zero. This has as results the matrix:

$$\begin{bmatrix} n & \sum x_i & \sum y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i \\ \sum y_i & \sum x_i y_i & \sum y_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \sum z_i \\ \sum x_i z_i \\ \sum y_i z_i \end{bmatrix} \tag{3.5}$$

Where n is the number of data set.

From eq (3.5), the n^{th} degree polynomial equation can be expressed as:

$$z_i = S(x_i, y_i) = \sum_{j=0}^{m} a_j \sum_{i=1}^{n} p_i(x_i, y_i) \tag{3.6}$$

The difference between $Z_f(x_i, y_i)$ and the surface elevation gives the residual $R(x_i, y_i)$ as equation (3.7) shows.

$$\sum_{i=1}^{n} R(x_i, y_i) = Z_f(x_i, y_i) - S(x_i, y_i) \tag{3.7}$$

In general the surface equation through the 2D least square method can be represented as [34]:

$$R^2 = Z_f^2 - Z_i^2 \tag{3.8}$$

In our cases, since the data set used are characterized by four point, the resulting equation coming from the computations operated by the tool will be of 1^st order, in the form:

$$z = a_0 + a_1 x + a_2 y \tag{3.9}$$

After that, the computation of the slope of the plane is made, in the same way of the first algorithm, in order to perform the rotation.

Mathematically, a rotation is possible through a specific matrix, called *rotational* *matrix,* R, which is used in the rotation of vectors and tensors whereas the coordinate system remains fixed. The general rule for applying the rotation matrix are the same as for the coordinate transformation matrix (see eq.(3.10)).

$$v' = v \cdot R \qquad \text{For the vectors} \tag{3.10}$$

The general definition of R, in 3D is:

$$R = \begin{bmatrix} cos(x', x) & cos(y', x) & cos(z', x) \\ cos(x', y) & cos(y', y) & cos(z', y) \\ cos(x', z) & cos(y', z) & cos(z', z) \end{bmatrix} \tag{3.11}$$

In the equation above, (x', x) represents the angle between the x' and x axes, (x', y) is the angle between the x' and y axes, etc. However, if the rotation happens along the x, y or z axes, the rotational matrix endures a reduction and simplifications in terms of arguments. This three particular rotational matrices are called *elementary rotation matrices* and their structures are shown in the eq(3.12).

$$Rot(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}$$

$$Rot(y, \beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix} \tag{3.12}$$

$$Rot(z, \gamma) = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In the equations above, the angles α, β and γ are the ones used in order to rotate the cloud point in the three cartesian directions.
In the algorithm the first matrix, intended to the rotation around x, is applied on the data coming from the folder because this is the first rotation computed by the algorithm. Instead, the others rotational matrices are applied on the updated data coming from the algorithm. In particular the rotational matrix around the y axes is applied to the updated data coming from the rotation around x, while the rotational matrix of the z axes is applied to the data coming from the rotation around y. All the procedure is performed consequentially by the algorithm.
The rotational transformations happens through a command called *rigid3D*, which stores information about the transformation and enables forward and inverse transformations (fig.3.12).

```
tformm = rigid3d(A,trans1)
```

Figure 3.12: *rigid3D* command.

The command just described presents another argument in addition to the rotational matrix A: the argument *trans1*. This is because the command *rigid3D* allows roto-translations transformations. But in our purpose the translational transformation is used once, at the end of the rotational transformations in order to place at the origin one of the edges of the board. So the arguments of this matrix are zero for both of the first two roto-translations transformations.
The 3D transformation is applied to the cloud point through the command *pctransform*, as shown in fig. 3.13. The ones coming from the transformation are updated with the definition of a new cell array named *ptCloudOut1*.

```
ptCloudOut1{i} = pctransform(ptCloud{i},tformm)
```

Figure 3.13: *pctransformation* command.

Once the board is in the desired position, the two algorithms start to perform the comparison.

In particular, they work at the same way for the comparison. So we are going to explicate how the first algorithm performs it, taking into account the same steps for the other algorithm.

The comparison is performed on the whole surfaces acquired by the camera in order to undestand the accuracy of the depth map. In order to do that, our attention is focused on the z coordinates.

The surfaces are selected by the algorithm one by one using the command *find-NeighborsInRadius*. Also in this step the calculation of the center of each surface of the board is made by hand.

The informations coming from the command are stored in a cell vector used to perform the computation of the mean value and the standard deviation of the whole points belonged to the surface. The mean value and the standard deviation in statistic are the most popular ways in order to calculate the variabilty of a set of data. In particular the mean value, or average, is defined as:

$$\mu = \frac{\sum_{i=1}^{n} x_i}{n} \tag{3.13}$$

In the equation above the numerator represents the sum of the acquisitions, while the denominator n represents the total number of the acquisitions.

Regarding the standard deviation, it shows how a set of data differs from the mean value, as figure 3.14 shows.

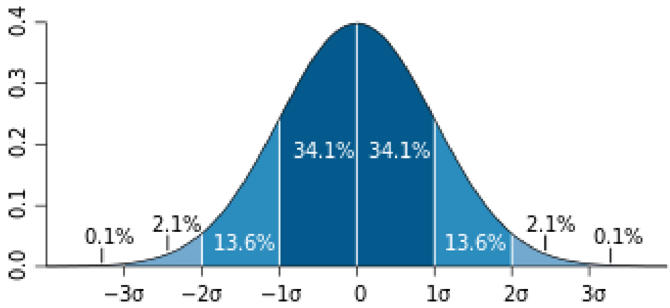


Figure 3.14: Standard deviation (example).

Mathematically it is defined as:

$$S = \sqrt{\frac{1}{N-1}\sum_{i_1}^{N} |A_i - \mu|^2} \tag{3.14}$$

Through Matlab's commands, they are calculated as figure 3.15 shows.

```
media_z_a{i}=mean(pt_a{1,i}.Location(1,3),1);
stand_a_z{i}=stand_a{1,i}(1,3);
```

Figure 3.15: Mean value and standard deviation commands.

Chapter 4

Results and conclusions

In this chapter, we are going to show the results gotten using the designed algorithms.
In particular the experiment is made at two different distances, 75 cm and 45 cm, to which the two algorithms have been applied. In this chapter, we will explain the results for both the cases making a comparison between the results at the same distance, then we will make a general final comparison among the overall results.
After that, we are going to conclude our work making explicit the possible future works.

4.1 Distance 75 cm

By placing the camera at the distance of 75 cm from the object, we acquire 30 images, with the five LED in a specific configuration: one LED at maximum light intensity and all the others at minimum intensity. This is made in order to increase the contrast of the image and consequently in order to have a well defined image for the point of view of the edges.
The rotated board used for the computations is shown in figure 4.1. In order to better understand the results, on the image we added the surfaces' names and the

point considered as reference.

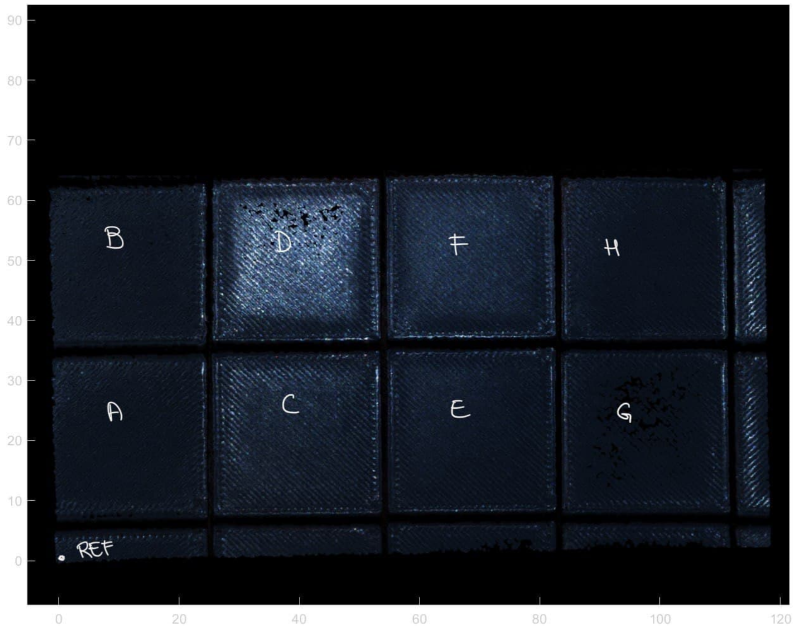


Figure 4.1: Rotated and translated board.

Each surface, as said, has a particular theoretical height that in the depth map computed by the algorithms is referred to a different reference. In order to understand the results, the following table contains both theoretical and practical values (referred to the reference used in the algorithms) of the heights.

Surface name	Absolute height [mm]	Relative height [mm]
A	13	2
B	16	5
C	15	4
D	18	7
E	13	2
F	16	5
G	11	0
H	14	3

The results of the first algorithm are shown in the following graph.

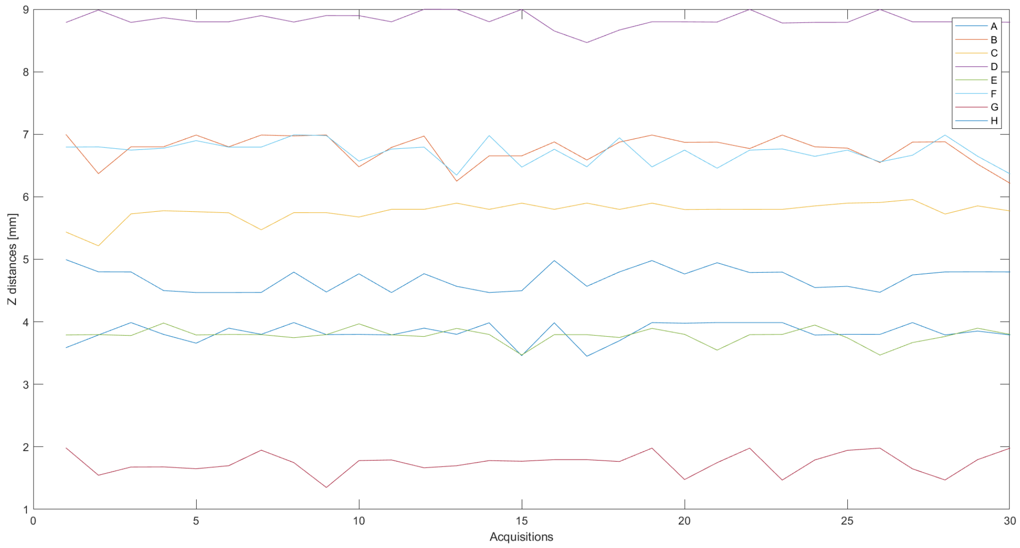


Figure 4.2: Results of the 1^{st} algorithm.

The graph above shows on the x-axis the thirty acquisitions taken. On the y-axis there are the values, in mm, of the mean value calculated for each surface of each acquisition.

The coloured lines represent, as the legend suggests, the eight surfaces on which our attention is focused.

In order to make understand the graph, there are some values showed for each surface. These points are taken trying to capture the maximum, the minimum and the middle values of the results in order to give an idea of the numerical value of them.

As figure 4.3 shows, the results of the first algorithm point out a difference from the theoretical heights. These results was expected and in particular the error from the theoretical value is 1.7 mm with a range of variation of 0.5 mm.

The values of standard deviation underline that the whole amount of data (more that 2 million) for each surface is less than 1 mm (see figure 4.4).

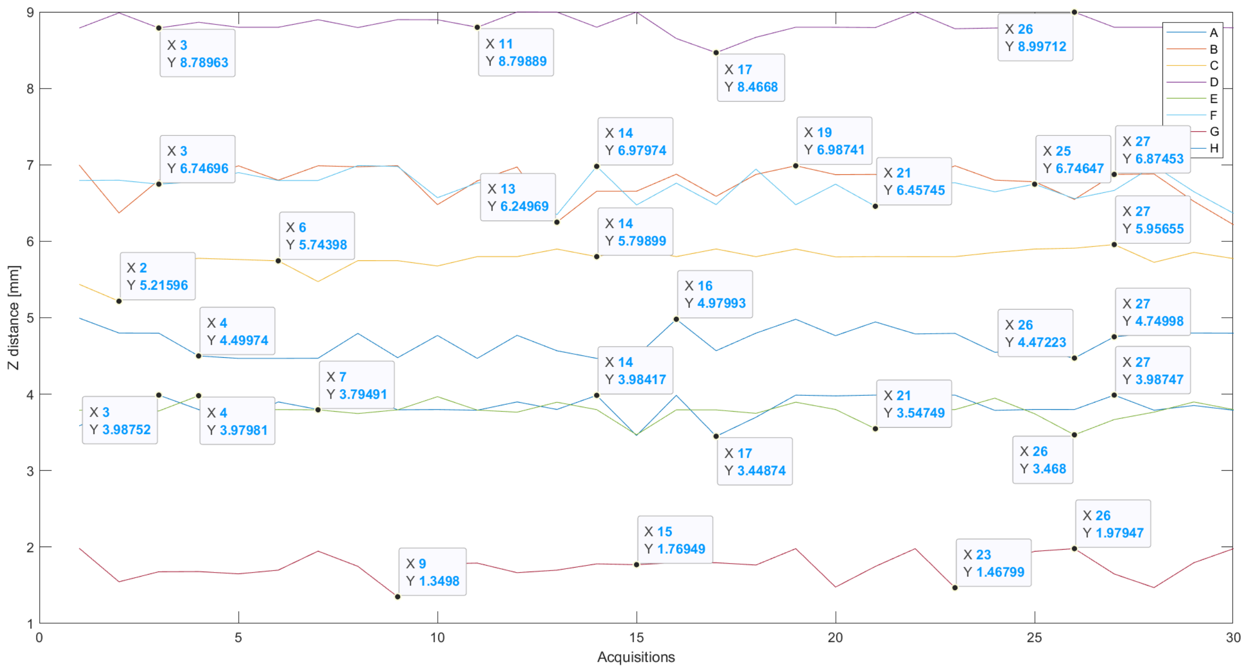


Figure 4.3: Results of the 1^{st} algorithm.

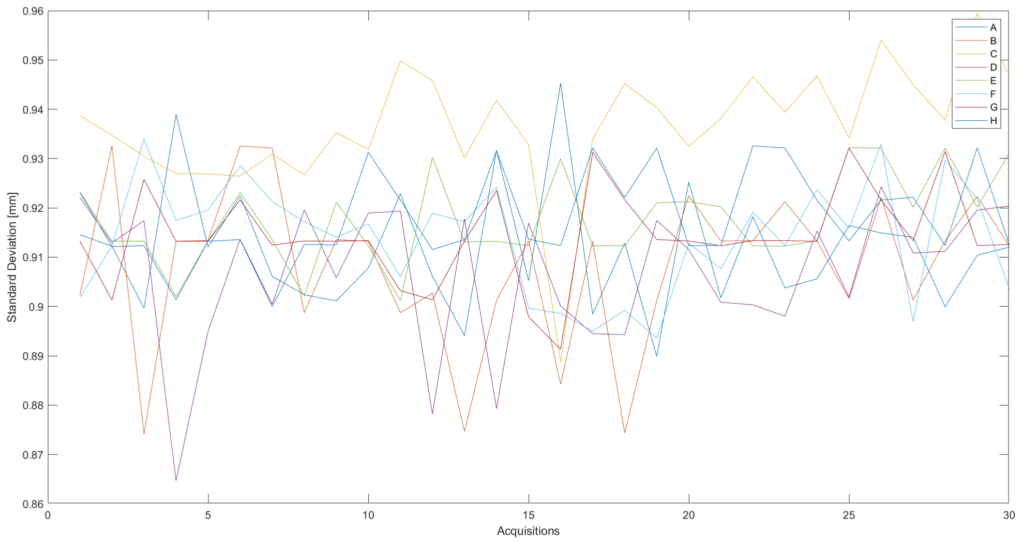


Figure 4.4: Standard deviation 1^{st} algorithm.

Regarding chart of the standard deviation, on the x-axes there are the total number of acquisitions taken, while on the y-axes there are the resulting values of standard deviation computed for each surface of each acquisition. It shows that the values are distant from the mean value in a constant way because the range on which the standard deviation swings is between 0.87 mm and 0.95 mm.

At this point, we introduce the graph computed by the results coming from the second algorithm (figure 4.5).

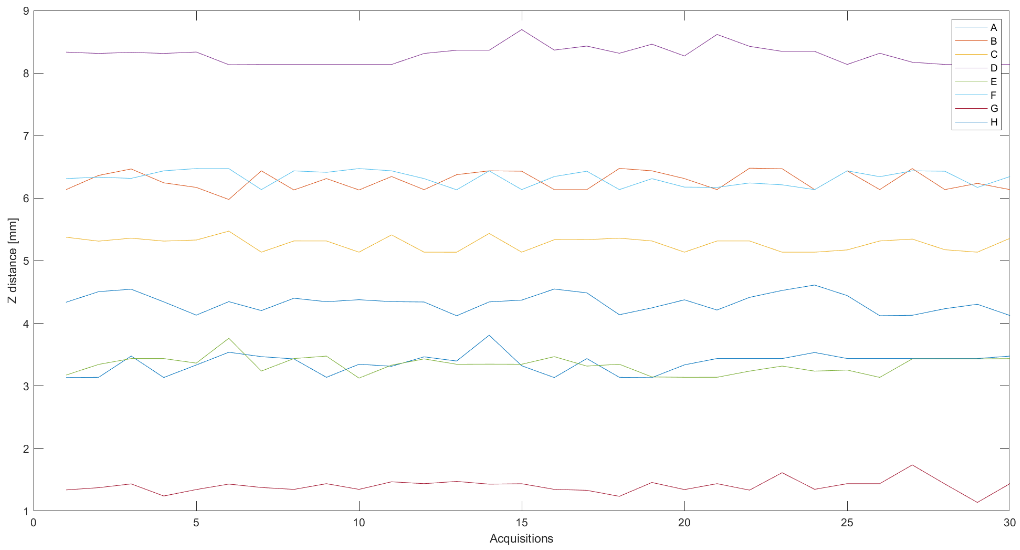


Figure 4.5: Results of the 2^{st} algorithm.

As before, we show some numerical values, as figure 4.6 shows.

The error is more precise than the error coming from the first algorithm, and precisely in this case it is 1.3 mm. The reason behind this is due to the different technique adopted for the computation of the plane equation required for the calculation of the slope of the board.

Also in this situation, the standard deviation computed by the algorithm is less than 1 mm for the whole acquisitions, and the range of variation of it is quite small, as figure 4.7 shows.

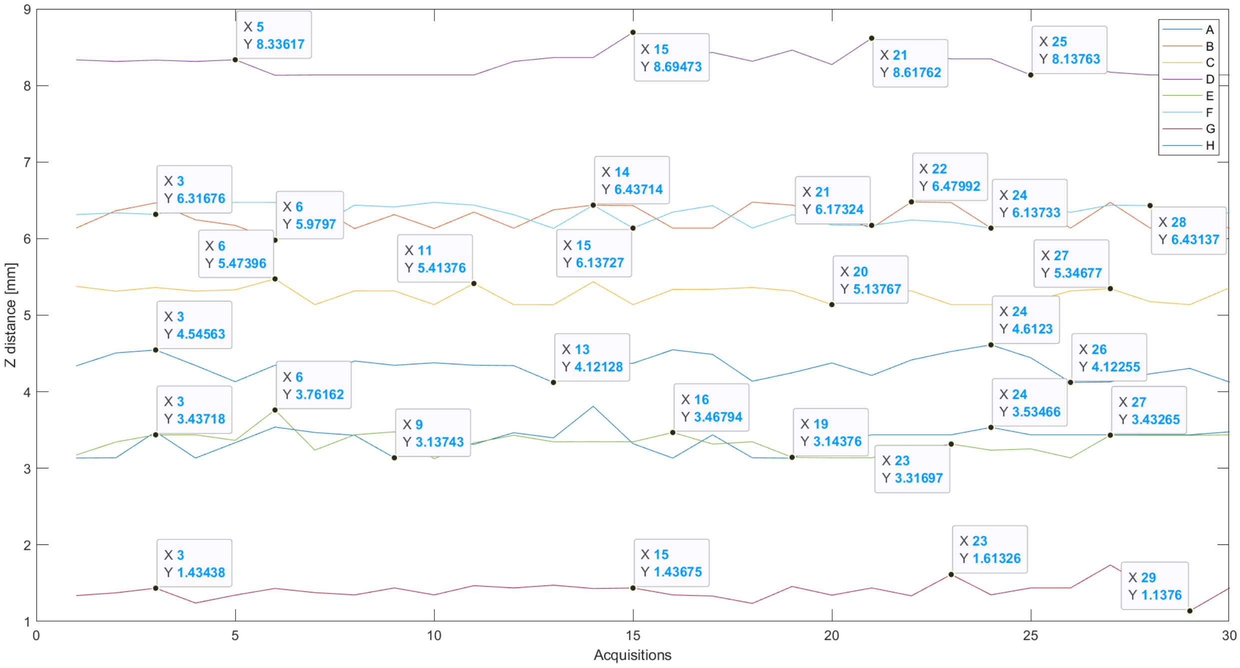


Figure 4.6: Results of the 2^{st} algorithm.

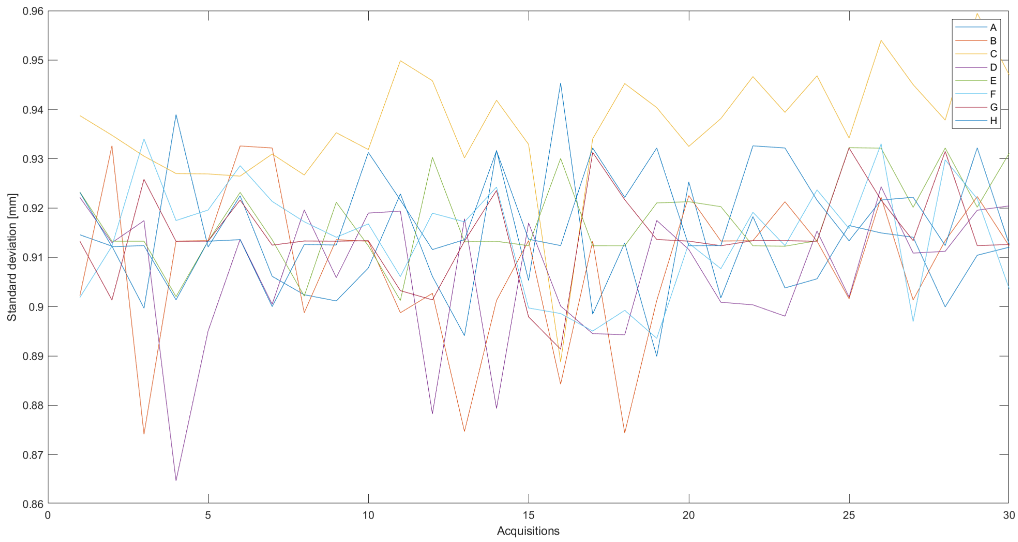


Figure 4.7: Standard deviation 2^{nd} algorithm.

4.2 Distance 45 cm

The procedure about the placement is the same as before, with the difference that in this case the distance of the camera from the board is 45 cm. This distance has

been chosen because of the limit of the camera during the calibration. The limit is due to the calibration target supplied by Raytrix. At distance less than 40 cm, with this specific target with point target of 2 mm, the calibration of the camera is not able to be performed by the software. So we chose a distance close to this limit to see its behaviour.
Also in this case in order to increase the contrast, we use the same configuration of the LED used before: one LED at maximum intensity and all the others at minimum intensity.
The representation of the board, after the roto-translation computations, is as figure 4.8 shows. Also in this case we explicit the name of the surface to better understand the variable of the graphs.

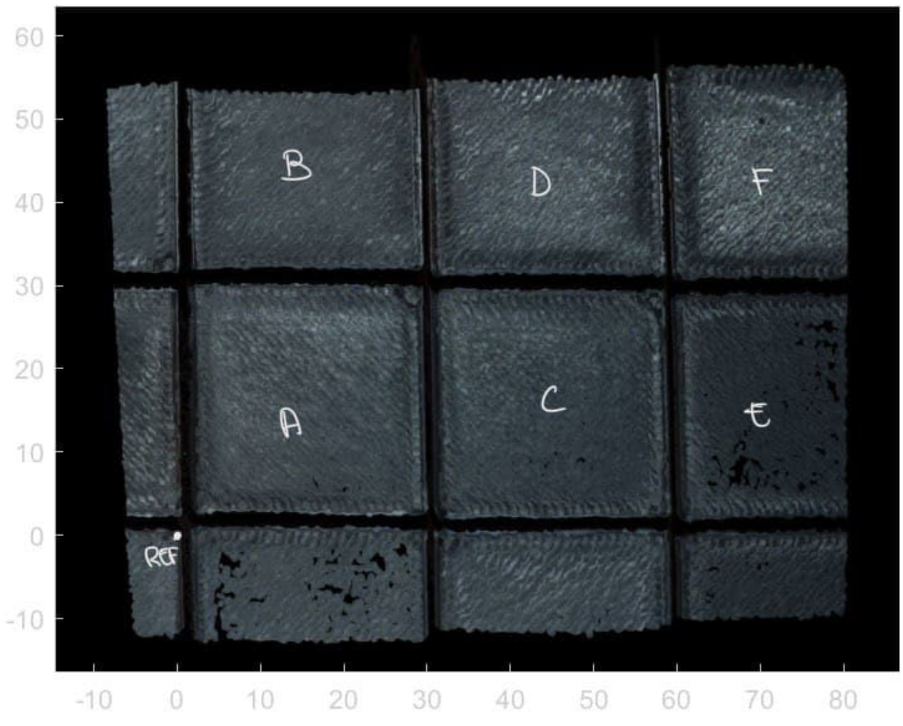


Figure 4.8: Rotated and translated board.

Also here the heights of the surfaces are referred to a reference different from the theoretical one so, as before, we differentiate the absolute and relative heights values in the following table.

Surface name	Absolute height [mm]	Relative height [mm]
A	18	5
B	16	3
C	16	3
D	13	0
E	14	1
F	11	-2

At this point, the results of the first algorithm are shown in figure 4.9.

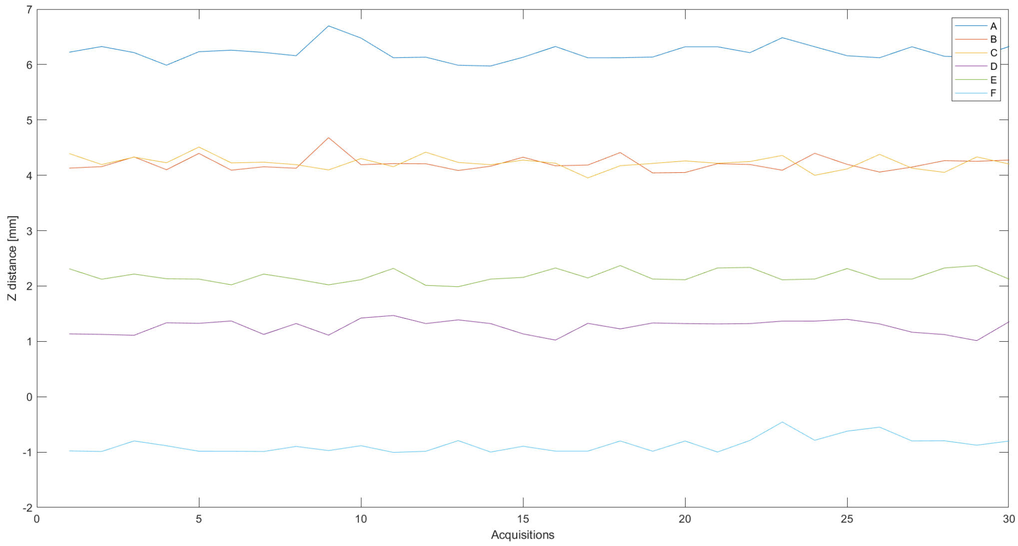


Figure 4.9: Results of the 1^{st} algorithm.

Here the results are nearer to the theoretical values showed in the table. As for the distance of 75 cm, some values are considered on the graph (see figure 4.10).

These results were expected because of the theory linked to the accuracy of the depth map: since the accuracy of the depth map is inversely proportional to the

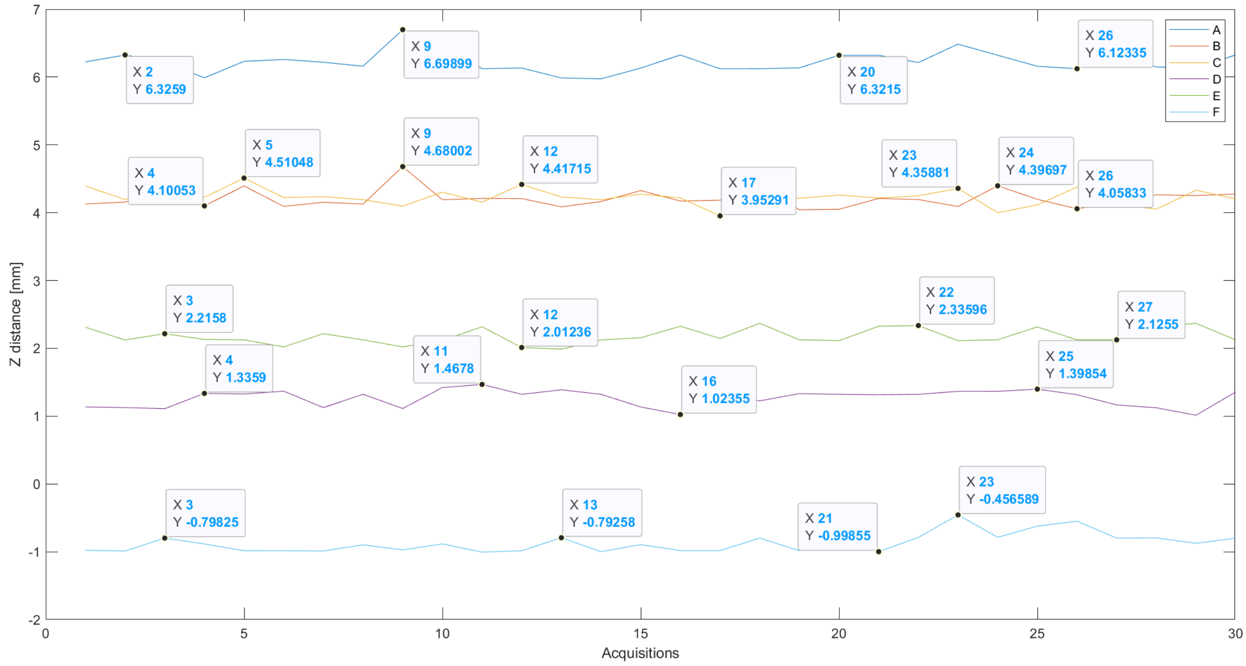


Figure 4.10: Results of the 1^{st} algorithm.

distance of the camera from the object, decreasing it, the accuracy increases. Here the distance error is 1.2 mm for the first algorithm.

The standard deviation is shown in figure 4.11.

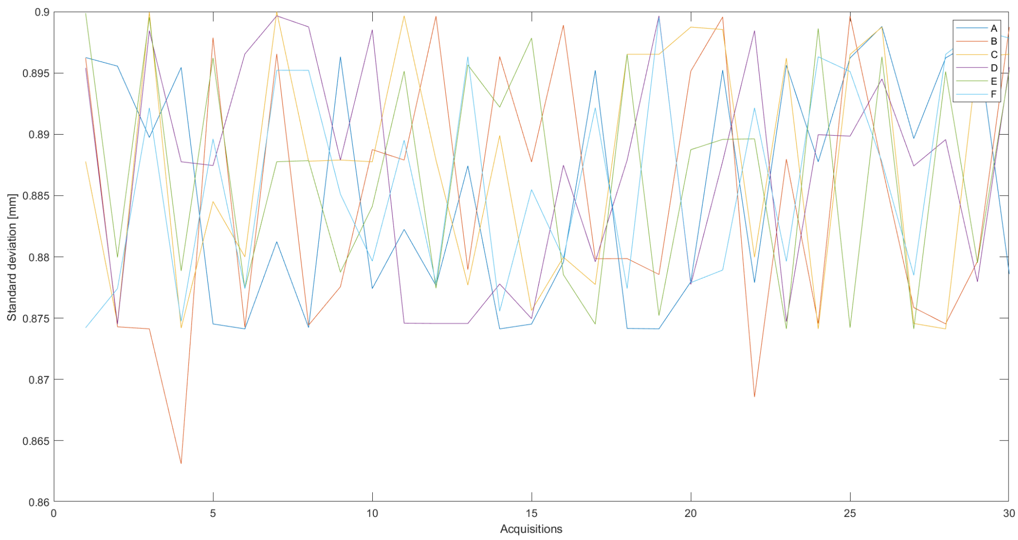


Figure 4.11: Standard deviation of the 1^{st} algorithm.

The range of variation in this case is 0.865 mm and 0.9 mm.

At this point the results coming from the 2^{nd} algorithm are shown (see figure 4.12).

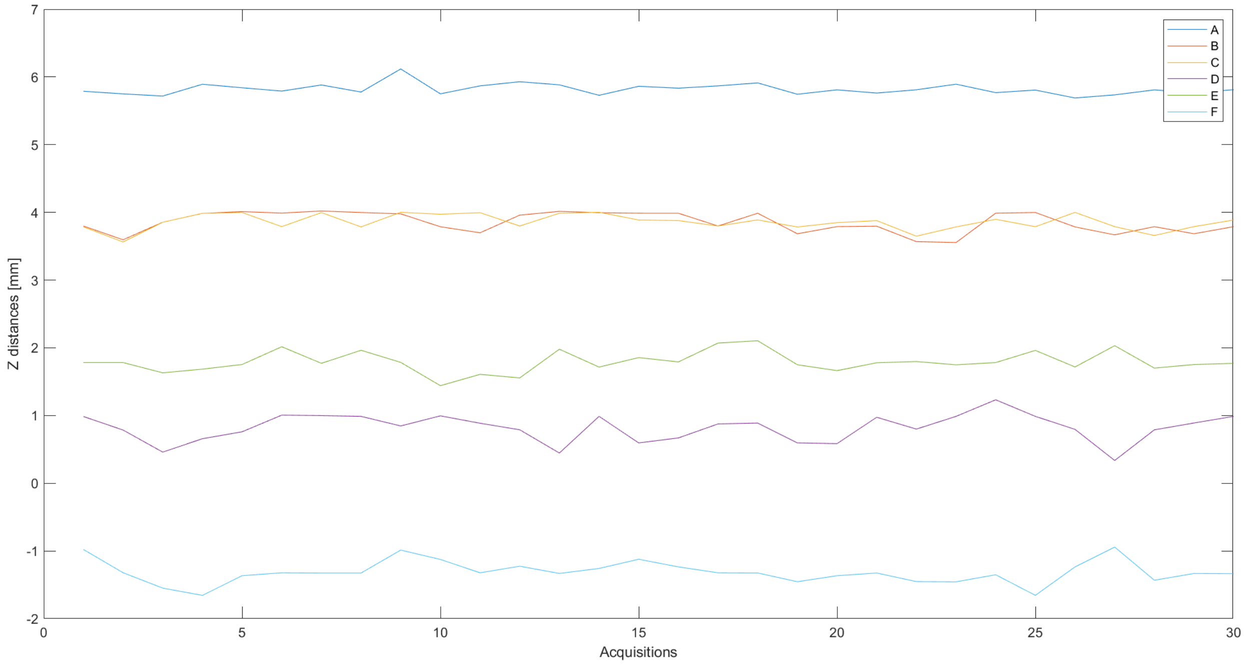


Figure 4.12: Results of the 2^{nd} algorithm.

As before, some values are added on the graph, as figure 4.13 shows.

As the results suggest, these distances are affected by a lower error equal to 0.9 mm.

Regarding the standard deviation of these values, as figure 4.14 shows, it is similar to the standard deviation of the first algorithm.

Comparing all the results, we can say that at the distance of 45 cm, the depth map computed by the camera is more precise. What we can see comparing the whole results is that the maximum range of variation of the values is equal to 0.5 mm among the different acquisitions. This is due two principal factors: the algorithm and the lighting system. For what concern the algorithm, since as starting point to calculate the center necessary to use the command to select the surface we take only, as reference, the first image acquired, the other acquisitions are considered

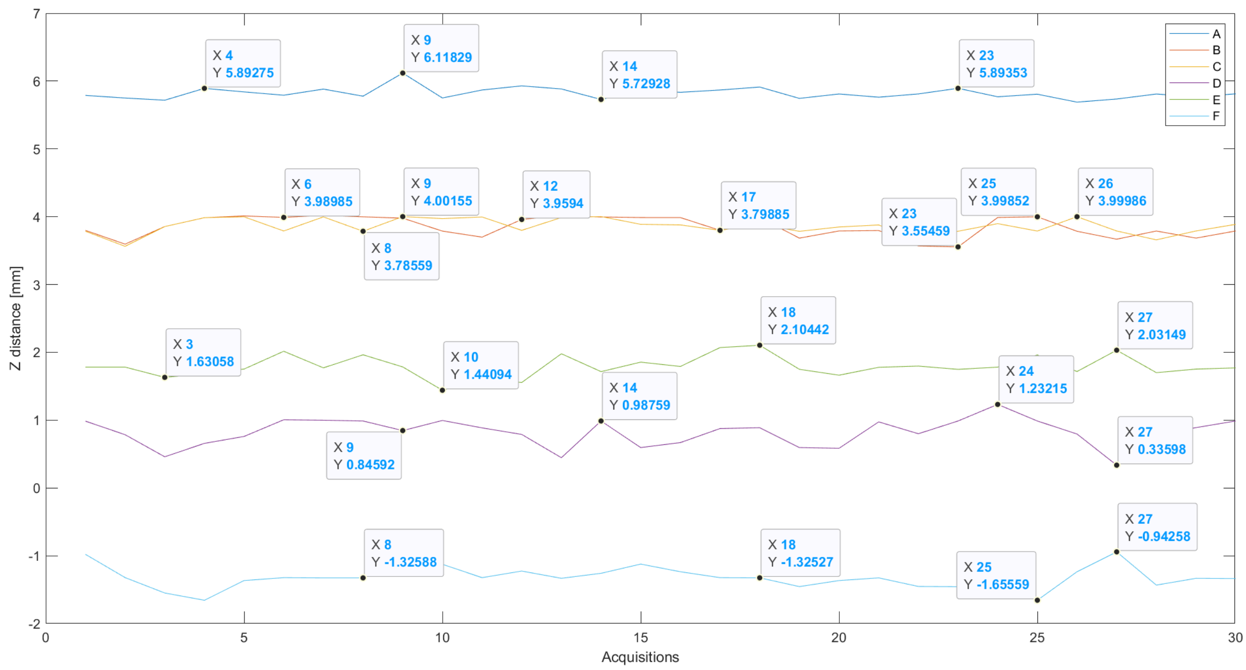


Figure 4.13: Results of the 2^{nd} algorithm.

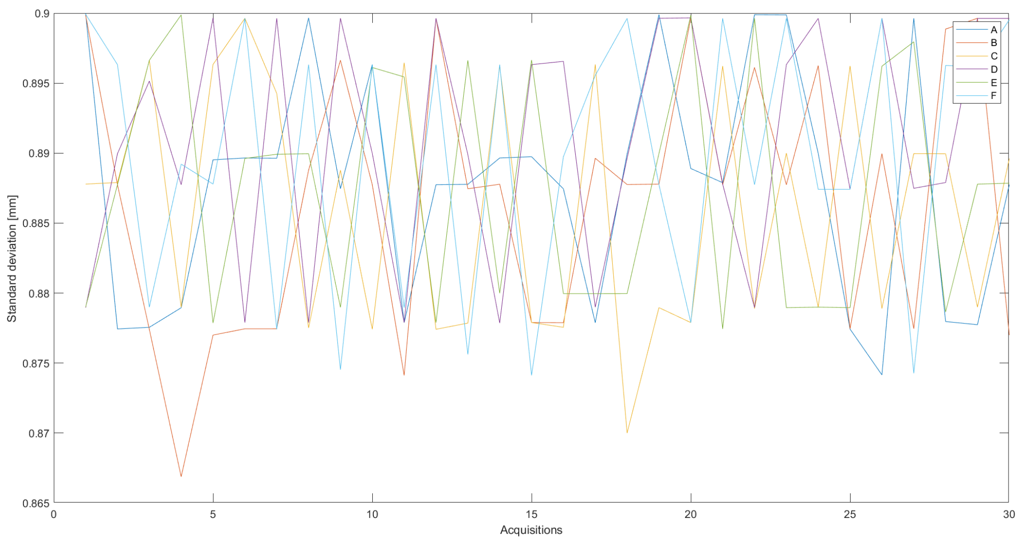


Figure 4.14: Standard deviation 2^{nd} algorithm.

only as comparison. So that exact point on the reference image will not be the same for the other acquisitions.

Regarding the lighting system, as we said in the theory part and during the description of the lighting system, the plenoptic camera needs to a lighting system

with high luminosity intensity possibly constant for the whole procedure. As the results show, the light is not constant enough.
For what concern about the difference of errors between the distance of 75 cm and 45 cm, this is due to the calibration. Before we said that the accuracy increases if we decreases the distance, but the reason why we obtain a better results at 45 cm rather than at 75 cm is also because of the calibration. At 75 cm, the software, used for the calibration of the camera, is less accurate in the recognition of the pattern presents on the target necessary to perform it. This fact has as consequence a worst calibration at 75 cm with respect to 45 cm.
Regarding the standard deviations, they are similar if we compared, at the same distance, the results of the two algorithms but they differ in the two distances. This difference is due to the reduction of distance between the camera and the object. Because of its dependence from the data, and being they more precise at 45 cm rather than 75 cm, the standard deviation will be better at the distance of 45 cm.

4.3 Conclusions and future works

At the end of our work, we can summarize the principal informations gotten by our research.
The plenoptic camera exploits, through the presence of the microlens array, the redundant informations coming from the images formed on each microlens. In addition, thanks to it, we are able to understand why in the plenoptic camera the refocusing, also after the acquisition of the image, is possible.
Through some mathematical definitions, we were able to explain the difference between the plenoptic camera 1.0 and 2.0.
In addition, our researches brought us to two interpretations about the explanation of the algorithm, being it nowadays unknown, that allows the focus in the plenoptic

camera.
We have defined the concept of depth map, our subject in the experiment. What come out from the theory about the depth map is that its accuracy is inversely proportional to the distance of the camera from the objects: increasing it, the accuracy decreases.
We explained, also, the algorithm behind the calibration of the camera. Calibration that is the most important aspects along with the lighting conditions. About it, we explained, for our particular plenoptic camera, the types of calibrations necessary and mandatory to be perform before any acquisition.
In the last part of our work, we defined how our particular object has been designed and developed, explaining the values of distances and heights.
In addition, the lighting system has been defined, focusing on the electronic issues. The algorithms necessary to perform the computations have been described.
From the two algorithms, we got that the camera measurements are affected by some errors. Considering the distance of 75 cm, the camera make an error equal to 1.8 mm with standard deviation equal to 0.9 mm; at the distance of 45 mm, the camera make an error equal to 1 mm with standard deviation equal to 0.85mm. The camera, considering the results, can be used for industrial application, as supplement instrumentation for the measurements.
Obviously, as we said in the previous sections, these results can be improved. So the future works on the plenoptic camera should be focused on testing it with a better lighting system, in terms of light intensity, able to generate more contrast.
In addition the calibration is an another important feature that can be improved, in particular the one regarding the metric calibration.
By considering the calibration, this has been, for distances greater that 1 m, an obstacle because of the dimension of the target used to perform it. Having it specific dimensions, the camera, around these distances, was not able to focus the image on it and we could not perform the calibration. So future works should be

focused, also, on testing the camera around these values using a calibration target with greater dimensions.

Bibliography

[1] A. Gershun. The light field. Moscow. *Journal of Mathematics and Physics, 18, 1936.*

[2] M. Levoy and P.Hanrahan. *Light field rendering, 1996* in Gaochang Wu, *Light Field Image Processing: An Overview, 2017.*

[3] C.Michael, G. Steven, J., S. Richard, G. Radek, and S. Rick. *The lumigraph* in Gaochang Wu *Light Field Image Processing: An Overview, 2017.*

[4] Gabriel Lippmann. Epreuves reversibles. Photographies integrals. *Comptes-Rendus Academie des Sciences*, 146:446-451, 1908.

[5] Hebert E Ives. *A camera for making parallax panoramagrams*, Josa, 17(6):435-437, 1929.

[6] E.H. Adelson and J.Y.A. Wang. *Single lens stereo with plenoptic camera.* IEEE transactions on pattern analysis and machine intelligence, 14(2):99-106, 1992.

[7] Ren Ng, Mark Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz and Pat Hanrahan. *Light field photography with a hand-held plenoptic camera.* Computer Science Technical Report CSTR, 2:11, 2005.

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

[9] Aaron Isaken, Leonard McMillan, and Steven J Gortler. *Dynamically reparameterized light field.* In *Proceeding of the 27th annual conference on Computer graphics and interactive tecniques*, pages 297-306. ACM Press/Addison-Wesley

Publishing Co., 2000.

[10] Ren Ng. *Fourier slice photography.* In *ACM Transactions on Graphics (TOG)*, volume 24, pages 735-744. ACM, 2005.

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

[12] Todor Georgiev and Andrew Lumsdaine. *Superresolution with plenoptic camera 2.0.*. Adobe Systems Incorporeted, Tech. Rep, 2009.

[13] Todor Georgiev and Andrew Lumsdaine. *Focused plenoptic camera and rendering.* Jornal of Electronic Imaging, 19(2):021106-021106, 2010.

[14] Todor Georgiev and Andrew Lumsdaine. *Reducing plenoptic camera artifacts.* In *Computer Graphics Forum*, volume 29, pages 1955-1968. Wiley Online Library, 2010.

[15] Zhan Yu, Jingy Yu, Andrew Lumsdaine and Todor Georgiev. *Color demosaicing in plenoptic cameras.*

[16] Tom E Bishop and Paolo Favaro. *Full-resolution depth map estimation from multiple aliased views.* In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on, pages 1622-1629.* IEEE, 2009

[17] Ng, R. *Digital light field photography.* PhD thesis, Stanford University, Stanford, USA, 2006. In N.Zeller, F. Quint, U. Stilla *Calibration and Accuracy Analysis of a Focused Plenoptic Camera.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Science, Volume ||-3, 2014.

[18] Perwass, C. and Wietzke, L. *Single lens 3d-camera with extended depth-of-field*, 2012. In *Calibration and Accuracy Analysis of a Focused Plenoptic Camera.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Science, Volume ||-3, 2014.

[19] Gortler, S. J., Grzeszczuk, R., Szeliski, R. and Cohen, M. F.. *The lumigraph*, 1996. In *Calibration and Accuracy Analysis of a Focused Plenoptic Camera.* ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information

Science, Volume ||-3, 2014.

[20] M. Wolffe. *How safe is the light during ophtalmic diagnosis and surgery*. Eye, 30(2):186-188, 2015.

[21] Perwass, C. and Wietzke, L. *Single lens 3d-camera with extended depth-of-field*, 2012. In *Human Vision and Electronic Imaging XVII*, Burlingame, California, USA.

[22] Strobel, L., Compton, J., Current, I., and Zakia, R. *Photographic Materials and Process*, 1986. In *Light Field Photograpghy with a Hand-held Plenoptic Camera*, Stanford University, 2005

[23] Ren Ng, Marc Levoy, Mathiue Bredif, Gene Duval, Mark Horowitz, et al. *Light Field Photograpghy with a Hand-held Plenoptic Camera*, CSTR 2005-02, Stanford university. 2005, Stanford University Comkputer science Tech Report. hal-02551481

[24] A. Lumsdaine and T. Georgiev. *Full resolution lightfield rendering*, Tech. Rep., Adobe System, 2008 In *Focusing plenoptic camera and rendering*, 2010.

[25] A. Lumsdaine and T. Georgiev. *The focused plenoptic camera.* In *Focusing plenoptic camera and rendering*, 2010.

[26] Todor Georgiev and Andrew Lumsdaine. *Focusing plenoptic camera and rendering*, Journal of Electronic Imaging 19(2), 021106 (Apr-Jun 2010).

[27] Marwah et al. in *Plenoptic cameras.*

[28] Christian Heinze, Stefano Spyropoulos, Stephan Hussmann and Christian Perwass. *Automated Robust Calibration Algorithm for 3D Camera Systems*, Senior Member IEEE, Faculty of Engineering, West Coast University of Applied Sciences, Heide Germany.

[29] Scheimpflug, T. *Improved Method and Apparatus for the Systematic Alteration or Distortion of Plane Pictures and Images by Means of Lenses and Mirrors for Photography and for other purposes*, Photometric Engineer- ing 32(3), GB Patent No. 1196. Filed 16 January 1904, and issued 12 May 1904. In *Automated*

Robust Calibration Algorithm for 3D Camera Systems, West Coast University of Applied Sciences, Heide Germany.

[30] Brown, D. C. *Decentering distortion of lenses*,Photometric Engineer- ing 32(3), 444-462, 1966.

[31] Johannsen, O. et al *On the Calibration of Focused Plenoptic Cameras*, http://hci.iwr.uni-heidelberg.de/Staff/bgoldlue/papers/JHGP13 gcpr inm.pdf [15.09.14]. In *Automated Robust Calibration Algorithm for 3D Camera Systems*, West Coast University of Applied Sciences, Heide Germany.

[32] Tsai, R. Y. *A Versatile Camera Calibration Technique for High- Accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses*, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, August 1987. In *Automated Robust Calibration Algorithm for 3D Camera Systems*, West Coast University of Applied Sciences, Heide Germany.

[33] Heinze, C. *Design and test of a calibration method for the calculation of metrical range values for 3D light field cameras*, Master's thesis, 98 pages, West Coast University of Applied Sciences, Heide, Germany, 2014. In *Automated Robust Calibration Algorithm for 3D Camera Systems*, West Coast University of Applied Sciences, Heide Germany.

[34] Dr. Tahseen F. Abbas, Omar A. Raheem, Abdulwahab N. Abbas *Surface Fitting and Representation By Using 2D Least Squares Method in CAD Application*, Eng. and Tech. Journal.