Analysis of Two Representative Algorithms of Depth Estimation from Light Field Images

by

Yutao Chen

A Report Submitted in Partial Fulfillment
of the Requirements for the Degree of
MASTER OF ENGINEERING
in the Department of Electrical and Computer Engineering

Abstract

Supervisor

Dr. Pan Agathoklis (Department of Electrical and Computer Engineering)

Co-Supervisor

Dr. Kin Fun Li (Department of Electrical and Computer Engineering)

Lightfield (LF) cameras offer many more of advanced features than conventional cameras. One type of LF camera, the lenslet LF camera is portable and has become available to consumers in recent years. Images from LF cameras can be used to generate depth maps which is an essential tool in several areas of image processing and can be used in the generation of various visual effects.

LF images generated by lenslet LF cameras have different properties that images generated from an array of conventional cameras and thus require different depth estimation approaches. To study and compare the differences of depth estimation from LF images, this project describes two existing algorithms for depth estimation. The first algorithm, from Korea Advanced Institute of Science and Technology, estimates the depth labels based on stereo matching theory, where each label is corresponding to a specific depth. The second algorithm, developed by University of California and Adobe Systems Company, takes full advantage of the LF camera structure to estimate depths from so-called refocus cue and correspondence cue, and combines the depth maps from both cues in a Markov Random Field (MRF) to obtain a quality depth map. Since these two methods apply different concepts and contain some widely used techniques for depth estimations, it is worthy to analyze and compare their advantages and disadvantages. In this report, the two methods were implemented using public domain software, the first method being called the DEL method and the second being called the DER method. Comparisons with respect to computational speed and visual quality of the depth information show that the DEL method tend to be more stable and gives better results than the DER method for the experiments carried out in this report.

Catalogue

List of Acronyms

DEL	Depth Estimation from Labeling [1]
DER	Depth Estimation from Refocusing [2]
LF	Light-Field
MRF	Markov Random Field
RMS	Root Mean Square
RMSE	Root Mean Square Error
VR	Virtual Reality

[1] "Depth Estimation from Labeling" is the abbreviation of "Accurate Depth Map Estimation from a Lenslet Light Field Camera".

[2] "Depth Estimation from Refocusing" is the abbreviation of "Depth from Combining Defocus and Correspondence Using Light-Field Cameras".

List of Figures

List of Tables

Acknowledgments

My deepest gratitude goes first and foremost to my supervisor, Dr. Pan Agathoklis, for his constant encouragement and guidance. He has walked me through all the stages of the works of this project. Without his consistent and illuminating instruction, this project report could not have reached its present form.

Second, I would like to express my heartfelt gratitude to Professor Kin Fun Li for all his kindness and help during my research.

Last my thanks would go to my beloved family for their loving considerations and great confidence in me all through these years of university time. I also owe my sincere gratitude to my friends and my fellow schoolmates who gave me their help and time in listening to me and helping me work out my problems during the difficult course of the project.

Chapter 1. Introduction

1.1 Light Field Camera

The light-field (LF) camera involved in this paper is also called lenslet light-field camera (Jeon et al., 2016) and plenoptic light-field camera (Ng et al., 2005). Photographs imaged by this kind of camera record light fields in space, which can be represented as a 4D plenoptic function.

1.1.1 Light Field and Plenoptic Function

The original form of a plenoptic function defines a light field with seven parameters as below:

$$L = L(\theta, \phi, \lambda, t, X, Y, Z) \tag{1.1}$$

Adelson and Bergen (1991) explained the concept of this 7D function as that one can imagine placing an idealized eye at a stereoscopic position (X, Y, Z) and recording the intensity of the light rays passing through the pupil center at a stereoscopic angle (θ, ϕ), for wavelength λ, at time t.

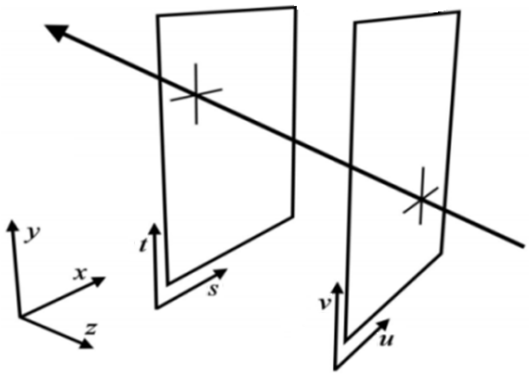


Figure 1.1: 4D light field representation in space (Dansereau and Bruton, 2004)

The 7D function can be simplified to a 5D function without the considerations of the wavelength λ and the time t, while it either can be reduced to a 4D function (Gortler et al., 1996) as $L(u, v, s, t)$. The 4D form, $L(u, v, s, t)$, only considers the location and direction of each ray in free space and supposing each ray has the same intensity value at every point along its propagation path (see figure 1.1). Here, (s, t) and (u, v) are coordinates of two parallel image planes. It is well-known that if the coordinates of two points in two parallel planes are given, the angle of the line between these two points is determinate. It means that the denotation $L(u, v, s, t)$ states not only the scene point position but also the angle of the ray between (s, t) and (u, v) planes.

1.1.2 Light-Field Camera Structure

The way a LF camera captures the 4D LF information can be explained with the help of the LF camera structure. The bottom of figure 1.2 shows that a microlens array is placed between the main lens and the photosensor of a LF camera, which does not exist in conventional cameras (the top of figure 1.2). It can be seen from the bottom of figure 1.2, after a ray emitted from an object point passes through the main lens and converges behind it and then the ray passes the microlens array. The ray is split into several individual rays by the microlens array before reaching the photosensor. When the plane of the main lens and the microlens array correspond to the (u, v) plane and (s, t) plane in figure 1.1, respectively, those two planes compose the 4D plenoptic model as mentioned above.

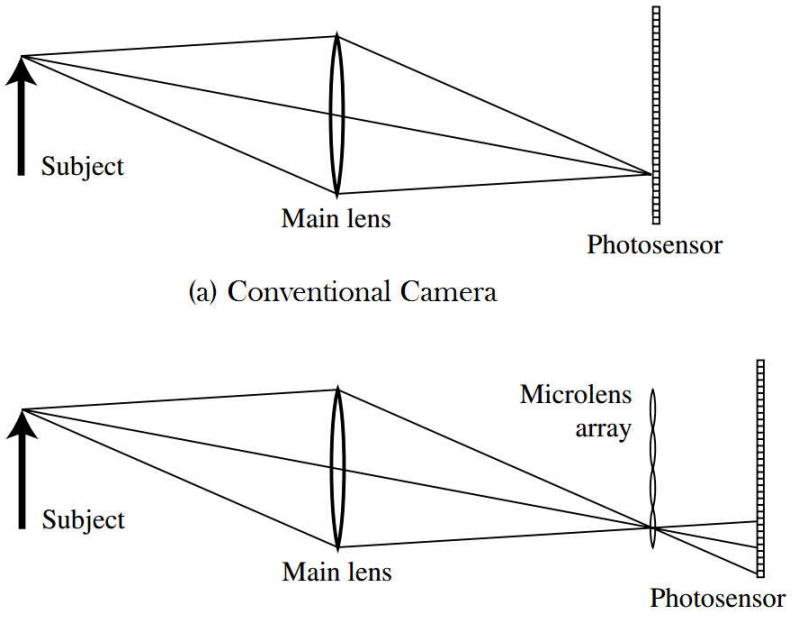


Figure 1.2: Imaging model of conventional camera and plenoptic camera: a. conventional camera structure; b. hand-held light-field camera structure

A more common notation of light field, $L(u, v, x, y)$, is used in digital LF cameras. Consider the LF camera equipped with a row of microlenses shown in figure 1.3, the light impinging on each microlens is split into three sub-parts, each corresponding to a particular incident angle. The corresponding part of sensor pixels beneath the microlens are called "macropixels", and a microlens and its macropixels compose a model of the pinhole camera. The image captured by an individual pinhole camera may be considered to be formed of macropixels, and each macropixel is subdivided into a set of three sub-pixels. The sub-pixels are labeled p_r, p_c, and p_l, since the light passing through the right

side, center, or left side of the main lens always strike at p_r, p_c, or p_l pixels no matter the object is in a distance equaling to (figure 1.3(a)), shorter than (figure 1.3(b)) or further than (figure 1.3(c)) the focusing distance.

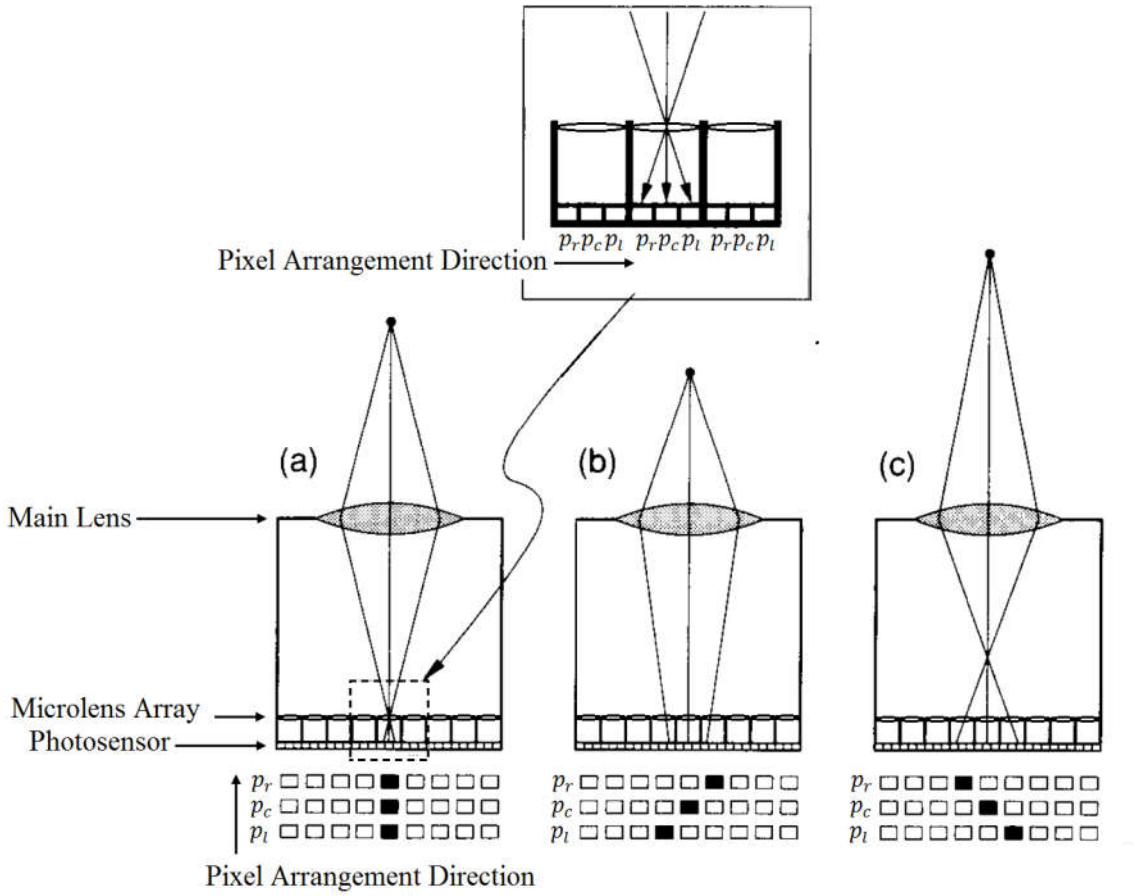


Figure 1.3: Array of miniature pinhole cameras placed at the image plane can be used to analyze the structure of the light striking each macropixel (Adelson and Wang, 1992, p.101)

In fact, each pinhole camera model forms an image, and this image captures the information about which sub-set of the light passed through a given sub-region of the main lens. If an object is on the plane of focus, as figure 1.3(a), then all three of the pixels p_r, p_c, and p_l of the center macropixel are illuminated. If the object is near or far, as shown in figure 1.3(b) and (c), the light is distributed across the pixels in a manner that is diagnostic of depth. For characterizing this distribution, separate sub-images are created from the p_r, p_c, and p_l pixels groups. The p_r sub-image describes light passing through the right side of the main lens, the p_c sub-image to light passing through the center, and the p_l sub-image to light passing through the left. The (u, v) coordinates of $L(u, v, x, y)$ are used to sort those three sub-images, since the p_r, the p_c and the p_l sub-images depict different parts of photograph imaged at the main lens plane, namely, the (u, v) plane. Either, the (x, y) coordinates of $L(u, v, x, y)$ correspond to the pixel locations in the

image plane of sub-images. The $L(u, v, x, y)$ is explained with a row of microlenses here, while the microlens row is changed to a microlens array in practice. As a default, x and y are called spatial coordinates since they represent the locations of pixels in sub-images, and u and v are called angular coordinates determining angles of rays. Figure 1.4 shows an example of a LF image denoted by $L(u, v, x, y)$.

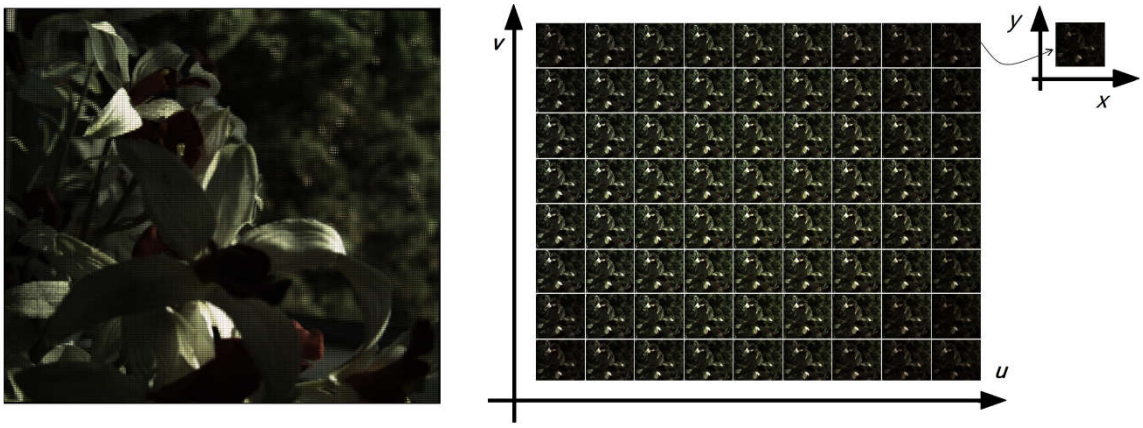


Figure 1.4: 9*9 LF sub-views (right) split from raw LF image (left)

1.1.3 Development History of Light-Field Imaging

After the description of the basic function of the LF camera, the history of the LF imaging is introduced here.

In 1908, Prof. Lippmann firstly proposed a remarkable technique for acquiring light field information to create "no glasses" stereoscopic 3D display. He entitled his technology "Photographies Integrales" or "Integral Photographs", and also laid the foundation of the light field imaging. After that, restricted by the lens imaging capability, much research was done to generate synthesized LF images by combining several images taken from different viewpoints at different times. The first one-step LF imaging solution was proposed in 1968 by Chutjian and Collier (1968), which imaged a LF picture with a one-time camera shot. After that, even though many techniques were developed to advance the process of Photographies Integrales, LF cameras were not achievable and affordable to ordinary researchers and customers until 2005. In 2005, the first hand-holding plenoptic camera was developed by Ng et al., (Ng et al., 2005) and it was called plenoptic camera 1.0. Later, several researchers contributed to the development of plenoptic cameras 2.0 including the methods to increase the spatial resolution of hand-hold LF cameras by Georgeiv and Intwala (2003), and Lumsdaine and Georgiev (2008), and a specialized plenoptic CCD sensor by Fife, Gamal, and Wong, (2008).

The history of LF imaging is simply introduced above, and more details can be found in Roberts and Smith's paper (2014).

Currently, two companies are leading the commercial market of LF cameras. One is Raytrix GmbH, founded by Christian Perwass and Lennart Wietzke, targeting industrial and scientific applications of light field camera. The other is Lytro, founded by Ren Ng, offering its first generation pocket-size camera from 2012, second generation high-resolution with a peak resolution of 4 megapixels from 2014, and an immerge high-end virtual reality (VR) video camera recently.

1.1.4 Features of Light-Field Camera

Before release, the most attractive feature of the first-generation LF camera published by Lytro is the refocus after shooting. The way to refocus a light-field image after its acquisition is outlined by Ng et al. (2005), while the refocusing process of the light filed is also called dramatic depth of field (Bishop and Favaro, 2012). The refocus technique allows customers to define what parts of the image being in focus or out of focus. Moreover, a LF image stores sufficient information to generate its corresponding depth map, which can be used to reconstruct 3D images and free view images of VR display.

1.2 Depth and Depth Map

In stereoscopic graphics, the depth value represents the distance from a scene point to a viewpoint. A depth map is an image representing depth values of all pixel in the corresponding image (see figure 1.5 right). Depth values are commonly defined as integral gray levels ranging from 0 to 255 representing the closest distance to the furthest. The gray-level representation of depth map can be used in various applications.

Figure 1.5: Picture and its depth map

A depth map is mostly used as a reference providing geometric information. One application from Seitz and Dyer (1996) implemented depth map to deal with image occlusions when creating synthetic views from two different located views. The removal and the substitute of background and object, as well as depth extension in stereoscopic videos, can also be achieved from using depth maps.

1.3 Existing Methods of Depth Estimation from Light-Field Image

Xu, Jin, and Dai (2015) presented a widely accepted classification of the existing methods of depth estimation for LF images: multiview stereo matching approaches and LF approaches.

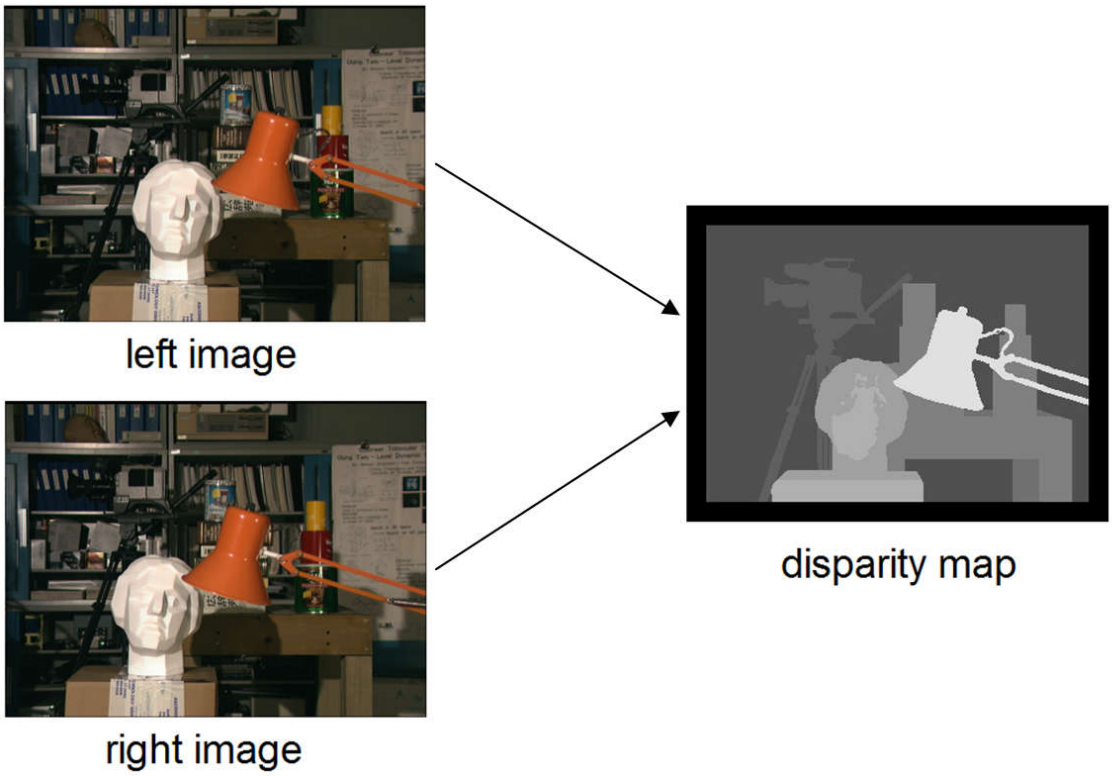


Figure 1.6: Stereo pair and its disparity map from stereo matching

A LF image can be treated as a multiview stereo image since it consists of sub-images of a scene captured from different angles. A particular example of multiview image is a stereo pair consisting of a left and a right image corresponding to human's left eye and right eye, respectively (see figure 1.6 left). To estimate depth map, a widely used way is to find disparities between two different views. The disparity refers to the distance between two corresponding pixels in the left image and the right, while the pixels are projected from a scene point. Section 2.1 explains the relationship between disparity and depth. However, the above disparity estimation approach cannot be directly applied to LF image because the narrow baseline (the distance between sub-aperture image pair) of LF image causes blurriness emergence during the estimation process. Therefore, more constraints and processes were added to estimate the depth maps from LF images. Bishop and Favaro (2010) proposed a stereo matching based algorithm with constraints of proper boundary conditions, where the term stereo matching was called LF photo-consistency matching. Jeon et al. (2016), Calderon, Parra, and Niño (2014), and Chen et al. (2017)

introduced the concept of multi-label shifts to perform stereo matching. Note that the depth estimation with high accuracy based on such approaches leads to extremely high computational complexity. Further, due to the lower resolution of the LF sub-aperture images, the quality of the generated depth from LF image is usually poorer than that from multiview acquisition system (Georgiev et al., 2013).

The LF approach to depth estimation is based on the structure of light field. Tao et al. (2014) shifted images to different focused depths based on the concept outlined by Ng et al. (2005) and processed the so-called defocus/refocus cues to determine depths. Tao et al. (2015) further developed the above depth estimation method through increasing the robustness of depth estimation model by adding one more depth constraint for depth combination. Similarly, Chen et al. (2010) analyzed the sharpness of focus-changing images and estimated depths from the most in-focus image. Further, Dansereau and Bruton (2004) developed an algorithm implementing the relationship between the slope line of epipolar image (EPI) and the depth, and Lv et al. (2015) worked out another algorithm not only utilizing the relationship between EPI slope and the depth but also the pixel correspondence of LF sub-images.

1.4 Objective and Motivation of Project

Even though there are lots of methods for LF depth estimations, not all of them result in accurate and high-quality depth maps. Therefore, the primary objective of the project is to implement and compare the performance of some existing methods of depth estimation from light field images.

Two representative algorithms are analyzed in details in this paper. The first one by Jeon et al. (2016) is entitled 'Accurate depth map estimation from a lenslet light field camera '. Tao et al. (2014) published the other method, named 'Depth from Combining Defocus and Correspondence Using Light-Field Cameras '. Those two approaches represent two main classes of existing depth estimation methods from LF images.

From the analyses of these two representative algorithms, the general approaches to estimate depth maps can be studied. Furthermore, the comparisons of these two representative methods indicates some advantages and disadvantages with respect to algorithm runtime and depth map quality.

For simplification, the method "Depth from Combining Defocus and Correspondence Using Light-Field Cameras" is abbreviated to depth estimation from refocusing (DER), while the "Accurate depth map estimation from a lenslet light field camera" is abbreviated to depth estimation from labeling (DEL).

1.5 Outline of Project Report

Five chapters constitute this report.

From chapter 2 to chapter 3, two representative algorithms, the DER method and the DEL methods, are described, respectively. Both of those chapters start from brief descriptions of employed basic concepts, followed by a more detailed description of the process at each step.

In chapter 4, the comparison and the analysis of these two methods in practice are presented, also with the discussions about their advantages and disadvantages from comparisons.

Chapter 5 provides conclusions from algorithm analysis and result comparisons, and describes the future works of depth estimation from light filed.

Chapter 2. DEL Method: "Accurate Depth Map Estimation from a Lenslet Light Field Camera"

This chapter describes the DEL method presented by Jeon et al. (2016). Based on the multi-label model of 3D scene construction, the DEL method applies the stereo matching of multiview images to estimate disparities. The content of this chapter begins with the descriptions of the basic concepts, followed by the step-by-step presentation of the DEL method.

2.1 Preliminaries

This section presents the way to convert disparity to depth, followed by the descriptions of a property of disparity in LF's used in image shifting. Then, three essential concepts are outlined: the multiview stereo matching, the multi-label model of 3D scene construction, and the image shifting based on the Fourier phase shift theorem for stereo matching. The disparity estimation in the DEL method is based on the above techniques.

2.1.1 Conversion from Disparity to Depth and Features of Disparity

The disparity is defined as the different location of a scene point seen by any two sub-images of a multiview image (a LF image is a multiview image). The DEL algorithm estimates disparities rather than depths directly. But, when certain intrinsic parameters (e.g. the focal length) of the camera are known, disparities are easy to be transformed to depths as following. As a simple example, the disparity-to-depth transform is illustrated by a model of shooting stereo image pair through two parallel cameras (c_l and c_r in figure 2.1). Figure 2.1 reveals two triangle relationship equations in geometry:

$$\frac{P_l}{f} = -\frac{h + X}{Z} \tag{2.1}$$

$$\frac{P_r}{f} = \frac{h - X}{Z} \tag{2.2}$$

where P_r and P_l are horizontal positions of the scene point P in the left and the right camera image planes. Equations 2.1 and 2.2 deduce an equation describing the relationship between the depth Z and the horizontal disparity $P_r - P_l$ as:

$$Z = \frac{2hf}{P_r - P_l} \tag{2.3}$$

where h is the half distance between two cameras and f is the camera focal length. f and h are constants and easy to be measured. If the values of P_r and P_l are known, the depth Z can be computed using equation 2.3. A simple relationship between the depth and the disparity is illustrated as above. More details about how to calculate the depth from the

disparity in more complicated systems, such as randomly placed camera systems, can be found in the paper by Wang, Ostermann and Zhang (2007), and the depth in such complicated systems still can be easily inferred from disparities. Therefore, provided adequate camera characteristic information, the depth estimation turns to be related to disparity estimation, and the disparity estimation is also called depth estimation in the DEL method.

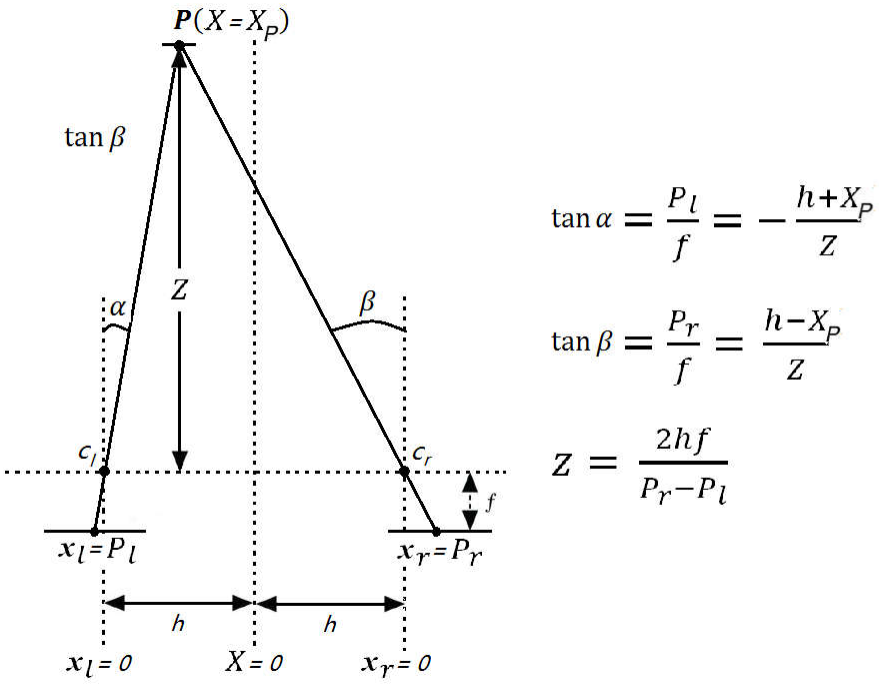


Figure 2.1: Model of stereo image capture consisting of two parallel placed cameras (X, x_l, and x_r: horizontal coordinates in space, in the left camera image plane and in the right camera image plane, respectively; X_P, P_l and P_r: horizontal positions of scene point P in space, in the left camera image plane, and in the right camera image planes, respectively; f: camera focal length; h: half distance between camera centers c_l and c_r; $P(X, Y, Z)$: 3D coordinate of point P; α and β: two angles)

When the distance between any two adjacent sub-lenses of a LF camera is always identical, the disparities in LF images have an essential feature. For demonstrating this feature, the stereo image capture model is rebuilt to a system with four parallel placed cameras as shown in figure 2.2. According to section 1.1.2, a LF camera can be treated as an array of cameras with constant intervals, while each camera in the array captures a sub-image from a different view angle. Therefore, the camera system in figure 2.2 can also be used to represent a LF camera consisting of 4 parallel placed sub-apertures.

With equation 2.3 and figure 2.2, an equation can be obtained as:

$$d_U = \frac{2hf}{Z} = d_{1,2} = d_{2,3} = d_{3,4} = {}^{d_{1,3}}/_{2} = {}^{d_{2,4}}/_{2} = {}^{d_{1,4}}/_{3} \tag{2.4}$$

where $d_{m,n} = |d_m - d_n|, (m, n = 1,2,3,4)$ means the disparity related to $Camera_m$ and $Camera_n$, and d_U is called the disparity unit for a point P.

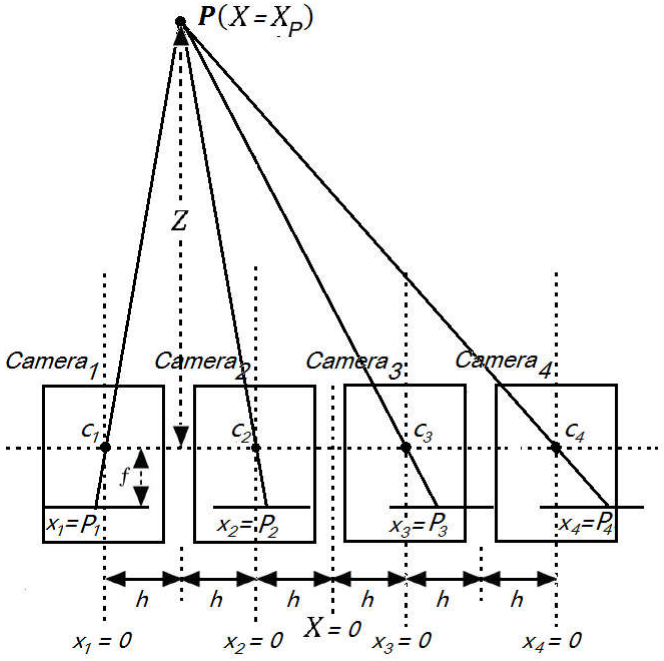


Figure 2.2: Model of three parallel placed cameras(P_1, P_2, P_3 and P_4: horizontal positions of scene point P in four camera image planes; f: camera focal length; c_1, c_2, c_3 and c_4: four camera centers; h: distance between neighbor camera centers; $P(X, Y, Z)$: 3D coordinate of point P)

Rewriting equation 2.4 to an inductive form as:

$$d_{m,n} = (m - n)d_U, \text{with } d_U = 2hf/Z \tag{2.5}$$

The disparity equation also can be extended to a 3D vector form in light fields as:

$$\boldsymbol{d}_{m,n} = \left(d^u_{m,n}, d^v_{m,n}\right) = \left((u_m - u_n)d_U, (v_m - v_n)d_U\right) \tag{2.6}$$

in which $d^u_{m,n}$ and $d^v_{m,n}$ are components of the disparity vector $\boldsymbol{d}_{m,n}$ in u-direction and v-direction, respectively, and (u_m, v_m) and (u_n, v_n) represent the coordinates of sub-images captured by $Camera_m$ and $Camera_n$ in the angular plane of light fields (see section 1.1.1).

From equation 2.6 it can be seen that disparities of a scene point P are integer multiples of the disparity unit d_U, while d_U is the disparity of P in any two adjacent sub-images, and the multiples linearly increase with the increasing intervals between sub-lenses. The

usage of this property in stereo matching is shown in the following section.

2.1.2 Disparity Estimation from Stereo Matching

The DEL method is based on stereo matching. According to stereo matching theory, all scene surfaces are assumed to be Lambertian surfaces (Lee, Ho and Kriegman, 2005) so that the visible luminance reflected from the surfaces to any observer at any angle is identical. It means that if a scene point can be seen by all LF sub-lenses, it will be imaged as a pixel in each sub-view. Also, the positions of the pixels depicting the same scene point in different sub-images may have disparities, but the irradiance values of those pixels are highly similar.

The irradiance similarity of corresponding pixels is an important cue to estimate disparity in stereo matching. The processes of conventional disparity estimation begin with shifting sub-images to a reference sub-image at possible disparities. If the capture of a scene point results in a particular disparity between a pixel in a non-reference sub-image and the other pixel in the reference image, the pixel in the non-reference sub-image will be relocated to the same position of the pixel in the reference sub-image after shifting the image at the disparity. The small variance of pixel irradiance between those two pixels is a measure to determine whether the associated disparity is the desired one. The variance in the DEL method is related to pixel irradiance similarity via pixel irradiance matching and directional gradient matching within an image window. Sections 2.2.1 describes the construction of this variance in detail.

The strategy of pixel-wise stereo matching in the DEL method is demonstrated in the following. Given a LF image, the distances between any sub-image and a reference sub-image in the (u, v) plane are constants. Several disparity units d_U's are pre-defined as trial variables. With these d_U's, if a pixel satisfies the stereo matching conditions when it shifted at a test disparity related to d'_U, the d'_U-related disparity is estimated as the desired disparity for the pixel.

2.1.3 Multi-label Model in 3D Scene Construction

As the pixel-wise stereo matching in the DEL method leads to pixel-wise distortions, a global optimization is needed to improve continuity and accuracy of the estimated disparity map. A method with high computational speed and high-quality output, named graph-cut optimization for multi-label, was employed as this optimization. One of the main reasons for the introduction of the multi-label model is to perform the graph-cut. Following the definition of the multi-label model, a depth label in the multi-label model depicts a plane in stereoscopic space, and all label planes are perpendicular to the optical axis of a camera (see figure 2.3). Also, the closest label plane to a camera is marked with the depth label N and the farthest with the depth label 1 when using a label set $\mathcal{L} = \{l | l = 1,2,3, \dots, N\}$ to label the 3D space. The depth label values of the label planes

decrease by one from current plane to the next further plane.

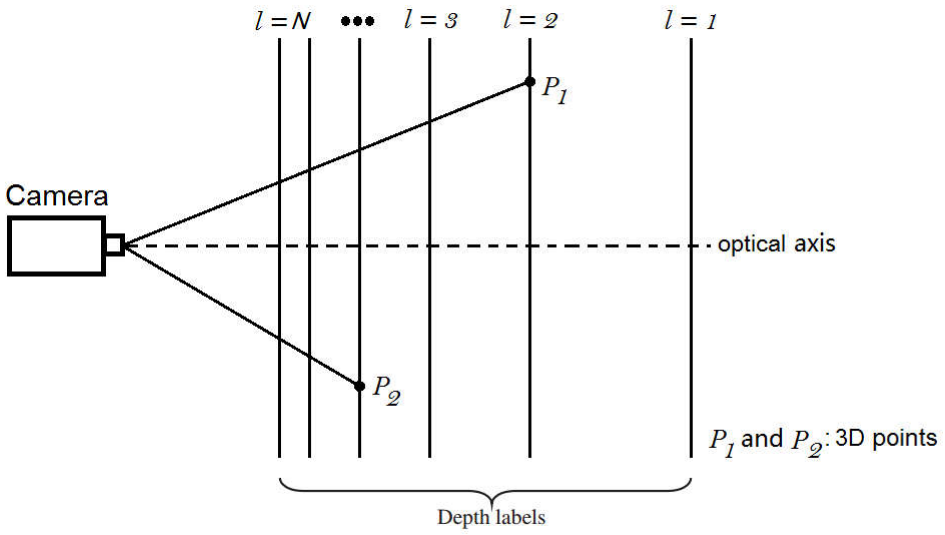


Figure 2.3: Multi-labels model in 3D space (Kolmogorov and Zabih, 2002, p.87)

A 3D-point P (e.g. P_1 and P_2 in figure 2.3) is the intersection of the light ray corresponding to P and a label plane l. It can be seen that all 3D points laid in a label plane have equivalent depths, which also means they have the same disparity. The points which are not on a corresponding label plane will be grouped to the nearest label plane to these points. To connect the label value to possible disparity, the unit disparity d_U defined in equation 2.4 is set to be

$$d_U = lk \tag{2.7}$$

where l $(l = 1,2, \dots, N)$ is the label value, k denotes the unit of the depth label in pixels. Then, for a scene point with a disparity unit d_U, the point disparities between two LF sub-images described in the equation 2.6 become

$$\boldsymbol{d}_{m,n} = \big((u_m - u_n)lk, (v_m - v_n)lk\big) \tag{2.8}$$

2.1.4 Image Shift Based on the Fourier Phase Shift Theorem

An essential step of disparity estimation through stereo matching is the image shift. We will use an example of the integer-pixel shift to explain the image shift concept. An expected disparity from the non-reference sub-image to the reference sub-image of a point is M pixels in the x-direction of the image plane. Remapping the pixels from their original position (x, y) to new posts at $(x + M, y)$, the pixel of the point in the non-reference sub-image will have the same position as the pixel of the point in the reference sub-image. However, an integer-pixel shift cannot always satisfy the image shift for LF images. This is not only because the desired disparity can be decimal times of a pixel, but

also because the narrow baselines (distances between two sub-lenses) of LF cameras lead to disparities less than 1 pixel. The DEL method introduced the 2D Fourier transform to solve the image shift issues in LF's. Following the Fourier phase shift theorem of digital image (Bracewell., 2004), a 2D shift of an image $I(x, y)$ with a vector $(\Delta x, \Delta y) \in \mathbb{R}^2$ in the spatial domain is same as the image multiplies a linear phase terms $exp^{2\pi i(\Delta x+\Delta y)}$ in the frequency domain, namely,

$$\mathcal{F}\{I(x + \Delta x, y + \Delta y)\} = \mathcal{F}\{I(x, y)\}exp^{2\pi i(\Delta x+\Delta y)} \tag{2.9}$$

where $\mathcal{F}\{\cdot\}$ denotes the discrete 2D Fourier transform. The shifted image can be obtained by the inverse Fourier transform $\mathcal{F}^{-1}\{\cdot\}$ as

$$I(x + \Delta x, y + \Delta y) = \mathcal{F}^{-1}\{\mathcal{F}\{I(x, y)\}exp^{2\pi i(\Delta x+\Delta y)}\} \tag{2.10}$$

The Δx and Δy are treated as the disparity components in the vertical direction and the horizontal direction, respectively. As a consequence of equation 2.8, the changes of Δx and Δy following the offsets of the non-reference sub-images to the reference sub-image:

$$\begin{aligned} \Delta x &= lk(u - u_r) \\ \Delta y &= lk(v - v_r) \end{aligned} \tag{2.11}$$

where (u, v) and (u_r, v_r) are the angular coordinates of the non-reference sub-images and the reference sub-image.

2.2 Disparity Estimation using the DEL Method

The generation of the disparity map using the DEL method requires four main steps.

Firstly, all LF sub-images excluding the reference one are substituted using the phase shift theorem. The shifted sub-images are used to build stereo matching cost functions with the reference sub-image (the cost function will be presented in section 2.2.1). This step is called cost volume construction, in which the shift degrees of each sub-image are related to the previously defined depth labels (see equation 2.11). If a label assembly $\mathcal{L} = \{l | l = 1,2,3, \dots, N\}$ is in use, the computations of constructed matching functions result in *N* cost slices for each sub-image.

As the cost volume construction focuses on measuring the similarity of pixels, it generates low signal-to-noise ratio disparity maps. Therefore, the second step, cost aggregation, is to locally remedy the previous noisy disparity maps by applying window-size guided filtering to each cost slice.

Thirdly, to globally improve the disparity map, a neighboring estimation named graph cut is used. The graph-cut optimization advances spatial smoothness of images while preserving image discontinuities (e.g. edges).

Finally, an iterative process is performed to refine the disparity map from discrete to non-discrete. The output of this step is the final disparity map of a LF image.

The details of these four steps are described in the following sections.

2.2.1 Cost Volume Construction

Two complementary cost volumes were used to match sub-view images: the absolute difference C_A and the gradient differences C_G.

C_A is defined as the minimum between two variables (equation 2.12). One is the absolute difference between corresponding pixel intensities I (or the root mean square (RMS) of RGB values), and the other is a threshold value τ_1 for robustness.

$$C_A(x, y, l) = min\{|I(u_r, v_r, x, y) - I(u, v, x + \Delta x, y + \Delta y)|, \tau_1\} \tag{2.12}$$

C_G denotes the cost of the directional gradient differences as

$$\begin{aligned} C_G(x, y, l) = \beta(u, v) min\{Diff_x(u_r, v_r, u, v, x, y, l), \tau_2\} \\ + \big(1 - \beta(u, v)\big) min\big\{Diff_y(u_r, v_r, u, v, x, y, l), \tau_2\big\} \end{aligned} \tag{2.13}$$

where τ_2 is another threshold value, and $Diff_x$ and $Diff_y$ are the differences between the directional gradients (I_x and I_y are x-directional and y-directional gradients) of the reference sub-image and the sub-images after shifting by $(\Delta x, \Delta y)$ as:

$$\begin{aligned} Diff_x(u_r, v_r, u, v, x, y, l) &= |I_x(u_r, v_r, u, v) - I_x(u, v, x + \Delta x, y + \Delta y)| \\ Diff_y(u_r, v_r, u, v, x, y, l) &= \big|I_y(u_r, v_r, u, v) - I_y(u, v, x + \Delta x, y + \Delta y)\big| \end{aligned} \tag{2.14}$$

The relative importance of two directional gradient differences $Diff_x$ and $Diff_y$ in equation 2.13 is controlled by $\beta(u, v)$, which is defined as:

$$\beta(u, v) = \frac{|u - u_r|}{|u - u_r| + |v - v_r|} \tag{2.15}$$

C_A and C_G are applied to construct a cost volume C for matching shifted sub-images to a reference image as below:

$$C(x, y, l) = \sum_{N_{u,v}} \sum_{(x,y) \in R_{x,y}} \big(\alpha C_A(x, y, l) + (1 - \alpha) C_G(x, y, l)\big) \tag{2.16}$$

where $N_{u,v}$ is the assembly of all coordinates in (u, v) plane. The summation within a rectangular region $R_{x,y}$ in the image plane is performed to reduce noise effect when finding correspondences using the sums of absolute differences, and $\alpha \in [0,1]$ controls the relevancy between C_A and C_G. For a depth label l', the cost $C(x, y, l')$ is called a cost slice, and the computations of function 2.16 for all label values result in N (N is the

number of distinct labels) cost slices.

2.2.2 Cost Aggregation

The purpose of cost aggregation is to locally eliminate the disparity outliers via smoothing while preserving edges. An ideal filter for this purpose should keep the value of pixels in borders unchanged and average the pixel values within local patches. One filter meeting the edge-preserving requirement is the bilateral filter (Paris et al., 2008), but the application of the bilateral filter requires many computations. A faster filter, named guided filter (He, Sun and Tang, 2013), had been introduced to the DEL method. The guided filter performs linear filtering and has almost comparable output quality to the bilateral filter.

The performance of the guided filter is defined as below. Given a guidance image I_g and an input image I_{in} (I_{in} can be identical to the guidance image), it is assumed that the output image I_{out} is a linear transform of a window w_p of I_g centered at a pixel p:

$$I_{out}(x,y) = a_p I_g(x,y) + b_p, \forall (x,y) \in w_p \tag{2.17}$$

where a_p and b_p are linear coefficients assumed to be constants in w_p. With equation 2.17, the difference between the output I_{out} and the input I_{in} is denoted by $n(x,y)$ as:

$$n(x,y) = I_{out}(x,y) - I_{in}(x,y) = a_p I_g(x,y) + b_p - I_{in}(x,y) \tag{2.18}$$

Then, the optimal solution of I_{out} can be obtained from minimizing the following cost function:

$$E(a_p, b_p) = \sum_{(x,y)\in w_p} \left(\left(a_p I_g(x,y) + b_p - I_{in}(x,y) \right)^2 + \epsilon {a_p}^2 \right) \tag{2.19}$$

where a penalty ϵ handles large ${a_p}^2$. Equation 2.19 obeys the linear ridge regression model (Draper and Smith, 1998), and its minimum is reached when a_p and b_p satisfy:

$$a_p = \frac{\frac{1}{|w_p|} \sum_{(x,y)\in w_p} I_g(x,y) I_{in}(x,y) - \mu_p \overline{I_{in}}}{\sigma_p^2 + \epsilon} \tag{2.20}$$

$$b_p = \overline{I_{in}} - a_p \mu_p \tag{2.21}$$

In equations 2.20 and 2.21, $|w_p|$ is the number of pixels in w_p, μ_p and σ_p^2 are the mean and variance of I_g in w_p, respectively, and $\overline{I_{in}}$ is the average for $I_{in}(x,y)$ across w_p. With the values of (a_p, b_p), the filtered output I_{out} is computed using equation 2.17. The procedure of guided filtering is shown in algorithm 2.1.

Algorithm 2.1: Guided Filter for Image (He, Sun and Tang, 2013, p.1400)

Input:

I_{in}: input image for filtering; I_g: guidance image;

r: filter window radius; ϵ: penalty

Output:

I_{out}

Steps:

1. $corr_{I_g} = \text{mean}(I_g .* I_g)$

   $corr_{I_g,I_{in}} = \text{mean}(I_g .* I_{in})$

2. $var_{I_g} = corr_{I_g} - \text{mean}(I_g) .* \text{mean}(I_g)$

   $cov_{I_g,I_{in}} = corr_{I_g,I_{in}} - \text{mean}(I_g) .* \text{mean}(I_{in})$

3. $a = cov_{I_g,I_{in}} ./ (var_{I_g} + \epsilon)$ *Eqn.(2.20)*

   $b = mean(I_{in}) - a .* mean(I_g)$ *Eqn.(2.21)*

4. $I_{out} = mean(a) .* I_g + mean(b)$ *Eqn.(2.17)*

$mean(.)$ is an average filter process

For analyzing the edge-preserving feature of the guided filter, two typical cases are considered:

Case 1: There are edges causing dramatic intensity changes within w_p, which results in a variance $\sigma_p^2 \gg \epsilon$. Therefore, $(a_p, b_p) \approx (1,0)$ can be computed from equations 2.20 and 2.21. The output computed from equation 2.17, in this case, is same as the guidance image, namely, $I_{out}(x,y) = I_g(x,y)$ within w_p.

Case 2: If the pixel values within w_p are equal to the same constant, the part of image I within w_p is a flat patch with $\sigma_k^2 \ll \epsilon$. Therefore, equations 2.20 and 2.21 result in $(a_k, b_k) \approx (0, \mu_k)$, and the output computed from equation 2.19 ,in this case, equals to the mean of intensity in w_k, i.e., $I_{out}(x,y) = \mu_k$ within w_p.

These two cases indicate that the guided filter keeps the edge pixel values unaltered and smoothes the other pixels within a window by taking the average.

In the DEL method, the reference image is set as the guidance image, and the guided filtering is done for all non-reference sub-views. A disparity map composed by depth labels can be generated after the guided filtering. The disparity value of each pixel in the disparity map is selected from the corresponding depth labels of the pixels in all cost slices. The depth label selection follows the winner-takes-all strategy. According to the definitions of label cost in section 2.2.1, it can be seen that the lower the cost volume value is, the more similar the corresponding pixels are, which means the depth label resulting in the lowest cost should be selected as the associated label to the desired

disparity. However, the label selection in the DEL method is embedded to the next step, the global optimization.

2.2.3 Global Optimization of Disparity Map via Graph Cut

The continuity and accuracy of disparity maps is further improved by a fast global optimization named graph-cut optimization.

The objective function of the graph-cut global minimization is built as below:

$$\underset{f}{minimize}\, E(f) = \lambda_{data} E_{data}(f) + \lambda_{smooth} E_{smooth}(f) \tag{2.22}$$

where f denotes a setting of assigning each pixel $p \in \mathcal{P}$ ($\mathcal{P}$ is a pixel assemble of an image) to a depth label $f_p = l$. The data term E_{data} measures how well the label f_p fits p given an observed data, the smoothness term E_{smooth} makes f smooth within spatial neighbors, and λ_{data} and λ_{smooth} control the weights of E_{data} and E_{smooth} to E.

Before defining E_{data} and E_{smooth}, an assembly $\mathcal{N}$ of neighboring pairs $\{p, q\}$ is introduced as:

$$\mathcal{N} \subset \{\{p, q\} | p, q \in \mathcal{P}\} \tag{2.23}$$

Under a 4-neighborhood system, the pixel p at (x_p, y_p) and q at (x_q, y_q) in the image plane are neighbors when their (x, y) index satisfy equation 2.24.

$$|x_p - x_q| + |y_p - y_q| = 1 \tag{2.24}$$

With neighboring pair assembly $\mathcal{N}$, the data term E_{data} is built as:

$$E_{data}(f_p) = C'(x, y, f_p) \tag{2.25}$$

where $C'(x, y, f_p)$ is the cost slice with $l = f_p$ after cost aggregation. The smoothness term E_{smooth} is defined as:

$$E_{smooth}(f_p) = \sum_{\{p,q\} \in \mathcal{N}} V_{\{p,q\}}(f_p, f_q) T(f_p \neq f_q) \tag{2.26}$$

where $T(\cdot) = 1$ if its argument is true otherwise $T(\cdot) = 0$, and $V_{\{p,q\}}(f_p, f_q)$ is a neighboring penalty function as:

$$V_{\{p,q\}}(f_p, f_q) = (I_p - I_q)^2 \tag{2.27}$$

Note that the introduction of $T(f_p \neq f_q)$ is to preserving edges during smoothness process.

The common way to minimize $E(f)$ requires high computational complexity, which is caused by three elements:

1) $E(f)$ probably has many local minima (e.g. $E(f)$ is not convex), which produces masses of computations in filtering local minima to a global minimum.

2) The space of possible labeling has dimension $|\mathcal{P}|$. $|\mathcal{P}|$ is equal to the pixel amount of an image, which tends to be many thousands for typical LF images.

3) The computational cost is significant for the multi-label case since there are multiple label options for labeling a pixel, and each labeling setting needs a computation of cost.

An appropriate graph-cut algorithm alleviates the effects of the above. The graph cut algorithm adopted in the DEL method is from Boykov, Veksler and Zabih (2001). The following demonstrates the definition of this algorithm and the brief descriptions about how this algorithm alleviates those three effects.

The basic idea of graph cut is to classify each pixel to an "object" or a "backgrounds" for a labeling. For example, if labeling a pixel to a label l reaches the minimum of $E(f)$, p is determined as an "object" of l. Otherwise, p is set to a "background". The optimal labeling is obtained after iterations until the objective function of graph cut is converged, which means there is no labeling change further decreasing $E(f)$. Due to that only two classes used, pixels can be classified only as binary segmentations with two labels modeled by a graph cut. However, it is possible to extend the binary-label case to multi-label case if one depth label is set as an "object" label and all the others are grouped to "backgrounds". The detailed procedure of multi-label graph cut will be demonstrated later after the description of α-expansion.

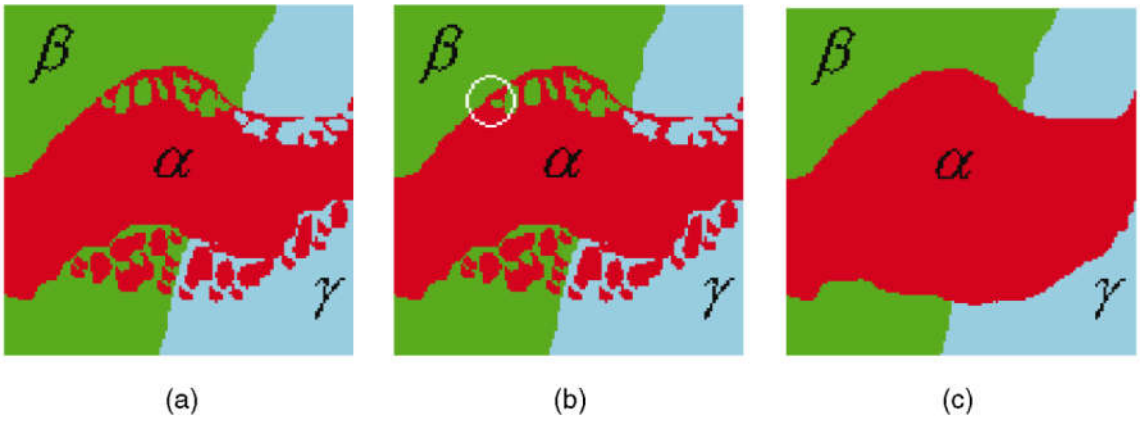


Figure 2.4: Examples of a standard and a large move. (a) a given initial labeling, where $\alpha, \beta, \gamma \in \mathcal{L}$; (b) a standard move which only changes the label of a single pixel (in the circled area); (c) α -expansion: allows a large number of pixels to change their labels to label α simultaneously. (Boykov, Veksler and Zabih, 2001, p.1255)

To get a faster computation, an important improvement of the used graph cut is the performance of expansion moves, instead of a standard move in certain common graph cut methods. Conventionally, during one iteration of minimizing $E(f)$, only a single pixel is allowed to change its label within a standard move (see figure 2.4(b)). As a consequence, finding the final optimum needs a large number of iterations. In contrast, within an α-expansion move, the label of all pixels differing from a chosen label $\alpha \in \mathcal{L}$ are allowed to be changed to α if this label change optimizes $E(f)$. Meanwhile, all pixels with current label α keep their label α (see figure 2.4). This expansion move is called α-expansion because label α is given a chance to grow.

Algorithm 2.2 shows the procedures of the graph-cut algorithm with α-expansion. A "for" loop for all labels is called an iteration, and the computation order for labels in an iteration can be fixed or arbitrary. An iteration is successful if a strictly better labeling is found after the iteration. The first unsuccessful iteration stops the algorithm as there is no labeling change which can further optimize the objective function. According to the experiments by Boykov, Veksler and Zabih (2001), the number of iterations is almost five until convergence, but the result after three iterations are practically the same.

Algorithm 2.2: α-expansion for Multi-Label Optimization
(Boykov, Veksler and Zabih 2001, p.1226)

f: initial labeling

repeat
  for each $\alpha \in \mathcal{L}$
    $\hat{f}_\alpha = argmin_{f_\alpha} E(f_\alpha)$, where f_α is an α-expansion of f
    if $E(\hat{f}_\alpha) < E(f)$
      set $f = \hat{f}_\alpha$
until converged

The analysis in the following show how graph cut decreases the high computational complexity caused by the three elements mentioned above.

Firstly, the base of graph cut space is a Markov Random Field (MRF), in which only neighbors have direct interactions with each other (Li, 2009). It has been shown that the global minimum in such a graph cut is the same as the local minima in certain degrees. Also, Boykov, Veksler and Zabih (2001) proved that when expansion moves are allowed a local minimum $\hat{f}$ is within a known factor c of the global minimum f^*:

$$E(\hat{f}) \leq 2cE(f^*) \tag{2.28}$$

where the factor c is the ratio of the largest non-zero value of $V_{\{p,q\}}(f_p, f_q)$ to the smallest

non-zero value of $V_{\{p,q\}}(f_p, f_q)$ for neighboring pairs $\{p, q\}$:

$$c = \max_{p,q \in \mathcal{N}} \left(\frac{\max_{f_p \neq f_q \in \mathcal{L}} V_{\{p,q\}}(f_p, f_q)}{\min_{f_p \neq f_q \in \mathcal{L}} V_{\{p,q\}}(f_p, f_q)} \right) \tag{2.29}$$

Therefore, the computation for eliminating local minima to find the global minimum can be decreased by considering function 2.28.

Secondly, the computational time is decreased via replacing the standard move by the α-expansion move. Although this replacement does not reduce the space of possible labeling, it significantly increases the labels of pixels able to change in one iteration. Namely, it reduces the number of iterations.

Thirdly, under an α-expansion, labels only have two options even in multi-label case: stay unchanged or change to α, which is less complicated than that labels are allowed to alter to all possible labels during one optimization step.

2.2.4 Iterative Refinement of Disparity Map

The last step in the DEL method is to enhance the disparity map from discrete to non-discrete. An algorithm, named spatial-depth super resolution for range images, by Yang et al. (2007) was employed for this.

The first step of the enhancement is to rebuild the cost volume based on the depth label map after graph-cut optimization. To permit large label variances, the cost should grow up with the increasing difference between the current label l_0 in disparity map and the potential label candidate l, and becomes constant when the label difference exceeds a search range. Hence, the cost is defined as a truncated quadric function:

$$C_Q(x, y, l) = min\{\eta * R, (l - l_0)^2\} \tag{2.30}$$

where R (e.g. $R = 5$) is the search range for depth labels, and η is a constant. The cost function is built in a squared difference form since quadratic polynomial interpolation will be used for pixel-wise cost minimization. The computations of the cost volumes $C_Q(x, y, l)$ result N cost slices for all label, followed by aforementioned guided filtering to all cost slices. Then, each pixel is set to the label producing the minimum C_Q at this pixel.

Due to that the depth labels l's are set as integers within a label range, the cost function is discontinuous (lack function values when label values are not integers). For reducing the discontinuities, a sub-pixel estimation algorithm is used based on quadratic polynomial interpolation. Because that if the cost function is continuous, the disparity with the minimum matching cost can be found (Yang et al., 2007), a continuous cost function

based on the quadratic polynomial model is defined as:

$$Q(l) = al^2 + bl + c, a > 0 \tag{2.31}$$

The minimum $Q(l^*)$ of $Q(l)$ reaches when the derivative of $Q(l)$ is 0:

$$Q(l^*)' = 2al^* + b = 0 \tag{2.32}$$

which results in the optimal l^* when

$$l^* = -\frac{b}{2a} \tag{2.33}$$

For calculating the parameters a and b, three discrete label candidates, l, l_+ and l_-, are introduced, where $l_+ = l + 1$ and $l_- = l - 1$ are the adjacent labels of l. Given the values of l, l_+ ,l_- and their corresponding costs C_Q's, a and b are solved by computing:

$$\begin{cases} C_Q(l) = al^2 + bl + c \\ C_Q(l_+) = al_+{}^2 + bl_+ + c \\ C_Q(l_-) = al_-{}^2 + bl_- + c \end{cases} \tag{2.34}$$

As a consequence, l^* is calculated from

$$l^*(x,y) = -\frac{b}{2a} = l - \frac{C_Q(x,y,l_+) - C_Q(x,y,l_-)}{2\left(C_Q(x,y,l_+) + C_Q(x,y,l_-) - 2C_Q(x,y,l)\right)} \tag{2.35}$$

One iteration of this disparity map refinement consists of cost volume construction (equation 2.30), guided filtering of cost slices, minimum cost selection, and refined disparity computation (equation 2.35). The experimental results of the DEL method revealed that four iterations are sufficient for appropriate results (Jeon et al., 2016).

Algorithm 2.3: Iterative Refinement for Disparity Map

```
input
    dmap: disparity map
output
    dmap_r: disparity map after iterative refinement
procedure
    for iteration = 1 to 4
        cost = cost(dmap)                    △build up costs for disparity map using Eqn. (2.30)
        cost_f = guided_filter(cost)         △ using guided filter to filter cost slice
        dmap_h = select_minimum_cost(cost_f) △ select minimum cost for each pixel
        dmap_r = refine(dmap_h)              △ pixel-wise refinement using Eqn. (2.31)
    end for

    return dmap_r
```

With the understandings about used theories, the procedure of the iterative refinement of disparity map is designed following algorithm 2.3.

2.2.5 The Overall Procedure of the DEL Method

Algorithm 2.4: Accurate Depth Map Estimation from a Lenslet Light Field Camera

input
$L(u, v, x, y)$: a light field image
$L(u_r, v_r, x, y)$: a sub-image of L as a reference
l: pre-defined labels at integral values, e.g. $l = 1,2,3, \dots, l_{max}$
k: a constant related to label unit
output
$l(x, y)$: a label map containing disparity information
procedure
for $(l = 1; l \leq l_{max}; l += 1)$
for each u
for each v
$(\Delta x, \Delta y) = lk(u - u_r, v - v_r)$ △ compute disparity components for Fourier Transform using Eqn. 2.11
$L(u, v, x + \Delta x, y + \Delta y) = \mathcal{F}^{-1}\{\mathcal{F}\{L(u, v, x, y)\} exp^{2\pi i(\Delta x + \Delta y)}\}$ △ Fourier Transform using Eqn. 2.10
$C(x, y, l) = Cost\big(L(u, v, x + \Delta x, y + \Delta y)\big)$ △ cost volume construction, see section 2.2.1
$C_f(x, y, l) = Filter\big(C(x, y, l)\big)$ △ guided filtering, see section 2.2.2
end for
end for
end for
$l_{gc}(x, y) = GraphCut\left(all\ C_f(x, y, l)\right)$ △ global optimization of graph cut, see algorithm 2.2 in section 2.2.3
$l(x, y) = Refine\left(l_{gc}(x, y)\right)$ △ iterative refinement to change label map from discrete to continuous, see algorithm 2.3 in section 2.2.4
return $l(x, y)$

The overall procedure of the DEL method follows the steps from section 2.2.1 to section 2.2.4. For estimating a disparity map from a LF image via the DEL method, a sub-image of the LF image, $L(u_r, v_r, x, y)$ is chosen as a reference to compute shifting vectors $(\Delta x, \Delta y)$ (equation 2.11). Also, the range of the depth label values l's has to be defined as integers, e.g. $l = 1,2,3, \dots, l_{max}$. For each l, u, and v, the corresponding LF sub-image $L(u, v, x, y)$ is shifted to $L(u, v, x + \Delta x, y + \Delta y)$ according to equation 2.36.

$$L(u, v, x + \Delta x, y + \Delta y) = \mathcal{F}^{-1}\{\mathcal{F}\{L(u, v, x, y)\} exp^{2\pi i(\Delta x + \Delta y)}\} \tag{2.36}$$

The cost volume $C(x, y, l)$ of the shifted LF image is constructed for the given l's (see

section 2.2.1). After that, the guided filtering (see section 2.2.2) is performed to each $C(x, y, l)$, which outputs a filtered cost slice $C_f(x, y, l)$. All $C_f(x, y, l)$'s are used to compute a smooth label map $l_{gc}(x, y)$ via graph-cut optimization (see section 2.2.3). The final output is a label map $l(x, y)$ containing disparity information, which is obtained from transforming the discrete $l_{gc}(x, y)$ to a continuous $l(x, y)$ by the iterative refinement described in section 2.2.4. Algorithm 2.4 shows the overall procedure of the DEL method.

Chapter 3. DER Method: Depth from Combining Defocus and Correspondence Using Light-Field Cameras

The DER method presented by Tao et al. (2014) estimates depths from defocus information and correspondence information of refocused LF images. The two estimations from two different sources may result in two different depth maps, which are combined using Markov Random Fields for outlier correction and global smoothness improvement.

The content of chapter 3 starts from the introduction of the refocusing process for LF images and the basic concept of depth estimation in the DER method, followed by the explanations about why both the defocus information and the correspondence information are used to create a final depth map. The DER method is outlined at the end.

3.1 Preliminaries

The key process of the DER method is the refocusing process of LF images and two kinds of information, the defocus information and the correspondence information, are being used for estimating depth. The estimated depths from these two kinds of information have complementary relationships and the final depth map is a combination of these depths. This section introduces the preliminaries of the DER method, which contain the method to refocus captured LF images, the approach used to extract the defocus information and the correspondence information, and the reciprocal relationship between depths from those two kinds of information.

3.1.1 Light-Field Image Synthesis for Refocusing

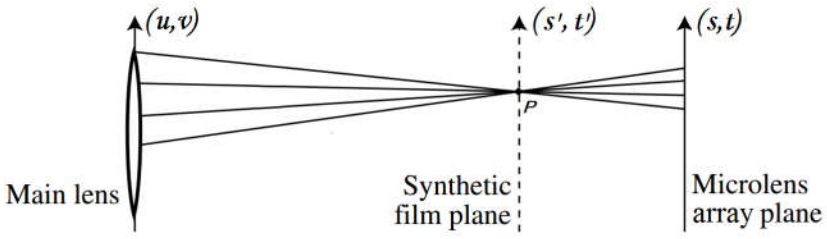


Figure 3.1: Conceptual model for synthetic photography (Ng et al., 2005, p.4)

Tao et al. (2014) introduced a refocusing process of light fields to create their DER algorithm. The objective of the refocusing process (Ng et al., 2005) is to use the light field between the main lens plane (u, v) and the microlens plane (s, t) to compute the synthetic photograph as if taken from the synthetic film plane (s', t') (see figure 3.1). Using the notation L' to denote the synthetic light field parameterized by the (u, v) plane and the synthetic (s', t') plane, $L'(u, v, s', t')$ defines the light traveling between (u, v) on the focal plane and (s', t') on the synthetic film plane.

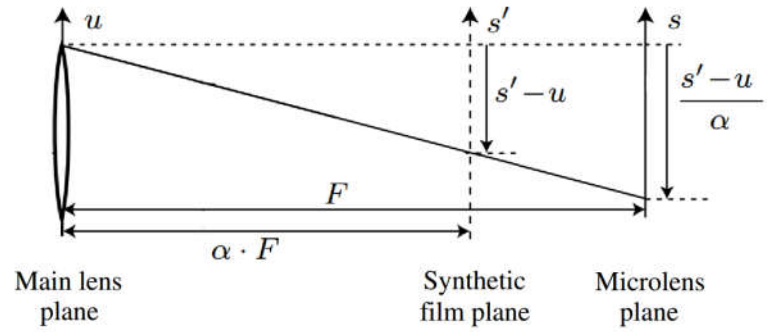


Figure 3.2: 2D model of light transmission within LF camera (Ng et al., 2005, p.4)

$L'(u, v, s', t')$ can be expressed by the known $L(u, v, s, t)$. In figure 3.2, only horizontal coordinates u, s' and s are considered. It can be seen that the value of the light intersecting plane u and s' also intersects the s plane at $u + (s' - u)/\alpha$. Therefore, $L'(u, s')$ is denoted as:

$$L'(u, s') = L\left(u, u + \frac{s' - u}{\alpha}\right) \tag{3.1}$$

where the refocusing parameter α controls the level of the changes of the focal length F. Extending the 2D form in equation 3.1 to a 4D form, the synthetic light field is given by:

$$L'(u, v, s', t') = L\left(u, v, u + \frac{s' - u}{\alpha}, v + \frac{t' - v}{\alpha}\right) \tag{3.2}$$

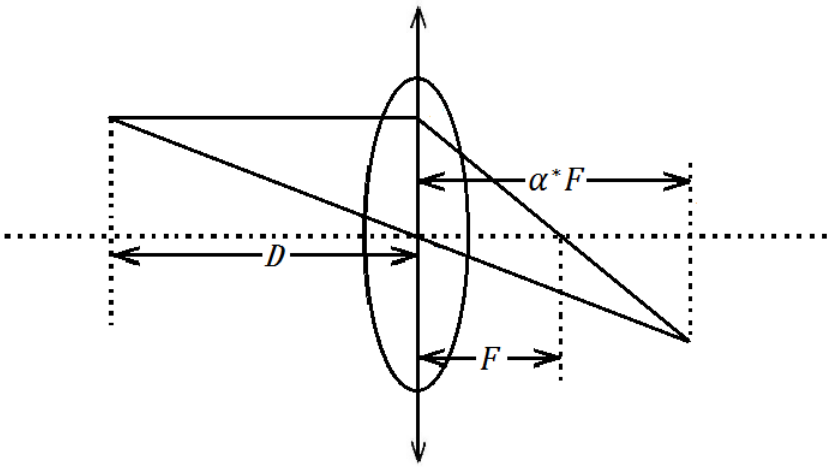


Figure 3.3: Demonstration of the principle of convex lens imaging

Also, according to the principle of convex lens imaging, when using $\alpha^* F$ to denote the in-focus length of a scene point, the depth D is connected to α by:

$$\frac{1}{D} = \frac{1}{F} - \frac{1}{\alpha^* F} \tag{3.3}$$

(see figure 3.3).

3.1.2 Features of Refocused LF Images for Depth Estimations

The focusing lengths $\alpha^* F$ are determined according to two properties of LFs.

Firstly, if a camera focuses on a scene point, all irradiance rays emitted from this point will converge to a single sub-lens. Otherwise, when the point is defocused, the rays are dispersed to several sub-lens. Therefore, for the LF image pixels depicting the same scene point but imaged at different focal lengths, the spatial contrast of the in-focus pixels is higher than that of the in-defocus pixels. This property associated with the differences of spatial contrast is called the defocus information of depth in the DER method.

Secondly, when all LF sub-images are refocused to a scene point, the corresponding pixels to this point in all LF images are relocated to the same position in the image plane. It is because that light rays emitted from a scene point are impinging on an individual sub-lens when the point is focused (see figure 3.4). Moreover, the corresponding pixels have similar irradiances since under the assumption of Lambertian surface (Lee, Ho and Kriegman, 2005) the light rays emitted from an object surface point have isotropic irradiance. This feature related to corresponding pixels is used as the correspondence information for estimating depths.

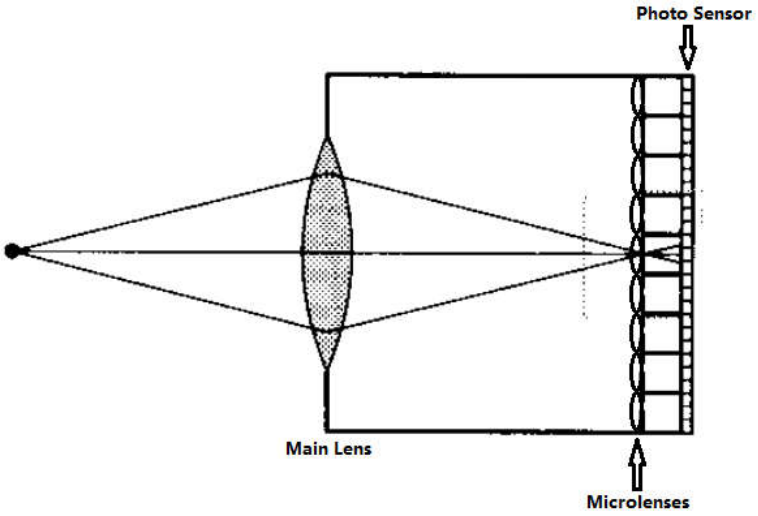


Figure 3.4: The image of a point focused at the microlens plane of a LF camera (Adelson and Wang, 1992, p.101)

3.1.3 Complementation across Depths from Defocus Information and Correspondence Information

The depth maps measured from the defocus information and the correspondence information complement limitations from each method. Therefore, the depths from the two kinds of information are fused to generate a more accurate depth map.

Table 3.1: Strengths and Weaknesses of Depth Estimation from Defocus Information and Correspondence Information (Tao et al., 2014, p.2)

	Occlusions	Repeating Textures	Interference Regions with High Spatial Contrast	Noise
Defocus	• more affected • more stable	• less affected	• more affected	• provide better support with noise
Correspondence	• less affected • unstable if affected	• more affected	• less affected	• more affected

Table 3.1 lists four complementary aspects between these two kinds of information. Regarding the correspondence information, the depth estimated from the information commonly suffers the effects from occlusions, repeating textures and noise. In comparison, even though the estimated depth from the defocus information also has a low tolerance to occlusions, the patch-based variance measurements of the defocus information improve the stability over occlusion regions. Meanwhile, measuring the defocus information results in better depths when dealing with repeating textures and noise. However, the measurement of the defocused information causes a new issue. The out-of-focus areas yield high contrast interfering with the defocusing estimation based on spatial contrast.

3.2 Depth Estimation through DER Method

The DER method consists of three main steps.

The first process is to construct the model for depth estimation. The construction begins with refocusing LF images at different α values, followed by the depth estimations from the defocus information and the correspondence information.

Secondly, after depth estimations from two kinds of information, pixel-wise depths are selected to compose two depth maps corresponding to the defocus information and the correspondence information, respectively. At the same time, the confidences of depth value in both maps, which is used as weights for the depth map combination, are calculated relying on peak ratios of the depth. This process, named depth and confidence

estimation, leads to a depth map and a confidence map for each kind of information.

It is necessary to globally propagate the depth maps derived from the pixel-wise measures. A Markov Random Field (MRF) model is employed to perform the global propagation, in which the depth maps from different kinds of information are combined. This process is entitled as the MRF global optimization.

3.2.1 Construction of Depth Estimation Model

The equation 3.4 is a transformation of equation 3.2, which is used to achieve refocusing effects from 4D light fields.

$$L_\alpha(x, y, u, v) = L_0\left(u, v, x + u\left(1 - \frac{1}{\alpha}\right), y + v\left(1 - \frac{1}{\alpha}\right)\right) \tag{3.4}$$

In equation 3.4, L_α denotes the refocused light field from the original light field L_0 by α, x and y are the coordinates of sub-image plane, and the central viewpoint in the angular plane is located at $(u, v) = (0,0)$.

To alleviate the effects from occlusions, L_α is averaged across the (u, v) plane as:

$$\bar{L}_\alpha(x, y) = \frac{1}{|N_{u,v}|}\sum_{u,v} L_\alpha(u, v, x, y) \tag{3.5}$$

where $|N_{u,v}|$ represents the total number of pixels within a window $N_{u,v}$ in the (u, v) plane. The contrast-based measure for the defocus information is defined as:

$$D_\alpha(x, y) = \frac{1}{|w_D|}\sum_{x,y \in w_D} |\Delta\, \bar{L}_\alpha(x, y)| \tag{3.6}$$

in which $|w_D|$ is the number of pixels within an image window w_D, and Δ is the 2D Laplacian operator used to compute spatial contrasts.

Meanwhile, for estimating depths from the correspondence information, the variance $\sigma_\alpha^2(x, y)$ for a given pixel is firstly calculated from:

$$\sigma_\alpha^2(x, y) = \frac{1}{N_{u,v}}\sum_{u,v}\left(L_\alpha(u, v, x, y) - \bar{L}_\alpha(x, y)\right)^2 \tag{3.7}$$

Then, the depth measure from the correspondence information is defined as the mean of the standard deviations over a pixel window w_C in the (x, y) plane as:

$$C_\alpha(x, y) = \frac{1}{|w_C|}\sum_{x,y \in w_C} \sigma_\alpha(x, y) \tag{3.8}$$

D_α and C_α are called the defocus measure and the correspondence measure for a refocusing parameter α, respectively. This leads for each LF sub-image to n (n is the amount of pre-defined α 's for refocusing) defocus measure responses and n correspondence measure responses.

3.2.2 Depth Selection and Confidence Estimation

The depth selections from the two kinds of information (the defocus information and the correspondence information) follows different winner-takes-all strategies. According to section 3.1.2, the optimal α's maximize spatial contrasts in the defocus measure and minimize the angular variances in the correspondence measure. Therefore, for each pixel of a depth map, one winner-takes-all strategy is set to select optimal α's (α_D^*) from the highest defocus response, and the other is to select optimal α's (α_C^*) from the lowest correspondence response as:

$$\alpha_D^*(x,y) = arg\max_{\alpha} D_\alpha(x,y) \tag{3.9}$$

$$\alpha_C^*(x,y) = arg\max_{\alpha} C_\alpha(x,y) \tag{3.10}$$

The depth value d of a pixel having n (n is a positive integer) possible levels is a projection from α as:

$$d\big(\alpha(x,y)\big) = \frac{n\alpha(x,y)}{\alpha_{max} - \alpha_{min}}, \alpha \in [\alpha_{min}, \alpha_{max}] \tag{3.11}$$

With equation 3.11, the depths $d_D^*\big(\alpha_D^*(x,y)\big)$ in the depth map from the defocus information and $d_C^*\big(\alpha_C^*(x,y)\big)$ in the depth map from the correspondence information can be obtained from their corresponding α_D^* and α_C^*.

d_D^*s and d_C^*s may differ since they are obtained using different information. Hence, for combining d_D^*s and d_C^*s in the following global optimization step, the weights of d_D^*s and d_C^* for depth combination are also estimated. Two α's involved to the same pixel are picked out to compute the combination weights. One α is the local optimizer α^* computed from equation 3.9 or 3.10, and the other is the next optimizer α_2^* corresponding to the second optimal measure. In fact, the larger the gap between the responses resulted by α^* and α_2^* is, the more confident α^* is to be a global optimizer. Consequently, the confidences as weights are defined to be proportional to the ratio of α^* and α_2^* as:

$$D_{conf}(x,y) = D_{\alpha^*}(x,y)/D_{\alpha_2^*}(x,y) \tag{3.12}$$

$$C_{conf}(x,y) = C_{\alpha_2^*}(x,y)/C_{\alpha^*}(x,y) \tag{3.13}$$

3.2.3 Global Optimization in Markov Random Field (MRF)

In brief, a MRF is a space within which only neighbors have direct interactions with each other (Li, 2009). Concerning a depth map, if the whole map is treated as a MRF, the depth value of a pixel only depends on the depth values of its neighbors.

The purpose of MRF global optimization is to correct outliers of depth maps. The outliers are mainly from the pixel-wise depth estimations (see equations 3.6 and 3.8), which lose depth consistency in global.

To achieve the purpose of global optimization in MRF, an optimization problem is built in the following. Given the optimal depths $d_D^*\left(\alpha_D^*(x,y)\right)$ and $d_C^*\left(\alpha_C^*(x,y)\right)$ from section 3.2.2, the MRF global optimization is to optimize the following function:

$$\begin{aligned} &\min_Z E(d) \\ &E(d) = E_{data}(d) + E_{smooth}(d) \end{aligned} \tag{3.14}$$

where E_{data} is the data term defined by

$$\begin{aligned} E_{data}\big(d(x,y)\big) = D_{conf}(x,y)\left|d(x,y) - d_D^*\left(\alpha_D^*(x,y)\right)\right| \\ +C_{conf}(x,y)\left|d(x,y) - d_C^*\left(\alpha_C^*(x,y)\right)\right| \end{aligned} \tag{3.15}$$

and E_{smooth} is the smoothness term with two penalty parameters: λ_{flat} controlling the partial derivative for flatness of the output depth estimation map, and λ_{smooth} controlling the second derivative kernel, which enforces overall smoothness:

$$\begin{aligned} E_{smooth}\big(d(x,y)\big) = (\lambda_{flat}\left(\left|\frac{\partial d}{\partial x}\right|_{(x,y)} + \left|\frac{\partial d}{\partial y}\right|_{(x,y)}\right) \\ +\lambda_{smooth}|\Delta\, d|_{(x,y)}) \end{aligned} \tag{3.16}$$

where the absolute value of partial derivatives and $\Delta\, d$ is performed as the Laplacian operations, which can be obtained from the convolutions between d and three kernels:

$$\begin{aligned} &\left|\frac{\partial d}{\partial x}\right|_{(x,y)} = \|d \otimes \boldsymbol{F}_1\|^2 && \boldsymbol{F}_1 = [-1 \quad 0 \quad 1] \\ &\left|\frac{\partial d}{\partial y}\right|_{(x,y)} = \|d \otimes \boldsymbol{F}_2\|^2 \quad \text{with} && \boldsymbol{F}_2 = [-1 \quad 0 \quad 1]^T \\ &|\Delta\, d|_{(x,y)} = \|d \otimes \boldsymbol{F}_3\|^2 && \boldsymbol{F}_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \end{aligned} \tag{3.17}$$

As there may be no optimizer make both the data term and the smoothness term to be minimal, the optimization problem is transformed into an iterative form so that the solution can get closer and closer to the minima. The iterative is a transformation of function 3.14 as:

$$\min_{\boldsymbol{z}} \boldsymbol{\delta}$$
$$\boldsymbol{\delta}(\boldsymbol{z}) = |\boldsymbol{A}\boldsymbol{z} - \boldsymbol{b}| \tag{3.18}$$

where $\boldsymbol{A}$ is a matrix, $\boldsymbol{z}$ is a variable vector, and $\boldsymbol{b}$ is a constraint vector. Before showing the definitions of $\boldsymbol{A}$, $\boldsymbol{z}$, and $\boldsymbol{b}$, several notations need to be introduced. The first notation is $[\cdot]_{diag}$. For a matrix $\boldsymbol{M}$ with r rows and c columns as:

$$\mathrm{M} = \begin{bmatrix} \mathrm{M}_{11} & \mathrm{M}_{12} & \cdots & \mathrm{M}_{1c} \\ \mathrm{M}_{21} & \mathrm{M}_{22} & & \mathrm{M}_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{M}_{r1} & \mathrm{M}_{r2} & \cdots & \mathrm{M}_{rc} \end{bmatrix}$$

$[\boldsymbol{M}]_{diag}$ reshapes the matrix $\boldsymbol{M}$ to a diagonal matrix $\boldsymbol{M}_{diag}$ as below:

$$M_{diag} = \begin{bmatrix} \mathrm{M}_{11} & 0 & & & \cdots\cdots & & & 0 \\ 0 & \mathrm{M}_{12} & & & & & & \\ & & \mathrm{M}_{1c} & & & & & \\ & & & \mathrm{M}_{21} & & & & \vdots \\ \vdots & & & & \mathrm{M}_{22} & \ddots & & \\ & & & & \ddots & \mathrm{M}_{2c} & & \\ & & & & & & \mathrm{M}_{r1} & \\ & & & & & & & \mathrm{M}_{r2} \quad 0 \\ 0 & & & & \cdots\cdots & & 0 & \mathrm{M}_{rc} \end{bmatrix} \tag{3.19}$$

Another notation, $[\cdot]_{vect}$, sequentially stretches a matrix to a vector (e.g. the transformation from $\boldsymbol{M}$ to $[\boldsymbol{M}]_{vect}$ in equation 3.20).

$$[M]_{vect} = [\mathrm{M}_{11}\ \mathrm{M}_{12} \cdots \mathrm{M}_{1c}\ \mathrm{M}_{21}\ \mathrm{M}_{22} \cdots \mathrm{M}_{2c} \cdots \mathrm{M}_{r1}\ \mathrm{M}_{r2} \cdots \mathrm{M}_{rc}]^T \tag{3.20}$$

Thirdly, a zero matrix with the same size as an image I is denoted by $[0]_{\mathrm{size}(I)}$. Also, the intensity of a LF central sub-image, I_c, is introduced as a reference image for smoothness process.

The initial matrix A and the initial vector b in equation 3.18 are presented as A_0 and b_0. With above notations, A_0 and b_0 are defined in equations 3.21 and 3.22.

$$\boldsymbol{A}_0 = \begin{bmatrix} A1 \\ A2 \\ A3 \end{bmatrix}, \text{ with } \begin{aligned} \boldsymbol{A}1 &= \begin{bmatrix} [D_{conf}]_{diag} \\ [C_{conf}]_{diag} \end{bmatrix} \\ \boldsymbol{A}2 &= \lambda_{flat} \begin{bmatrix} \left[\left|\frac{\partial I_c}{\partial x}\right|_{(x,y)}\right]_{diag} \\ \left[\left|\frac{\partial I_c}{\partial y}\right|_{(x,y)}\right]_{diag} \end{bmatrix} \\ \boldsymbol{A}3 &= \lambda_{smooth}\left[|\Delta I_c|_{(x,y)}\right]_{diag} \end{aligned} \tag{3.21}$$

with

$$\boldsymbol{b}_0 = \begin{bmatrix} \boldsymbol{b}1 \\ \boldsymbol{b}2 \\ \boldsymbol{b}3 \end{bmatrix}, \quad \begin{aligned} \boldsymbol{b}1 &= \begin{bmatrix} [\boldsymbol{D}_{conf} .* \boldsymbol{Z}_D]_{vect} \\ [\boldsymbol{C}_{conf} .* \boldsymbol{Z}_C]_{vect} \end{bmatrix} \\ \boldsymbol{b}2 &= \begin{bmatrix} [[0]_{\text{size}(I_c)}]_{vect} \\ [[0]_{\text{size}(I_c)}]_{vect} \end{bmatrix} \\ \boldsymbol{b}3 &= [[0]_{\text{size}(I_c)}]_{vect} \end{aligned} \tag{3.22}$$

Using i to denote the index of an iteration, the optimal vector $\boldsymbol{z}_i^*$ is the solution for minimizing $\boldsymbol{A}_{i-1}\boldsymbol{z} - \boldsymbol{b}_{-1}$ in the least square sense as:

$$\boldsymbol{z}_i^* = \underset{\boldsymbol{z}}{argmin}(\boldsymbol{A}_{i-1}\boldsymbol{z} - \boldsymbol{b}_{i-1}) \tag{3.23}$$

where $\boldsymbol{z}$ is a vector containing the depths $d(x, y)$ for each (x, y). With the notation shown in equation 3.20, $\boldsymbol{z}$ can be written as $\boldsymbol{z} = [\boldsymbol{d}]_{vect}$. The gap, $\boldsymbol{\delta}_i$, between the current optimizer and the constraints can be got from computing function 3.24.

$$\boldsymbol{\delta}_i = |\boldsymbol{A}_{i-1}\boldsymbol{z}_i - \boldsymbol{b}_{i-1}| \tag{3.24}$$

An error, $error_i$, consists of $\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T$ and a constant softness factor $\epsilon_{softness}$, and a root mean squared error (RMSE), $RMSE_i$, are defined as:

$$error_i = \sqrt{\boldsymbol{\delta}_i\boldsymbol{\delta}_i^T + \epsilon_{softness}^2} \tag{3.25}$$

$$RMSE_i = mean(error_i) \tag{3.26}$$

Setting a converge fraction $\tau_{converge}$, the iteration stops when the RMSE meets the following condition:

$$\left|\frac{RMSE_{i-1} - RMSE_i}{RMSE_i}\right| < \tau_{converge} \tag{3.27}$$

If function 3.27 is not satisfied, an error weight matrix $\boldsymbol{W}$ is computed as

$$\boldsymbol{W}_i = [1./error_i]_{diag} \tag{3.28}$$

The error weight matrix $\boldsymbol{W}_i$ is used to soften $\boldsymbol{A}$ and $\boldsymbol{b}$ for the next iteration:

$$\begin{aligned} \boldsymbol{A}_i &= \boldsymbol{W}_i \boldsymbol{A}_{i-1} \\ \boldsymbol{b}_i &= \boldsymbol{W}_i \boldsymbol{b}_{i-1} \end{aligned} \tag{3.29}$$

Based on the description by Tao et al.'s (2014), the procedures of the applied iterative optimization is concluded to algorithm 3.1.

Algorithm 3.1: Iterative Optimization for Depth Map Combination and Enhancement

Object:

iteratively find optimizer z_i^* to minimize $|A_{i-1}z_i - b_{i-1}|$

Input:

A_0: initial data matrix — d_D:initial depth map from defocus

b_0: initial constraint vector — d_C: initial depth map from correspondence

$\epsilon_{softness}$: softness factor — $\tau_{converge}$: converge fraction

Procedure:

$RMSE_0 = 0$ △ Set initial RMSE to 0

$z_1^* = \underset{z}{argmin}(A_0 z - b_0)$ △ *Eqn. (3.23)*: compute first optimizer

$\delta_1 = |A_0 z_1^* - b_0|$ △ *Eqn. (3.24)*: initial error

$error_{\,1} = \sqrt[2]{\delta_1^2 + \epsilon_{softness}^2}$ △ *Eqn. (3.25)*: soften initial error weight factor

$RMSE_1 = mean(error_1)$ △ *Eqn. (3.26)*: average the initial error weight factors

for iteration i $(i > 0)$

if $\left(\left|\frac{RMSE_{i-1} - RMSE_i}{RMSE_i}\right| < \tau_{converge}\right)$ break

△ *Eqn. (3.27)*: if the current optimum compared to the previous optimum is below a converge fraction $\tau_{converge}$, stop the iteration

$w_i = [1./error_{\,i}]_{diag}$ △ *Eqn. (3.28)*: compute error weight matrix

$A_i = w_i * A_{i-1}$ △ *Eqn. (3.29)*: modify matrix *A* for next iteration

$b_i = w_i * b_{i-1}$ △ *Eqn. (3.29)*: modify vector *b* for next iteration

$z_{i+1}^* = \underset{z}{argmin}(A_i z - b_i)$ △ *Eqn. (3.19)*

$\delta_{i+1} = |A_i z_{i+1}^* - b_i|$ △ *Eqn. (3.20)*

$error_{\,i+1} = \sqrt{\delta_{i+1}^2 + \epsilon_{softness}^2}$ △ *Eqn. (3.21)*

$RMSE_{i+1} = mean(error_{\,i+1})$ △ *Eqn. (3.22)*

end for

3.2.4 The Overall Procedure of the DER Method

Algorithm 3.2: Depth from Combining Defocus and Correspondence Using Light-Field Camera (Tao et al., 2014)

input

$L_0(u,v,x,y)$: a light field image

α_{min} and α_{max}: the minimum and the maximum values of the parameter α

α_{step}: the gap between two adjacent αs

output

$d(x,y)$: a depth map

procedure

for $(\alpha = \alpha_{min}; \alpha \leq \alpha_{max}; \alpha += \alpha_{step})$

$L_\alpha(u,v,x,y) = refocus(L_0(u,v,x,y),\alpha)$

△ refocus light field image using equation 3.4

$D_\alpha(x,y) = defocus(L_\alpha(u,v,x,y))$

△ compute defocus response using equation 3.6

$C_\alpha(x,y) = corresp(L_\alpha(u,v,x,y))$

△ compute correspondence response using equation 3.8

end for

$\alpha_D^*(x,y) = \arg\max(D_\alpha(x,y))$ △ select optimal α for defocus

$\alpha_C^*(x,y) = \arg\max(C_\alpha(x,y))$ △ select optimal α for correspondence

$\{D_{conf}(x,y), C_{conf}(x,y)\} = conf(\{D_\alpha(x,y), C_\alpha(x,y)\})$

△ compute confidence using equations 3.12 and 3.13

$d(x,y) = MRF\left(\alpha_D^*(x,y), \alpha_C^*(x,y), D_{conf}(x,y), C_{conf}(x,y)\right)$

△ global optimization in Markov Random Field, see section 3.2.3

return $d(x,y)$

As a summary, this section describes the procedure of the DER method. Displayed in algorithm 3.2, the DER method procedure follows the steps from section 3.2.1 to section 3.2.3.

From the beginning of the DER method, for each α, all sub-images $L_0(u,v,x,y)$ of a LF image are refocused to $L_\alpha(u,v,x,y)$ (see equation 3.4), and the defocus measure $D_\alpha(x,y)$ and the correspondence measure $C_\alpha(x,y)$ are computed for $L_\alpha(u,v,x,y)$. Then, the optimal α's including α_D^*'s and α_C^*'s are selected for the depth maps from the defocus

information and the correspondence information, respectively. Meanwhile, the confidence $D_{conf}(x, y)$ and $C_{conf}(x, y)$ are computed as weights fro the depth map combination in MRF optimization step. With α_D^*, α_C^*, $D_{conf}(x, y)$ and $C_{conf}(x, y)$, the final depth map $d(x, y)$ is the output of the global optimization in MRF.

Chapter 4. Analysis and Comparisons of Studied Depth Estimation Methods

This chapter describes the experimental environments, and the comparison between the DEL method and the DER method, with respect to runtime and quality of the depth estimation.

4.1 Experimental Environments and Algorithm Illustrations

For evaluating the algorithm performance, the datasets offered by Heidelberg Collaboratory and the modified Matlab code (some invoked functions are written in c++) have been used. In the rest of this section, the reason why the Heidelberg Collaboratory datasets have been chosen and the output of each algorithm, step by step, is presented.

4.1.1 Experimental Datasets and Method

Before examining the algorithm performance, two problems have to be solved. Firstly, the in-hand LF camera, the first generation Lytro camera, captures photographs in high noise and low resolution. Using those photographs will lead to additional interferences in experimental results. Secondly, the DEL method and the DER method provide disparity map and depth map, respectively. Since some necessary camera parameters are unknown, it is impossible to transform or inversely transform from depth maps to disparity maps.

The benchmarks and datasets of densely sampled 4D light fields generated by Heidelberg Collaboratory (Honauer et al., 2016) are introduced to solve the above issues. The Heidelberg Collaboratory 4D light field datasets provide high SNR and high resolution LF images, which eliminate the effects from camera noise and low resolution. Furthermore, the datasets offer ground truth disparity maps and ground truth depth maps for each LF image, which means there are ground truth references for both the disparity map and the depth map. The advantages of these datasets have been described by Wanner, Meister and Goldluecke (2013).

4.1.2 Parameter Configurations and Algorithm Illustration of the Depth Estimation from Labeling (DEL) Method

The parameter values used in the DEL method are listed in table 4.1. The third column of table 4.1 also states the meaning of these parameters and where they are used. The disparity values in a disparity map can be deduced from the disparity value scale (a color bar) shown on the right side of each disparity map. The disparity values range from 0 to 75, while the higher the disparity value is, the closer the object is to the camera.

Table 4.1: Parameter configurations of DEL method

Sign	Value	Remark
N	75	Number of pre-defined depth labels for labeling 3D space
k	0.02	Label unit for image shift in Eqn. 2.7
α	0.5	Relevance controller of two costs in Eqn. 2.16
$R_{x,y}$	3*3	Window size of average denoising in Eqn. 2.16
τ_1	0.5	Truncation value for robustness in Eqn. 2.12
τ_2	0.5	Truncation value for robustness in Eqn. 2.13
r	5	Radius of guided filter windows $\lvert w_p \rvert$ in Eqn. 2.20
λ_{data}	2	Data constraint coefficient of graph cut in Eqn. 2.22
λ_{smooth}	0.009	Smooth constraint coefficient of graph cut in Eqn. 2.22

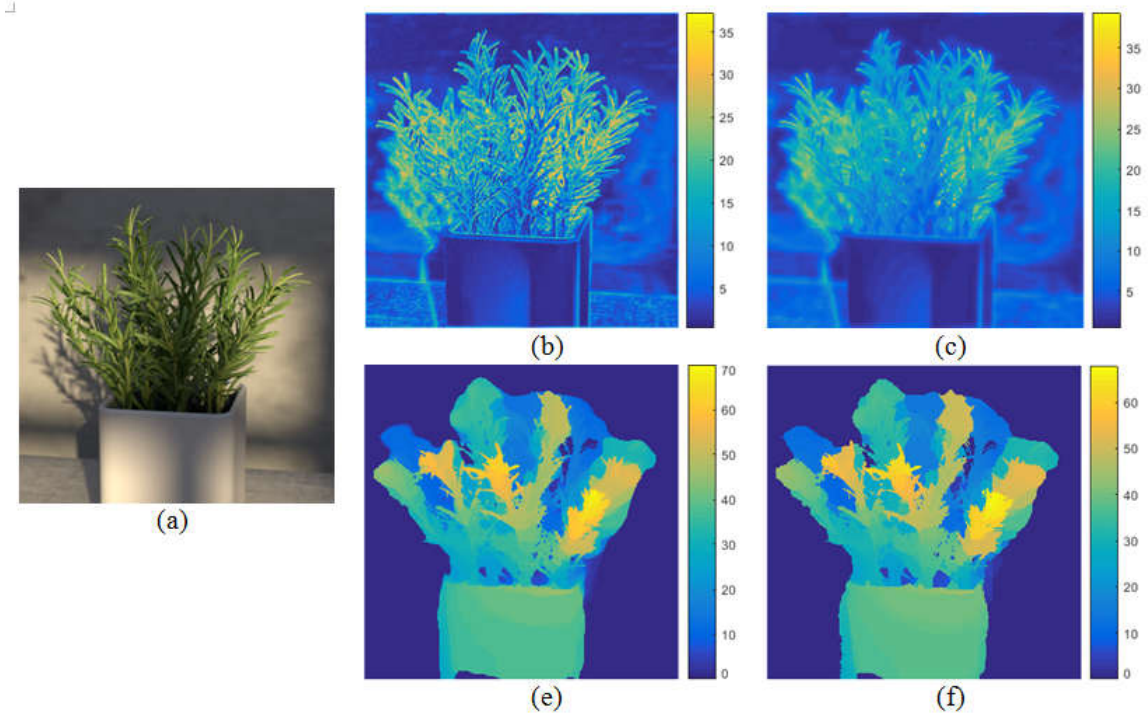


Figure 4.1: Illustration of the DEL method: (a) the central LF sub-image; (b) cost slice at *l=30* before guided filtering; (c) cost slice at *l=30* after guided filtering; (d) disparity map after graph-cut optimization; (e) non-discrete label map obtained from iterative refinement.

The LF image *rosemary* from the Heidelberg Collaboratory datasets is used to illustrate the DEL method process. Figure 4.1(a) displays the central view of *rosemary*. With the input image *rosemary*, the cost volume construction in the DEL method outputs N (N is the number of pre-defined depth labels for labeling 3D space) cost slices including the cost slice at label $l = 30$ (figures 4.1(b)). All cost slices are processed using guided filtering, and figure 4.1(c) shows the cost slice at label $l = 30$ after guided filtering. Through the aggregation of processed cost slices by the guided filter, the graph-cut optimization creates a smooth disparity map (figure 4.1(e)), and the iterative refinement further enhances the disparity map from discrete to non-discrete (see figure 4.1(f)).

4.1.3 Parameter Configurations and Algorithm Illustration of the Depth Estimation from Refocusing (DER) Method

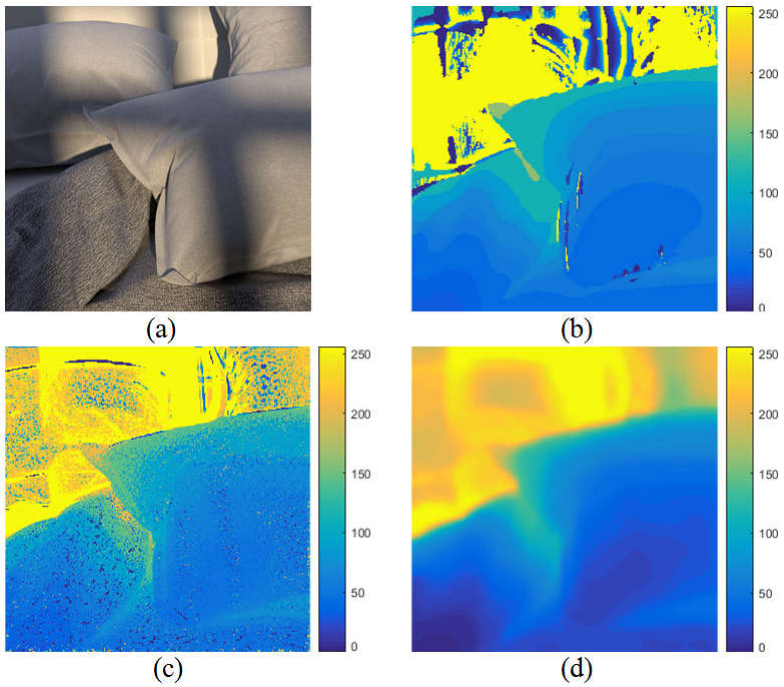


Figure 4.2: Flow demonstration of the algorithm of refocusing measurement. (a): central view of the LF image *pillows*; (b): depth map from defocus information; (c): depth map from correspondence information; (d): combined depth map of (b) and (c) in MRF

Table 4.2: Parameter configurations of refocusing method measures

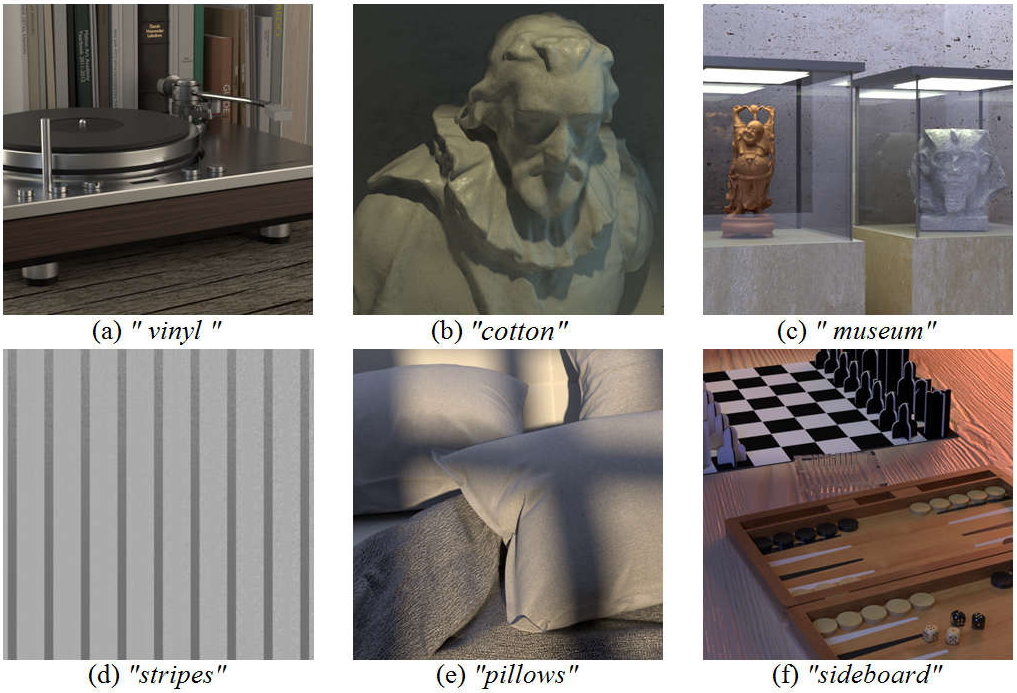
Sign	Value	Remark
n	256	Amount of depth levels
α	0.2 to 2	Range of refocusing parameter α
r_D	9	Radius of defocus measure window W_D in Eqn. 3.7
r_C	9	Radius of correspondence analysis window W_C in Eqn. 3.9
$[\lambda_1^{init}, \lambda_2^{init}]$	[1,1]	Weight controller in Eqn. 3.19
λ_{smooth}	2	Overall smoothness controller in Eqn. 3.20
λ_{flat}	2	Flatness controller in Eqn. 3.20
$\epsilon_{softness}$	1.0	Iteration softening factor in Eqn. 3.33
$\tau_{converge}$	1	Converge fraction for global minimization in MRF

The DER method process is illustrated with the LF image *pillows* (see its central view in figure 4.2(a)). Following the experimental parameter values listed in table 4.2, the DER method results in a final depth map (figure 4.2(d)). The final depth map is a combination of the depth map from the defocus information (figure 4.2(b)) and the one from the correspondence information (figure 4.2(c)). Each depth map is on the left side of a depth scale (a color bar), to which the depth values ranging from 0 to 255 in the depth map can be determined by reference. Also, the smaller the depth value is, the closer the image is to

the camera.

4.2 Analysis and Comparisons of Algorithm Runtime

The practical runtimes of the methods are obtained from experiments. Six LF images are picked out to test the runtimes of both the DEL method and the DER method, which are *vinyl*, *cotton*, *museum*, *stripes*, *pillows,* and *boardgames* from the Heidelberg Collaboratory LF image datasets. Those six LF images have same image size with $u = v = 9$ and $x = y = 512$, and their central sub-images are shown in figure 4.3.



(a) *" vinyl "* (b) *"cotton"* (c) *" museum"*

(d) *"stripes"* (e) *"pillows"* (f) *"sideboard"*

Figure 4.3: Central views belonging to the LF images for runtime tests

Table 4.3: Experimental runtimes in DEL method

LF image name	* Valid depth label	Process Runtime (sec)				
		Cost volume construction	Cost aggregation	Global optimization	Iterative refinement	Total
vinyl	51	$0.29 * 10^4$	$0.42 * 10^3$	$0.30 * 10^3$	$0.55 * 10^3$	$0.42 * 10^4$
cotton	75	$0.28 * 10^4$	$0.46 * 10^3$	$0.30 * 10^3$	$0.83 * 10^3$	$0.44 * 10^4$
museum	75	$0.30 * 10^4$	$0.42 * 10^3$	$0.29 * 10^3$	$0.78 * 10^3$	$0.45 * 10^4$
stripes	25	$0.27 * 10^4$	$0.38 * 10^3$	$0.14 * 10^3$	$0.27 * 10^3$	$0.35 * 10^4$
pillows	75	$0.30 * 10^4$	$0.42 * 10^3$	$0.27 * 10^3$	$0.70 * 10^3$	$0.44 * 10^4$
boardgames	73	$0.27 * 10^4$	$0.38 * 10^3$	$0.24 * 10^3$	$0.69 * 10^3$	$0.41 * 10^4$
Average	/	$0.29 * 10^4$	$0.40 * 10^3$	$0.26 * 10^3$	$0.64 * 10^3$	$0.40 * 10^4$
* Valid label: the label at whose plane has at least one scene point intersecting						

A valid depth label is an element of the pre-defined depth label set and there is at least

one scene point imaged of the LF image intersect the depth plane corresponding to this label. The numbers of valid depth labels are presented with the DEL method runtimes. It can be seen that the runtimes of two DEL method steps, the global optimization via graph cut and the iterative refinement, increase with the growing number of valid depth label.

Table 4.3 shows the runtimes of the four main steps of the DEL algorithm. For the six chosen LF images, the most time-consuming step to the lowest one are the cost volume construction, the iterative refinement, the cost aggregation, and the global optimization.

Table 4.4: Experimental runtimes in DER method

LF Image Name	Process Runtime (sec)			
	Depth Estimation Model Constructions	Depth Selection and Confidence Estimation	Global Optimization in MRF (number of iteration)	Total
vinyl	$1.67 * 10^3$	$1.34 * 10^2$	$0.63 * 10^2(2)$	$1.86 * 10^3$
cotton	$1.73 * 10^3$	$1.38 * 10^2$	$0.61 * 10^2(2)$	$1.93 * 10^3$
museum	$1.44 * 10^3$	$1.25 * 10^2$	$0.24 * 10^2(0)$	$1.59 * 10^3$
stripes	$1.59 * 10^3$	$1.23 * 10^2$	$0.25 * 10^2(0)$	$1.74 * 10^3$
pillows	$1.47 * 10^3$	$1.22 * 10^2$	$0.54 * 10^2(2)$	$1.64 * 10^3$
boardgames	$1.44 * 10^3$	$1.22 * 10^2$	$0.54 * 10^2(2)$	$1.62 * 10^3$
Average	$1.56 * 10^3$	$1.28 * 10^2$	$0.47 * 10^2(1)$	$1.73 * 10^3$

Three main steps constitute the DER method, which are the depth estimation model constructions, the depth selection, and confidence estimation, and the global optimization in MRF. For the same LF images used to test the DEL method, the experimental runtimes of the DER method are shown in table 4.4. As the number of iterations increase, the higher the runtime of the global optimization in MRF is, the number of iterations are shown in the brackets after the runtimes of the global optimization in MRF.

From table 4.4, it can be seen that the step for constructing the depth estimation model takes the majority of time for the DER method, while the steps, depth selection and confidence estimation, require less time and the runtime of the MRF global optimization is the least time-consuming step among all three steps.

Table 4.5: Total runtimes of the DER method and the DEL method

LF Image Name	Process Runtime (sec)	
	DER Method	DEL Method
vinyl	$0.19 * 10^4$	$0.42 * 10^4$
cotton	$0.1.9 * 10^4$	$0.44 * 10^4$
museum	$0.16 * 10^4$	$0.45 * 10^4$
stripes	$0.17 * 10^4$	$0.35 * 10^4$
pillows	$0.16 * 10^4$	$0.44 * 10^4$
boardgames	$0.16 * 10^4$	$0.41 * 10^4$
Average	$0.17 * 10^4$	$0.40 * 10^4$

Contrasting table 4.3 and table 4.4, both the DEL method and the DER method spent the majority of the time in LF image shifting (LF image shifts in the DEL method and LF image refocusing process in the DER method). Further, collecting the total runtime for each LF image from table 4.3 and 4.4 in table 4.5, it can be seen from table 4.5 that the average runtime of the DEL method ($0.40 * 10^4$) is much higher than the time of the DER method ($0.17 * 10^4$).

4.3 Analysis and Comparisons of Estimated Depth Maps

The depth map quality is analyzed and compared only from visual observations. This is because that the DER method fails to clearly label image depths (Jeon et al., 2016), and from our experiments, the disparity map outputs produced by the DEL method also have a large number of disparity values which are different than the ground truth maps of the offered benchmarks. As mentioned before, due to the lack of some necessary parameters depth-to-disparity conversion, the outputs from the DEL method and the DER method cannot be compared directly. Instead, they are compared to corresponding ground truth disparity maps and depth maps provided by the Heidelberg Collaboratory. For easy comparisons, the disparity maps and depth maps are presented as grey-scale images. In a grey-scale image, the higher the grey value of a pixel is, the whiter it is shown. The white color corresponds to the shortest depth in a disparity map and the longest depth in a depth map.

The qualitative tests are classified based on four aspects. The first aspect checks some challenges in stereo matching of the LF images from Heidelberg Collaboratory datasets. Those challenges include the irradiance deformation caused by transparency and reflection and the effects of low texture. Secondly, the quality of the depth maps is analyzed based on visual perception by considering depth smoothness, depth continuity, and edge preservation. Finally, the possibility of the loss of local depth information in the DER method is demonstrated.

4.3.1 Influences of Low Texture, Transparency, and Reflection on Depth Estimation

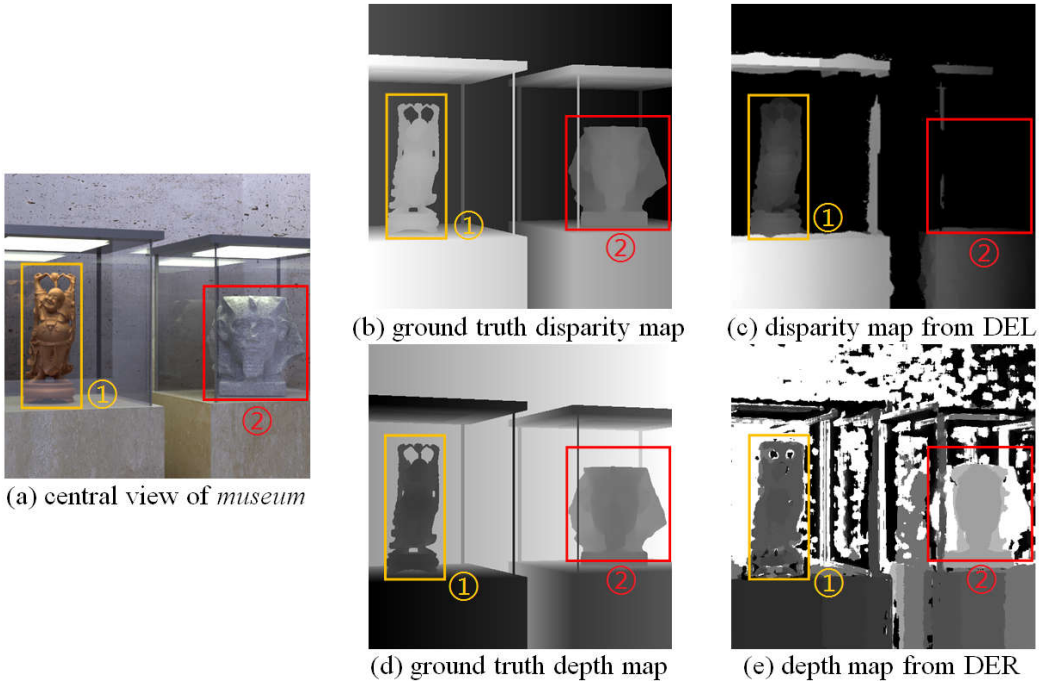
In this section for each experiments the results are evaluated using five images corresponding to a used LF image. They correspond to the central sub-view of the LF image, the ground truth disparity map, the estimated disparity map from DEL method, the ground truth depth map, and the depth map from DER method. If it is needed, the sub depth maps (from the defocus information and the correspondence information) of the DER method will be observed to check which sub map makes the most contribution to the performance of the combined depth map.

- **Transparency**

It is known that the depth estimations in both the DEL method and the DER method are related to irradiance, hence almost any distortions to the original irradiance will affect the results from both methods. Transparency deforms the irradiance emitted on the scene surface. If a transparent plane is placed between an object and a camera, the irradiance rays reflected from the object will be weakened when passing through the transparent plane.

The transparency effects are observed from two marked regions in the LF image *museum*

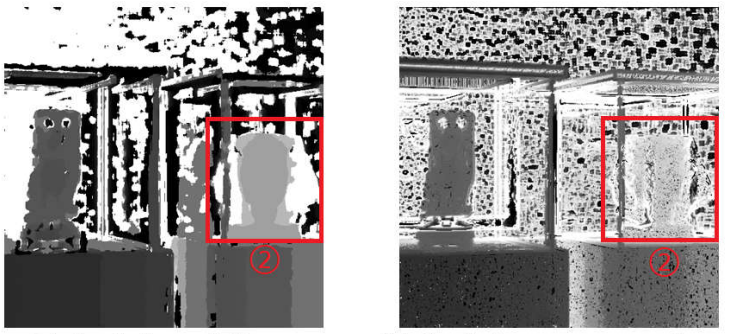
(see figure 4.4). Transparency appears within region 1 and region 2. The differences are that the color contrast and the depth of the object in region 1 behind the transparent plane are higher and lower than those in region 2.



(a) central view of *museum*
(b) ground truth disparity map
(c) disparity map from DEL
(d) ground truth depth map
(e) depth map from DER

Figure 4.4: Qualitative comparison of effect of transparency

Compared to the ground truth maps (figure 4.4(b) and figure 4.4(d)), both the DEL method and the DER method output more-or-less correct depth in region 1, but the DEL method lost the depth information of the Sphinx head statue in region 2 while the DER method keeps them (see figure 4.4(c) and figure 4.4(e)).



(a) depth from defocus
(b) depth from correspondence

Figure 4.5: Sub depth maps of LF image *museum* from defocus information and correspondence information in the DER method

Further checking the sub depth maps from the DER method before the combination of them in MRF (figure 4.5), the visually correct depths in region 2 is mainly from the sub

depth map derived from defocus information (figure 4.5(a)). It shows that the depth from defocus measure may lessen the effect of transparency on depth estimations.

- **Reflection**

Reflection also affects irradiance. For example, when a mirror surface reflects mirror images to a camera, the camera senses the mirror surface at a depth of the mirror image instead of the true depth of the mirror surface. Also, the brightness of a surface with specular reflection is higher than the brightness of the surface without specularity.

In figure 4.6, two regions in the central view of the LF image *vinyl* have been highlighted, where specularity and mirror reflection appear in the regions with label 1 and label 2, respectively.

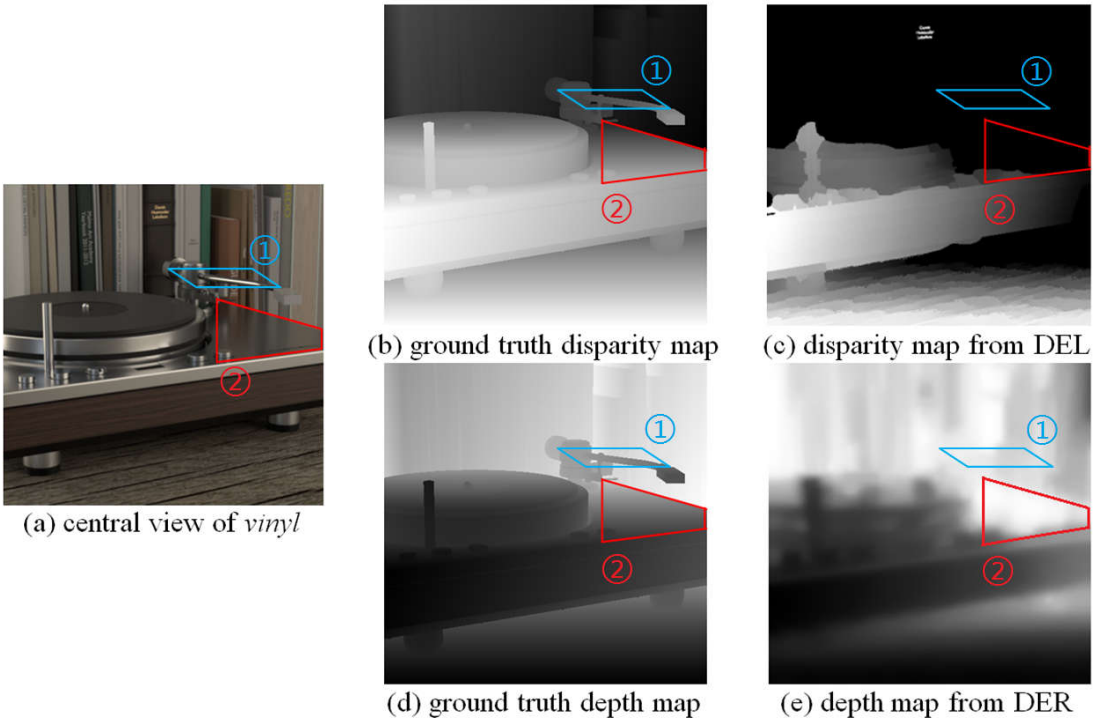


Figure 4.6: Qualitative comparison of effect of reflection

From figures 4.6(c) and 4.6(e), it can be seen that both methods output incorrect depth in regions 1 and 2.

- **Low Texture**

The depth estimation model in both the DEL and the DER method are based on the match of irradiance of pixels. Hence, the similar image patterns inside low texture areas confuse the estimate of the desired depth. Two LF images, *stripes* and *pyramids*, are used to test the effects of low texture (see figures 4.7 and 4.8).

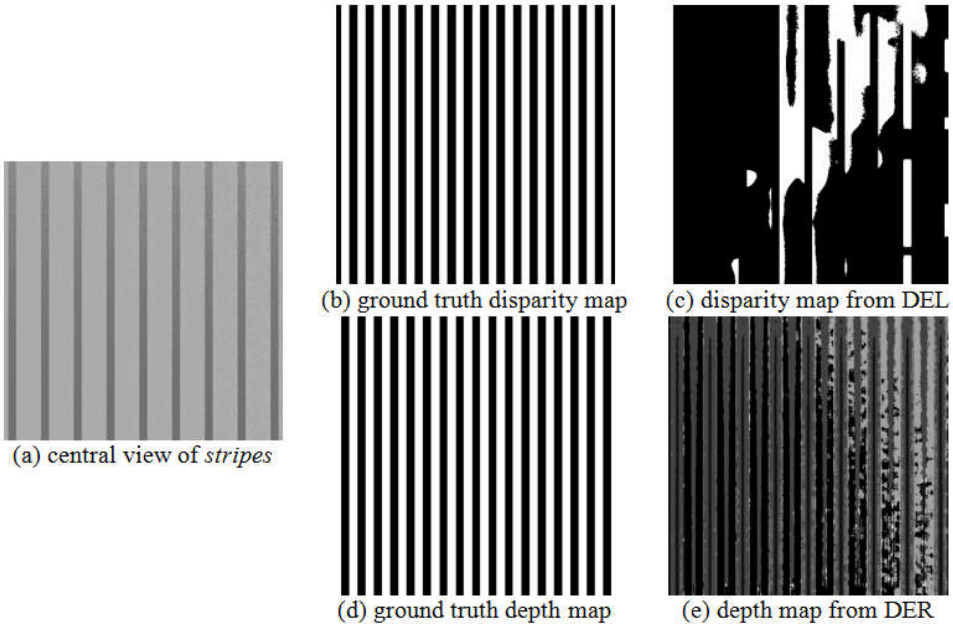


Figure 4.7: 1st qualitative comparison of effect of low texture

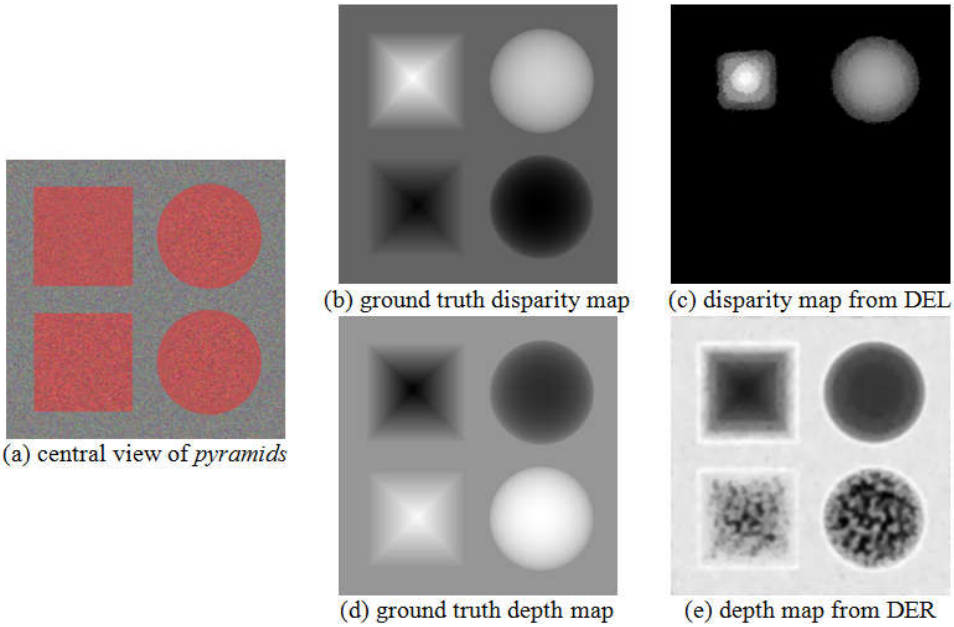
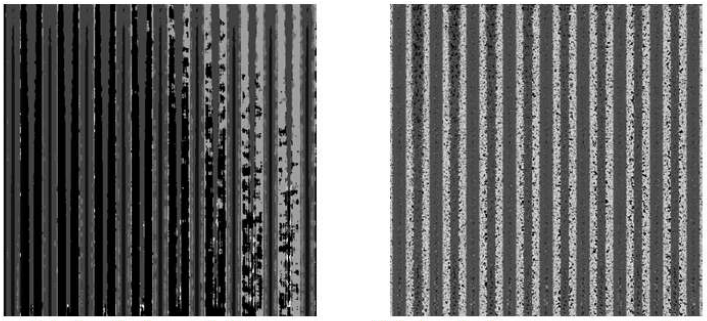


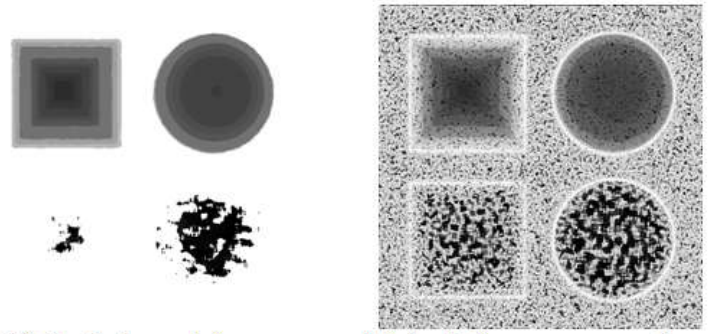
Figure 4.8: 2nd qualitative comparison of effect of low texture

From figures 4.7(a) and 4.8(a), it can be seen that the entire images can be treated as regions of low texture. The disparity maps (figures 4.7(c) and 4.8(c)) and the depth maps (figures 4.7(e) and 4.8(e)) estimated by both methods are not strictly correct when compared to ground truth disparity maps (figures 4.7(b) and 4.8(b)) and depth maps (figures 4.7(d) and 4.8(d)). But, the depth maps from the DER method tend to lead to more accurate depth information, such as the left side of *stripes*, and the bottom square and the bottom round of *pyramids*.



(a) depth from defocus (b) depth from correspondence

Figure 4.9: Sub depth maps of the LF image *stripes* from defocus information and correspondence information in the DER method



(a) depth from defocus (b) depth from correspondence

Figure 4.10: Sub depth maps of the LF image *pyramids* from defocus information and correspondence information in the DER method

Further, the depth from correspondence information in the DER method offers the majority of the more depth information, as can be observed from figures 4.9(b) and 4.10(b).

4.3.2 Qualitative Analysis and Comparisons of Depth Maps from Overall Perception

The analysis and comparisons in this section focus on the quality of the full depth maps. The quality is discussed based on three aspects: depth continuity, depth smoothness, and edge preservation.

- **Depth Continuity**

The image regions for the comparison of depth continuity are highlighted with red rectangles in figure 4.11.

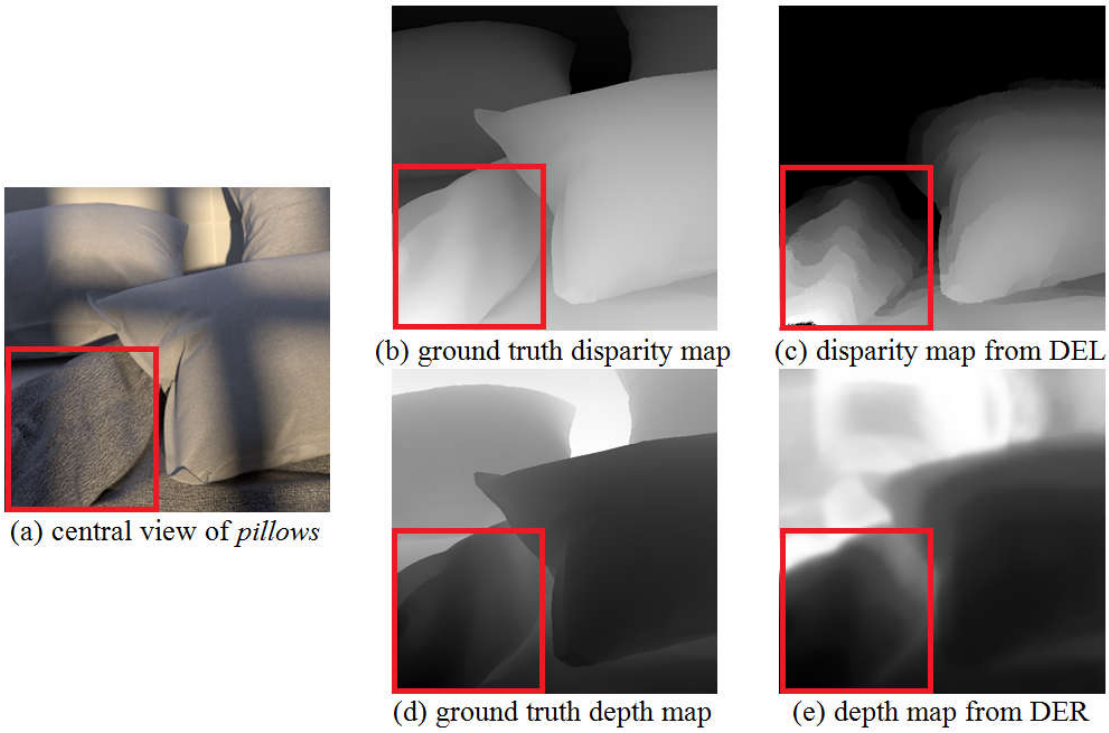


Figure 4.11: Qualitative comparison of depth continuity

Even though an iterative refinement is used to obtain non-discrete disparity maps, distinct discrete depths still can be seen in the disparity map from the DEL method (figure 4.11(c)). By comparison, the depth map from the DER method (figure 4.11(e)) presents continuous depths.

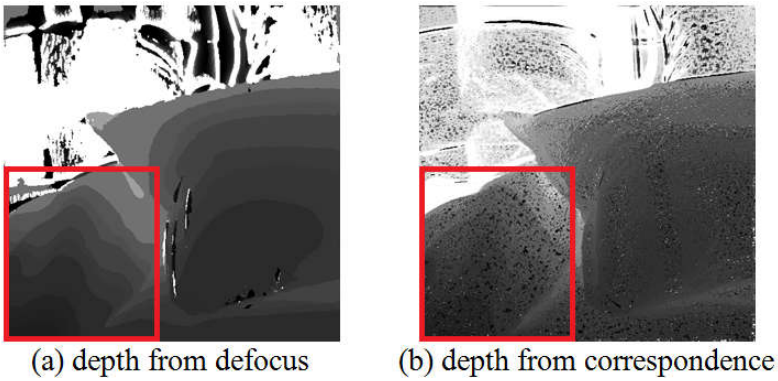


Figure 4.12: Sub depth maps of the LF image *pillows* from defocus information and correspondence information in the DER method

This depth continuity is due to the depth map combination in the DER method since only the depths estimated from the correspondence information are continuous (see figure 4.12).

• **Depth Smoothness**

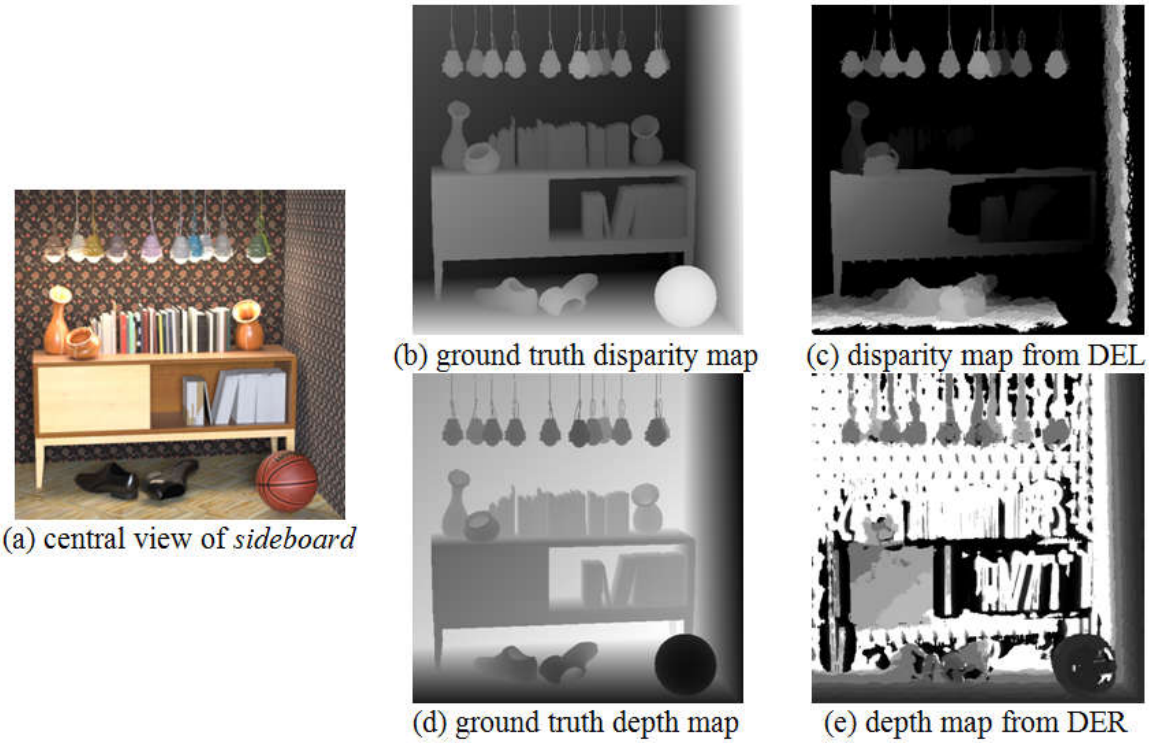


Figure 4.13: Qualitative comparison of depth smoothness

Using the LF image *sideboard* (figure 4.13) to observe depth smoothness, the depth map from the DER method (figure 4.13(e)) is unsmooth particularly the depths in the image background.

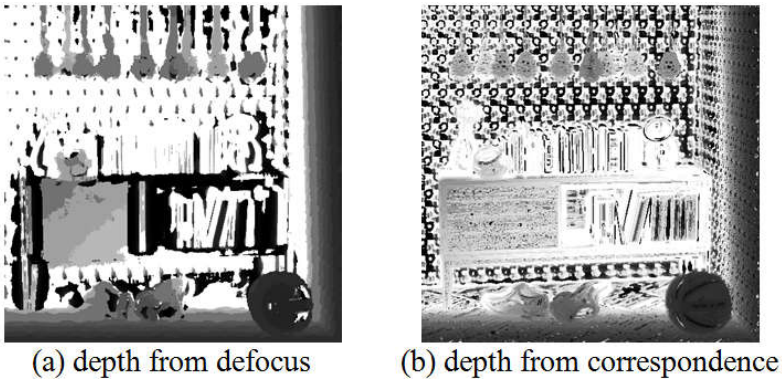


Figure 4.14: Sub depth maps of the LF image *sideboard* from defocus information and correspondence information in the DER method

From figure 4.14, both the depth form defocus and the depth from correspondence result in unsmooth depth maps. This is due to the fact that the unsmooth depth regions exceed the size of the window for smoothing the combined depth map in MRF. This results in

the combined depth map being unsmooth as well. In contrast, the disparity map from the DEL method (figure 4.13(c)) is much smoother which is due to the contribution of the graph-cut optimization (described in section 2.2.3).

- **Edge Preservation**

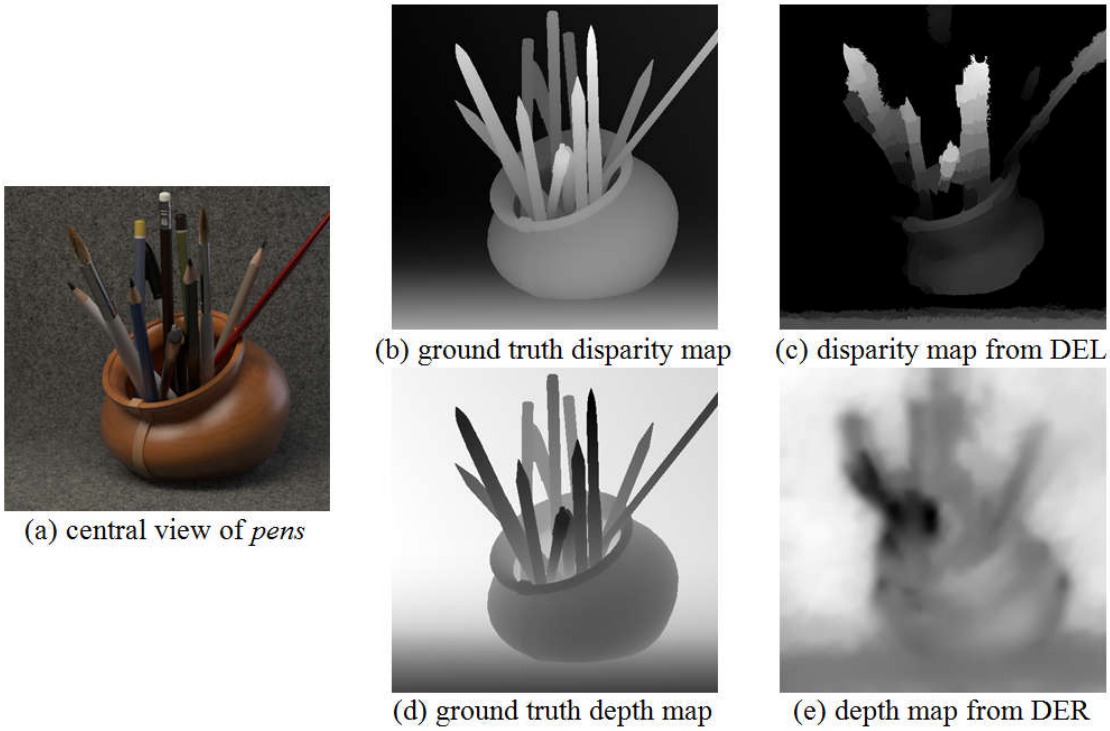

(a) central view of *pens*
(b) ground truth disparity map
(c) disparity map from DEL
(d) ground truth depth map
(e) depth map from DER

Figure 4.15: Qualitative comparison of edge preservation

Using the LF image *pens* as input, it is obvious in figure 4.15 that the disparity map from the DEL method (figure 4.15(c)) preserves clear edges of the image, while the edges are blurred in the depth map estimated from the DER method (figure 4.15(e)). The good performance of edge preservation in the DEL method is from the employed guided filter (see section 2.2.2) and the graph-cut optimization (see section 2.2.3).

4.3.3 Possible Loss of Local Depth Information Due to the Applied Global Optimization in the DER Method

To demonstrate the possible local depth inaccuracy in the DER method, five images belonging to a LF image are shown together. They are the central view of the LF image used, the given ground truth depth map, the final depth map from the DER method, the depth map from defocus, and the depth map from correspondence. The LF images *boardgames* and *kitchen* are used to show the experimental results.

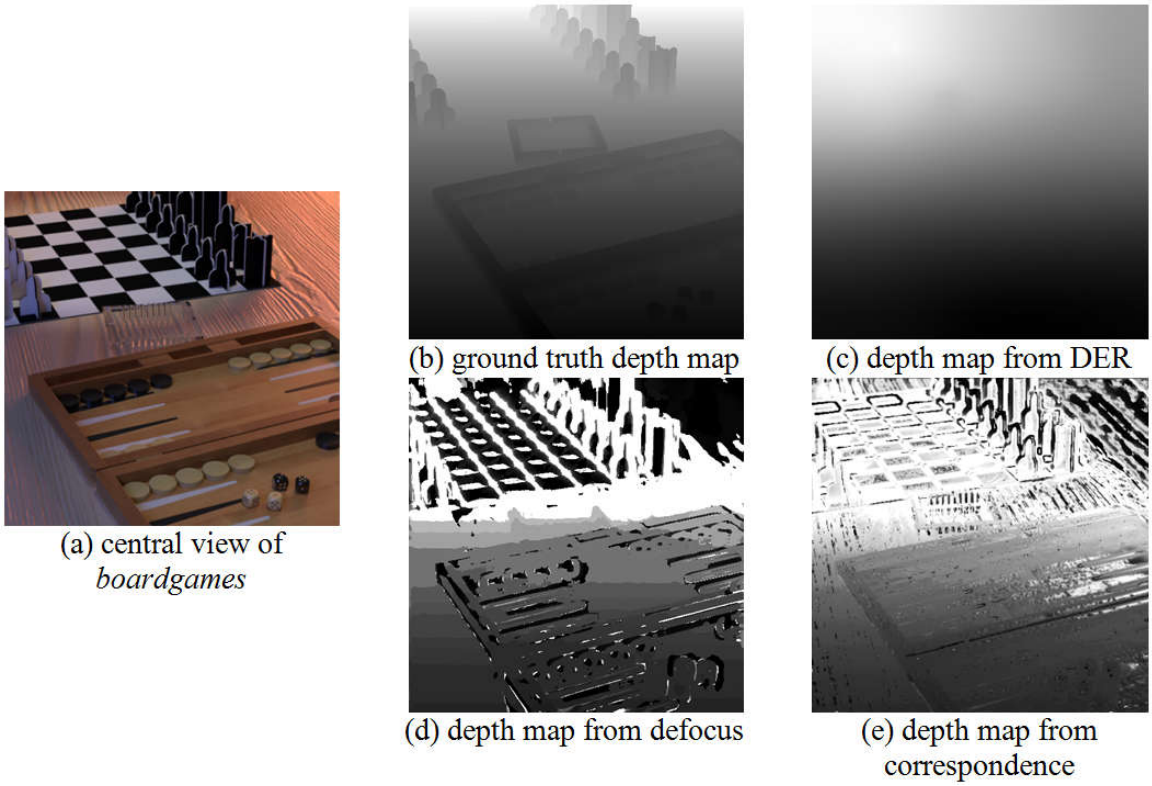


Figure 4.16: 1st demonstration of the loss of local depth information in the DER method

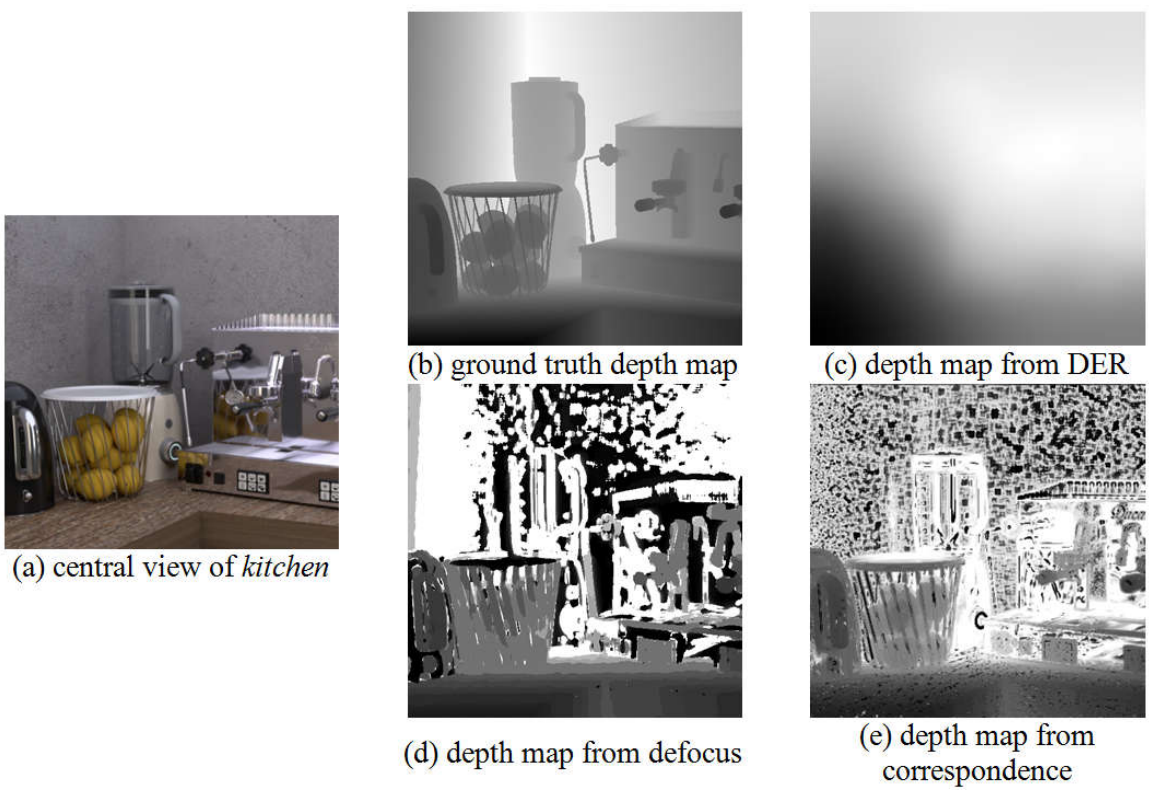


Figure 4.17: 2nd demonstration of the loss of local depth information in the DER method

From these experiments, it can be seen that the estimated depth maps from the DER method may be excessively smooth due to the loss of detailed depths of the objects in the images. This can be seen in figure 4.16, when comparing the depth maps from the DER method (figures 4.16(c) and 4.17(c)) to corresponding ground truth depth maps (figures 4.16(b) and 4.17(b)). However, this does not occur in the depth maps from the defocus information (figures 4.16(d) and 4.17(d)) and the depth map from correspondence information (figures 4.16(e) and 4.17(e)). Therefore, this issue is most likely caused by the MRF global optimization.

4.3.4 Depth Map Quality Discussion and Conclusion

Table 4.6: Comparisons of Depth Map Quality

Compared Item	DEL Method	DER Method
Transparency tolerance	Low	High
Reflection tolerance	Low	Low
Texture tolerance	Low	Low, but higher than the DEL method
Depth continuity	Non-discrete but approximately discrete	Continuous
Depth smoothness	Smooth in both local and global sense	Smooth in local sense but not always in global
Edge preservation	Clear	Ambiguous
Possibility of losing local depth information	No	Yes

Table 4. 5 summarizes the quality comparisons presented in the previous sections.

In table 4.5, the performance of both methods for the cases of low texture, transparency, and reflection is summarized. These three cases represent tough challenges in stereo matching (Farinella, Battiato and Cipolla, 2013), while the stereo matching is the theoretical basis of the DEL method. Therefore, it is not a surprise that the DEL method has low tolerances of transparency, reflection, and low texture.

Further, it can be seen from table 4.5 that the DER method performs better than the DEL method in cases of transparency, low texture, and depth discontinuity. However, the DER method introduces three new issues that are nonexistent in the DEL method. The first one is the ambiguous edges. The ambiguous edges could be solved by using a guided filter or a similar filter and improving the global optimization to improve edge preservation. The second and the third issues are the unsmooth depth map in a global sense and the possibility of local depth information loss. They severely decrease the quality of the depth maps and improvements are required such as applying a more advanced global optimization method. Note that the possibility of local depth information loss found in this report confirms a comparative result by Jeon et al. (2016), which is said the DER method failed to clearly label the depth.

Chapter 5. Conclusion and Future Works

5.1 Conclusion

Two representative methods, the DEL method and the DER method, were used to study depth estimation from LF images. The DEL method is developed based on previous work on the depth estimations from stereo images obtained using conventional cameras. The DER method, on the other hand, takes advantage of LF image properties to estimate depths.

In chapter 1, the preliminaries about Light Fields (LFs), LF cameras, and depth estimation are presented and a brief review of the depth estimation from LF images is given. In the same chapter, the objective and the motivation for this project as well as the outline of the report are also presented.

The DEL method is presented in chapter 2. The theoretical basis and the main steps of the algorithm are discussed. Further, the working of the algorithm is described.

In Chapter 3, the DER method used for depth estimation from LF images is presented. Together with the DEL described in chapter 2, the performance of these two methods will be analyzed and compared in chapter 4.

In chapter 4, these two methods were applied to a set of LF images with 9 by 9 sub-images from the Heidelberg Collaboratory 4D Image Datasets and their performance was compared with respect to computation time and the visual quality of the resulting depth and disparity maps. Experimental results from the implementation of these two methods show that the computational time required to generate depth maps using the DER method is less that the computation time required by the DEL method. The time advantage of the DER method will become larger when dealing with larger LF images with higher resolution sub-images than the chosen ones. Regarding image quality, the estimated depth maps from the DEL method appear to perform better with respect to edge preservation and depth smoothness in the global sense, while the depth maps from the DER method seem to perform better in handling transparent images, low texture images, and depth continuity.

It should be noted that using the DER method the resulting maps appeared to be unsmooth in the global sense and there is a possibility of losing local depth information. This indicates that the results of this method may be inferior with respect to quality. Therefore, from experiments performed in this report, the DEL method appears, comparatively speaking, to lead to more accurate and robust depth maps than the DER method.

5.2 Future Works

The work done in this project can be extended in the following four areas:

Firstly, to further evaluate those two algorithms, more experiments are needed. In this report only some of the LF images available in the dataset were used and therefore, not all known challenges in depth estimation have been tested. Therefore, several experiments can be done to test the evaluate the performance of these methods, including to using LF images from commercial LF cameras

Secondly, in this paper, the depth map quality was analyzed and compared based on visual observations. Conclusions from only visual observations are not satisfactory and it is desirable to complement them with other, more objective criteria. Additional methodologies analyzing the depth map quality could be, for example, be used based on the evaluation of the mean square depth error or other criteria used in the literature.

Thirdly, an updated version of the DER method should be studied as a continuation of this paper. This updated method, published by Tao et al. (2015), is entitled "Depth from shading, defocus, and correspondence using light-field angular coherence", which is targeted at amending some shortcomings of the DER method. Therefore, it is worthy to check whether the updated DER method can overcome the issues of the DER method observed in this project.

Fourthly, it would be interest to develop a method based on a combination of the DEL method and the DER method. Since the performance of the DEL and the DER method tend to be complementary, i.e. one performs better where the other fails, a combined method may be able to retain the advantages of both and eliminate the disadvantages.

Reference

Adelson, E. and Bergen, J. (1991). *The Plenoptic Function and the Elements of Early Vision*, Computational Models of Visual Processing, pp. 3–20. Available at: http://persci.mit.edu/pub_pdfs/elements91.pdf [Accessed 10 May 2017].

Adelson, E. and Wang, J. (1992). Single lens stereo with a plenoptic camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2), pp.99-106.

Bishop, T. and Favaro, P. (2011). Full-Resolution Depth Map Estimation from an Aliased Plenoptic Light Field. *Computer Vision – ACCV 2010*, pp.186-200.

Bishop, T. and Favaro, P. (2012). The Light Field Camera: Extended Depth of Field, Aliasing, and Superresolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5), pp.972-986.

Bolles, R., Baker, H. and Marimont, D. (1987). Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1), pp.7-55.

Boykov, Y., Veksler, O. and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), pp.1222-1239.

Bracewell R. (2004). *Fourier Analysis and Imaging*. 2003 Edition. Springer, pp.155-157.

Calderon, F., Parra, C. and Nino, C. (2014). Depth map estimation in light fields using an stereo-like taxonomy. *2014 XIX Symposium on Image, Signal Processing and Artificial Vision*.

Chen, Q., Zhang, Y., Cao, X., Zhang, Y. and Xiong, H. (2016). Depth map estimation with 4D light fields using confocal stereo. *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*.

Chen, Y., Yi-Chin Wu, Chih-Hung Liu, Wei-Chih Sun and Chen, Y. (2010). Depth map generation based on depth from focus. *2010 International Conference on Electronic Devices, Systems and Applications*.

Chutjian, A. and Collier, R. (1968). Recording and Reconstructing Three-Dimensional Images of Computer-Generated Subjects by Lippmann Integral Photography. Applied Optics, 7(1), p.99.

Dansereau, D. and Bruton, L. Gradient-based depth estimation from 4D light fields. *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No.04CH37512)*.

Delong, A., Osokin, A., Isack, H. and Boykov, Y. (2011). Fast Approximate Energy Minimization with Label Costs. *International Journal of Computer Vision*, 96(1), pp.1-27.

Dongarra, J. and Sullivan, F. (2000). Guest Editors Introduction to the top 10 algorithms. *Computing in Science & Engineering*, 2(1), pp.22-23.

Draper, N. and Smith, H. (1998). *Applied regression analysis*. 3rd ed. New York.

Drineas, P., Mahoney, M., Muthukrishnan, S. and Sarlós, T. (2010). Faster least square approximation. *Numerische Mathematik*, 117(2), pp.219-249.

Fang, J., Varbanescu, A., Shen, J., Sips, H., Saygili, G. and Van Der Maaten, L. (2012). Accelerating Cost Aggregation for Real-Time Stereo Matching. *2012 IEEE 18th International Conference on Parallel and Distributed Systems*.

Farinella, G., Battiato, S. and Cipolla, R. (2013). *Advanced Topics in Computer Vision*. 1st ed. Springer London, pp.143-179.

Fife, K., El Gamal, A. and Wong, H. (2008). A 3MPixel Multi-Aperture Image Sensor with 0.7µm Pixels in 0.11µm CMOS. *2008 IEEE International Solid-State Circuits Conference - Digest of Technical Papers*. pp. 48–50.

Georgeiv, T. and Intwala, C. (2003) *Light Field Camera Design for Integral View Photography*. Available at: http://tgeorgiev.net/IntegralView.pdf [Accessed 10 May 2017].

Georgiev, T., Yu, Z., Lumsdaine, A. and Goma, S. (2013). Lytro camera technology: theory, algorithms, performance analysis. *Multimedia Content and Mobile Devices*.

Gortler, S., Grzeszczuk, R., Szeliski, R. and Cohen, M. (1996). The lumigraph. *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*. Available at: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/Gortler-SG96.pdf [Accessed 10 May 2017].

He, K., Sun, J. and Tang, X. (2013). Guided Image Filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6), pp.1397-1409.

Honauer, K., Johannsen, O., Kondermann, D. and Goldluecke, B. (2016). *A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields*. Available at: http://lightfield-analysis.net/benchmark/paper/lightfield_benchmark_accv_2016.pdf [Accessed 10 May 2017].

Hosni, A., Rhemann, C., Bleyer, M., Rother, C. and Gelautz, M. (2013). Fast Cost-Volume Filtering for Visual Correspondence and Beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(2), pp.504-511.

Huijin Lv, Kaiyu Gu, Yongbing Zhang and Qionghai Dai (2015). Light field depth estimation exploiting linear structure in EPI. *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*.

Ives, F. (1903). *Parallax stereogram and process of making same*. Available at: https://patentimages.storage.googleapis.com/pdfs/US725567.pdf [Accessed 10 May 2017].

Jeon, H., Park, J., Choe, G., Park, J., Bok, Y., Tai, Y. and Kweon, I. (2015). Accurate depth map estimation from a lenslet light field camera. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kolmogorov, V. and Zabih, R. (2002). Multi-camera Scene Reconstruction via Graph Cuts. *Computer Vision — ECCV 2002*, pp.82-96.

Kuang-Chih Lee, Ho, J. and Kriegman, D. (2005). Acquiring linear subspaces for face recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5), pp.684-698.

Lumsdaine, A. and Georgiev, T. (2008) *Full Resolution Lightfield Rendering*. Available at: http://tgeorgiev.net/FullResolution.pdf [Accessed 10 May 2017].

Ng, R., Levoy, M., Bredif, M., Duval, G., Horowitz, M. and Hanrahan, P. (2005). *Light Field Photography with a Hand-held Plenoptic Camera*. Available at: https://graphics.stanford.edu/papers/lfcamera/lfcamera-150dpi.pdf [Accessed 10 May 2017].

Paris, S., Kornprobst, P., Tumblin, J. and Durand, F. (2008). Bilateral Filtering: Theory and Applications. *Foundations and Trends® in Computer Graphics and Vision*, 4(1), pp.1-75.

Perwass, C. and Wietzke, L. (2012). Single lens 3D-camera with extended depth-of-field. *Human Vision and Electronic Imaging XVII*.

Roberts, D. and Smith, T. (2014). *The History of Integral Print Methods An excerpt from: "Lens Array Print Techniques"*. Available at: http://lenticulartechnology.com/files/2014/02/Integral-History.pdf [Accessed 10 May 2017].

Seitz, S. and Dyer, C. (1996) *View Morphing*, Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH' 96. pp. 21–30. Available at: http://homes.cs.washington.edu/~seitz/papers/sigg96.pdf [Accessed 10 May 2017].

Tao, M., Hadap, S., Malik, J. and Ramamoorthi, R. (2013). Depth from Combining Defocus and Correspondence Using Light-Field Cameras. *2013 IEEE International Conference on Computer Vision*.

Tao, M., Srinivasan, P., Malik, J., Rusinkiewicz, S. and Ramamoorthi, R. (2015). Depth from shading, defocus, and correspondence using light-field angular coherence. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Wang, Y., Ostermann, J. and Zhang, Y. (2007). *Video processing and communications*. 1st ed. Taipei: Pearson Education Taiwan, pp.394-409.

Wanner, S., Meister, S. and Goldluecke, B. (2013). Datasets and Benchmarks for Densely Sampled 4D Light Fields. *Vision, Modeling, and Visualization*. Available at: https://pdfs.semanticscholar.org/1a86/e03c229adb5b94e1f43f8508f033f74e94ae.pdf [Accessed 10 May 2017].

Xu, Y., Jin, X. and Dai, Q. (2015). Depth estimation by analyzing intensity distribution for light-field cameras. *2015 IEEE International Conference on Image Processing (ICIP)*.

Yang, Q., Yang, R., Davis, J. and Nister, D. (2007). Spatial-Depth Super Resolution for Range Images. *2007 IEEE Conference on Computer Vision and Pattern Recognition*.