


8-2016

Three dimensional moving pictures with a single imager and microfluidic lens

Chao Liu
Purdue University

Follow this and additional works at: https://docs.lib.purdue.edu/open_access_dissertations

 Part of the [Computer Engineering Commons](#), and the [Electrical and Computer Engineering Commons](#)

Recommended Citation

Liu, Chao, "Three dimensional moving pictures with a single imager and microfluidic lens" (2016). *Open Access Dissertations*. 800.
https://docs.lib.purdue.edu/open_access_dissertations/800

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.

**PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance**

This is to certify that the thesis/dissertation prepared

By Chao Liu

Entitled

THREE DIMENSIONAL MOVING PICTURES WITH A SINGLE IMAGER AND MICROFLUIDIC LENS

For the degree of Doctor of Philosophy

Is approved by the final examining committee:

Charles A. Bouman

Co-chair

Lauren Chrstopher

Co-chair

Edward J. Delp

Paul Salama

To the best of my knowledge and as understood by the student in the Thesis/Dissertation Agreement, Publication Delay, and Certification Disclaimer (Graduate School Form 32), this thesis/dissertation adheres to the provisions of Purdue University's "Policy of Integrity in Research" and the use of copyright material.

Approved by Major Professor(s): Lauren Christopher, Co-chair

Approved by: Venkataramanan Balakrishnan

Head of the Departmental Graduate Program

07/27/2016

Date

THREE DIMENSIONAL MOVING PICTURES
WITH A SINGLE IMAGER AND MICROFLUIDIC LENS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Chao Liu

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

August 2016

Purdue University

West Lafayette, Indiana

ACKNOWLEDGMENTS

Foremost, special thanks to my advisor Dr. Lauren Christopher of the Department of Electrical and Computer Engineering for her support of my studies. I would also like to thank my committee members Dr. Bouman, Dr. Delp and Dr. Salama for their guidance.

TABLE OF CONTENTS

	Page
LIST OF TABLES	v
LIST OF FIGURES	vi
ABSTRACT	ix
1 Introduction	1
2 Depth Estimation Methods	4
2.1 Stereo imaging	4
2.2 Light field imaging	6
2.3 Depth from focus	6
2.4 Depth from defocus	8
3 Optics and DfD Method	10
3.1 Lens Systems and Defocus/Depth Relationships	10
3.2 Modelling Defocus: Point Spread Function	12
3.3 Depth of defocus field	13
3.4 Summary	16
4 New Research Improvements	18
4.1 Algorithm overview	18
4.2 Initial depth map generation	19
4.3 Preprocessing	21
4.4 Revised MAP Estimation using texture information	23
4.5 Summary	25
5 Regularized Depth from Defocus	26
5.1 MAP-MRF	26
5.2 EM/MPM	28
5.3 Graph-cuts	34

	Page
5.4 Summary	41
6 Real Lens/Camera simulation	42
6.1 Lens simulation	43
6.1.1 spherical aberration	43
6.1.2 Coma	47
6.1.3 Distortion	48
6.2 Simulate Camera digital image processing pipeline	52
6.2.1 Pipeline introduction	53
6.2.2 Noise	54
6.2.3 Illumination and Contrast ratio	60
6.2.4 Resolution effects	62
6.3 Summary	63
7 Experimental Results from Camera with Microfluidic Lens	64
7.1 Microfluidic lens	65
7.2 True camera results	68
8 Potential application	75
9 Summary and future work	77
LIST OF REFERENCES	79
VITA	83

LIST OF TABLES

Table	Page
5.1 Experimental results comparison, RMSE	34
5.2 Graph cut experimental results comparison, RMSE	40
6.1 Noisy inputs experimental results comparison, RMSE	59
7.1 Average running time for each starting picture (EM/MPM)	73
7.2 Average running time for frame to frame processing (EM/MPM)	73
7.3 Average running time for each starting picture (Graph-cut)	74
7.4 Average running time for frame to frame processing (Graph-cut)	74

LIST OF FIGURES

Figure	Page
2.1 Binocular Stereo Geometry [16]	5
2.2 depth from focus	7
3.1 Lens system with object at focused position	11
3.2 Lens system with object at defocused position	11
3.3 Illustration of Depth of defocus field	16
4.1 Proposed algorithm overview	19
4.2 Initial MAP Estimation	20
4.3 Preprocessing procedure	21
4.4 Example of input and output of preprocessing procedure	22
4.5 Example of the benefits of preprocessing procedure. (a) All in-focus image (b) Defocus image (c) Depth ground truth (d) Texture image (e) Initial depth map (f) Final depth map after using texture information	23
4.6 Revised MAP Estimation	24
5.1 General MAP Estimation block diagram	27
5.2 (a) In-focus image (b) Synthetic defocus image (c) Ground truth (d) Initial depth map (grayscale input) (e) depth map(color input) without texture information (f) final depth map	31
5.3 Middlebury results (a) In-focus image (b) Ground truth (c) EDfD results (d) SA results (e) Shape from defocus results (f) 3D view maps	32
5.4 Comparison with other methods on Middlebury image data	33
5.5 Graph Cut for segmentation example	35
5.6 Middlebury results 1 (a) In-focus image (b) EDfD (use EM/MPM) (c) EDfD (use Graph-Cut) (d) Ground truth	37
5.7 Middlebury results 2 (a) In-focus image (b) EDfD (use EM/MPM) (c) EDfD (use Graph-Cut) (d) Ground truth	38

Figure	Page
5.8 Middlebury results 3 (a) In-focus image (b) EDfD (use EM/MPM) (c) EDfD (use Graph-Cut) (d) Ground truth	39
5.9 Graph-cut results Comparison with other methods	40
6.1 Physical image formation process	42
6.2 Spherical aberration example (a) No aberration (b) Spherical aberration [34]	43
6.3 Spherical aberration	44
6.4 spherical aberration result (a) In-focus image (b) Defocus image (c) Depth map	45
6.5 spherical aberration result (PSF known) (a) In-focus image (b) Defocus image (c) Depth map result by using a known PSF (d) Depth map result (No aberration)	46
6.6 Coma illustration (a) No aberration (b) Coma aberration [34]	47
6.7 Coma aberration [35]	48
6.8 Distortion [36]	49
6.9 Barrel Distortion	50
6.10 Depth map result with Barrel Distortion (a) In-focus image (b) Defocus image (c) Depth map result(with Barrel Distortion) (d) Depth map result(without distortion)	51
6.11 Depth map result after correcting distortion (a) In-focus image after correcting barrel distortion. (b) Defocus image after correcting barrel distortion. (e) is the ground truth of depth map. (c) represents the resulting depth map with the correction-first method, and (d) is the depth map using EDfD-first.	52
6.12 Image signal processing pipeline (ISP)	54
6.13 Intensity dependent noise (a) In-focus image with intensity-dependent noise (b) Defocus image with intensity-dependent noise (c) In-focus image without noise (d) Defocus image without noise	56
6.14 EDfD example result with noise effect (a) EDfD result using noisy inputs (b) EDfD result using noise-free inputs (C) Ground truth	57
6.15 Middlebury EDfD result with noise effect	58
6.16 Teddy with noise effect, RMSE	59
6.17 EDfD example result under different Illumination	60

Figure	Page
6.18 RMSE of EDfD example results under different exposures	61
6.19 Resolution effects (a) In-focus image (original size) (b) Defocus image (original size) (c) EDfD result (use (a) and (b)) (d) In-focus image (half size) (e) Defocus image (original size) (f) EDfD result (use (d) and (e))	63
7.1 Single imager system	64
7.2 Effective focal length vs liquid lens voltage	66
7.3 optical power vs liquid lens voltage	67
7.4 optical power vs response time	68
7.5 Train and gift box (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image	69
7.6 Basket and Malaysia (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image	70
7.7 Dog and gift box (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image	71
7.8 Basket and train (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image	72
8.1 EDfD result of a skull (a) In-focus image (b) Defocus image (c) EDfD depth map	76

ABSTRACT

Liu, Chao Ph.D., Purdue University, August 2016. Three Dimensional Moving Pictures with a Single Imager and Microfluidic Lens. Major Professor: Lauren Christopher.

Three-dimensional movie acquisition and corresponding depth data is commonly generated from multiple cameras and multiple views. This technology has high cost and large size which are limitations for medical devices, military surveillance and current consumer products such as small camcorders and cell phone movie cameras. This research result shows that a single imager, equipped with a fast-focus microfluidic lens, produces a highly accurate depth map. On test material, the depth is found to be an average Root Mean Squared Error (RMSE) of 3.543 gray level steps (1.38%) accuracy compared to ranging data. The depth is inferred using a new Extended Depth from Defocus (EDfD), and defocus is achieved at movie speeds with a microfluidic lens. Camera non-uniformities from both lens and sensor pipeline are analysed. The findings of some lens effects can be compensated for, but noise has the detrimental effect. In addition, early indications show that real-time HDTV 3D movie frame rates are feasible.

1. INTRODUCTION

Depth inference is a key research area for modeling 3D objects in the 3D environment; for consumer electronics, robotics, and computer vision. In consumer electronics, depth maps are used in Depth Image Based Rendering (DIBR) displays, they are used as part of improved efficiency 3D compression algorithms, and can be used in future virtual reality.

Depth may be inferred using stereo disparity [1]; however this requires multiple source images where two cameras or complex optics are needed to achieve the left-right views. Depth also may be found by ranging techniques, but this requires additional transmit and receive hardware. New light-field or integral imaging cameras can produce depth [2], but the microlens array reduces the maximum imager resolution capability. None of the current 3D imaging systems is easily miniaturized to fit with the form factor of a small consumer camera, such as the type in cell phones and tablet devices. For medical devices such as endoscopes, the large size of the imaging system limits the applications. Military surveillance applications such as unmanned vehicles have limited space for cameras, and would benefit from 3D videos. The size and cost of the current systems includes two imagers and/or expensive lens arrays or ranging devices. Depth from defocus inference [3–5] requires only one imager capturing two focus images, which can be done with a standard camera with varying focus. Inferring depth is done by a pixel-by-pixel comparison of two or more defocused images, where the object’s blur radius is related to its distance. This depth inference uses Bayesian and Markov Random Field (MRF) statistical structure [6–8]. The published data are promising, but the classical approach can be improved by combination with other computational imaging techniques. The motivation of this research is to extend the classical DfD to Extended Depth from Defocus (EDfD) and using a fast focus optics to make a real-time system.

The new EDfD algorithm is using a new optimization function, extended to adapt to both the image color data and high frequency image data. This research shows significant depth accuracy improvements compared to the currently published DfD techniques. Depth is important in new consumer electronics products in order to create immersive 3D experiences for the user with new 3D displays. Accurate depth information is also needed for improved compression efficiency and for super-resolution techniques. A method for enhancing a ranging cameras resolution was reported in [9], which used Markov Random Field methods with the 2D image to provide a more accurate depth result for DIBR display. This reference uses a ranging camera in addition to the visible light imager. Another thread of research explores 2D to 3D conversion in two representative papers, the first uses edge information from the 2D image [10] to provide a depth map from a hypothesis depth map starting point; the second provides a depth map specifically for outdoor scenes using the dark channel (the effect of haze in the image) to estimate depth [11]. The results from EDfD show significant quality improvement compared to these two papers, and EDfD is generally applicable to a variety of scenes.

For the EDfD method, fast focus optics is required. New bio-inspired microfluidic lenses [12, 13] allow a time-domain approach for the very fast focus change. These new lenses use two fluids and electrostatic forces to rapidly change the shape of a very small lens. To design the total system then requires balancing the maximum focus speed of the microfluidic lens with the capability and accuracy of the depth inference.

Based on my previous research [14], this thesis presents a new extended DfD depth inference method, together with a fast focus lens which enables depth map generation of an average accuracy 3.543 RMSE compared to ground truth, and small size due to a single imager. The computational complexity is similar to other methods, with opportunity for further improvements. The results are shown for synthetic blur images for accuracy testing and for a single imager matched with microfluidic lens for generating the 2 focus images. Chapter 2 introduces different depth estimation methods including depth from defocus algorithm. Chapter 3 provides the theoretical

background for optics and DfD model. Chapter 4 describes the new improvements to the state of the art. Chapter 5 illustrates regularization methods for depth from defocus. Chapter 6 simulates the effects from lens and camera. Chapter 7 presents the experimental results, and finally Chapter 8 contains the goals and plans for future research.

2. DEPTH ESTIMATION METHODS

Depth acquisition methods can be broadly classified as optical and non-optical methods. Non-optical methods are based on technologies like Magnetic and Ultrasound. Working with Lasers, the non-optical methods could get accurate single point depth information, but they require very expensive computations to achieve a dense depth map. Optical methods usually could provide acceptable depth accuracy from images. Here are two kinds of Methods: Active Method and Passive Method. Active methods are the methods using controlled energy beams like structure light [4]. But they are constrained by the environment. Passive methods are more applicable without any environmental constraint and are widely employed in many areas [4]. The research in this thesis belongs to passive optical depth recovery which will be presented in the next Section.

Monocular and Binocular are two kinds of Optical depth estimation techniques. Binocular vision technologies, for example, Depths from Stereo imaging requires at least two images captured from different viewpoints. By comparing these images, the disparity between the images is related to the actual depth. Monocular techniques estimate depth by using only one single camera. Depth is determined by using the relative size of the objects, the distribution of light and shade, movement at a different distance, and the amount of focus or defocus. Monocular vision techniques include: Depth from Focus, Depth from Defocus and so on.

2.1 Stereo imaging

As described in [15], stereo imaging systems use two or more images which captured from different viewpoints as input to calculate depth. Every viewpoint is separated from others by some distance. By doing this way, the depth information can

be computed by the disparity information between these images. The typical stereo system capturing two images is shown in Figure 2.1

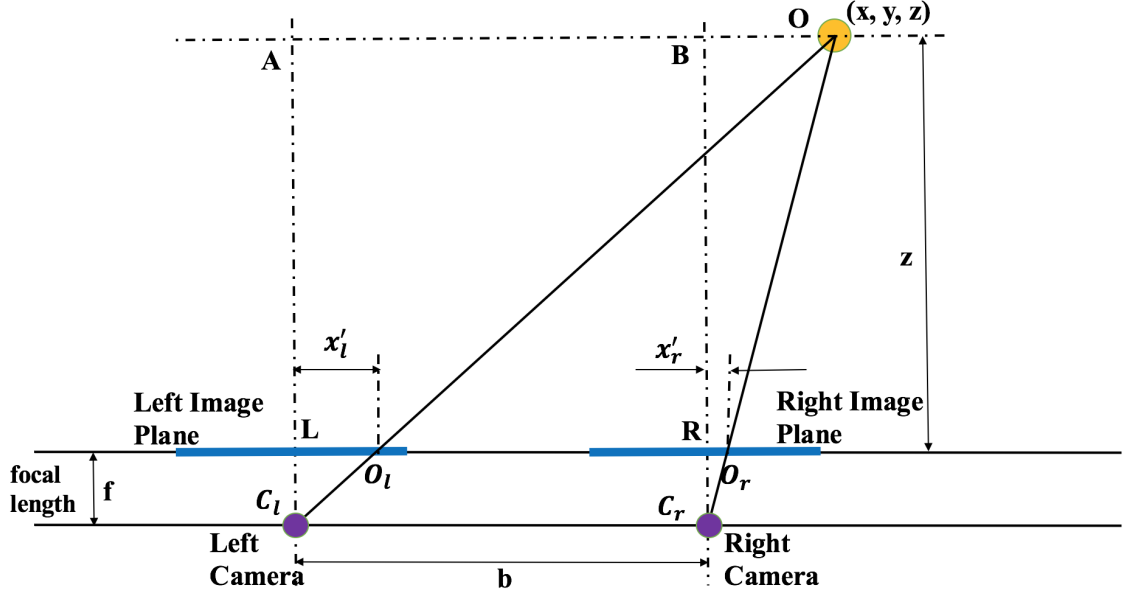


Fig. 2.1. Binocular Stereo Geometry [16]

As introduced in [16], Figure 2.1 shows a model of stereo imaging system. O_l and O_r are detected points of object O in the left and right image planes, respectively. By using geometry similarity property, as shown by equations 2.1 and 2.2:

$$\frac{x}{z} = \frac{x'_l}{f} \quad (2.1)$$

$$\frac{x - b}{z} = \frac{x'_r}{f} \quad (2.2)$$

Depth Z can be calculated by combining equations 2.1 and 2.2,

$$z = \frac{bf}{x'_l - x'_r} \quad (2.3)$$

So if parameters f and b are known, the depth map of the whole image can be estimated by calculating the disparity $(x'_l - x'_r)$ of each pair of pixels between corresponding image points. However, how to establish the correspondence between the

objects in the two images is a challenge in stereo imaging. It requires unique matching points to create pairs of relationship. This kind of relationship will be hard to establish when the scene has uniform intensity or occlusions.

2.2 Light field imaging

A LightField camera is one kind of special camera which has a microlens array in front of the imaging sensor. The microscopic lens splits the light rays into many tiny images depending on the corresponding microlens position in the array. Depth information of each pixel can be calculated by tracking each light trace. Although light-field imaging cameras can calculate depth map with acceptable results [2], the limitation of the size of the microlens array on the imager lessens the resolvable resolution.

2.3 Depth from focus

Depth from Focus (DFF) uses the camera parameters to estimate the depth of an object. A sequence of images captured at different lens positions and the sharpness of focus is measured for each one. Then the actual depth is calculated by using the lens law, shown in Figure 2.2.

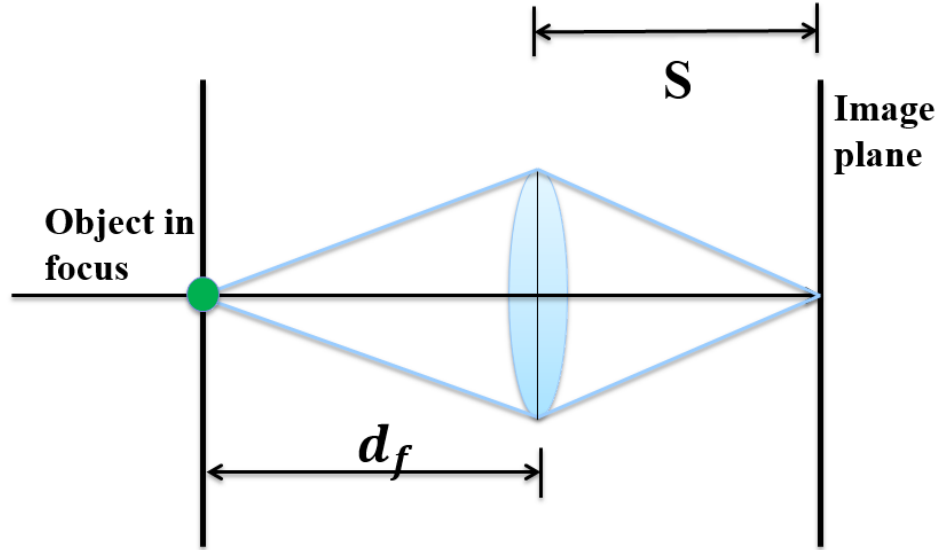


Fig. 2.2. depth from focus

When the object at distance d_f from the lens is in-focus, the image is formed at a distance S on the image sensor. The relation between the focal length of the lens f , the object distance d_f and the image distance S is given by the lens law:

$$\frac{1}{f} = \frac{1}{d_f} + \frac{1}{S} \quad (2.4)$$

In practice, to get different sharp focus images on different objects, a series of images are captured by adjusting either the focal length f or the image distance S . The critical step is how to measure focus. Brenner [17] proposed a method based on summing the squares of the horizontal first derivative. Similarly, the focus could also be measured by convolving the image with either a 3x3 or a 5x5 Laplacian operator [18]. Other methods [19] use image histogram, image statistics or correlation..

Depth from focus method is monocular and can calculate the actual depth using the lens law. Different from stereo imaging, it does not have the correspondence problem. However, to get accurate depth map for each object, DFF requires 10 to

12 images as input. Extra time is therefore needed to adjust the camera parameters before capturing each image, during which the scene must remain stationary.

2.4 Depth from defocus

In theory, all light rays from the same point of an object should be converged at the same point on the image plane, if the point is at the in-focus position. However, if the object is not at the in-focus position, on the image plane, there will not be a clear point but a blurred circular disc. The basic idea of Depth from Defocus (DFD), is to measure the radius of the blur and relate it to the actual depth using the simple lens law. DfD also does not have the correspondence matching problem. In comparison to DFF, the DFD methods only need a few images (usually 2) to compute a reliable depth map.

Subbarao and Gurumoorthy [20] proposed a method for recovering depth by measuring the blurring degree of an edge. The degree of a blurred edge is then fed into Line Spread Function computation. However, Subbarao and Gurumoorthy's method is only powerful for isolated edges.

Based on the inhomogeneous reverse heat equation, Namboodiri and Chaudhuri [21] proposed to estimate the blur information and depth. The heat equation is formed by the Gaussian point spread function. The difference between the observed image and the reconstructed image is then used to estimate the depth information.

Zhuo [22] presented how to recover the defocus map from a single image. The spatially varying defocus blur at the edge locations is estimated in this method. On the input defocus image, the blur is added by a Gaussian kernel. The comparison between the gradients of input and re-blurred images determines the blur amount. By propagating the blurring amount at all the image's edges, the full defocus map is formed.

Many other DfD techniques use two or more images captured by different camera settings to estimate the depth map. For example, Chaudhuri [23] proposed an

algorithm that recovers depth information from a pair of defocus images. In that algorithm, the blur parameter was modeled as Markov Random Field (MRF). Simulated Annealing was used as the Optimization algorithm. More details about DfD Method used in this research will be discussed starting from next chapter.

3. OPTICS AND DFD METHOD

The purpose of this chapter is to explain some of the fundamental theory that is used in Depth from Defocus (DfD) methods. This chapter has been designed to illustrate the main theoretical elements, and has been organized into three sections,

- Lens systems and defocus/depth relationships
- Modelling defocus blur
- Depth of Defocus field

3.1 Lens Systems and Defocus/Depth Relationships

Figure 3.1 shows a single thin lens system. The light rays from the object pass through the thin lens and then converge on the image plane at distance S . The basic equation of this single lens system is given by (3.1):

$$\frac{1}{d_f} + \frac{1}{S} = \frac{1}{f} \quad (3.1)$$

Where the focal length of the thin lens is defined as f , the distance between the lens and the object is defined as d_f , and S represents the distance between the lens and image plane.

When the object is not at the focused position, the light rays will not be converged at the focus point but some other point with distance v . And in the image plane, a defocus blur of radius R is formed as shown in Figure 3.2.

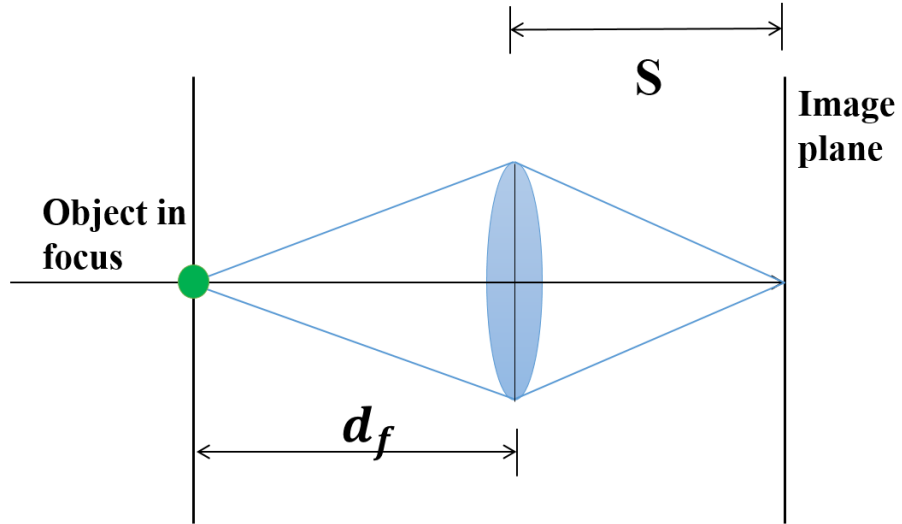


Fig. 3.1. Lens system with object at focused position

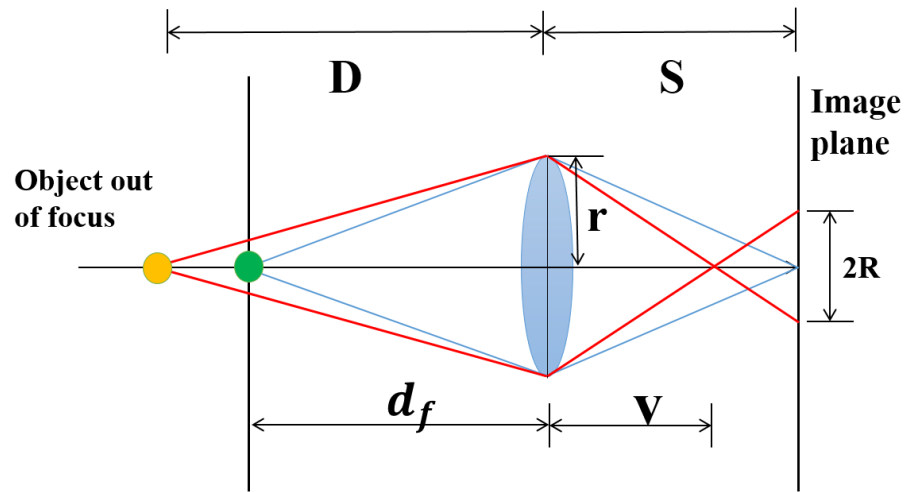


Fig. 3.2. Lens system with object at defocused position

In Figure 3.2, D is defined as the distance between the lens and object at out of focus position, and r is the radius of the lens. Therefore, the relationship is:

$$\frac{1}{d_f} + \frac{1}{S} = \frac{1}{f} \quad (3.2)$$

Also based on geometry theory:

$$\frac{R}{r} = \frac{S - v}{v} \quad (3.3)$$

Then combining equations (3.3) and (3.2), equation (3.4) is formed.

$$R = rS \left(\frac{1}{f} - \frac{1}{D} - \frac{1}{S} \right) = \frac{rd_f f}{d_f - f} \left(\frac{1}{d_f} - \frac{1}{D} \right) \quad (3.4)$$

If the following camera settings are given;

- f : the focal length of the lens
- d_f : the distance of the focused object from the lens
- r : effective radius of the lens

The radius of the blur circle, R , is a non-linear monotonically increasing function of D , the distance between object and lens. This implies the image captured by camera would have increasing blur for increasing distance between the object and lens.

3.2 Modelling Defocus: Point Spread Function

As mentioned in section 3.1, the radius of defocus blur is related to the actual depth. Then estimation of the depth can be converted to estimating defocus blur level. As is known, for each pixel in one image, the defocus blur can be modeled by convolving one in-focus image with a point spread function (PSF).

The point spread function is a geometric result after the light rays passes through the lens. If incident light energy is A units, the the focused image can be expressed as $A\delta(x, y)$. Here $\delta(x, y)$ is the Dirac delta function [24]. And if $h(x, y)$ is defined as the response function of the input signal $\delta(x, y)$ in the lens system. Based on the

assumption that the blurred point light is circular in shape, the intensity distribution can be modeled as:

$$h(x, y) = \begin{cases} \frac{1}{\pi r^2} & \text{if } x^2 + y^2 \leq r^2 \\ 0 & \text{otherwise} \end{cases} \quad (3.5)$$

In order to avoid lens diffraction, as suggested in [5], a symmetric two-dimensional Gaussian function $h(x, y)$ (3.6) is used to model the PSF.

$$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.6)$$

Where σ is 2D Gaussian blur parameter such that

$$\sigma = k \times R \quad \text{for } k > 0 \quad (3.7)$$

k is a constant proportional characteristic for a given lens. And σ and R are both defined in millimeters(mm). Referencing Eq. 3.8, σ in pixel can also be calculated based on the relationship between R in pixels and R in millimeters (3.8). $Sensor_width_{mm}$ stands for the width of camera sensor in millimeter; $Image_width_{pixel}$ is the width in pixel of the image take from camera. It determined by the resolution sensor.

$$R_{pixel} = \frac{R_{mm}}{Sensor_width_{mm}} \times Image_width_{pixel} \quad (3.8)$$

Once PSF $h(x, y)$ is known, a defocused image is denoted by a convolution:

$$b(x, y) = f(x, y) * h(x, y) \quad (3.9)$$

The 2D Gaussian blur parameter σ is proportional to R , therefore the depth D can be calculated using Equation (3.4).

3.3 Depth of defocus field

As presented in previous sections, for a near-focus defocus image, the objects closer to camera are in-focus and the objects far away from camera are out of focus.

The defocus blur increases as the distances of objects to camera increases. However, if the distance is too large, the differences in blur cannot be distinguished. This can be demonstrated as follows: By rearranging Eq. 3.4,

$$R = \frac{D - d_f}{D} \times \frac{f}{d_f - f} \times r \quad (3.10)$$

Where r is effective radius of lens which is proportional to aperture size. Aperture size is defined as:

$$Aperture = \frac{f}{f_{number}} \quad (3.11)$$

Here f is focal length of the lens, f_{number} determines the size of iris. So Eq. 3.11 can be modified to Eq. 3.12

$$R = \frac{D - d_f}{D} \times \frac{f}{d_f - f} \times \frac{f}{f_{number}} \quad (3.12)$$

For fixed focal length f , F-stop number f_{number} , and focus distance d_f , the blur radius is proportional to the distance D of the out of focus object as the distance changes. In order to compute the resolvable depth field of view and resolvable depth step size, these equations will set the limits. First, we combine Eq. 3.7, Eq. 3.8 and Eq. 3.10. Next, use $D1$ and $D2$ to define the depth of objects located at different locations. If the difference between $D1$ and $D2$ is small and $D1$, $D2$ are big enough, here is no difference of the defocus blur for the objects on these two distances shown in images. At that point, the radius of blur circle values R in pixels are less one 1 pixel, which can be calculated by the equation below. This will set the limit on the resolvable depth step size.

$$\left| \frac{1}{D_2} - \frac{1}{D_1} \right| \times d_f \times \frac{f}{d_f - f} \times \frac{f}{f_{number}} \times \frac{Image_width_{pixel}}{Sensor_width_{mm}} < 1 \quad (3.13)$$

T and Q are used here to simplify the notation (Eq. 3.14, Eq. 3.15), and Eq. 3.12 and Eq. 3.13 are rewritten to Eq. 3.14 and Eq. 3.15

$$T = \frac{f}{d_f - f} \times \frac{f}{f_{number}} \quad (3.14)$$

$$Q = \frac{Image_width_{pixel}}{Sensor_width_{mm}} \quad (3.15)$$

$$R = \frac{D - d_f}{D} \times T \quad (3.16)$$

$$\left| \frac{1}{D_2} - \frac{1}{D_1} \right| \times d_f \times T \times Q < 1 \quad (3.17)$$

In order to find the point at which the maximum depth is indecipherable from infinity, the simplification is made assuming $D_1 > D_2$, then $\left| \frac{1}{D_2} - \frac{1}{D_1} \right|$ has a maximum value at $1/D_2$ when D_1 is infinity. So based on Eq. 3.17, when depth $D \geq d_f \times T \times Q$, the blur radius R will remain the same value $T - 1/Q$. For a given d_f , the radius of defocus blur, R , increases as the depth increases. While the increasing rate of R is lower and lower until it meets its largest value: $T - 1/Q$.

For example, if one specific camera has the settings as shown below:

- $f = 9mm$
- $f_{number} = 3.7$
- $Sensor_width_{mm} = 4.9mm$
- $Image_width_{pixel} = 1024\ pixels$
- Focus distance $d_f = 1\ m$

Then regarding to Equ. 3.14 and Equ. 3.15, $T = 0.0221$ and $Q = 208.33$. So maximum radius of blur R_{max} can be calculated by using $R_{max} = T - 1/Q = 0.0172m$ and the corresponding maximum depth D_{max} can be calculated by using $D_{max} = d_f \times T \times Q = 4.604m$. This means if use this camera with the settings above, if the object distance is larger than D_{max} , the radius of defocus blur will not change but keep the value R_{max} . So for this case, the workable region for EDfD is from 1m to 4.604m.

In EDfD algorithm, defocus blurs are divided into 256 steps. For this case, the range for R is from 0 to 0.0172. So step interval is $0.0172/255$. Figure 3.3 shows defocus blur step is a non-linear monotonically increasing function of depth until to the maximum depth position.

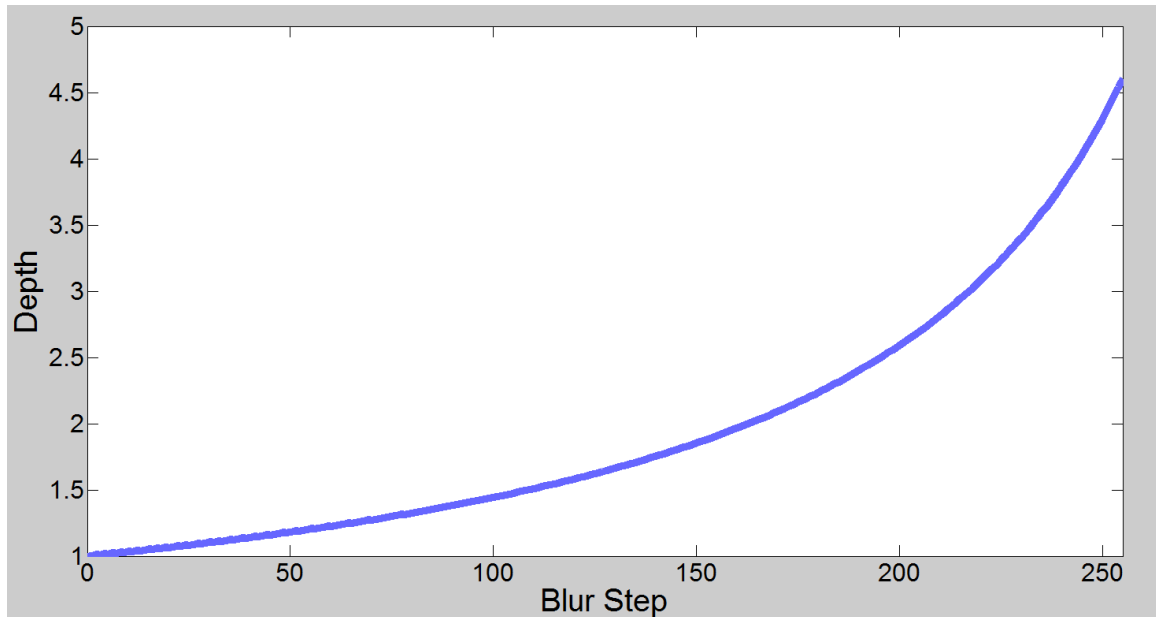


Fig. 3.3. Illustration of Depth of defocus field

3.4 Summary

This chapter introduced the method of modeling defocus blur by using Gaussian Point Spread Function. Also by finding the relationship between defocus blur radius

and actual depth, the radius of the blur circle is shown to be a non-linear monotonically increasing function with depth. Therefore, the depth estimation problem is equivalent to estimating defocus blur level. This research presents a new analysis of the depth of defocus field, one of the innovations of this research. If camera settings are given, the workable field for EDfD can be determined. Additionally, the defocus blur steps are also non-linear monotonically increasing functions related to depth until to the maximum depth position. After introducing the relationship between depth and defocus blur, new research improvements of the EDfD method will be presented in next chapter.

4. NEW RESEARCH IMPROVEMENTS

This new EDfD algorithm is extended from classical depth from defocus method. For the EDfD method, color, edge and texture information, are added to improve the accuracy of depth estimation. Section 4.1 introduces the overview of this EDfD algorithm. Section 4.2-4.4 show the benefits of incorporating color, edge, and texture information.

4.1 Algorithm overview

The classical DfD algorithm compares individual pixels of the defocused image to the all in-focus image passed through the Gaussian filters, according to the energy function of Equation (5.10). The implementation of this research is shown in Figure 4.1. In contrast to the traditional approaches which only have used grayscale images as input images, EDfD research takes advantage of the color images. An all-focus image and a defocused image of the same scene is the input to the EDfD. The first step converts both of these two color images into YCbCr channel. The Y channel contains the intensity of color image, and the Cb and Cr channels are added to improve the accuracy of depth estimation.

After splitting the two input images into three channels, a new preprocessing procedure is used on the in-focus image before doing MAP estimation. The preprocessing procedure has two main tasks. Image processing is used to distinguish textured and texture-less regions of the image. Second, the edges in the image are isolated with a highpass filter.

Next an initial depth map is combined with the output of previous steps as input to the revised MAP estimator, and the final depth map is the output.

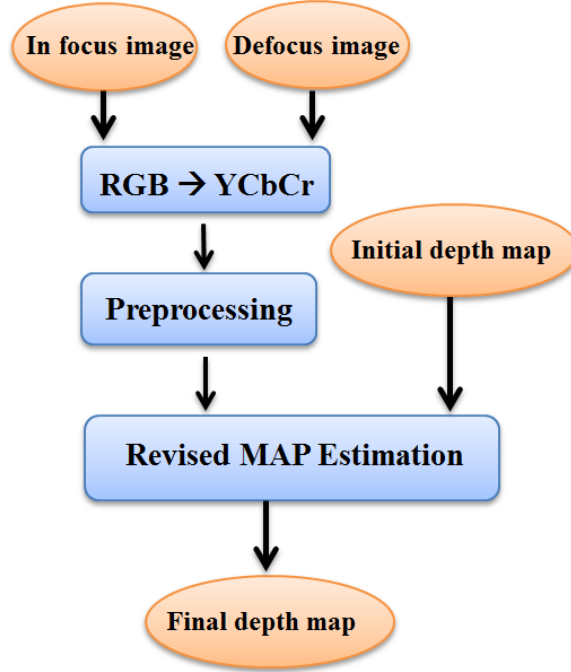


Fig. 4.1. Proposed algorithm overview

4.2 Initial depth map generation

Initial depth map generation is a very important procedure which is the baseline of the whole algorithm. The new approach is to use the EM/MPM optimization algorithm in the MAP Estimator. In Figure 4.2, the greyscale all in-focus image I_{inf} and defocused image I_{def} are the input to the initial MAP estimator. 256 levels of blurred images $I_{b1}, I_{b2}, \dots, I_{b256}$ are created by applying 256 different Gaussian filters to I_{inf} . The Gaussian blur parameters are chosen with equal step size. At the same time, depth class label map I_s is initialized as a MRF with the same image size as I_{inf} and I_{def} . Starting from I_s as initial depth map, $I_s, I_{def}, I_{b1}, I_{b2}, \dots, I_{b256}$ are passed to the initial MAP estimator.

For each pixel c with depth class label k ($k = 1, 2, \dots, 256$), the data term, $d(c, k)$, and smoothness term, $prior(c, k)$, are calculated using Equation (4.1) and (4.2). Based on Equation (5.10) the energy function can be expressed as (4.3).

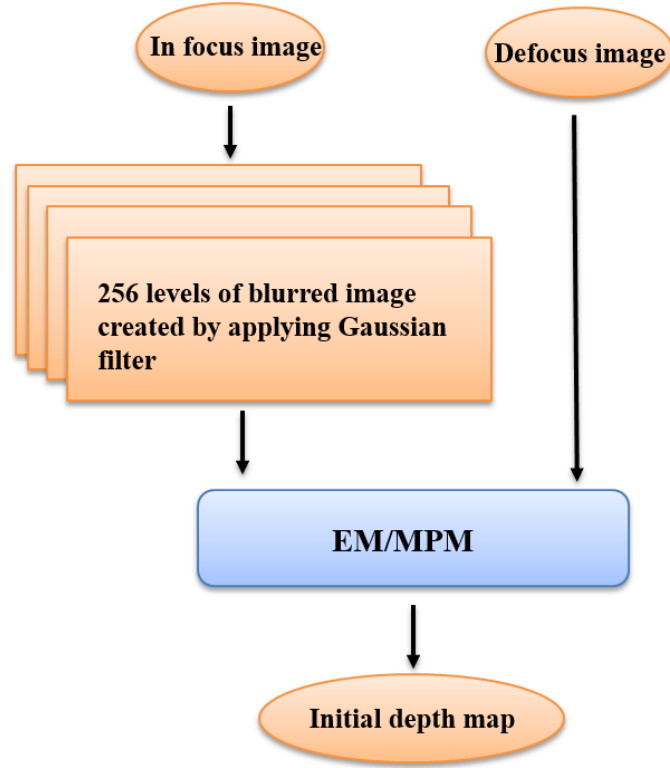


Fig. 4.2. Initial MAP Estimation

$$d(c, k) = |I_{def}(c) - I_{bk}(c)| \quad (4.1)$$

$$prior(c, k) = \sum_{r \in N_c} |I_s(r) - S_c(k)| \quad (4.2)$$

$$\logpost(c, k) = \log \sigma_k + \frac{(I_{def}(c) - I_{bk}(c))^2}{2\sigma_k^2} + \sum_{r \in N_c} |I_s(r) - S_c(k)| \quad (4.3)$$

Finally, the initial depth map I_s is generated by optimizing $\logpost(c, k)$ for each pixel using EM/MPM.

4.3 Preprocessing

From the initial depth map shown in Figure 4.2, new image processing is used to improve the quality of depth map. One challenging case is where some regions in the image have little or no details with which to infer the depth. For the traditional DfD algorithm, Gaussian filter would remove the low frequency objects in the scene which do not contain edges (spatial high frequencies), and the inference algorithm then does not have enough detail to choose one solution. So the initial depth map would have some ambiguous depth values in some texture-less regions. The baseline algorithm can achieve an accurate result in a textured region or on the edges. However to handle the texture-less regions, two new preprocessing functions are introduced. As shown in Figure 4.3, the input to the preprocessing is one in-focus image. The first function uses a highpass filter to find the edges, and then generates a highpass image with the same size as the input. The second function is a texture region identifier which determines whether this region is texture-less.

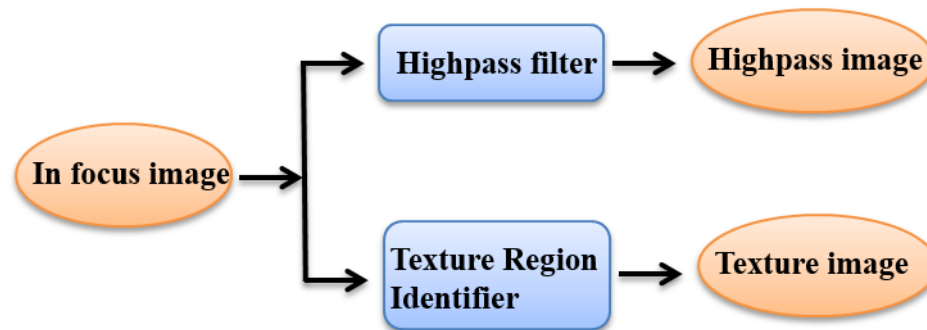


Fig. 4.3. Preprocessing procedure

Figure 4.4 illustrates an example of input and output of the preprocessing procedure. Column 1 shows the in-focus image. Column 2 shows the highpass image output after applying the filter. Column 3 shows the textured image output from the textured region identifier. As defined in [8], the texture-less regions are regions where the squared horizontal intensity gradient averaged over a square window is below a

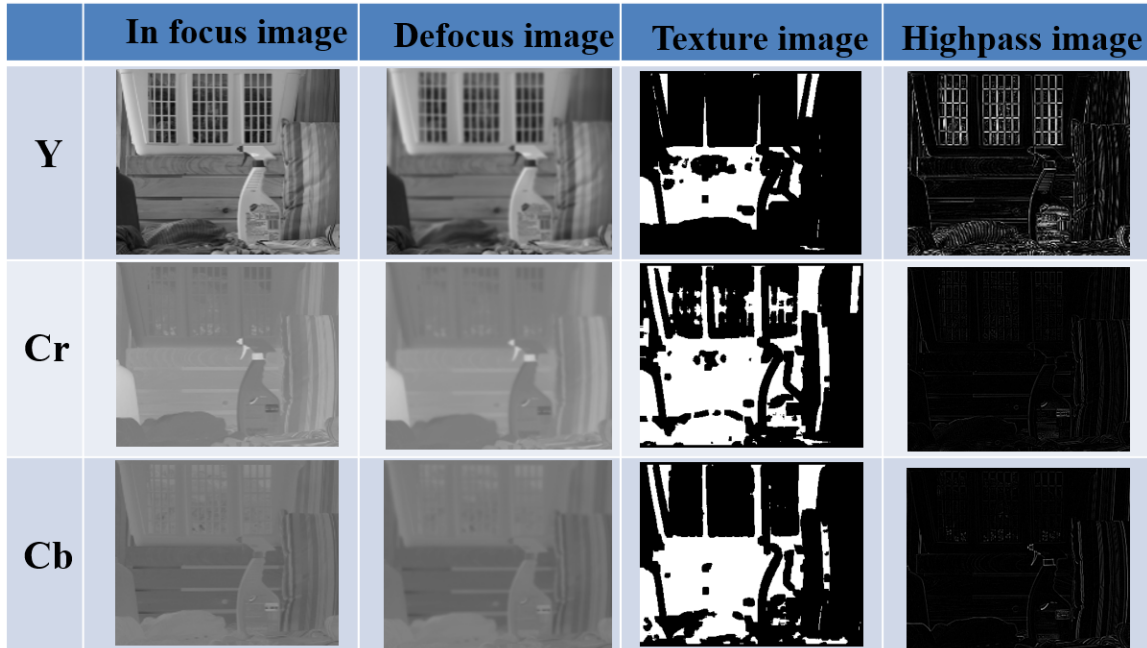


Fig. 4.4. Example of input and output of preprocessing procedure

given threshold. As Figure 4.4 shows, the textured images are binary ones, where a white region means texture-less and the black region is textured.

Figure 4.5 illustrates the benefits of implementing preprocessing on small texture-less regions. Figure 4.5(a) shows a synthetic all in-focus image with a no-texture region in the center. Figure 4.5(c) shows the synthetic ground truth of a texture-less region and textured region at different depths. As Figure 4.5(e) shows, the traditional method in an initial depth map can only find accurate results in a textured region or on the boundaries. The preprocessing results in the Figure 4.5(f) showing the improved final depth map (much closer to the ground truth) used as input to “Revised MAP Estimation” in next subsection.

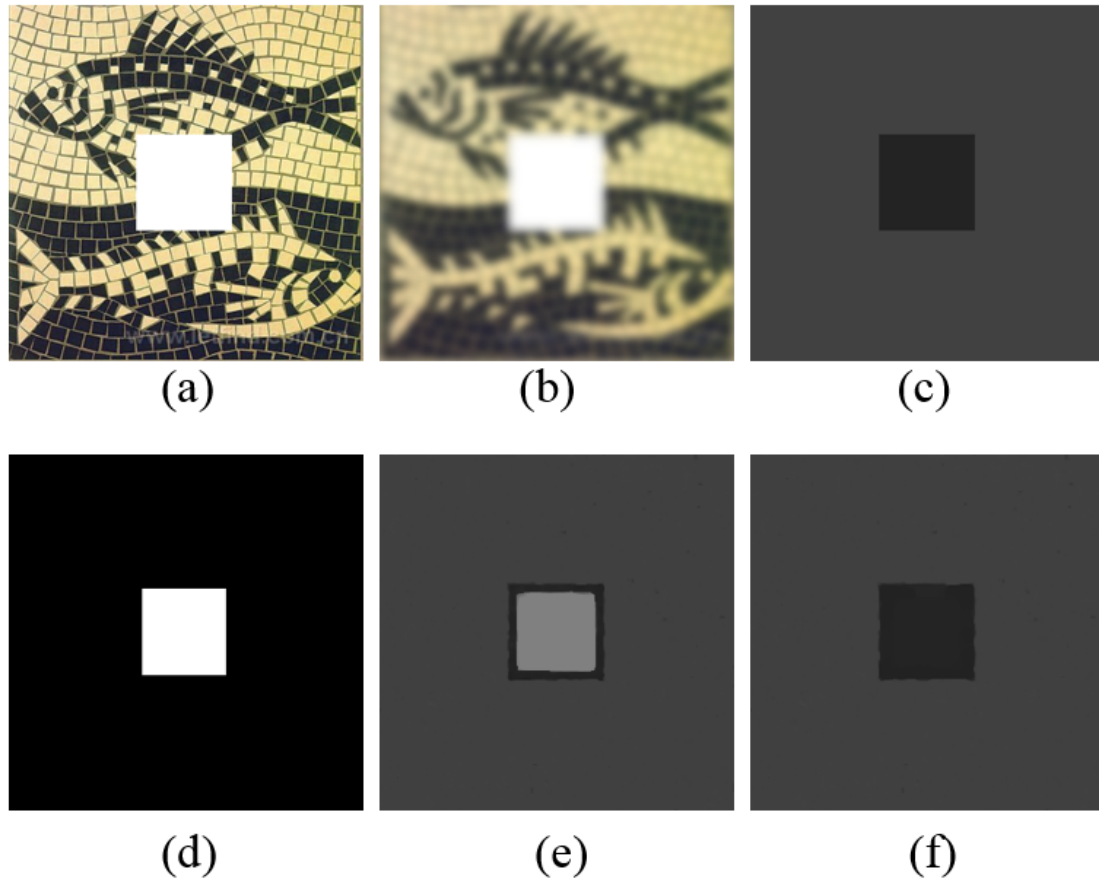


Fig. 4.5. Example of the benefits of preprocessing procedure. (a) All in-focus image (b) Defocus image (c) Depth ground truth (d) Texture image (e) Initial depth map (f) Final depth map after using texture information

4.4 Revised MAP Estimation using texture information

In the next section, Equation (5.10) is introduced as the energy function, this is formed from two terms: a data term and a smoothing term. These terms are modified by the texture information using the weighting factor β . Since the texture-less region has few details to infer the depth, the goal is to de-emphasize the data term, and rely more on the prior smoothing term in the optimization. Therefore, for each channel (Y,Cb,Cr), it is important to maintain the weighting factor in textured

regions; and modify the weighting in texture-less regions. The new research result is due to providing a higher weighting on the neighboring pixels which are on the boundary of these texture-less regions.

The decision tree for this adaptation is shown in Figure 4.6. For each channel, the first step is to identify if pixel c belongs to texture-less region. If not, the next step is to determine whether pixel c is on the edge. If “Yes”, then a smaller value, β_1 , is given to β , otherwise β is set to be a larger value, β_2 . The last step follows equation (5.10) for MAP estimation.

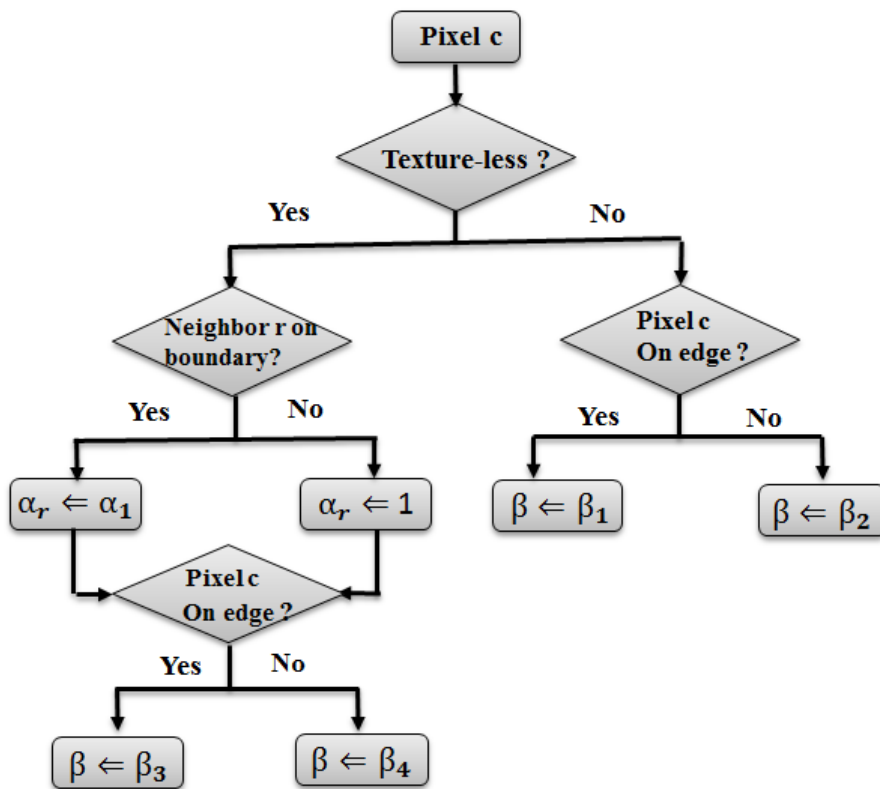


Fig. 4.6. Revised MAP Estimation

If pixel c belongs to texture-less region, the 8 neighboring pixels will be checked first to form a new modified energy function, introduced in equation (4.4). A new weighting factor α_r is involved. If neighbor pixel r is on the boundary of texture-less region which means it could have a higher probability of the correct depth, then α_r

will be set a large value α_1 , typically bigger than 1. Otherwise, α_r equals 1. If at least one neighbor pixel r is found on the boundary which has the similar intensity to the center pixel c , then c is merged into a textured region.

The next step is the same as in the textured region, if a pixel c is on the edge, then a smaller value, β_3 , is given for β , otherwise β is set to be a larger value β_4 ($\beta_4 > \beta_3 > \beta_2 > \beta_1$).

$$S_c = \arg \min \left\{ \log \sigma_{S_c} + \frac{(g_c - b_{k_c})^2}{2\sigma_{S_c}^2} + \beta \sum_{r \in N_c} \alpha_r |S_r - S_c| \right\} \quad (4.4)$$

4.5 Summary

This chapter presents the new improvement of this research from classical depth from defocus method. Color information is added to improve the accuracy of depth estimation. Another innovation uses the edge and texture information determine the relative weights of the data and smoothing terms in the energy function. Based on this information, ambiguous nature of blur in the textureless areas is substantially improved. The EDfD algorithm was introduced in this chapter and more details about different regularization algorithms used in this research will be presented in next chapter.

5. REGULARIZED DEPTH FROM DEFOCUS

The energy function of EDfD algorithm is developed in section 5.1. Different regularization algorithms are also introduced in this chapter. EM/MPM method (section 5.2) gives better results compared with other methods. While graph-cut method gives even better performance which is shown in section 5.3.

5.1 MAP-MRF

The general MAP Estimation technique has been widely used in such applications such as denoising, deblurring and segmentation. In this research, it is combined with Markov Random Filed (MRF) and Bayesian statistical estimator to estimate depth label for each pixel as shown in Figure 5.1.

Two input images are used to determine the blur. The first is an all-focus or in-focus image $f(x, y)$, and the second is the defocused image $g(x, y)$. So $g(x, y)$ can be represented as:

$$g(x, y) = f(x, y) * h(x, y) + w(x, y) \quad (5.1)$$

Where $h(x, y)$ is the space-variant blur function modeled by the Gaussian kernel, and $w(x, y)$ is the noise.

Let S denote the depth label of pixel, then a prior distribution $p(s)$ can be used with a Markov Random Field (MRF) model. The blur is quantized to 256 classes (8 bits) of space-variant blur parameter σ . Then, based on Equation (5.1), the a posteriori probability distribution of S can be expressed as: $P(S = s|G = g)$. Using Bayes equation, the closed form of the distribution is given below (5.2)(5.3):

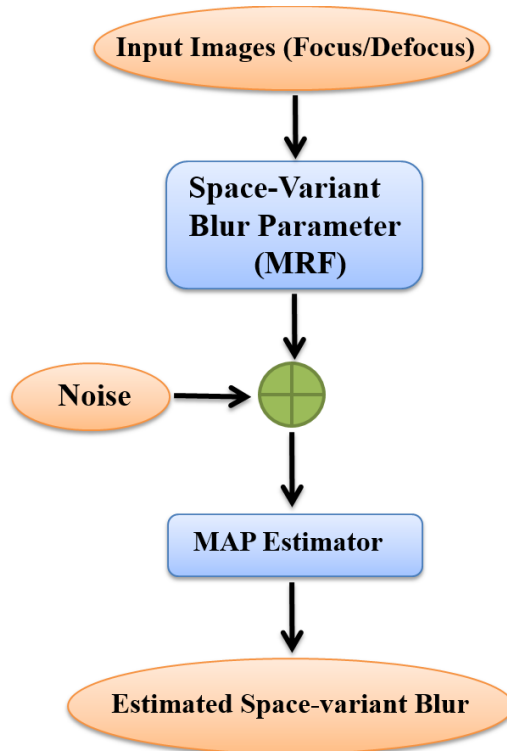


Fig. 5.1. General MAP Estimation block diagram

$$P(S = s|G = g) = \frac{P(G = g|S = s) P(S = s)}{P(G = g)} \quad (5.2)$$

$$p(s) = \frac{1}{z} \exp \left(-\beta \sum_{r \in N_c} |S_r - S_c| \right) \quad (5.3)$$

Maximizing $P(S = s|G = g)$ is equivalent to minimizing the energy function described by Equation (5.4), as shown in [4]. This is done on a pixel by pixel basis, so the blur class (value) will vary over the image.

$$U(S) = |g(x, y) - f(x, y) * h(x, y)|^2 + \beta \sum_{r \in N_c} |S_r - S_c| \quad (5.4)$$

This energy function has two terms. The first term, the data-dependent term, is the

mean squared error difference of the blur image and a particular choice of blur kernel convolved with the in-focus image. The second term, sometimes called the smoothing term, calculates the differences in choice of depth classes in every 8-neighbor clique. This second term, the Bayesian prior, measures how different a choice of depth is from its immediate neighbors. In Equation (5.3), S_c is depth class label of center pixel c ; S_r is depth class label of neighbor r ; N_c is defined as all 8 neighbors of center pixel c . α is a weighting factor which balances the data term and smoothing term. The better choice of blur class value will minimize this energy function, allowing the convolution, $b(x, y)$, to be closer to the true defocus $g(x, y)$, while at the same time providing a smoothness among all neighboring pixels.

5.2 EM/MPM

In order to find the best choice of blur label for each pixel, optimization process is needed. The MAP optimization reported in Chaudhuri [4] uses Simulated Annealing (SA) as the optimization process. The choice in this research is EM/MPM, which has some advantages compared to SA, both in convergence speed and in optimization over local areas. As will be seen in the results, the performance is compared between SA and EM/MPM methods on the same test data, and EM/MPM is chosen because of its overall better accuracy.

The general EM/MPM algorithm consists of two parts: Expectation Maximization (EM) and Maximization of Posterior Marginals (MPM) [25]. The EM algorithm finds the estimates for Gaussian mean and variance, while MPM classifies the pixels into N class labels, using estimated parameters from EM.

The Gaussian mixture model used here means that Equation (5.2) is modified into (5.5) and (5.6). Here $\sigma_{S_c}^2$ is variance of each class; μ_{s_c} is mean for each class; s_c is blur class of the pixel c ; g_c is the pixel in the input defocussed image at location c ; θ is the vector of means and variances of each class.

$$p_{s|g}(s|g, \theta) = \frac{f_{g|s}(g|s, \theta)p_s(s)}{f_g(g|\theta)} \quad (5.5)$$

$$f_{g|s}(g|s, \theta) = \prod_{c \in C} \frac{1}{\sqrt{2\pi\sigma_{S_c}^2}} \exp \left\{ -\frac{(g_c - \mu_{s_c})^2}{2\sigma_{S_c}^2} \right\} \quad (5.6)$$

At the beginning of this process, a random blur class label is initialized into every pixel in S . An evenly distributed vector of means and variances is used as a starting point for the classes. Then, the estimate of S is formed by iterating several times through the whole image. At each iteration, two steps are performed: the expectation step and maximization step. First maximization step is performed based on Equation (5.7), (5.8) and (5.9), then in expectation step, iterating using MPM to find the best log-likelihood of the probability that a particular pixel belongs to one of the 256 blur classes.

$$\mu_k(c) = b_k(c) = f(c) * h_k(c) \quad (5.7)$$

$$\sigma_k^2 = \frac{1}{N_k} \sum_{c \in C} (g_c - \mu_k(c))^2 p_{S_c|g}(k|g, \theta) \quad (5.8)$$

$$N_k = \sum_{c \in C} p_{S_c|g}(k|g, \theta) \quad (5.9)$$

For MPM, convergence is achieved by choosing the best blur class label which minimizes the expected value of the number of misclassified pixels as proved in [7]. The final energy function is calculated in the log domain, eliminating constants and exponentials as shown in equation (5.10).

$$S_c = \operatorname{argmin} \left\{ \log \sigma_{S_c} + \frac{(g_c - b_{k_c})^2}{2\sigma_{S_c}^2} + \beta \sum_{r \in N_c} |S_r - S_c| \right\} \quad (5.10)$$

Before implementing the proposed algorithm on video camera, the accuracy has been verified by introducing a synthetic blur based on images that have corresponding

real ranging ground truth. For this purpose, the test images and ground truth images from the Middlebury 3D imaging website [26, 27] were used.

Middlebury does not have defocus images, only all-focus, so this research uses the Middlebury ranging camera high resolution ground truth images and the in-focus images to generate synthetic defocussed images. At each pixel c in the ground truth image, there was assigned a blur parameter θ_c based on the depth ground truth brightness. A total of 256 levels of blur are linearly mapped corresponding to the 256 levels of brightness (brighter means closer to the camera). As mentioned in previous section, the blur function is assumed to be Gaussian. After applying these various Gaussian blurs to each pixel in the all in-focus image, a synthetic defocus image is generated. Finally, the in-focus image and synthetic defocus image are used as two input images for verifying the accuracy of the proposed EDfD algorithm.

Figure 5.2 shows the experimental results of the Middlebury data. Figure 5.2(a) and (c) are the in-focus image and ground truth, respectively. These scenes are directly downloaded from the Middlebury website. Figure 5.2(b) is the synthetic defocus image generated by the method above. Figure 5.2(d), (e) and (f) are initial, intermediate and final depth map results. Figure 5.2(d) shows the initial depth map result which using the greyscale image as input with the new EM/MPM optimization method. Figure 5.2(e) shows the intermediate result after adding in the color components of the image. This YCbCr data provides more information for improving MAP estimation. The figure 5.2(d) and (e) comparison shows that adding color information reduces misclassifications. However, some problems still appear in the texture-less regions. Finally, in Figure 5.2(f), the depth map result is includes the full EDfD method and the accuracy is improved significantly in small texture-less regions, due to the new EDfD.

Figure 5.3 compares depth map results of six different images from the Middlebury dataset, with two techniques from the DfD literature. Column (a) shows the source input in-focus images. Column (b) shows ground truth ranging camera depth. In Column (c), the images are depth map results using the EDfD method. The results

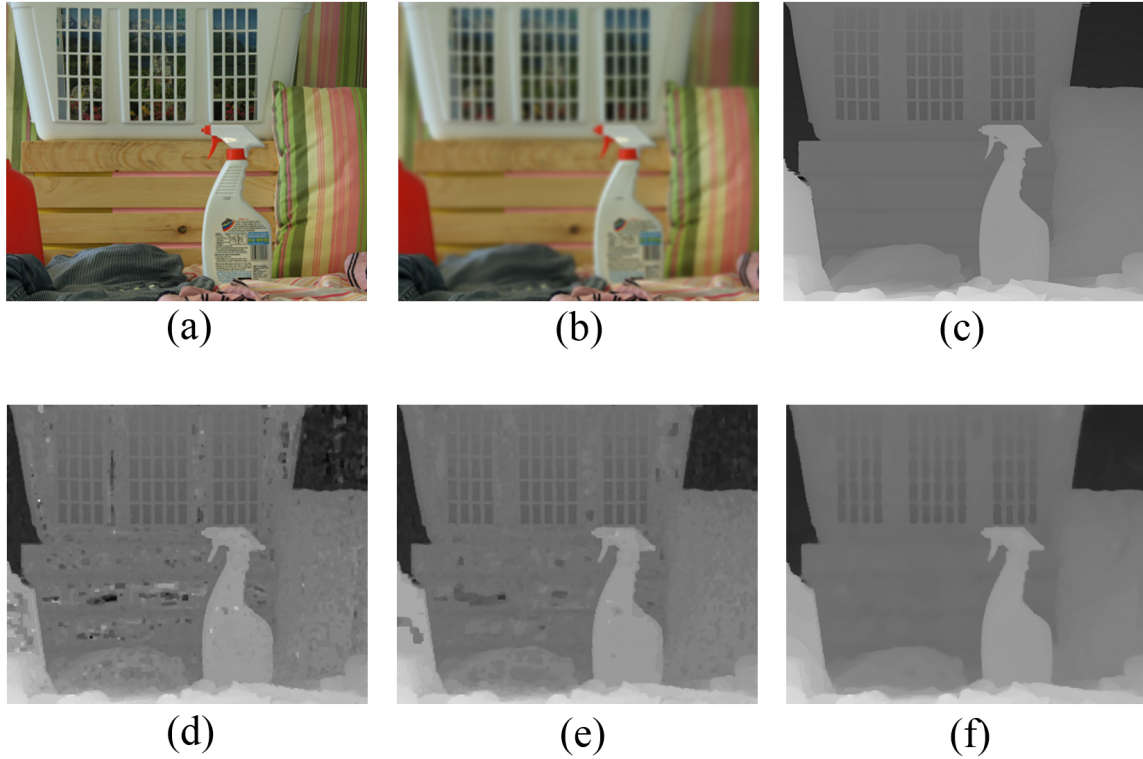


Fig. 5.2. (a) In-focus image (b) Synthetic defocus image (c) Ground truth (d) Initial depth map (grayscale input) (e) depth map(color input) without texture information (f) final depth map

shown in Column (d) and (e) are using Chaudhuri's DfD method [28] and Favaro's Shape from Defocus method [29] respectively. Chaudhuri's DfD method is based on traditional DfD algorithm, the difference is that it uses Simulated Annealing (SA) as the optimization method for MAP estimation. The Shape from Defocus algorithm uses two defocussed images as input. One is far-focus image and another is near-focus image. In order to fairly compare this method with EDfD, the number of classes was increased to 256 levels. Column (f) contains the 3D view maps using depth information from the EDfD results.

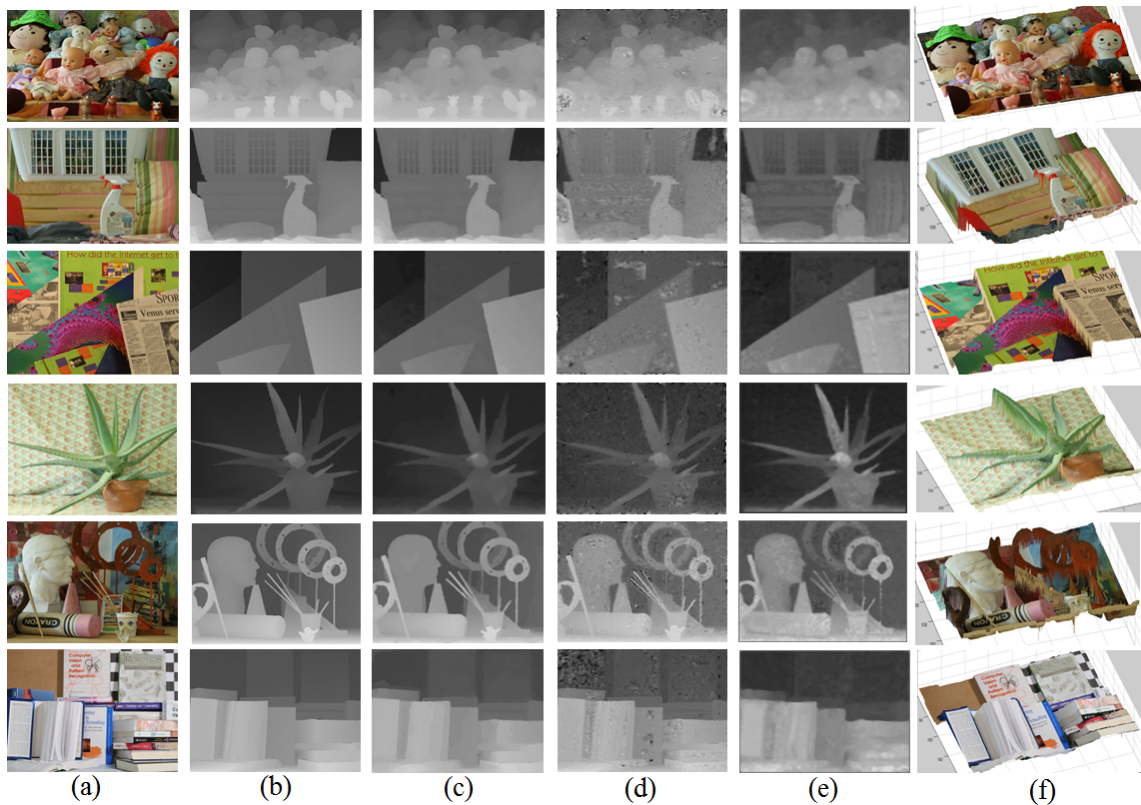


Fig. 5.3. Middlebury results (a) In-focus image (b) Ground truth (c) EDfD results (d) SA results (e) Shape from defocus results (f) 3D view maps

Using the Root Mean Square Error (RMSE) of the calculated depth map against the ground truth, Table 5.1 and Figure 5.4 compare the proposed EDfD results to the results using other methods. Eight sample images are compared from the Middlebury dataset: Aloe, Art, Baby, Books, Doll, Laundry, Poster and Teddy. The EDfD method is shown against four different methods. Two methods are the closest previous literature methods: Simulated Annealing (SA), Chaudhuri's [4] DfD, method and Favaro's [29] Shape from Defocus method (SfD). In addition two new additional methods were explored: CME (Color plus the EM/MPM) and GME (Gray plus EM/MPM). These two methods are used to generate intermediate and initial results respectively as illustrated in Figure 5.2(d) and (e). From Table 5.1 and Figure 5.4,

it is shown that for each test image, the proposed EDfD method achieves the most accurate results. While the average RMSE for EDfD is 4.677, which indicates the error rate is about $4.677/256=0.018$. The average accuracy is 98.18%.

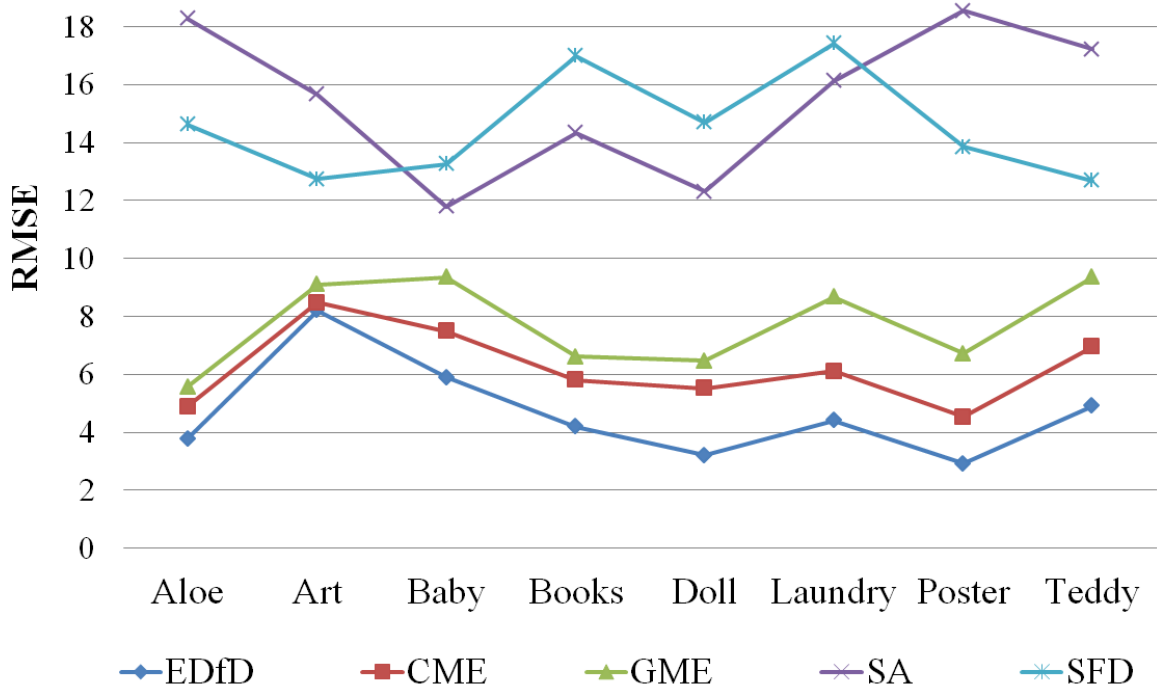


Fig. 5.4. Comparison with other methods on Middlebury image data

Table 5.1
Experimental results comparison, RMSE

Image	EDFD	CME	GME	SA	SFD
Aloe	3.7825	4.8843	5.5787	18.2668	14.6188
Art	8.2107	8.4762	9.0997	15.6589	12.7344
Baby	5.8874	7.4889	9.3466	11.7794	13.2551
Books	4.1916	5.7961	6.6030	14.3384	16.9813
Doll	3.2111	5.5235	6.4619	12.3158	14.6848
Laundry	4.4057	6.1087	8.6766	16.1305	17.4083
Poster	2.9186	4.5216	6.7107	18.537	13.8500
Teddy	4.8989	6.9491	9.3475	17.2097	12.6796

5.3 Graph-cuts

In the field of computer vision, Graph Cuts is usually used as a very powerful energy optimization algorithm. Applications like image segmentation and stereo imaging are associated with minimum cut of weighted graphs [30] that represent the linkages between the pixel values. For a normal weighted graph, it always consists vertices, V , and edges, E . If the edges do not have direction, the graph is called an undirected graph. The "Graph" in Graph Cuts, is a special undirected graph $G = \langle V, E \rangle$, where V and E are the sets of vertices and edges, respectively. This kind of Graph usually contains another special node called a terminal. Here are two types of terminals: source, S , and the sink, T . All the vertices should connect with terminals. For the graph G in the Graph Cut method, here are two types of edges [30]:

- N-link: the edges connect the pixels with their neighbors.
- T-link: the edges connect the pixels with terminals.

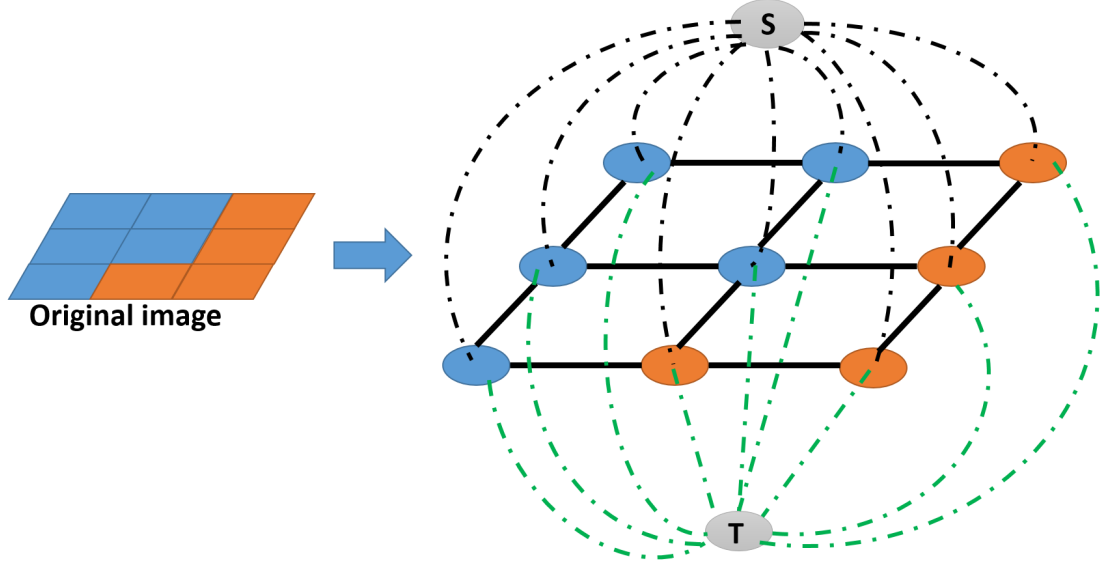


Fig. 5.5. Graph Cut for segmentation example

Figure 5.5 shows a S-T graph of an image. Each pixel corresponds to an vertex in S-T graph. The figure has these two types of edges. The solid line represents an n-link which connect pairs of neighboring pixels. The dashed line represents a t-link which connects pixels and terminals.

Every edge in this S-T graph has a non-negative weight or cost. Cutting an N-link edge will have a penalty cost for neighboring pixels. And similarly, cutting a T-link edge will lead a cost for assigning the corresponding label to the pixel. So after one cut, the cost of all edges has the minimum value, it is called minimum-cut. The max-flow/min-cut method developed by Boykov and Kolmogorov [30] used the energy function shown below (Eq. 5.11) to obtain the minimum cut of S-T graph.

$$E(L) = \sum_{s \in S} D_s(L_s) + \sum_{(s,r) \in N} V_{s,r}(L_s, L_r) \quad (5.11)$$

where L is a set of labels for each pixel in image, $D_s()$ is a data penalty function of pixel s . $V_{r,s}()$ indicates the similarity of the pixel with its neighbors. And N is the set of all pairs of neighboring pixels. By minimizing the energy function, the

original image can be segmented into different parts. The research cited proved that finding the minimum cut is the same as to finding the maximum flow. The most common algorithm to find maximum flow are the pushrelabel algorithm [31] and the FordFulkerson algorithm [32].

Depth from defocus algorithm can be described as assigning a label to each pixel in such a way that an energy function (Eq. 5.10) is minimized. The energy function is a map from the set of all possible labels and is minimized when the segmentation best conforms to a cut model. By using graph-cut algorithm to minimize energy function (Eq. 5.10), 256 blur classes are used as nodes and the pixels in initial depth map are used as vertices for the *S-T graph*.

For this kind of *multi-label graph-cut* problem, Boykov et. al. [33] proposed a fast approximation algorithm called *α -expansion* which is used in this thesis. *α -expansion* is an iterative optimization method. In every iteration, for each pixel, new labels would be obtained if here are better than choices the current ones. The energy function will finally converge when here is no better label could be found.

Figure 5.6, Figure 5.7 and Figure 5.8 compare depth map results of eight different images from the Middlebury dataset. For each figure, Column (a) shows the source input in-focus images. In Column (b), the images are depth map results using the EDfD with EM/MPM method. The results shown in Column (c) is using EDfD with Graph-cut. Column (d) shows ground truth ranging camera depth.

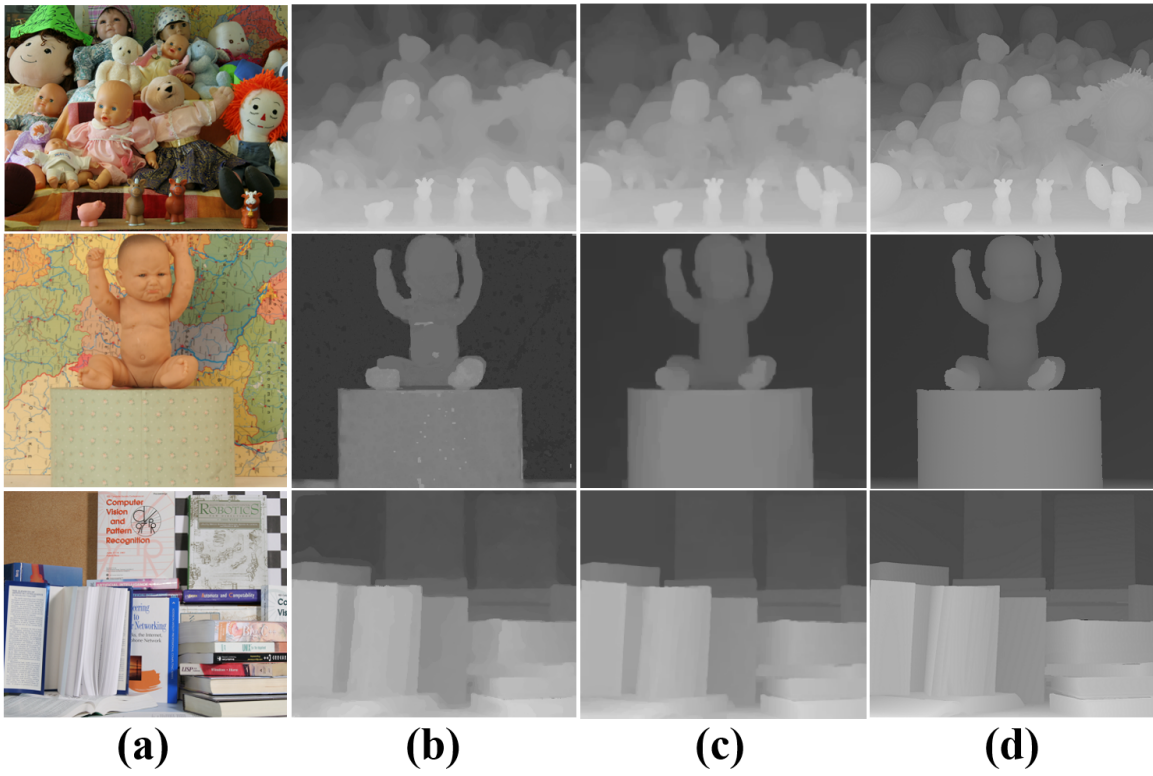


Fig. 5.6. Middlebury results 1 (a) In-focus image (b) EDfD (use EM/MPM) (c) EDfD (use Graph-Cut) (d) Ground truth

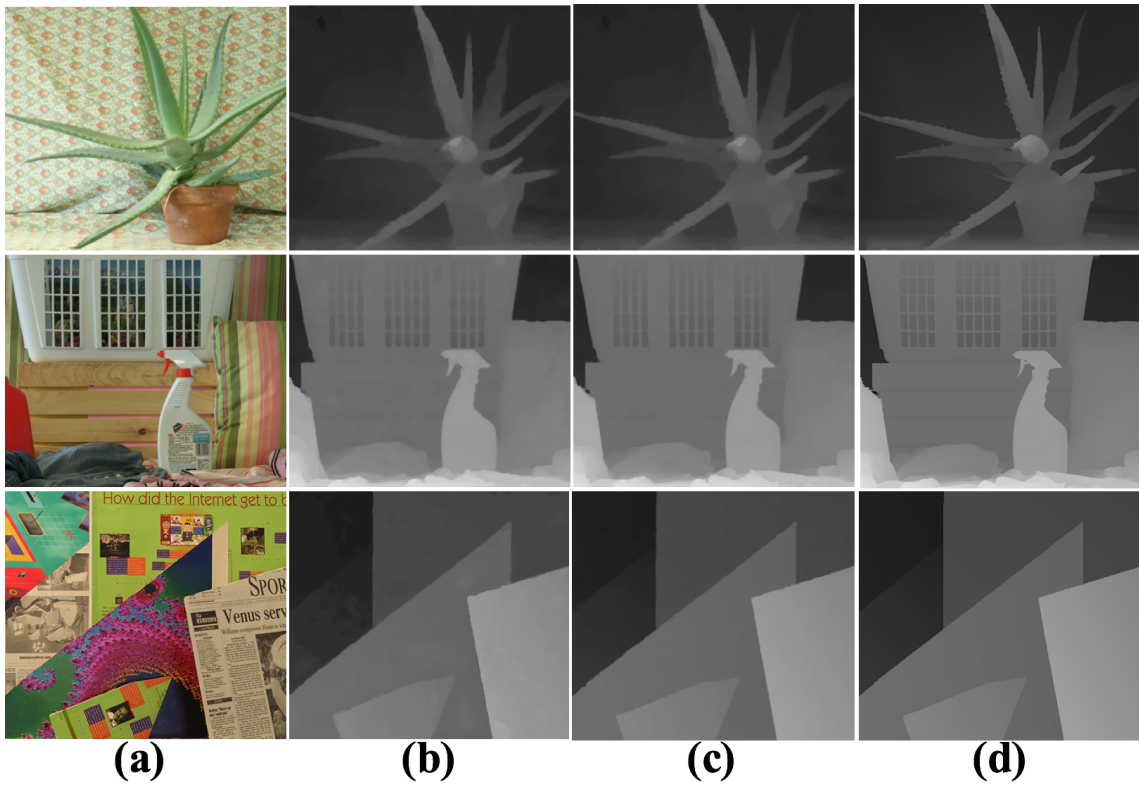


Fig. 5.7. Middlebury results 2 (a) In-focus image (b) EDfD (use EM/MPM) (c) EDfD (use Graph-Cut) (d) Ground truth

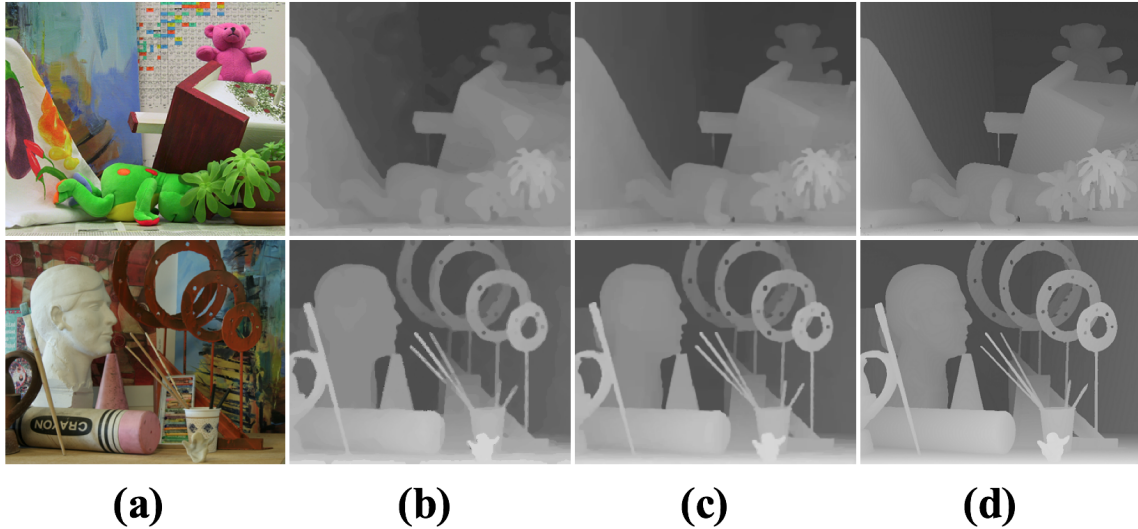


Fig. 5.8. Middlebury results 3 (a) In-focus image (b) EDfD (use EM/MPM) (c) EDfD (use Graph-Cut) (d) Ground truth

As was presented in the previous section, we again use the Root Mean Square Error (RMSE) of the to evaluate the calculated depth map against the ground truth. The updated Table 5.2 and Figure 5.9 compare the proposed EDfD(Graph-cut) results to the results using EDfD(EM/MPM) and other methods. Eight sample images are again compared from the Middlebury dataset: Aloe, Art, Baby, Books, Doll, Laundry, Poster and Teddy. Besides EDfD(EM/MPM), Simulated Annealing (SA), Shape from Defocus method (SFD), CME (Color plus the EM/MPM) and GME (Gray plus EM/MPM) are illustrated. From Table 5.2 and Figure 5.9, it is shown that for each test image, the proposed EDfD(Graph-cut) method achieves the most accurate results. While the average RMSE for EDfD is 2.773, which indicates the error rate is about $3.543/256=0.0138$. The average accuracy is 98.62%.

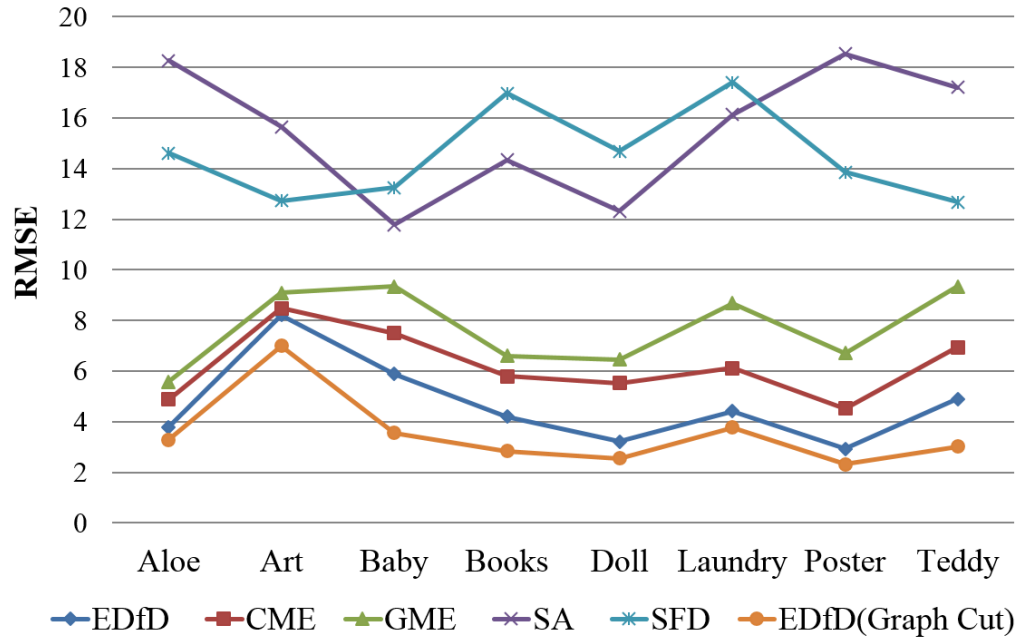


Fig. 5.9. Graph-cut results Comparison with other methods

Table 5.2
Graph cut experimental results comparison, RMSE

Image	GRAPH CUT	EDFD	CME	GME	SA	SFD
Aloe	3.2924	3.7825	4.8843	5.5787	18.2668	14.6188
Art	6.9989	8.2107	8.4762	9.0997	15.6589	12.7344
Baby	3.5495	5.8874	7.4889	9.3466	11.7794	13.2551
Books	2.8431	4.1916	5.7961	6.6030	14.3384	16.9813
Doll	2.5582	3.2111	5.5235	6.4619	12.3158	14.6848
Laundry	3.7763	4.4057	6.1087	8.6766	16.1305	17.4083
Poster	2.3128	2.9186	4.5216	6.7107	18.537	13.8500
Teddy	3.0151	4.8989	6.9491	9.3475	17.2097	12.6796

5.4 Summary

In this chapter, several results of EDfD algorithm with different regularization methods are illustrated. By comparing with some other DfD methods, the new EDfD method using EM/MPM or Graph cuts has much better performance. However, the examples introduced in this chapter are all synthetic images, so in next chapter, a real lens and camera system is used, and the affect the EDfD performance under various impairments will be discussed.

6. REAL LENS/CAMERA SIMULATION

Since the accuracy of the proposed EDfD method was refined by using synthetic images; the next step is to verify that a camera system can achieve the same quality result. The main blocks of digital camera system are shown in Fig 6.1. A scene reflects the light towards the camera, the lens in the camera focuses the light to the image sensor that captures the light information and converts it into digital signals. Finally, the image processing pipeline (ISP) is used to get a high quality digital image.

The EDfD algorithm could be influenced by several parts of this process, such as the lens, sensor, and ISP. The simulation in this chapter will include the effects from lens distortion, relative illumination and optical blur. Also sensor noise, sensor resolution and illumination are performed in the sensor simulation section.

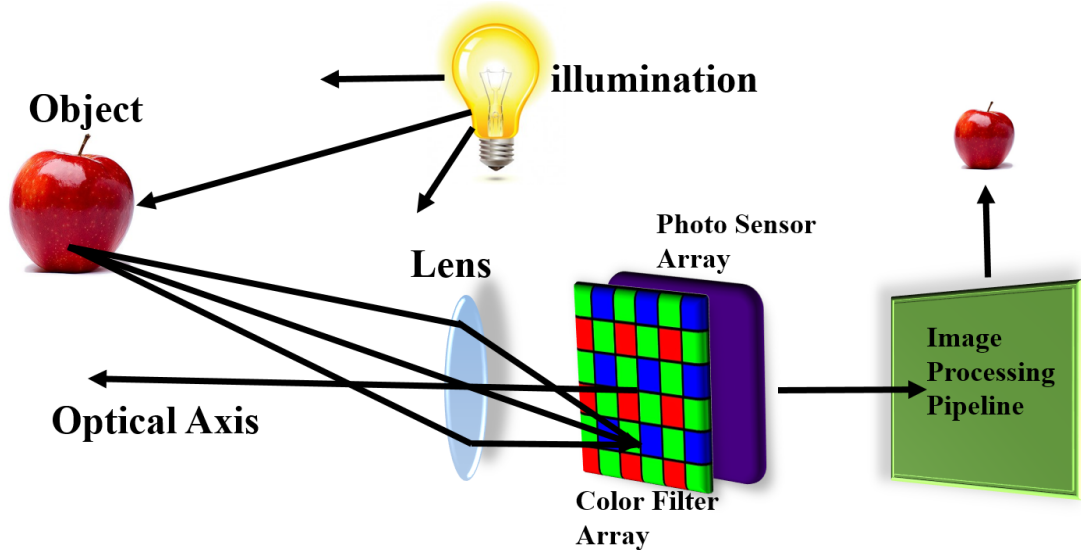


Fig. 6.1. Physical image formation process

6.1 Lens simulation

Ideally, for a perfect lens system, if the light rays from the same object point, they could converge to the one point in the image plane. However, the lens sometimes is not perfect and could cause focus errors. This phenomenon is called lens aberration. In this section, three types of lens aberrations will be described: spherical aberration, coma, and distortion.

6.1.1 spherical aberration

Spherical aberration is one common lens aberration. This kind of lenses has spherical surfaces that the parallel light rays cannot converge to the same point. As introduced in [34], Figure 6.2(a) shows 4 dots in-focus with no aberration. Figure 6.3(b) shows these dots at in-focus position but has spherical aberration.

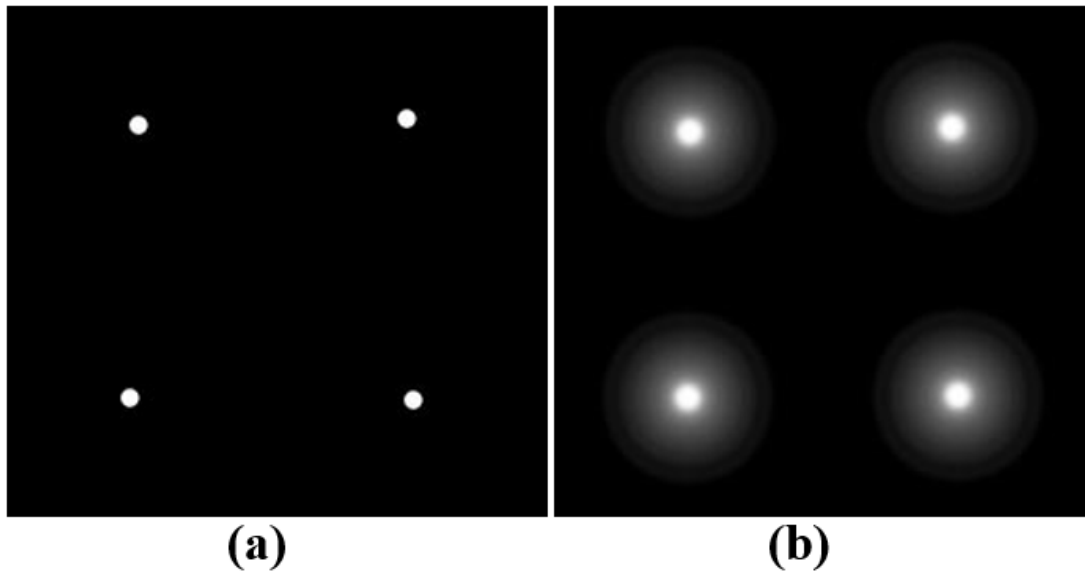


Fig. 6.2. Spherical aberration example (a) No aberration (b) Spherical aberration [34]

In the ideal lens case, all the parallel rays should focus to same distance. However, if the lens has a spherical surface, as shown in Figure 6.3, the light rays further away

from optic axis will have a shorter focus distance. Similarly, the light rays closer to optic axis will have a further focus distance compared with the accurately focused point.

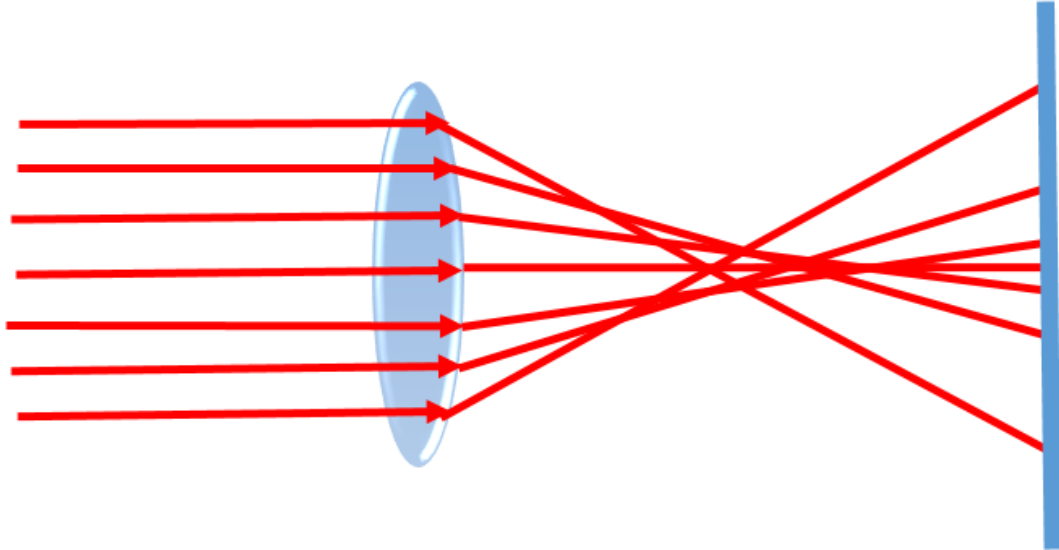


Fig. 6.3. Spherical aberration

As discussed in the previous chapter, the optical system can also be described by the point spread function (PSF). However, the PSF varies for each point in space due to optical aberrations. If using $I(x, y)$ to represent the output of an optical system, $I(x, y)$ can be represented by an ideal input image $I_1(u, v)$, convolving a PSF, $P(x, y)$ (Eq. 6.1)

$$I(x, y) = \iint_{-\infty}^{\infty} I_1(u, v) P(x - u, y - v) dudv \quad (6.1)$$

If the optical system has aberrations, the PSF should be spatially varying. And if assuming all objects in a scene have the same depth, the PSF varies only in x, y directions (Eq. 6.2).

$$I(x, y) = \iint_{-\infty}^{\infty} I_1(u, v) \sum_{i=1}^N w_i(u, v) P_i(x - u, y - v) dudv \quad (6.2)$$

Where p_i are the basis PSF, N represents the number of basis PSF and w_i are the corresponding weights.

For a 3D space (2D+depth), the space variant filtering is described as (Eq. 6.3):

$$I(x, y) = \iint_{-\infty}^{\infty} I_1(u, v) \sum_{i=1}^N w_i(x, y, z) P_i(x - u, y - v) dudv \quad (6.3)$$

This new PSF is also dependent on a third variable z , which represents the depth value.

Based on Eq.6.2, a new all-in-focus image with spherical aberration is simulated as Fig. 6.5(a). Fig. 6.5(b) shows the defocus image generated by using Eq.6.3. Fig. 6.5(c) is the EDfD depth map result calculated based on (a)(b) and has a strong effect from the lens aberration.



Fig. 6.4. spherical aberration result (a) In-focus image (b) Defocus image (c) Depth map

However, it is still possible to fix the depth map error. If the point spread function in Eq.6.2 can be calculated or inferred, an accurate depth map can still be achieved. For a fixed, known lens, this can be calculated and compensated for. This compensation is shown in Fig. 6.5. As in the previous uncompensated example; (a) and (b) are in-focus and defocus images respectively, and are both affected by spherical aberration. The compensated method is used, and (c) is depth map result by using EDfD method with a known aberration-PSF. (d) is depth map results calculated from

the input pairs without spherical aberration. Since (c) and (d) are equivalent, it is concluded that accurate depth map results can be achieved based on a spherically compensated known PSF.

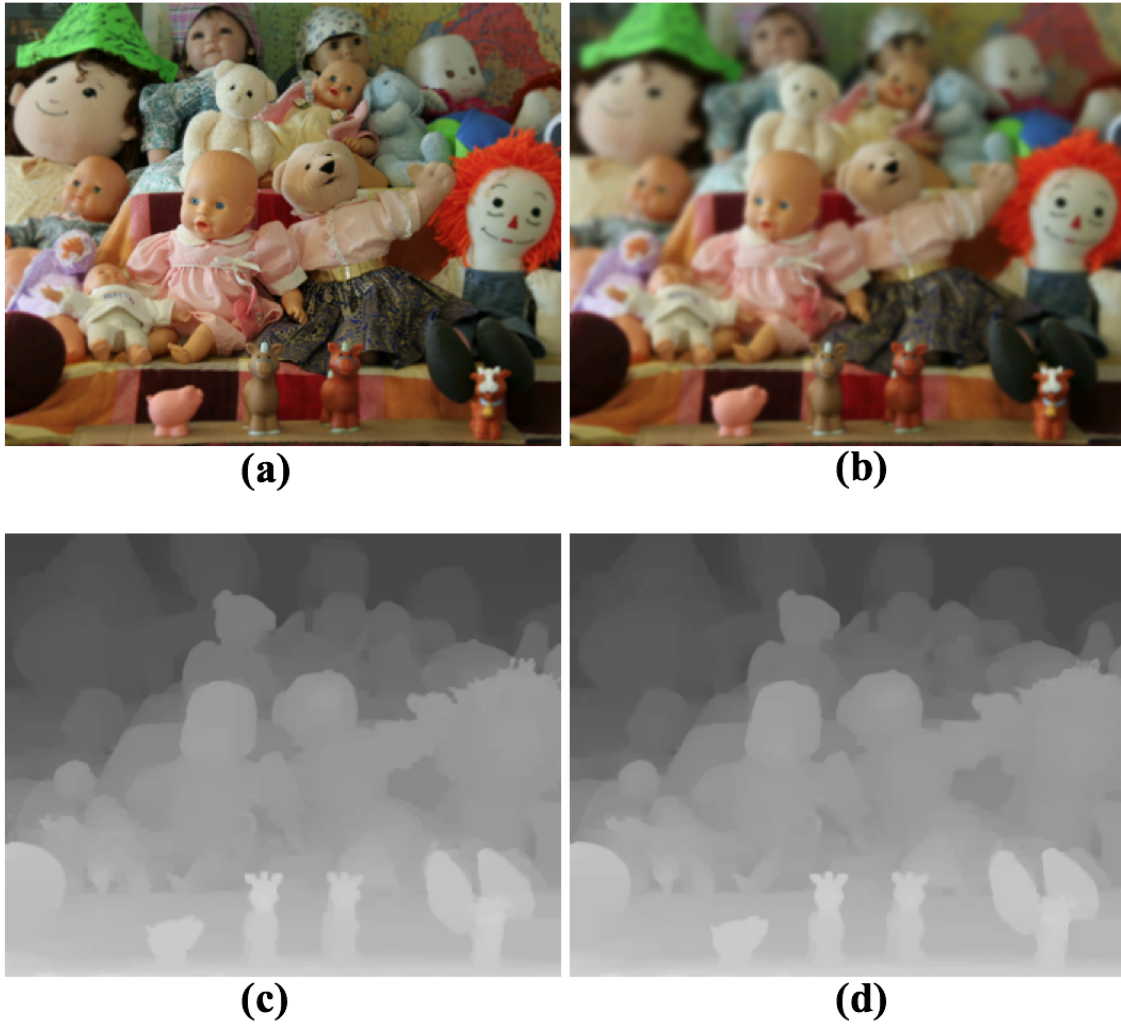


Fig. 6.5. spherical aberration result (PSF known) (a) In-focus image (b) Defocus image (c) Depth map result by using a known PSF (d) Depth map result (No aberration)

For the lens, an additional optical component can be used to reduce the spherical aberration. For multiple lenses, some lens elements like symmetric doublets could be applied to eliminate the spherical aberration.

6.1.2 Coma

Similar to the spherical aberration, coma is also a common aberration but caused by off-axis light rays. A lens with a large coma could generate a sharp image at the field center, and a more blurred image near the edge locations.

As introduced in [34], Figure 6.6(a) shows 4 dots in-focus with no aberration. Figure 6.3(b) shows these dots at in-focus position but has coma aberration.

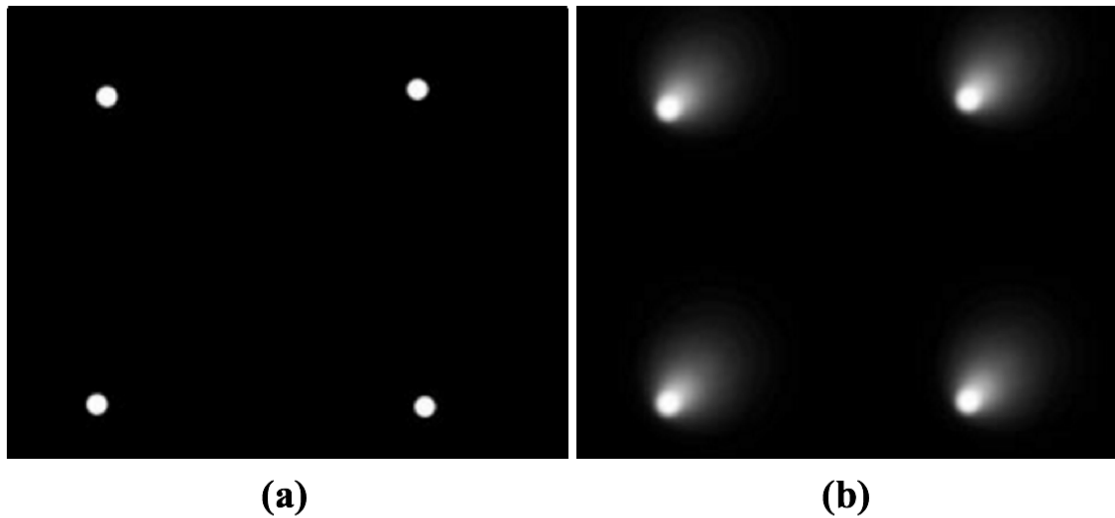


Fig. 6.6. Coma illustration (a) No aberration (b) Coma aberration [34]

Fig 6.7 shows how light rays could be affected by a lens with coma. Especially the off-axis light rays, passing through the lens, finally focus on the image plane with different sizes of circles and project at slightly different positions.

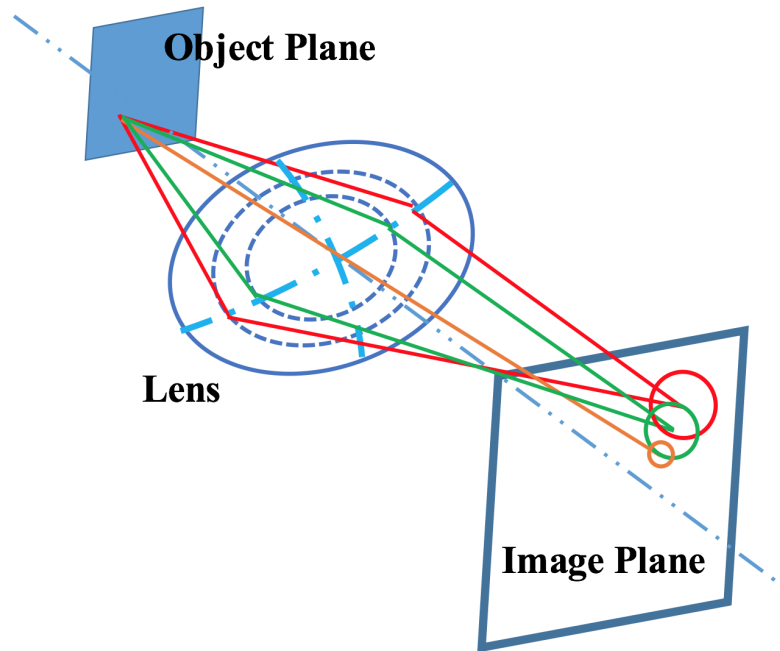


Fig. 6.7. Coma aberration [35]

Similar to the spherical aberration, an accurate depth map can be generated if the PSF is clear. Coma can be corrected by bending the light using added lens element for a single lens. Also combining the symmetric lenses could achieve a better creation which is a better solution to solve coma problem.

6.1.3 Distortion

Lens distortion does not change the color or the sharpness of the image but its shape. Here are two types of distortion: barrel distortion and pincushion distortion [36] (Fig. 6.8).

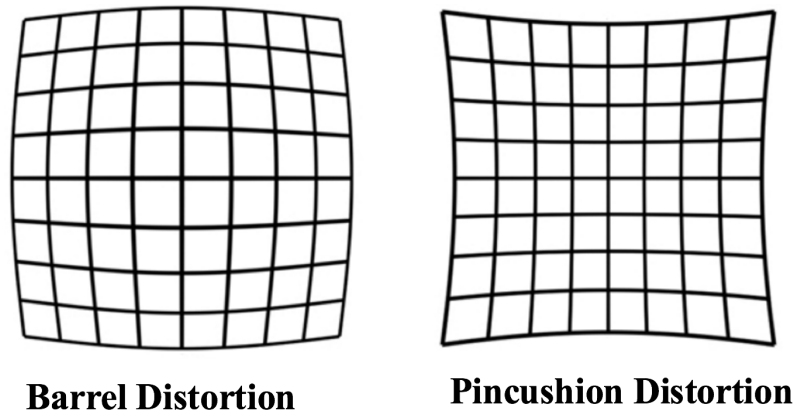


Fig. 6.8. Distortion [36]

Here barrel distortion is used as an example. As shown in Fig. 6.9, the object is placed at out of focus position. If the lens has no distortion, light rays (red lines) stop at the lens position and converge at point A in the virtual image plane. If the lens has barrel distortion, light rays (purple lines) first stop in front of the lens, and then converged at point B in the virtual image plane which locates closer to the spindle.

In image plane, due to lens theory, the object at out of focus position will be a blur disc for both of these two situations. Based on math geometric theory, R' will be larger than R which means the object appears stronger blurry in image plane if the lens has barrel distortion. According to 3.4, the radius of blur disc is proportional to depth. So barrel distortion of the lens could lead to error estimation of depth.

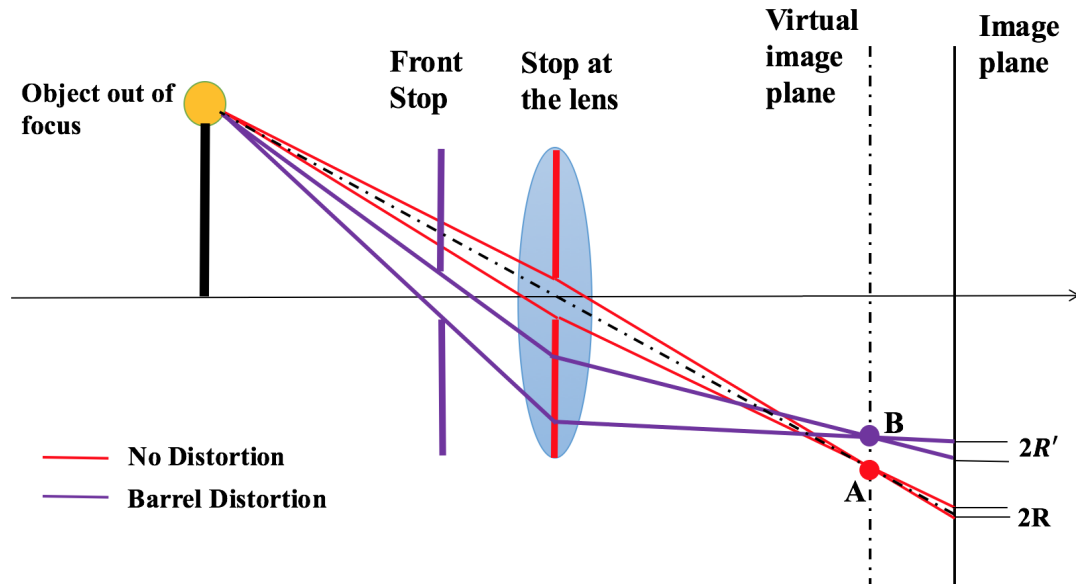


Fig. 6.9. Barrel Distortion

Fig. 6.10 shows an example of calculating depth map when the lens has barrel distortion. (a), (b) show in-focus image and defocus image with barrel distortion. (c) is the depth map calculated by (a) and (b). (d) represents ground truth of depth map. Comparing (c) and (d), depth map errors are demonstrated, affected by barrel distortion.

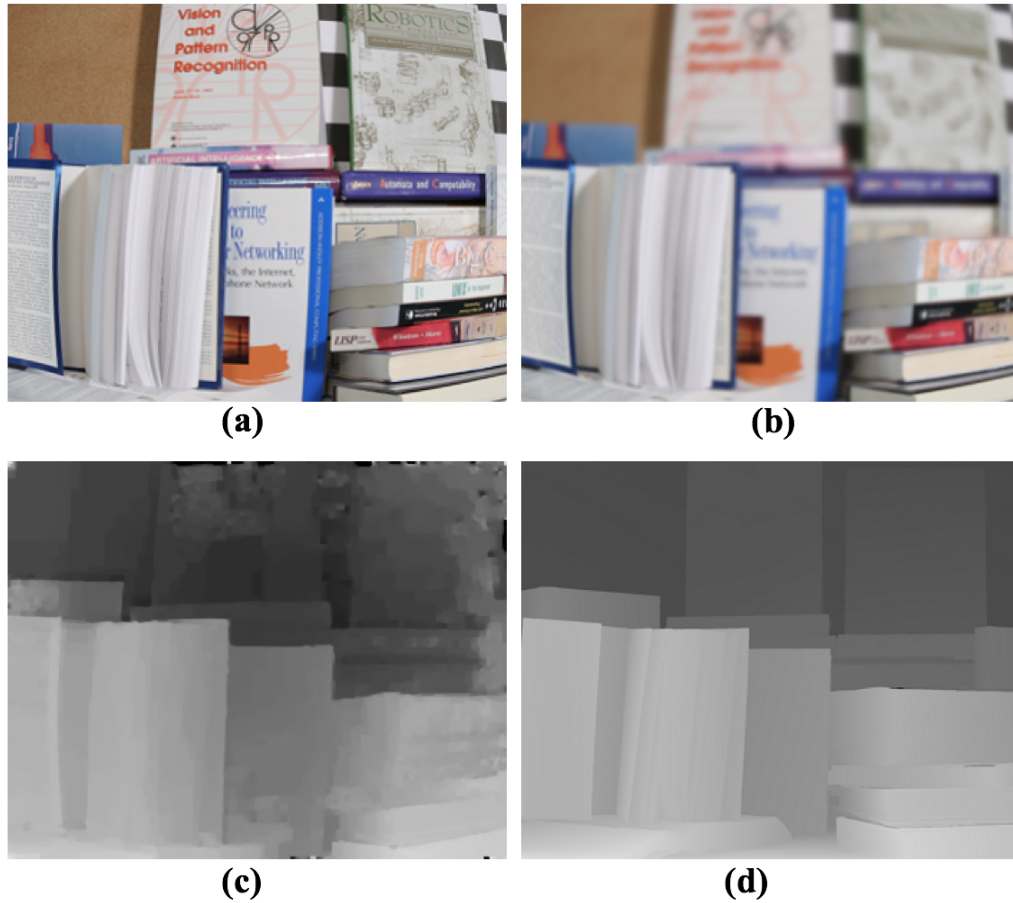


Fig. 6.10. Depth map result with Barrel Distortion (a) In-focus image (b) Defocus image (c) Depth map result(with Barrel Distortion) (d) Depth map result(without distortion)

In order to improve the quality of depth map, it is important to minimize these lens distortions. One way is to use the optical methods, as is suggested above with the other lens aberrations. Another way is to use the image processing tools of camera calibration for correction. Two methods have been explored in this thesis: Correction-first and EDfD-first. Correction-first is correcting the barrel distortion for in-focus and defocus images first, then EDfD is used to generate depth map; EDfD-first is correcting the barrel distortion of depth map directly, from the distorted input pairs. An example is shown in Fig. 6.11. Examples (a) and (b) are in-focus and defocus images after correcting the barrel distortion. (e) is the ground truth of depth

map. (c) and (d) represent depth map images by using Correction-first and EDfD-first respectively. It is shown that the quality of depth map can be improved after correction and Correction-first method is improved over the EDfD-first method.

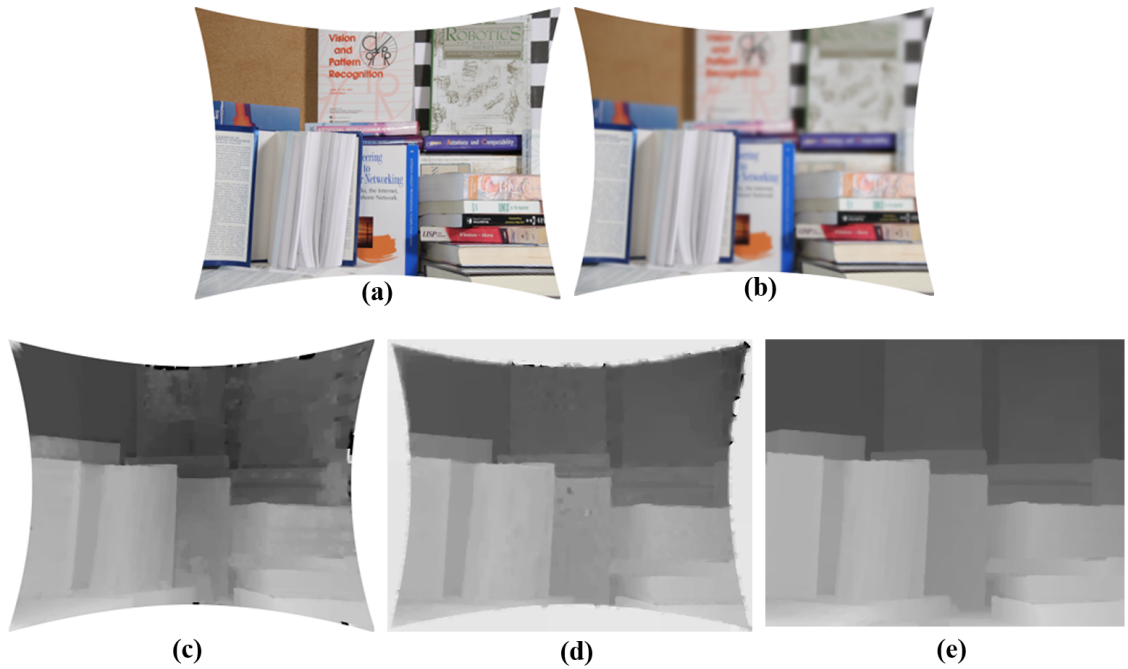


Fig. 6.11. Depth map result after correcting distortion (a) In-focus image after correcting barrel distortion. (b) Defocus image after correcting barrel distortion. (e) is the ground truth of depth map. (c) represents the resulting depth map with the correction-first method, and (d) is the depth map using EDfD-first.

6.2 Simulate Camera digital image processing pipeline

After capturing the light information, the camera converts it into digital signals from the sensor. The image signal processing pipeline (ISP) is used to generate a final digital image output with high quality [37].

While the EDfD algorithm could be affected by several modules on the pipeline. The simulation in this section will include the major effects from sensor noise, illumination and contrast.

6.2.1 Pipeline introduction

An example of typical ISP is shown in Fig. 6.12. For color cameras, the way to get a color image out is to put a filter on top of imaging sensor [37]. Usually a Bayer pattern color filter is chosen. The image sensor does not sense red green and blue for each pixel, it senses one color for each pixel. Then interpolation is needed to generate the color information of the pixels by using adjacent pixels. This is called demosaicing, and it is the primary job of ISP. In addition, the ISP also controls autofocus, exposure, and white balance for the camera system. Things like noise reduction, color correction, gamma correction, edge enhancement, contrast enhancement, and conversion between color spaces etc are also included. Recently, correcting for lens imperfections like vignetting or color shading coming from the imperfect lens system has been added as well.

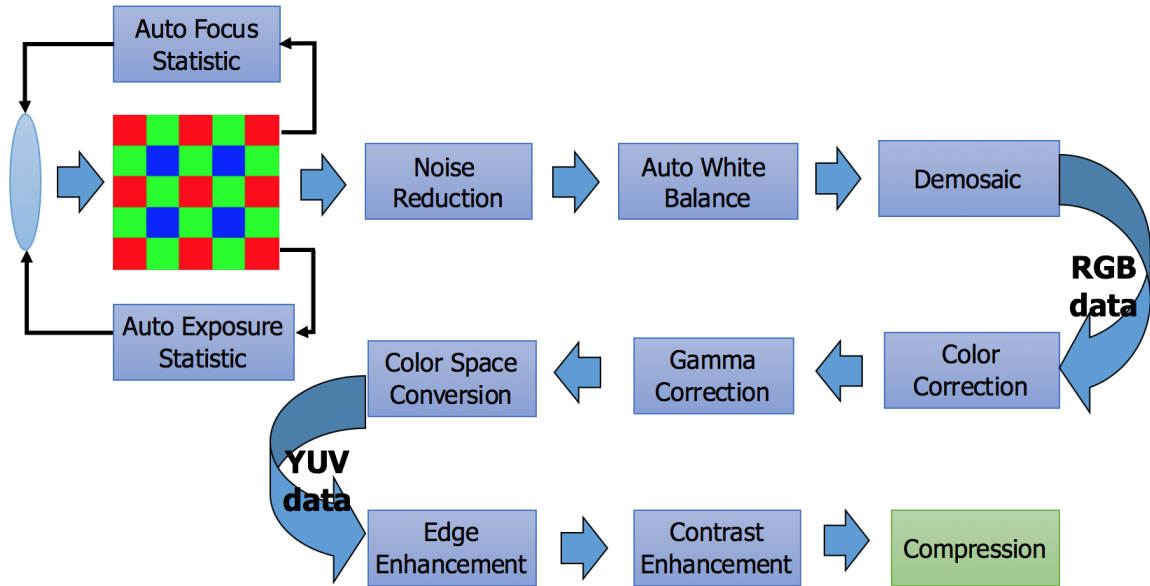


Fig. 6.12. Image signal processing pipeline (ISP)

6.2.2 Noise

CMOS image sensors are widely used in the market. However, the images captured from CMOS image sensors could contain noise, especially under low light conditions. In order to test the EDfD with noise, we introduce the following noise models. Generally, here are two types of noise from CMOS image sensor: fixed-pattern noise (FPN) and temporal random noise. FPN is easy to eliminate because it has the same spatial location frame to frame. However, temporal random noise is known as photon shot noise, and it is much more difficult to remove. Usually it can be approximated by the Gaussian distribution [38]. A special additive White Gaussian Noise (AWGN) model is used to describe it in [38]. This results in a standard deviation of temporal noise that is proportional to pixel intensity: the higher intensity value, the larger standard deviation of noise [39].

As presented in [39], a noisy pixel can be noted by Equation 6.4:

$$g = f + f^\gamma \cdot u + v \quad (6.4)$$

Where g is the pixel with noise, f is pixel without noise. u and v are zero-mean random variable with variances σ_u^2 and σ_v^2 . So the standard deviation of the noise can be expressed as [39]:

$$\sigma^2 = f^{2\gamma} \cdot \sigma_u^2 + \sigma_v^2 \quad (6.5)$$

Based on the suggestion from [39], γ is set to 0.5. So the Equation 6.5 is rewritten to Equation 6.6. In this equation, the noise variance is linearly related to the pixel intensity value.

$$\sigma^2 = f \cdot \sigma_u^2 + \sigma_v^2 \quad (6.6)$$

Fig. 6.13 illustrates an example of images with intensity-dependent noise added to them. σ_v^2 is set as 10^{-4} and σ_u^2 is set as 6×10^{-3} . Fig. 6.13(c) and (d) are in-focus and defocus images without noise respectively; (a) is an in-focus image with intensity-dependent noise; (b) shows a defocus image with intensity-dependent noise as well.

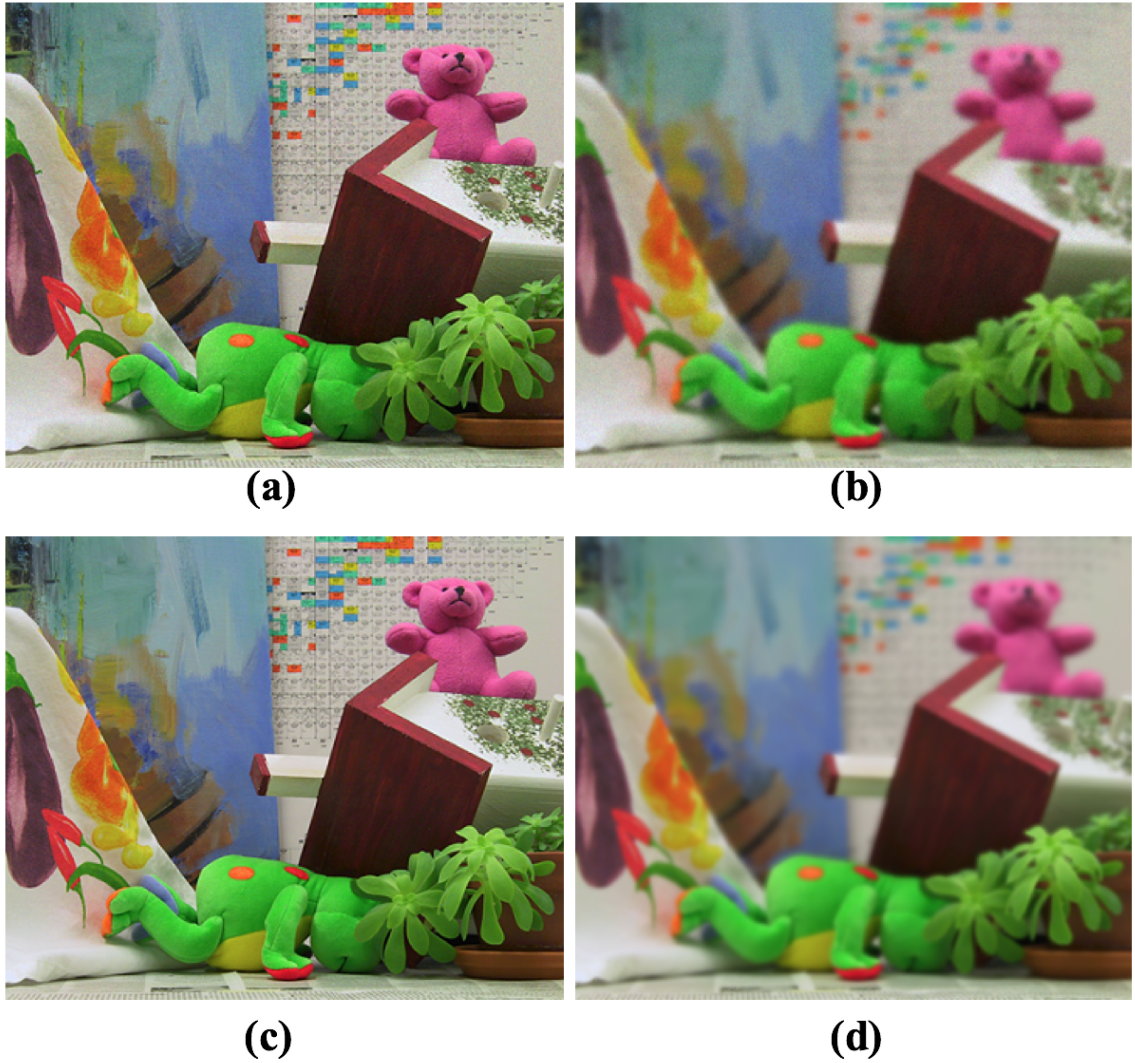


Fig. 6.13. Intensity dependent noise (a) In-focus image with intensity-dependent noise (b) Defocus image with intensity-dependent noise (c) In-focus image without noise (d) Defocus image without noise

By using Fig. 6.13(a) and (b) as input for EDfD algorithm, depth map result is shown in Fig. 6.14(a). Compared with depth map result without noise effect and Ground truth of depth map in Fig. 6.14(b) and (c) respectively. This example shows that intensity-dependent noise will highly affect EDfD result. This is seen especially in "white" regions (e.g. lower left corner of Fig. 6.13(a) and (b)), where the pixels

there have large intensity values and have stronger noise. This leads to large mistakes in the calculated depth map (e.g. lower left corner of Fig. 6.14(a) and (b)).

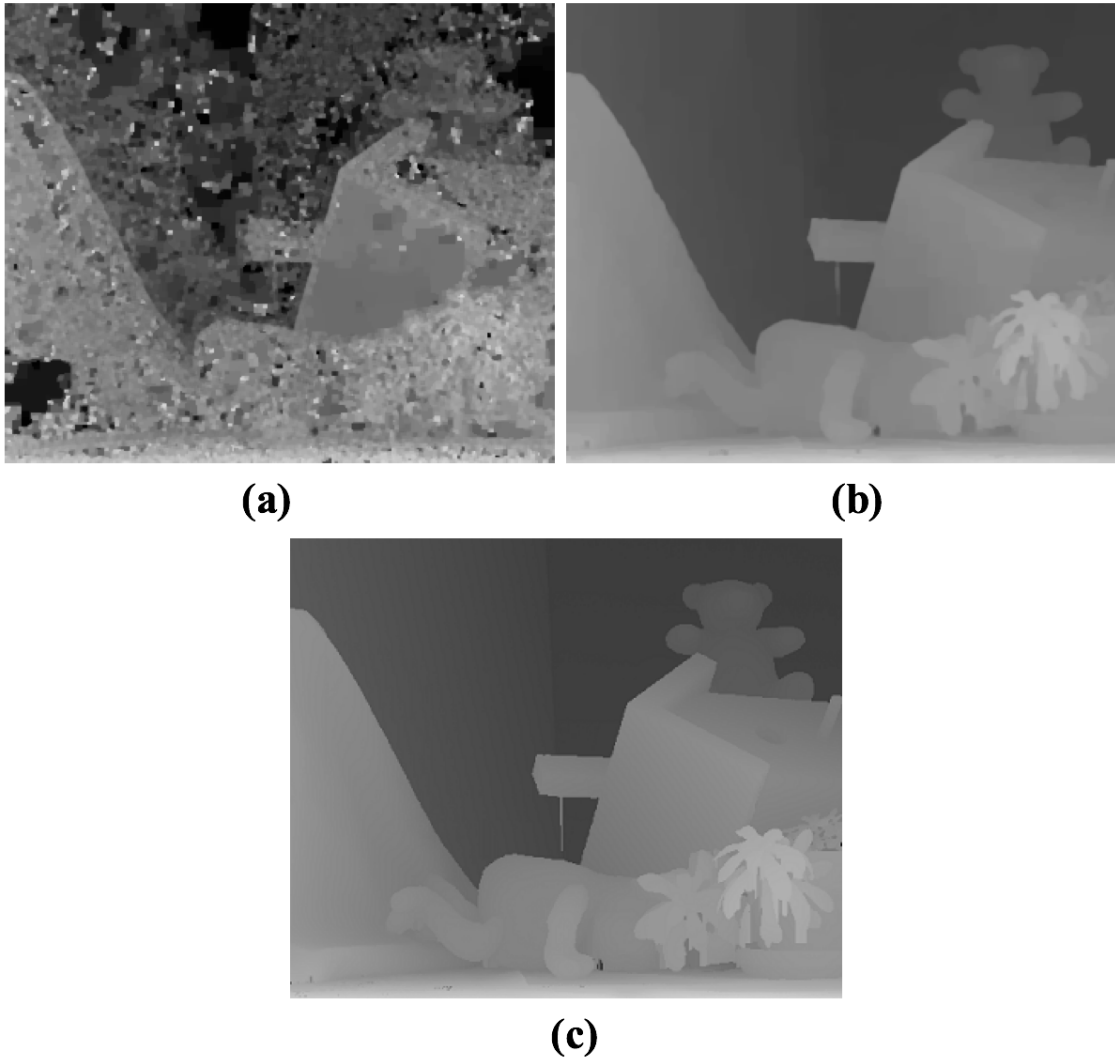


Fig. 6.14. EDfD example result with noise effect (a) EDfD result using noisy inputs (b) EDfD result using noise-free inputs (C) Ground truth

Figure 6.15 shows RMSE of the calculated depth map against the ground truth, The updated Table 6.1 and Figure 6.15 compare the EDfD(Graph-cut) results using noise-free inputs to the results using noisy inputs. Eight sample images are still

compared from the Middlebury dataset: Aloe, Art, Baby, Books, Doll, Laundry, Poster and Teddy. From Table 6.1 and Figure 6.15, it is shown that for each test image, the proposed EDfD(Graph-cut) method is highly affected by noise. And using Middlebury dataset Teddy as an example, refer to Eq. 6.6, fixing σ_v^2 and changing σ_u^2 from $0.1 * 10^{-3}$ to $8 * 10^{-3}$, the RMSEs are increased from 7.3248 to 29.1242 as shown in Figure 6.16.

It is shown that sensor noise has a significant effect on EDfD performance. Spatial or temporal noise reduction methods will be developed in future research.

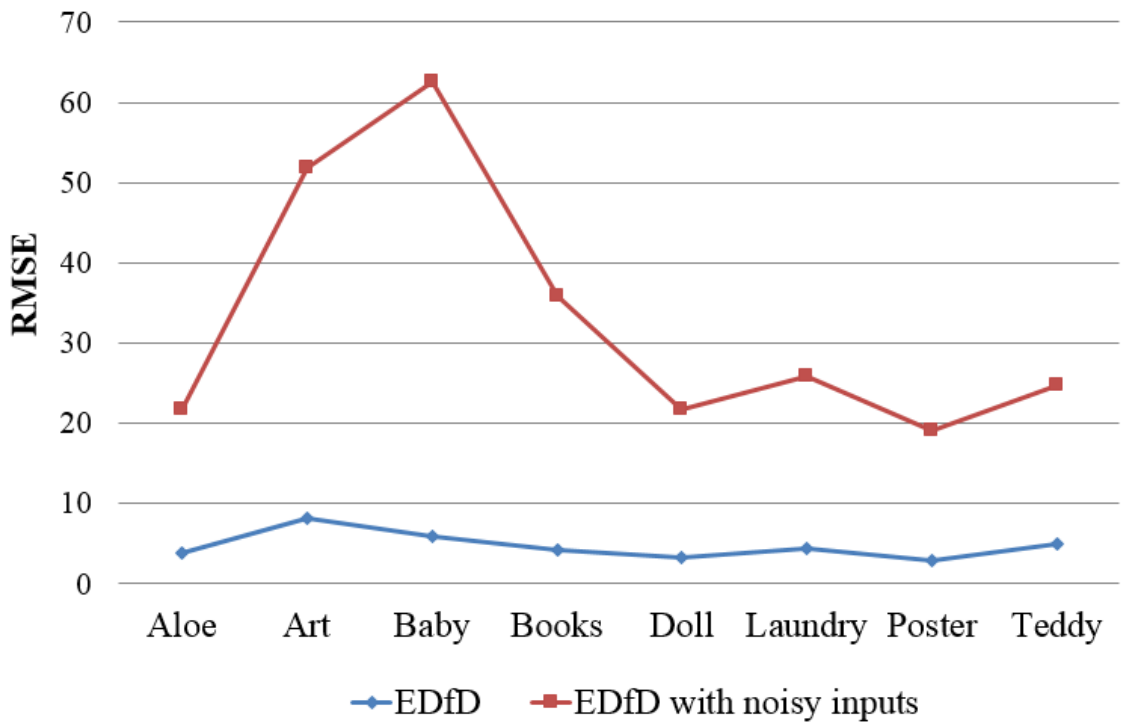


Fig. 6.15. Middlebury EDfD result with noise effect

Table 6.1
Noisy inputs experimental results comparison, RMSE

Image	EDFD(GRAPH CUT)	EDFD(GRAPH CUT)+NOISE
Aloe	3.2924	21.6847
Art	6.9989	51.8614
Baby	3.5495	62.6769
Books	2.8431	35.8835
Doll	2.5582	21.8150
Laundry	3.7763	25.8648
Poster	2.3128	19.0883
Teddy	3.0151	24.6516

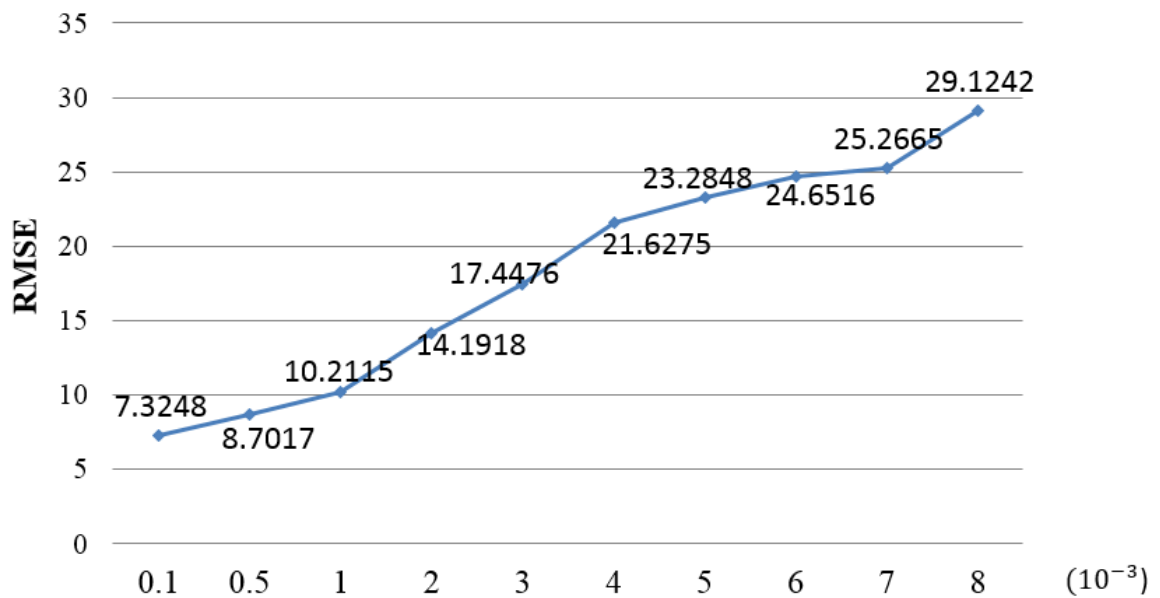


Fig. 6.16. Teddy with noise effect, RMSE

6.2.3 Illumination and Contrast ratio

For the EDfD method, the in-focus and defocus image are captured at different times. In order to avoid getting different gain for each pair, auto exposure function will be inhibited, at least during the pair's acquisition. So the exposure time and gain will be set as fixed number. The contrast ratio of output image will only be affected by the illumination of the scene. To better understand the effects of image contrast ratio, the several Middlebury images are chosen from different exposures but the same illumination to understand the brightness effect on the EDfD algorithm performance..



Fig. 6.17. EDfD example result under different Illumination

Fig. 6.17 represents EDfD results under different exposure times but same illumination. The image exposure data are sourced from Middlebury. Each row shows one type of contrast. From top to bottom, the exposure time are 4000ms, 1000ms and 250ms respectively. From left to right, each column shows in-focus image, defocus image and depth map EDfD result respectively.

It is shown that under low illumination, the EDfD result will be worse than normal and high. However, the error rates do not increase more than 35%. This confirms that the exposure difference (and illumination difference) will not have a strong effect on EDfD algorithm compared with the effect of noise.

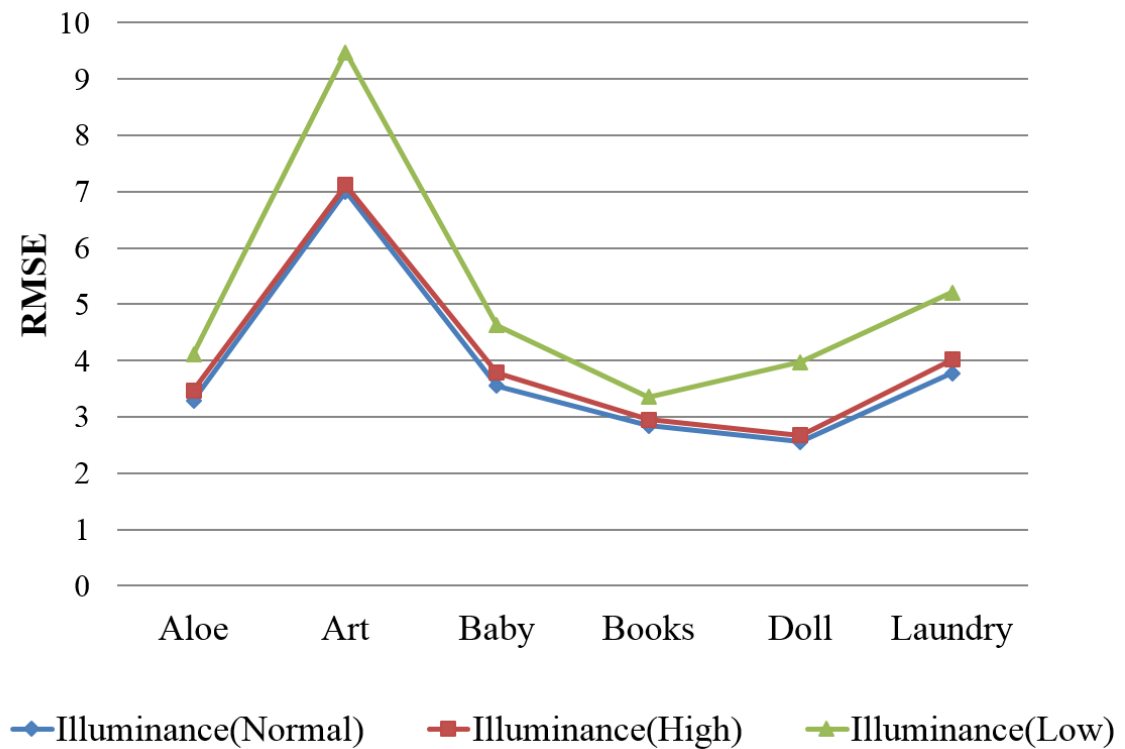


Fig. 6.18. RMSE of EDfD example results under different exposures

6.2.4 Resolution effects

The camera sensor usually has different resolution settings (e.g. 1280x720, 640x360) which lead to output images having different size. Refer to Eq. 3.8, if using the same camera and lens, camera sensor's width will be a fixed number, and for the same depth, the radius of defocus blur in millimeters will not change. So the radius, R , in pixels will change with the same scale factor as the changes of image width or height when changing image resolution (assuming image width and height are changed using the same scale factor). And regarding Eq. 3.7, the 2D Gaussian blur σ should also be changed with the same scale number.

For example, for one defocus image, the resolution is 1280x720 and the maximum σ corresponding to the largest depth value is 3. If the image resolution is reduced to 640x360, the maximum σ should be 1.5.

This can be demonstrated as shown in Figure 6.19. In Figure 6.19, (a), (b) and (c) are original size in-focus image, defocus image and corresponding depth map calculated by proposed EDfD method, respectively. (d) and (e) are the half-size version of (a) and (b) that both width and length of images are one-half of original ones. These images are scaled by using Bicubic interpolation [40]. (f) shows the EDfD result by using (d) and (e) as input, and the maximum σ is set as one-half of the value used for original size. As is shown, the depth map result has the same quality as original one. So resolution does not have a strong effect on the RMSE of EDfD algorithm.

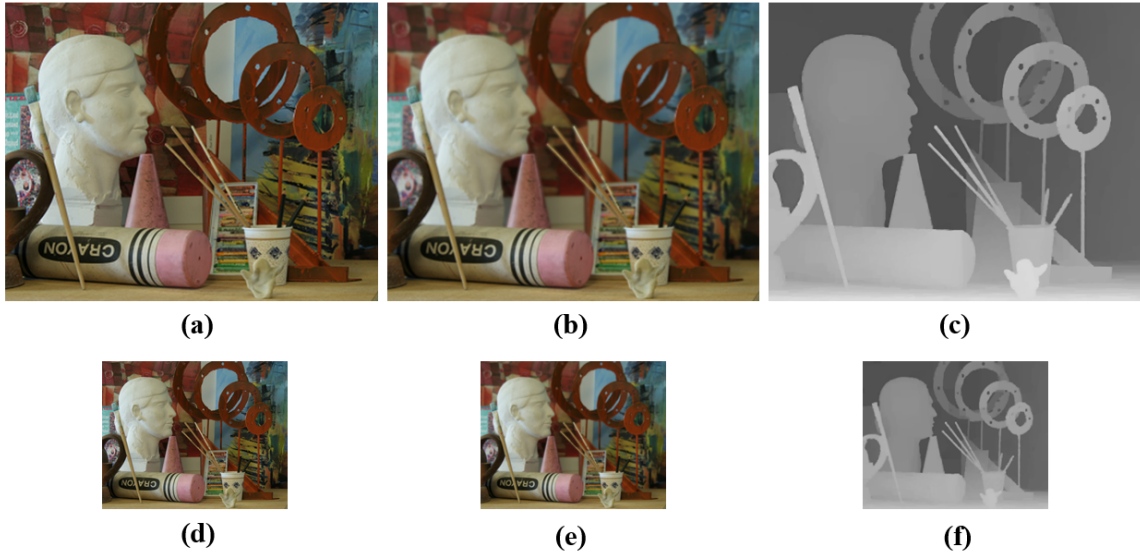


Fig. 6.19. Resolution effects (a) In-focus image (original size) (b) Defocus image (original size) (c) EDfD result (use (a) and (b)) (d) In-focus image (half size) (e) Defocus image (original size) (f) EDfD result (use (d) and (e))

6.3 Summary

This chapter discussed several important impact factors in real Lens and camera systems which affect the accuracy of EDfD result. For the lens: if it has lens aberrations like spherical aberration, coma, and distortion, it will affect the EDfD results. However, the known aberrations can be fixed, or the PSF can be calculated using experiments; then the accurate depth map result can still be achieved. For the camera ISP: illumination, contrast ratio, and resolution differences are not the major problems for the favorable EDfD results. However, this research finds that the signal-dependent noise from CMOS image sensor does have a significant effect on EDfD performance. How to reduce the noise while preserving the original image information will be important research in future.

7. EXPERIMENTAL RESULTS FROM CAMERA WITH MICROFLUIDIC LENS

Since the accuracy of the proposed EDfD method was refined by using synthetic images; the next step is to verify that a camera with a microfluidic lens can achieve the same quality result. A single imager with a fast-focus microfluidic lens is needed. Some focus and optical performance experiments with this lens were introduced in previous papers [41,42].



Fig. 7.1. Single imager system

Figure 7.1 shows the single imager system which is used in this research. The system consists five components: lens focus controller, microfluidic lens, complementary metal-oxide-semiconductor (CMOS) imager, CMOS imager development board and a desktop computer (not shown in Figure 7.1).

The image is formed on the imager, then the camera passes data to development board in real time. There is another board installed in the computer which is connected to the development board. The computer sends commands to lens focus controller. By changing the voltage, the microfluidic lens can change the focus settings and different focus images appear on the imager. Once the system is connected, the video stream is sent to the computer and observed on the monitor.

7.1 Microfluidic lens

In this research, an electrowetting microfluidic lens [12] is used to capture the focused and defocused images in real time. The technology uses the electrowetting principle and transparent liquids to create a lens. The innovation of this technology is the focal length can be fast changed by only adjusting the voltage added on this particular liquid lens.

Fig. 7.2 represents the relationship between effective focal length and voltage based on experiments using CASPIAN C-39N0-16 module which equipped Arctic 39N0 Liquid Lens. Blue dots are results from whole lens module, and orange dots are from liquid lens only. The minimum working voltage for this liquid lens is 42V and effective focal length starts from 16mm - 16.5mm. As is shown, the focal length decreases as the voltage increases.

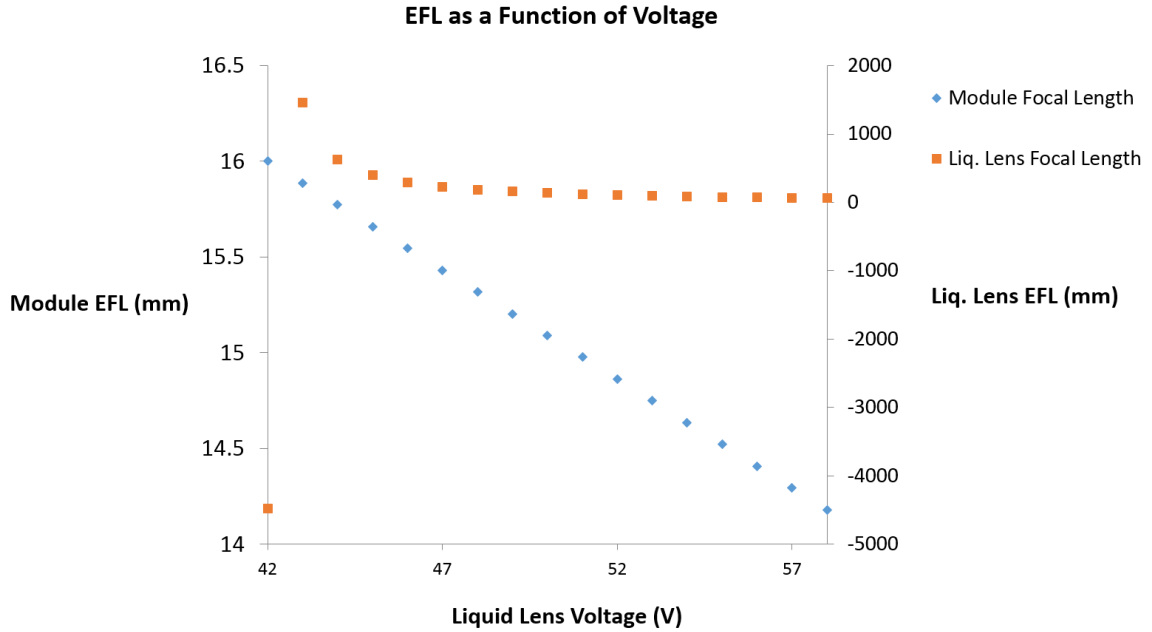


Fig. 7.2. Effective focal length vs liquid lens voltage

This particular lens specification guarantees that the focus can be adjusted continuously up to 60 frames per second. It also has a very fast response time and wide focus range from 10cm to infinity. In this research, the voltages for capturing in-focus and defocus images are 52.4V and 53.1 respectively. Fig. 7.3 shows relationship of optical power (also named diopter) and voltage. Optical power is the inverse of focal length. As is shown in this figure, optical power is linearly relative with voltage 7.1. As voltage is only changed less than 1V, optical power changes around 1 optical power. By using 7.4 as reference, changing 1 optical power corresponds to less than 10ms response time.

$$Optical_Power = Voltage - 42.1 \quad (7.1)$$

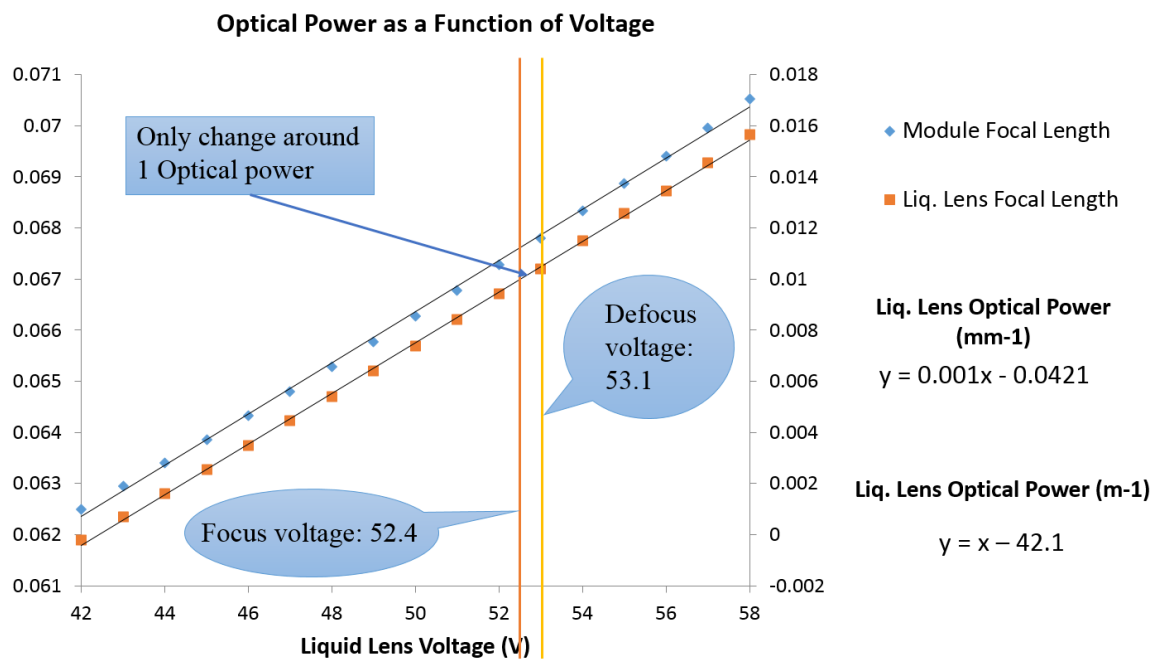


Fig. 7.3. optical power vs liquid lens voltage

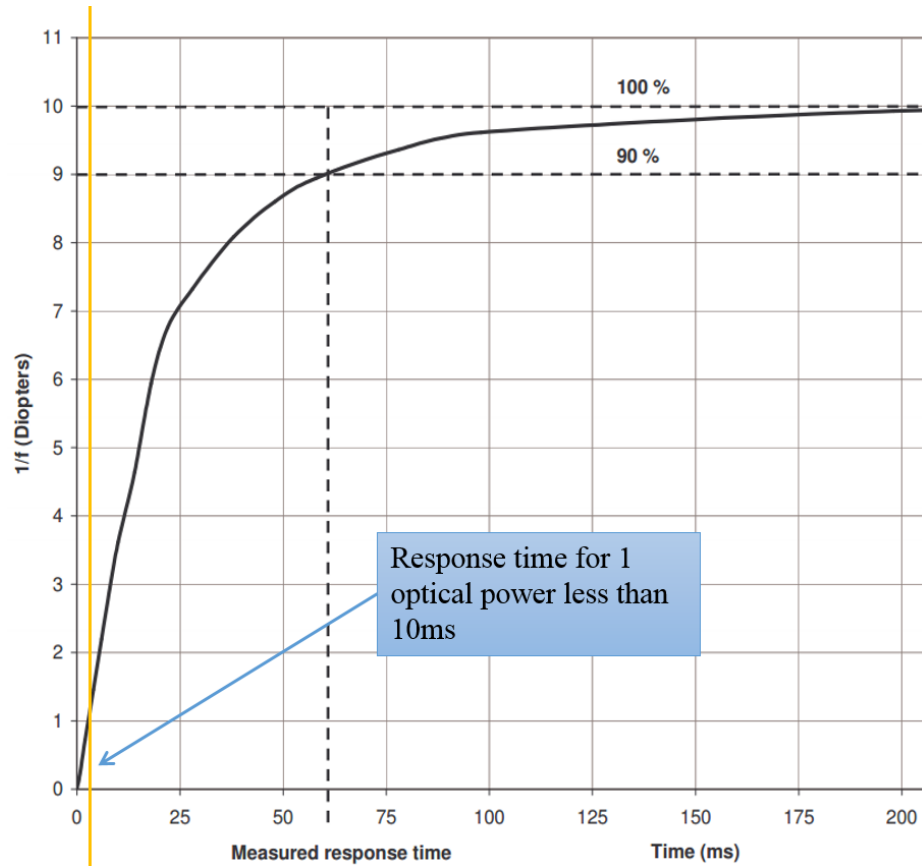


Fig. 7.4. optical power vs response time

7.2 True camera results

By using this single imager system, both still and motion images can be collected. Figure 7.5 to Figure 7.8 show four different collected images captured by this single imager system. In every figure, (a) are in-focus images are captured by the camera. The (b) images are the defocused images which are captured directly by the camera at a different lens voltage. The column (c) are the depth maps generated by the EDfD algorithm. In (d) the 3D view maps of EDfD depth maps can be seen. Finally, in (e) are shown the 3D view maps which are generated by in-focus images and depth maps.

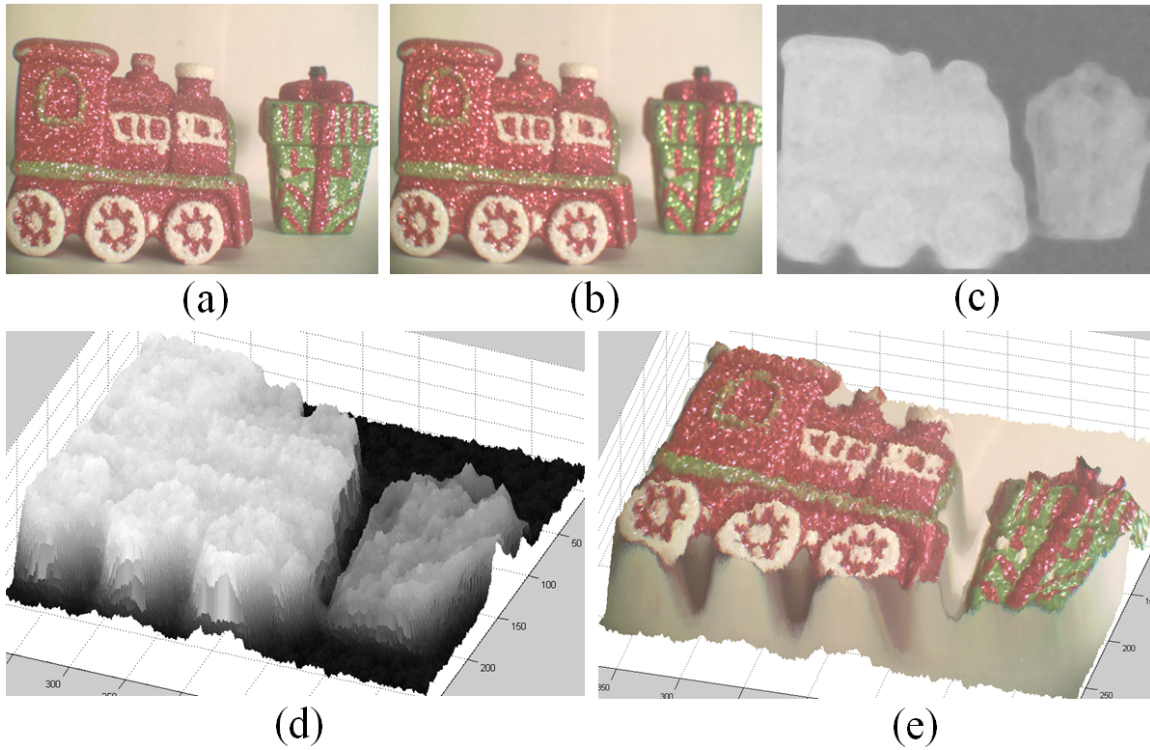


Fig. 7.5. Train and gift box (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image

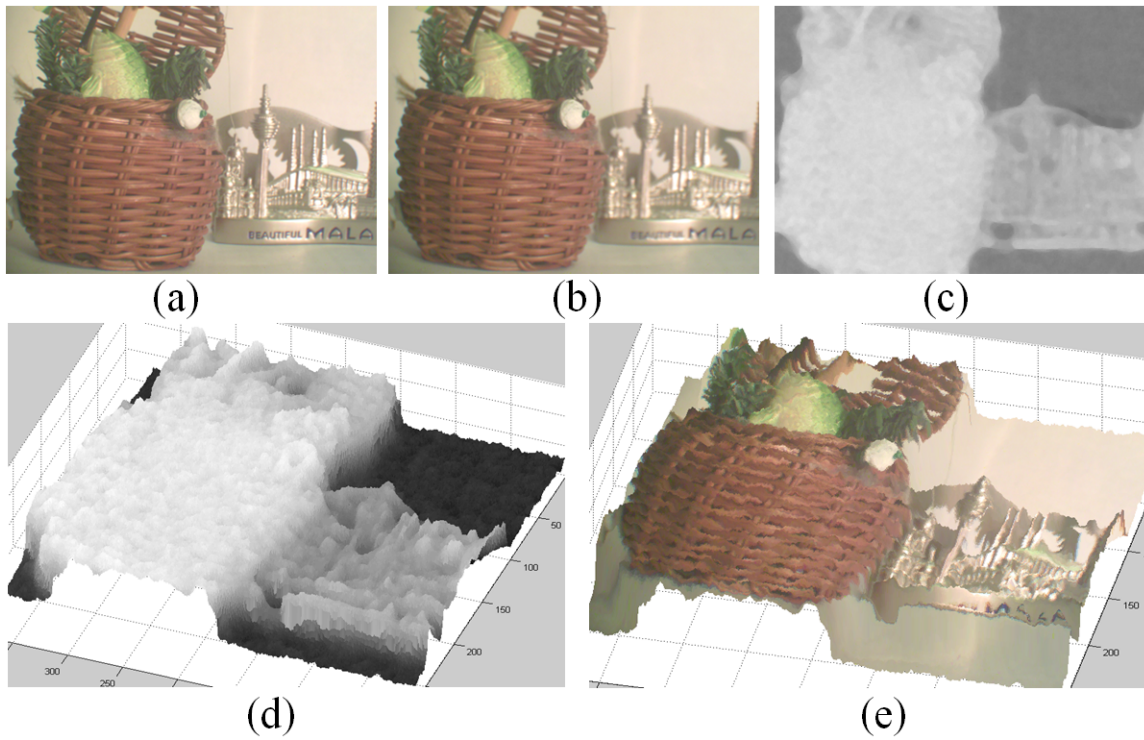


Fig. 7.6. Basket and Malaysia (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image

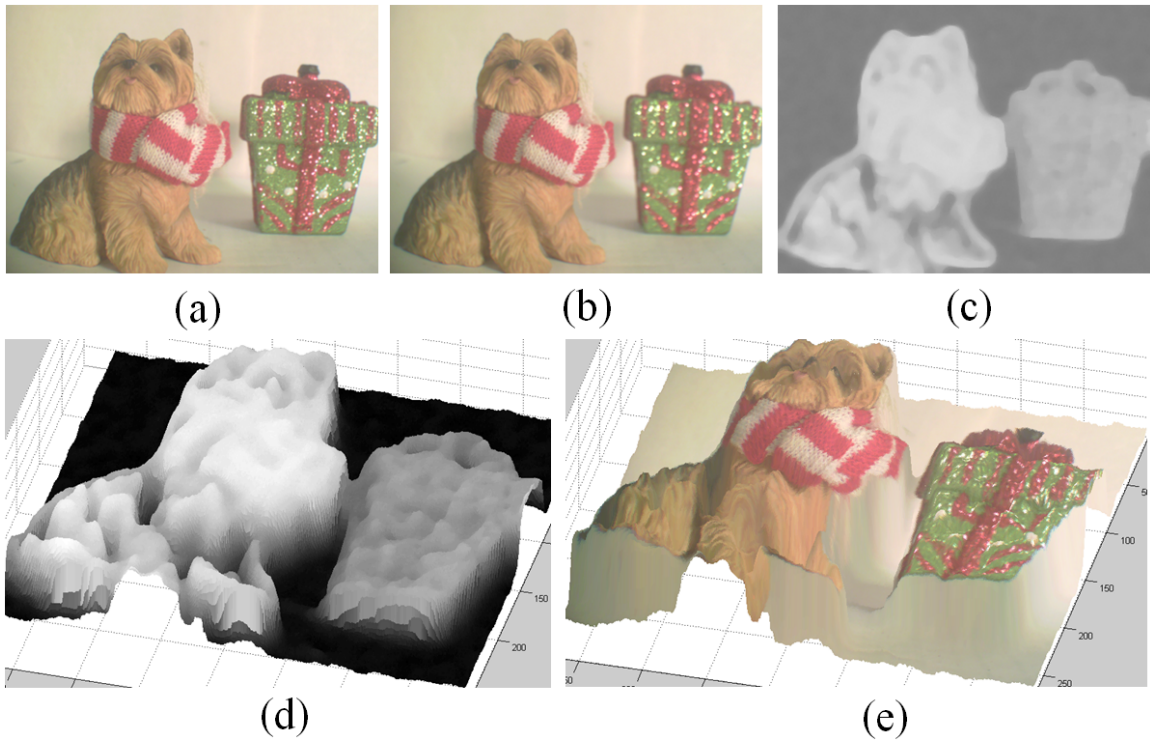


Fig. 7.7. Dog and gift box (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image

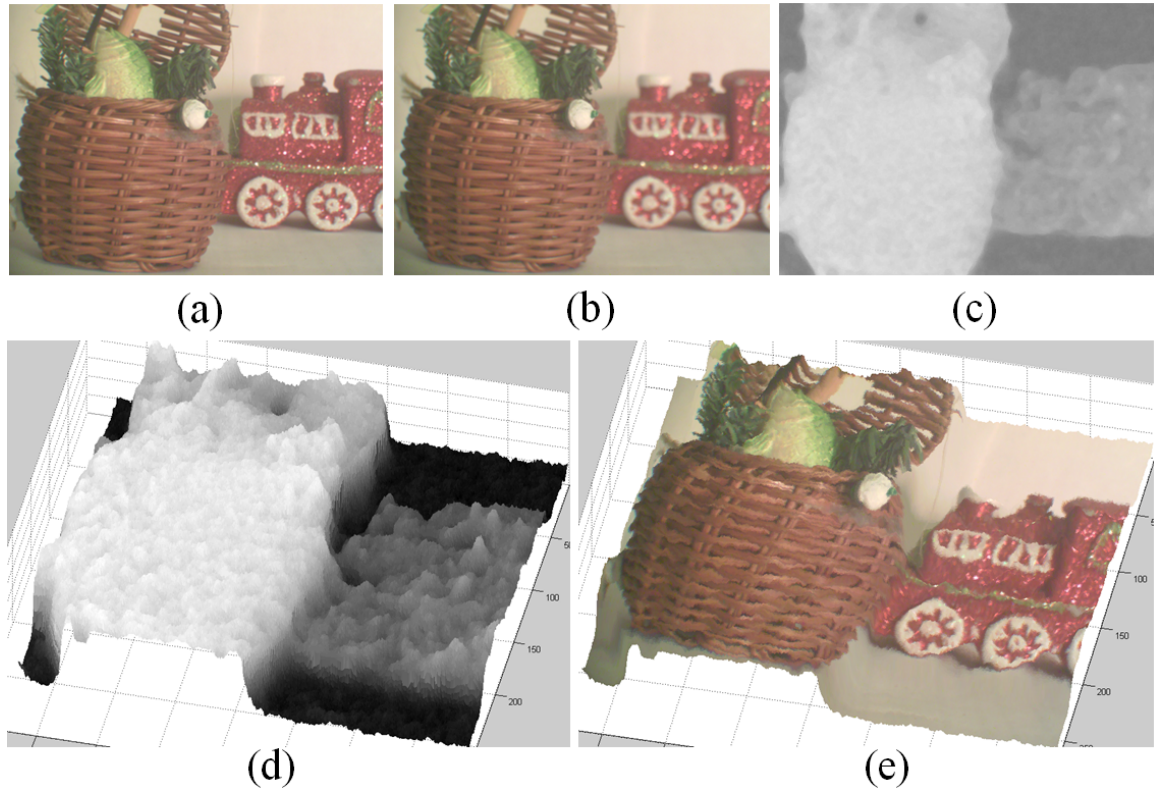


Fig. 7.8. Basket and train (a) in-focus image captured by camera (b) defocus image captured by camera (c) EDfD depth map (d) 3D view map of EDfD depth map (e) 3D view map of in-focus image

In order to confirm the real time operation of the lens and algorithm, the algorithms running time was tested on PC with single CPU. The size of test images were 640 by 480 and OpenCV library was used for the research. The average running time of the EDfD(EM/MPM) components is summarized in Table 7.2. As shown in this table, the iterative MAP-EM/MPM is the dominant factor. Table 7.1 only shows the starting frame, not a frame to frame processing. For frame to frame processing, Table 7.2 reflects that the initial depth generation is no longer needed because the calculated depth map of previous frame is used as initial depth map. Because this is a good estimate, the MAP-EM/MPM step converges to final result much faster than the starting frame. With the Middlebury data, the starting picture requires 40

iterations for convergence, however for the frame-to-frame speed using the previous depth map only requires 8 iterations. For these experiments, the research has not yet taken advantage of any parallelism.

Table 7.1
Average running time for each starting picture (EM/MPM)

STEPS	RUNNING TIME(S)
Initial depth generation	27.532
Preprocessing	0.143
Gaussian blur generation	28.724
MAP-EM/MPM (40 iterations)	425.879

Table 7.2
Average running time for frame to frame processing (EM/MPM)

STEPS	RUNNING TIME(S)
Preprocessing	0.143
Gaussian blur generation	28.724
MAP-EM/MPM (8 iterations)	85.617

The running time is further improved by using EDfD(Graph-cut). As shown in Table 7.3, "Initial depth generation", "Preprocessing" and "Gaussian blur generation" will remain the same. The table shows that the algorithm runs much faster due to the dominant factor of Graph-Cut.

Table 7.3
Average running time for each starting picture (Graph-cut)

STEPS	RUNNING TIME(S)
Initial depth generation	27.532
Preprocessing	0.143
Gaussian blur generation	28.724
Graph-cut	32.618

Table 7.4
Average running time for frame to frame processing (Graph-cut)

STEPS	RUNNING TIME(S)
Preprocessing	0.143
Gaussian blur generation	28.724
Graph-cut	24.571

By using the same dataset, the calculated the running time for SA-DfD and SFD was researched. For SA-DfD, the running time was tested on the same PC and also using OpenCV library. The average running time is 238.491s. And for SFD, the running time was tested using Matlab and running parallel on 8 CPUs. The average running time is 77.182s, using the 8 times parallelism. Compared the running time with these two algorithms, the EDfD research in the same order of magnitude, but is not fast enough yet for real time use in 30 frames per second movie cameras. One option is parallel execution in software, where up to 8 times improvement is feasible in EDfD speed with multi-core hardware. In addition, our previous research [43] which employs FPGA parallelism, showed that the hardware implementation of the EM/MPM function achieves over 100 times speed improvement. So, the conclusion is that this EDfD research is capable of real-time operation, given hardware future improvements.

8. POTENTIAL APPLICATION

This research could be used in many potential applications. In the medical field, using 3D over 2D images improves diagnosis accuracy and speed of procedures. This implies a strong potential for small, compact 3D cameras, such as described in this thesis. The output to a display can be a 2D image plus its depth map, which is a natural format to use in virtual augmented reality. This can be applied to image-guided surgery, for example. Therefore, this 3D camera and EDfD research gives the opportunity to get a 2D plus depth image in real-time.

Cooperating with University of Colorado Denver, this research is contributing as an important part of a new computer vision aided stereotactic system for brain surgery. The new system creates a real time three dimensional (3D) view and location guidance for the surgeon during the operation based on multi-view imaging, 3D image rendering, pattern recognition and real time 3D display techniques.

Figure 8.1 shows an example of preliminary result. (a) is an in-focus image of skull and (b) is a defocused one. The in-focus image and defocus image are both captured by using single camera equipped with microfluidic lens. (c) presents the depth map result by using EDfD algorithm.

Another application which under development is using 3D camera (this method) to evaluate bicyclist behavior analysis to inform the research toward transportation safety. In the transportation industry safety systems are becoming more autonomous, and pedestrian and bicyclist behavior needs to be analyzed in 3D. However, previous applications are based on the videos which are recorded by surveillance cameras or the cameras installed in vehicles. The cameras are set up on bicycles. So the videos recorded are from the first-person perspectives. Moreover, by using the particular lens - microfluidic lens, focus and defocus images are captured with fast speed. Using

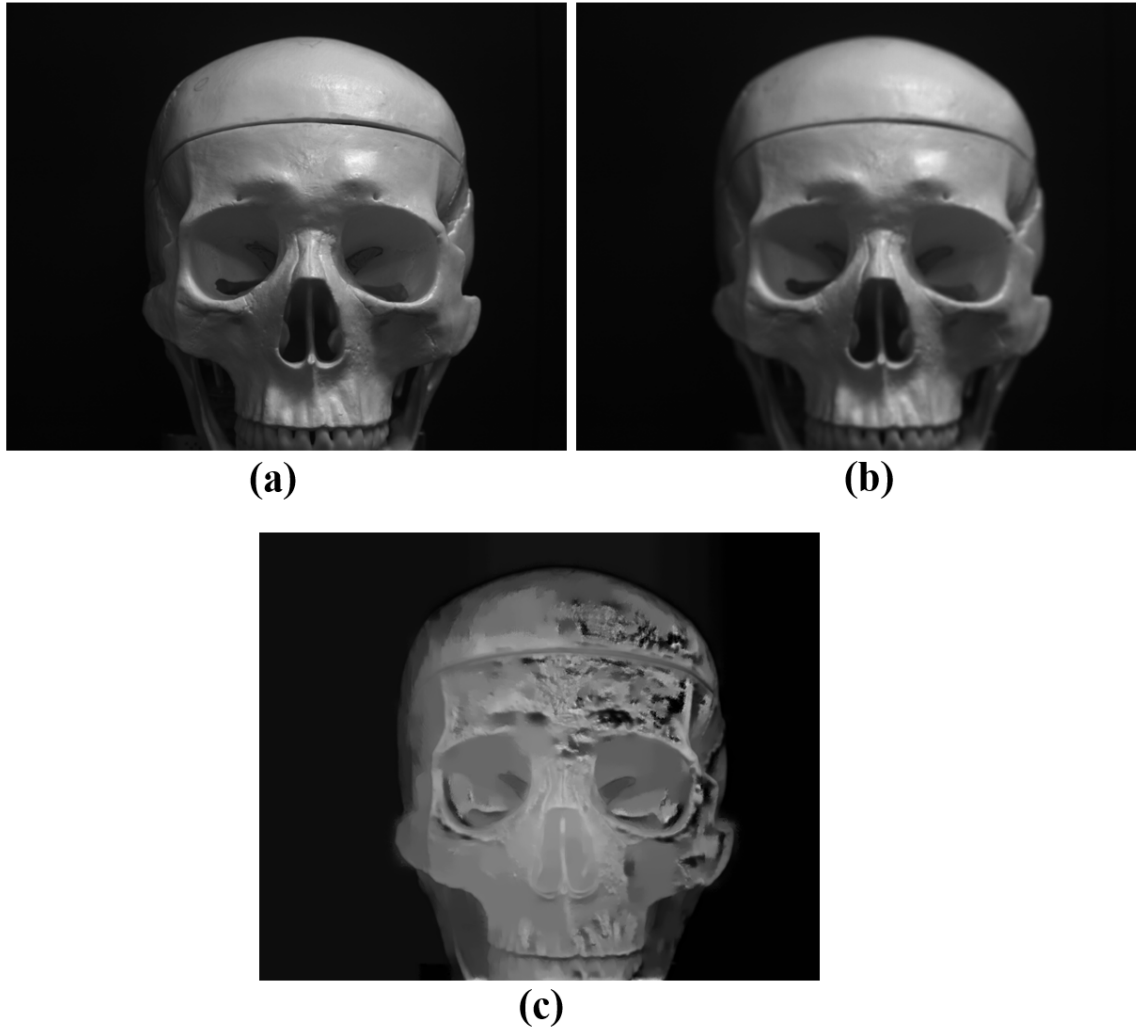


Fig. 8.1. EdfD result of a skull (a) In-focus image (b) Defocus image
(c) EdfD depth map

EdfD algorithm in the post-processing stage, the depth information can be calculated which is a very useful statistical parameter for analyzing the behavior of bicyclists.

9. SUMMARY AND FUTURE WORK

The motivation of this research is to find a low-cost real-time imaging system to achieve an accurate depth map. Compared with multi-camera methods, single camera method has a significant improvements in the cost and miniature size of the equipment.

Depth from Defocus (DfD) is one of the methods which produces a depth map using one single camera imager. One contribution of this research was the introduction of a new method of calculating depth information. The experiments show a very favorable accuracy result for the new Extended Depth from Defocus (EDfD) method, achieving an average Root Mean Squared Error (RMSE) of 3.543 gray level steps (1.38%) compared to ranging ground truth. This is an improvement of nearly two times when compared to standard techniques (SA-DfD and SFD).

Compared with traditional DfD methods, the EDfD algorithm is improved in several aspects: First, color information is added. Three channel images give the algorithm more reliable data to improve the accuracy of depth estimation. Second is the edge and texture information used to determine textured and textureless regions. Based on this information, the textureless areas can be resolved. Third, unlike the other methods, the EDfD algorithm showed the EM/MPM or Graph-Cuts as regularization methods showed much better performance.

Another innovation of this research is choosing the microfluidic lens for the EDfD imaging system. The low cost microfluidic lens is very suitable for small medical or consumer electronics devices. Also, it is capable of focus changes by only changing the controlling voltage. Moreover, the voltage can be changed at speeds fast enough to match the movie camera speed. It makes real-time frame rates feasible. Although the EDfD algorithm speed is not yet real time, the speed is within the same order of mag-

nitide as the competing algorithms, and it has significantly improved performance compared to other methods on a standard PC platform.

Further research help define the limits of the optics combined with the EDfD. First, the working area of EDfD is defined by the camera lens and parameter settings. If the camera settings are fixed, the maximum depth resolvable can be calculated. The depth of objects outside that range are equivalent to infinity. We also show that the defocus blur step is a non-linear monotonically increasing function related to depth up to the maximum depth position.

Camera's non-uniformities like lens aberration, lens distortion, different noise conditions and different contrast ratios were tested. If lens aberrations and distortions can be fixed by optics or could be measured by doing experiments, by incorporating them into the PSF function an accurate depth map can still be achieved. However, the signal dependent noise from imaging sensor is concluded as the primary detrimental effect for the EDfD algorithm. So in future, research on the effects of noise reduction on the EDfD will be developed.

Finally, the algorithm speed is not yet real time. However, in our previous work with similar algorithms, we confirm that a hardware speed improvement can be used to approach real time movie speeds. Further improvements to the algorithm can be made to reduce the computational load, such as incorporating motion information in the optimization.

LIST OF REFERENCES

LIST OF REFERENCES

- [1] D. Scharstein and R. Szeliski, “A taxonomy and evaluation of dense two-frame stereo correspondence algorithms,” *International journal of computer vision*, vol. 47, no. 1-3, pp. 7–42, 2002.
- [2] T. E. Bishop and P. Favaro, “The light field camera: Extended depth of field, aliasing, and superresolution,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 5, pp. 972–986, 2012.
- [3] A. P. Pentland, “A new sense for depth of field,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 4, pp. 523–531, 1987.
- [4] S. Chaudhuri, A. Rajagopalan, and S. Chaudhuri, *Depth from defocus: a real aperture imaging approach*, vol. 3. Springer New York, 1999.
- [5] J. Ens and P. Lawrence, “An investigation of methods for determining depth from focus,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 15, no. 2, pp. 97–108, 1993.
- [6] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 721–741, 1984.
- [7] J. Marroquin, S. Mitter, and T. Poggio, “Probabilistic solution of ill-posed problems in computational vision,” *Journal of the American Statistical Association*, vol. 82, no. 397, pp. 76–89, 1987.
- [8] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, “A comparative study of energy minimization methods for markov random fields with smoothness-based priors,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 6, pp. 1068–1080, 2008.
- [9] J.-I. Jung and Y.-S. Ho, “Depth image interpolation using confidence-based markov random field,” *Consumer Electronics, IEEE Transactions on*, vol. 58, no. 4, pp. 1399–1402, 2012.
- [10] C.-C. Cheng, C.-T. Li, and L.-G. Chen, “A novel 2D-to-3D conversion system using edge information,” *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 1739–1745, 2010.
- [11] T.-Y. Kuo and Y.-C. Lo, “Depth estimation from a monocular view of the outdoors,” *Consumer Electronics, IEEE Transactions on*, vol. 57, no. 2, pp. 817–822, 2011.

- [12] B. Berge, "Liquid lens technology: principle of electrowetting based lenses and applications to imaging," in *Micro Electro Mechanical Systems, 2005. MEMS 2005. 18th IEEE International Conference on*, pp. 227–230, IEEE, 2005.
- [13] M. Agarwal, R. Gunasekaran, P. Coane, and K. Varahramyan, "Fluid-actuated variable focal length polymer microlens system," in *MOEMS-MEMS Micro & Nanofabrication*, pp. 170–181, International Society for Optics and Photonics, 2005.
- [14] C. Liu and L. A. Christopher, "Three dimensional moving pictures with a single imager and microfluidic lens," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 2, pp. 258–266, 2014.
- [15] R. Jain, R. Kasturi, and B. G. Schunck, "Stereo imaging," in *Machine Vision*, vol. 5, pp. 289–308, McGraw-Hill New York, 1995.
- [16] W. E. Crofts, *The generation of depth maps via depth-from-defocus*. PhD thesis, University of Warwick, 2007.
- [17] J. F. Brenner, B. S. Dew, J. B. Horton, T. King, P. W. Neurath, and W. D. Selles, "An automated microscope for cytologic research a preliminary evaluation.," *Journal of Histochemistry & Cytochemistry*, vol. 24, no. 1, pp. 100–111, 1976.
- [18] M. Subbarao and T.-C. Wei, "Depth from defocus and rapid autofocusing: a practical approach," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pp. 773–776, IEEE, 1992.
- [19] H. Mir, P. Xu, and P. Van Beek, "An extensive empirical evaluation of focus measures for digital photography," in *IS&T/SPIE Electronic Imaging*, pp. 90230I–90230I, International Society for Optics and Photonics, 2014.
- [20] M. Subbarao and N. Gurumoorthy, "Depth recovery from blurred edges," in *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR'88., Computer Society Conference on*, pp. 498–503, IEEE, 1988.
- [21] V. P. Namboodiri and S. Chaudhuri, "Recovery of relative depth from a single observation using an uncalibrated (real-aperture) camera," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–6, IEEE, 2008.
- [22] S. Zhuo and T. Sim, "Defocus map estimation from a single image," *Pattern Recognition*, vol. 44, no. 9, pp. 1852–1858, 2011.
- [23] A. N. Rajagopalan and S. Chaudhuri, "An MRF model-based approach to simultaneous recovery of depth and restoration from defocused images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 7, pp. 577–589, 1999.
- [24] E. W. Weisstein, "Normal product distribution," Wolfram Research, Inc., 2003.
- [25] M. L. Comer and E. J. Delp, "The EM/MPM algorithm for segmentation of textured images: Analysis and further experimental results," *Image Processing, IEEE Transactions on*, vol. 9, no. 10, pp. 1731–1744, 2000.

- [26] D. Scharstein and R. Szeliski, “High-accuracy stereo depth maps using structured light,” in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–195, IEEE, 2003.
- [27] D. Scharstein and C. Pal, “Learning conditional random fields for stereo,” in *Computer Vision and Pattern Recognition, 2007. CVPR’07. IEEE Conference on*, pp. 1–8, IEEE, 2007.
- [28] A. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, “Depth estimation and image restoration using defocused stereo pairs,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 11, pp. 1521–1525, 2004.
- [29] P. Favaro and S. Soatto, “A geometric approach to shape from defocus,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 3, pp. 406–417, 2005.
- [30] Y. Boykov and V. Kolmogorov, “An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision,” in *Energy minimization methods in computer vision and pattern recognition*, pp. 359–374, Springer, 2001.
- [31] B. V. Cherkassky and A. V. Goldberg, “On implementing the pushrelabel method for the maximum flow problem,” *Algorithmica*, vol. 19, no. 4, pp. 390–410, 1997.
- [32] L. R. Ford and D. R. Fulkerson, “Maximal flow through a network,” *Canadian journal of Mathematics*, vol. 8, no. 3, pp. 399–404, 1956.
- [33] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [34] L. Norman, “A practical guide to lens aberrations and the lonely speck aberration test,” <http://www.lonelyspeck.com/a-practical-guide-to-lens-aberrations-and-the-lonely-speck-aberration-test/>, 2015.
- [35] LUXORION, “Optical aberrations,” <http://www.astrosurf.com/luxorion/report-aberrations2.htm>.
- [36] Click It Up A Notch, “Lens distortion: What every photographer should know,” <http://clickitupanotch.com/2014/06/lens-distortion/>, 2014.
- [37] H. S. Park, “Architectural analysis of a baseline isp pipeline,” in *Theory and Applications of Smart Cameras*, pp. 21–45, Springer, 2016.
- [38] Y.-M. Baek, J.-G. Kim, D.-C. Cho, J.-A. Lee, and W.-Y. Kim, “Integrated noise modeling for image sensor using bayer domain images,” in *Computer Vision/Computer Graphics Collaboration Techniques*, pp. 413–424, Springer, 2009.
- [39] X. Liu, M. Tanaka, and M. Okutomi, “Estimation of signal dependent noise parameters from a single image,” in *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pp. 79–82, IEEE, 2013.
- [40] R. G. Keys, “Cubic convolution interpolation for digital image processing,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, no. 6, pp. 1153–1160, 1981.

- [41] A. J. Bower, R. M. Bunch, P. O. Leisher, W. Li, and L. A. Christopher, "Correction of a liquid lens for 3d imaging systems," in *SPIE Photonics Europe*, pp. 842909–842909, International Society for Optics and Photonics, 2012.
- [42] L. A. Christopher and W. Li, "A single-imager, single-lens video camera prototype for 3d imaging," in *IS&T/SPIE Electronic Imaging*, pp. 82900Z–82900Z, International Society for Optics and Photonics, 2012.
- [43] Y. Sun and L. A. Christopher, "3d image segmentation implementation on fpga using the em/mpm algorithm," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pp. 670–673, IEEE, 2010.

VITA

VITA

Chao Liu received his B.S degree in Electrical Engineering from Southeast University, Nanjing, China in 2009, and the M.S degree from Purdue University. He joined the Ph.D. program in Electrical and Computer Engineering at Purdue University, West Lafayette, Indiana in 2011. His research interests include real-time 3D system, 2D+depth image processing algorithm, Camera sensor and ISP, Computer vision and Machine learning.