Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Weixu Li

Entitled
A 2D PLUS DEPTH VIDEO CAMERA PROTOTYPE USING DEPTH FROM DEFOCUS IMAGING AND A SINGLE MICROFLUIDIC LENS

For the degree of Master of Science in Electrical and Computer Engineering

Is approved by the final examining committee:

Dr. Lauren Christopher
Chair

Dr. Maher Rizkalla

Dr. Paul Salama

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Lauren Christopher

Approved by: Brian King 06/21/2011
Head of the Graduate Program Date

Graduate School Form 20
(Revised 9/10)

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:

A 2D PLUS DEPTH VIDEO CAMERA PROTOTYPE USING DEPTH FROM DEFOCUS IMAGING AND A SINGLE MICROFLUIDIC LENS

For the degree of Master of Science in Electrical and Computer Engineering

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22*, September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Weixu Li
_____
Printed Name and Signature of Candidate

07/08/2011
_____
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

A 2D PLUS DEPTH VIDEO CAMERA PROTOTYPE USING DEPTH FROM DEFOCUS IMAGING AND A SINGLE MICROFLUIDIC LENS

A Thesis

Submitted to the Faculty

of

Purdue University

by

Weixu Li

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Electrical and Computer Engineering

August 2011

Purdue University

Indianapolis, Indiana

To my family.

ACKNOWLEDGMENTS

Special thanks to Dr. Lauren Christopher who taught me how to think of massive ways of handling image processing problems, to Bradley A. Meyer, Chao Liu and Yan Sun who had always been there for help through my research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

ABSTRACT

Li, Weixu. M.S.E.C.E., Purdue University, August 2011. A 2D Plus Depth Video Camera Prototype Using Depth from Defocus Imaging and A Single Microfluidic Lens. Major Professor: Lauren A. Christopher.

A new method for capturing 3D video from a single imager and lens is introduced in this research. The benefit of this method is that it does not have the calibration and alignment issues associated with binocular 3D video cameras, and allows for a less expensive overall system. The digital imaging technique Depth from Defocus (DfD) has been successfully used in still camera imaging to develop a depth map associated with the image. However, DfD has not been applied in real-time video so far since the focus mechanisms are too slow to produce real-time results. This new research result shows that a Microfluidic lens is capable of the required focal length changes at 2x video frame rate, due to the electrostatic control of the focus. During the processing, two focus settings per output frame are captured using this lens combined with a broadcast video camera prototype. We show that the DfD technique using Bayesian Markov Random Field optimization can produce a valid depth map.

1. INTRODUCTION

During the past few decades, computer vision has become increasingly important. This is because tasks in several industrial applications must be performed in the 3D world. The scene in the 3D world is created by gathering reliable depth information. There are several techniques that are currently available to find depth information. They can be classified into two types: active and passive. The active type needs an artificial source of energy, laser or visible light. Passive methods have a wider range of applicability in industrial and military applications where constraints may prevent the use of artificial energy sources. One interesting passive method is Depth from Defocus (DfD). It was first proposed by Pentland [1], and then contributed by other researchers such as Chaudhuri [2] [3]. The advantage of DfD is that it only needs a single instead of multiple cameras, so the overall cost of the system can be reduced. In this thesis, we combine the DfD algorithm with a microfluidic lens. Using DfD, it is possible to generate the image with its depth map from a single perspective view with multiple focuses. In the earlist work contributed by Pentland [1] [4], he addressed the problem using the focused and the defocused images of the same scene. In his work, he estimated the blur parameter of a local region by comparing the focused and the defocused images. When applying DfD to a scene, The blur obtained from a real camera can be modeled as the space-variant Gaussian blur as suggested in [5]. In this thesis, the strategy we use is different from Pentland's, but similar to the method described in [3]. We made several improvements when designing the algorithm to produce a better result.

In order to find the blur parameter when applying DfD to the real world, we must develop an estimation algorithm. The estimation algorithm described in the literature models a depth map of an image as a Markov Random Field (MRF) [6]. According to the DfD technique, the blur parameter value is related to the depth of

the scene. Since the change of depth in a scene is usually gradual, it tends to have local dependencies. The MRF has the ability to capture these local dependencies. We therefore model the space-variant blur parameter as a MRF.

Several approaches are described in the literature. Cooper and Subramonia [7] suggested the Maximum a Posteriori (MAP) for the blur parameter estimation. Chaudhuri and Rajagopalan [2] indicated that it is possible to restrict the computations to local regions, which reduces the computational load. MAP probability is generally the criteria used for optimization. It has been proved that the MRF-MAP model is suitable for many vision-related problems [6] [8] [9].

In our research, a camera with single microfluidic lens is used to capture the focus and the defocus images. We combine DfD with a microfluidic lens to capture two focus settings per output frame. Our goal is to show that the DfD combined with microfluidic lens system is capable of producing a depth map for each frame in real time. In order to verify the capability of the system, the following specific objectives are discussed:

1. Develop an algorithm using DfD and Markov random field optimization to produce the depth map for each frame. In this thesis, our algorithm is modeled in software, which will be ported to hardware in the future in order to increase the speed.
2. Develop a control software for the microfluidic lens focus change.
3. Analyze the results and discuss if they are valid depth maps.
4. Verify the capability of the system.

Several aspects of this system and results are covered in this thesis. In Chapter 2, DfD theory and Markov random field optimization is covered, with the corresponding mathematics. The microfluidic lens is introduced in Chapter 3 by giving its working principles and performance. In Chapter 4, the overall system flow with implementation method is shown, algorithm details and calculation steps are presented. Chapter 5 shows the results we obtained. The results are from two catagories: synthetic images and camera images. In this chapter, we discuss the results and evaluate the

performance for each algorithm we test. Finally, the conclusion is given in Chapter 6 that the system is capable of estimating depth information of a scene in real time.

2. THEORY

In this chapter, we describe the theoretical basis of this work. We begin with the real aperture imaging system, then go through DfD technique, and end this chapter by discussing the MAP-MRF estimation.

2.1 Real Aperture Imaging

The lens equation for real aperture imaging is given by:

$$\frac{1}{D} + \frac{1}{v} = \frac{1}{f} \tag{2.1}$$

Where D is the distance of the object from the lens, f is focal length of the lens, and v is the distance between the lens and the image detector. When a point of light is defocused, the light rays coming through a lens are intercepted and spread within an area in the image plane. Figure 2.1 shows an image formation process when the light source is defocused.

2.2 DfD

In order to find the depth D, we use the DfD algorithm to estimate the depth given the focused and the defocused image of a scene. If a single point light source is defocused, Chaudhuri [2] indicates that the point spread function (PSF) should be used to describe the image intensity from the camera. By the ray tracing model described in [10], the object point should be imaged into a circular area. However in practice, the image formation process is affected by the brightness attenuation, the image will be a circle with the brightness falling from the center to the border gradually, viewed as blur. Thus, for a real camera system, the PSF of it can be

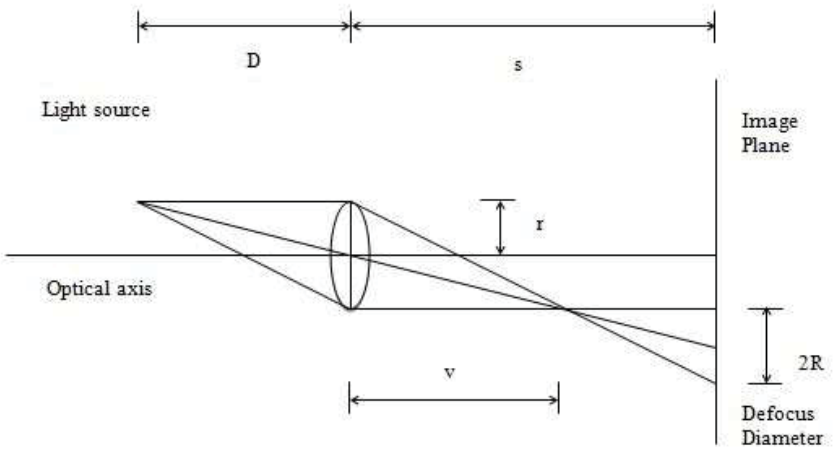


Fig. 2.1. Image formation process when a light source is defocused

approximately modeled as a symmetric 2D Gaussian function as suggested in [11] described by:

$$h(x,y) = \frac{1}{2\pi\sigma^2} exp \frac{-(x^2+y^2)}{2\sigma^2} \tag{2.2}$$

Where σ is the blur parameter, and x and y correspond to horizontal and vertical coordinates.

In this research, we modeled the local blur in an image by a 2D spatially varying Gaussian blur kernel. The local blur value is therefore specified by the blur parameter. When the source point is not in focus, its image is a blur circle whose radius is described by a blur parameter σ defined as:

$$\sigma = \rho\, r\, v(\frac{1}{f} - \frac{1}{v} - \frac{1}{D}) \tag{2.3}$$

The blur standard deviation can be calculated knowing the focal length of the optics f, the distance between the lens and the image detector v, the radius of the lens aperture r, the camera constant ρ, and the depth of the object in the scene D.

2.3 Markov Random Field

MRF is a graphical model in which a set of random variables have the Markov property. Given a MRF consisting of a set of random variables $X = Xv$ indexed by V. The Markov properties are:

1. Pairwise Markov property: Any two non-adjacent variables are conditionally independent given all other variables.

2. Local Markov property: A variable is conditionally independent of all other variables given its neighbors.

3. Global Markov property: Any two subsets of variables are conditionally independent given a separating subset.

MRF is characterized by the conditional distributions according to its definition, which is called local characteristics of the random field. In most of the image processing problems, it is usual to expect that the value of pixel does not depend on the pixels outside its immediate neighborhood.

2.4 MAP Estimation

Given two registered images I_{in} and I_{out}, where the object of interest is in focus in I_{in} and out of focus in I_{out}, a MAP estimation can be applied to calculate σ. Depth can then be calculated using Equation 2.3 and the known fixed camera parameters.

The defocused image of a scene can be represented as follows

$$g(x, y) = f(x, y) * h(x, y) + w(x, y) \tag{2.4}$$

where $f(.)$ is the focused image, $h(.)$ is the space-variant blur function that can be modeled by Gaussian function using Equation 2.2, $w(.)$ is the noise and $g(.)$ is the observed defocused image.

Let S denote a MRF which consists of the space-variant blur parameter $\sigma_{x,y}$. In this thesis, we assume S takes k classes. We define the noise field as W and observed

image field as G, assuming S and W are independent. Since S is modeled as MRF, the energy function can be written as

$$U(S) = \beta \sum_{c \in C_S} V_c(S) + |g(x,y) - f(x,y) * h(x,y)|^2, \tag{2.5}$$

Where C_S is the set of all cliques, β is the weighting factor and neighborhood weighting function of clique c is denoted as V_c, which can be expressed as

$$V_c(S) = \sum_c prior, \tag{2.6}$$

with the prior obtained from Equation 2.7, below:

$$prior = \begin{cases} 1 & if\ the\ pixel\ has\ a\ different\ class\ from\ its\ neighbor. \end{cases} \tag{2.7}$$

Figure 2.2 is the block diagram of MAP-MRF space-variant blur parameter estimation scheme. The a posteriori probability distribution of S is $P(S = s|G = g)$. MAP problem then can be used to estimate the space-variant blur parameter by applying Bayes' rule, which is given by Equation 2.8 and 2.9.

$$p(S = s|G = g) = \frac{P(G = g|S = s)P(S = s)}{P(G = g)} \tag{2.8}$$

$$MAP = max(\frac{P(G = g|S = s)P(S = s)}{P(G = g)}) \tag{2.9}$$

Our task then is computing the estimate of s. It had also been proved in [2] that

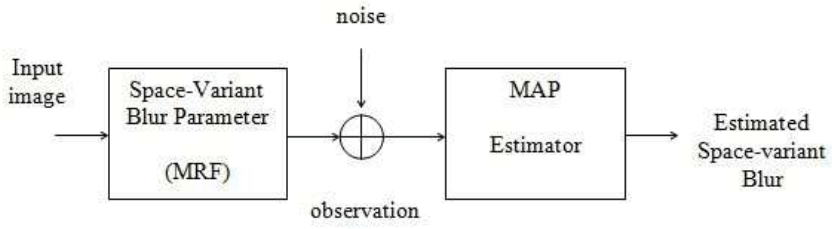


Fig. 2.2. Block diagram of MAP-MRF space-variant blur parameter estimation scheme

computing the MAP estimate of s is equivalent to minimizing the energy function $U(S)$. Thus an algorithm is needed to calculate the minimum of the energy function. In this thesis, we tested three computational algorithms: Simulated Annealing (SA); Iterated Conditional Modes (ICM) [12]; and maximization of posterior marginals (MPM) [13].

3. MICROFLUIDIC LENS AND CAMERA PROTOTYPE

In this chapter the microfluidic lens and camera prototype will be discussed by giving their working principles and performance. The microfluidic lens is an optimal component in our system development since it has the ability to change focus multiple times in a single second. This will enable us to get the depth information of a scene in real time.

3.1 Microfluidic Lens

A typical electrowetting microfluidic lens is shown as Figure 3.1. In this thesis the ARCTIC 416 lens is chosen, supplied by Varioptic Lens [14]. In order to capture focused and defocused images in real time, this lens guarantees that the focus can be adjusted continuously up to 30 frames per second. It also has a very fast response time and wide focus range from 10 cm to infinity.

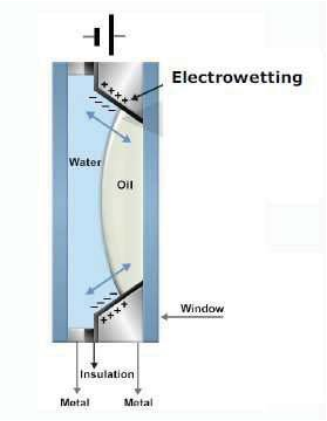


Fig. 3.1. Basic components of an electrowetting microfluidic lens [15]

In this lens, a hydrophilic coating is applied to the interior part of the lens chamber. A non-conducting fluid is placed in the bottom region. The insulating fluid is put on the contact surface between the metal and the fluid which has a large difference in refractive index with respect to water. Applying a voltage across the electrodes will force the fluid toward the metal part. Then as shown in Figure 3.1, the shape change of the fluid will change the optical properties of the whole lens. In this research, we can change the focal point by controlling the voltage applied to the lens.

3.2 Camera Prototype

Figure 3.2 shows the camera system we use in this research. The system is con-

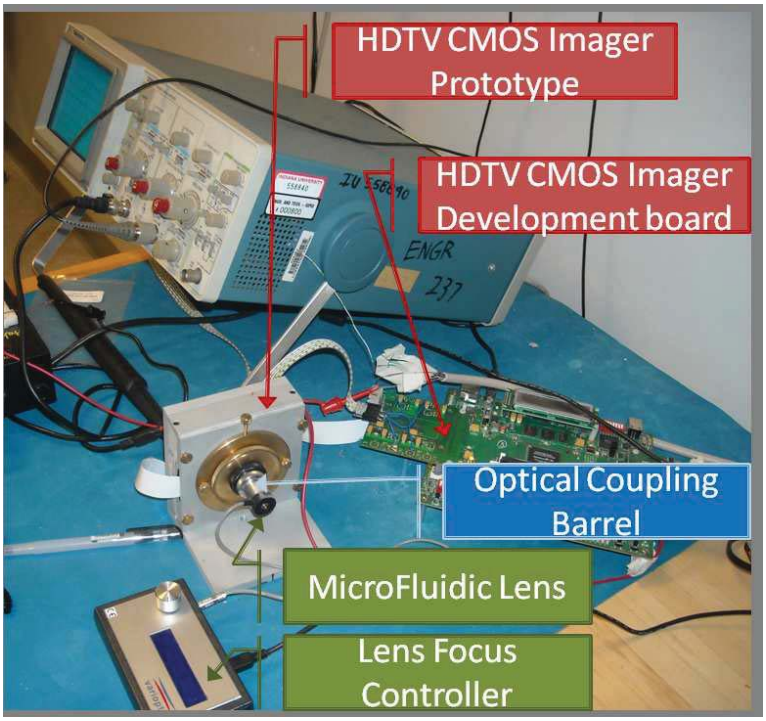


Fig. 3.2. Single camera system

sists of five components: The lens focus controller, microfluidic lens, Complementary Metal-Oxide-Semiconductor (CMOS) imager, CMOS imager development board, and

the computer. The image is formed on the imager, then the imager passes it to development board in real time. There is another board installed in the computer which is used to connect to the development board. Once the system is connected, the video stream is sent to the computer and observed on the monitor. Figure 3.3 is the working flow of this camera system. With this system, we are able to collect the test data, both still and motion images. An example of test data is shown in Figures 3.4(a) and (b), the image of the scene is taken in two different focuses, one is focused on the nearer object, the other is focused on the farther one.

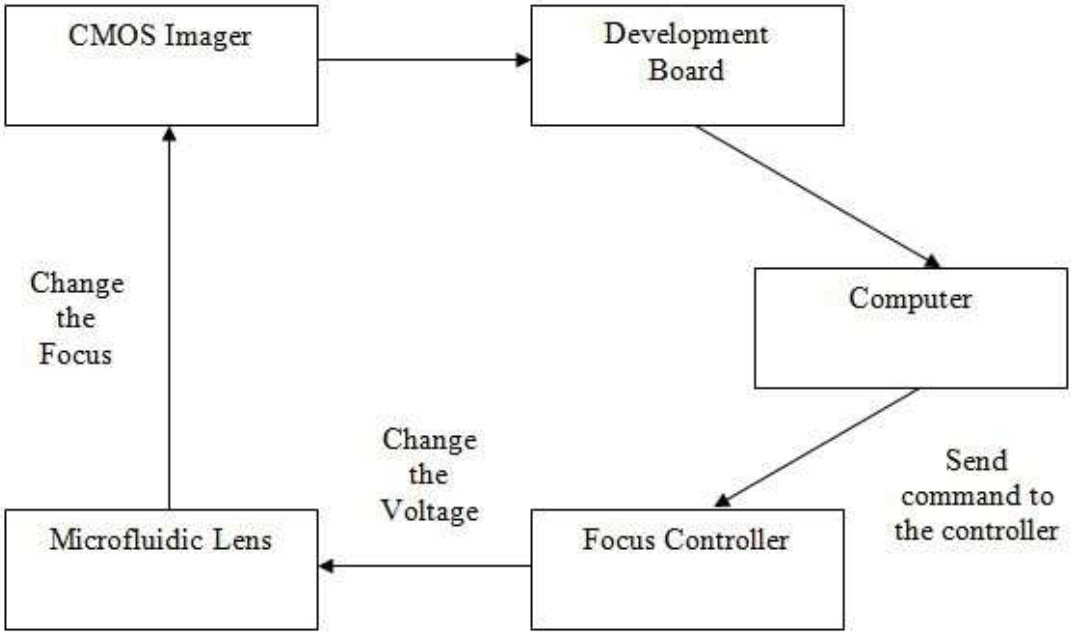
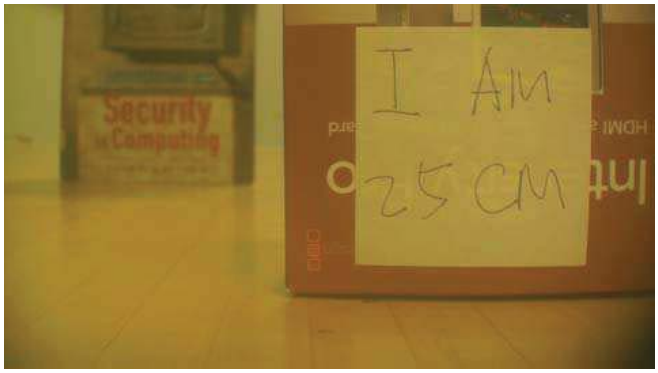
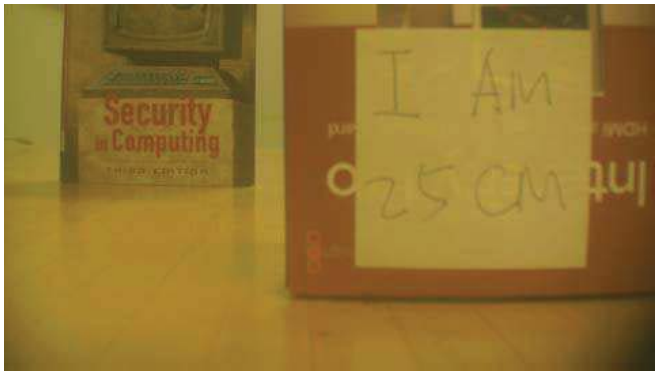


Fig. 3.3. Block diagram of the camera system flow



(a)



(b)

Fig. 3.4. Focused at (a) 25cm and (b) 75cm

4. IMPLEMENTATION PROCESS

As discussed in Chapter 2 and Chapter 3, we need a way to process the images in the DfD algorithm. In this thesis we use OpenCV [16] library, which is one the most powerful function libraries in image processing. It is an open source library which has over 2000 optimized algorithms and many useful functions. We use it to create multiple levels of blurred images from the original observed focused image, the function is called cvSmooth(). Figure 4.1 is an example of a set of blurred image created by OpenCV based on Equation 2.2.

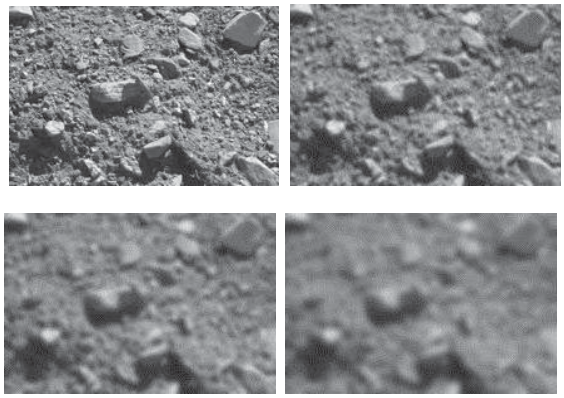


Fig. 4.1. Multiple levels of blurred images created by OpenCV

4.1 General System Flow

Figure 4.2 shows the overall system flow for DfD algorithm. As described in chapter 2, two inputs are required in order to calculate the space-variant blur parameter using a MAP estimator, one observed defocused image and a set of blurred image created by convolution. With the help of a microfluidic lens, we can capture focused and defocused images within a single frame time. Once the input image pair is prepared,

the algorithm will pass it to the MAP estimator which use three different algorithms referenced in chapter 2. The calculation inside the MAP estimator is a loop to make sure that the energy function converges to its minimum. Then MAP estimator outputs the blur map that contains all space-variant blur parameters. Therefore the depth map of a scene can be calculated from the blur map using Equation 2.3. In this thesis, the experiments are performed on two categories of images, one is synthetic images, the other is real aperture images.

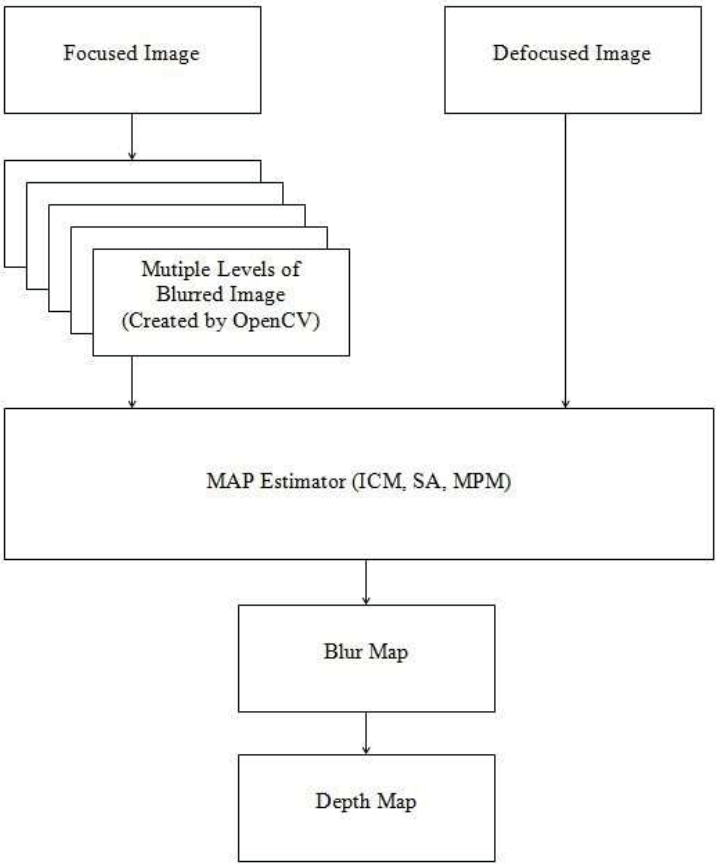


Fig. 4.2. DfD System Flow

4.2 More Details of DfD Algorithm

Following is the outline for DfD algorithm:

1. We initialize the data structure for the observed focused image I_0 and defocused image I_1. We create multiple levels of blurred images by Gaussian smooth function cvSmooth() in OpenCV I_{G1}, I_{G2},..., I_{G16}; using $k = 1, 2, 3, ..., 16$ to label them; Initialize I_B as a MRF with the same image size.

2. Load the input image pair. Images are either synthetic or captured from the camera.

3. Apply cvSmooth() function supplied by OpenCV to I_0 , create a set of blurred image. For this we always choose 16 levels. The 16 Gaussian blur parameters are chosen from 0.5 to 2.0 with equal step size.

4. Pass I_1, I_{G1}, I_{G2},...,I_{G16} to MAP estimator.

5. Loop through each pixels in raster order by x and y.

a. Starting from I_B as the default depth map. Calculate the absolute difference $D(x, y, k)$ for each class k using Equation 4.1.

$$D(x, y, k) = |I_{Gk} - I_0|^2 = |g(x, y) - f(x, y) * h(x, y)|^2 \tag{4.1}$$

b. Calculate the prior term $P(x, y)$ associate to the neighborhood system using Equation 2.6,

$$P(x, y) = \beta \sum V_c(S) \tag{4.2}$$

c. Calculate the logpost $L(x, y, k)$ for each class using Equation 4.3

$$L(x, y, k) = D(x, y, k) + P(x, y) \tag{4.3}$$

d. Optimize $L(x, y, k)$ by applying either MAP-ICM, MAP-SA, or MPM. Find the optimal solution (class number) for this pixel. Update this pixel value for I_B.

6. Calculate the average cost $Ave(I_B)$ following Equation 4.4.

$$Ave(I_B) = \frac{\sum_{s \in S} L(x, y, k)}{total\ number\ of\ pixels\ in\ the\ image} \tag{4.4}$$

7. Repeat step 6 until difference of average cost between iterations is below a predetermined threshold.

4.3 Controlling the Microfludic Lens

In order to verify that the focal length of the lens can be changed 2x video frame rate, software is designed for the purpose of controlling the microfluidic lens. In this thesis, we use VPS-048 controller supplied by Varioptic Lens as our control box. Figure 4.3 shows the details of VPS-048 controller. As shown in Figure 4.3, the RMS voltage applied to the lens holder can be changed easily by turning the knob on top of the controller. However, our goal is to make this change back and forth in a fixed frequency, so a software is needed to to fulfill this task.

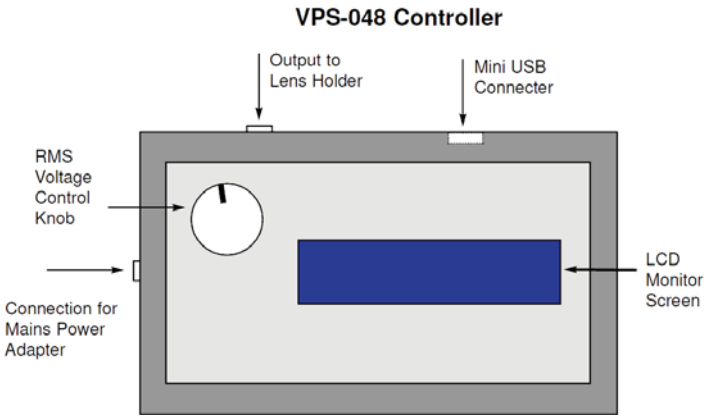


Fig. 4.3. The VPS-048 Controller [17]

Fortunately there is a communication protocol supplied by Varioptic, which we use to talk to the controller through a signal generated by computer. This signal will be transmitted by an USB cable trough the mini USB connecter on VPS-048. A standard protocol (char[16] in C++) is described as : STX / MODE / XXX / YYY / ZZZ / TTT / CRC / ETX, where:

1. STX: 0x02: Starting frame of the protocol.

2. MODE: 0x34, 0x33, 0x32. 0x34 is the fast mode, 0x33 and 0x32 are the slow mode. In this work we use 0x34 mode.

3. XXX: Lens voltage ($Vrms$); YYY: Lens current (μA); ZZZ: IC voltage ($3.3V$, predefined by Varioptic); TTT: IC current (mA).

4. CRC: Checksum.

5. ETX: Ending frame of the protocol.

Using fast mode, a single command is executed in 600 μsec, time is needed to decode the USB frame, process the command and send the data to lens driver. We can get a focused and a defocused image in a single frame time if a single command is executed less than 16.67 $msec$. Therefore we are sure that the performance of microfluidic lens meets our requirements. However, we still have 2 sources of uncertain timing, which can contribute to jitter in the timing of the lens focus change:

1. PC to USB timing.

2. Imager to video capture in PC timing.

When we try to change the focus in 30 frames per second, the timing accuracy suffers from the above two uncertainties. More work needs to be done to solve these problems for a robust system.

5. EXPERIMENTS AND RESULTS

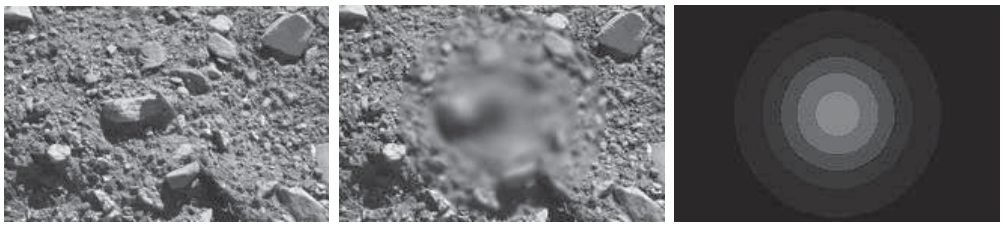
In this chapter, we discuss the performance of DfD algorithm based on the results in estimating the space-varying blur parameter. The test is applied both on synthetic image and real image sets from the camera. In our experiments, the blur function was assumed to be Gaussian with parameter $\sigma_{x,y}$. The number of levels, k, for the blur parameter was predefined to be 16. For all the depth map, larger pixel value stands for longer distance. In this chapter the pixel value in a image stands for the blur $\sigma_{x,y}$, which can be used directly to calculate the depth given the camera constant.

5.1 Synthetic Images

In this research we choose several synthetic blurred images as test images. The blur effect on them is assumed to be Gaussian blur. To evaluate the performance of DfD algorithm, the first step is to generate synthetic images with various space-varying blur. Then the DfD algorithm should be able to estimate the synthetic blur parameters for each image.

5.1.1 Rock Image

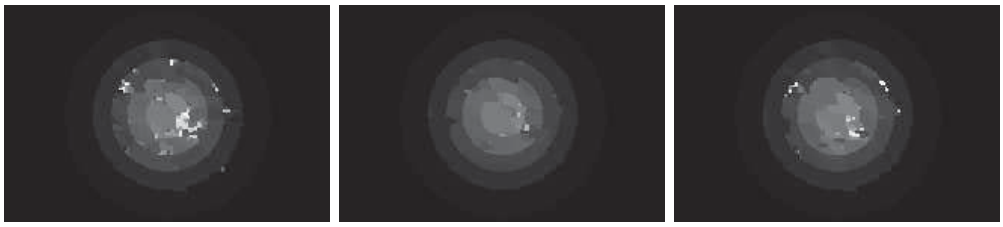
In our first set of experiments, a scene with rocks is defocused by space-varying blur which has the shape of concentric-disks. Figures 5.1(a), (b) show the focused and defocused images of rocks respectively, Figure 5.1 (c) shows the synthetic blur map applied to this scene, in which the pixel value stands for the number of class k multiplied by 8 for observation convenience.



(a) (b) (c)

Fig. 5.1. (a) Original focused rock image. (b) Synthetic defocused rock image. (c) Actual class label of blur parameters

Here the image has the size of 216x144, the value chosen for weighting factor was $\beta = 19$. Figures 5.2 (a), (b) and (c) shows the estimated blur map obtained from MAP-SA, MAP-ICM and MPM.



(a) (b) (c)

Fig. 5.2. Blur map of rock image obtained by (a) MAP-SA (b) MAP-ICM (c) MPM

After having the estimated blur map from the DfD algorithm, we are able to evaluate the performance of the algorithms. Figure 5.3 is surface plot of actual blur map, Figures 5.4, 5.5, and 5.6 are the blur maps estimated by MAP-SA, MAP-ICM and MPM. Table 5.1 summarizes the performance of these three algorithms.

From the performance of these three algorithm we can see that MAP-ICM works the best for this set of test images. It has the least convergence iterations and the smallest average cost.

Table 5.1
Performance of the three algorithms (rock image)

Algorithm	Iterations to Converge	Average Cost
MAP-SA	72	4.303
MAP-ICM	9	3.333
MPM	14	3.842

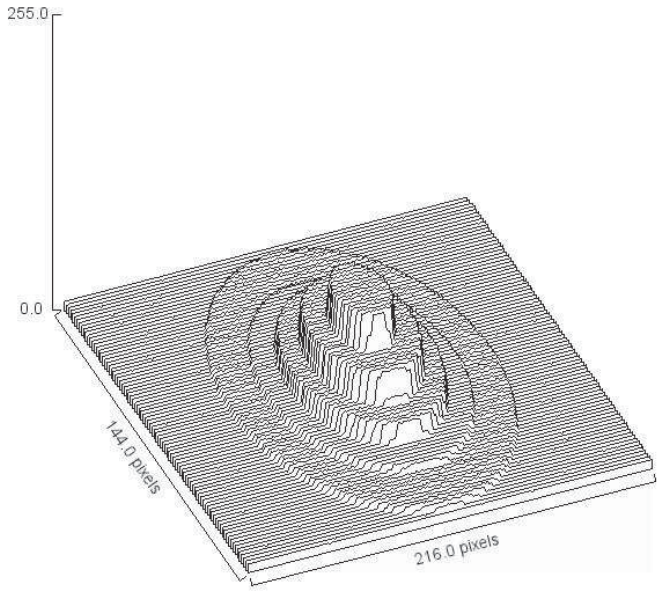


Fig. 5.3. Rock Image: Surface plot of actual blur map

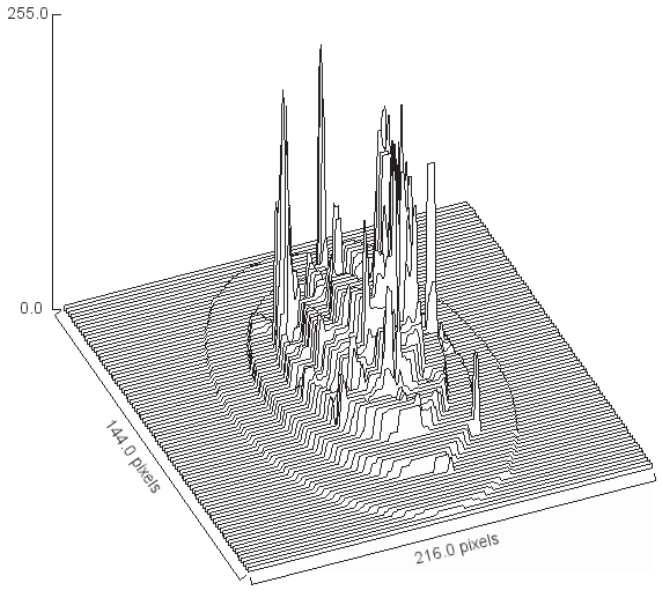


Fig. 5.4. Rock Image: Surface plot of blur map obtained by MAP-SA

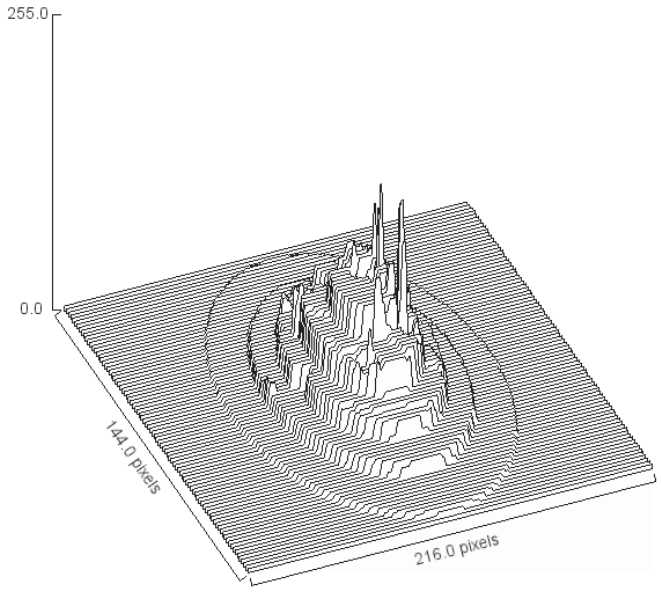


Fig. 5.5. Rock Image: Surface plot of blur map obtained by MAP-ICM

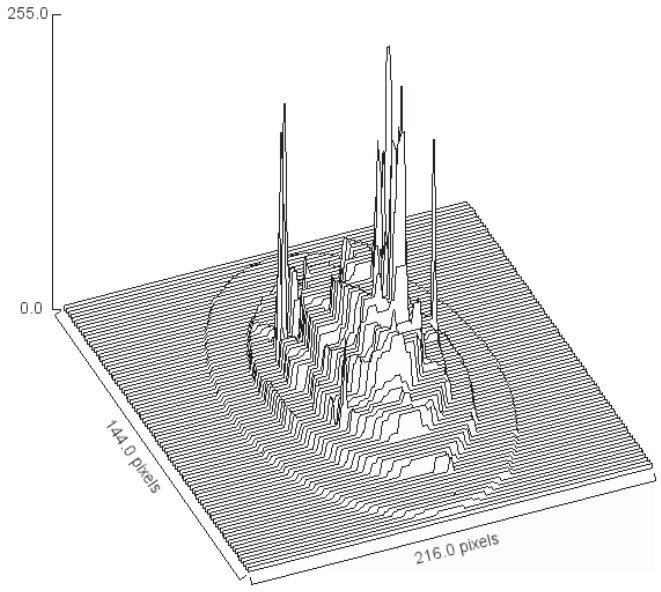


Fig. 5.6. Rock Image: Surface plot of blur map obtained by MPM

As shown in Figure 5.2. The algorithm produces some labeling mistakes when the blur parameter is either at the high end or the low end. That is mainly because at the low end, the defocused image has to close to the focused image; at the high end however, the defocused image becomes so blurry that there is not enough detail for DfD algorithm as a reference. To reduce those labeling mistakes, a 2D median filter can be added to the original results; Figure 5.7 shows the result after applying the median filter to the result obtained by MAP-ICM and Figure 5.8 is the surface plot. It can be observed that the result has improved.

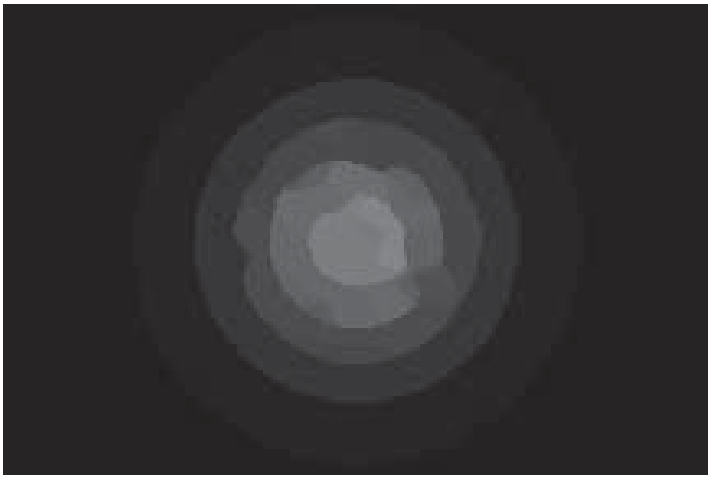


Fig. 5.7. Blur map of rock image after applying median filter

5.1.2 Fish Image

The second test is performed on a set of fish mosaic images. In contrast with the first test, here we use red, green, and blue (RGB) color images as the test data. In this process, the algorithm will convert the RGB color images to black and white (B/W) using Equation 5.1, then apply the algorithm to them.

$$BW = 0.212671 * RR + 0.715160 * GG + 0.072169 * BB \tag{5.1}$$

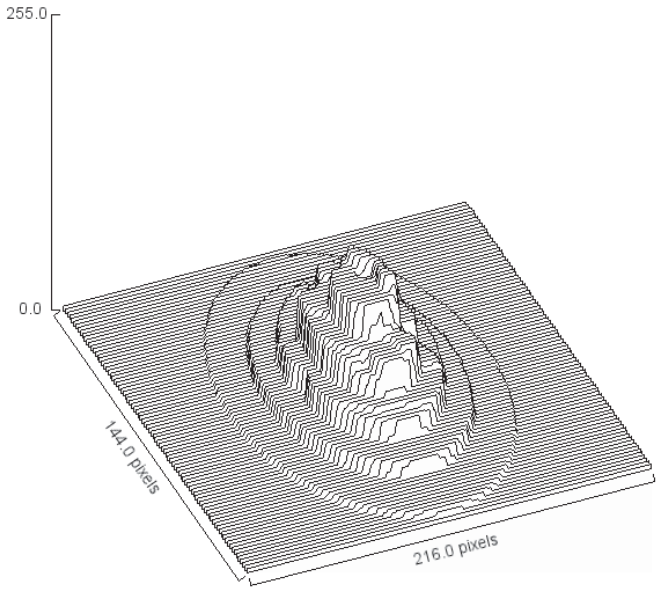


Fig. 5.8. Blur map surface plot after applying median filter

Here BW is the grey level of the B/W image, the range of BW is 0 to 255 since the data type of our test image is 8-bit unsigned character. RR, GG, and BB stand for the 8-bit pixel values of the red, green, and blue channel. Figures 5.9 and 5.10 show the focused and defocused image of fish mosaic image respectively, Figure 5.11 shows the synthetic blur map applied to this scene.

Here the image has the size of 450 by 450, the value chosen for weighting factor was $\beta = 19$. Figures 5.12, 5.13 and 5.14 shows the estimated blur map obtained from MAP-SA, MAP-ICM and MPM.

From the result we can see that the algorithm performance is also acceptable when applied to RGB color image. Similar to the B/W image, the results also suffered from several miss labeling (white dots in the result). We can use median filter to get rid of them as we did for B/W images. Table 5.2 summarizes the performance of this set of tests. Figures 5.15 is surface plot of actual blur map, 5.16, 5.17, and 5.18 are the blur maps estimated by MAP-SA, MAP-ICM and MPM. , Figure 5.19 shows the result after applying the median filter to the result obtained from MPM and Figure

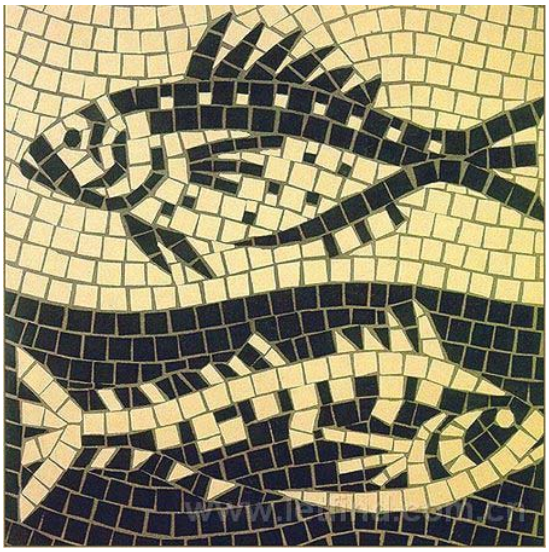


Fig. 5.9. Original focused fish image

5.20 is its surface plot. It can be seen that the result is improved after applying the median filter.

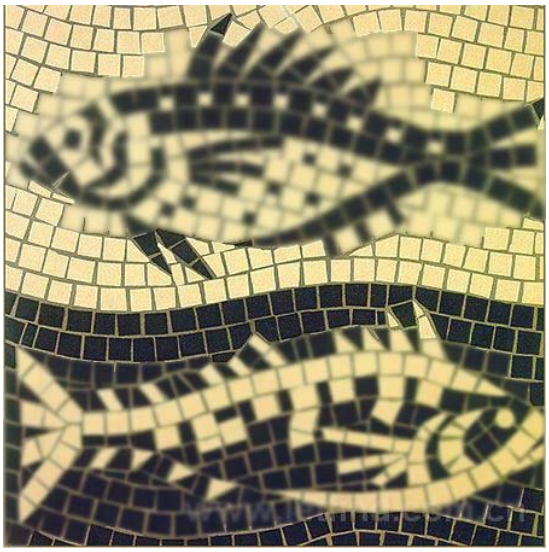


Fig. 5.10. Synthetic defocused fish image

Fig. 5.11. Actual class label of blur parameters

From the performance of these three algorithms we can see that MAP-ICM works the best for this set of test images. It has the least convergence iterations and the smallest average cost. Combined with the results of rock image, we conclude that

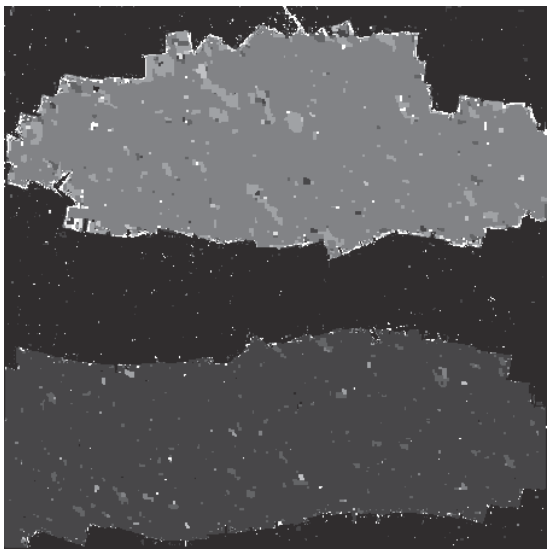


Fig. 5.12. Blur map of fish mosaic image obtained from MAP-SA

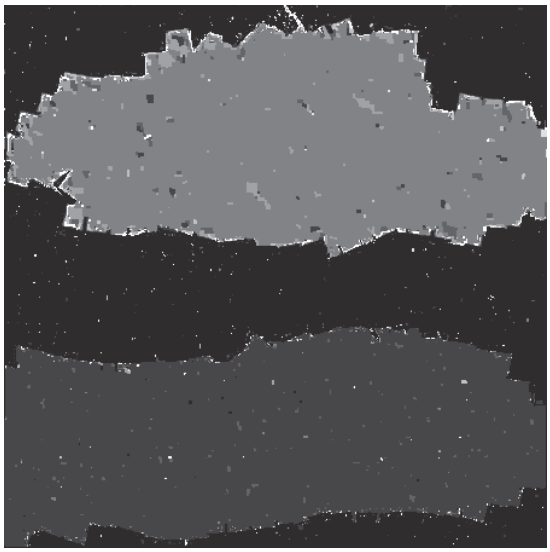


Fig. 5.13. Blur map of fish mosaic image obtained from MAP-ICM

MAP-ICM has the best performance among the three algorithms when dealing with the synthetic images. Therefore we choose MAP-ICM as our estimator when applying DfD algorithm to the real aperture images.

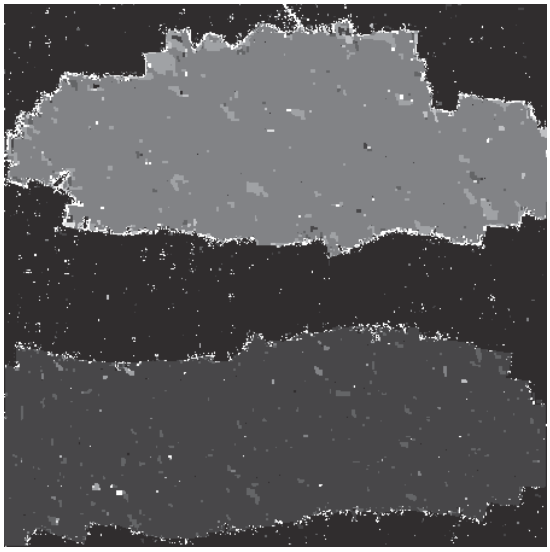


Fig. 5.14. Blur map of fish mosaic image obtained from MPM

Table 5.2
Performance of the three algorithms (fish image)

Algorithm	Iterations to Converge	Average Cost
MAP-SA	155	1.421
MAP-ICM	11	1.183
MPM	15	4.102

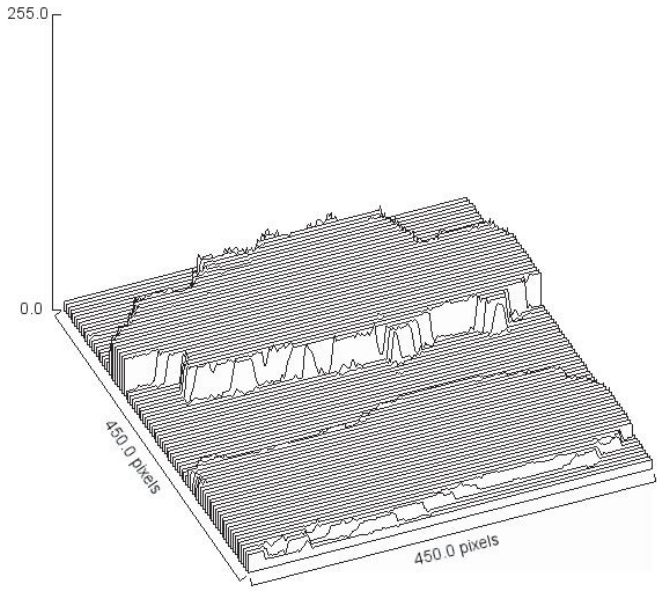


Fig. 5.15. Fish mosaic image: surface plot of actual blur map

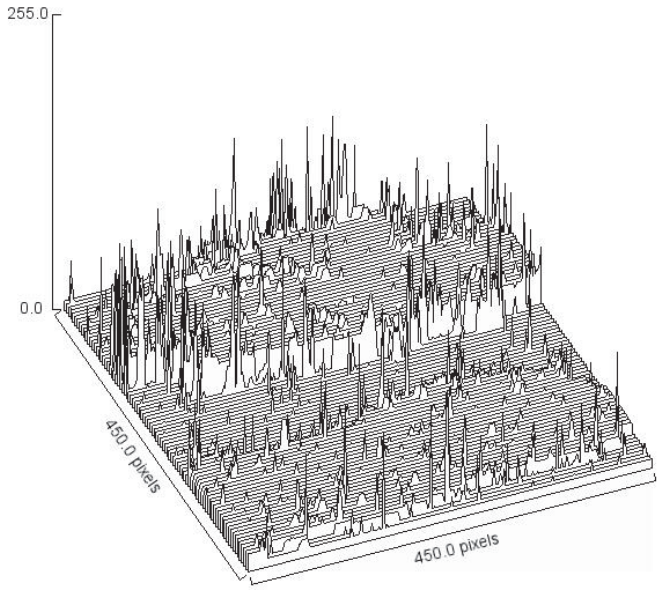


Fig. 5.16. Fish mosaic image: surface plot of blur map obtained from MAP-SA

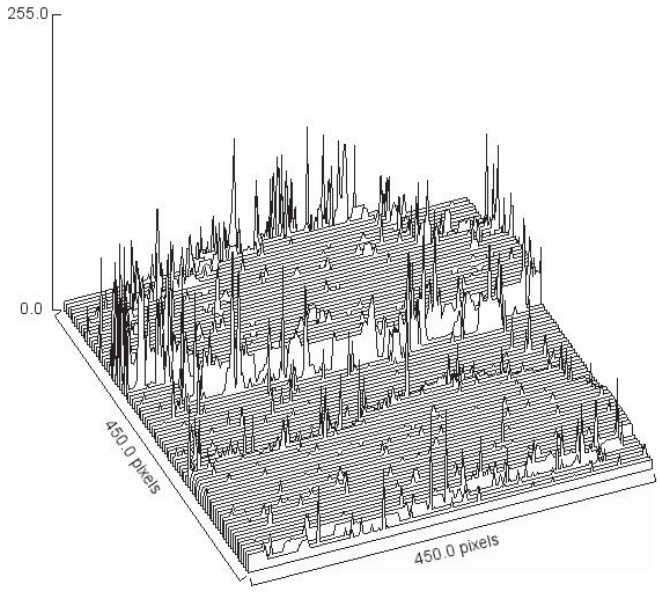


Fig. 5.17. Fish mosaic image: surface plot of blur map obtained from MAP-ICM

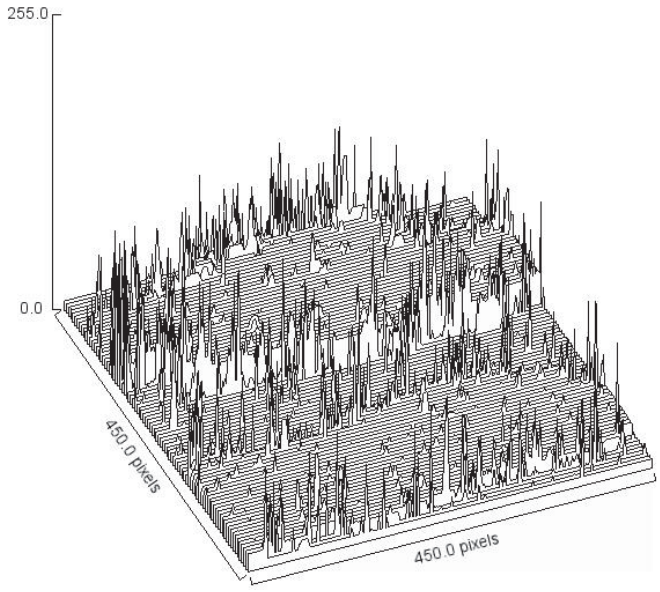


Fig. 5.18. Fish mosaic image: surface plot of blur map obtained from MPM

Fig. 5.19. Blur map of fish mosaic image after applying median filter

In literature Chaudhuri and Rajapopalan [2] had performed a recovery on the space-variant depth (i.e. blur) of a scene given the focused and defocused image. In order to evaluate our DfD algorithm, we compare our result to [2] and in [1]. The estimate of the depth obtained by using the method in [1] are quite noisy, whose average ranging error is 6.58 percent. The root mean squre (RMS) error for blur parameter σ is 0.51. The estimates of depth obtained by the proposed method in [2] is much better, which has a average ranging error of 4.05 percent. The RMS error for blur parameter σ is reduced to 0.13. In our first set of experiments, the estimates of depth of the rock image obtained by using MAP-ICM was shown in Figure 5.5. The average ranging error is 1.887 percent, the RMS error for blur parameter σ is also 0.13. In the second experiment, the original depth obtained by DfD algorithm is not quite satisfactory, which has a lot of noise on the edges. The ranging error is over 9 percent. However after applying a median filter to the original result, the ranging error reduces to about 1.822 percent and the RMS error for blur parameter σ is 0.05.

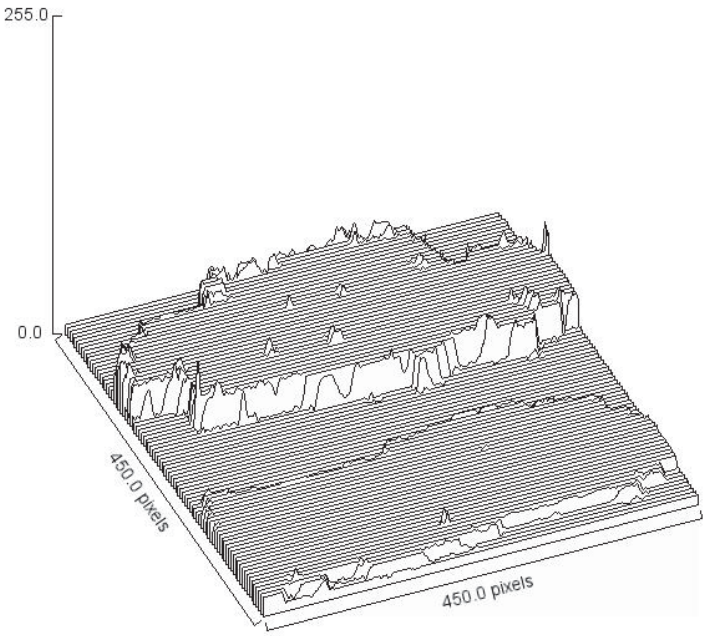


Fig. 5.20. Blur map surface plot after applying median filter

Therefore we conclude that there is a significant improvement in the estimate of σ accuracy using our method. Table 5.3 summarizes the performances of the above three methods.

In Chaudhuri's experiment, 64 levels of synthetic blur parameters was used as reference, while in this thesis we use 16 levels. If the defocused image has the blur parameters which are close to the steps, our settings will prevent it from jumping to the other levels. However if the blur parameters of the defocused image is laying between two levels, our setting will cause some quantization error. Future work remains to compare ours at 64 levels.

Table 5.3
Performance of the three algorithms (fish image)

Method	Pentland	Chaudhuri	Our method (rock)	Our method (fish)
Ranging error	6.58	4.05	1.887	1.822
RMS error	0.51	0.13	0.13	0.05

5.2 Images Captured from Camera with Microfluidic Lens

In order to verify if the DfD algorithm can be applied to real time, we need to test it also on image captured from a real camera. In this research, our test data is collected from a single camera system that has been discussed in chapter 3.

Figure 5.21 shows the scheme when handling the images from a real camera. Different from the synthetic ones, images obtained from a real camera with blur effect come along with the noise and lens defects. Therefore, instead of simply applying the same algorithm, we segment the image in parallel to the MAP estimator. Here our development is based on the assumption that a segmented object has a same depth over its surface. First a MAP estimation is performed, then with the help of segmented image (we call it an atlas the algorithm will calculate the dominant pixels for each class inside a certain region. The algorithm will put the whole region into the class which turns out to be dominating the region.

Figure 5.22 is our test image set, (a) is the focused image, (b) is the defocused image. There are two objects in the scene, one is located at 25cm from the camera, the other is at 50cm. As described in Fig 5.21, we first segment Fig 5.22 (a) using pyramid segmentation. The segmentation function is supplied by OpenCV, Figure 5.23 shows the result of pyramid segmentation. Once the original segmentation result is prepared, the algorithm will process it with a median filter to remove detail. Fig 5.24 is the output of the median filter, which becomes the atlas for the MAP estimator.

In this research, we find that DfD algorithm appeared to be more accurate when estimating the region which has the details. We have improved our algorithm based on this fact by writing a post process module. Figure 5.25 shows the edges after applying Canny [18] edge detection to the original focused image. Figure 5.26 (a) shows the blur map obtained directly from DfD algorithm. We pass three inputs: blurmap from DFD algorithm; edges of the scene; and the atlas. We first find all the pixels belonging to the edges in the image, then we perform a region growing by 10 pixels toward all directions, where the pixel value is obtained from the MAP

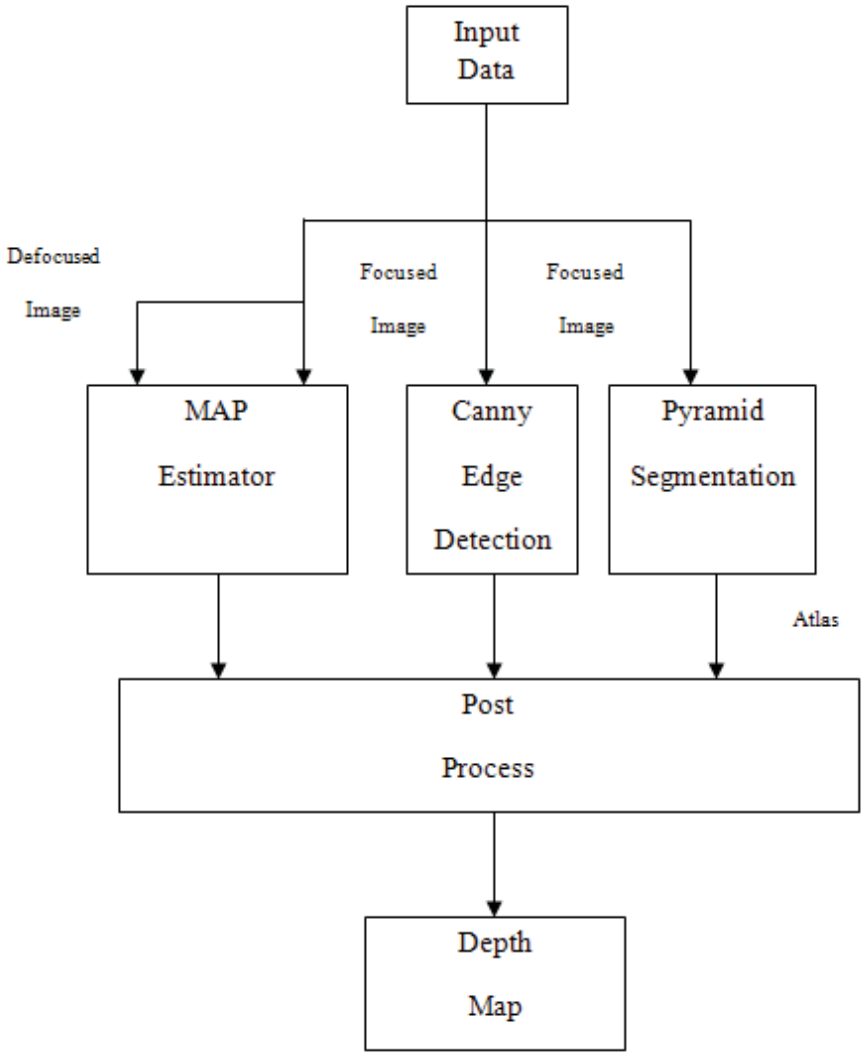
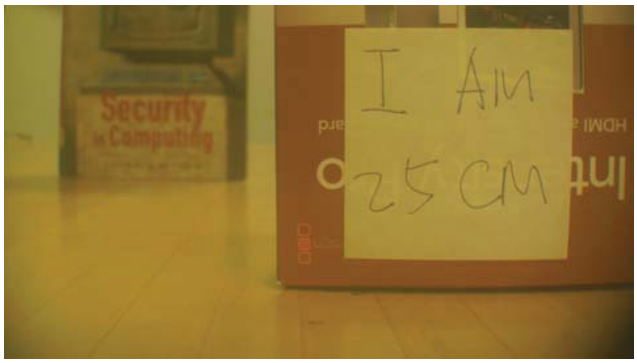


Fig. 5.21. Segmentation-DfD algorithm scheme for images obtained from single camera

estimation result. Figure 5.26 (b) shows the blur map after ignoring all the pixels which are not close to the edges, where black region is where the pixels are eliminated. In another words, Figure 5.26(b) is our final reference when calculating which class



(a)

(b)

Fig. 5.22. Test images obtained from single camera: (a) Focused at 25cm (b) Defocused image

Fig. 5.23. Original segmentation result using pyramid segmentation

Fig. 5.24. The atlas for MAP estimator

is dominating each segmented area. Finally, we calculate the the number of pixels in each class inside every segmented region shown in Figure 5.24, and assign the class label to the largest.

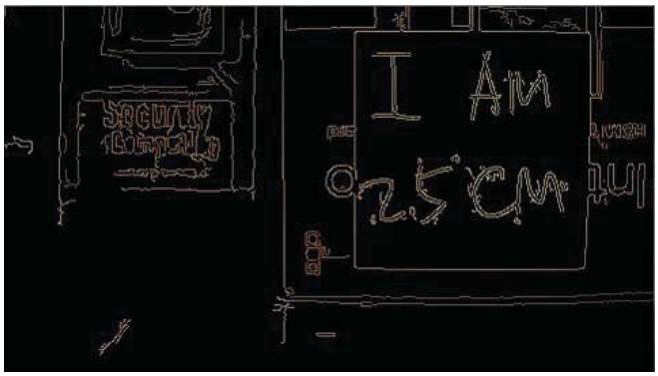
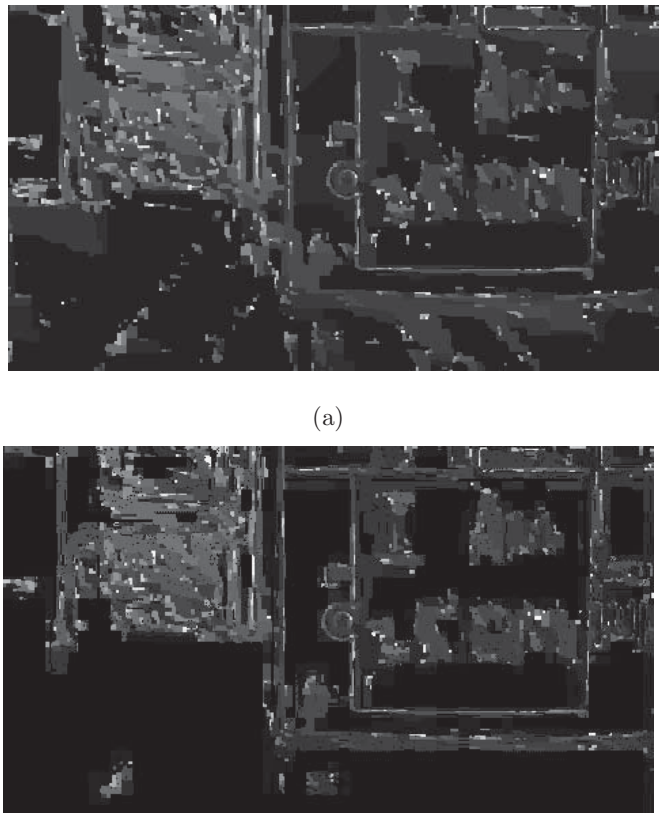


Fig. 5.25. Edges on the test image

After post processing the algorithm outputs the final blur map. Figure 5.27 presents this output. It is very clear that Figure 5.27 is an improvement compared to Figure 5.26(a) because more pixels are estimated correctly. The weakness of the pyramid segmentation is that it can only estimate the depth information when the objects are segmented properly in a scene. Where there is no detail in image we do



(a)

(b)

Fig. 5.26. (a)Blur map of test images obtained from DfD algorithm (b) Pixels picked from the original blur map after adding the edge information

not find the depth, and we assign $k = 0$. Other depth cues can be used to get the entire depth information of a scene, such as vanishing point or motion.

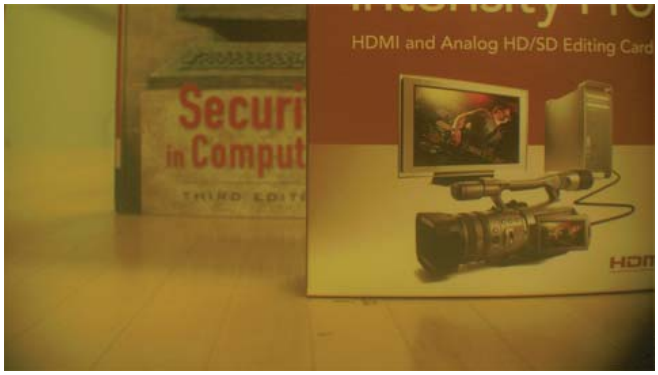
In the literature Namboodiri, Chaudhuri, and Hadap [3] proposed a combined segmentation method similar to ours. However, we use pyramid segmentation and Canny edge detection instead of graph-cut. Compared to their result, our results are more clear and well grouped.

However there are also some weaknesses in our method. As we can see in Figures 5.28 (a) and (b), the pyramid segmentation does not produce good result if the following happens:

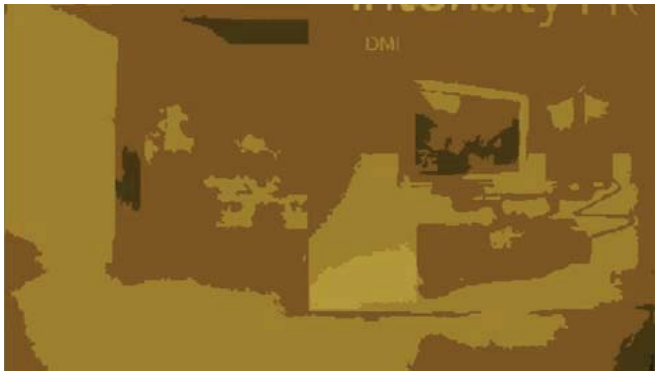
Fig. 5.27. Blur map obtained from segmentation-DfD algorithm

1. One object is blocking the other.
2. The objects have similar colors.

Another weakness is caused by our assumption that the objects are flat depth when we dealing with the images captured from camera. If the object surface is non-flat or not vertical to the lens plane, our algorithm will output an incorrect result. We also find that the experimental result of camera scenes are inferior to the synthetic, and more research is required to find the difference. In general, the depth inference is not valid for flat areas, this problem is common in DfD, Depth from Disparity, and other depth finding techniques. Therefore, other computer vision cues such as vanishing point or motion analysis may improve results for flat colored scene of objects.



(a)



(b)

Fig. 5.28. Problem caused by pyramid segmentation:(a) test image (b) segmented image

6. CONCLUSION

In this thesis we have proposed a new system structure which can be used to obtain the depth information of a scene in real-time using a new DfD algorithm. It use a MAP estimator and the MRF framework, augmented by an efficient segmentation and edge detection method. The most important advantage of this system is it only needs a single camera. Compared to two camera disparity, the system cost is lowered

We have described the mathematical basis for this research in Chapter 2. The microfluidic lens we used in this work is discussed in Chapter 3, as well as our camera system. To verify that our system is capable of doing the job mentioned above, we first introduced the performance of the Microfluidic lens in chapter 4. Given the response time, we show that the camera system is able to get the focused and defocused image in a single frame time. With the results given in chapter 5, we show that our algorithm is capable of producing the valid depth map both for synthetic images and real aperture images. Also in Chapter 5, we compare our method to 2 other methods, and show our advantages and limitations.

During our experiments, two difficulties came from the camera system. We have a prototype with optics that are imperfect, and the camera speed has a maximum of 30Hz. Future work will be to fix those problems. We believe that we can improve our result based on better camera-lens system. Although the microfluidic lens is able to change focus twice in a single frame time, the software loop controlling it needs to be free of timing jitter. The throughout of the software solution does not complete in a single frame time. Table 6.1 shows the running time for each step. Therefore interesting work in the future is to port our algorithm to hardware, thereby improving the throughout significantly. The steps that would most benefit from hardware acceleration are: create gaussian blur; MAP estimator; and post processing. Another interesting application of our system is medical imaging. Since the microfluidic lens

Table 6.1
Running time for each step in DfD algorithm (450x450 image)

Steps	Running Time
Create Gaussian Blur	3.356s
Loading Data	0.125s
Segmentation	0.170s
Canny Edge Detection	0.143s
MAP-ICM	4.873s
Post Processing	0.842s
Total	9.509s (ICM)

is small, it can be added to a probe, and can be used to collect depth information during surgery.

In this thesis we have developed an DFD algorithm, we have (<2%) error and 0.05 RMS error result for DFD where image has valid edges. Our execution time on CPU for the whole algorithm is less then 2 minutes.

LIST OF REFERENCES

LIST OF REFERENCES

[1] A. Pentland, "Depth of Scene from Depth of Field," in *Image Understanding Workshop*, pp. 253–259, 1982.

[2] S. Chaudhuri and A. Rajagopalan, *Depth From Defocus: A Real Aperture Imaging Approch*. Springer, 1999.

[3] S. C. V. Namboodiri and S. Hadap, "Regularized Depth from Defocus," in *IEEE 15th International Conference on Image Processing*, pp. 1520–1523, 2008.

[4] A. Pentland, "A New Sense for Depth of Field," *IEEE Trans. Pattern Anal. Machine intell.*, vol. 9, pp. 523–531, 1987.

[5] J. Ens and P. Lawrence, "An Investigation of Methods for Detemining Depth from Focus," *IEEE Trans. Pattern Anal. Machine intell.*, vol. 15, pp. 97–107, 1993.

[6] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distribution and the Bayesian Distribution of Images," *IEEE Trans. Pattern Anal. Machine intell.*, vol. 6, pp. 721–741, 1984.

[7] D. Cooper and J. Subramonia, *The Use of Markov Random Fields in Estimating and Recognizing Objects in 3D Space*. Springer, 1993.

[8] P. M. S. Kapoor and U. Desai, "Depth and Image Recovery using Markov Random Field," *IEEE Trans. Pattern Anal. Machine intell.*, vol. 16, pp. 1117–1122, 1994.

[9] S. Z. Li, "Markov Random Field Models in Computer Vision," 1994.

[10] J. S. Y. Hung and D. Copper, "Model-Based Segmentation and Estimation of 3D Surface from Two or More Intensity Images using Markov Random Fields," in *IEEE Intl. Conf. on Pattern Recognition*, pp. 390–397, 1990.

[11] G. Surya and Murali Subbarao, "Depth from defocus by changing camera aperture: A spatial domain approach," 1993.

[12] D. Rowe, D. B. Rowe, and S. J. Press, "Bayesian Factor Analysis by Gibbs Sampling and Iterated Conditional Modes," 1994.

[13] J. M. S. Mitter and T. Poggio, "Probabalistic Solution of Ill-Posed Problems in Computational Vision," *Journal of the American Statistical Association*, vol. 82, pp. 76–89, 1987.

[14] Varioptic, "Varioptic Home Page," http://www.varioptic.com/en/home.html.

[15] T. Garnier, "The Liquid Lens Technology," http://www.varioptic.com /pdf/whitepaper/VariopticLiquidLensTechnology20101202.pdf.

[16] G. Bradski and A. Kaebler, *Learning OpenCV: Computer Vision with the OpenCV Library.* O'REILLY, 2008.

[17] T. Garnier, "Varioptic Liquid Lens Developer Kits," http://www.varioptic.com/ img/res/products/MASDDeveloperKitsDescriptionDecember10.pdf.

[18] J. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Anal. Machine intell.*, vol. 8, pp. 679–698, 1986.