High-Speed Optics based Depth Mapping for Automated Active Safety System

by

JooHun Kim

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author ...... Signature redacted ....
Department of Electrical Engineering and Computer Science
May 20, 2016

Certified by.................... Signature redacted .
Kamal Youcef-Toumi
Professor
Thesis Supervisor

Accepted by .... Signature redacted
Dr Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

High-Speed Optics based Depth Mapping for Automated Active Safety System

by

JooHun Kim

Submitted to the Department of Electrical Engineering and Computer Science
on May 20, 2016, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

The usage of image processing in automotive active safety systems has increased dramatically in the recent decades. Main interest of the digital signal processing has been in the area of communications or networks, but with various advancements in the digital image processing, new type of applications has become possible. The advancement of digital camera technology along with the fast pace of development of faster microprocessors, it has become possible to implement more advanced image processing tasks for high-speed applications. The availability of image processing results in a timely fashion opens up new possibilities. In this project, sponsored by Ford, we will look at the use of machine vision system to build a standalone system capable of providing real-time depth map for automated emergency brake systems in cars.

Thesis Supervisor: Kamal Youcef-Toumi
Title: Professor

Acknowledgments

First of all, I would like to thank Prof. Kamal Youcef-Toumi for giving me an opportunity to participate in this amazing project. It has been a tough, but a wonderful experience, and it was a great chance for me to experience the real life applications of the materials that I have learned here at MIT. I would also like to thank Iman Soltani Bozchalooi, for all the help he has provided in the course of the research, as well as his early contributions to the project. His early work and ideas were the founding stone of this project. Also, I would love to say thanks to Mohsen Lakehal-ayat and Ford, for initiating such an awesome project, and funding the whole process. I also have to say thanks to all my friends, especially my fellow MIT ARA students and brothers of PKT for their love and support, that allowed me to overcome any hardships or obstacles ahead of me. Lastly, I cannot say more thanks to my parents and my sister who were always there with an unconditional support, and guided me through any hard times. Because of them, everything I have done was possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Focus of our reasearch is to use active lens to a build depth map for application of emergency brake system. Depth mapping is a technic that builds distance map of the 3-dimensional environment from various sensors. Depending on the type of sensor, many kinds of applications are possible. In this thesis, we will focus on the vision system, especially on the Depth from Focus algorithm.

Chapter one describes various kinds of depth mapping system, introduce what active lens is, and breifly explain different vision systems.

Chapter two describes more in-depth detail about depth from focus and focus measure, including it's limitation and choice of the system to be implemented.

Chapter three describes system design process of depth mapping system with active lens, and how one can build such system.

Chapter four describes the performance optimization of the algorithm, and show how parallelizaton can speed up the system. Performance analysis of current system are also discussed.

1.1 Motivaton of Research

Since its invention, automotive has been the major means of transportation for mankind. Automotive industry is among the largest here in the United States and elsewhere in the word, producing more than 87 million unit per day [26]. There has been a number of major renovations in the car industry, but one thing that has not changed much is the way how cars are operated by drivers. In 19th century and still in 21st century, human takes the full responsibility for navigating the car in a more or less similar fashion. During the past decade there has been an increased interest in the development of fully automated systems that control the car, and a lot of research is conducted in this area. Development of cars with more complex active safety systems is one step along this path. There are various forms of active safety in the cars ranging from lane detection to obstacle avoidance or emergency braking. Regardless of the methodology all these functionalities require a reliable map of the surrounding. For example, Emergency brake system refers to a set of sensors and system that could activate a brake so that the automotive can be stopped soon enough to avoid a possible life threatening accident. Such systems are widely studied because of their importance in controlling the automotive and the safety of the passengers. There are great number of distractions on the road, and these distractions tends to cause a harmful result. In an attempt to prevent such situation, many automotive manufacturers are focusing on the development of reliable system to stop the car in case of obstacle in front. Sensors that have been studied include laser, stereo-vision system, and radar. Each of the technologies available today benefit from certain advantages and suffer from other disadvantages. In this project, I will further study the potentials of vision systems for collision avoidance and emergency braking in the cars. More specifically I will investigate the development of mono-vision system for reliable depth mapping with application to active safety systems.

1.2 What is Depth Mapping

Depth mapping is a technic of mapping surrounding three dimensional environment to obtain information about the structure of the scenery. Informations about the distance and three dimensional structure of the environment has a wide range of applications. At early stage, the depth map has been widely studied in computer graphics and photography. Nowadays, the interest in the mapping of the distance has increased rapidly in robotics, motion planning, and autonomous driving cars.

1.2.1 Various Approaches to Depth Mapping

To obtain informations of surrounding environment, several different sensors are widely used. The most popular sensors include laser, radar, sonar and vision system. These systems have different advantages and disadvantages. Laser systems use laser to measure the distance to objects in the surrounding environment. It uses the time of flight of the coherent light reflected from objects[8]. Laser systems, sometimes also called Lidar, are usually most accurate in the measurement of the distance and has the most wide range of coverage in terms of the distance. However, laser sensor systems are usually expensive and is not easy to integrate into existing hardwares systems. Radar and sonar, similar to Lidar, makes the measurement of objects using reflection travel time, and they are somewhat cheaper than laser sensor systems. However, their coverage is usually much shorter than the laser system due to the fact that radio wave or ultrasonic waves are less coherent than lasers and more prone to diffraction and obstacles[6]. These systems are also hard to integrate into the existing devices.

1.2.2 Vision System

Among the various approaches to the three dimensional environment reconstruction systems, the most widely studied systems are vision system. Vision systems, which relies on the commercial cameras, compared to other sensor based systems, are the most cost effective and most easy to integrate into existing hardware systems, such as automobiles.

Structure from Motion

Vision systems can be classied into two sub-groups, based on the approach, which are Structure from Motion and Depth from Focus/Defocus. Structure from motion, sometimes also called SLAM(Simultaneous localization and mapping), is a process of estimating three-dimensional structure of environment from two-dimensional image sequence obtained from motion of the camera. It couples the local motion of camera to different images, and estimates the motion by matching the charateristic points, like SIFT or SURF, which are pixels that can be robustly matched[21]. Using the characteristic pixel correspondances, one can make estimation about how other pixels would have transformed between motion, and from which one can build the distance map of those pixels. Typical process of depth mapping procedure for structure from motion is given in figure 1-1.

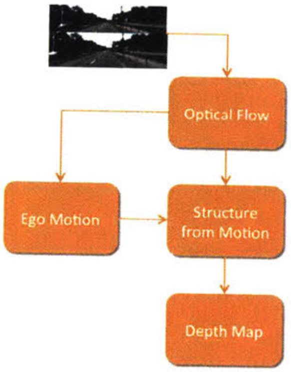


Figure 1-1: Sub-procedures in structure from motion algorithm. Ego-motion estimate refers to estimating the motion of camera and optical-flow refers to pixel matching problem.

Most of the structure from motion algorithms use stereo-vision, because the pixel matching problem becomes easier compared to mono-vision systems. In addition, the distance between the camera can be used as the reference distance that could be used to convert the pixel distance into actual physical distance. Mono-vision systems usually suffers from the lack of such reference distance, problem also known as scale drift[23]. Due to these advantages, stereo-vision systems are most widely studied, but to make the system more cost effective, mono-vision systems are gaining more and more interest.

Structure from motion has already been widely applied in robotics[15], motion planning[20], and autonomous driving[5], and couples well with these application due to the fact that these applications inheritently involve motion. However, due to the fact that algorithm relies on the motion of the camera, structure from motion algorithm cannot

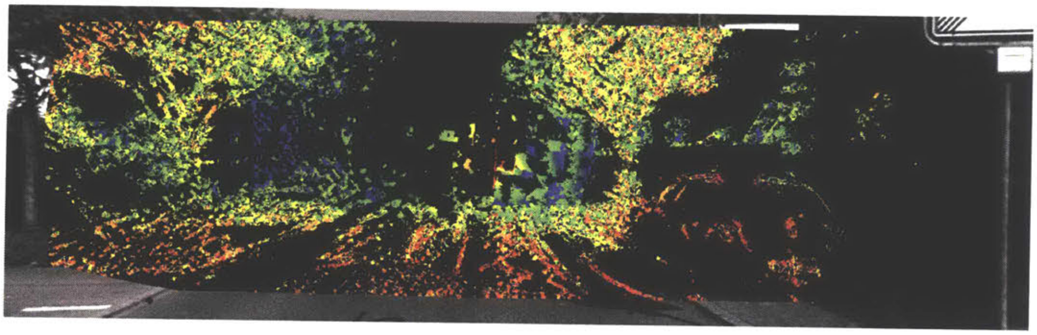


Figure 1-2: Depth map obtained using structure from motion for mono-vision system. KITTI dataset has been used for testing the implementation.

give any information about surrounding environment when the camera is stationary or slowly moving. In addition, structure from motion algorithms usually suffers from the long runtime, because most of the structure from motion algorithms are optimization problem, which are known to take long time and hard to scale to solve larger problem[11].

Depth from Focus/Defocus

Depth from Focus refers to the class of algorithms that measures the degree of focus to estimate the distance to individual pixels. By measuring the degree of focus, one can determine at which optical setup the pixel is best focused, and use that information to estimate distances. The advantage of depth from focus/defocus algorithm compared to structure from motion, is that it does not require the camera motion, therefore can be used for stationary environment. In addition, the depth from focus algorithms are usually easy to scale and are eay to reduce the runtime. In our work, we will utilize the depth from focus algorithm to build the depth map. More detailed analysis of depth from focus/defocus will be discussed in Chapter 2.

1.3 Active Lens

In this thesis, we will use depth from focus to build a depth mapping system. Depth from focus algorithms require images taken from different lens parameters, i.e. focal length. Active lens provides effective method to change the optical parameters of our system. In this section, we will look at how active lens operates, and how it can be controlled.

1.3.1 What is Active Lens

Lens is an optical device that affects the focus of light. It is composed of material with various index of refraction. Lens is defined by its focal length, which characterizes how much the light is refracted. Lens is an integral part of any modern optical system and is used everywhere from everyday camera device to complicated optical system.

Most of the lens are usually built from glass or solid material with refractive properties. Therefore, once it is built, it's optical characteristics cannot change. Therefore, to change the focal length of optical systems, either combinations of different lens are used, or has to be replaced by other lens with desired focal legnth. Thus, it becomes hard to change the focal length in a high speed applications due to these issues.

Active lens is a lens that has a variable focal length. It is consisted with a liquid material that has refractive properties. The liquid material is sealed inside the elastic polymer membrane. Around this polymer membrane, there is an electromagnetic actuator with coil that is used to exert pressure on the membrane, which changes the curvature of the membrane and liquid material inside. Therefore, the amount of the current applied will dictate the focal length of the lens[17]. More detailed design of the active lens EL-10-30-C series by optotune is shown in figure 1-3.

For the active lens system EL-10-30-C, the focal power is calibrated such that it is proportional to the amount of current that is applied to the coil in the electromagnetic actuator.

1.3.2 Controlling Active Lens

The active lens can be controlled manually to have different focal length, and optotune provides a communication protocol for developers to build custom software.

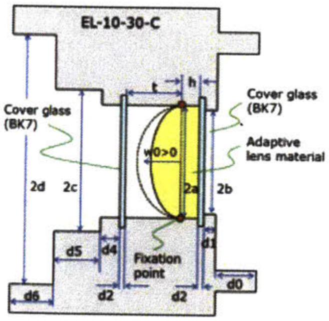


Figure 1-3: Structure of Active lens EL-10-30-C from optotune manual[17].

Communication with the active lens is done over a USB cable, and one can send proper set of signals to change the focal length or the driving mode of the signal. Optotune lens provides various different driving mode as well as lens calibration with fast response time, which allows the developers to build high-speed applications with active lens[16]. For our application, we will use the focal power mode to control the active lens so that our overall focal plane is located at specific location.

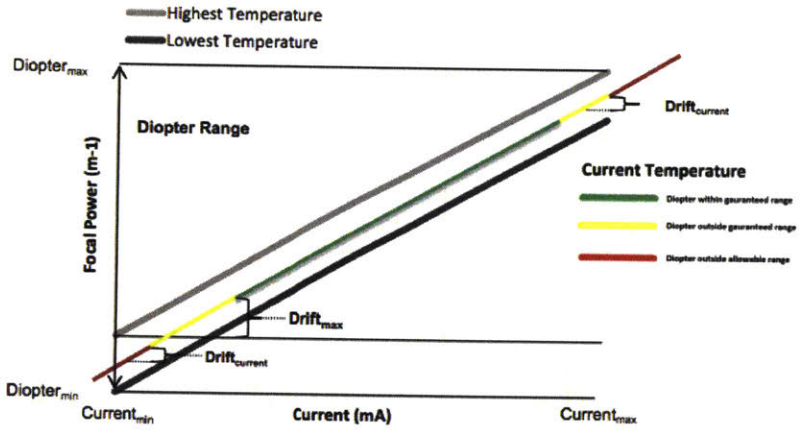


Figure 1-4: Relation of focal power of the lens with amount of current applied. Focal power mode of optotune lens provides drift compensation from the temperature change[16].

More detailed explanation of the communication protocal can be found at the optotune Lens Driver 4 manual[16] and the code for driving the lens can be found in the appendix A.

Chapter 2

Depth from Focus and Focus Measure Algorithm

Depth estimation has been an important problem in the computer vision, as a dimension is lost in the image acquisition process of projecting three dimensional world to two dimensional image. However, by varying the camera parameters, we can acquire different set of images that has different information about the scenery. Depth from Focus/Defocus referes to the set of algorithms that measure the distance to various objects of the scenery from the images obtained using different focal setting.

2.1 Depth from Focus

2.1.1 Depth from Focus Algorithms

Depth from Focus refers to the class of technics that uses the degree of focus to estimate the distance to individual pixels. By measuring the degree of focus, one can determine which optical parameters give the best focus for each pixel, and use that information to estimate the distance.

It is known for decades, that the optical system can be modelled as a linear system as the lens itself can be understood as a linear operator[12]. As a result, when image of point is formed on the sensor, one can understand the optical system to be a transfer function that is radially symmetric, called point-spread function(PSF). The point-spread function depends on a distance from a point to lens, focal length of the lens and the aperture of the optical system. Therefore, when one can make estimation about the structure of point-spread function of camera system, one can also estimate the distance to the point when other optical parameters are known.

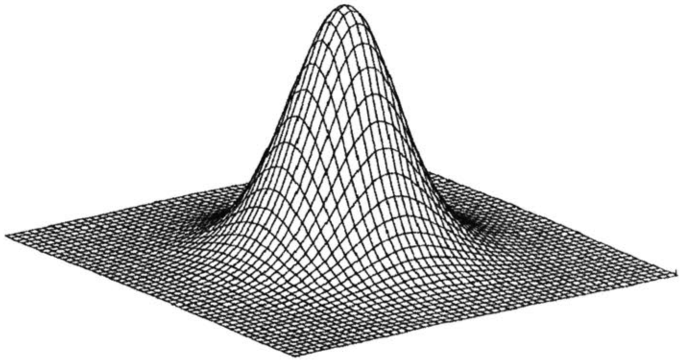


Figure 2-1: Two dimensional gaussian model of Point Spread Function.

Because we can model the optical lens system as a linear system, frequency analysis can be used to understand the process of imaging from scenery. Essentially, one can understand the process of defocusing as a low-pass filter in the frequency domain. Thus, by analyzing the degree of the spread using fourier transform, we can study the effect of various camera parameter on the resulting image.

At the earliest development of the depth mapping algorithms, many of the literatures focused on the passive depth mapping system. Early work done by Petland[18] showed that using images from different camera settup can provide the informations about the distance to each pixels. In his paper, he showed that by using a two different camera with one long depth of field lens, and the other with short depth of field lens one can extract the depth information of individual pixels. In the analysis of degree of focus, Petland used focal gradient or Laplacian operator to measure how well individual pixels are focused. Other approaches include informational entropy analysis of two different images from different optical setup[1] and etc. Some of the more recent researches also showed that additional structure to the lens allows one to recover the depth information with mono-vision system, with one image as well[10].

2.1.2 Limitation of Depth from Focus: Depth of Field

Many of the depth from focus algorithm relies on the fact that the degree of focus will change depending on the relative distance between focal plane and location of the object. In idealized thin lens model, we assume that the image will only be perfectly focused when focal plane and distance to scenery is exactly same. In such model, the underlying assumption is that the pixel size of the image sensor in camera system has infinitly small radius, and therefore, image will only be focused when the image of the point source on camera sensor is also an infinitely small point. However, in real camera, image sensor has a finite size and cannot be infinitely small. Due to finite size of the pixel on the image sensor, there is a region in space where object in various distances are all focused. Such region is called a depth of field, and will depend on the intrinsic camera parameters.

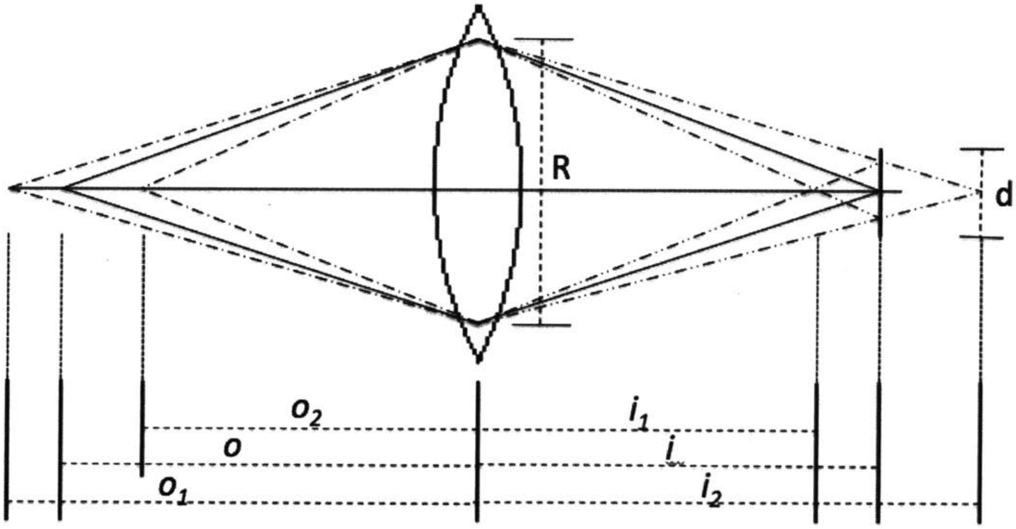


Figure 2-2: Thin Lens model with aperture R, sensor diameter d.

Assume a thin lens model, with a finite pixel size d on camera sensor. The camera is settup with focal distance f, F-Number $N(= f/R)$. Using thin lens model, we can write the relation between distance of an object from lens o and distance of an image from lens i.

$$\frac{1}{i} + \frac{1}{o} = \frac{1}{f} \tag{2.1}$$

By varying the distance of the object around o, the image of the point source at distance o_1 will at a location i_1. However, if the image that is formed at different distance is within the radius of pixel, resulting image will still be percieved as a point. Sometimes the radius of this permissible circle is called the circle of confuision. Geometrically, we can determine the range of image location i_1 and i_2 that is still percieved as a point at the camera sensor.

$$\frac{i_2 - i}{i_2} = \frac{d}{R} \rightarrow i_2 = \frac{i}{1 + d/R} \tag{2.2}$$

$$\frac{i - i_1}{i_1} = \frac{d}{R} \rightarrow i_1 = \frac{i}{1 - d/R} \tag{2.3}$$

Again, using thin lens equation, the object distance o_1 and o_2 satisfies following.

$$\frac{1}{i_1} + \frac{1}{o_1} = \frac{1}{f} \tag{2.4}$$

$$\frac{1}{i_2} + \frac{1}{o_2} = \frac{1}{f} \tag{2.5}$$

Combining above equations, we can write o_1, o_2 as,

$$o_1 = \frac{of^2}{f^2 + dN(o - f)} \tag{2.6}$$

$$o_2 = \frac{of^2}{f^2 - dN(o - f)} \tag{2.7}$$

One can write depth of field DOF as the difference between o_1 and o_2.

$$DOF = \frac{2Ndo^2}{f^2}\left(\frac{o - f}{o}\frac{1}{1 - (\frac{dN(o-f)}{f^2})^2}\right) \tag{2.8}$$

For the limit where $f \ll o$, we can simplify the euqation.

$$DOF = \frac{2Ndo^2}{f^2} = \frac{2d}{Rf}o^2 \tag{2.9}$$

Therefore, every object in the region DOF around o will be in focus all together. The concept of depth of field is widely used in photography as well. In the field of photograpy, it is known that increasing F-Number N, therefore decreasing R, the aperture of the lens, depth of field increases, and entire image will be focused.

However, for the depth from focus systems, depth of field degrades the depth resolution of depth mapping system. Many of those system rely on the fact that at different distance, degree of focus will change. Since all objects in the region of depth of field will be focused all together, every system that relies on the degree of focus will suffer from the effect of depth of field and it imposes a limit on the accuracy of the distance measurement on such systems.

From above equation, one can observe that $DOF \propto o^2$, i.e. further objects suffer more from the effect of depth of field compared to the closer object, thus putting a limit on the coverage of depth mapping system. We can optimize the set up of the camera to minmize the effect of depth of field as much as possible, by increasing the aperture size R, or by decreasing pixel size d, to increase the coverage as much as possible.

2.2 Focus Measure Algorithms

In previous section, we have looked at various depth from focus algorithms and analyzed the limits on such systems. In this section, we will focus our attention on the depth from focus algorithms with focus measures, and compare their performances. Lastly, we will look more closely at the particular choice of focus measure algorithm, Variance of Wavelet Coefficient, that has been used for the implementation of our depth mapping system.

2.2.1 Focus Measure

Many of depth from focus algorithms can also be catagorized in a broader categroy of an algorithms called focus measure. Focus measure refers to the set of algorithms that measures the degree of the focus, which can be used to measure the degree of focus for either each pixel or entire image. These algorithms can be understood as an operator that takes the image, and returns a degree of focus for an individual pixels or for an entire image. Figure 2-3 is an example of applying the focus measure operator.

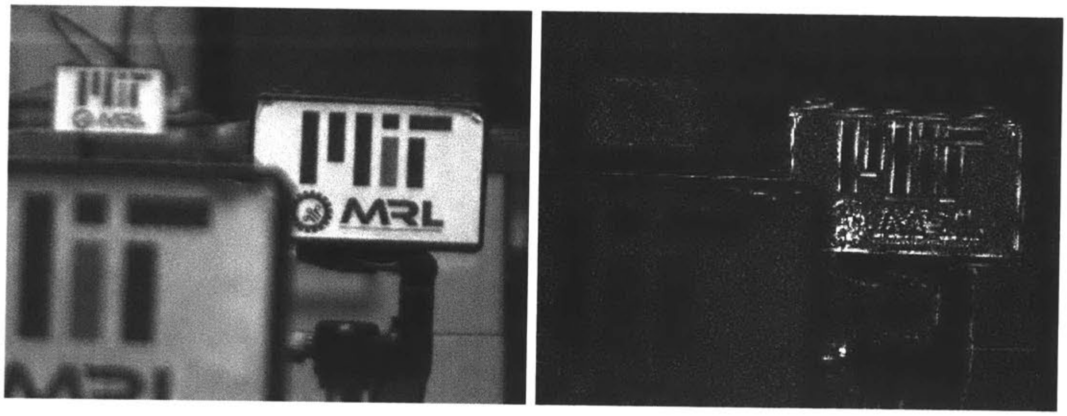


Figure 2-3: Example of Focus Measure operator for object at 2m distance. We have used Variance of Wavelet Coefficients.

Many of focus measure algorithms relies on the basic idea that well focused images typically have high spatial frequency component compared to blurry images. Images that are in focus represent sharper edges and contain more detailed texture compared to a blurry image. Using well known frequency analysis technic we can extract the amount of information contained in the high frequency domain and by comparing the high frequency content in the images with different focal length, we can make decision on at which focal length, the image is in focus.

This concept of focus measure algorithms make the procedure of depth estimation to be a simple search task, as now we can quantize the degree of focus, which allows one to find the depth by finding the best focused image and its optical setup.

2.2.2 Comparing Different Focus Measure Algorithms

For years, many focus measure operators has been introduced and analyzed. Several researchers have evaluated and compared different focus measure algorithms[9, 24, 14]. What most of the literature agree is that ideal focus measure algorithm is unimodal, monotonic, and reaches maximum only when the image or pixel is all focused[28]. The focus measure operators that have been introduced showed different performances in different settings. The paper by S. Petruz et al[22], attempted to make a thorough comparison of various focus measure algorithms in different settings and compared the performance of each focus measure operators. They grouped 36 different focus measure algorithms into a smaller categories, and showed Laplacian based and Wavelet based operator perform the best in various settings[22]. Based on their research, we were able to reduce the choice of which focus measure algorithm to use for our depth mapping system, and with in-house comparion of these measures, we decided to use the variance of wavelet coefficient as our focus measure. The result of such comparison is give in figure 2-4.

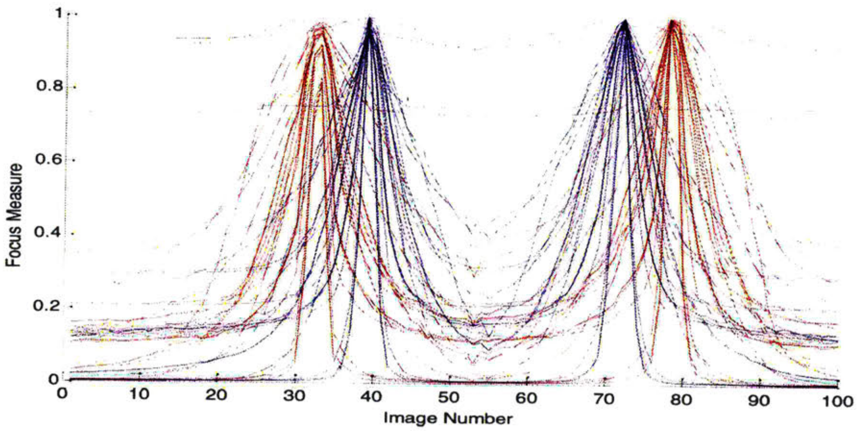


Figure 2-4: In-house comparison of 36 focus measures from S. Petruz. Experiment performed by I. Soltani. Variance of Wavelet coefficients shows the most distinct peak for the test scenery[7].

In the remainder of this chapter, we will look at what wavelet tranform is and how variance of wavelet coefficient is defined as a focus measure operator.

2.2.3 Discrete Wavelet Transform

Wavelet transform is an analysis of signal by representing it with a certain orthonormal series generated by wavelet. These wavelets, similar to sinusodial waves in fourier transform, constitutes the basis functions that compose the signal, and they are defined to be orthonormal to each other, similar to the sinusoidals. However, different to the sinusodial signals, the wavelets are defined to be a finite signal, with different length. Being finite length signal, decomposition by wavelets indicates decomposition by transient signals.

These orthogonal basis can be constructed by a scaled and shifted version of the mother wavelet $\psi(x)$.

$$\psi_{ij} = 2^{i/2}\psi(2^i x - j) \tag{2.10}$$

Because the basis for the decompositions are transient signals, wavelet decomposition often capture the transient behavior of the signal better than short-time fourier tranform. Especially, when the signal that one wish to capture from longer sequence of signal is known, it shows superior performance compared to other types of frequency analysis technics. There have been several literatures on different wavelets, and one of the most widely used wavelets are Daubechies wavelets, family of wavelets found by I. Daubechies in 1988. Daubechies wavelets have maximal number of vanishing moments and has finite length[3].

For discrete signals, wavelet transform can also be understood as an octave-band filter banks. The filter bank refers to an array of band pass filters that separates the signal into frequency sub-band. Short-time fourier transform, which sometimes is also refered to as time-dependent discrete fourier transform, is type of filter bank that separates the signal into equally sized frequency sub-bands. However, for the discrete wavelet transform, the sub-bands do not have equal size, but rather changes depending on the frequency. To be more specific, at each stage of analysis, the low-pass band is splited in half, resulting in octave-band decomposition. In addition, the set of wavelet filters can be built to satisfy perfect reconstruction condition, which indicates that information is not lost in the process of decomposition and therefore has equal amount of information as the origianl signal.

The advantage of the wavelet transform over fourier transform becomes evident when we compare their sub-band structure. As in figure 2-5 fourier transform has equal resolution for every frequency band, therefore, does not contain any information about the time locality. However, wavelet tranform's resolution in frequency domain increases for low frequency band, and decreases for high frequency band. The change

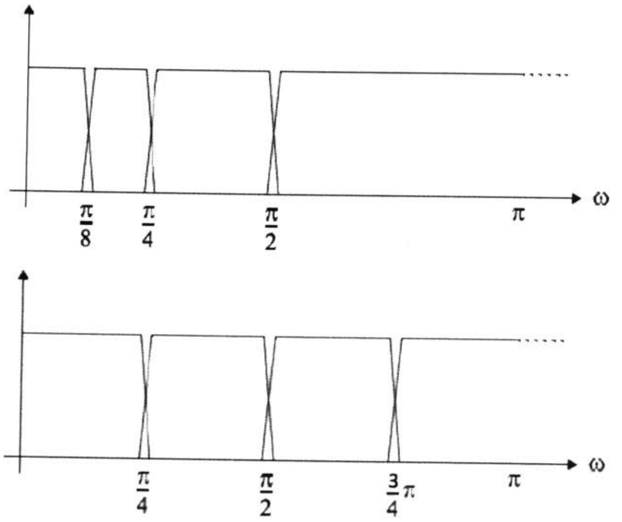


Figure 2-5: Octave band decomposition in discrete wavelet transform compared with short-time fourier transform. Frequency resolution increases for lower frequency band.

in the resolution depending on the frequency band provides information about the time locality of the discrete time signal, thus capturing transient behaviors in the signal better than fourier transform. Discrete wavelet transform can be implemented using a cascade of filters as in figure 2-6. Similarly, for two-dimensional signal, one can decompose the signal as in figure 2-7.

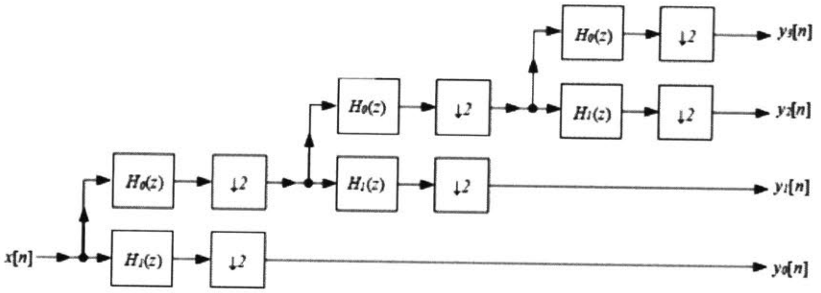


Figure 2-6: Discrete wavelet transform as a cascade of convolution.

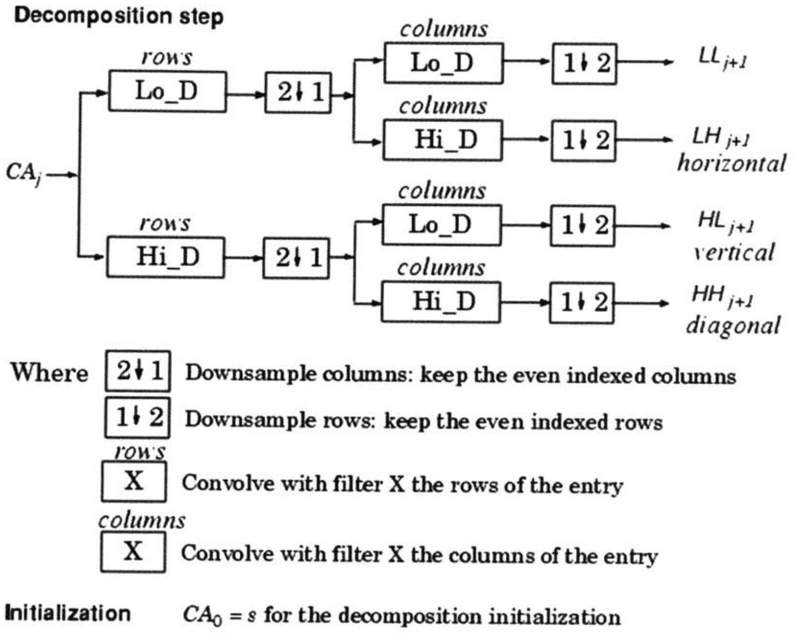

Decomposition step
rows
Lo_D
2↓1
columns
Lo_D
1↓2
LL_{j+1}
columns
Hi_D
1↓2
LH_{j+1} horizontal
CA_j
rows
Hi_D
2↓1
columns
Lo_D
1↓2
HL_{j+1} vertical
columns
Hi_D
1↓2
HH_{j+1} diagonal
Where 2↓1 Downsample columns: keep the even indexed columns
1↓2 Downsample rows: keep the even indexed rows
rows X Convolve with filter X the rows of the entry
columns X Convolve with filter X the columns of the entry
Initialization $CA_0 = s$ for the decomposition initialization

Figure 2-7: Two dimensional discrete wavelet transform as a cascade of convolution[25].

2.2.4 Variance of Wavelet Coefficients

Variance of Wavelet Coefficient is a focus measure operator that reflects the strength of high-frequency detail in the image. It computes the variance of the wavelet transform coefficient in high frequency sub-band, and therefore can be understood as summation over the window of high frequency energy[4].

For a given image $I(x, y)$, consider the window E of size w by l. Then, one can consider it's window in corresponding wavelet decomposition in two dimension for 1^{st}-level LH, HL, and HH. Denoting each decomposed image as W_{LH}, W_{HL}, and W_{HH}, one can write variance of wavelet coefficients as follows.

$$FM = \frac{1}{wl}[\sum_{(i,j)\in E_{LH}} (W_{LH}(i,j) - \mu_{LH})^2 \tag{2.11}$$

$$+ \sum_{(i,j)\in E_{HL}} (W_{HL}(i,j) - \mu_{HL})^2 \tag{2.12}$$

$$+ \sum_{(i,j)\in E_{HH}} (W_{HH}(i,j) - \mu_{HH})^2] \tag{2.13}$$

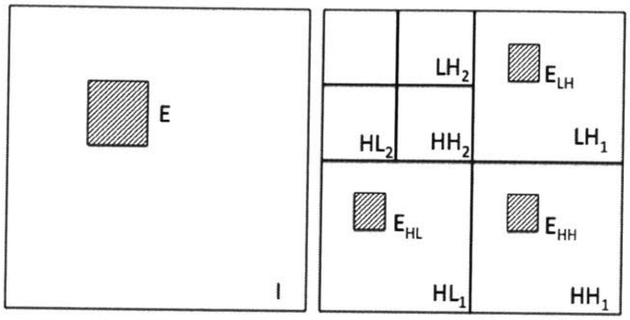


Figure 2-8: Variance of Wavelet Coefficients.

One can generalize this to further level of decomposition by applying smaller window at every level. At n^{th}-level decomposition, the effective window size is given by $[\frac{w}{2^n}, \frac{l}{2^n}]$. As the degree of focus increase in individual pixel window, variance of the wavelet coefficients will increase as the high frequency content in each window will increase and reach maximum when the pixel is all-focused. One can also change the window size to control the resolution of the resulting depth map.

Chapter 3

Depth Mapping Algorithm and Active Lens

In previous section, we have showed that variance of wavelet coefficients can be used to measure the degree of focus for individual pixels, and how it is computed. Using this focus measure operator, one can show at which focal distance, each pixel becomes in focused, and determine the distance to the object where that pixel belongs. In this section, we will study how one can build depth mapping system using focus measure operator and active lens. In addition, we will investigate what noise filtering criteria can be used to eliminate the noisy data points to acquire reliable depth map of the scenery.

3.1 Focus Measure and Depth Mapping

3.1.1 System Design

Basic setup of our system consist of basic camera system attached with normal optical lens and active lens attached in front of the lens. Entire optics system is mounted on a tripod to acquire images of the scenery. We have used Basler camera acAC640-750um, which allows upto 751fps capturing, and normal lens with focal length of 25.2mm. We can calibrate the settings of our optical system to minimize the effect of depth of field. In this section we will generalize the depth of field that was discussed in Section 2.1.2 to our optics setup of one lens system with active lens attached. We can model our system as two thin lens system as given in figure 3-1. Assume that the object is at distance i from the active lens and, and active lens is calibrated such that it is perfectly focused. From the calibration of the active lens, we get the following equation.

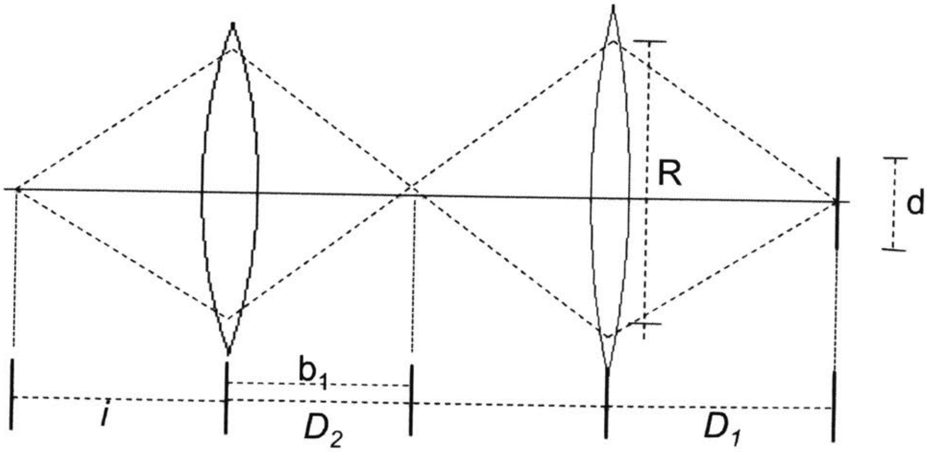


Figure 3-1: Modelling the active lens system. The distance between camera sensor and ordinary lens is D_1 and distance between two lenses are D_2.

$$\begin{gathered}\frac{1}{i} + \frac{1}{b_1} = \frac{1}{f_2} \\ \frac{1}{D_2 - b_1} + \frac{1}{D_1} = \frac{1}{f_1}\end{gathered} \tag{3.1}$$

Above equation tells us that if active lens is calibrated such that the focal plane is located at i, the image of point source at the distance i from the optics system, will be a point on the image sensor as well. The variable b_1 is a dummy variable in the intermediate step where the image from first lens(i.e. the active lens) is formed.

Now let's move our object to distance o such that image will not be perfectly focused. We can write similar relation between the new location of object, and the location where the image of an object is formed.

$$\begin{aligned}\frac{1}{o}+\frac{1}{b_2}&=\frac{1}{f_2}\\ \frac{1}{D_2-b_2}+\frac{1}{a}&=\frac{1}{f_1}\end{aligned} \tag{3.2}$$

The difference of the equation 3.1 and equation 3.2, is that because in general the distance to the current focal plane i and the distance to object o is different, image of the object will not be formed on the image sensor (i.e. D_1) but will be at some location a. In theory, object in new distance will not be perfectly focused and it's energy will be spread over a circular radius r. However, due to the finite size of pixel on the image sensor, if r is smaller than the size of pixel d, image on the sensor would still be perceived as a point, and therefore will seemed to have been all focused.

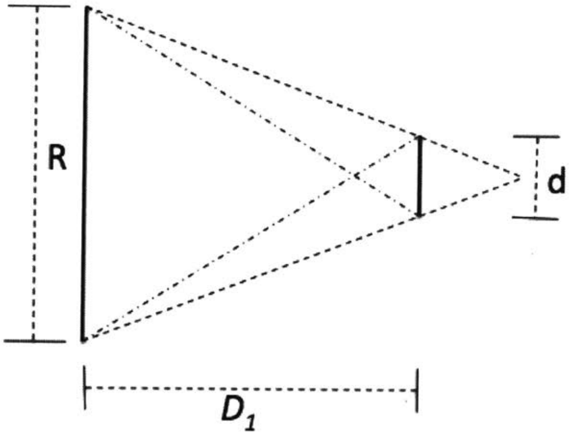


Figure 3-2: Triangle similarity relation.

We can derive the allowed distance a in relation to pixel size d using simple geometry.

$$d=R|\frac{D_1-a}{a}|=R|(\frac{D_1}{a}-1)|\rightarrow\frac{1}{a}=\frac{1}{D_1}(1\pm\frac{d}{R}) \tag{3.3}$$

Now, let's write the radius of the point spread function d in terms of camera parameters (f_1, D_1, D_2) and new object distance o and original object distance i. We start by eliminating the dummy variable b_1, b_2.

$$\begin{aligned} \frac{1}{b_1} = \frac{1}{f_2} - \frac{1}{i} = \frac{i - f_2}{i f_2} &\rightarrow b_1 = \frac{i f_2}{i - f_2} \\ \frac{1}{D_2 - b_1} = \frac{1}{f_2} - \frac{1}{D_1} = \frac{D_1 - f_2}{D_1 f_2} &\rightarrow D_2 - b_2 = \frac{D_1 f_2}{D_1 - f_2} \end{aligned} \tag{3.4}$$

Adding the two equations, we get the following relationship between f_2, i.

$$D_2 = \frac{i f_2}{i - f_2} + \frac{D_1 f_1}{D_1 - f_1} \rightarrow \frac{1}{f_2} = \frac{1}{i} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}} \tag{3.5}$$

We can write a similar relation for the object distance o and its image location a by eliminating the dummy variable b_2.

$$\begin{aligned} \frac{1}{b_2} = \frac{1}{f_2} - \frac{1}{o} = \frac{o - f_2}{o f_2} &\rightarrow b_2 = \frac{o f_2}{o - f_2} \\ \frac{1}{D_2 - b_2} = \frac{1}{f_2} - \frac{1}{a} = \frac{a - f_2}{a f_2} &\rightarrow D_2 - b_2 = \frac{a f_2}{a - f_2} \end{aligned} \tag{3.6}$$

Again, adding above two equations,

$$D_2 = \frac{o f_2}{o - f_2} + \frac{a f_1}{a - f_1} \rightarrow \frac{1}{o} = \frac{1}{f_2} - \frac{1}{D_2 - \frac{a f_1}{a - f_1}} \tag{3.7}$$

Now replacing $\frac{1}{a}$ and $\frac{1}{f_2}$ from above, we get the following result for the distance o.

$$\begin{aligned} \frac{1}{o} &= \frac{1}{f_2} - \frac{1}{D_2 - \frac{a f_1}{a - f_1}} \\ &= \frac{1}{i} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}} - \frac{1}{D_2 - \frac{a f_1}{a - f_1}} \\ &= \frac{1}{i} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}} - \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1 \pm f_1 \frac{d}{R}}} \\ &= \frac{1}{i} + \frac{\frac{D_1 f_1}{D_1 - f_1}\left(\frac{\pm \frac{f_1}{D_1 - f_1} \frac{d}{R}}{1 \pm \frac{f_1}{D_1 - f_1} \frac{d}{R}}\right)}{\left(D_2 - \frac{D_1 f_1}{D_1 - f_1}\right)\left(D_2 - \frac{D_1 f_1}{D_1 - f_1}\left(\frac{1}{1 \pm \frac{f_1}{D_1 - f_1} \frac{d}{R}}\right)\right)} \end{aligned} \tag{3.8}$$

The relation given in above equation seems quite complicated, however, we can make a guess that the difference $\frac{1}{o} - \frac{1}{i}$ itself is independent of object's distance i or the new distance o. In addition, the D_2 dependence of $\frac{1}{o} - \frac{1}{i}$ is simply inversely quadratic. Therefore, we can get smallest difference between i and o, i.e. depth of field when D_2 is very large.

The dependence of depth of field on D_1 however, is very complicated. Finding a closed form solution of D_1 that maximizes the difference between i and o can be done by differentiating the difference by D_1. Setting the derivative to be 0, we get the following value of D_1.

$$D_1 = \frac{D_2 f_1}{D_2 - f_1}\sqrt{1 \pm \frac{d}{R}} \tag{3.9}$$

Thus, to minimize the effect of depth of field, we can set lens to sensor distance to be 49mm and lens to active lens distance to be as far as it can be.

3.1.2 Depth Mapping with Active Lens

After optimizing the optical setup, we can now use active lens to construct the depth map of the scenery. Basic idea of depth mapping with active lens is by changing the focal power of the active lens, each pixel points wii be in focus at different instance and from which we can determine the distance to the object where that pixel belongs.

The active lens allows us to change its focal power by communicating over USB connection. By sending proper signal to the active lens, we can change it's focal length, and change the overall focal power of entire optical system. To build a depth map, we would like to capture images with focal plane at equally spaced distances. Again, using the thin lens model, from figure 3-1, we see that following expression holds.

$$\begin{gathered} \frac{1}{i} + \frac{1}{b} = \frac{1}{f_2} \\ \frac{1}{D_2 - b} + \frac{1}{D_1} = \frac{1}{f_1} \end{gathered} \tag{3.10}$$

We can eliminiate b, and get following formula.

$$\begin{aligned}\frac{1}{f_2} &= \frac{1}{i} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}} \\ &= \frac{1}{i} + \beta \end{aligned} \tag{3.11}$$

i.e. required focal power of the active lens is inversely proportional to the distance of required focal plane plus some constant that is dependent on the camera calibration. Therefore, to obtain images with focal plane at equal distances, we need to set the focal power to change inversely proportional to distances. We have calibrated the parameter β from equation 3.11 of the system so that every image we obtain has a focal plane located at every 5 cm starting from 50 cm to 10 m.

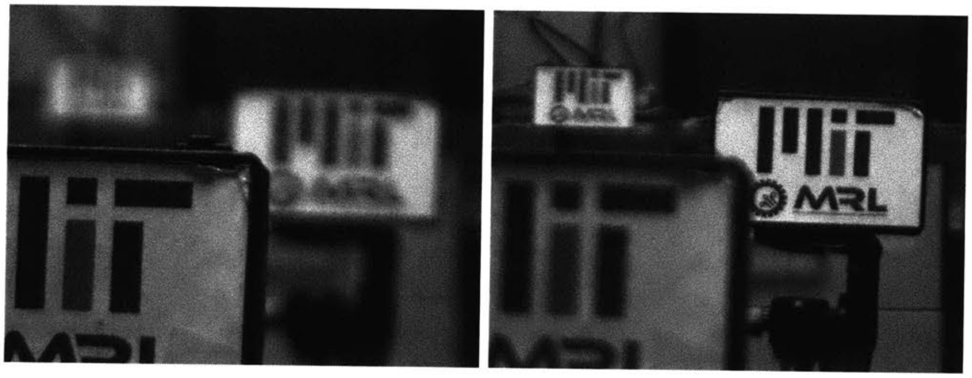


Figure 3-3: Images of three objects with different focal plane location. Image on the left is focused at 1m, where as the image on the right is focused at 2m

To test our depth mapping system, we have placed 3 MIT logos at 1m, 2m and 4m away from the lens. The set of images captured will be defocused based on the location of focal plane as in figure 3-3.

After capturing these images, we apply variance of wavelet coefficient operator for each images, which will measure the degree of focus for each pixel. Typical signal we find will show well localized maximum as in figure 3-4. At the image index where the pixel is in focus will show maximum degree of focus, and we can find the distance to pixel based on the image index. We can perform this operation for every pixel, and get prilimiary depth map as in figure 3-5.

The basic idea of finding the maximum among focus measures of different focal setting was studied here at Mechatronics Research Laboratory at MIT, and to show the proof of concept, depth mapping system was built using kurtosis and smoothness index as an error criteria using Labview[7].

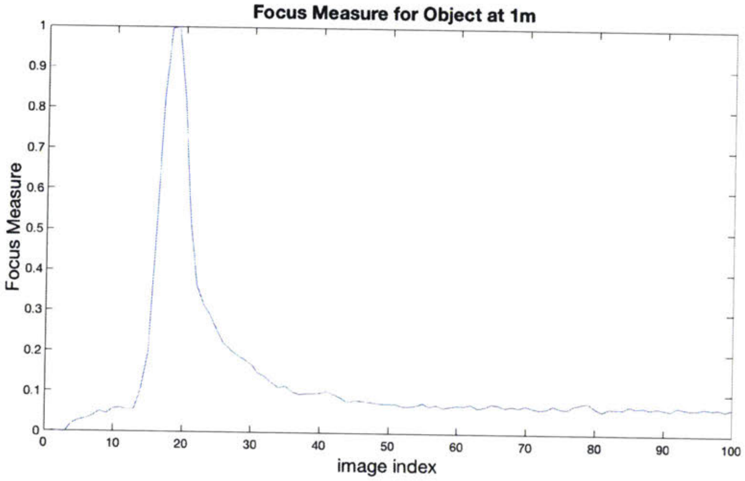


Figure 3-4: Change in energy in high frequency band. Object is located 1m away from the lens.

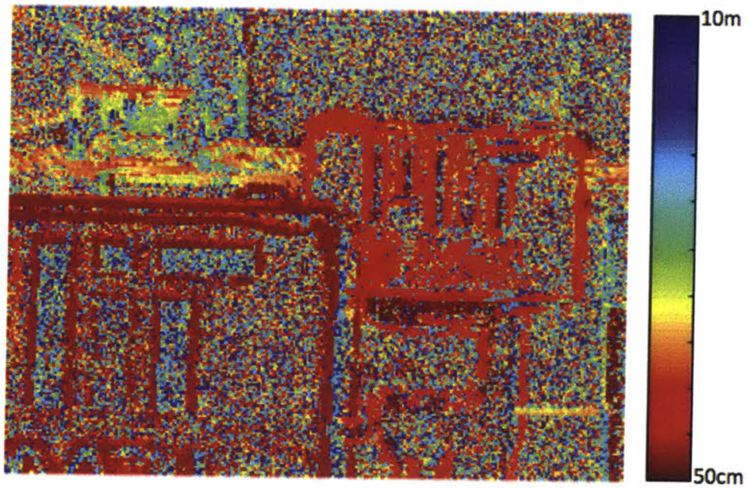


Figure 3-5: Prilimary Depth Map from finding the maximum of Focus Measure. Colorbar on the right indicates the coverage of Depth from 50 cm to 10m. Edges which has relatively high frequency energy show clear depth information.

3.2 Improving Depth Mapping Algorithm

In previous section, we showed how the active lens is used in our depth mapping system, and the process of building depth map. In this section, we will discuss some of the methods that could improve our system; Heisenberg Uncertainty criteria, Second level decomposition and variable threshold.

3.2.1 Filtering Noise

Prilimary depth map shows a good performance for finding depth informations, but also contain various noisy pixels. These pixels are the points that don't have enough texture so that the defocusing does not change the underlying energy content of pixel. To build a reliable depth map, we need to filter these points and label them to be unknown.

Noise Criteria

To distinguish noise to signal, one could use well-known statistical measures, such as Kurtosis, Standard deviation, or Signal-to-Noise ratio. In house experiment on these measures has been tested using Kurtosis, Standard deviation and smoothness index[7]. However, due to the fact that these measures are statistical measures, it ignores the structure or the shape of the signal, and usually does not perform well as in figure 3-6. Therefore we need a better error criteria to filter out the noisy points.

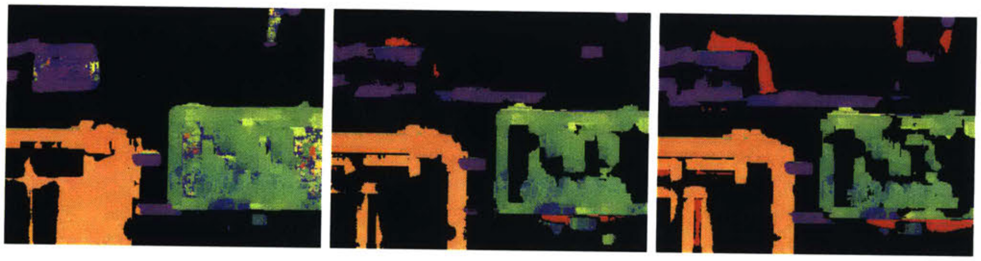


Figure 3-6: Depth Maps from using other criterias. From left, noise criteria used are Standard Deviation, SNR, Kurtosis. Calibration and resolution is different.

Key idea to find a better criteria is to observe that signal and noise show very different behavior over image index. From figure 3-7, one can observe that the signal, compared to the noise are usually well localized. In other word, focus measure over the image index shows very trasient behavior, such that it will show sharp transition around the peak, whereas the noise is more spread out over the image index.

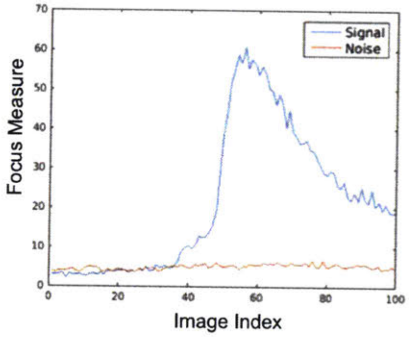


Figure 3-7: Typical change of energy in high frequency band. Signal shows clear localization compared to noise.

Therefore, one can use a criteria that measures how well localized the signal is to distinguish signals from noise. To measure the locality of the signal, we can borrow the idea of locality from Quantum Mechanics, and Heisenberg Uncertainty principle[4].

$$\sigma^2 = \frac{\int_{-\infty}^{+\infty}(x-x_0)^2 f(x)^2}{\int_{-\infty}^{+\infty} f(x)^2} \tag{3.12}$$

Heisenberg Uncertainty principle defines how one can measure 'locality' of the signal around the point x_0. One can also understand above equation as a quadratic cost function around x_0, i.e. the more spread out the function is around x_0, higher the uncertainty is. When the function is all focused at one point, for example a dirac delta function $f(x) = \delta(x - x_0)$, uncertainty measure is 0. Also using functional analysis, one can show that uncertainty measure is maximized for the case when the function $f(x)$ is maximally flat.

Therefore, one can understand that above definition of uncertainty will measure the locallity of the signal well, and can be used as an error criteria to distinguish signal

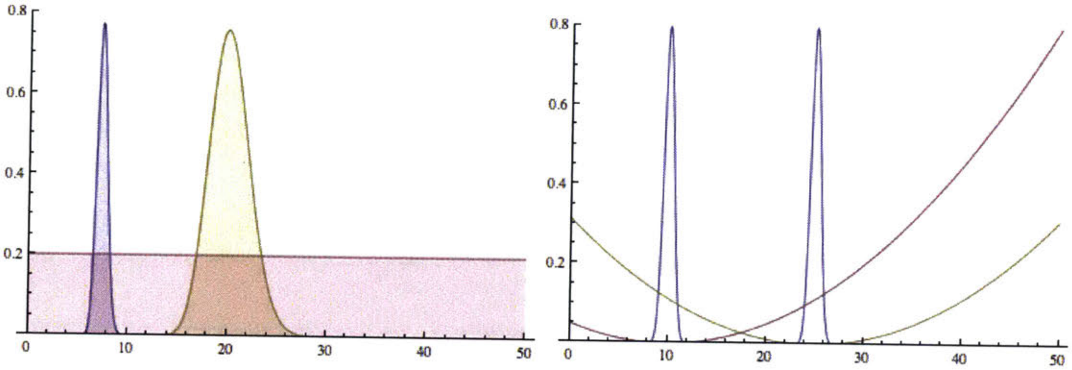


Figure 3-8: Different signals and their locality.

to noise for our system.

Utilizing Heisenberg Depth Resolution

In order to use depth uncertainty criteria for our system, one must convey the fact that integration will be done in discretized finite space instead of continuous infinite space. Due to the fact that the integration space is finite, the cost function $(x - x_0)^2$ becomes asymmetric, and cause a bias for the values that are close to the boundary. One could mitigate such problem by normalizing the cost function as well. Therefore, the discretized version of uncertainty measure for the focus measure FM_i can be written as follows.

$$\sigma^2 = \frac{\sum_i (i - i_0)^2 FM_i^2}{\sum_i FM_i^2 \sum_i (i - i_0)^2} \tag{3.13}$$

Applying this criteria for our depth mapping system, we get the depth map as in figure 3-9.

We can observe that Heisenberg Uncertainty criteria performs much better compared to the statistical measures, such as Kurtosis, Signal-to-Noise ratio, in both coverage, and false positive ratios. From now on, we will refer to the system with Heisenberg Uncertainty criteria combined with finding maximum location, as Heisenberg Uncertainty system, and the error measure as Heisenberg Depth Uncertainty.

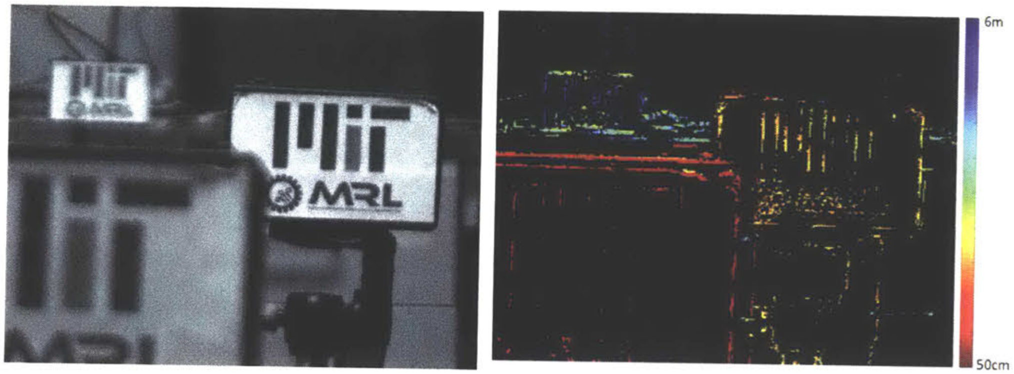


Figure 3-9: Resulting Depth Map of applying Heisenberg Depth Uncertainty. Depth scale has been reduced to 6m so it is easier to compare the different objects visually.

3.2.2 Second Level Decomposition

First level wavelet decomposition is equivalent to have a high pass filter with cutoff frequency around $\pi/2$. We can therefore, think about improving the system by extending the cutoff so that low textured pixels will contribute to the energy, which may allow us to find depth information for some of the pixels that were unknown.

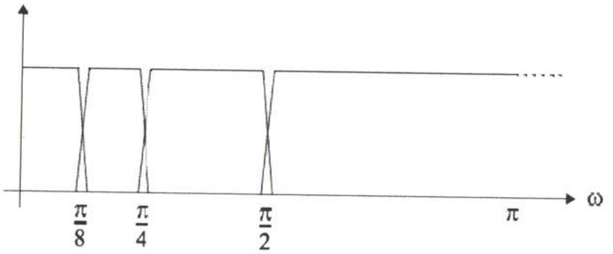


Figure 3-10: Octave band decomposition in discrete wavelet transform.

In wavelet decomposition, it is quite simple to extend the filter to have a lower frequency cutoff. In frequency domain, wavelet decomposition is an octave band filter bank, and we can easily decrease the frequency cutoff by extending to lower levels of decomposition. In addition, because the wavelet filters are orthogonal to each other, any information we gain from lower level decomposition will not overlap with the information that is currently known, and therefore will only improve the system in terms of the number of pixels that we can recover the depth information.

Thus, to improve the detection rate of our system, we can extend the frequency cutoff by including second level wavelet decomposition. Extending the filter to lower levels can be implemented quite simply by applying the first level decomposition recursively to lower levels. Resulting depth map from including second level decomposition is given in figure 3-11.

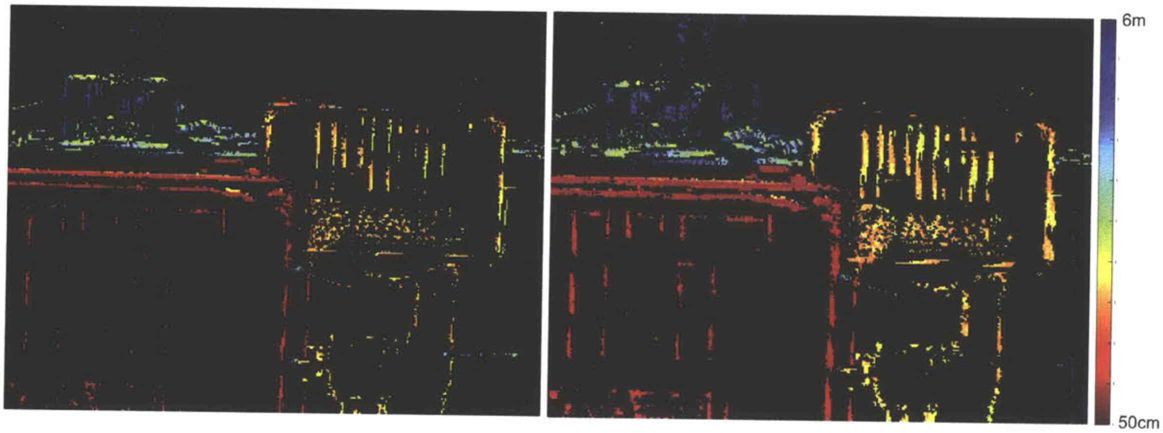


Figure 3-11: Depth map including second level decomposition. One can observe that more pixel depth are recovered.

From figure 3-11, one can easily observe that the detection rate of pixels vastly improved from single level decomposition. Considering same error threshold for Heisenberg Uncertainty criteria, the detection rates improves from 7.8% to 15%.

We can also draw a histogram of depth distribution for both single level decomposition and second level decomposition. The histogram, of the depth map we obtain, where x-axis corresponds to the depth of each pixel, and y-axis corresponds to the number of pixels within that depth bucket is given in figure 3-12.

Histogram analysis shows that the number of pixels that we recover from second level decomposition almost doubles compared to the single level decomposition. In addition, one can also see that the peaks in the histogram, which corresponds to the three objects in the system, have become more distinct for the second level decomposition compared to single level decomposition.

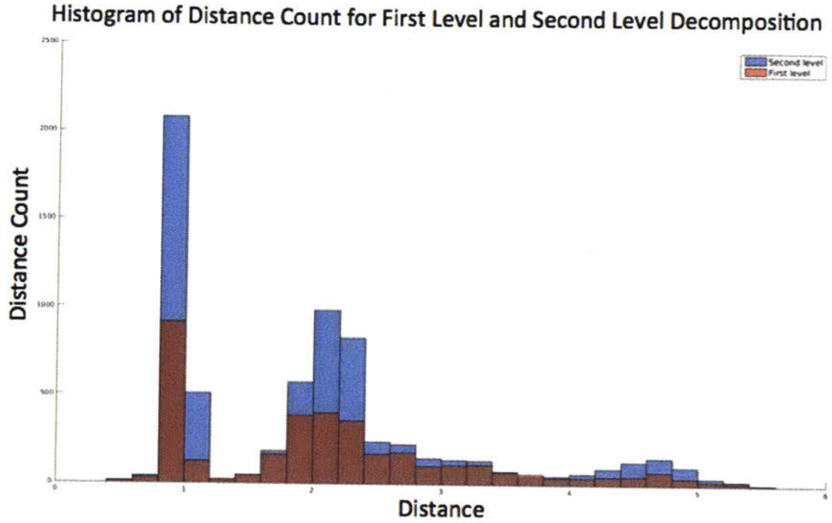


Figure 3-12: Histogram of pixel depths recovered from first level and second level decomposition. The number of pixels that are recovered doubles by incorporating the second level decompositions.

3.2.3 Variable Threshold for Heisenberg Uncertainty Depth Resolution

In previous section, we have seen that adding second level decomposition enhances our depth mapping system, by increasing the detection rate for the pixels, and producing depth information for more pixels. We can further improve this system by applying variable threshold for Heisenberg Depth Uncertainty.

To test how focus measure changes as the object is located further from the lens, we have placed an object at several distances and measured the change in focus measure. From the figure 3-13, the experiments on same objects on different distances, one may find that the locality of the energy distribution over the image index degrades as the object is futher from the camera. This is natural result of the depth of field; further objects become harder to distinguish. This fact natrually leads to the idea of variable threshold for Heisenberg Uncertainty criteria. Further objects would naturally have a worse localization compared to closer objects, and thus will be more likely to be classified as a noise. One can test this effect by looking at the maximum/minimum error that we find for each depth. Figure 3-14 shows the maximum and minimum

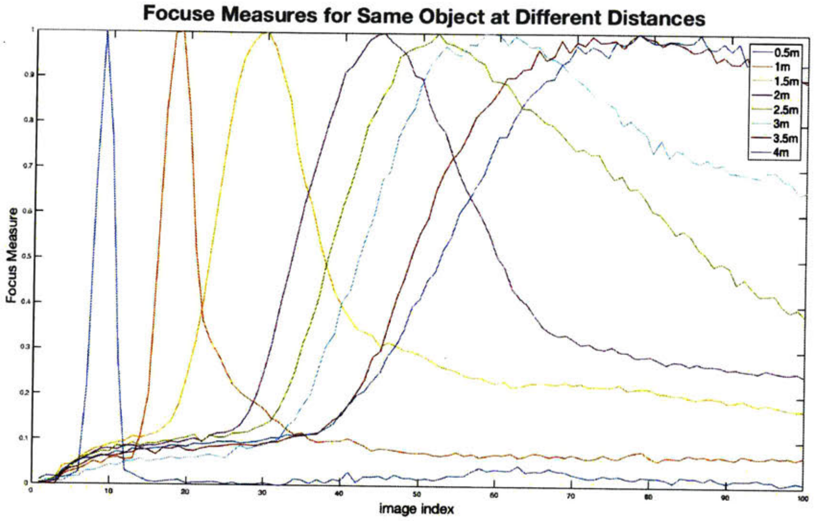


Figure 3-13: Change in energy distribution for same objects in different distances. One could observe the locality degrades as the object is at further distance.

Heisenberg Depth Uncertainty for each depth. The figure is obtained by finding for each depth, maximum and minimum Heisenberg Depth Uncertainty.

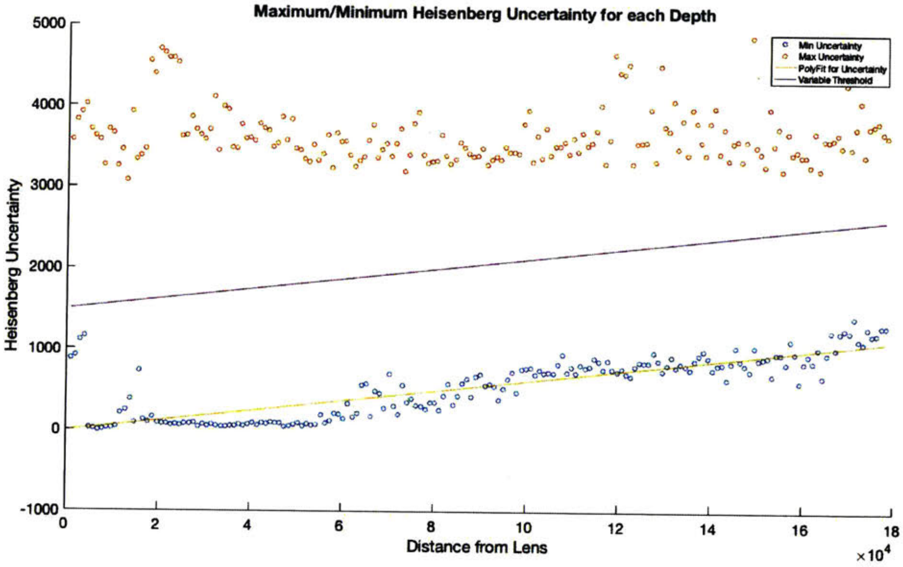


Figure 3-14: Maximum/Minimum Heisenberg Uncertainty for different distances.

From the above analysis, we can find two properties. First, the maximum error for each plot is almost uniform regardless of the depth, and is close to the Heisenberg Depth Uncertainty for maximally flat signal. For our experiment setup, of 200 image frames, maximally flat signal(i.e. $f(x) = 1$) gives Heisenberg Depth Uncertainty of 4000, which is very close to the average maximum uncertainty we obtain from the experiment. Second, we observe that the minimum value of Heisenberg Uncertainty Depth Resolution, increases as the depth increase. That is, the best signal we find at further distance shows larger Heisenberg Depth Uncertainty than the best signal at closer distance. This precisely conveys the fact we have found from the previous experiment; Energy distribution of further objects are less localized than energy distribution of the closer objects. From this fact, it is obvious that we must apply different threshold to filter the depth information with Heisenberg Depth Uncertainty. Using the minimum values of the Heisenberg Depth Uncertainty, we can come up with a linear model that explains the behavior of minimum Heisenberg Depth Uncertainty, and use that relation to find a threshold for filtering the noise by adding a constant shift to the linear model.

Below is the resulting depth map from applying variable threshold which improves our detection rate from 15% to 16.1%.

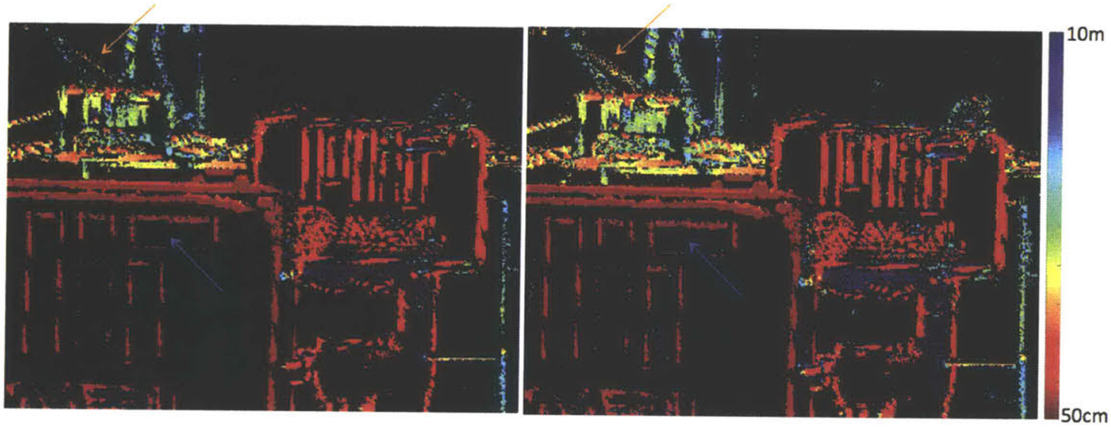


Figure 3-15: Depth Map using variable threshold for Heisenberg Depth Uncertainty. The number of pixels we recover increases, with some noise.

Looking at the resulting depth map, we could observe that some of the missing points at further distances are now recovered. However, the points that were classified as error are now classified to be at further distance because the error criteria for further points has been eased.

3.3 Modeling the Energy Distribution

In previous sections, we have seen how we can increase the detection rate by including second level decomposition as well as by applying the variable threshold for Heisenberg Depth Uncertainty. In this section, we will discuss the improvement in somewhat different direction, that is to use less number of images to find depth map with similar performance.

3.3.1 Energy Distribution of Variance of Wavelet Coefficient

Current depth mapping system uses 200 images with different focal plane that is separated by 5cms. This gives us a depth resolution of 5cm, which is better or on par with other depth mapping systems that have been studied. Our system shows high depth resolution, because the energy distribution over image index is well localized for our methodology, especially for closer objects.

The variance of wavelet coefficient shows clear and distinct peak at the location where the pixel is in focus. Therefore, our system requires well localized signal, and needs many images acquired for different focal plane. For a passive mapping system, requiring many images is not a big issue. However, to build a depth mapping system for moving vehicle, we need to reduce the number of images captured and processing time as minimal as possible. In our current system, we were able to mitigate such problem with a camera that can capture 751 fps, but such device is both expensive and sensitive to lighting condition.

Therefore, in this section, we will look at how we can use less number of images to build a depth map and still get a similar performance with the previous approach, by modeling the energy density itself, and fitting our experiment to the model.

3.3.2 Point Spread Function of Active Lens System

The process of defocusing in the digital image can be understood with the concept of the point-spread function as discussed in section 2.11. In this section, we will derive the structure of point spread function using geometrical optics. Consider the two-thin lens model that we have considered in the section 3.3.1. We have used the thin lens equation to model how the point source in the world is spreaded out on the image sensor, and showed, if the spread of the point ousrce is within the size of the pixel, the resulting image would still be in focus, and therefore result in the effect of depth of field.

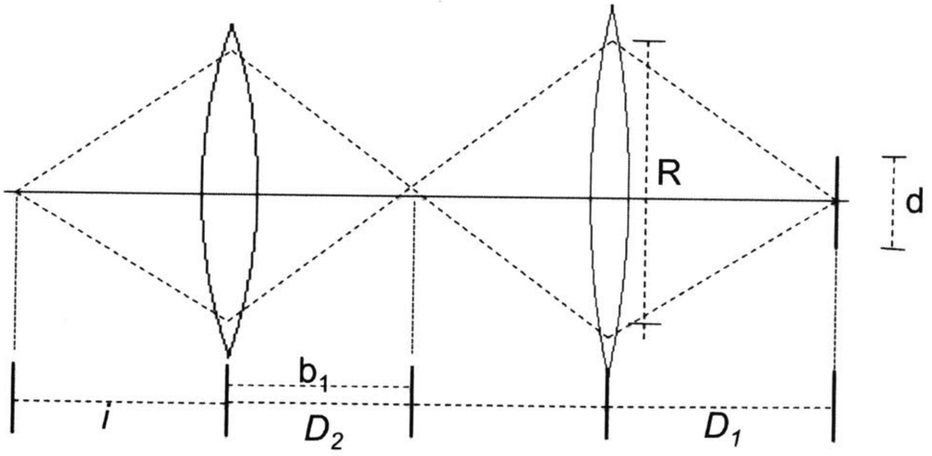


Figure 3-16: Modelling the active lens system. The distance between camera sensor and ordinary lens is D_1 and distance between two lenses are D_2.

Extending the anaylsis from Section 3.1.1, we can derive how the spread of light intensitiy will change, by varying the focal length of active lens. Following analysis is comparable with the analysis from Section 3.1.1. Consider the same setup of two thin lens system, where the camera parameters are set exactly the same as before. Then let's assume that the object is at distance o from the active lens, and active lens is calibrated such that it's focal plane is located at distance i. From the calibration of the active lens, we get the following equation.

$$\begin{aligned} \frac{1}{i} + \frac{1}{b_1} = \frac{1}{f_2} \\ \frac{1}{D_2 - b_1} + \frac{1}{D_1} = \frac{1}{f_1} \end{aligned} \tag{3.14}$$

Above equation tells us that if active lens is calibrated such that the focal plane is located at i, the image of point source at the distance i from the optics system, will be a point on the image sensor as well. The variable b is a dummy variable in the intermediate step where the image from first lens(i.e. the active lens) is formed.
We can write similar relation between the location of actual object, and the location where the that object is formed.

$$\begin{aligned} \frac{1}{o} + \frac{1}{b_2} = \frac{1}{f_2} \\ \frac{1}{D_2 - b_2} + \frac{1}{a} = \frac{1}{f_1} \end{aligned} \tag{3.15}$$

The difference of the equation 3.14 and equation 3.15, is that because in general the distance to focal plane i and the distance to object o is different, image of the object will not be formed on the image sensor (i.e. D_1) but will be at some location a. Therefore, the image that will formed on the image sensor, will not be a point, but will be spreaded to some radius r.
We can derive the radius of the spread r using simple geometry.

$$r = R|\frac{D_1 - a}{a}| = R|(\frac{D_1}{a} - 1)| \tag{3.16}$$

Now, lets write the radius of the spread r of point source in terms of camera parameters (f_1, D_1, D_2) and object distance o and image index i. We start by eliminating

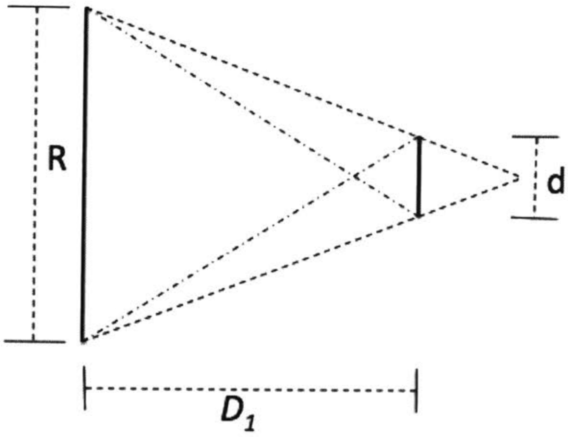


Figure 3-17: Triangle similarity relation.

the dummy variable b_1, b_2.

$$\frac{1}{b_1} = \frac{1}{f_2} - \frac{1}{i} = \frac{i - f_2}{i f_2} \rightarrow b_1 = \frac{i f_2}{i - f_2}$$
$$\frac{1}{D_2 - b_1} = \frac{1}{f_2} - \frac{1}{D_1} = \frac{D_1 - f_2}{D_1 f_2} \rightarrow D_2 - b_2 = \frac{D_1 f_2}{D_1 - f_2} \tag{3.17}$$

Adding two equations, we get following relationship between f_2, i.

$$D_2 = \frac{i f_2}{i - f_2} + \frac{D_1 f_1}{D_1 - f_1} \rightarrow \frac{1}{f_2} = \frac{1}{i} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}} \tag{3.18}$$

This relation shows the required focal power of the active lens to have a focal plane located at distance i from the active lens.

We can write similar relation for the object distance o and its image location a by eliminating dummy variable b_2.

$$\frac{1}{b_2} = \frac{1}{f_2} - \frac{1}{o} = \frac{o - f_2}{o f_2} \rightarrow b_2 = \frac{o f_2}{o - f_2}$$
$$\frac{1}{D_2 - b_2} = \frac{1}{f_2} - \frac{1}{a} = \frac{a - f_2}{a f_2} \rightarrow D_2 - b_2 = \frac{a f_2}{a - f_2} \tag{3.19}$$

Again, adding above two equations,

$$D_2 = \frac{of_2}{o - f_2} + \frac{af_1}{a - f_1} \rightarrow \frac{1}{a} = \frac{1}{i} - \frac{1}{D_2 - \frac{of_2}{o-f_2}} \tag{3.20}$$

This relation shows the location of the image of an object at distance o from the lens, when the active lens is calibrated to have a focal power of f_2.

Now to find the radius r, let's write the image location a in terms of object distance o and image index i by eliminating the focal power of active lens f_2.

$$\begin{aligned}
\frac{1}{a} &= \frac{1}{f_1} - \frac{1}{D_2 - \frac{1}{\frac{1}{f_2} - \frac{1}{o}}} \\
&= \frac{1}{f_1} - \frac{1}{D_2 - \frac{1}{\frac{1}{i} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}} - \frac{1}{o}}} \\
&= \frac{1}{f_1} - \frac{1}{D_2 - \frac{1}{\frac{1}{i} - \frac{1}{o} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}}}}
\end{aligned} \tag{3.21}$$

Radius of the spread can be written as following.

$$\begin{aligned}
r &= R|(\frac{D_1}{a} - 1)| \\
&= R|(\frac{D_1}{f_1} - 1 - \frac{1}{D_2 - \frac{1}{\frac{1}{i} - \frac{1}{o} + \frac{1}{D_2 - \frac{D_1 f_1}{D_1 - f_1}}}})|
\end{aligned} \tag{3.22}$$

To write a simler expression, we will introduce three parameters A, B and Δ that has following relationships.

$$\begin{aligned}
A &= \frac{D_1}{f_1} - 1 \\
B &= \frac{D_2}{f_1} - 1 \\
\Delta &= \frac{1}{i} - \frac{1}{o}
\end{aligned} \tag{3.23}$$

Parameter A and B entails the relations between camera's calibration and Δ encodes the relation between focal plane i and object distance o.

Now the radius r can be written as follows.

$$r = R|(A - \frac{A+1}{B+1-\frac{1}{f_1\Delta+\frac{A}{AB-1}}})| \tag{3.24}$$

Now as a final step, to get the spread over the image sensor(i.e. the radius of point spread function), we can just divide r by the pixel size d. Therefore, the radius of point spread function can be written,

$$\sigma = \frac{R}{d}|(A - \frac{A+1}{B+1-\frac{1}{f_1\Delta+\frac{A}{AB-1}}})| \tag{3.25}$$

As a check on the validity of the model, we can check that setting $i = o$, i.e. $\Delta = 0$, we get $\sigma = 0$, therefore, all-focused image. To model the point spread function, we will use a Gaussian Filter with variance σ as a model of defocusing.

3.3.3 Energy Distribution over Image Index

Point Spread Function and Energy Distribution

With the model of the point spread function, we can now consider how the energy distribution would change over the image index. One simple test that we can perform is to have a test image, and apply Gaussian Kernel with variance σ that we have obtained and measure its energy. Following is the result of such testing.

From this test, one can observe some key factors that we have seen from the actual experiment with active lens. First, we can observe that as in real world experiment, the energy distribution shows asymetric distribution. From the experiment, we have seen that the energy distribution shows sharper transition before it reaches maximum, and decays more slowly after it reaches the maximum point. We can observe similar characteristic in our model as well. Second, we can obviously see that the depth of field, which can be measured in our toy model as the number of image indicies that is saturated at maximum, increases as the target object distance becomes further. For the camera calibration, we can actually calculate the depth of field from our model,

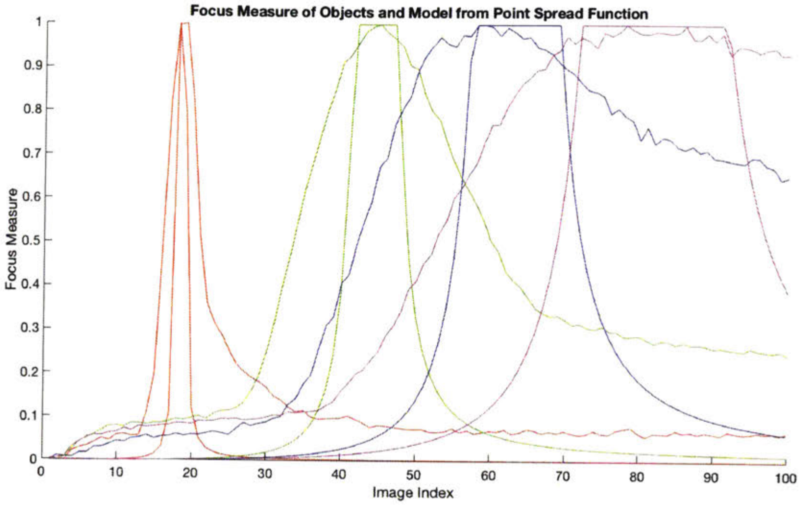


Figure 3-18: Energy Distribution obtained from point spread function model. Objects were placed at 1m, 2m, 3m, and 4m from the lens.

and get a result that matches our experiment on depth of field. From this result, we can conclude that our model explains the general structure of the energy distribution over image index.

However, if we compare the model with our actual experiment, we could see that two does not match as well as we have expected. Key difference between the experimental result and the model is that the transition from in focus image to defocus image is rather smooth for the experimental result, whereas for the model, transition is very sharp. Such behavior happens for every model energy distribution for various object distance, and therefore we can conclude that this model is not suitable to fit our measurements to find the distance to object.

Finding Point Spread Function from Experiment

We can investigate what the problem is by analyzing the difference between the radius of point-spread function from model and experiment. To compare them in parallel, let's first consider how to find the radius of point-spread function itself from the experimental data.

Consider a two dimensional discrete Gaussian Kernel, of variance σ in time domain.

$$f[n,m] = \frac{1}{\sigma\sqrt{2\pi}} e^{-(n^2+m^2)/2\sigma^2} \tag{3.26}$$

Its discrete time fourier transform pair is also a gaussian, but with variance $1/\sigma$.

$$F(\omega_n, \omega_m) = e^{-(\omega_n^2+\omega_m^2)\sigma^2/2} \tag{3.27}$$

Now, consider the integration over a fourier domain. Variance of wavelet coefficient will be the high frequency band component of this integration squared.

$$\iint d\omega_n d\omega_m F(\omega_n, \omega_m) = \iint d\omega_n d\omega_m e^{-(\omega_n^2+\omega_m^2)\sigma^2/2} \tag{3.28}$$

which is a gauss integral. From the gaussian integral, we know that above integration will be proportional to $1/\sigma$, and therefore, the energy transfer function will be proportional to $1/\sigma^2$.

Because the transfer function reduces by $1/\sigma^2$ as a function of variance of gaussian filter, we can imagine that high frequency band will also decrease by same rate. Although such approximation is dependent on the type of band pass filter and as well as the structure of underlying image, we could believe that it will generally show similar relationship with some constant factor. Therefore, we can think that the radius of point-spread function from our experimental data will have a relation as equation 3.26 upto a constant factor. For focus measure $FM_{exp}(i)$ measured in experiment, we can write the radius of point-spread function as,

$$\sigma_{exp}(i) \sim \frac{1}{\sqrt{FM_{exp}(i)}} \tag{3.29}$$

Using above relation, we can compare the radius of PSF from experiment and model, to analyze the discrepency between two energy distribution over the image index. Following is the plot of comparing the radius of PSF for model and experiment of an object at distance 2m away from the lens.

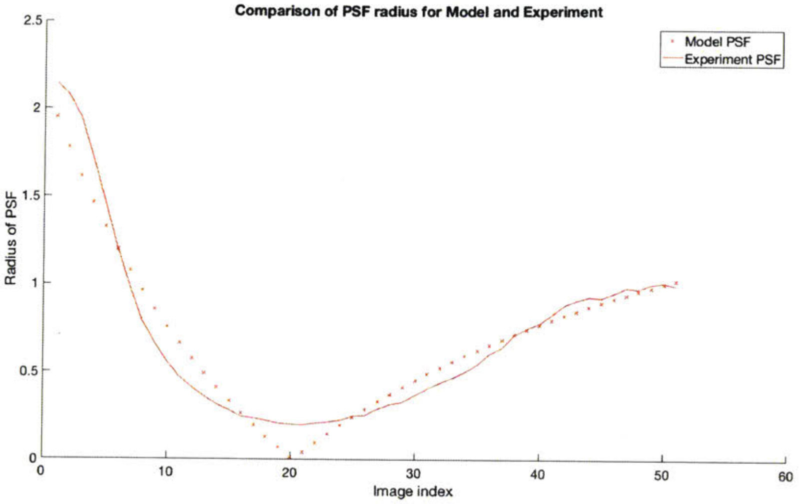


Figure 3-19: Comparison of PSF model and experiment PSF.

Figure 3-19, it becomes evident what the difference is in the model and the experiment. Both model and experiment behaves similarly at the image index that is far away from the location of object. However, as we move closer to the image index where object is located, our model decreases rapidly reaching 0, and makes a sharp turn and starts to increase, where as in the experiment, the transition around the minimum value is more smooth. Especially the fact that the model shows a sharp turn at the minimum value causes the model to have discontinuous first derivative (i.e. non-differentiable) and results in the rapid change in the energy distribution around the image index where energy becomes maximum.

Applying Kernel around Local minima

To mitigate this issue, one can introduce artificial smoothing around the local minimum of the model point-spread function radius. One of the possible choice is to use reciprocal gaussian kernel around the local minimum. Multiplying reciprocal gaussian function around the local minimum would make the function differentiable at local minima, and we can expect that would soften our transition from out of focus index number to in focus.

Another option is to use reciprocal of log normal distribution, which shows similar

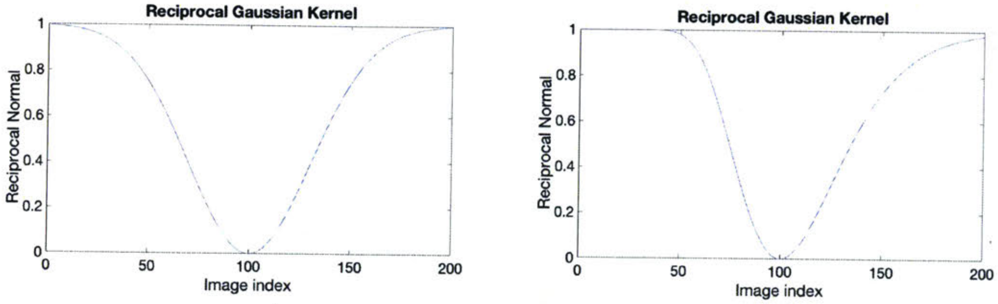


Figure 3-20: Reciprocal Normal Kernel and LogNormal Kernel.

effect of more smooth transition of energy distribution, and also can enhance asymmetric structure of the energy distribution. Result of imposing reciprocal log-normal around local minima is plotted on figure 3-21. In order to get a good fit, we need to scale and add some constant shift to the original model.

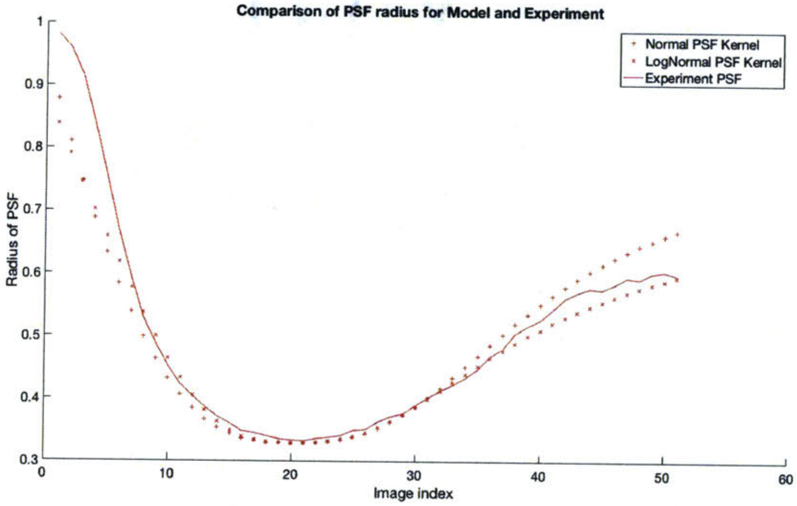


Figure 3-21: Normal Kernel and LogNormal Kernel compared with experiment PSF.

Introducing arbitrary kernel into the function may seem implausible, because it seems to be a pure mathematical trick. However, we can think more carefully about the model and can be convinced that such choice can be interpreted to have a physical interpretation. At the local minima, our point spread function will have zero radius. That means at an image index where focal plane perfectly matches the distance to an object, point source will also be a point on the image sensor, and no energy will be lost. In other word, we can think of the transfer function from light source to image sensor to be a dirac-delta function. However, in more realistic setup, we know that such perfect transfer is impossible due the diffraction and other anomallies rising from wave-like behavior of the light. That is, the geometrical optics and thin lens model that we have used to model our point spread function is highly idealized model of the real world. Therefore, we can be convinced that introducing artificial kernel can be understood as a compensation for the effect of fourier optics.

Now we could use our model to fit our data and make comparison. Using the relation of equation 3.29, we get following relationship for the energy distribution.

$$FM_{theory}(i) \sim \frac{1}{\sigma^2_{model}} \tag{3.30}$$

Comparison with the experimental data is shown in figure 3-22. Model has been optimized with the proper parameters to have minimum error.

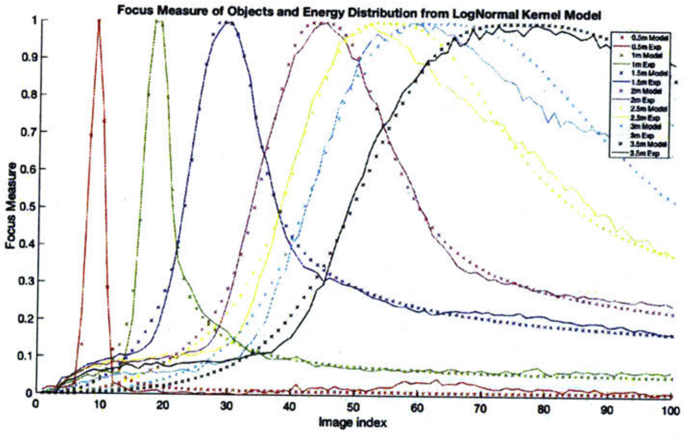


Figure 3-22: Result of optimized fitting to experiment. LogNormal kernel around the minimum smoothes the transition at maximum value.

As we can see from above figure, our model matches the experiment result very well. It maintains the both characteristics that we have seen from previous attempt, which is to have asymmetric structure, and worsening of resolution of depth for further distances. In addition, by introducing LogNormal kernel around the local minima, we now have more smooth transition, which matches the experiment result very well.

3.3.4 Depth Map from Model Fitting

Now with the model, we can use it to find the depth of individual pixels. Given the measurements of energy distribution for some number of image indicies, we can try to match it with our model. Model has three parameters, which are distance to object, variance of LogNormal kernel, and constant multiplied to match overall fit. We can use optimization algorithms to find a set of parameters that minimize the error, which can be calculated as a sum of difference of squares for each measurement and expected energy distribution. We can perform such optimization process for every pixel points, and use the optimal value of depth we found as our expectation of the distance to that pixel point. In addition, we can use the error of the fit itself as a criteria for the noise to signal comparison. High noise would indicate less likeliness of the pixel being a signal.

Below is the resulting depth estimation of the pixels from fitting the model compared with the depth map from Heisenberg depth uncertainty using same energy measurement as before; 5cm resolution upto 10m.

We can observe that the resulting depth map from model fitting is comparable to the Heisenberg depth uncertainty method. It successfully capture the distance to each pixel well, and shows similar error characteristics as in Heisenberg depth uncertainty.

Using less frames with fixed interval

The strength of modelling the energy distribution comes from the fact that we now may estimate the depth with fewer measurements. For Heisenberg depth uncertainty method, we had to have images at high depth resolution, due to the fact that the signal we find will be highly transient. Therefore, if we decrease the sampling rate, i.e. acquire images at longer distances in between, we won't be able to make accurate estimation of the depths. However, with the model of energy distribution over image

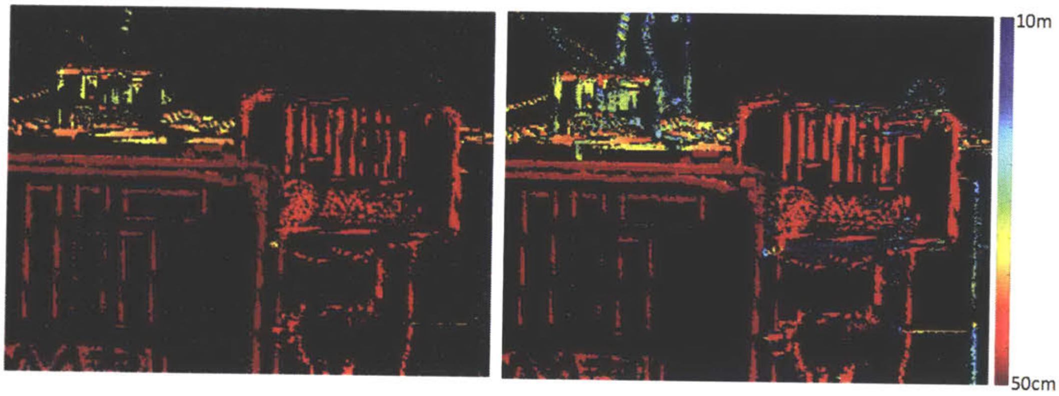


Figure 3-23: Resulting depth map from model fitting and Heisenberg Depth Uncertainty. Both maps have threshold value such they have equal number of known pixels.

index, we can now try to make an estimate of each pixel depth with fewer points. depth parameter will be determined from the overall shape of the measurement, and therefore will not depend only on maximum location of the measurement. We can test whether this argument is true by sampling our original measurement. In order to use 100 frames, which corresponds to 10cm depth resolution, we just need to use our data at every even numbered index. Similarly we can test the resulting depth map for 25, 10, 5 frames.

From the figure 3-24, we can observe few important characteristics. First, the resulting depth map is on par with Heisenberg depth uncertainty even with 25 frames of images. Now we are using 1/8th of the image number compared to Heisenberg depth uncertainty method, and getting similar depth map. Therefore, we can see that our argument that modelling will help to use less number of images is correct. Second, as we decrease the number of images, for example using 10 images, performance of the depth estimation degrades more severely for closer objects. This is an obvious result as we have discussed previously. For closer objects, the signal will be more localized compared to further objects, and as a result, will have highly transient signal. That means, if we decrease the number of the frames objects at closer distance will more likely to be not focused, and therefore our method will fail to find distance to those objects.

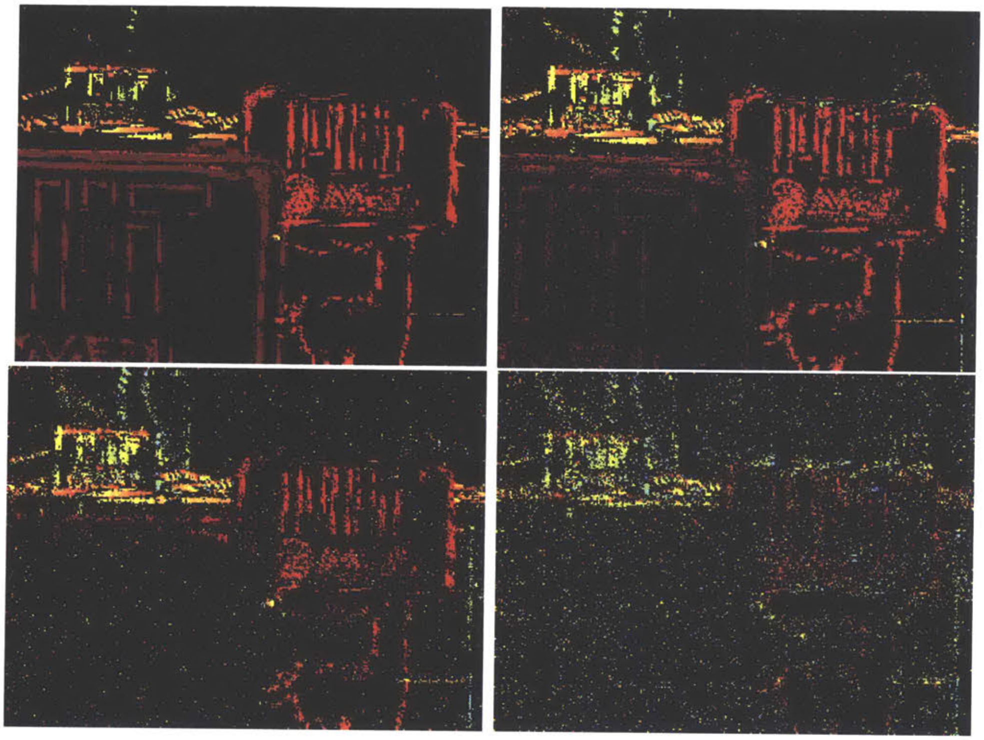


Figure 3-24: Depth Map using less number of images with fixed interval. From left to right, top to bottom, algorithm uses 200, 25, 10, and 5 frames.

Using less frames with variable interval

There is a simple remedy for such issue. That is to vary our sampling rate depending on the distance. Simply speaking, we could take more images at closer focal plane, and take fewer images at further focal plane and we can determine how one should vary the sampling rate based on the depth of field. Starting from image index 1, which has focal plane located at 50cm, we add amount of depth of field from that point. Then, we iterate this process until we reach 10m point.

Above is the result of using 40 frames for a distance from 50cm to 10m wtih variable sampling rate, where distance between each image index's focal plane is determined based on depth of field, and sub-sampled version of using 40 frames, which uses 20, 10 frames individually. As we could observe, comparing with the depth map we have obtained from using fixed sampling rate, the performance is enhanced dramatically for closer pixels and many pixels that were missing is now restored.

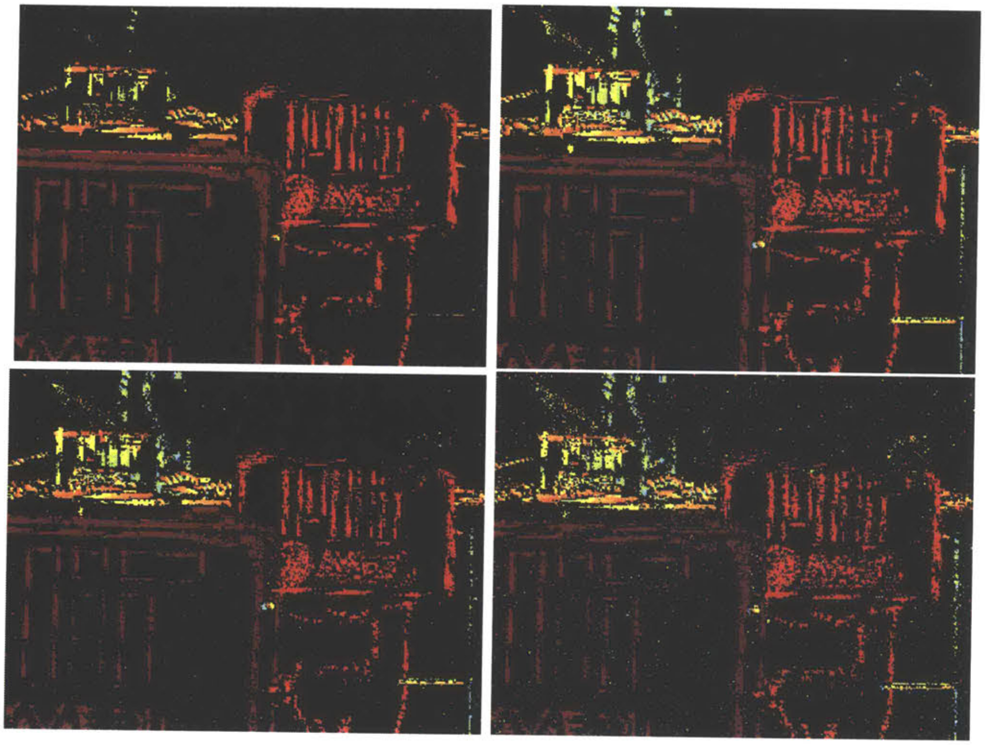


Figure 3-25: Depth Map using less number of images with variable interval. From left to right, top to bottom, algorithm uses 200, 40, 20, and 10 frames.

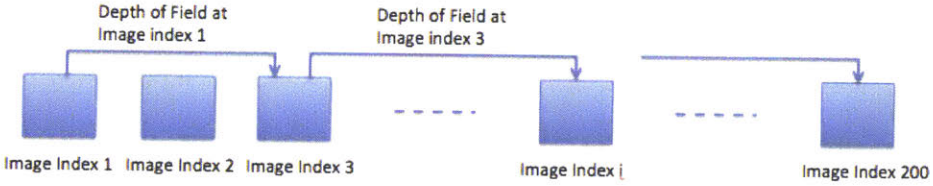


Figure 3-26: Algorithm to obtain variable sampling distance. Starting from image index 1, we add amout of depth of field and move to next image index. This process is iterated until we arrive 10m away from lens. To use fewer frames, we sub-sample this sequence. e.g. to use 20 frames, use every other image of this sequence.

We can compare the performance of two methods by looking at the distribution of the distance for each objects. The pixels that belongs to the same object will have a same distance, and by looking at the distribution of depth for those pixels, we can see the performance of our system. Below are the analysis of two methodologies using 40 frames using the histogram of the depth for the pixels that belong to objects at 1m and 2m distance.

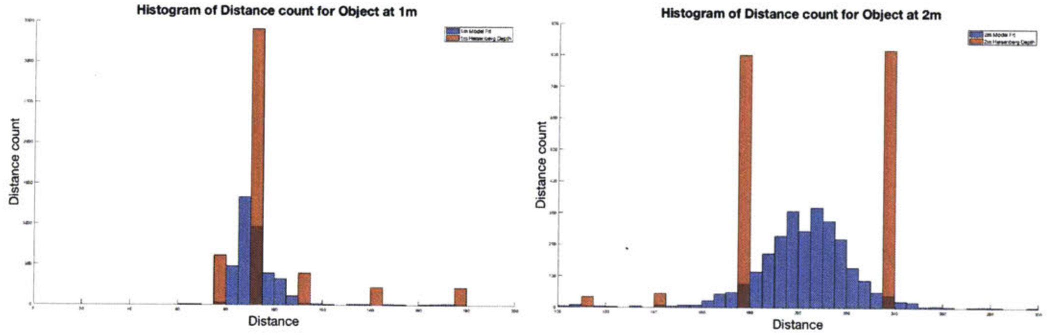


Figure 3-27: Comparison of the distribution of distance for object at 1m and 2m away from lens for model fitting with 10 frames, and Heisenberg Depth Uncertainty with 10 frames.

As we can observe, the performacė for two methods are similar for the closer objects, due to the high sampling rate at the closer distance. However, as the objects becomes further, the distance between each image index becomes further, and therefore results in an inaccurate estimation for the Heisenberg Depth Uncertainty method. For the model fitting, the estimation is more centered around a truth distance, and shows better estimation of the distance to object.

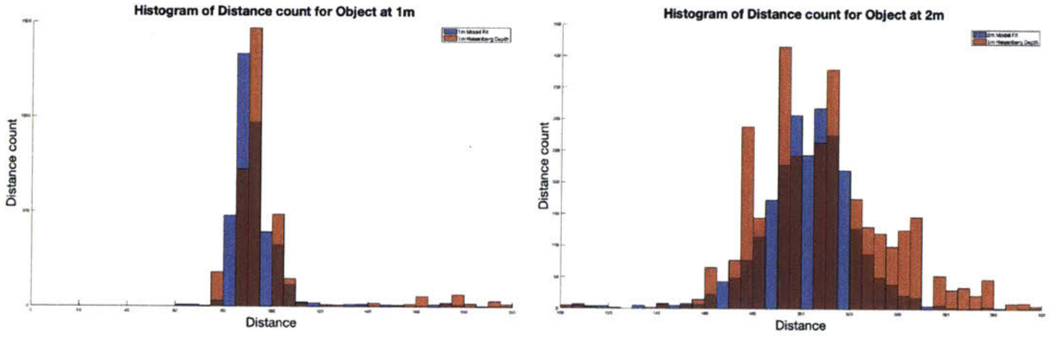


Figure 3-28: Comparison of the distribution of distance for object at 1m and 2m away from lens for model fitting with 10 frames, and Heisenberg Depth Uncertainty with 200 frames.

As a final comparison, figure 3-28 is the same analysis of histogram for model fitting using 10 frames and Heisenberg Depth Uncertainty using 200 frames of images. From the histogram, we can observe that two methods show very similar distribution near the ground truth distance. Therefore, we can conclude that the model fitting allows us to build a depth map with similar performace using much less number of images. Table 3-1, is the measurement of coverage, average distance of pixels and standard deviation of distance for object at distance 1m, 2m and 4m. One could see the performance of model fitting with LogNormal Kernel using 10 Frames, compared to Heisenberg Depth Uncertainty with 200 Frames shows on par or even better performance.

	Heisenberg Uncertainty (200 Frames)				Model with LogNormal Kernel (10 Frames)			
	Coverage	Avg Distance	StanDev	Error	Coverage	Avg Distance	StanDev	Error
Object 1m	13.65%	103.13 cm	22.84 cm	3.13%	15.66%	97.88 cm	14.06 cm	2.12%
Object 2m	28.73%	208.65 cm	43.93 cm	4.33%	23.80%	202.75 cm	34.86 cm	1.38%
Object 4m	37.12%	418.21 cm	125.57 cm	4.55%	41.41%	408.05 cm	99.07 cm	2.01%

Table 3.1: Depth Map statistics for objects in scene. Coverage, average distance, standard deviation, and error in prediction are calculated.

3.4 Performance Measurement of Depth Mapping System

3.4.1 Experimental setup for testing performance

As a final remark, we have measured the performance of Heisenberg Depth Uncertainty system and model fitting for various distances. We made a measurement for an object at various distances. For model fitting, we have used 40 variable interval images. For each experiment, we have measured coverage, average distance of pixels, standard deviation of distances and error in depth estimate against the ground truth.

Figure 3-30, shows the setup for the experiment. We have used Basler camera acA2000-165um, to get a better field of view. The target object is 30cm × 30cm checker board, and the object was placed starting from 1m to 8m at every 10cm apart at the beginning and at every 20cm.

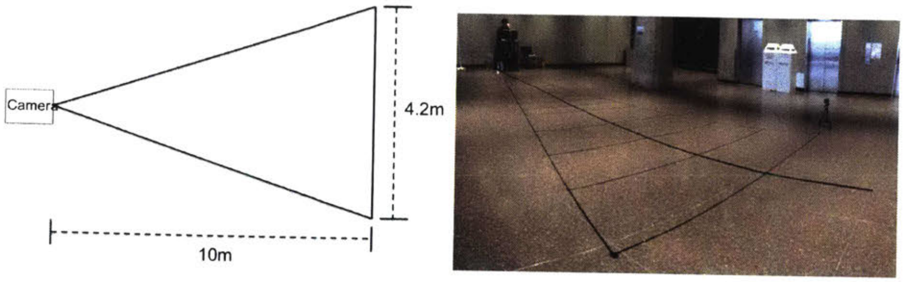


Figure 3-29: Experiment setup with field of view. Field of view of the camera at 10m distance is 4.2m wide. Black line on the floor indicates the limit on the field of view.

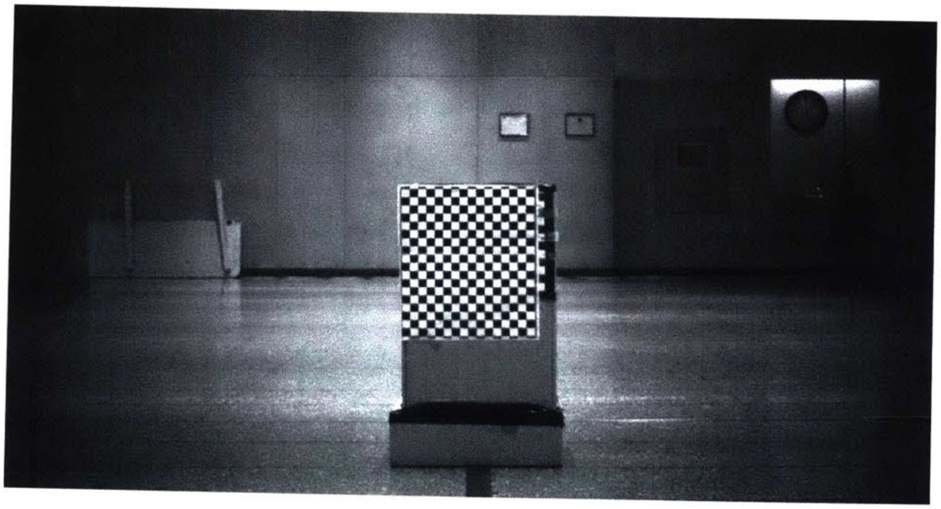


Figure 3-30: Checker board with 30 × 30 is placed at 4m distance from the lens, and camera is calibrated such that the target is at the center of the field of view.

3.4.2 Analysis of the Performance

For each set of experiment, we have obtained depth estimation using both Heisenberg depth uncertainty and model with LogNormal kernel. Then for each region of interest, we have averaged the distance and obtained standard deviation of the distribution. Detailed results are given in Appendix B, and the result of the analysis are given in the figure, 3-31, 3-32.

Figure 3-31 shows the estimated distance to the object as well as the standard deviation around the estimated average. Blue line indicates the ground truth, and red points with error-bar indicates the average distance and standard deviation. Black line indicates the expected depth of field from the camera calibration. The result

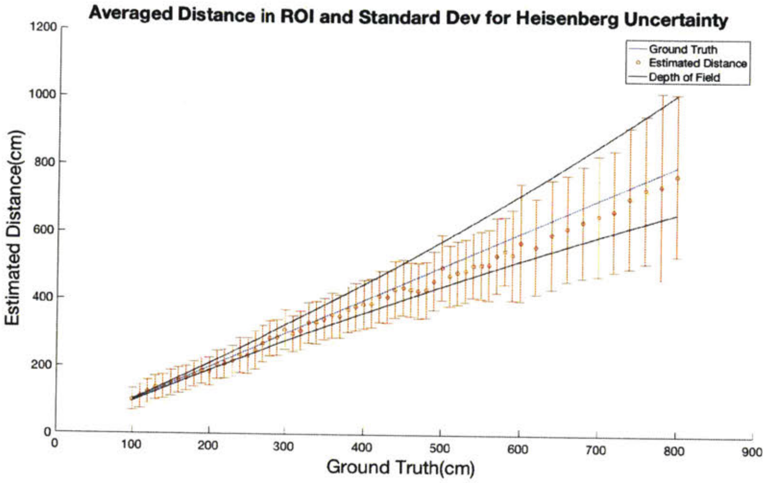


Figure 3-31: Analysis of the depth estimation using Heisenberg Depth Uncertainty. Black lines indicate the expected depth of field.

shows that the estimation falls well within the depth of field limit, and the standard deviation in the estimate becomes larger as the object is further from the lens. Error compared to the ground true distance, is below 15% for every measurements.

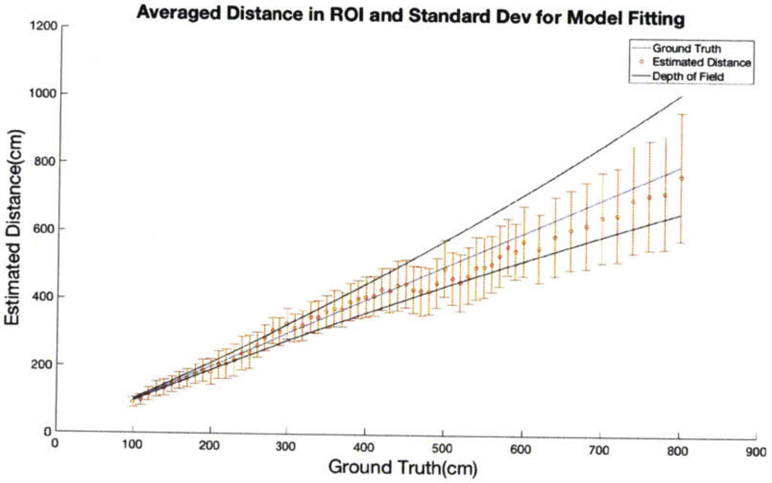


Figure 3-32: Analysis of the depth estimation using model fitting with LogNormal kernel. Black lines indicate the expected depth of field.

Similar analysis is done for model fitting with LogNormal kernel. Figure 3-32 shows the estimated distance to the object as well as the standard deviation around the estimated average for depth estimation with model fitting. Model fitting with LogNormal kernel shows very similar result the Heisenberg Uncertainty system. It shows that the estimation falls well within the depth of field limit, and the standard deviation in the estimate becomes larger, as the object is further from the lens. Again, the error compared to the ground true distance is below 15% for every measurements.

To compare the performance of two methods, we can compare the standard deviation in the measurement. Smaller standard deviation refers to more precise estimation of the distance. Figure 3-33, shows the standard deviation of two methods compared with expected depth of field from our experiment.

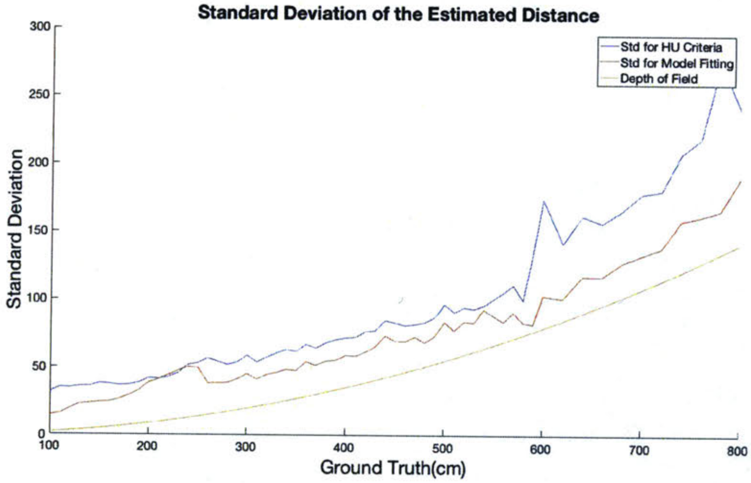


Figure 3-33: Comparing the standard deviation of two method. Model fitting shows smaller standard deviation compared to Heisenberg Depth Uncertainty.

Comparing two modesl, both model clearly shows that the standard deviation of the measurement increases as the object is further from the lens. In addition, model fitting with LogNormal kernel shows smaller standard deviation compared to Heisenberg Uncertainty system, and is closer to the depth of field limit imposed by the optical setup. Therefore, we can expect the model fitting approach to give more accurate result than the Heisenberg Uncertainty system.

As a conclusion, both method gives a depth map that is realiable upto 8m distance, with 15% accuracy at worst, and fits well with the theoretical expectation of the effect of depth of field.

Chapter 4

Performance Optimization and Conclusion

To build a depth map for active safety system for automobile, the runtime is a crucial factor that must be considered. Many of the depth from focus algorithms that has been studied focuses on the passive depth mapping systehm and therefore was free of concerns with regard to runtime. However, as our target is to build a depth mapping system for a moving vehicle, consideration of the runtime is essential.

In the final chapter, we will look at how system can be made to run in real time using basic parallelizaition, and conclude the paper with final remarks on the performace of the system.

4.1 Parallelization of Depth Mapping System

Parallelization is a design procedure that makes the algorithm to utilize multiple processors in parallel. When the given tasks are indepdent of each other, one can process these in parallel using multiple cores in the computer system. In modern computer architecture, parallel structure is an integral part of CPUs and other hardware system, and therefore careful consideration of parallelization is essential.

The depth mapping system we have built has many sub-procedures that are not dependent on each other, and therefore can be parallelized. In this section, we will look at what parallelization is and how we can minimze the runtime of our system by designing the system to be parallel compatible.

4.1.1 Parallelization and Parallelism

Since its development in 1958, the size of integrated circuits has decreased exponentially. The modern integrated circuit's width between the conducting line is near tens of nanometers of scale[27]. The famous Moore's law predicted the performance of integrated circuit doubles every two years[13], which well predicted the advance of performance of integrated circuits. However, as the width between the conducting line has decreased to nano scale, the problem of heat dissipation, as well as fundamental limits imposed by quantum mechanics, hinders further reduction of the size of integrated circuit[19].

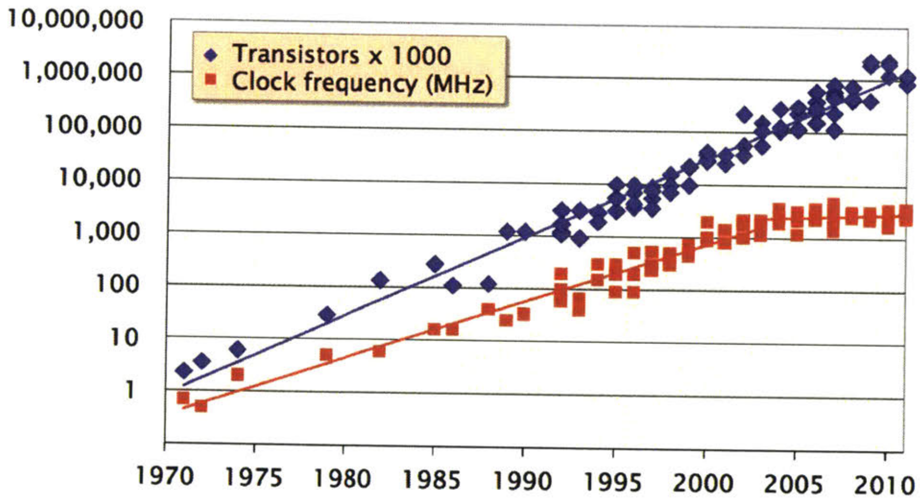


Figure 4-1: Moore's law of showing the number of transistor increase for integrated circuits. However, since early 2000, clock frequency started to asymptotically approach it's limit[2].

As an althernative to making smaller and smaller chip, set of researchers suggested using parallel system. In essence, the idea behind parallel system is to use multiple integrated circuits to process several tasks at once. Such idea has been studied for decades, and now many modern computer languages nowadays support different packages so that even students can develop their own parallel system. The idea of parallel computing has evolved into a cloud computing technology and development of super computers as well.

4.1.2 Designing Parallel System

Designing the parallel system requires a careful consideration of work distribution as well as race condition. However, for our depth mapping system, we have highly indepdent sub-procedures, and therefore can think of simple implementation of parallelized system. In this section, we will discuss how we can implement parallelized depth mapping system using Cilk extension to c++ system. Our code base uses Opencv algorithms for image processing, and the cpu is Intel(R) Xeon(R) CPU E5-2609 v3 with 12 cores of 1.90GHz each.

Let's look at the procedure of our depth mapping system in series. We sub-group our entire procedure to three smaller procedures as in figure 4-2.

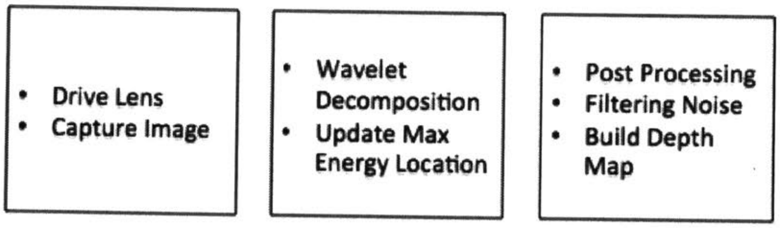


Figure 4-2: Sub-procedures that consist the depth mapping system.

First sub-procedure is the image capturing process. To acquire different images with different focal plane, we must send out a signal to the active lens, wait so that the actuation of active lens is saturated, and then capture the image. These images has to be stored somewhere in the memory so that they can be processed.

Second sub-procedure is to process the captured images. These captured images are passed to variance of wavelet operators, and the degree of focus for individual pixels are measured for each image. This information will be used to build a depth map, as well as for post processing for the noise.

Third and final step is the building depth map and post processing of the noise. With

the wavelet decomposed images from previous step, now we must find the maximum location of the energy distribution to find the distance to pixel, and process the possible noisy points to create realiable depth map.
We can think of the most simple procedure that builds depth map as follows. For every captured image, we will process it with wavelet decomposition, and update the maximum information on the run. After all images are processed, we will process the noise with Heisenberg depth uncertainty crieteria, to build the final depth map. Below is the runtime analysis of such system.

- Activate and Initialize lens driver
- Initialize camera
- Image grab thread (200 images)

i. Send proper current signal to lens driver (from 50cm to 10m)	5 ms
ii. Grab image, and save it as opencv Mat format	10ms
iii. Compute Wavelet decomposition	16ms
Total Runtime	6 sec

Table 4.1: Runtime analysis of the serial procedure of depth mapping system.

As we can observe, the total runtime of the system is 6 sec, which cannot be run in realtime for moving vehicle. In addition, such procedure cannot be parallelized as each step (image acquisition, image processing) is dependent on each other. Lastly, the image capturing process of the camera is not able to utilize the high-speed frame rate due to the fact that camera must wait for the image processing is done.
From such analysis, we can think of following system; Split the process of image capturing, image processing, and post processing step. By spliting the capturing and processing step, we are able to acheive three optimizations. First, by not waiting for the processing of the image is to be done, we can let the camera to capture the image continuously, which allows one to utilize the high-speed frame rate of the camera. Second, by saving each images individually, each processing step is now independent of each other, and therefore can be parallelized. Lastly, we can parallelize the capturing and processing steps itself, by processing the previous set of images while capturing next set of images. Such procedure can be drawn as in figure 4-3.

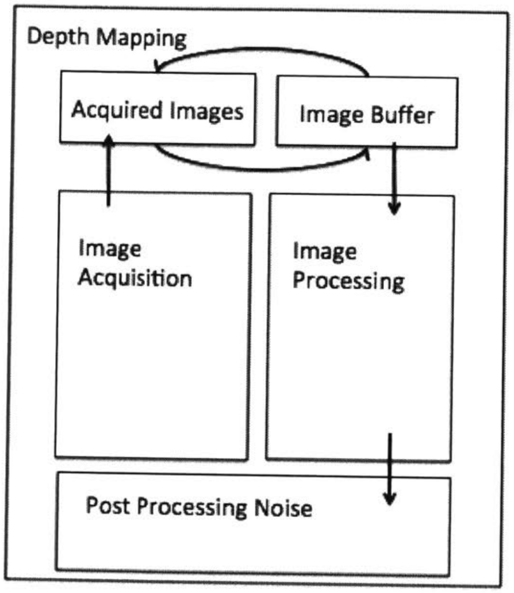


Figure 4-3: Parallel procedure of depth mapping system. Image acquisition and image processing are done in parallel. Acquired images are stored, and image processing thread processes images from previous run which is stored in image buffer.

4.1.3 Performance Measurement

By spliting the image capturing process with image processing, we now can let the camera run as fast as it can to capture images. The Basler Camera acA640-750um, has a frame rate of 751 fps, which allows us to capture 200 images in 0.26sec. However, to mitigate the problem of exposure time for the camera, we will let the camera to run at 300 fps so that we can capture 200 images in 0.67sec.

For the image processing, we can easily imagine that in theory the image processing can all be done in parallel, and therefore get $O(n)$ linear performance optimization. We can test the performance optimization using multiple cores, by using cilkview which measures the performance improvements with different number of cores. Test has been performed on Heisenberg Depth Uncertainty method with 200 images from 50cm to 10 m.

From the analysis of cilkview, we can see that the performance improvement is almost linear. Using 12 cores, we were able to achieve 9x performace improvement compared to previous system where the runtime was 6 sec, and now the image processing pro-

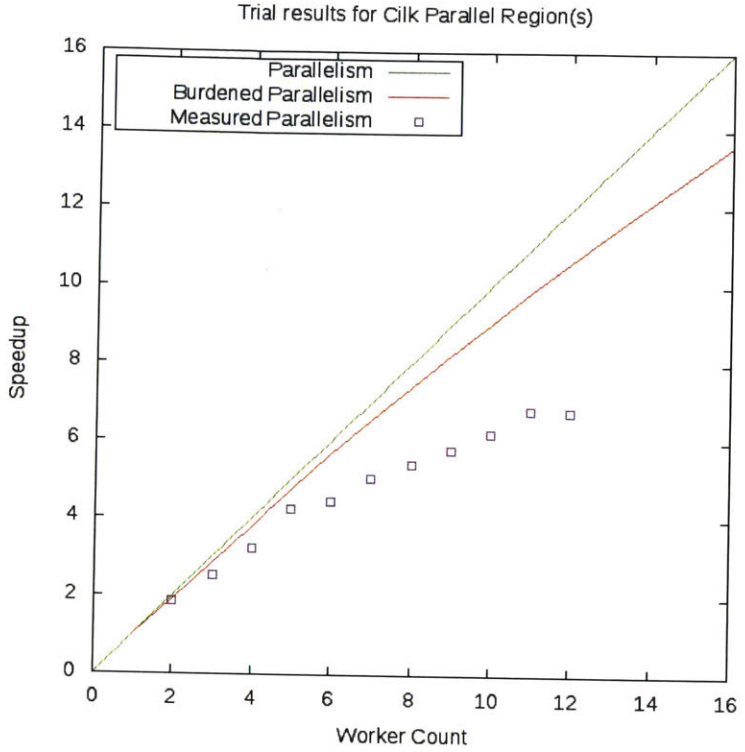


Figure 4-4: Result of analysis of the cilkview on the depth mapping system. Performance are measured from 1 cores to 12 cores.

cedure can be ran in 0.61 secs on average.

Now combining two procedures in parallel, we can run the whole procedure in 0.70 sec on average, which can be used for real time applications for depth mapping.

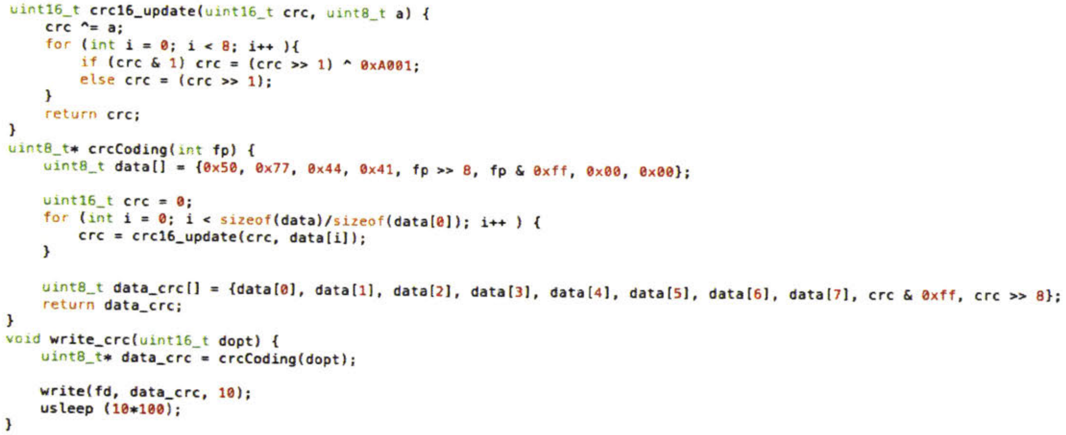
4.2 Conclusion

This thesis was first attempt to utilize the active lens and depth from focus to build a depth map for longer range than indoor environment. In this thesis, we have considered various approaches to building a depth map of the scenary using active lens. Starting from basic focus measure algorithms, we have considered different approaches to filter the noise in the depth information, and how to use less number of images using model fitting. Lastly, we have seen how parallelization of our system can speed up our system such that it can also be used in real time applications such as active safety system.

Through our analysis, we showed that the depth map we have built using focus measure algorithms combined with other criteria is comparable with other vision systems, and can easily be expanded to have better accuary and speed, and therefore can be used for various high speed applications involving dense depth mapping of the environment.

Appendix A

C++ Code for Driving Lens



```
uint16_t crc16_update(uint16_t crc, uint8_t a) {
    crc ^= a;
    for (int i = 0; i < 8; i++ ){
        if (crc & 1) crc = (crc >> 1) ^ 0xA001;
        else crc = (crc >> 1);
    }
    return crc;
}
uint8_t* crcCoding(int fp) {
    uint8_t data[] = {0x50, 0x77, 0x44, 0x41, fp >> 8, fp & 0xff, 0x00, 0x00};

    uint16_t crc = 0;
    for (int i = 0; i < sizeof(data)/sizeof(data[0]); i++ ) {
        crc = crc16_update(crc, data[i]);
    }

    uint8_t data_crc[] = {data[0], data[1], data[2], data[3], data[4], data[5], data[6], data[7], crc & 0xff, crc >> 8};
    return data_crc;
}
void write_crc(uint16_t dopt) {
    uint8_t* data_crc = crcCoding(dopt);

    write(fd, data_crc, 10);
    usleep (10*100);
}
```

Figure A-1: C++ code for driving the active lens. Crc code must be added at the end of the signal to prevent the corruption in the communication via USB connection.

Appendix B

Tables for Performance Measurement

B.1 Heisenberg Depth Uncertainty

B.2 Model Fitting with LogNormal Kernel

Object	Coverage	Avg Dist	Std Dist	Error
100 cm	3.40 %	102.03 cm	31.81 cm	2.03 %
110 cm	7.71 %	110.87 cm	34.95 cm	0.79 %
120 cm	8.78 %	124.69 cm	34.69 cm	3.91 %
130 cm	12.44 %	137.38 cm	35.86 cm	5.68 %
140 cm	14.92 %	141.04 cm	35.84 cm	0.74 %
150 cm	18.34 %	150.75 cm	38.22 cm	0.50 %
160 cm	18.86 %	158.92 cm	37.64 cm	0.67 %
170 cm	15.39 %	165.92 cm	36.67 cm	2.40 %
180 cm	14.85 %	177.32 cm	37.06 cm	1.49 %
190 cm	17.19 %	187.36 cm	38.54 cm	1.39 %
200 cm	20.50 %	185.15 cm	41.79 cm	7.42 %
210 cm	20.54 %	205.65 cm	41.36 cm	2.07 %
220 cm	23.39 %	209.06 cm	42.69 cm	4.97 %
230 cm	25.59 %	217.32 cm	45.75 cm	5.51 %
240 cm	31.79 %	234.44 cm	51.77 cm	2.32 %
250 cm	34.35 %	233.71 cm	53.02 cm	6.52 %
260 cm	34.72 %	250.31 cm	56.54 cm	3.73 %
270 cm	40.87 %	269.16 cm	54.19 cm	0.31 %
280 cm	40.79 %	285.64 cm	51.90 cm	2.01 %
290 cm	40.32 %	287.86 cm	53.65 cm	0.74 %
300 cm	40.00 %	312.54 cm	58.76 cm	4.18 %
310 cm	39.67 %	298.90 cm	53.73 cm	3.58 %
320 cm	37.86 %	308.62 cm	57.33 cm	3.55 %
330 cm	39.05 %	331.60 cm	60.63 cm	0.49 %
340 cm	39.29 %	333.07 cm	62.97 cm	2.04 %
350 cm	39.98 %	343.66 cm	61.75 cm	1.81 %
360 cm	40.85 %	355.01 cm	66.94 cm	1.39 %
370 cm	40.65 %	352.79 cm	64.24 cm	4.65 %
380 cm	41.61 %	371.76 cm	68.08 cm	2.17 %
390 cm	39.99 %	380.69 cm	70.47 cm	2.39 %
400 cm	40.87 %	388.11 cm	71.84 cm	2.97 %
410 cm	41.05 %	391.03 cm	72.38 cm	4.63 %
420 cm	40.58 %	411.80 cm	76.51 cm	1.95 %
430 cm	40.94 %	410.95 cm	77.36 cm	4.43 %
440 cm	39.02 %	432.57 cm	84.88 cm	1.69 %
450 cm	39.09 %	438.82 cm	83.08 cm	2.48 %
460 cm	36.61 %	432.67 cm	81.04 cm	5.94 %
470 cm	38.91 %	429.04 cm	81.90 cm	8.71 %
480 cm	36.71 %	431.83 cm	83.39 cm	10.04 %
490 cm	36.98 %	458.92 cm	87.13 cm	6.34 %
500 cm	35.91 %	498.69 cm	96.58 cm	0.26 %
510 cm	35.33 %	473.82 cm	90.92 cm	7.09 %
520 cm	37.43 %	483.49 cm	94.43 cm	7.02 %
530 cm	36.81 %	489.79 cm	93.25 cm	7.59 %
540 cm	39.39 %	504.46 cm	95.93 cm	6.58 %
550 cm	39.83 %	505.86 cm	100.71 cm	8.03 %
560 cm	41.13 %	506.93 cm	105.54 cm	9.48 %
570 cm	41.60 %	535.24 cm	110.91 cm	6.10 %
580 cm	41.70 %	548.63 cm	99.73 cm	5.41 %
590 cm	38.41 %	536.38 cm	134.25 cm	9.09 %
600 cm	43.13 %	573.11 cm	174.25 cm	4.48 %
620 cm	43.52 %	561.49 cm	141.75 cm	9.44 %
640 cm	43.73 %	597.01 cm	161.94 cm	6.72 %
660 cm	45.67 %	615.57 cm	156.74 cm	6.73 %
680 cm	45.05 %	634.83 cm	165.99 cm	6.64 %
700 cm	45.98 %	653.08 cm	178.27 cm	6.70 %
720 cm	45.89 %	668.17 cm	180.59 cm	7.20 %
740 cm	44.07 %	707.18 cm	208.09 cm	4.44 %
760 cm	42.02 %	733.16 cm	219.21 cm	3.53 %
780 cm	36.31 %	743.69 cm	275.69 cm	4.66 %
800 cm	37.55 %	775.35 cm	240.59 cm	3.08 %

Table B.1: Experiment result of estimated distance to an object, using Heisenberg Depth Uncertainty. Distance has been estimated by averaging over the region of interest, and the standard deviation has been computed. Error has been calculated against the ground true distance.

Object	Coverage	Avg Dist	Std Dist	Error
100 cm	5.87 %	91.98 cm	14.50 cm	8.02 %
110 cm	8.70 %	99.13 cm	15.82 cm	9.88 %
120 cm	9.33 %	116.63 cm	19.30 cm	2.81 %
130 cm	12.38 %	130.96 cm	22.78 cm	0.74 %
140 cm	13.94 %	136.39 cm	23.45 cm	2.58 %
150 cm	14.82 %	145.73 cm	24.19 cm	2.84 %
160 cm	14.62 %	156.68 cm	24.58 cm	2.08 %
170 cm	12.51 %	164.28 cm	26.36 cm	3.36 %
180 cm	12.31 %	176.57 cm	29.12 cm	1.91 %
190 cm	13.17 %	187.65 cm	32.82 cm	1.24 %
200 cm	15.26 %	184.10 cm	38.62 cm	7.95 %
210 cm	14.29 %	205.13 cm	41.00 cm	2.32 %
220 cm	15.73 %	207.81 cm	44.29 cm	5.54 %
230 cm	16.91 %	218.50 cm	47.38 cm	5.00 %
240 cm	20.04 %	239.17 cm	50.17 cm	0.35 %
250 cm	23.44 %	243.92 cm	49.55 cm	2.43 %
260 cm	24.58 %	265.08 cm	38.24 cm	1.95 %
270 cm	28.44 %	286.78 cm	38.43 cm	6.21 %
280 cm	27.52 %	307.43 cm	38.53 cm	9.80 %
290 cm	24.93 %	305.45 cm	41.18 cm	5.33 %
300 cm	23.06 %	328.07 cm	45.12 cm	9.36 %
310 cm	22.33 %	315.74 cm	41.10 cm	1.85 %
320 cm	19.36 %	323.20 cm	44.59 cm	1.00 %
330 cm	20.24 %	348.93 cm	46.12 cm	5.74 %
340 cm	19.97 %	348.87 cm	48.35 cm	2.61 %
350 cm	23.24 %	364.31 cm	47.64 cm	4.09 %
360 cm	23.00 %	374.68 cm	54.18 cm	4.08 %
370 cm	23.60 %	372.43 cm	51.56 cm	0.66 %
380 cm	22.88 %	393.93 cm	54.65 cm	3.67 %
390 cm	22.00 %	402.66 cm	55.23 cm	3.25 %
400 cm	21.35 %	409.53 cm	58.86 cm	2.38 %
410 cm	20.96 %	413.13 cm	58.55 cm	0.76 %
420 cm	20.32 %	432.95 cm	61.43 cm	3.08 %
430 cm	19.07 %	429.63 cm	65.51 cm	0.09 %
440 cm	15.30 %	442.97 cm	73.43 cm	0.67 %
450 cm	14.91 %	449.82 cm	69.72 cm	0.04 %
460 cm	12.16 %	432.56 cm	69.27 cm	5.96 %
470 cm	12.25 %	428.14 cm	72.75 cm	8.91 %
480 cm	11.12 %	427.36 cm	68.43 cm	10.97 %
490 cm	10.56 %	452.64 cm	73.24 cm	7.62 %
500 cm	7.99 %	495.59 cm	83.65 cm	0.88 %
510 cm	9.01 %	466.25 cm	77.48 cm	8.58 %
520 cm	9.23 %	453.99 cm	84.00 cm	12.69 %
530 cm	11.31 %	475.30 cm	83.35 cm	10.32 %
540 cm	12.04 %	497.97 cm	92.87 cm	7.78 %
550 cm	14.33 %	499.43 cm	88.35 cm	9.19 %
560 cm	15.81 %	510.36 cm	83.98 cm	8.86 %
570 cm	15.28 %	533.81 cm	90.81 cm	6.35 %
580 cm	16.02 %	560.71 cm	83.37 cm	3.33 %
590 cm	16.03 %	547.54 cm	81.86 cm	7.20 %
600 cm	17.40 %	577.84 cm	103.01 cm	3.69 %
620 cm	19.40 %	557.35 cm	101.08 cm	10.10 %
640 cm	17.98 %	591.46 cm	117.46 cm	7.58 %
660 cm	21.46 %	610.98 cm	117.23 cm	7.43 %
680 cm	23.00 %	624.83 cm	127.32 cm	8.11 %
700 cm	22.42 %	648.78 cm	133.02 cm	7.32 %
720 cm	22.07 %	655.96 cm	138.22 cm	8.89 %
740 cm	20.24 %	702.27 cm	158.51 cm	5.10 %
760 cm	13.01 %	715.09 cm	161.67 cm	5.91 %
780 cm	10.47 %	721.15 cm	166.08 cm	7.54 %
800 cm	14.94 %	771.50 cm	190.24 cm	3.56 %

Table B.2: Experiment result of estimated distance to an object, using model fitting with LogNormal kernel. Distance has been estimated by averaging over the region of interest, and the standard deviation has been computed. Error has been calculated against the ground true distance.

Bibliography

[1] V. Michael Bove. Entropy-based depth from focus. *JOSA A*, pages 561–566, 1993.

[2] Aparna Chandramowlishwaran Charles E Leiserson, Saman P Amarasinghe. *6.172 Performance Engineering of Software Systems.* Lecture Notes in Computer Science. Massachusetts Institute of Technology, Cambridge, MA, 2014.

[3] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. Philadelphia: Society for industrial and applied mathematics, 1992.

[4] Bradley J Nelson Ge Yang. Wavelet-based autofocusing and unsupervised segmentation of microscopic images. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 3, 2003.

[5] Amnon Shashua Gideon P. Stein, Ofer Mano. A robust method for computing vehicle ego-motion. *Proceedings of IEEE Intelligent Vehicles Symposium*, 2000.

[6] Alan T. Brooker Graham Brooker. Introduction to sensors for ranging and imaging. *SciTech Pub. Incorporated*, 2009.

[7] Kamal. Youcef-Toumi Iman Soltani Bozchalooi, Mohsen. Lakehal-Ayat. Depth mapping using active lenses. Internal presentations: Mechatronics Research Laboratory, Oct - Dec, 2014 Jan - Sep, 2015.

[8] Ruigang Yang Jiejie Zhu, Liang Wang and James Davis. Fusion of time-of-flight depth and stereo for high accuracy depth maps. *In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference. IEEE*, pages 1–8, 2008.

[9] Eric Krotkov. Focusing. *International Journal of Computer Vision 1.3*, pages 223–237, 1988.

[10] Fredo Durand Levin Anat, Rob Fergus and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics (TOG)*, 26(3), 2007.

[11] Rachid Deriche Luc Robert. Dense depth map reconstruction: A minimization and regularization approach which preserves discontinuities. *Computer Vision-ECCV'96. Springer Berlin Heidelberg*, pages 439–451, February 1996.

[12] E. Wolf M. Born. *Principles of Optics*, pages 203–232, 459–490. Pergamon, Oxford, UK, 1970.

[13] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics, 38: 8*, 1965.

[14] Arman Nikzad Murali Subbarao, Tae Choi. Focusing techniques. *Optical Engineering 32.11*, pages 2824–2836, 1992.

[15] Bernard Espiau Nicolas Andreff, Radu Horaud. Robot hand-eye calibration using structure-from-motion. *The International Journal of Robotics Research 20.3*, pages 228–248, 2001.

[16] Optotune. *Electrical Lens Driver 4 Manual.*

[17] Optotune. *Fast Electrically Tunable Lens EL-10-30 Series.*

[18] Alex. P. Petland. A new sense for depth of field. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 4*, pages 523–531, February 1987.

[19] James. R. Powell. The quantum limit to moore's law. *Proceedings of the IEEE 96.8*, pages 1247–1248, 2008.

[20] Andrew J. Davison Richard A. Newcombe. Live dense reconstruction with a single moving camera. *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE*, 2010.

[21] Andrew Zisserman Richard Hartley. *Multiple view geometry in computer vision.* Cambridge university press, 2003.

[22] Miguel Angel Garcia Said Pertuz, Domenec Puig. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition 46.5*, pages 1415–1432, 2013.

[23] Manmohan Chandraker Shiyo Song. Robust scale estimation in real-time monocular sfm for autonomous driving. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[24] Yasuo Nakagawa Shree K. Nayar. Shape from focus: An effective approach for rough surfaces. *International Conference on Robotics and Automation*, pages 218–255, 1990.

[25] http://www.mathworks.com/help/wavelet/ref/wavedec2.html.

[26] http://oica.net/category/production-statistics/.

[27] http://www.intel.com/content/www/us/en/silicon-innovations/intel-14nm-technology.html.

[28] Steven A. Shafer Yalin Xiong. Depth from focusing and defocusing. *Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on. IEEE*, pages 68–73, 1993.