Three Dimensional Scene Recovery from Image Defocus

A Dissertation Presented

by

Gopal Surya

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

State University of New York
at
Stony Brook

December 1994

Copyright © by
Gopal Surya
1994

State University of New York
at Stony Brook

The Graduate School

Gopal Surya

We the dissertation committee for the above candidate for the Doctor of Philosophy degree, hereby recommend acceptance of this dissertation.

Muralidhara Subbarao, Advisor, Associate Professor
Department of Electrical Engineering

John Murray, Associate Professor, Chairman
Department of Electrical Engineering

Petar Djuric, Assistant Professor
Department of Electrical Engineering

Gene Gindi, Associate Professor
Department of Radiology

This dissertation is accepted by the Graduate School.

Graduate School

Abstract of the Dissertation
Three Dimensional Scene Recovery from Image Defocus

by

Gopal Surya

Doctor of Philosophy

in

Electrical Engineering

State University of New York at Stony Brook

1994

Advisor: Prof. Muralidhara Subbarao

Three dimensional scene information consists of *geometric* and *photometric* information. The distance and shape of objects in a scene constitute the geometric information, while color and image irradiance of objects constitute the photometric information. *Active* methods of scene recovery use energy beams such as laser, infrared light and sonar. *Passive* methods, on the other hand, use only the ambient illumination of the scene. This dissertation deals

with a new passive method of scene recovery using defocus information. The method is based on a new spatial domain deconvolution formula. It requires only two blurred images taken with different camera parameters. The camera parameters include position, focal length and aperture size of the optical system in the camera. The computations in the method are local and hence it can be used for obtaining the depth-map of a scene in real-time. The method has been implemented on a prototype camera system and a large number of experiments have been performed. The experiments indicate that the method is useful in practical applications such as autofocusing and robot vision. The method has also been extended to continuously focus on moving objects. A simple camera configuration has been suggested for real-time focusing on moving objects. This dissertation also deals with the problem of recovering the focused image of a scene from its blurred images. A spatial domain technique and a Fourier domain technique have been investigated for focused image recovery. Experimental results and comparisons between the different techniques are provided. The experimental results have also been compared with results obtained by computer simulation.

To the people who are deprived of vision

Contents

Abstract . . . . . **iii**

List of Figures . . . . . **xiv**

List of Tables . . . . . **xv**

Acknowledgements . . . . . **xvi**

1 Introduction **1**

1.1 Importance of Vision 1

1.2 Approaches to 3-D Shape Recovery 2

1.3 Dissertation Overview 3

2 Camera Models **8**

2.1 Image formation 8

2.2 Geometric Optics 12

2.3 Wave Optics 14

2.4 Gaussian Model 16

2.5 Image Normalization 21

3 Depth from Focus and Defocus **23**

3.1 Introduction 23

3.2 Focused Image Surface 24

3.3 Focus Measures 25

3.4 Window Size for Focus Measure Computation 30

3.5 Advantages of Depth-from-Focus (DFF) 31

3.6 Disdvantages of Depth-from-Focus (DFF) 32

3.7 Depth-from-Defocus (DFD) 32

3.8 Edge based Methods 34

3.9 Algorithms for Arbitrary Scenes 37

3.9.1 Matrix Based Regularization 41

3.9.2 Differential Methods 42

4 Depth-from-Defocus: A Spatial Domain Approach **44**

4.1 Introduction 44

4.2 S Transform 45

4.2.1 Forward S-Transform 45

4.2.2 Inverse S-Transform 48

4.3 Determining Distance 49

4.4 Smoothed Differentiation Filters 56

4.5 Implementation 58

4.5.1 SPARCS 58

4.5.2 STM1 63

4.5.3 Error Analysis 72

4.5.4 Error in Stereo 78

4.5.5 STM2 79

4.5.6 3 - Dimensional Objects 80

4.6 Conclusions 86

5 Continuous Focusing of Moving Objects 88

5.1 Introduction 88

5.2 Distance of Moving Objects (CSTM) 89

5.3 Implementation 93

5.3.1 Experiments 96

5.4 Conclusions 104

6 Focused Image Recovery 106

6.1 Introduction 106

6.2 Spatial Domain Approach 108

6.2.1 S - Transform 109

6.2.2 Experiments 111

6.3 Second Method 122

6.3.1 Camera calibration for PSF 123

6.3.2 Calibration Experiments 126

6.3.3 Experimental Results 131

6.4 Experiments with unknown σ and 3-D objects 132

6.5 Conclusion 142

7 Experiments with an Off-the-Shelf Camera 149

7.1 Introduction 149

7.2 S-Transform Method 149

7.3 Thresholds for Focus Measures 153

7.3.1 Gradient Magnitude Squared 153

7.3.2 Laplacian Squared 156

7.3.3 Brightness Variance 159

7.4 3-D Shape Recovery 160

7.4.1 Inclined Plane 161

7.4.2 Cone Object 164

7.4.3 Microscopic Objects 167

7.5 Conclusion 167

8 Conclusion 171

8.1 Summary 171

8.2 Future Research 174

A S Transform 175

A.1 Forward S Transform 175

A.2 Inverse S Transform 176

A.3 Two-Dimensional Case 177

B Lens Data and Image Database 179

B.1 Lens Data 179

B.2 Image Database 180

B.3 3-D Cone Object Database 184

Bibliography **185**

List of Figures

2-1 Image formation in a convex lens 9

2-2 A two-lens camera model 10

2-3 PSF for blur circle radius of 1 pixel 16

2-4 PSF for blur circle radius of 3 pixels 17

2-5 PSF for blur circle radius of 6 pixels 17

2-6 PSF for blur circle radius of 9 pixels 18

2-7 OTF for blur circle radius of 1 pixel 18

2-8 OTF for blur circle radius of 3 pixels 19

2-9 OTF for blur circle radius of 6 pixels 19

2-10 OTF for blur circle radius of 9 pixels 20

3-1 Image formation with a 3-D scene 26

3-2 Depth-from-Focus 27

3-3 Classification of DFD algorithms 35

4-1 Least square filters 57

4-2 (a) Block diagram of SPARCS 59

4-3 Lens step Vs best focused distance 62

4-4 Flow chart of STM 64

4-5 Focused and blurred images with object at step 10 67
4-6 Focused and blurred images with object at step 60 68
4-7 Calibration plots for STM1 69
4-8 Histograms for an object at step 10 70
4-9 Histograms for an object at step 60 70
4-10 Test images in the database 71
4-11 Mean results at 200 Lux illumination 74
4-12 Mean results at room illumination 74
4-13 Mean results at 400 Lux illumination 75
4-14 Relative error 76
4-15 Absolute error 77
4-16 Radius of blur circle with lens at step 70 77
4-17 Calibration plots for STM2 82
4-18 STM2 results at normal illumination 82
4-19 STM2 results at 400 Lux illumination 83
4-20 Results for a cone object 84
4-21 Teddy bear object 85
4-22 Results on teddy bear 85

5-1 Beam splitter implementation 91
5-2 Calibration for several lens positions 92
5-3 Flow chart of CSTM 95
5-4 Objects used in CSTM experiments 97
5-5 Some CSTM results 102

6-1 Focused character image 112
6-2 Restoration with 0 steps of blur 113
6-3 Restoration with 10 steps of blur 114
6-4 Restoration with 20 steps of blur 115
6-5 Restoration with 30 steps of blur 116
6-6 Restoration with 40 steps of blur 117
6-7 Restoration with 50 steps of blur 118
6-8 Cylindrical and Gaussian PSF models 119
6-9 Images of blurred step edges 127
6-10 LSFs obtained from step edges 129
6-11 PSF radius from step edges 130
6-12 PSFs using Inverse Abel Transform 130
6-13 (b) Simulation with geometric optics PSF 134
6-14 (b) Simulation with Gaussian PSF 136
6-15 (d) Depth estimation with restoration for step 76 141
6-16 (c),(d) Depth estimation with restoration for 3-D object . . . 144
6-17 (d) Restored using actual PSF (Inv. Abel Transform) 147

7-1 Calibration Plot 151
7-2 Experimental Results using STM 151
7-3 Objects used for threshold calculations, po115 and pa150 . . . 155
7-4 Focus Measures with Object po115 156
7-5 Focus Measures with Object pa150 157
7-6 Focus Measures with Object po115 158
7-7 Focus Measures with Object pa150 159

7-8 Focus Measures with Object po115 160
7-9 Focus Measures with Object pa150 161
7-10 (a) and (b) Images of Character Poster at lens steps 121 and 175 162
7-11 Depth Map of Inclined Plane without Thresholding 163
7-12 Depth Map of Inclined Plane with Thresholding 164
7-13 (a) and (b) Images of a cone at lens steps 121 and 175 165
7-14 Depth Map of a Cone Object 166
7-15 Reconstructed Image of the cone 166
7-16 (a) and (b) Images of a mustard seed taken 105 microns apart 168
7-17 Depth Map of a Mustard Seed 169
7-18 Reconstructed Image of the Mustard Seed 169

List of Tables

Acknowledgements

I would first like to express my sincere thanks to my advisor Prof. M. Subbarao for suggesting this topic to me and for his guidance and inspiration during the entire period of my research. I am greatly indebted to Prof. J. Murray, Prof. P. Djuric and Prof. G. Gindi for consenting to be on my committee and for their comments and suggestions.

I owe a great debt of gratitude to my fellow students Dr. M.C. Lu and Mr. T. Wei for letting me use many of their computer programs and images and for helping me in many other ways. Special thanks to Ms. M. Krause for her kindness to graduate students.

This work was supported in part by the National Science Foundation under Grant IRI-8821923 and the Olympus Optical Corporation.

Chapter 1

Introduction

1.1 Importance of Vision

Visual perception is a powerful source of information to humans and animals. It is essential for survival activities like gathering food and avoiding predators. In humans, the visual system helps to perceive the shape, color, distance and other characteristics of objects in a three-dimensional scene. The central goal of computer vision research is to provide machines with visual capabilities comparable to humans. The applications of this research are in machine and robot vision, autonomous vehicle navigation, medical imaging etc. This research could also help us understand the sensory and computational aspects of human vision.

1.2 Approaches to 3-D Shape Recovery

The shapes and distances of objects in a scene constitute the 3-D geometric information in the scene, while the color and image irradiance constitute the photometric information. There are many approaches to recover the 3-D information of a scene. Active methods use lasers, sonar or infrared rays to bounce off an object and determine its distance. Passive methods on the other hand use the natural illumination of the scene and do not use any active energy sources. Some of the passive techniques are based on focus, stereopsis, shading and motion parallax.

Stereopsis is the method of obtaining 3-D shape information using binocular vision. Two images of a scene are recorded by two different cameras (or eyes) located at different positions in space. Features such as edges are extracted from one image and matched with the corresponding features on the other image. The relative displacement of a feature from one image to another is known as *disparity*. By measuring disparity, depth information can be inferred through triangulation.

Shape recovery from shading is based on the fact that the brightness of a surface depends on its orientation with respect to the light source and the observer. The shape of an object may be inferred by estimating the surface orientation at different points in its image.

Focus as a cue for depth and shape recovery is a relatively new idea in computer vision. It is well known that objects at different distances from a

lens are focused at different distances from the lens. A 2-D image detector when placed at a particular distance from the lens, registers an image which is blurred by different amounts at different points in the image plane. The amount of blur at a region in an image depends on the shape and distance of the corresponding object in the scene. There are two approaches to recovering 3-D information from focus. In the traditional approach, the most focused (sharpest) image along every possible direction is obtained by taking many images of the scene and searching for the most focused image. In another approach, there is no search involved and the focusing position along each direction is directly estimated by measuring the amount of blur in that direction. The blur is estimated from just one or two images, which are recorded at one or two known lens positions. The latter approach is termed "Depth-from-Defocus" to distinguish it from the search method which is termed "Depth-from-Focus". This research falls into the Depth-from-Defocus category.

1.3 Dissertation Overview

This dissertation is organized as follows. Chapter 2 describes the process of image formation in an optical system. Camera models with single and multiple lenses are described. The concepts of focused and blurred images are introduced and mathematical models of the point spread functions of camera systems based on geometric and wave optics are discussed. Actual values of the point spread function and the optical transfer function based on geometric and wave optics have been plotted for different values of blur. The plots

indicate that for high values of blur, the two models are almost identical. The chapter concludes with a discussion of the image magnification and brightness normalization issues that arise when using two or more images of a scene.

In Chapter 3, we first introduce the Depth-from-Focus (DFF) technique for 3-D shape recovery. DFF is essentially a search method, requiring the acquisition and processing of a large number of images. The concept of a *Focused Image Surface* is explained followed by a survey of some of the commonly used focus measures. The issues involved in the choice of window size for focus measure computation are discussed. The advantages and disadvantages of this technique over other techniques have also been listed.

Recently some researchers have attempted to recover 3-D shape information directly by measuring the amount of blur in an image (Depth-from-Defocus or DFD). DFD methods do not require focusing the object. They directly estimate the distance of an object from a measure of the level of defocus. These methods require only a few images (about 2-3) for recovering distance of an object, as compared to DFF methods which require a large number of images (typically about 10-12 for a planar object). In addition, only a few images are sufficient to determine the distance of all objects in the scene using DFD methods, irrespective of whether the objects are focused are not. For a 3-D scene, DFF methods require many more images. In the second part of Chapter 3, we review some of the recent work on Depth-from-Defocus (DFD). A few algorithms based on the processing of blurred step edges are described first. For arbitrary objects, some of the Fourier domain methods and a matrix based regularization method are reviewed. There is also a brief

note on the differential DFD methods for 3-D shape recovery.

In Chapter 4, we provide the theoretical basis and implementation details of a new method for determining distance based on image defocus. The method is based on a new Spatial Domain Convolution/ Deconvolution Transform, which is introduced in [113] recently. This method is called the *S-Transform Method* or STM. STM requires only two images taken with different camera parameters such as lens position, focal length and aperture diameter. Both images can be arbitrarily blurred and neither of them needs to be a focused image. STM therefore, is very fast in comparison with the Depth-from-Focus (DFF) methods, which search for the lens position or focal length of best focus. Also, STM involves only simple local operations and can easily be implemented in parallel to obtain the depth-map of a scene. It has been implemented on an actual camera system. Experiments on the performance of STM and their results on real-world objects are described in this chapter. The results indicate that the accuracy of STM compares well with DFF methods and is useful in practical applications. The utility of the method is demonstrated for rapid autofocusing of electronic cameras.

Chapter 5 is an extension of the S-Transform Method for continuously focusing on moving objects. Such a situation may arise in an autofocusing video camera and in robotic vision, where the objects in the scene may be slowly moving. We describe a spatial domain method based on STM. The method is named *Continuous STM* or CSTM. Two variations of CSTM - CSTM1 and CSTM2 - are described. CSTM1 is a straightforward extension of STM described in Chapter 4. It involves calibration of the camera for a

number (about 6 in our implementation) of discrete lens positions. In CSTM2 the camera is calibrated just once corresponding to one lens position. The calibration data corresponding to other positions are obtained by transforming the data of the one lens position for which the camera is calibrated. We provide the implementation details and experimental results.

In many applications it is required to obtain the focused image of a 3-D scene. In Chapter 6 we investigate two methods for using a blur parameter σ for recovering the focused images of objects in the scene from their blurred images. The first method of focused image recovery is based on the Spatial Domain Convolution/ Deconvolution Transform (S-Transform) proposed in [113]. This method uses only the blur parameter σ which is a measure of the spread of the camera's PSF. In particular, the method does not require a knowledge of the the exact form of the camera PSF. The second method, in contrast to the first, requires complete information about the form of the camera PSF. We describe a camera calibration procedure for measuring the camera PSF for various degrees of image blur. The calibration procedure is based on recording and processing the images of blurred step edges. In the second method, the focused image is obtained through a deconvolution operation in the Fourier domain using the Wiener filter. For both methods of recovering the focused image, results of experiments on an actual camera system are presented. The results of the first method are compared with the results obtained using two commonly used PSF models– cylindrical based on geometric optics, and a 2-D Gaussian. The results of the second method are compared with simulation results.

Finally, in Chapter 7, we summarize the contribution of this research and indicate some avenues for future research.

Chapter 2

Camera Models

In this chapter we review some of the basic ideas of image formation in optical systems. We first consider a very simple model of a camera consisting of a single lens. In practice, however, camera systems consist of multiple lenses. Some of the terminology associated with such a system of lenses will be introduced next. Finally we discuss some of the commonly used models of blur based on geometric and wave optics.

2.1 Image formation

Consider a camera system consisting of a single convex lens as shown in Figure 2.1. If there is a point source p at a distance u from the lens, then a *sharp* or *focused* image p' is formed on the other side of the lens at a distance v. The distances u and v are related by the well-known lens formula,

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

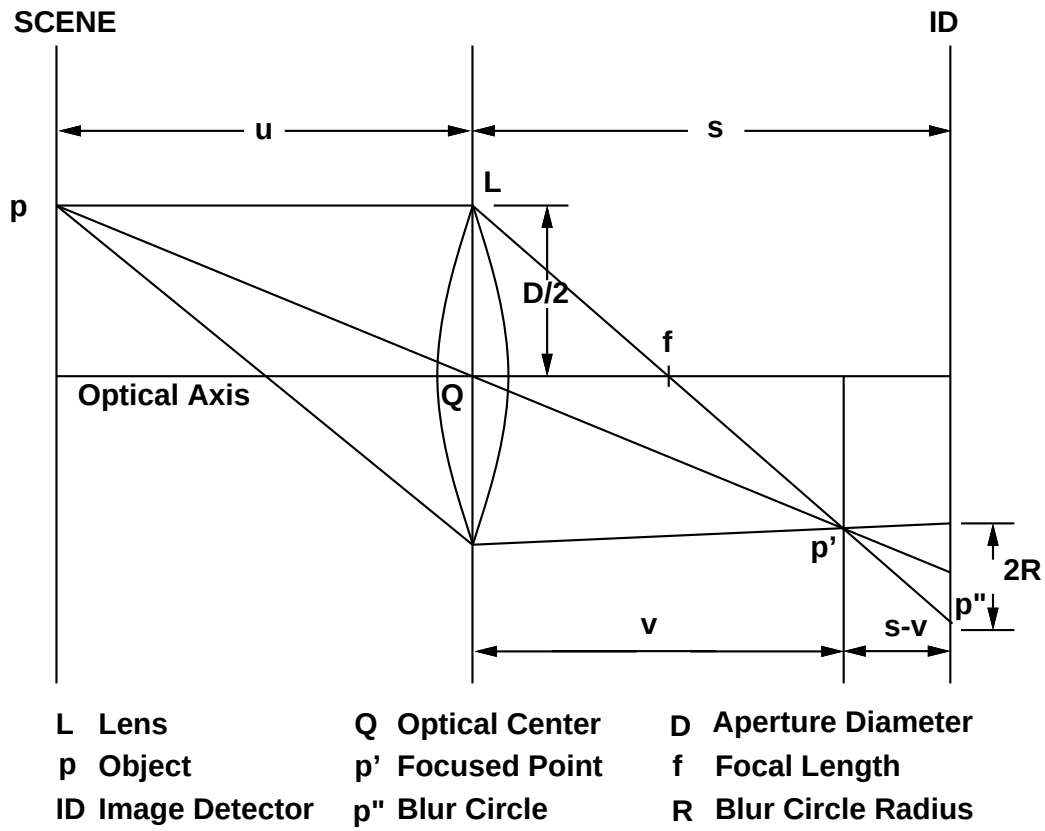


Figure 2-1: Image formation in a convex lens

where f is the focal length of the lens. If an image detector (ID) such as a photographic film or a two-dimensional CCD array is placed exactly at a distance v, a sharp image p' of the point p is formed. However, if the image detector is placed at a distance $s, s \neq v$, then a blurred image p'' of the point source is formed. The blurred image of a point source is known as the *Point Spread Function (PSF)* and will be denoted by $h(x, y)$. The Fourier Transform of the PSF is the *Optical Transfer Function (OTF)*.

In practice, camera systems consist of a number of lenses. Figure 2.2 shows a camera system with two lenses. Many of the terms associated with such systems are explained in [41]. The *aperture stop* (AS) is the element

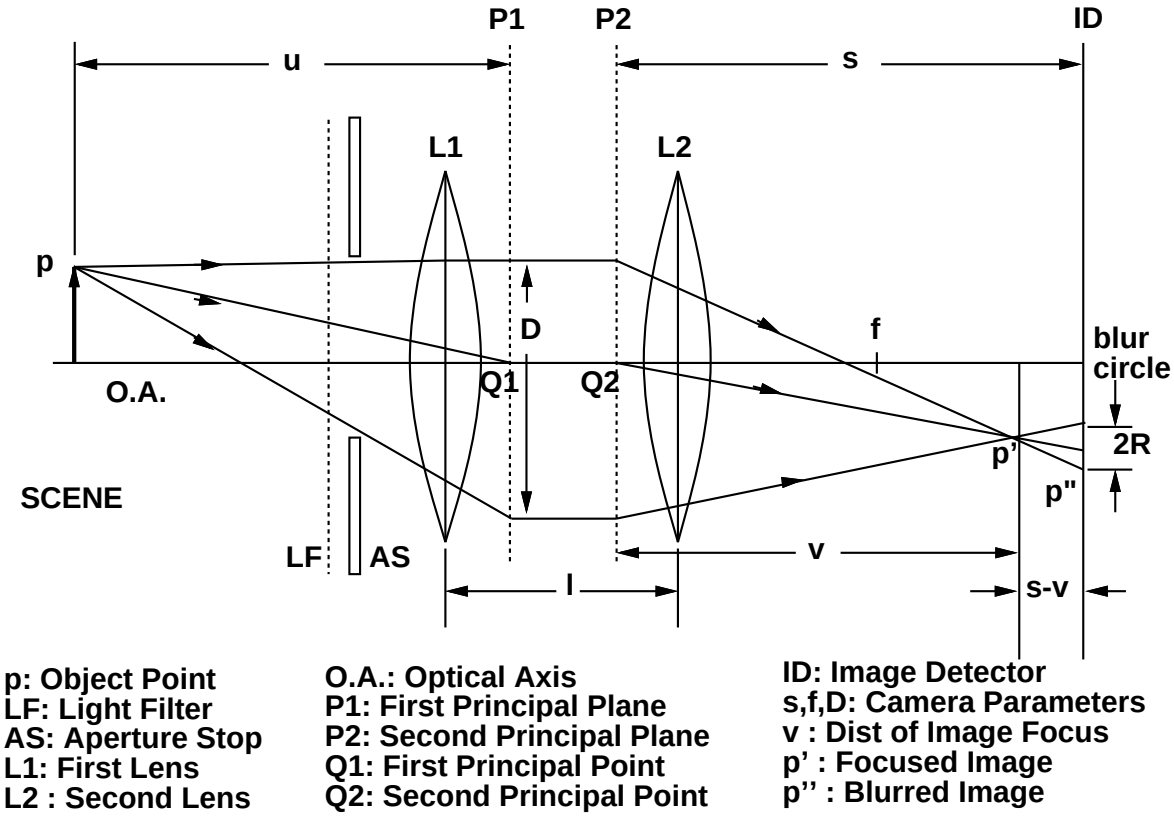


Figure 2-2: A two-lens camera model

of the imaging system which physically limits the angular size of the cone of light accepted by the system. In a simple camera, the iris diaphragm acts as an aperture stop with a variable diameter. The *field stop* is the element that physically restricts the size of the image. The *entrance pupil* is the image of the aperture stop as viewed from the object space, formed by all the optical elements preceding it. It is the effective limiting element for the angular size of the cone of light reaching the system. Similarly, the *exit pupil* is the image of the aperture stop, formed by the optical elements following it.

Whenever there is a system of lenses, consisting of two or more lenses, it

can be shown that the lens formula (2.1) is still valid in the form,

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f_{eff}} \tag{2.2}$$

where f_{eff} is the *effective focal length* of the system. However, the distances u and v will have to be measured from two unique points on the optical axis, namely, the *first principal point* (Q1) and the *second principal point* (Q2). Imaginary planes erected perpendicular to the optical axis at these points, are known as the *first principal plane* (P1) and the *second principal plane* (P2) respectively.

The distance s, focal length f and the aperture diameter D will be referred together as *camera parameters* and denoted by

$$\mathbf{e} = (s, f, D). \tag{2.3}$$

If, instead of a point source, there is a planar object perpendicular to the optical axis, with the focused intensity distribution $f(x, y)$, then the blurred image corresponding to $f(x, y)$ can be expressed as

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

where $*$ denotes the convolution operation. Equation (2.4) is only an approximation and is valid only in the region close to the optical axis (paraxial approximation). In the Fourier domain, equation (2.4) can be expressed as

$$G(\omega, \nu) = F(\omega, \nu)H(\omega, \nu) \tag{2.5}$$

where $G(\omega, \nu)$, $F(\omega, \nu)$ and $H(\omega, \nu)$ are the Fourier Transforms of $g(x, y)$, $f(x, y)$ and $h(x, y)$ respectively.

2.2 Geometric Optics

Geometric optics is based on simple ray-tracing and often yields useful results. Usually camera systems have circular apertures and hence the PSF is circularly symmetric. According to geometric optics, the blurred image of a point source (PSF), is a circular patch of constant intensity and is known as the *blur circle*. In Figure 2.1 let R be the radius of blur circle and D the diameter of the lens aperture and s be the distance from the lens to the image detector plane. Also let q be the scaling factor defined by $q = 2R/D$. From similar triangles in Figure 2.1 we have

$$q = \frac{2R}{D} = \frac{s-v}{v} = s\left[\frac{1}{v} - \frac{1}{s}\right] \tag{2.6}$$

Substituting for $\frac{1}{v}$ from equation (2.1) in the above equation we obtain,

$$q = s\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{2.7}$$

Therefore,

$$R = q\frac{D}{2} = s\frac{D}{2}\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{2.8}$$

Note that q and therefore R can be either positive or negative depending on whether $s \geq v$ or $s < v$. In the former case the image detector plane is behind the focused image of p and in the latter case it is in front of the focused image of p. It will be seen later in this chapter that it is required to normalize the images with respect to magnification. Without loss of generality we shall assume that magnification has been normalized corresponding to $s = s_0$. The normalized radius $R' = \frac{s_0 R}{s}$ of the blur circle can be expressed as a function

of camera parameter setting $\mathbf{e}$ and object distance u as

$$R'(e;u) = \frac{Ds_0}{2}\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{2.9}$$

According to geometric optics the intensity within the blur circle is approximately constant. If we assume the camera to be a lossless system (i.e., no light energy is absorbed by the camera system) then

$$\int\int h(x,y)\ dx\ dy \ = \ 1 \tag{2.10}$$

because the light energy incident on the lens is taken to be one unit. Using these facts, we obtain the PSF to be a cylindrical function

$$h_g(x,y) = \begin{cases} \frac{1}{\pi R'^2} & \text{if } x^2+y^2 \leq R'^2 \\ \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

where h_g is the PSF according to paraxial geometric optics. Due to circular symmetry, we can represent

$$h_g(x,y) = h(r) \quad \text{where} \quad r = \sqrt{x^2+y^2} \tag{2.12}$$

If a PSF is circularly symmetric then the corresponding OTF, $H(\omega,\nu)$ is also circularly symmetric. Accordingly $H(\omega,\nu)$ can be expressed as

$$H(\omega,\nu) = H(\rho) \quad \text{where} \quad \rho = \sqrt{\omega^2+\nu^2} \tag{2.13}$$

The OTF corresponding to the PSF $h_g(x,y)$ is given by

$$H_g(\rho) = \frac{J_1(2\pi R'\rho)}{\pi R'\rho} \tag{2.14}$$

where J_1 is the first order Bessel function.

The *spread parameter* is a quantity used to characterize a PSF and is defined as the standard deviation of the distribution of the PSF $h(x, y)$. Using polar coordinates it can be shown that the spread parameter σ_g, corresponding to $h_g(x, y)$ is $\frac{R'}{\sqrt{2}}$. Therefore from equation (2.8) we have

$$\sigma_{h_g} = mu^{-1} + c \tag{2.15}$$

where

$$m = -\frac{Ds_0}{2\sqrt{2}} \tag{2.16}$$

and

$$c = \frac{Ds_0}{2\sqrt{2}} \left[\frac{1}{f} - \frac{1}{s}\right] \tag{2.17}$$

Thus, for a given parameter setting (i.e., for a given value of camera parameters s, f and D), the spread parameter depends linearly on the inverse distance u^{-1}.

2.3 Wave Optics

Wave optics takes into account the wave nature of light, instead of the simple rectilinear propogation model of light considered by geometric optics. The PSF based on wave optics was first derived by Hopkins [51] and is also discussed in [77, 103, 76, 111].

According to Levi and Austing [77] the OTF corresponding to a focus defect Δ is given by

$$H(\rho, \Delta) = \frac{4}{\pi} \int_{\rho}^{1} \sqrt{1 - t^2}\; cos\left[2\pi\Delta\rho(t - \rho)\right] dt \tag{2.18}$$

The focus defect Δ is a measure of defocus and has been shown to be equal to the radius of blur circle expressed in reduced units,

$$\Delta = \frac{R'}{\lambda(s/D)} \tag{2.19}$$

where λ is the wavelength of light. The corresponding PSF for wave optics, $h_w(r, \Delta)$ can be found by the inverse Fourier-Bessel Transform given by

$$h_w(r, \Delta) = 2\pi \int_0^\infty H(\rho, \Delta) J_0(2\pi\rho r)\rho d\rho \tag{2.20}$$

where J_0 is the Zeroth Order Bessel function of the first kind and r is the reduced radial distance on the image detector plane.

The spread parameter of the PSF corresponding to wave optics was evaluated indirectly in [111] and was found to be of the form

$$\sigma_w^2 \approx \sigma_g^2 + \sigma_0^2 \tag{2.21}$$

where σ_w, σ_g are the spread parameters of wave and geometric optic PSFs respectively and σ_0 is a constant equal to the spread parameter of the PSF obtained with a point source whose image is focused according to geometric optics (i.e., $\sigma_g = 0$) [111].

For comparison, the OTF and PSF corresponding to wave and geometric optics for different amounts of blur have been plotted in Figures 2.3 - 2.10. Figures 2.3 - 2.6 show the PSF values obtained for a blur circle of radius 1 pixel, 3 pixels, 6 pixels and 9 pixels in our camera system. The radius of blur circle is plotted in terms of Δ. The PSF for wave optics was obtained by numerical integration using equations (2.18) and (2.20). In the plots the solid line shows the PSF using wave optics and the dotted line shows the PSF with

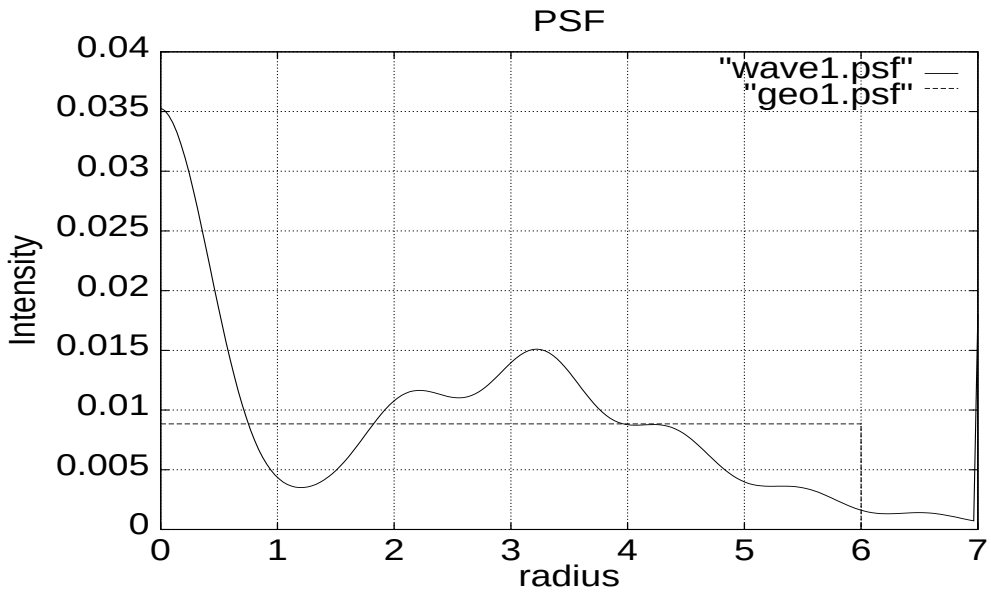


Figure 2-3: PSF for blur circle radius of 1 pixel

geometric optics. It can be seen that for higher degrees of blur, there is a closer resemblance between the two models. Figures 2.7 - 2.10 show the OTFs, again for a blur circle radius of 1, 3, 6 and 9 pixels. The OTF for wave optics was obtained using equation (2.18) and for geometric optics was obtained using equation (2.14). Once again, it can be seen that for large values of blur the two OTF models are almost identical.

2.4 Gaussian Model

Some researchers [55, 93, 101, 106] considered the polychromatic illumination, lens aberrations etc. and proposed a two-dimensional Gaussian model

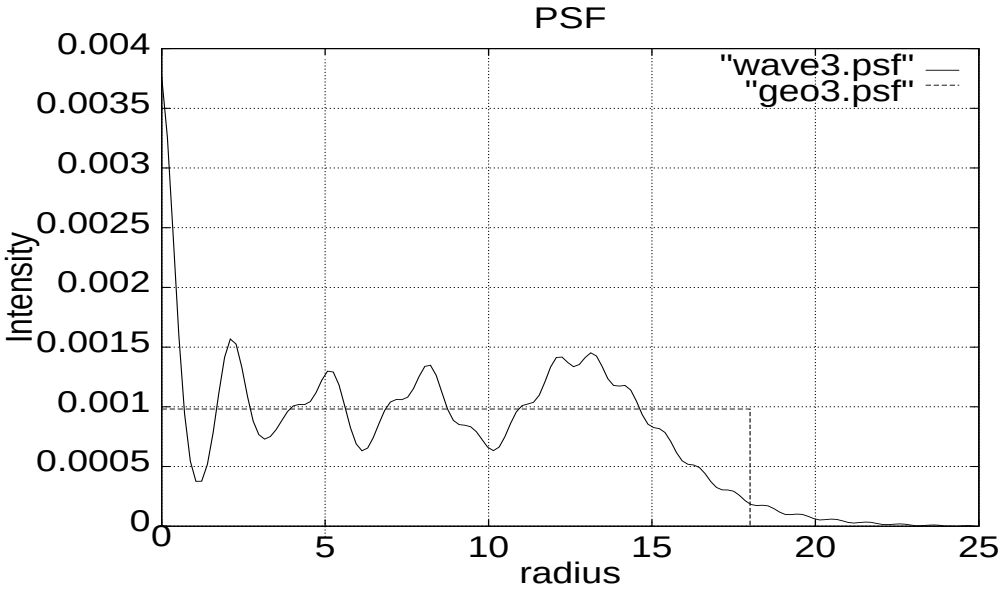


Figure 2-4: PSF for blur circle radius of 3 pixels

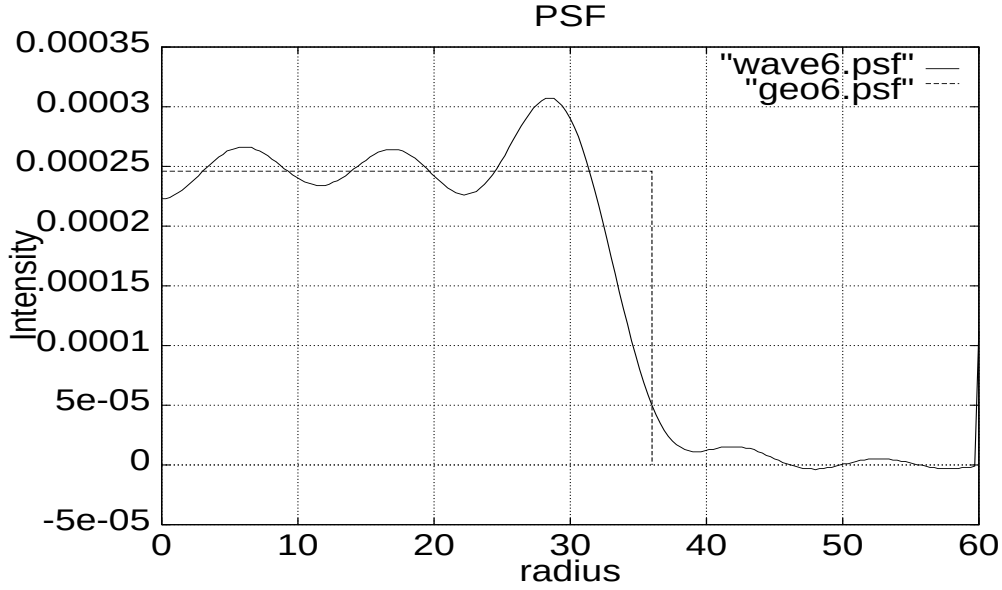


Figure 2-5: PSF for blur circle radius of 6 pixels

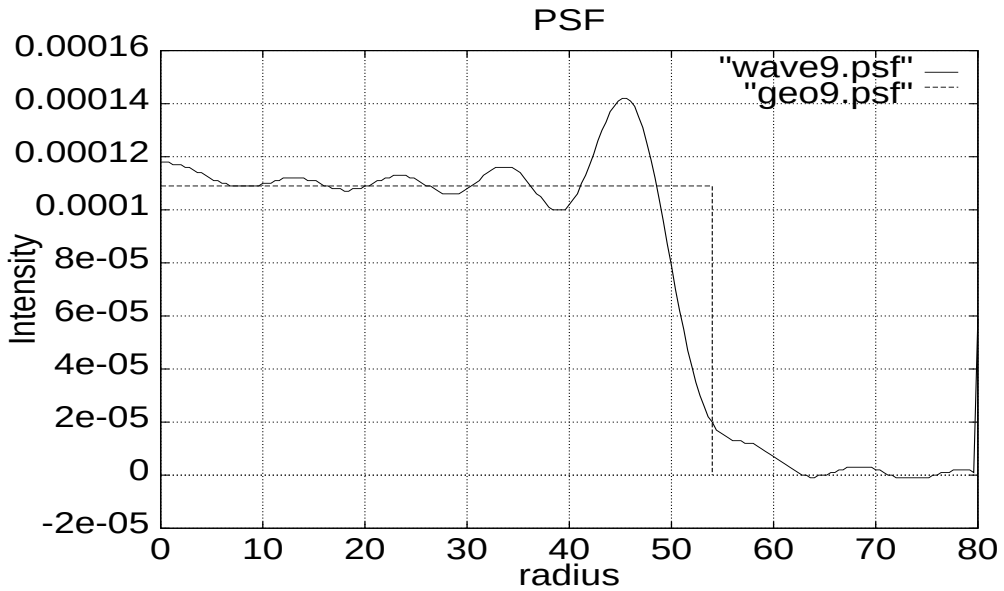


Figure 2-6: PSF for blur circle radius of 9 pixels

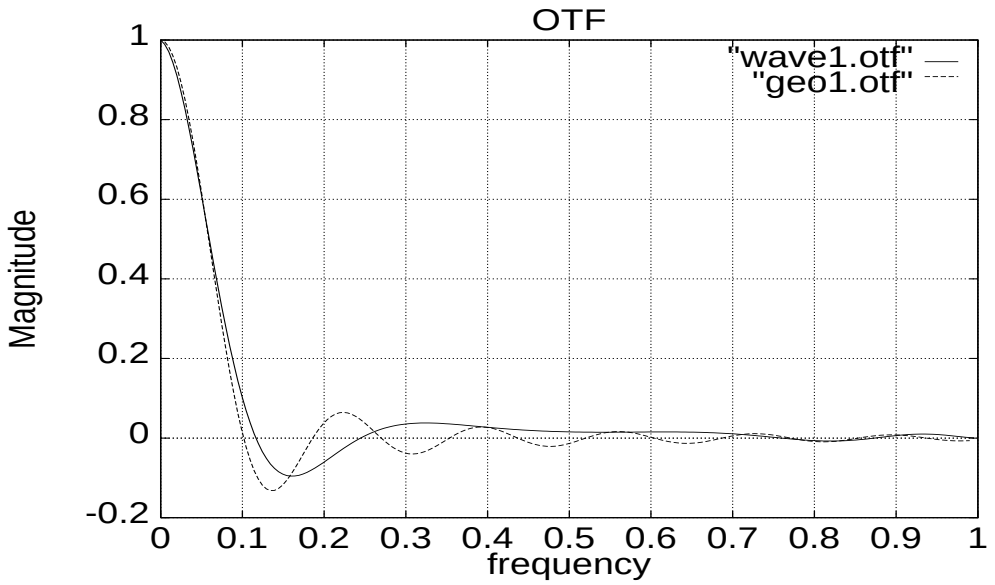


Figure 2-7: OTF for blur circle radius of 1 pixel

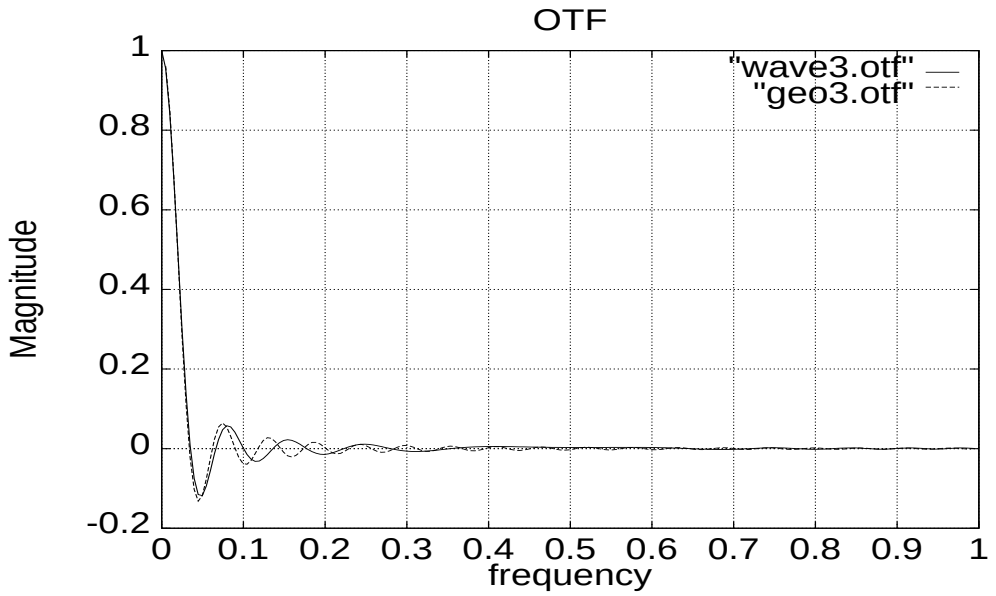


Figure 2-8: OTF for blur circle radius of 3 pixels

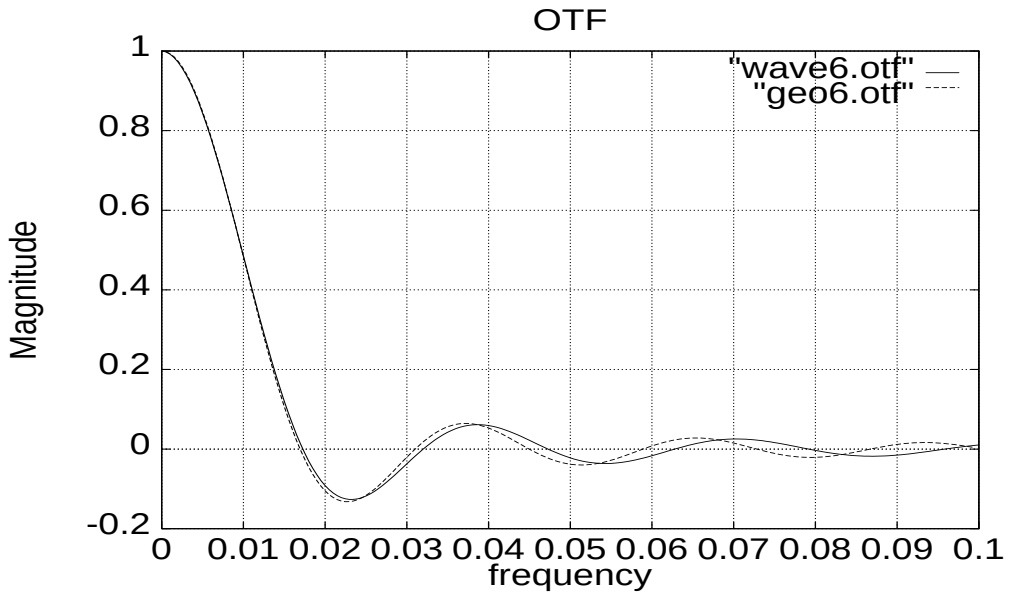


Figure 2-9: OTF for blur circle radius of 6 pixels

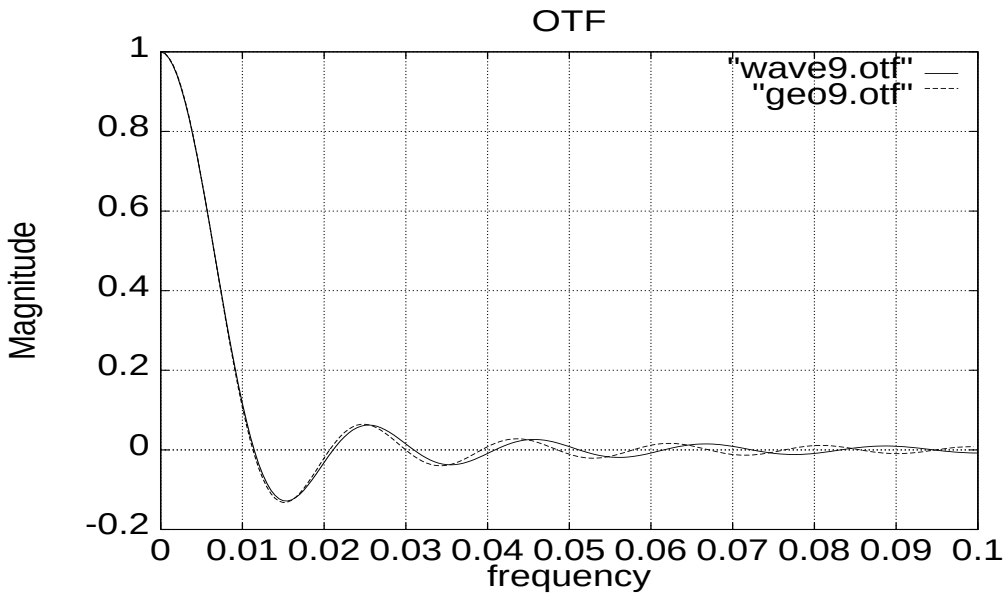


Figure 2-10: OTF for blur circle radius of 9 pixels

for the PSF. Accordingly, the PSF is defined as

$$h_n(x,y) = \frac{1}{2\pi\sigma_n^2} e^{-\frac{x^2+y^2}{2\sigma_n^2}} \tag{2.22}$$

where σ_n is the spread parameter corresponding to the Gaussian PSF. In practice, it is found that [108, 109] σ is proportional to R', i.e.

$$\sigma \;=\; k\,R' \;\text{ for }\; k > 0 \tag{2.23}$$

where k is a constant of proportionality characteristic of the given camera. Except when σ is very small (in which case diffraction effects dominate), in most practical cases

$$k = \frac{1}{\sqrt{2}} \tag{2.24}$$

is a good approximation [107, 109].

In general, equation (2.15) is approximately valid for all the PSF models considered so far and hence in general,

$$\sigma = mu^{-1} + c. \tag{2.25}$$

Equation (2.25) indicates that if we measure the spread parameter of the PSF of a blurred image, we can possibly recover the distance u of the object in terms of the camera parameters. This argument will be further pursued in Chapter 4 and Chapter 5 for determining distance from image defocus. In practice, none of the PSF models discussed above may be exactly valid and in Chapter 6, we will describe a method of experimentally determining the PSF of a camera by using step edges.

2.5 Image Normalization

In a practical camera system, if two images $g_i(x, y)$ for $i = 1, 2$ are taken at camera parameter settings of $\mathbf{e}_i$, then image magnification and mean image brightness may change even though nothing has changed in the scene. For example, moving the lens away from the image detector will increase image magnification (because magnification is proportional to s) and changing the aperture diameter changes mean image brightness (which is proportional to $\pi(D/2)^2$). In order to compare the blur in images g_1 and g_2 in a correct and consistent manner, they must be first normalized with respect to these factors. Normalization with respect to image brightness is carried out by dividing the image brightness at every point by the mean brightness of the image.

Normalization with respect to image magnification is more complicated. It can be done by image interpolation and resampling such that all images correspond to the same field of view [110]. The relation between an original image $g(x, y)$ taken with $s = s_1$ and the corresponding magnification normalized image $g_n(x, y)$ taken with $s = s_0$ is given by $g_n(x/s_1, y/s_1) = g(x/s_0, y/s_0)$. However, in most practical applications, the magnification change is less than 3% and can be ignored.

Chapter 3

Depth from Focus and Defocus

3.1 Introduction

Three dimensional scene information consists of photometric and geometric information. Photometric information consists of color and image irradiance of objects and geometric information consists of distances and shapes of objects in the scene. One of the central problems in computer vision is to recover the geometric and photometric information of a scene from 2-D images of the scene. In the next few chapters we focus on recovering the geometric information and in Chapter 6 we address the problem of recovering the photometric information.

There are many approaches to 3-D depth recovery and are surveyed in [11, 12, 60, 64, 98, 104]. Active methods such as infrared and sonar techniques use an energy source for the purpose. Passive methods on the other hand, are more attractive as they use only the ambient illumination. The most important among the passive methods are based on stereo disparity, focus and

motion parallax.

In this chapter we first outline the basis of Depth-from-Focus (DFF) methods and survey some of the focus measures that have been proposed. We also indicate the advantages and pitfalls of this approach over other techniques. Next we review some of the previous work on Depth-from-Defocus.

3.2 Focused Image Surface

Figure 2.1 shows the image formation in a simple optical system, such as a convex lens. If P is a point source at a distance u from the camera system, a sharp image p' of P is formed at a distance v from the lens. The quantities u and v are related to the focal length f of the lens by the lens formula,

$$\frac{1}{u} + \frac{1}{v} = \frac{1}{f} \tag{3.1}$$

As mentioned in Chapter 2, if the image detector is placed exactly at a distance v from the camera, a very sharp image p' will be registered. If the image detector is placed at some other distance $s, s \neq v$, a blurred image of the point source is formed. The blurred image of the point source, is known as the *Point Spread Function* (PSF) of the optical system and various models of the PSF have been discussed in Chapter 2.

The problem of focusing is to find and adjust the value of focal length f or the distance s between the lens and the image detector (Figure 2.1) or both, so that a specified object is in focus. One way of focusing then is to vary f and/or s in small steps until the observed image of the object is in sharpest focus. Once the values of f and s which correspond to focusing the object

are found, the distance of the object can be calculated using the lens formula. Therefore DFF is essentially a search method, which requires acquiring and processing many images. The search may be optimized by using a binary search or a Fibonacci search [67] technique. Typically about 10-12 images are required for a planar object, in our implementation.

In a 3-D scene, however, different points in the scene will be at different distances from the camera. Correspondingly, the sharpest image of each point in the scene is formed at a specific distance from the camera. Figure 3.1 shows a scene with a 3-D object. The sharpest image of the 3-D object is formed along a surface, known as the "Focused Image Surface" (FIS). It can be seen in Figure 3.1, that if a 2-D image detector is placed at a distance v_1 from the lens, only points on the image detector which intersect the FIS are sharpest (focused), while other points are blurred. If a sequence of images of the scene are taken, with the image detector positions varying from S_1 to S_2 as shown in Figure 3.2, we obtain a three dimensional image volume. The problem then is to recover the FIS from the image volume. In our implementation, about 25 images of the scene are required to recover the shape of a typical 3-D object such as a cone.

3.3 Focus Measures

In the previous section we mentioned that 3-D shape recovery from DFF involves searching for the most focused image among a sequence of images. In this section we discuss the criteria for searching. The search is based on

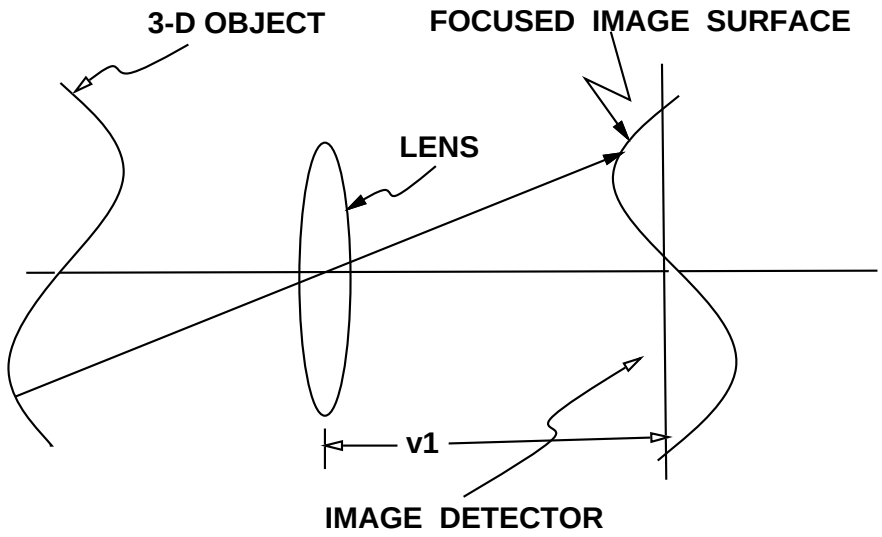


Figure 3-1: Image formation with a 3-D scene

maximizing a focus measure and the choice of a focus measure is very critical to the performance of a DFF method. Considering the nature of the blurring process as some kind of low pass filtering in the spatial frequency domain, many different measures of focus have been proposed. Krotkov [67, 68], evaluated many of the focus measures. Other surveys of focus measures may be found in [35, 119]. Here we enumerate some of the commonly used focus measures.

1. Gradient Magnitude

The thresholded gradient magnitude (tenengrad) scheme was investigated by Tenenbaum [121] and Schlag [100]. The gradient magnitude is defined as

$$|\bigtriangledown g(x,y)| = \sqrt{\left(\frac{\partial g}{\partial x}\right)^2 + \left(\frac{\partial g}{\partial y}\right)^2} \tag{3.2}$$

where $g(x,y)$ is an image. The focus measure is obtained by summing all the

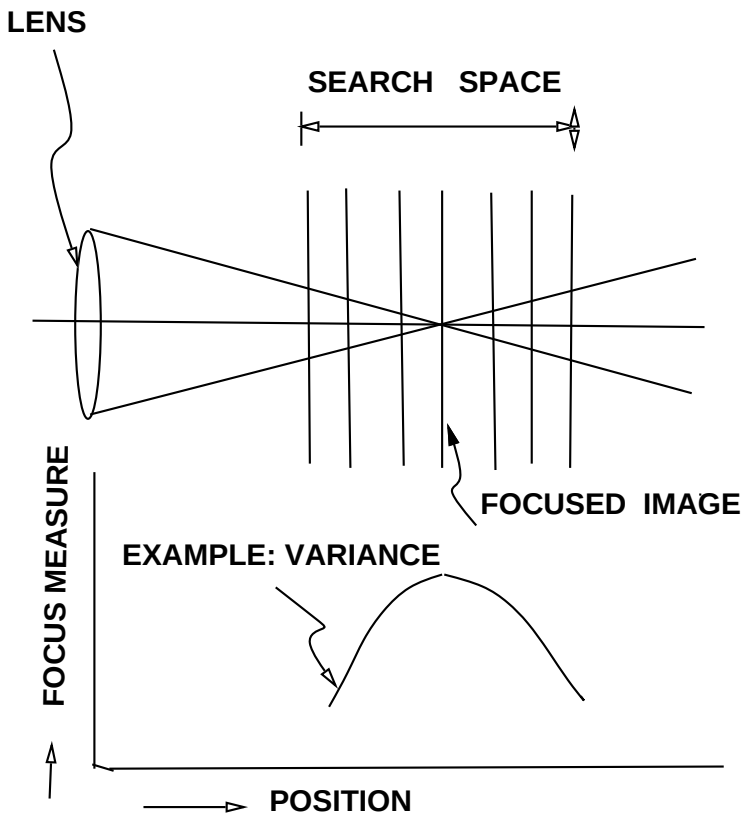


Figure 3-2: Depth-from-Focus

values of $|\bigtriangledown g(x,y)|$, which are greater than a threshold value. The partial derivatives may be estimated by many discrete operators.

2. Laplacian

The Laplacian of an image is obtained as

$$\bigtriangledown^2 g(x,y) = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2} \tag{3.3}$$

The Laplacian is a kind of high pass filter, which measures the high frequency content of an image. By summing up the square of the Laplacian over an image region, a focus measure may be obtained. There are many discrete operators which yield an approximate value of the Laplacian of an image.

3. Gray Level Variance

By viewing the gray levels in an image as random variables, the variance of an image is defined as

$$\sigma^2 = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} g(x,y) - \mu^2 \tag{3.4}$$

where μ is the mean of the gray level distribution and size of the image being $N \times N$.

4. Sum Modulus Difference

The sum-modulus-difference was proposed by Jarvis [60] and is computed as

$$SMD_x = \sum_x \sum_y |g(x,y) - g(x,y-1)| \quad \text{and} \tag{3.5}$$

$$SMD_y = \sum_x \sum_y |g(x,y) - g(x+1,y)| \tag{3.6}$$

The focus measure is then obtained as

$$SMD = SMD_x + SMD_y \tag{3.7}$$

5. Sum Modified Laplacian

The sum-modified Laplacian was recently proposed by Nayar [88]. It is computed as

$$SML = \sum_x \sum_y \left| \frac{\partial^2 g(x,y)}{\partial x^2} \right| + \left| \frac{\partial^2 g(x,y)}{\partial y^2} \right| \tag{3.8}$$

It is a variation of a simple Laplacian, and Nayar used it effectively on microscope images.

6. Bandpass Filters

Subbarao *et al* [119] recently proposed some focus measures, which perform bandpass filtering on an image, instead of the usual high pass filtering.

Considering geometric optics, the OTF of an optical system is circularly symmetric and a cross section looks as shown in Figure 2.7 - 2.10. There is a main lobe and many side lobes in the OTF. If the focused image of an object has high energy frequency content in the side lobes, then the focus measures such as Laplacian and variance may exhibit local maxima. This complicates the task of searching for the global maximum of a focus measure. In order to overcome the problem caused by the OTF sidelobes, lowpass filtering with a Gaussian was proposed. The combination of the lowpass filter and highpass filter (Laplacian or variance), results in a bandpass filter.

Another advantage of lowpass filtering is the reduction of noise and the attenuation of aliased frequencies near the high end of the spectrum.

3.4 Window Size for Focus Measure Computation

If a planar object is in the scene, then a focus measure can be computed over the entire image, by summing up or integrating over the entire image. For example,

$$\sigma^2 = \frac{1}{N^2} \sum_x \sum_y g(x, y) - \mu^2 \tag{3.9}$$

where the summation is done over a region of size $N \times N$. But as indicated in Figure 3.1, if the scene consists of 3-D objects, then different points of the scene are at different distances from the camera and hence different regions of the image are blurred by different degrees. Most of the researchers solve this problem by considering a small window size of about 15×15 pixels and assume that the surface is planar in those small regions (piecewise planar approximation). If the window size is made too small, the focus measure is more susceptible to the image overlap problem and noise and may lead to the problem of local maxima. Hence, there is a trade-off involved in choosing a window size for focus measure computation. It has been suggested in [119] that the focus measure should be computed and maximized over the focused image surface (FIS) to yield more accurate shape recovery as compared to the methods using the piecewise planar approximation.

3.5 Advantages of Depth-from-Focus (DFF)

In general, the advantages of Depth-from-Focus (DFF) methods over other methods of depth estimation are the following.

1. No Correspondence Problem: In binocular stereo vision, two rays of light from the same point on an object are obtained from two imaging systems which have a different view or projection. Combining the conjugate points on both images is known as the correspondence problem. Most of the algorithms for solving the correspondence problem are computation intensive and heuristic in nature.

There is no correspondence problem in DFF methods, as only a single optical system is used.

2. No Occlusion Problem: Occlusion is another serious problem with binocular stereo vision. If there are occluding surfaces (one surface in the foreground and another in the background), then all the objects present in one image may not be present in the other image. Hence establishing correspondence will be even more difficult.

There is no occlusion problem with DFF methods, if a relatively small window size is chosen for focus measure computation.

3. Parallel Implementation: Since computation of a focus measure usually involves local computations, a DFF algorithm can be implemented with parallel processing elements.

4. Single Imaging System: The requirement of a single imaging system for DFF leads to a more compact hardware (instrument). In some applications such as microscopes and endoscopes, it is not feasible to have two or more

cameras, as the objects may be too small and the overall size of the system may be a critical design issue.

3.6 Disdvantages of Depth-from-Focus (DFF)

1. Large number of images: The most important disadvantage of DFF methods is that it requires a large number of images, taken with different camera parameters. Usually, for a planar object perpendicular to the optical axis, about 10-12 images are required to obtain a depth estimate using DFF. If the scene consists of 3-D objects the number of images may be very large (about 25 in our experiments).

2. Stationary Scene: Moving the lens to a different position for acquiring an image involves mechanical motion of the lens system and hence could be time consuming. Moreover, the scene has to remain stationary during the entire period of time taken to acquire all the images. If the scene consists of objects which are moving slowly, DFF methods are diffcult to use.

The Depth-from-Defocus (DFD) algorithm that we investigate in the subsequent chapters requires only two or three images and hence eliminates some of the limitations of DFF.

3.7 Depth-from-Defocus (DFD)

The previous sections described some of the Depth-from-Focus (DFF) algorithms. DFF methods are essentially search methods, which search for a set of camera parameters corresponding to the sharpest image of the scene.

Recently some researchers have pursued a different class of algorithms known as the Depth-from-Defocus (DFD) algorithms. In this section we review some of the recent work on DFD.

DFD methods do not require focusing the object. They directly estimate the distance of an object from a measure of the level of defocus. These methods require only a few images (about 2-3) for recovering distance of an object, as compared to DFF methods which require a large number of images (typically about 10-12 for a planar object). In addition, only a few images are sufficient to determine the distance of all objects in the scene using DFD methods, irrespective of whether the objects are focused are not. For a 3-D scene, DFF methods require many images (typically about 25 images).

As mentioned in the previous sections, the image of a scene formed by an optical system contains information about the distance (or depth) of objects in the scene. Objects at a particular distance are focused, whereas other objects are defocused (blurred) by different degrees. The amount of blur depends on the distance of the objects and also the characteristics of the imaging system. By measuring the amount of blur at any given point in an image, it seems possible that the distance of the corresponding point in the scene could be determined.

The central problem in DFD is then to measure the *amount* of blur in a small region of an image. The image $g(x, y)$ recorded on an image detector, is a result both of the characteristics of the scene and of the lens system. It has been shown in equation (2.5) that in the Fourier domain,

$$G(\omega, \nu) = F(\omega, \nu)\, H(\omega, \nu) \tag{3.10}$$

where $G(\omega, \nu)$ is the Fourier Transform of the observed image, $F(\omega, \nu)$ is the Fourier Transform of the focused image and $H(\omega, \nu)$ is the Fourier Transform of the PSF, $h(x, y)$. If the underlying focused image $f(x, y)$ is known (or its Fourier Transform $F(\omega, \nu)$ is known), possibly the PSF $h(x, y)$ can be recovered. Conversely, if $H(\omega, \nu)$ is known, the focused image $f(x, y)$ could possibly be recovered from the corresponding blurred image $g(x, y)$. The latter is a case of image restoration. However, in the general case, neither $f(x, y)$ nor $h(x, y)$ is known, resulting in two unknown functions and hence requiring two images for a solution.

There are a number of DFD algorithms falling into different categories and varying largely in application and complexity. The classification of the different algorithms is shown in Figure 3.3. In the following sections we briefly review these algorithms.

3.8 Edge based Methods

Some of the early DFD algorithms were based on the assumption that the underlying focused image is a sharp step edge. Pentland [90] modeled a blurred step edge as a result of convolving a focused image with a Gaussian PSF. He showed that if $C(x, y)$ is the Laplacian of the observed image, then the spread σ of the Gaussian is related to $C(x, y)$ by

$$\ln \left[\frac{b}{\sqrt{2\pi}\sigma^3}\right] - \frac{x^2}{2\sigma^2} = \ln \left|\frac{C(x, y)}{x}\right| \tag{3.11}$$

where b is the magnitude (or height) of the step edge and x, y the image coordinate system with its origin on the edge and x - axis perpendicular to

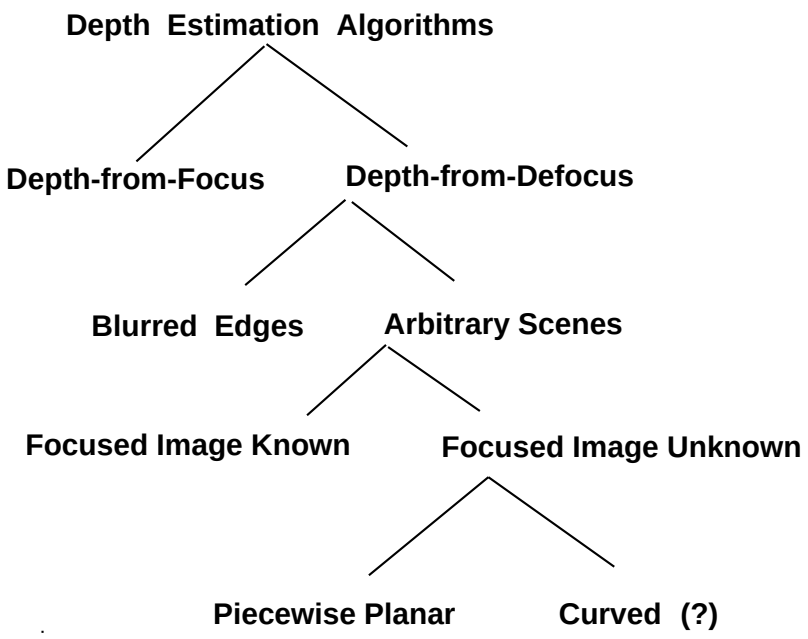


Figure 3-3: Classification of DFD algorithms

the edge. In the above equation b and σ are the only unknowns. He solved for these unknowns by formulating the above equation as a linear regression in x^2. The depth was then computed from σ. Pentland applied his method to an image of a natural scene and showed that depth of edges could be classified as being small, medium or large.

Grossman [48] showed experimentally that useful depth information can be obtained from blurred edges. In addition to step edges, he considered edges of finite width, such as ramp edges. Grossman however provided no theoretical justification for his method.

Subbarao and Gurumoorthy [107] presented a new approach for finding the distance of step edges. In their approach, the image $f(x, y)$ of a step edge is expressed as

$$f(x, y) = a + b\, u(x) \tag{3.12}$$

where $u(x)$ is a unit step function. The blurred image $g(x, y)$ of the step edge $f(x, y)$ is expressed as

$$g(x, y) = h(x, y) * f(x, y) \tag{3.13}$$

The line spread function (LSF) of the system is then shown to be

$$\theta(x) = \frac{\frac{\partial g}{\partial x}}{\int_{-\infty}^{\infty} \frac{\partial g}{\partial x} dx} \tag{3.14}$$

The *spread parameter* σ_l of the LSF is defined as

$$\sigma_l^2 = \int_{-\infty}^{\infty} (x - \overline{x})^2 \theta(x) dx \tag{3.15}$$

where $\overline{x}$ is the *center of mass* of the distribution defined by

$$\overline{x} = \int_{-\infty}^{\infty} x\theta(x) dx \tag{3.16}$$

Subbarao and Gurumoorthy showed that the spread parameter σ_l is related to the distance of the object u by

$$\sigma_l = mu^{-1} + c \tag{3.17}$$

The computational method of this approach is much simpler than that of Pentland. Also, the solution for the edge height b and the spread σ_l are given in closed form. More importantly, the method just assumes that the PSF is circularly symmetric and is not restricted to the Gaussian or cylindrical models. Subbarao [109] later extended his algorithm to include simple objects such as blobs, stripes and smooth edges (in contrast to sharp step edges).

Lai, Fu and Chang [75] generalized the edge based method of Pentland [90] to include edges with any arbitrary orientation. It is an optimization method and is claimed to be less sensitive to noise as it does not involve any differentiation.

3.9 Algorithms for Arbitrary Scenes

The methods discussed in the previous section are all edge based methods. They are based on the assumption that the scene consists of sharp step edges. A natural scene, on the other hand, consists of many arbitrary intensity patterns. Recently a few researchers have attempted to develop DFD algorithms, which work for arbitrary scenes.

Pentland [92, 93] proposed an algorithm where he used two images of the same scene, one obtained with a pin-hole camera and the other with a wide

aperture camera. The image obtained with a pin-hole camera is focused everywhere, while the other image is blurred by different amounts in different regions, depending on the distance of the corresponding points from the camera. Pentland was able to compare corresponding points in the two images and measure the change in focus, which is related to depth.

He considered the first image $f_1(r,\theta)$ as a result of convolving the focused image $f_0(r,\theta)$ with a Gaussian $G(r,\sigma_1)$. Thus

$$f_1(r,\theta) = f_0(r,\theta) * G(r,\sigma_1) \tag{3.18}$$

Similarly, a second blurred image could be expressed as

$$f_2(r,\theta) = f_0(r,\theta) * G(r,\sigma_2) \tag{3.19}$$

Taking the Fourier Transforms and natural logarithm, it was shown that

$$\ln \frac{\sigma_2^2}{\sigma_1^2} + \rho^2 2\pi^2(\sigma_2^2 - \sigma_1^2) = \ln F_1(\rho) - \ln F_2(\rho) \tag{3.20}$$

where $F_1(\rho,\theta)$ and $F_2(\rho,\theta)$ are Fourier Transforms of $f_1(r,\theta)$ and $f_2(r,\theta)$ respectively and

$$F_1(\rho) = \int_{-\pi}^{\pi} F_1(\rho,\theta)d\theta \tag{3.21}$$

and

$$F_2(\rho) = \int_{-\pi}^{\pi} F_2(\rho,\theta)d\theta. \tag{3.22}$$

Since the first image is obtained with a pin-hole camera ($\sigma_1 = \epsilon$ for some value ϵ), the above equation becomes

$$k_1\sigma_2^2 + k_2\ ln\sigma_2 + k_3 = \ln F_1(\rho) - \ln F_2(\rho) \tag{3.23}$$

Thus the difference in Fourier power is a monotonic increasing function of the blur in the second image.

In Pentland's implementation the image was first filtered with a bandpass filter. Parseval's theorem was then used to obtain the Fourier power in a narrowband by computing the energy in the spatial domain. In experiments over a 1-cubic meter work space, he reported a standard error of 2.5 percent.

The main disadvantage of Pentland's algorithm is that it assumes that the focused image is known. In his experiments the focused image was obtained with a pin-hole camera. A pin-hole camera has a very long exposure time, and the scene has to remain stationary during the entire period. Also the PSF of a pin-hole camera will be dominated by diffraction effects which have to be taken into account. An image detector such as a CCD array, is more susceptible to noise, at the low intensity levels at which a pin-hole camera operates.

Subbarao and Wei [114, 115] proposed a Fourier domain approach, called DFD1F, for determining depth using defocus information. Their method requires two images taken with two different sets of camera parameters. If $g_1(x, y)$ is the first blurred image, then the corresponding Fourier Transform $G_1(\omega, \nu)$ can be expressed as

$$G_1(\omega, \nu) = F(\omega, \nu)H_1(\omega, \nu) \tag{3.24}$$

where $F(\omega, \nu)$ is the Fourier Transform of the focused image and $H_1(\omega, \nu)$ is the Fourier Transform of the PSF corresponding to the first set of camera parameters. For a second blurred image, a similar relation is obtained,

$$G_2(\omega, \nu) = F(\omega, \nu)H_2(\omega, \nu) \tag{3.25}$$

To illustrate the idea of DFD1F let the PSF be a 2-D Gaussian,

$$H(\omega, \nu) = e^{-\frac{1}{2}\sigma^2(\omega^2+\nu^2)} \tag{3.26}$$

Dividing equations (3.24) and (3.25) and simplifying using equation (3.26),

$$\sigma_1^2 - \sigma_2^2 = -\frac{2}{\omega^2 + \nu^2} \ln \frac{|G_1(\omega, \nu)|}{|G_2(\omega, \nu)|} \tag{3.27}$$

σ_1 and σ_2 are related to the camera constants α and β as

$$\sigma_1 = \alpha\sigma_2 + \beta_2 \tag{3.28}$$

The final equation for σ_2 is of the form

$$(\alpha^2 - 1)\sigma_2^2 + 2\alpha\beta\sigma_2 + \beta^2 = -\frac{2}{\omega^2 + \nu^2} \ln \frac{|G_1(\omega, \nu)|}{|G_2(\omega, \nu)|} \tag{3.29}$$

In the above quadratic equation σ_2 is the only unknown. In general, two solutions for σ_2 will be obtained and methods for resolving the two-fold ambiguity are discussed in [115]. The distance u is directly obtained from σ_2. Repeating the above procedure in all image neighborhoods a rough depth map of the scene could be obtained.

Subbarao and Wei implemented their method on an actual camera system and reported an RMS error of 3.5 percent in lens position, over a large number of experiments. The advantages of DFD1F are that it requires only two images, both of which can be arbitrarily blurred. Also, DFD1F is not restricted to any particular form of the PSF. The only drawback of DFD1F is that it requires a relatively large window size of about 128 $\times$ 128 pixels for computation.

3.9.1 Matrix Based Regularization

Ens [35, 36] proposed a matrix based regularization method for estimating depth using defocus information. He enumerated many problems with the inverse filtering approach used by other algorithms and proposed a new method based on regularization theory.

In his method two images $i_1(x, y)$ and $i_2(x, y)$ are acquired with two different defocus operators (PSFs). If $h_1(x, y)$ and $h_2(x, y)$ are the corresponding PSFs, then

$$i_1(x, y) = s(x, y) * h_1(x, y) + n_1(x, y) \tag{3.30}$$

and

$$i_2(x, y) = s(x, y) * h_2(x, y) + n_2(x, y) \tag{3.31}$$

where $n_1(x, y)$ and $n_2(x, y)$ are noise in the image acquisition process. The goal of the algorithm is to find a new PSF $h_3(x, y)$ such that

$$i_1(x, y) * h_3(x, y) = i_2(x, y) \tag{3.32}$$

It was shown that for geometric optics $h_3(x, y)$ is a unique indicator of depth. However in the general case, uniqueness of $h_3(x, y)$ could not be proved. Ens argued that $h_3(x, y)$ must belong to a family of patterns, which can be known *a priori*. The problem was formulated as minimization of the functional

$$\| [i_{1T}] .h_3 - i_2 \|^2 + \lambda \| [C] .h_3 \|^2 = minimum \tag{3.33}$$

where $[i_{1T}]$ is a Toeplitz matrix formed from $i_1(x, y)$

λ is a scalar parameter

and $[C]$ is a matrix which minimizes the magnitude of the second term if h_3 belongs to the expected family of patterns.

The Euler equation for h_3 was solved as

$$h_3 = \left([i_{1T}]^T [i_{1T}] + \lambda [C]^T [C]\right)^{-1} [i_{1T}]^T i_2 \tag{3.34}$$

Solving equation (3.34) was found to be extremely computation intensive. Additionally, finding $[C]$ is difficult for anything other than simple parametric families. Hence an iterative method was proposed, which iteratively minimizes

$$\sum_{x=0}^{N-k} \sum_{y=0}^{N-k} \left[i_1(x,y) [\times] \hat{h}_3(x,y) - i_2(x,y)\right]^2 = minimum \tag{3.35}$$

A table of $h_3(x,y)$ values corresponding to different distances of the object was obtained by calibration. The table was searched for the best value of $h_3(x,y)$ that minimizes equation (3.35). In Ens' experiments object distance was varied from 0.80 to 0.95 meters and an RMS error of 2.8 percent in terms of the expected range was reported.

The important disadvantages of this method are that it is computationally too expensive and it requires calibration by experimental determination of PSF for different distances.

3.9.2 Differential Methods

Subbarao [105] proposed some depth estimation algorithms for arbitrary objects by considering the effect of differential changes in camera parameters on the amount of blur in an image. The camera parameters could be position of the image detector plane, camera aperture diameter or focal length. In all the

cases, the algorithms involved the computation of the change in power spectral density across two blurred images. The implementation of these algorithms is not feasible as a differential change in camera parameters produces only a differential change in blur, which is difficult to measure.

Chapter 4

Depth-from-Defocus: A Spatial Domain Approach

4.1 Introduction

In this chapter we provide the theoretical basis and implementation details of a new method for determining distance based on image defocus. The method is based on a new Spatial Domain Convolution/ Deconvolution Transform or S-Transform, introduced recently [113]. This method is called the *S-Transform Method* or STM. STM requires only two images taken with different camera parameters such as lens position, focal length and aperture diameter. Both images can be arbitrarily blurred and neither of them needs to be a focused image. STM therefore, is very fast in comparison with the Depth-from-Focus (DFF) methods, which search for the lens position or focal length of best focus. Also, STM involves only simple local operations and can easily be implemented in parallel to obtain the depth-map of a scene. It has been implemented on an actual camera system named SPARCS. Experiments on

the performance of STM and their results on real-world objects are described in this chapter. The results indicate that the accuracy of STM compares well with DFF methods and is useful in practical applications. The utility of the method is demonstrated for rapid autofocusing of electronic cameras.

4.2 S Transform

A new Spatial-Domain Convolution/ Deconvolution Transform (S Transform) has been developed for images and n-dimensional signals [113] for the case of arbitrary order polynomials. The transform has been defined for both continuous signals and discrete signals. Here we summarize briefly only those results relevant to STM. Details may be found in [113].

4.2.1 Forward S-Transform

Let $f(x, y)$ be an image which is a two variable cubic polynomial defined by

$$f(x, y) = \sum_{m=0}^{3} \sum_{n=0}^{3-m} a_{m,n} x^m y^n \tag{4.1}$$

where $a_{m,n}$ are the polynomial coefficients. (This restriction on the form of f will be relaxed later.) Let $h(x, y)$ be a rotationally symmetric point spread function. The moments of the point spread function are defined by

$$h_{m,n} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^m y^n h(x, y)\, dxdy \tag{4.2}$$

Now consider the convolution of the image $f(x, y)$ and the point spread function $h(x, y)$

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x - \zeta, y - \eta) h(\zeta, \eta) \, d\zeta d\eta \tag{4.3}$$

Since f is a cubic polynomial, it can be expressed in a Taylor series as

$$f(x - \zeta, y - \eta) = \sum_{0 \leq m+n \leq 3} \frac{(-\zeta)^m}{m!} \frac{(-\eta)^n}{n!} f^{m,n}(x, y) \tag{4.4}$$

$$\text{where} \qquad f^{m,n}(x, y) \equiv \frac{\partial^m}{\partial x^m} \frac{\partial^n}{\partial y^n} f(x, y) \tag{4.5}$$

Using equation (4.4) in equation (4.3) we obtain

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{0 \leq m+n \leq 3} \frac{(-1)^{m+n}}{m!n!} f^{m,n}(x, y) \zeta^m \eta^n h(\zeta, \eta) \, d\zeta d\eta \tag{4.6}$$

Interchanging the order of summation and integration,

$$g(x, y) = \sum_{0 \leq m+n \leq 3} \frac{(-1)^{m+n}}{m!n!} f^{m,n}(x, y) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \zeta^m \eta^n h(\zeta, \eta) \, d\zeta d\eta \tag{4.7}$$

$$= \sum_{0 \leq m+n \leq 3} \frac{(-1)^{m+n}}{m!n!} f^{m,n}(x, y) h_{m,n} \tag{4.8}$$

Equation (4.8) expresses the convolution of a function $f(x, y)$ with another function $h(x, y)$, as a summation involving the derivatives of $f(x, y)$ and moments of $h(x, y)$. This corresponds to the *forward S-Transform.*

To illustrate in more detail, the effect of blur on a cubic polynomial, equation (4.1) can be expanded and written as

$$\begin{aligned} f(x, y) &= a_{0,0} + a_{1,0}x + a_{0,1}y + a_{2,0}x^2 + a_{1,1}xy \\ &\quad + a_{0,2}y^2 + a_{3,0}x^3 + a_{2,1}x^2y + a_{1,2}xy^2 + a_{0,3}y^3 \end{aligned} \tag{4.9}$$

There are 10 terms in the above equation. When $f(x, y)$ is blurred with a circularly symmetric PSF $h(x, y)$, then all the odd moments of $h(x, y)$ vanish.

Therefore,

$$h_{0,1} = h_{1,0} = h_{1,1} = h_{0,3} = h_{3,0} = h_{2,1} = h_{1,2} = 0 \text{ and } h_{2,0} = h_{0,2}. \tag{4.10}$$

Using this in equation (4.8) and writing out all the terms in the summation we get,

$$g(x,y) = f(x,y)h_{0,0} + f^{2,0}(x,y)h_{2,0} + f^{0,2}(x,y)h_{0,2} \tag{4.11}$$

Differentiating equation (4.9) twice with respect to x, we get

$$f^{2,0}(x,y) = 2a_{2,0} + 6a_{3,0}x + 2a_{2,1}y \quad \text{and} \tag{4.12}$$

$$f^{0,2}(x,y) = 2a_{0,2} + 6a_{0,3}y + 2a_{1,2}x \tag{4.13}$$

Using these results in equation (4.11), $g(x,y)$ can be expressed as

$$\begin{aligned} g(x,y) &= (a_{0,0} + 2a_{2,0} + 2a_{0,2}) + (a_{1,0} + 6a_{3,0}h_{2,0} + 2a_{1,2}h_{2,0})x \\ &\quad + (a_{0,1} + 6a_{0,3}h_{2,0} + 2a_{2,1}h_{2,0})y + a_{2,0}x^2 + a_{1,1}xy + a_{0,2}y^2 \\ &\quad + a_{3,0}x^3 + a_{2,1}x^2y + a_{1,2}xy^2 + a_{0,3}y^3 \end{aligned} \tag{4.14}$$

It can be seen that the quadratic and higher order terms are unaffected by the process of blurring and only the constant and linear terms are affected. But the constant and linear terms are affected by the higher order terms. If an image region does not have too much variation in intensity, then the higher order terms will be relatively very small in magnitude and hence the process of blurring will not affect such an image region. As we will see later, we cannot determine the distances of such objects which are smoothly varying in intensity.

4.2.2 Inverse S-Transform

To derive a deconvolution formula, a similar approach is followed. From the definition of a point spread function, the volume enclosed by it is unity. Therefore,

$$h_{0,0} = 1 \tag{4.15}$$

Using this result and equation (4.10) in equation (4.8) we find that most of the terms vanish and hence

$$g(x,y) = f(x,y) + \frac{h_{2,0}}{2}\left\{f^{2,0}(x,y) + f^{0,2}(x,y)\right\} \tag{4.16}$$

From equation (4.16) it is possible to express $f(x,y)$ as

$$f(x,y) = g(x,y) - \frac{h_{2,0}}{2}\left\{f^{2,0}(x,y) + f^{0,2}(x,y)\right\} \tag{4.17}$$

Applying $\frac{\partial^2}{\partial x^2}$ to the above expression on either side and noting that derivatives of order higher than 3 are zero (because f is cubic), we obtain

$$f^{2,0}(x,y) = g^{2,0}(x,y) \tag{4.18}$$

Similarly applying $\frac{\partial^2}{\partial y^2}$ we get

$$f^{0,2}(x,y) = g^{0,2}(x,y) \tag{4.19}$$

Therefore using equations (4.18) and (4.19) in equation (4.17) we obtain

$$f(x,y) = g(x,y) - \frac{h_{2,0}}{2} \bigtriangledown^2 g(x,y) \tag{4.20}$$

where $\bigtriangledown^2$ is the Laplacian operator. Equation (4.20) is a deconvolution formula. It expresses the original function $f(x,y)$ in terms of the convolved

function $g(x, y)$, its derivatives and the moments of the point spread function. In the general case this corresponds to the *Inverse S-Transform* [113].

Using the definitions of the moments of h and the definition of the spread parameter σ_h of h, it can easily be shown that

$$h_{2,0} = h_{0,2} = \frac{\sigma_h^2}{2} \tag{4.21}$$

The above deconvolution formula can then be written as

$$f(x, y) = g(x, y) - \frac{\sigma_h^2}{4} \bigtriangledown^2 g(x, y) \tag{4.22}$$

Equation (4.22) is a very useful deconvolution formula and in the next section we describe the application of this formula to the problem of distance estimation from blurred images.

4.3 Determining Distance

In this section we develop a theoretical basis for determining distance. Let $f(x, y)$ be the focused image of a planar object at distance u. The *focused image* $f(x, y)$ at a point (x, y) of a scene is defined as the total light energy incident on the camera aperture (entrance pupil) during one exposure period from the object point along the direction corresponding to (x, y) [112].

Let $g_1(x, y)$ and $g_2(x, y)$ be two images of the object recorded for two different camera parameter settings $\mathbf{e}_1$ and $\mathbf{e}_2$ where

$$\mathbf{e_1} = (s_1, f_1, D_1) \quad \text{and} \quad \mathbf{e_2} = (s_2, f_2, D_2). \tag{4.23}$$

Let us assume that the images g_1 and g_2 are normalized with respect to magnification, brightness, and other factors such as sensor response and vignetting as necessary [110].

For a planar object perpendicular to the optical axis, the blur circle radius R' is a constant over the image of the object. In this case the camera acts as a linear shift invariant system. Therefore g_i will be equal to the convolution of the focused image $f(x, y)$ with the corresponding point spread function $h_i(x, y)$. In brief this can be expressed by

$$g_1(x, y) = h_1(x, y) * f(x, y) \text{ and} \tag{4.24}$$

$$g_2(x, y) = h_2(x, y) * f(x, y) \tag{4.25}$$

Let the spread parameter σ_h for h_1 be σ_1 and for h_2 be σ_2.

Now from equation (2.25) we can write

$$\sigma_1 = m_1 u^{-1} + c_1 \tag{4.26}$$

where

$$m_1 = -\frac{D_1 s_0}{2\sqrt{2}} \text{ and } c_1 = \frac{D_1 s_0}{2\sqrt{2}} \left[\frac{1}{f_1} - \frac{1}{s_1}\right]. \tag{4.27}$$

Similarly we obtain

$$\sigma_2 = m_2 u^{-1} + c_2 \tag{4.28}$$

where

$$m_2 = -\frac{D_2 s_0}{2\sqrt{2}} \text{ and } c_2 = \frac{D_2 s_0}{2\sqrt{2}} \left[\frac{1}{f_2} - \frac{1}{s_2}\right]. \tag{4.29}$$

Therefore,

$$u^{-1} = \frac{\sigma_1 - c_1}{m_1} = \frac{\sigma_2 - c_2}{m_2}. \tag{4.30}$$

σ_1 can then be expressed in terms of σ_2 as

$$\sigma_1 = \alpha\sigma_2 + \beta. \tag{4.31}$$

where

$$\alpha = \frac{m_1}{m_2} \text{ and } \beta = c_1 - c_2\frac{m_1}{m_2}. \tag{4.32}$$

We assume that in a small image neighborhood the focused image $f(x, y)$ can be adequately approximated by a cubic polynomial in (x, y) as in equation (4.1). This assumption will be relaxed in the next section. In our application, the image neighborhood is of size 9×9 pixels. Now we can apply the results from the previous section, particularly the deconvolution expression (4.22) and obtain the following relations:

$$f = g_1 - \frac{1}{4}\,\sigma_1^2\ \nabla^2\, g_1 \tag{4.33}$$

$$f = g_2 - \frac{1}{4}\,\sigma_2^2\ \nabla^2\, g_2 \tag{4.34}$$

In the above two relations, the dependence of all functions on (x, y) is understood but has been dropped from notation only for convenience. These two relations express the focused image f in terms of the blurred (observed) images g_1, g_2, and the spread parameters σ_1 and σ_2. Equating the right hand sides of equations (4.33) and (4.34) we obtain

$$g_1 - \frac{1}{4}\,\sigma_1^2\ \nabla^2\, g_1 = g_2 - \frac{1}{4}\sigma_2^2\,\nabla^2\, g_2\ . \tag{4.35}$$

It can be easily verified for a third degree polynomial f that

$$\nabla^2 g_1 = \nabla^2 g_2 \tag{4.36}$$

Therefore, in equation (4.35) $\nabla^2 g_1$ and $\nabla^2 g_2$ can be replaced by

$$\nabla^2 g \;=\; \frac{(\nabla^2 g_1 + \nabla^2 g_2)}{2} \tag{4.37}$$

Further, using relations (4.31) and (4.35) we obtain

$$a\,\sigma_2^2 \;+ b\,\sigma_2 \;+ c \;= 0 \tag{4.38}$$

where

$$a \;=\; \frac{1}{4}(\alpha^2 - 1)\,\nabla^2\, g \tag{4.39}$$

$$b \;=\; \frac{1}{2}\,\alpha\,\beta\;\nabla^2\, g \tag{4.40}$$

$$c \;=\; (g_2 - g_1) + \frac{1}{4}\,\beta^2\;\nabla^2\, g \tag{4.41}$$

The values of α and β are determined using the camera parameter values in relations (4.27), (4.29) and (4.32). The Laplacians $\nabla^2 g_1$ and $\nabla^2 g_2$ are computed from the two observed images g_1 and g_2. Therefore, the coefficients a, b and c can be computed from a knowledge of the camera parameters and the observed images using relations (4.39), (4.40) and (4.41). Having computed the coefficients a, b and c, we can solve for σ_2 by solving the quadratic equation (4.38). The distance u of the object is then obtained from equation (4.28). The fact that the quadratic equation results in two solutions for σ_2 does pose a problem. The easiest way to overcome this two-fold ambiguity is to reduce the equation to a linear equation by forcing the coefficient a to be zero. This approach is followed in one version of our implementation. As an alternative, one may record a third image g_3 and solve for σ_2 again using g_3 in place of g_1. The common root for σ_2 in the two cases gives the correct root. However, the ambiguity persists if the two roots for the second pair of images are the same

as that for the first pair of images. (This condition arises very rarely). This is the basic principle of STM.

We now describe the modifications necessary to make STM applicable in practice. Due to noise, the two focused images derived from the two blurred images may not be the same and equation (4.35) may not be valid. Another reason why equation (4.35) may not hold exactly is due to the influence of the neighboring windows on the current window being processed. This problem is known as the image overlap problem. In order to minimize these effects and make the method robust in the presence of noise, the following modifications are made. From equations (4.35) and (4.37) we get

$$g_1 - g_2 = \frac{1}{4}\,(\sigma_1^2 - \sigma_2^2)\;\; \bigtriangledown^2 g \tag{4.42}$$

Squaring first and then integrating over a small region around the point (x, y) we get

$$\int\int (g_1 - g_2)^2\; dxdy = \frac{1}{16}(\sigma_1^2 - \sigma_2^2)^2\;\; \int\int (\bigtriangledown^2 g)^2\; dxdy \tag{4.43}$$

which can be expressed as

$$(\sigma_1^2 - \sigma_2^2)^2 = G^2 \tag{4.44}$$

where

$$G^2 = 16\;\frac{\int\int (g_1 - g_2)^2 dxdy}{\int\int (\bigtriangledown^2 g)^2\; dxdy} \tag{4.45}$$

Thus,

$$(\sigma_1^2 - \sigma_2^2) = G' \tag{4.46}$$

where $G' = \pm G$.

The sign of G' is ambiguous, but this ambiguity is not inherent. It was introduced by the squaring of equation (4.42). The ambiguity can be resolved from the given images g_1 and g_2 in one of several ways. As one example, if g_1 is more blurred than g_2 then $\sigma_1^2 > \sigma_2^2$ and therefore the sign is positive, otherwise the sign is negative. It is easy to determine which of g_1 and g_2 is more blurred. From the theory of Depth-from-Focus methods [119] it is well-known that the gray-level variance of an image is a good measure of the degree of focus of the image. Therefore, if v_1 , v_2 are the gray-level variances of g_1, g_2 respectively, then the sign is positive if $v_1 < v_2$ and negative otherwise. Therefore

$$G' = \begin{cases} +G & \text{if } \ v_1 < v_2 \\ \\ -G & \text{otherwise} \end{cases}$$

Now substituting for σ_1 in terms of σ_2 using equation (4.31) into equation (4.46) yields

$$\sigma_2^2(\alpha^2 - 1) + 2\alpha\beta\sigma_2 + \beta^2 = G' \tag{4.47}$$

The above equation can be solved as a quadratic in σ_2.

In our experiments, two variations of STM, named STM1 and STM2, were implemented. In STM1 the lens position was changed in acquiring the two images g_1 and g_2. This resulted in changing the parameters s and f of the camera but the aperture diameter remained the same (i.e. $f_1 \neq f_2$ and $s_1 \neq s_2$ but $D_1 = D_2$). In this case we get $\alpha = 1.0$ and therefore the above quadratic equation in σ_2 reduces to a linear equation. Therefore we get the

unique solution:

$$\sigma_2 = \frac{G' - \beta^2}{2\beta} \tag{4.48}$$

In STM2, only the diameter of camera aperture was changed in acquiring the two images g_1 and g_2. All other camera parameters remained constant (i.e. $s_1 = s_2$ and $f_1 = f_2$ but $D_1 \neq D_2$). In this case we get $\beta = 0.0$ and $\alpha = D_1/D_2$. Therefore the quadratic equation in σ_2 reduces to

$$\sigma_2 = \pm\sqrt{\frac{G'}{\alpha^2 - 1}} \tag{4.49}$$

In this case we get two solutions for σ_2. One way to obtain a unique solution is to set $s_2 = f_2$. In this case the sign of the right hand side above is negative. This is the approach used in our implementation.

Ideally it should be possible to compute the value of σ_2 at one pixel (x, y) in the image and obtain an estimate of the distance. But because of noise and digitization, it is necessary to combine information from many pixels in an image region. In our implementation several alternatives were tried. Finally the following scheme worked well: σ_2 was computed at each pixel in a neighborhood of size 48×48 and a histogram of the values was obtained. The histogram was smoothed by a Parzen window [32] and the mode of the resulting distribution was taken to be the best estimate of σ_2. Once σ_2 is determined the object distance u can be obtained using a look-up table or calculated from equation (4.28).

4.4 Smoothed Differentiation Filters

In the previous section we assumed a local cubic polynomial model for the focused image $f(x, y)$ in deriving STM. This assumption can be removed by using a set of smoothing filters proposed by Meer and Weiss [85] so that STM can be applied to arbitrary focused images. Meer and Weiss [85] have proposed a set of discrete filters for smoothing images and estimating their derivatives. These filters essentially provide an efficient way for fitting polynomials to image brightness in small neighborhoods through simple separable convolution operation. The polynomial fitting is implicit and it is done subject to least-square error minimization. In our implementation we used one version of the filters where all data points have equal weight. The filters are based on Chebyshev polynomials as described in Appendix A in [85].

The filter for image smoothing by fitting a quadratic or cubic polynomial is

$$L_0(n) = -\frac{3[5n^2 - (3N^2 + 3N - 1)]}{(2N-1)(2N+1)(2N+3)} \tag{4.50}$$

where the support of the filter is $n = -N, -(N-1), \cdots, -1, 0, 1, \cdots, N-1, N$. This filter is separable and therefore can be first applied along rows and then along columns. The effective smoothing convolution kernel in this case is $L = L_0 * L_0^T$ where L_0^T is the transpose of L_0. The filter for estimating the second order image derivatives is

$$L_2(n) = -\frac{30[3n^2 - N(N+1)]}{N(N+1)(2N-1)(2N+1)(2N+3)} \tag{4.51}$$

In the implementation of STM, the result of applying the above filter along rows and columns were summed to get an estimate of the Laplacian of the

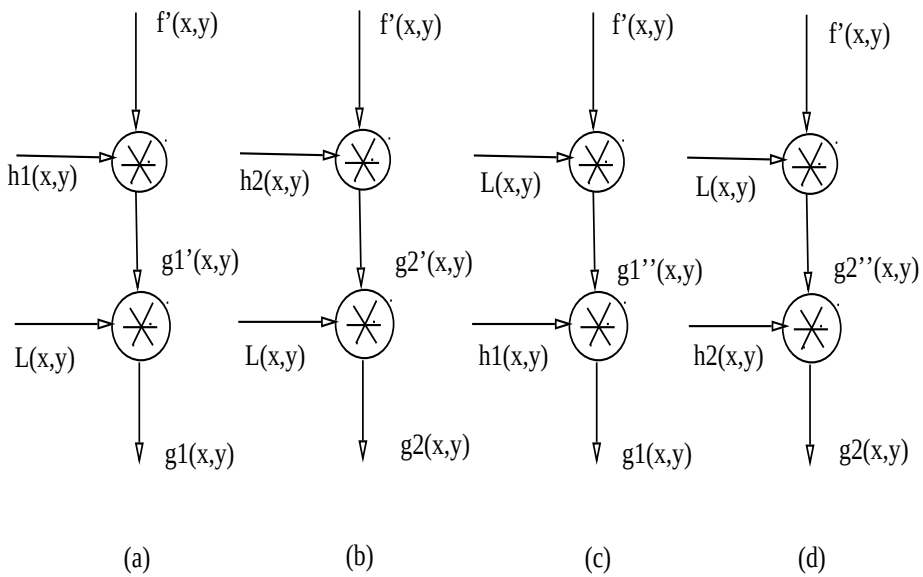


Figure 4-1: Least square filters

image. In our implementation N was chosen to be 4 so that the window size becomes 9×9 pixels.

Let the underlying focused image of an object be $f'(x, y)$ which is not a cubic polynomial, and let the two blurred images of f' corresponding to two point spread functions h_1 and h_2 be g_1' and g_2' respectively (Figure 4.1). Then we have $g_1' = h_1 * f'$ and $g_2' = h_2 * f'$. Now consider the effect of smoothing the blurred images g_1' and g_2' using a filter such as $L = L_0 * L_0^T$ which fits a cubic polynomial. If g_1 and g_2 are the smoothed images corresponding to g_1' and g_2' respectively, then we have $g_1 = L * (h_1 * f')$ and $g_2 = L * (h_2 * f')$ (Figure 4.1, (a) and (b)). Using commutative and associative properties of the convolution operation, we can write $g_1 = h_1 * (L * f')$ and $g_2 = h_2 * (L * f')$. Therefore, the smoothed images g_1 and g_2 can be thought of as the blurred images of

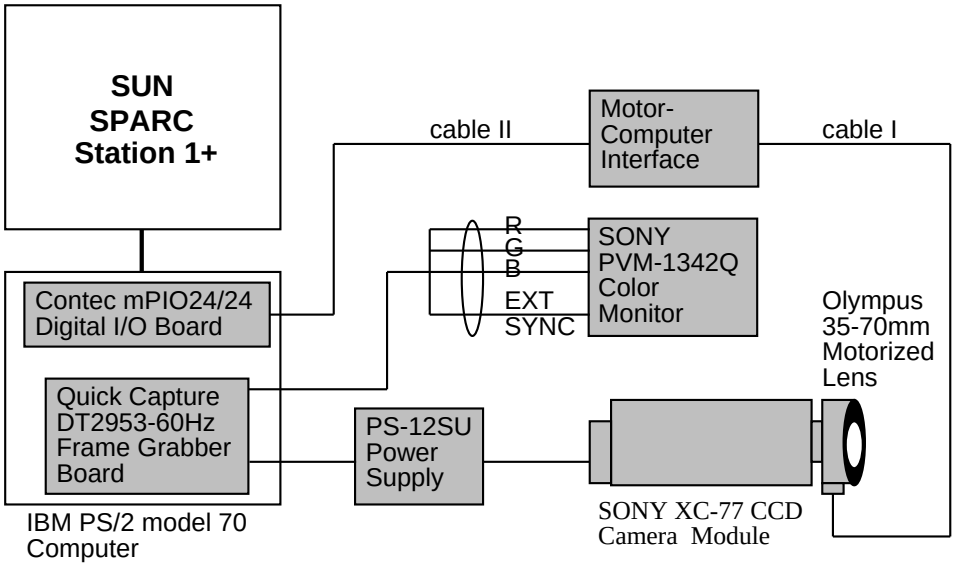
the focused image f where $f = L * f'$ (Figure 4.1, (c) and (d)). However, f is the result of fitting a cubic polynomial to the underlying focused image f'. Therefore g_1 and g_2 are the blurred images of a hypothetical focused image f which can be modeled by a cubic polynomial. For this reason, STM can be applied to g_1 and g_2 to estimate σ_2.

4.5 Implementation

4.5.1 SPARCS

STM described above was implemented on a camera system named Stonybrook Passive Autofocusing and Ranging Camera System (SPARCS). SPARCS was built over the last three years in our laboratory. A block diagram of the system is shown in Figure 4.2 (a). SPARCS consists of a SONY XC-77 CCD camera and an Olympus 35-70 mm motorized lens. Images from the camera are captured by a frame grabber board (Quickcapture DT2953 of Data Translation). The frame grabber board resides in an IBM PS/2 (model 70) personal computer. The captured images are processed in the PS/2 computer. The experimental setup is also shown in Figure 4.2 (b) and (c).

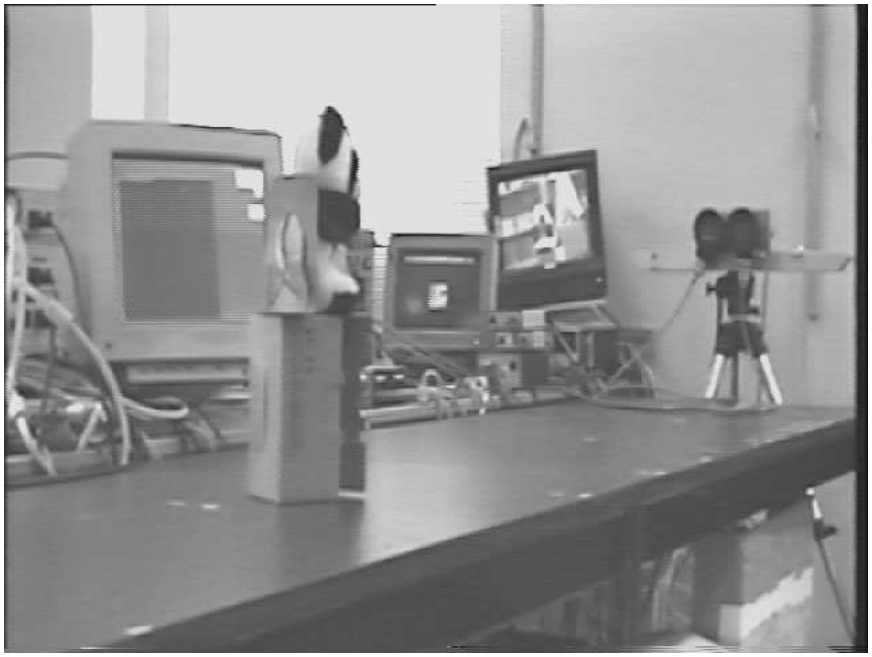
The lens system consists of multiple lenses and focusing is done by moving the front lens forward and backward. The lens can be moved either manually or under computer control. To facilitate computer control of the lens movement there is a stepper motor with 97 steps, numbered 0 to 96. Step number 0 corresponds to focusing an object at distance infinity and step number 96 corresponds to focusing a nearby object, at a distance of about 55cm from



Stonybrook Passive Autofocusing and Ranging Camera System-SPARCS - is a prototype camera system developed at the Computer Vision Laboratory for experimental research in robotic vision, State University of New York at Stony Brook.

Figure 4-2: (a) Block diagram of SPARCS

(b)



(c)

Figure 4.2 (b) and (c). Experimental setup

the lens. The motor is controlled by a microprocessor, which can communicate with the IBM PS/2 through a digital I/O board (Contec mPIO24/24). Pictures taken by the camera can be displayed in real-time on a color monitor (SONY PVM-1342 Q). The images acquired and stored in the IBM PS/2 can be transferred to a SUN workstation. In effect, the system is set up such that, a C program running on the PS/2 can move the lens to any desired step number and take pictures and process them.

Table 4.1 shows some important data of the lens used in SPARCS. In this table, the first column specifies lens position in terms of step number of the stepper motor. The second column is the focal length f, third column is the parameter s which specifies the distance between the image detector and the second principal plane of the lens, and the last column specifies the distance D_0 of an object which will be in best focus when the lens position is as specified in the first column. This data was obtained by the manufacturer of the lens by computer simulation and provided to us. This data is for the case when the zoom setting on the lens is 35 mm focal length. It is clear from this data that when the lens step number is changed, not only the parameter s but also the focal length changes by a small amount. Figure 4.3 shows a plot of the lens step number (the first column) along the x-axis and the reciprocal of best focused distance $1/D_0$ along the y-axis. This plot indicates that the lens step number and the reciprocal of best focused distance have an almost linear relationship. This is in fact predicted by the lens formula (3.1). Based on this relationship, we often find it convenient to specify distances of objects in terms of lens step number rather than in units of length such as meter. For

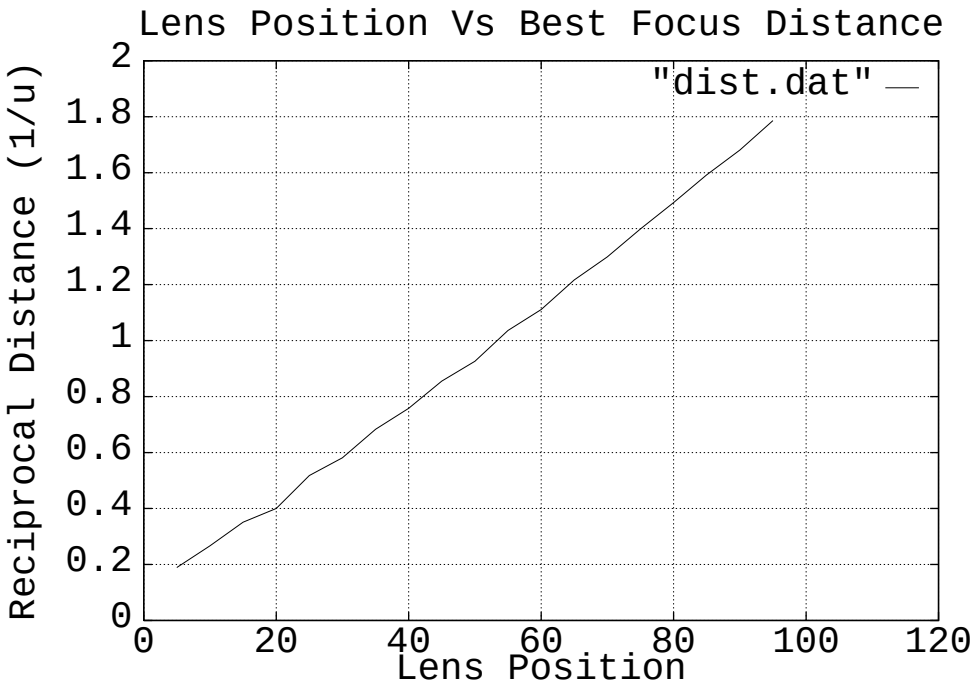


Figure 4-3: Lens step Vs best focused distance

example, when the "distance" of an object is specified as step number n, it means that the object is at such a distance D_0 that it would be in best focus when the lens is moved to step number n. The precise relationship between n and D_0 is given in Figure 4.3. In SPARCS, some experiments based on DFF methods indicated that the data of Table 4.1 should be shifted by 12 lens steps, i.e. a value of 12 should be added to each of the entries in the first column. We believe that this is due to mechanical assembly error between the lens and the CCD camera. We have taken this fact into account in reporting the results of our experiments in the following discussion.

A missing piece of information in Table 4.1 is the dependence of the diameter of camera aperture on lens position. We believe that the diameter

also changes by a small percentage when the lens moves from one end to the other, but this data is not available to us. Therefore we have taken the diameter to be constant. The diameter was calculated from the F-number and the zoom focal length.

As mentioned before, two versions of STM were implemented. In the first version named STM1, only the lens position was changed in obtaining two images g_1 and g_2, but the diameter of the lens aperture was not changed. Changing the lens position changes the parameters s and f as shown in Table 4.1. In the second version of STM named STM2, only the diameter of the lens aperture was changed but the other camera parameters s, f were unchanged in obtaining the two images. First we present the results for STM1 and then for STM2.

4.5.2 STM1

The overall operation of SPARCS for finding distance and autofocusing of an object is summarized as a flow chart in Figure 4.4. The stepwise operation is also explained briefly with comments below. In the experiments, initially, the zoom setting of the lens was set to be 35 mm focal length and the F-number was set to be 4. The camera gain was set to +6db.

The lens is first moved to step 10 and a first image $g_1(x, y)$ is obtained. Optionally we can specify the number of image frames (typically 4) to be recorded which are then averaged to reduce noise. Such frame averaging is particularly needed under low illuminations, and in the presence of flickering illumination such as fluorescent lamps. This was clearly evident from a number

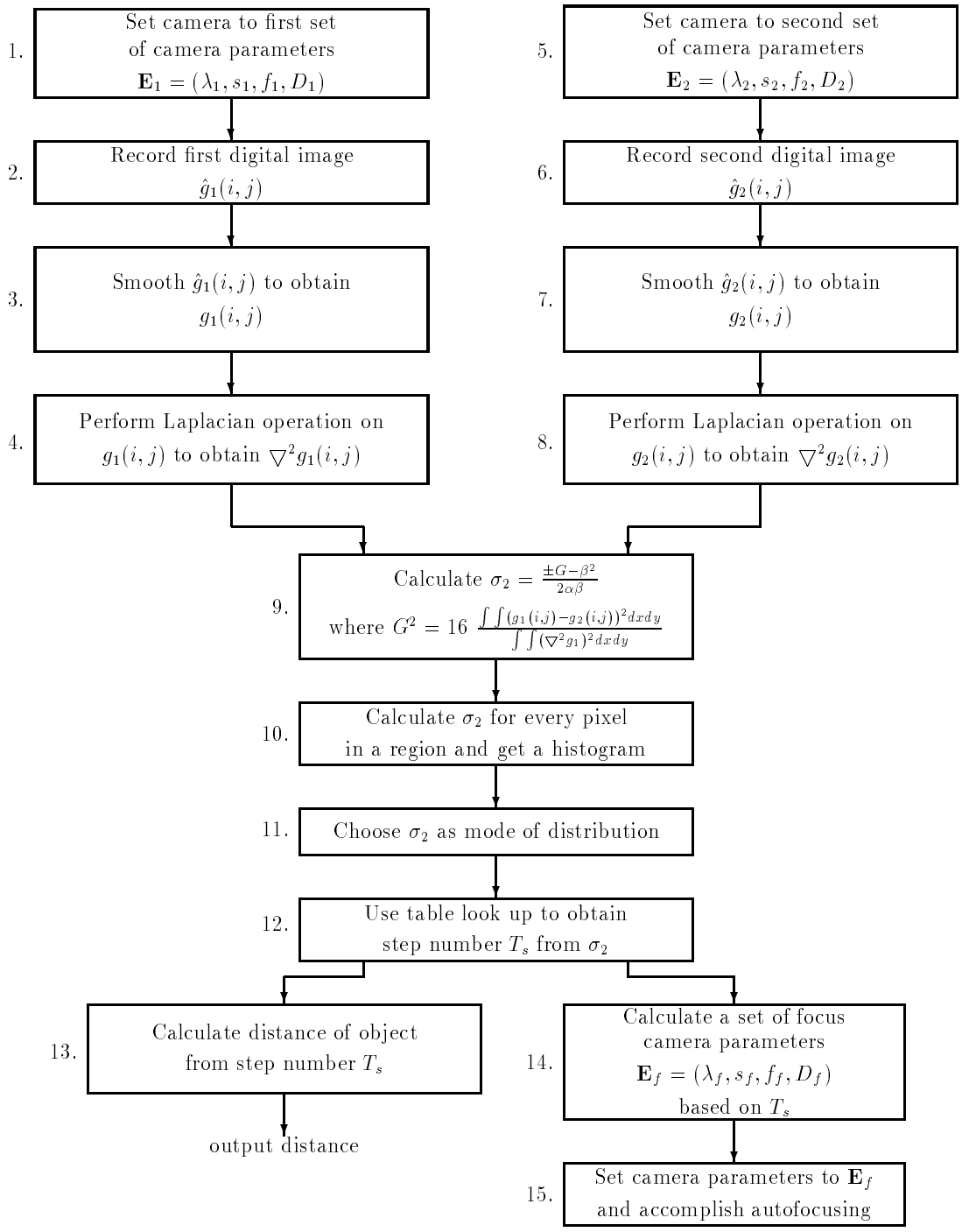


Figure 4-4: Flow chart of STM

of tests on SPARCS.

The lens is then moved to step 40 and a second image $g_2(x, y)$ is recorded. Again several frames may be recorded and averaged. The object to be ranged/ focused can be selected by specifying a region in the image. The default region is the center of the image. The size of the region is also an option and the default size is 72×72. The two images are then normalized with respect to brightness. This is done by dividing the grey level of each pixel by the mean grey level of the entire image. Our implementation does not normalize the images with respect to other types of distortions such as vignetting and sensor response characteristics, as their effects are not significant for our camera. As mentioned earlier we have also ignored the magnification normalization, as the change in magnification due to change in lens position was found to be negligible (about 2%).

The images are then smoothed using the least-squares polynomial fit filters proposed by Meer and Weiss [85]. The filter coefficients are derived from equation (4.50) and the filter size is 9×9. The Laplacian of the two smoothed images are then obtained using the differentiation filters of Meer and Weiss [85] given by equation (4.51).

The sign of G' is found by computing the gray-level variances of the original (unsmoothed) images g_1 and g_2. G^2 is calculated at every pixel by integrating over a 9×9 window centered at the pixel. G' is then calculated at every pixel. The value of the camera constants α and β are calculated from a knowledge of the camera parameters (see Table 4.1). An estimate of σ_2 is then obtained at every pixel using equation (4.48). Due to border effects of

smoothing filter and integration, the estimates of σ_2 is limited to the interior 48×48 region of the original 72×72 images. A histogram of the estimated σ_2 is computed. The bin size of the histogram was 0.1 (the expected range of σ_2 was from about -10.0 to +10.0). The histogram was smoothed using a Parzen Window of size 5. The mode of the histogram was taken to be the best estimate of σ_2. This value is used to estimate the distance of the object. In autofocusing application, from σ_2, the lens step number which will bring the object to focus is determined. The lens is then moved to this step number to accomplish autofocusing.

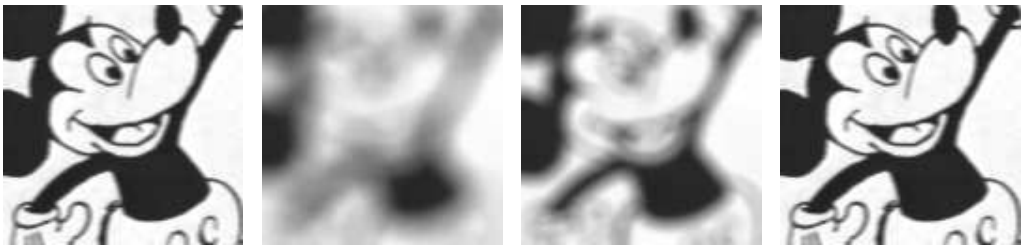
Once the value of σ_2 is estimated, equation (4.28) can be used to determine the distance of the object. However, in our implementation, for obtaining the object distance or lens step number for focusing from the computed value of σ_2, a look-up table is used. The look-up table itself is obtained through calibration and this method was found to be more accurate than the direct method of using equation (4.28). The calibration procedure used by us is as follows. First an object is placed at a known distance and then σ_2 is obtained exactly as described above. This procedure is repeated for several different objects at the same distance and the average σ_2 and the distance are recorded. This gives one entry of the look-up table. All other entries are obtained by repeating the above procedure for all possible distances of the objects. As for autofocusing, the relation between object distance and the best focused lens position was obtained by using a DFF algorithm which is based on the maximization of the energy of smoothed image gradient magnitude [119]. For every possible distance, the best focused lens step number was determined for

Focused 10 40 70

Figure 4-5: Focused and blurred images with object at step 10

several different objects and the mean was recorded in the look-up table.

In our experiments, it was found that if the object is very close to the camera, then the image g_1 is very highly blurred. In this case the results of ranging were unreliable. In such a case, a third image g_3 was taken after moving the lens to step number 70 and the image pair g_2 and g_3 were used in the estimation of distance. The use of this third image is due to practical reasons. Theoretically, only two images are necessary and sufficient. In our implementation, whenever the use of the first two images g_1, g_2 (taken at steps 10 , 40 respectively) resulted in an object distance greater than step 45, the third image g_3 was taken (at step 70). Some of the blurred images taken with lens steps 10, 40 and 70 along with the focused image are illustrated in Figures 4.5 and 4.6. It can be seen from the figures that one of the images can be highly blurred depending on the distance of the object, suggesting the need for a third image occasionally. The results of calibration are shown in Table 4.2. The calibration plots are shown in Figure 4.7. The first plot labeled "sigma2 (10 - 40)" was obtained with lens step positions of 10 and 40 and the second plot labeled "sigma2 (40 - 70)" was obtained with lens step positions



Focused 10 40 70

Figure 4-6: Focused and blurred images with object at step 60

of 40 and 70. It can be observed that for step numbers greater than 70, there are no proper values for sigma in the first plot.

Two typical histograms of σ_2 are shown in Figures 4.8 and 4.9. In these figures, the histogram obtained with first pair of images (g_1, g_2) is indicated by the plot "lens steps 10 - 40" and the histogram obtained with the second pair of images (g_2, g_3) is indicated by the plot "lens steps 40 - 70". In Figure 4.8, the object is closer to step 10 and hence the plot has a sharper peak for the first pair. In Figure 4.9, the object is closer to step 70 and hence the second pair of images yields a sharper histogram. In general as the object moves farther away from either of the two positions where the images are taken, the histogram becomes more and more flat. The Q-factor (ratio of peak value to width at half the peak value) of the histogram may be used as a goodness measure of the results.

Experiments were performed on five objects at normal room illumination (about 200 to 300 lux), five objects at 200 lux illumination, and 10 objects at 400 lux illumination. For each object the distance was varied from step 5 (step 10 in some cases) to step 95. All these images have been saved in an

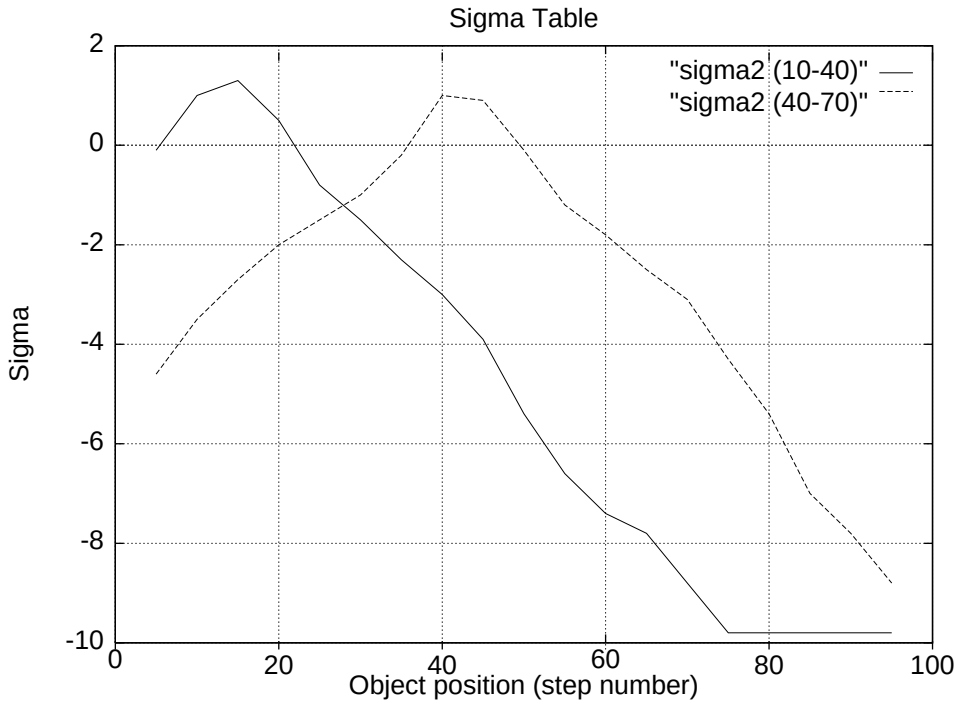


Figure 4-7: Calibration plots for STM1

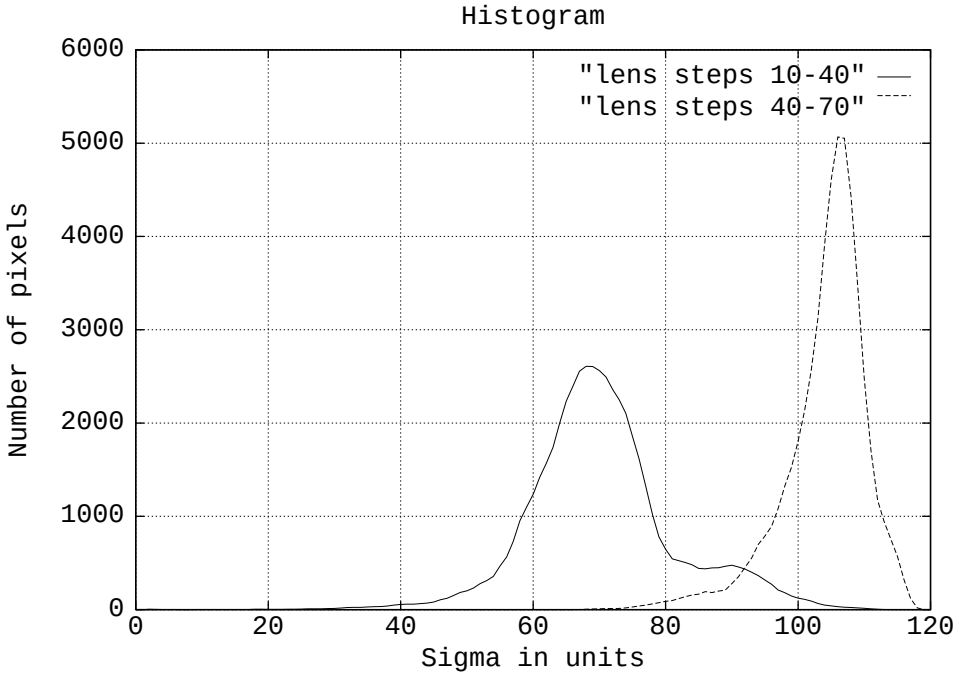


Figure 4-8: Histograms for an object at step 10

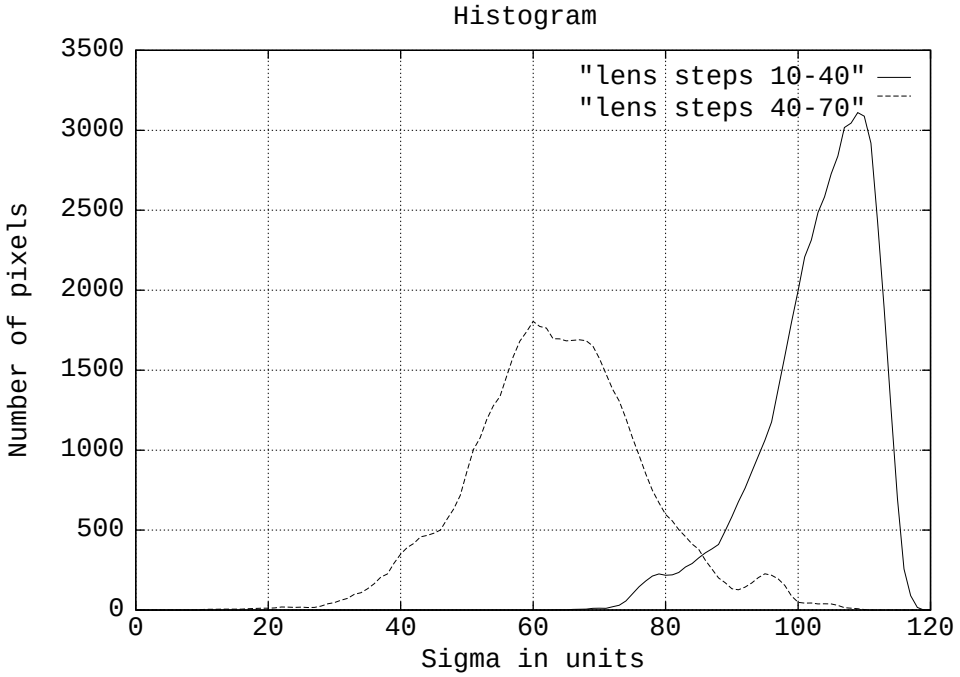


Figure 4-9: Histograms for an object at step 60

Figure 4-10: Test images in the database

image database named SPARCS.DB1. Some of the objects in the database are shown in Figure 4.10. The total number of experiments is 355. The results are tabulated in Tables 4.3, 4.4 and 4.5 along with the mean and standard deviations. The mean results are also plotted in Figures 4.11 - 4.13. The actual distance of the object measured in step numbers is along the x axis, and the estimated distance (in step number) is along the y axis. Under ideal conditions, the plots would have been diagonals running from bottom-left to top-right, which is indicated on the plots by the "ideal" curve. Some of the objects were very difficult ones such as thin lines and edges. The Root Mean Square (RMS) error was calculated for each of the three cases. Out of 97 steps, the RMS errors were 1.48 steps at room illumination, 2.26 steps error at 200 lux illumination, and 2.28 steps at 400 lux illumination. Since there are 97 steps, this error corresponds to about 2.5 percent error.

4.5.3 Error Analysis

We shall use the accuracy achieved by Depth-from-Focus methods as a benchmark against which to compare the accuracy of STM. The DFF methods usually take a large number of images (about 10-12) and search for the sharpest focus position by maximizing some focus measure. Many different focus measures have been proposed and the performances of many of them are nearly the same [119]. Since DFF methods involve exhaustive search for the focused position, we believe that the accuracy that can be obtained by any DFD method (which takes just 2-3 images) can at best be equal to a DFF method. Hence we shall call the results obtained by the DFF method

Step	FL	SI	D0
0	36.235	36.180	9034.0
5	36.086	36.132	5300.0
10	35.938	36.083	3750.0
15	35.72	36.035	2850.0
20	35.646	35.988	2500.0
25	35.502	35.941	1930.0
30	35.359	35.894	1720.0
35	35.217	35.848	1465.0
40	35.076	35.802	1320.0
45	34.937	35.757	1170.0
50	34.798	35.712	1080.0
55	34.661	35.667	965.0
60	34.524	35.622	900.0
65	34.389	35.578	822.0
70	34.255	35.534	770.0
75	34.121	35.491	715.0
80	33.989	35.448	670.0
85	33.859	35.406	628.0
90	33.728	35.363	595.0
95	33.380	35.250	560.0

Table 4.1. Lens Data

Stepno	$\sigma_{2,1}$	$\sigma_{2,2}$
05	-0.1	-4.6
10	1.0	-3.5
15	1.3	-2.7
20	0.5	-2.0
25	-0.8	-1.5
30	-1.5	-1.0
35	-2.3	-0.2
40	-3.0	1.0
45	-3.9	0.9
50	-5.4	-0.1
55	-6.6	-1.2
60	-7.4	-1.8
65	-7.8	-2.5
70	-8.8	-3.1
75	-9.8	-4.3
80	-9.8	-5.4
85	-9.8	-7.0
90	-9.8	-7.8
95	-9.8	-8.8

Table 4.2.

Dist.	Estimated Distance						
Step	Tiger	Face	GS	Line	Offcam	Mean	Std. Dev
10	10	10	10	9	9	9.6	0.49
15	15	14	15	16	14	14.8	0.75
20	19	19	20	22	20	20.0	1.10
25	24	25	25	29	25	25.6	1.74
30	30	30	28	33	31	30.4	1.62
35	33	37	32	35	36	34.6	1.85
40	39	42	37	40	40	39.6	1.62
45	43	47	42	43	44	43.8	1.72
50	50	49	50	50	48	49.4	0.80
55	55	54	55	56	53	54.6	1.02
60	60	59	60	63	59	60.2	1.47
65	65	63	66	67	65	65.2	1.33
70	68	68	71	71	69	69.4	1.36
75	73	74	76	76	74	74.6	1.20
80	78	79	82	81	80	80.0	1.41
85	83	84	86	86	85	84.8	1.17
90	89	88	90	94	92	90.6	2.15

Table 4.3. Results at room illumination

Distance Vs Sigma

Dist.	Estimated Distance						
Step	Tgr	Face	Tbr	Edge	GS	Mean	Std. Dev
5	5	3	5	8	3	4.8	1.83
10	8	9	9	12	11	9.8	1.47
15	13	15	14	18	15	15.0	1.67
20	19	18	20	22	19	19.6	1.36
25	24	25	25	25	26	25.0	0.63
30	29	30	29	29	32	29.8	1.17
35	34	34	34	35	35	34.4	0.49
40	40	40	42	38	39	39.8	1.33
45	45	43	45	48	43	44.8	1.83
50	49	47	55	52	49	50.4	2.80
55	54	55	54	60	54	55.4	2.33
60	58	60	60	61	59	59.6	1.02
65	63	65	68	59	65	64.0	2.97
70	69	68	72	65	71	69.0	2.45
75	74	75	74	70	75	73.6	1.85
80	79	80	82	76	80	79.4	1.96
85	83	87	93	80	85	85.6	4.26
90	88	95	91	84	89	89.4	3.61

Table 4.4. Results at 200 lux

Dist.	Estimated Distance											
Step	c1	c2	sb	fa	ft	gl	gs	mk	mn	tg	Mean	Std. Dev
10	9	9	9	7	10	11	10	10	9	11	9.5	1.12
15	11	12	13	15	12	13	15	14	13	13	13.1	1.22
20	12	16	18	18	18	19	19	20	20	19	17.9	2.26
25	20	20	22	24	25	23	26	24	25	24	23.3	1.95
30	30	31	31	31	31	30	31	30	30	29	30.4	0.66
35	37	36	37	35	33	34	35	34	33	35	34.9	1.37
40	41	41	40	41	36	40	41	41	37	40	37.8	1.72
45	46	45	45	45	44	45	44	44	46	44	44.8	0.75
50	48	49	48	50	47	49	48	48	50	49	48.6	0.92
55	54	52	53	54	53	53	53	53	54	53	53.2	0.60
60	60	59	60	59	59	59	59	58	58	59	59.0	0.63
65	68	65	66	66	64	62	66	65	63	64	64.9	1.64
70	72	71	70	71	70	71	70	71	70	70	70.6	0.66
75	76	76	77	75	74	75	74	75	75	75	75.2	0.87
80	83	81	82	82	80	80	85	81	81	82	81.7	1.42
85	92	88	88	87	83	84	86	83	85	89	86.5	2.73
90	95	95	92	91	91	84	85	93	92	94	91.2	3.63
95	95	95	92	91	94	92	81	95	95	95	92.5	4.10

Table 4.5. Results at 400 lux

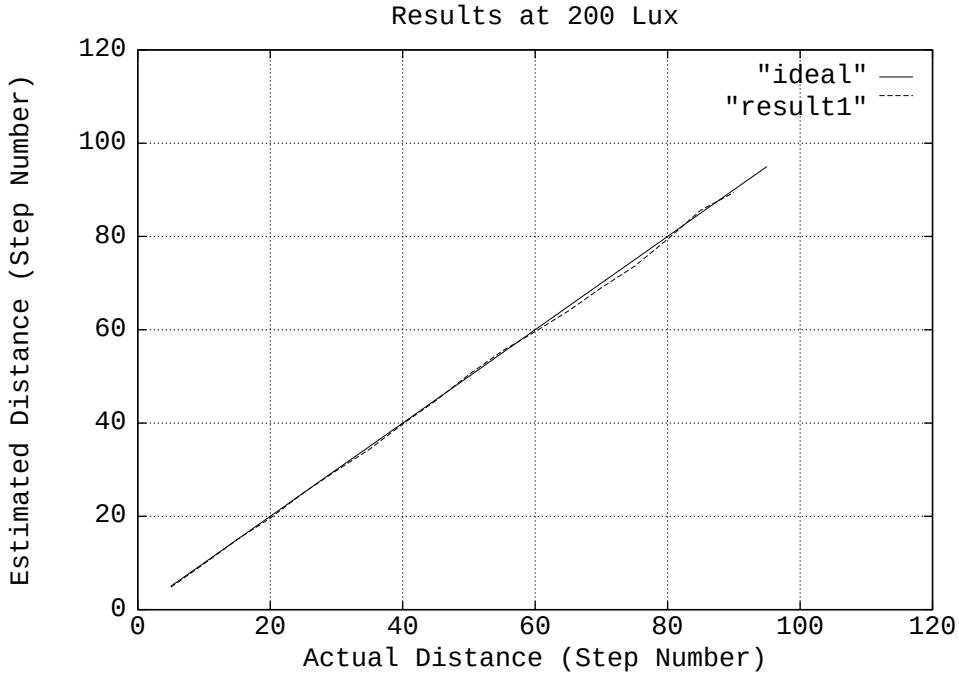


Figure 4-11: Mean results at 200 Lux illumination

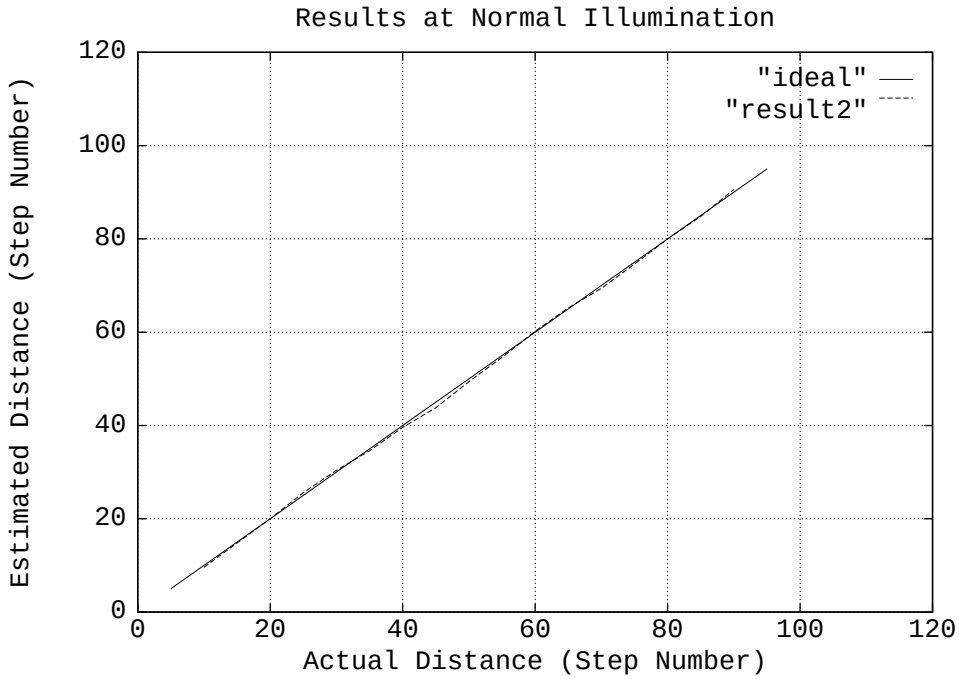


Figure 4-12: Mean results at room illumination

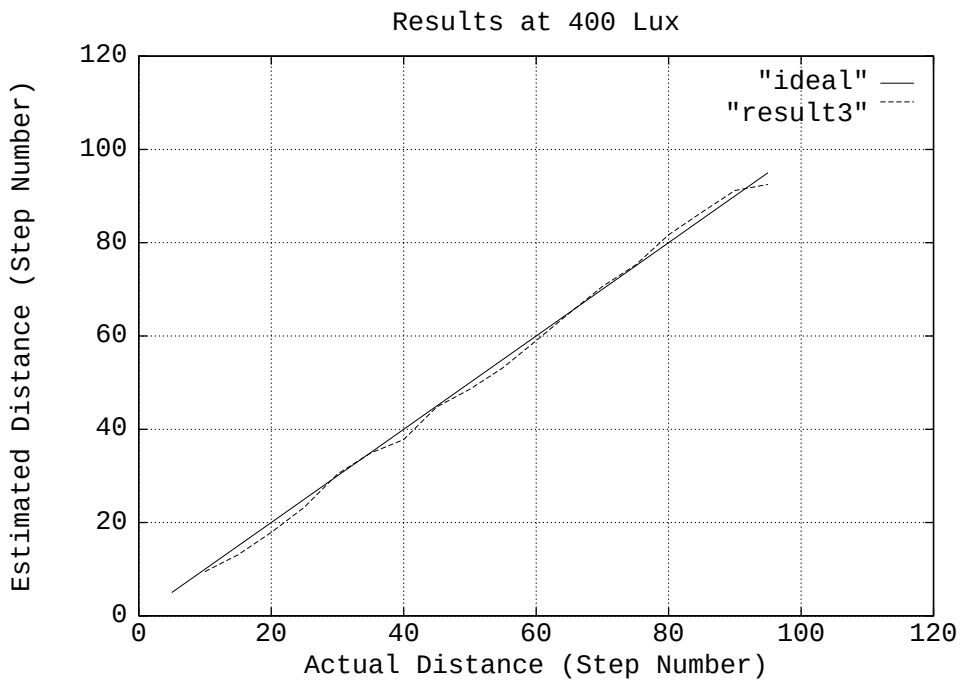


Figure 4-13: Mean results at 400 Lux illumination

as DFF.BST. A number of experiments were performed with a DFF method using the same objects used for the STM experiments. These experiments yielded an RMS error of 1.52 steps out of 97 steps. The RMS error of about 2.25 steps for STM compares well with this, considering the fact that only 2-3 blurred images are used.

The relationship between the reciprocal of the object distance $1/u$ versus the lens step number is almost linear (see Figure 4.3) and can be expressed as

$$1/u = ax + b \tag{4.52}$$

where x specifies lens position. For our camera, the lens position is specified in terms of a motor step number where each step corresponds to a displacement of about 0.03mm. The RMS errors mentioned above are for the lens position and it gives a good indication of the performance of the method for application

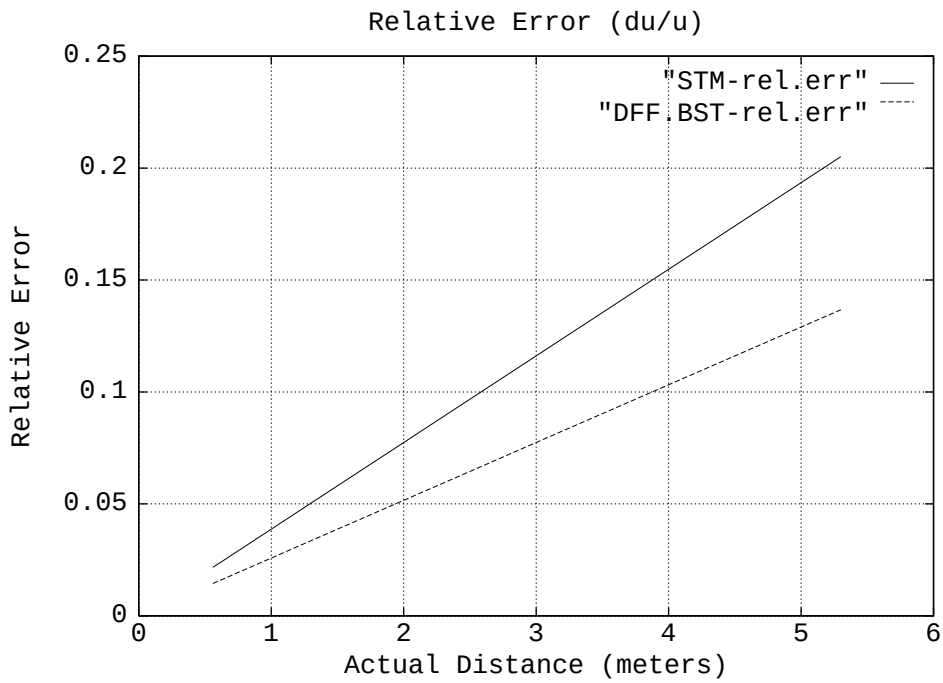


Figure 4-14: Relative error

in rapid autofocusing of cameras. In order to compute the error in terms of object distance, we have to consider the error differentials in equation (4.52):

$$|\delta(1/u)| = a|\delta x| \tag{4.53}$$

$$\Rightarrow |\frac{\delta u}{u}| = a|\delta x|u \tag{4.54}$$

$$\Rightarrow |\delta u| = a|\delta x|u^2 \tag{4.55}$$

From the above relations we see that the relative (percentage) error $|\frac{\delta u}{u}|$ in actual distance u increases linearly with distance, and the absolute error $|\delta u|$ in actual distance increases quadratically with distance. For our camera, using a Depth-from-Focus method [119] the constants were found to be $a = 0.0172$ and $b = -0.1143$.

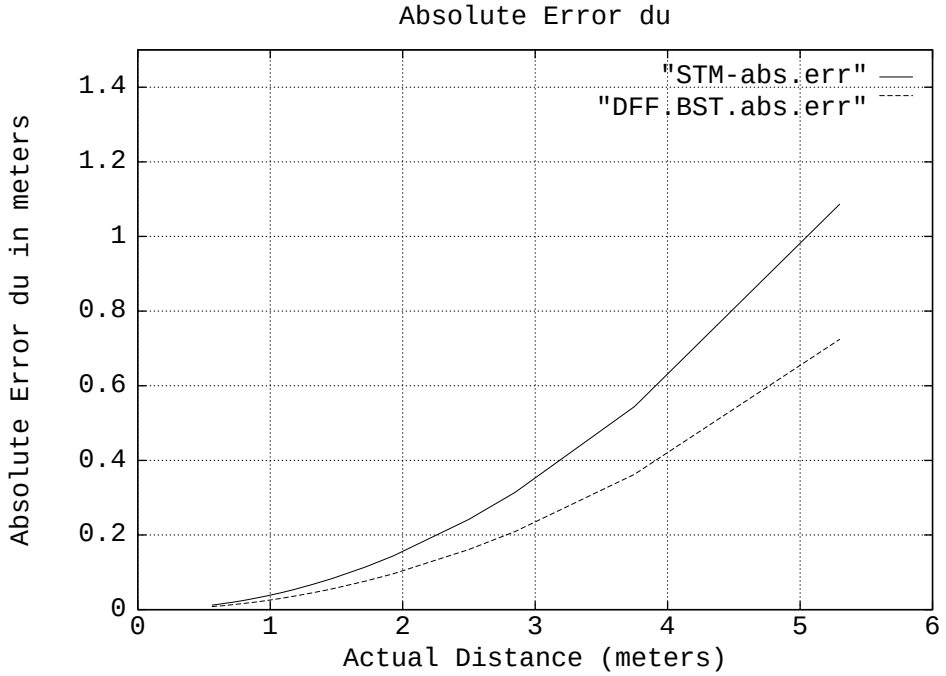


Figure 4-15: Absolute error

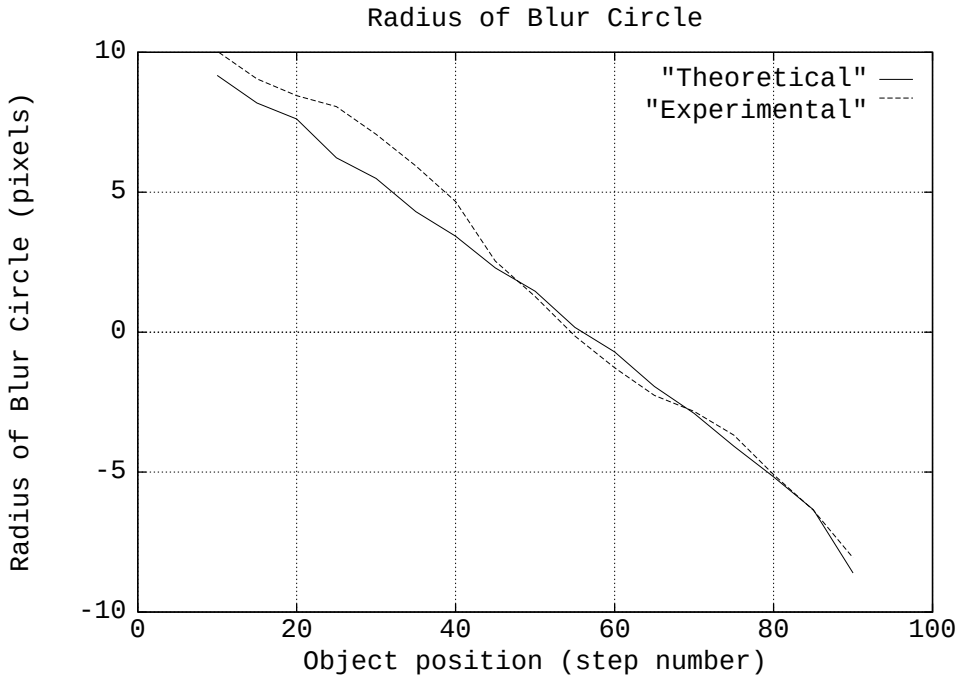


Figure 4-16: Radius of blur circle with lens at step 70

Setting $|\delta x|$ to be the RMS error of 1.52 steps for DFF and 2.25 steps for STM1 respectively, a plot of relative error $|\frac{\delta u}{u}|$ is shown in Figure 4.14 and a plot of the absolute error is shown in Figure 4.15. In Figure 4.14 we see that for STM the percentage error in distance at 0.6 meter is about 2.3% and increases linearly to about 20% at 5 meter distance. This compares well with the error obtained by the DFF approach of about 1.6% at 0.6 meter and increasing linearly to about 12.5% at 5 meter distance. Figure 4.15 shows that for STM, absolute error increases quadratically from 1.3 cms at 0.6 meter to about 1.0 meter at 5 meters distance. The corresponding numbers for the DFF method are 1 cm at 0.6 meter and about 0.6 meter at 5 meters distance.

A comparison between the actual radius of blur circle R' obtained from an experiment (using equation (2.23)) and that predicted by the geometric optics model (equation (2.9)) is shown in Figure 4.16. It can be seen that the error in terms of the radius of blur circle is less than 1 pixel for most distances.

4.5.4 Error in Stereo

Considering a simple stereo configuration, with the optical axes of the two cameras in parallel, the disparity l can be expressed as

$$l = \frac{bf}{u} \tag{4.56}$$

where b is the distance between the two optical axes (the baseline width), f is the focal length of the two cameras and u is the object distance. Considering error differentials as before we get

$$\delta l = -\frac{bf}{u^2}\delta u \tag{4.57}$$

The relative error can then be obtained as

$$\frac{\delta u}{u} = -\frac{\delta l.u}{bf} \tag{4.58}$$

As in the case of DFF or DFD, the relative error in stereo is linearly increasing with object distance. A typical value for δl is about 1 pixel (0.013 mm), f is about 35 mm and b is about 10 times f. Using these values at a distance u of 0.6 meter, in equation (4.58) we obtain a relative error of about 0.064 percent. The corresponding figure for DFF is about 1.6 percent. Thus, for this typical configuration stereo is about 25 times more accurate than focus.

This analysis suggests that by combining stereo and focus, the accuracy of DFF can be improved.

4.5.5 STM2

The procedure for calibration and experiments for STM2 is similar to STM1. The calibration results are shown in Tables 4.6 and 4.7. The average sigma values are plotted in Figure 4.17. The experiment is first tried with two pictures g_1 and g_2 taken at lens position fixed at 0 but F-numbers 4 and 8 respectively. Fixing the lens position at step 0 assures that the focused image is always behind the image detector (because objects at infinity are focused at step 0 and all other objects come to focus at higher step numbers). Therefore a unique solution is obtained for σ_2. If the estimated distance is greater than step 60, then the object is assumed to be too close to the camera and two more pictures are taken at step 60. These two are then used in estimating the distance.

Experiments on STM2 were conducted on four different objects at room illumination (about 200 - 300 Lux) and ten different objects at 400 Lux illumination. These objects were the same ones used in STM1 experiments and all the images are available as a database. For each object the experiment was repeated by moving the object to different distances from step 10 to step 95 in steps of 5. Thus the total number of experiments is $18 \times 14 = 252$. The results are tabulated in Tables 4.8 and 4.9. The mean values of the results and the standard deviations are also shown in these tables. The mean values of the results are plotted in Figures 4.18 and 4.19. The overall RMS error is about 2.25 steps out of 97 steps. This accuracy is very similar to that of STM1.

4.5.6 3 - Dimensional Objects

In the previous section, planar objects were used so that a rigorous performance and error analysis could be done. Here we give the results of determining distance of some 3D objects. In figure 4.20, (a) and (b) are two images of a cone taken with lens positions 40 and 70. The cone is about 1.5 meters long with black and white stripes on it. The axis of the cone is placed roughly along the optical axis of the camera and the tip is about 0.7 meter from the camera. The images were divided into overlapping regions of 32×32 pixels and STM1 algorithm was used to get one depth estimate at every 4 pixel intervals. The resulting depth- map is shown in Figure 4.20 (c). The depth-map is roughly in agreement with the ground truth.

Figure 4.21 shows a 3-D object (Teddy Bear), which has a depth variation of about 10 cm. The result of running STM1 on it is shown in Figure 4.22.

Dist.	Sigma Values			
Step	tg	fa	gs	mean
10	2.2	2.3	2.1	2.2
15	2.5	2.6	2.5	2.53
20	2.8	2.9	2.8	2.83
25	3.3	3.4	3.3	3.33
30	3.8	3.8	3.7	3.77
35	4.2	4.2	4.2	4.2
40	4.8	4.7	4.5	4.67
45	5.1	4.9	5.0	5.0
50	5.5	5.5	5.2	5.4
55	5.9	5.7	5.4	5.67
60	6.2	6.1	5.7	6.0
65	6.5	6.4	5.7	6.2
70	6.7	6.6	6.3	6.53
75	7.0	7.0	6.2	6.73
80	7.3	7.2	6.7	7.07
85	7.6	7.1	6.6	7.10
90	8.2	7.1	6.4	7.23
95	8.4	7.9	6.2	7.5

Table 4.6

Sigma with lens step 0

Actual Dist.		Estimated Dist. (step)					
Step	meters	Tgr	Face	GS	Edge	mean	std. dev.
10	3.750	10	11	10	13	11.00	1.22
15	2.850	14	16	14	17	15.25	1.30
20	2.500	19	20	19	21	19.75	0.83
25	1.930	24	25	24	24	24.25	0.43
30	1.720	30	30	29	31	30.00	0.71
35	1.465	35	35	35	37	35.50	0.87
40	1.320	41	40	38	40	39.75	1.09
45	1.170	46	43	45	47	45.25	1.48
50	1.080	51	51	47	50	49.75	1.64
55	0.965	55	58	51	63	56.75	4.38
60	0.900	58	63	57	65	60.75	3.34
65	0.822	65	63	65	70	65.75	2.59
70	0.770	69	69	70	75	70.75	2.49
75	0.715	75	73	74	80	75.50	2.69
80	0.670	81	79	80	84	81.00	1.87
85	0.628	85	84	85	88	85.50	1.50
90	0.595	90	89	90	94	90.75	1.92
95	0.560	95	94	94	96	94.75	0.83

Table 4.8. Results at normal illumination

Dist.	Sigma Values			
Step	tg	fa	gs	mean
10	3.6	3.7	3.5	3.6
15	3.3	3.4	3.3	3.33
20	3.0	3.1	3.0	3.03
25	2.6	2.6	2.6	2.6
30	2.2	2.2	2.2	2.2
35	1.8	1.8	1.8	1.8
40	1.4	1.4	1.4	1.4
45	1.1	1.3	1.0	1.13
50	1.0	1.0	0.8	0.93
55	1.3	1.5	1.0	1.27
60	1.5	1.8	1.4	1.57
65	1.9	1.8	1.9	1.87
70	2.2	2.2	2.3	2.23
75	2.7	2.6	2.8	2.7
80	3.3	3.1	3.2	3.2
85	3.6	3.5	3.6	3.57
90	4.1	4.0	4.1	4.07
95	4.5	4.4	4.4	4.43

Table 4.7.

Sigma with lens step 60

Actual Dist.		Estimated Distance (step)											
Step	meters	c1	c2	fa	ft	gl	gs	mk	mn	sb	tg	mean	std. dev.
10	3.750	10	10	11	11	11	10	10	10	10	10	10.3	0.46
15	2.850	17	11	15	17	15	14	14	13	14	14	14.4	1.68
20	2.500	17	17	20	20	20	19	19	20	19	19	19.0	1.09
25	1.930	23	23	25	25	25	25	24	23	25	24	24.2	0.87
30	1.720	28	27	30	30	29	30	30	28	31	29	29.2	1.17
35	1.465	35	32	36	35	35	37	32	33	36	36	34.7	1.68
40	1.320	51	38	42	42	40	40	38	39	40	39	40.9	3.62
45	1.170	46	44	46	45	45	45	43	45	45	44	44.8	0.87
50	1.080	51	50	50	46	50	44	49	51	50	50	49.1	2.16
55	0.965	55	54	55	54	55	55	55	56	56	54	54.9	0.70
60	0.900	63	59	60	60	62	60	60	56	58	59	59.7	1.85
65	0.822	64	68	65	68	63	56	63	62	64	62	63.5	3.23
70	0.770	64	67	70	72	71	64	70	72	67	70	68.7	2.86
75	0.715	74	73	75	75	75	72	73	74	73	75	73.9	1.04
80	0.670	78	78	80	82	95	80	78	80	80	81	81.2	4.78
85	0.628	83	82	85	85	84	84	84	84	84	84	83.9	0.83
90	0.595	89	88	90	92	92	92	88	89	89	90	89.9	1.51
95	0.560	95	95	88	95	95	95	95	95	95	95	94.3	2.10

Table 4.9. Results at 400 Lux

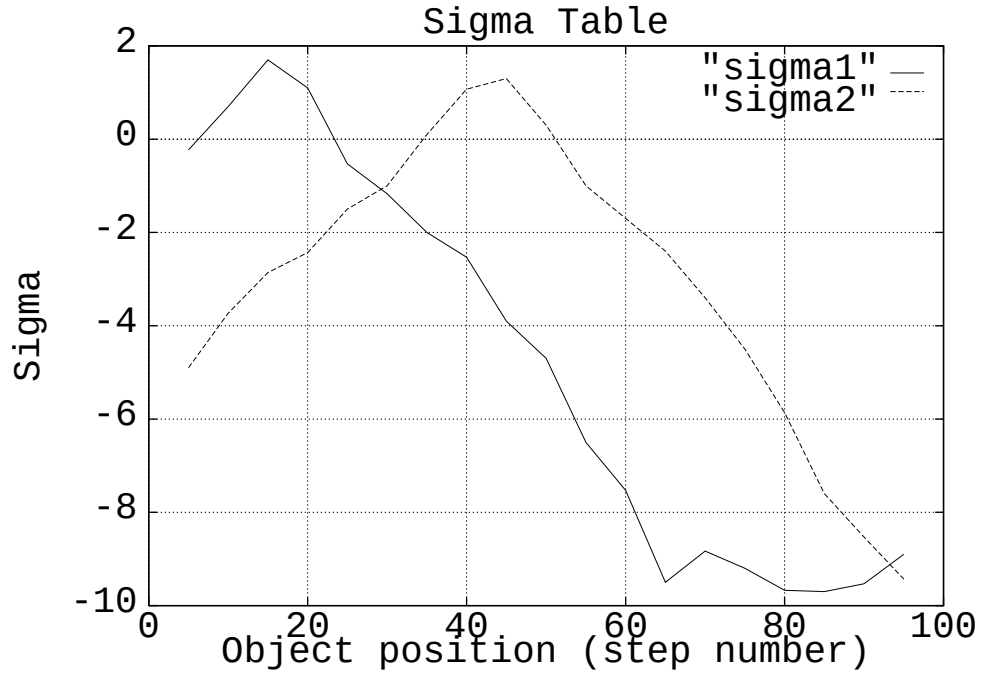


Figure 4-17: Calibration plots for STM2

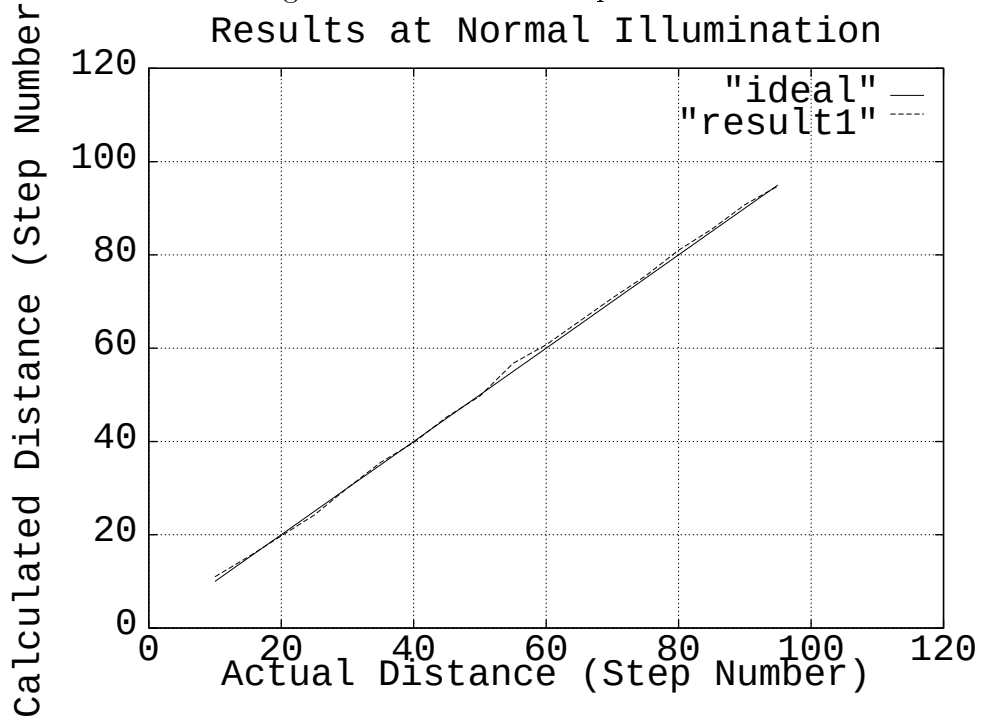


Figure 4-18: STM2 results at normal illumination

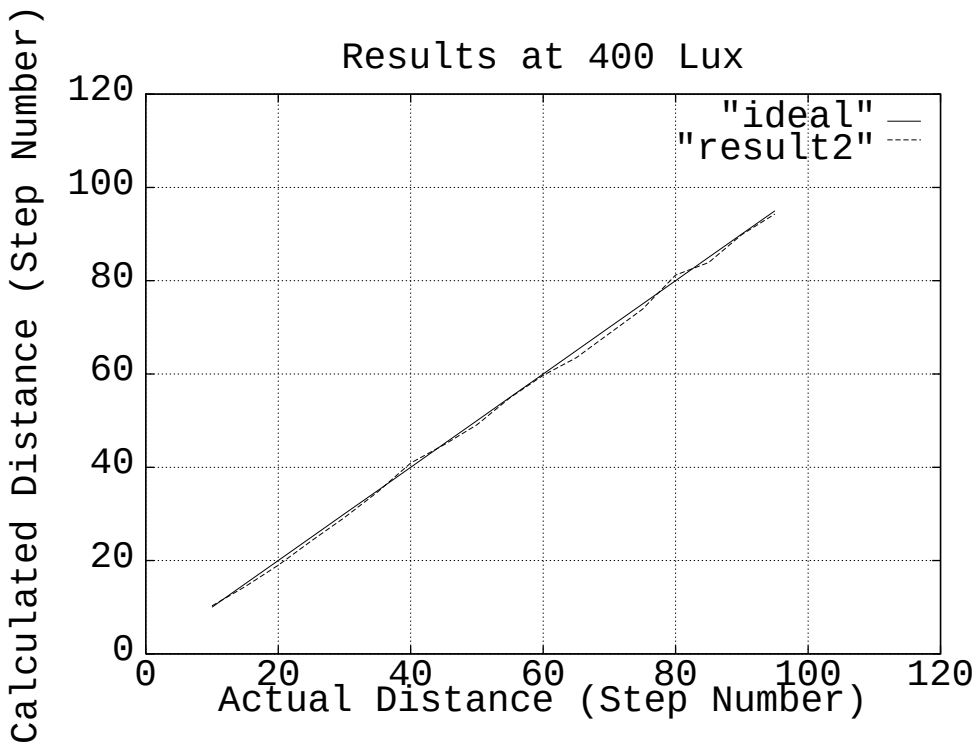
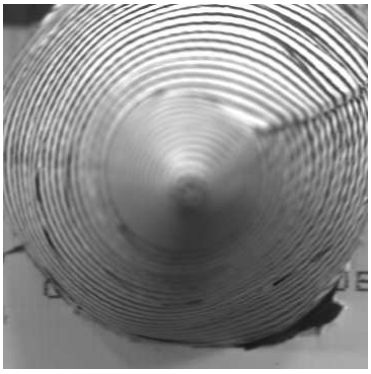


Figure 4-19: STM2 results at 400 Lux illumination

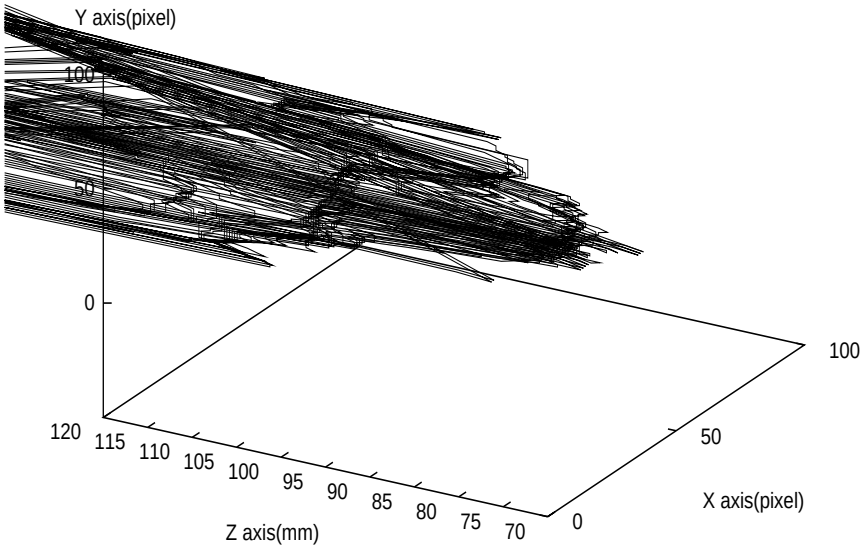
The face of the Teddy Bear was the region chosen for processing. We get an approximate depth estimate in such cases where the depth discontinuities are not too large.

In the case of 3-D objects, blurred images cannot be modeled as the result of convolving the focused image with the PSF of the camera. Therefore the distance estimated by STM will be in error. The error depends on the shape and appearance of objects. For objects with small depth variations, STM gives an estimate of "average" distance of the objects in the image window being processed.



(a) Step 40

(b) Step 70



(c) Depth Map for Cone.

Figure 4-20: Results for a cone object

Figure 4-21: Teddy bear object

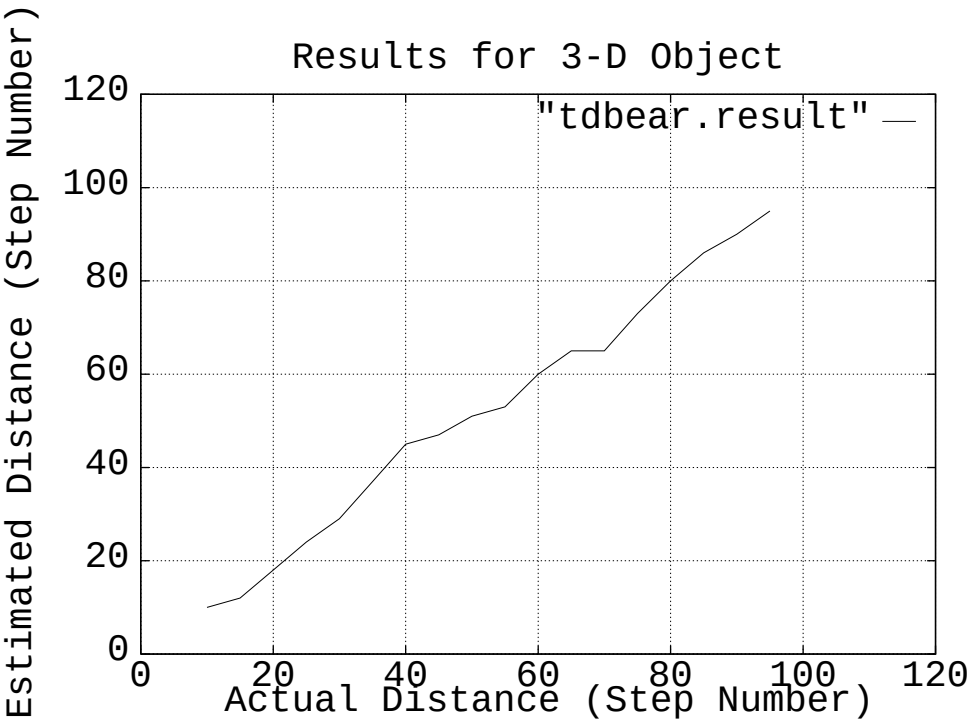


Figure 4-22: Results on teddy bear

4.6 Conclusions

In this chapter we have presented the theory and implementation of a new DFD method named STM. It has been successfully demonstrated on an actual camera system built by us. Experimental results indicate that STM is useful for passive ranging and rapid autofocusing. The ranging accuracy of STM is high for nearby objects and decreases with increasing distance. The ranging accuracy of this method can be improved somewhat by using a DFF method which searches for the best focused position in a small interval near the distance estimated by STM.

If we want to find the distance of only one object, then one could use binary or Fibonacci search in a DFF method. The order of complexity is about the same for both binary and Fibonacci search. However if one wants to obtain a coarse depth-map of a scene (e.g. the cone object in the previous section), DFF requires a large uniformly spaced sequence of images whereas STM needs only two to three images. In the case of search methods there is no closed form solution for the distance of objects and that is what distinguishes Depth-from-Focus and Depth-from-Defocus.

The distance of "plain" objects such as white walls which do not exhibit reflectance variation under uniform illumination cannot be determined by STM. However a random illumination pattern can be projected onto such objects to make them "textured". STM can then be used.

The wavelength of light λ can be considered as another camera parameter because focal length changes with wavelength [14]. STM can be implemented by taking two pictures using two different colors. Color filters may be placed

in front of the lens for the purpose.

Most existing camera systems (including our camera) are designed to maximize the depth-of-field since the goal is to obtain a "good" image of the scene for viewing by humans. However this minimizes the accuracy when ranging is concerned, since maximizing depth of field reduces the difference in blur between objects at different distances. Therefore, STM can be made much more accurate by designing cameras with small depth of field for the purpose of ranging.

Chapter 5

Continuous Focusing of Moving Objects

5.1 Introduction

In the previous chapter we described a Depth-from-Defocus algorithm, known as the S-Transform Method or STM, for determining distances of objects. STM is a spatial domain method and uses only two images taken with different camera parameters to estimate the distance of an object. Two variations of STM were described, one where the lens position and focal length are changed and another where the diameter of camera aperture is changed.

In this chapter, we address the problem of continuously focusing the camera on a moving object by changing the lens position. Such a situation may arise in an autofocusing video camera and in robotic vision, where the objects in the scene may be slowly moving. Wei and Subbarao [125] have recently proposed a Fourier domain approach for focusing on moving objects. They reported a focusing accuracy of about 4.3%. Here, we describe a spatial domain method based on STM. The method is named *Continuous STM* or CSTM.

The focusing accuracy of CSTM is about 2.3 - 3%. Using CSTM it is also possible to obtain denser depth-maps of the scene, than what can be obtained by using Fourier domain methods, such as [125].

Two variations of CSTM - CSTM1 and CSTM2 - are described. CSTM1 is a straightforward extension of the STM described in Chapter 4. It involves calibration of the camera for a number (about 6 in our implementation) of discrete lens positions. In CSTM2 the camera is calibrated just once corresponding to one lens position. The calibration data corresponding to other positions are obtained by transforming the data of the one lens position for which the camera is calibrated. Experiments show that the difference in performance between CSTM1 and CSTM2 is marginal. We describe experiments on the SPARCS system and also propose a fast and inexpensive camera system for implementing CSTM.

5.2 Distance of Moving Objects (CSTM)

Let $g_1(x, y)$ and $g_2(x, y)$ be two images of the object recorded for two different camera parameter settings $\mathbf{e}_1$ and $\mathbf{e}_2$ where

$$\mathbf{e_1} = (s_1, f_1, D_1) \quad \text{and} \quad \mathbf{e_2} = (s_2, f_2, D_2). \tag{5.1}$$

The images g_1 and g_2 are normalized with respect to magnification, brightness, and other factors such as sensor response and vignetting as necessary [110].

In the previous chapter, we have seen that the spread parameter σ_2 of the PSF of the image $g_2(x, y)$ can be found from

$$\sigma_2^2(\alpha^2 - 1) + 2\alpha\beta\sigma_2 + \beta^2 = G' \tag{5.2}$$

$$\text{where } G'^2 = 16\ \frac{\int\int (g_1 - g_2)^2 dxdy}{\int\int (\nabla^2 g)^2\ dxdy} \tag{5.3}$$

The above equation can be solved as a quadratic in σ_2.

In our experiments, $D_1 = D_2$ and therefore $\alpha = 1.0$. In this case the above quadratic equation in σ_2 reduces to a linear equation. Therefore we get the unique solution:

$$\sigma_2 = \frac{G' - \beta^2}{2\beta} \tag{5.4}$$

Ideally it should be possible to compute the value of σ_2 at one pixel (x, y) in the image and obtain an estimate of the distance. But because of noise and digitization, it is necessary to combine information from many pixels in an image region. σ_2 is computed at each pixel in a neighborhood of size about 48×48 pixels and a histogram of the values is obtained. The histogram is smoothed by a Parzen window and the mode of the resulting distribution is taken to be the best estimate of σ_2. Once σ_2 is determined the object distance u can be obtained using a look-up table or calculated from equation (4.28). The distance u can then be substituted into the lens formula to obtain v. Moving the lens such that $s = v$ in Figure 2.1 results in autofocusing the object. In our experiments, the lens position v for focusing the object was determined using a look-up table with σ_2 as the index to the table. The look-up table itself was created by a calibration process. This approach was found to be more accurate than using equation (4.28) and the lens formula.

In the case of STM, the two images g_1 and g_2 are recorded at lens positions S_1 and $S_1 + \Delta S$. However, in the case of a moving object, the two images g_1 and g_2 will have to be recorded at the same instant. They cannot be recorded in sequence one after the other because the object would have moved during

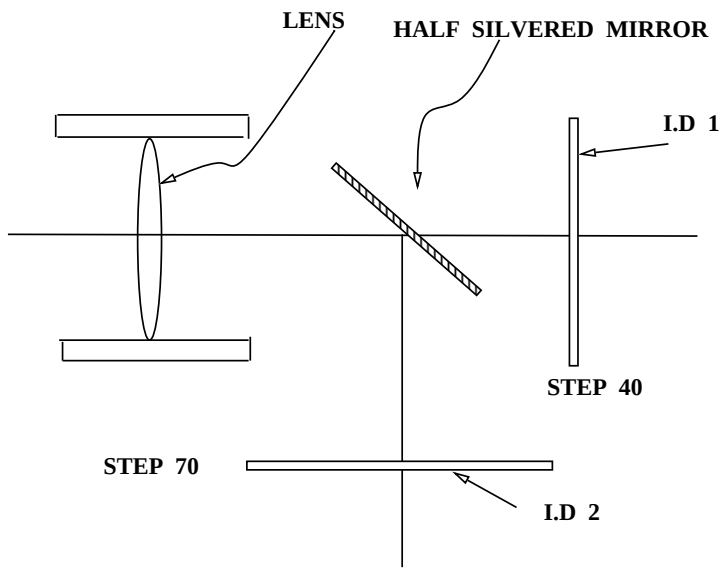


Figure 5-1: Beam splitter implementation

the time it takes to move the lens from one position to the other. Figure 5.1 shows a scheme for recording g_1 and g_2 simultaneously using a beam splitter and two image detectors ID1 and ID2. The effective distance of ID1 and ID2 from the lens are set to be S_1 and $S_2 = S_1 + \Delta S$.

Let ID1 be the image detector in Figure 5.1 on which we want to continuously record the focused image of a moving object. When the lens is moved along the optical axis to focus the moving object, both S_1 and S_2 change but $S_2 - S_1 = \Delta S$ remains the same. CSTM1 is based on this camera architecture. CSTM1 is a direct extension of STM applied to moving objects using a camera system which is similar to the one shown in Figure 5.1. In general, look-up tables have to be used for better accuracy in focusing. In the case of CSTM1, look-up tables have to be obtained for a number of discrete val-

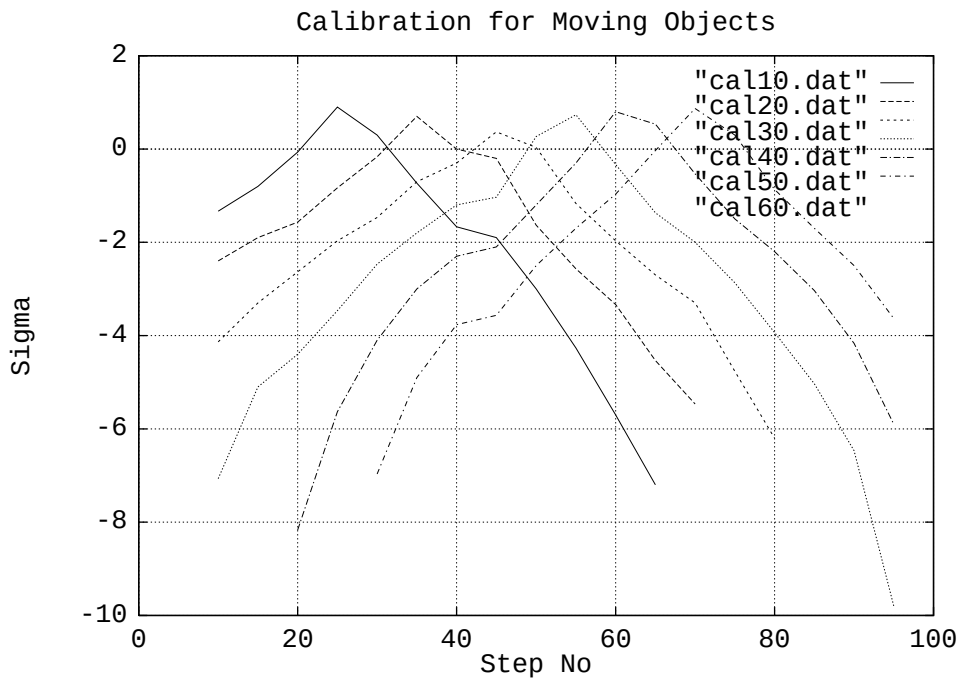


Figure 5-2: Calibration for several lens positions

ues of S_1. The data at other values of S_1 are obtained through interpolation. Therefore, CSTM1 involves camera calibration for a number of lens positions.

CSTM2 is based on the following observation. It can be shown that if we take two images g_1 and g_2 keeping the aperture diameter constant ($D_1 = D_2$), then $\alpha = 1.0$ and $\beta = c_1 - c_2$. So equation (4.31) becomes

$$\sigma_1 = \sigma_2 + c_1 - c_2. \tag{5.5}$$

Therefore, if we know $\sigma = \sigma_2$ for a lens position S_2, then we can compute $\sigma = \sigma_1$ for any other lens position S_1, by adding a known constant $c_1 - c_2$. Consequently, if a look-up table is available to find the lens focused position using σ_2, then the same look-up table can be transformed to obtain the lens focused position using σ_1. The validity of equation (5.5) can be verified ex-

perimentally. Figure 5.2 shows calibration data for several lens positions. The X-axis denotes object distances specified in lens step number and Y-axis is the blur parameter σ (see next section for more details). It is found that the data for different lens positions are roughly shifted versions of the same curve. Equation (5.5) however is not exact, but only a good approximation for actual camera systems because our PSF model is not exact.

A very similar situation arises whenever the lens is removed and put back. The new image detector position will be slightly shifted with respect to the previous one. However, since σ_2 is always estimated relative to step 70 of the image detector position, the estimation of σ_2 will not be affected by the assembly error. Also, since the calibration plots in Figure 5.2 appear almost as shifted versions of each other, the focusing results also will be unaffected by the assembly error. However, in obtaining the distance of the object from σ_2, a look-up table is used, which is a table of σ_2 versus object distance. This table will be affected by the assembly error. Hence the relative displacement of the image detector should be determined through calibration and the look-up table should be corrected.

5.3 Implementation

The overall operation of SPARCS for finding distance and autofocusing of a moving object is summarized as a flow-chart in Figure 5.3. The stepwise operation is also explained briefly with comments below. In the experiments, initially, the zoom setting of the lens was set to be 35 mm focal length and

the F-number was set to be 4. The camera gain was set to +6db.

The lens is first moved to S_1 and a first image $g_1(x, y)$ is obtained. Optionally we can specify the number of image frames (typically 4) to be recorded which are then averaged to reduce noise. Such frame averaging is particularly needed under low illuminations, and in the presence of flickering illumination such as fluorescent lamps. This was clearly evident from a number of tests on SPARCS.

The lens is then moved to S_2 and a second image $g_2(x, y)$ is recorded. Again several frames may be recorded and averaged. The object to be ranged/focused can be selected by specifying a region in the image. The default region is the center of the image. The size of the region is also an option and the default size is 72×72. The two images are then normalized with respect to brightness.

An estimate of σ_2 is then obtained at every pixel using equation (5.4). Due to border effects of smoothing filter and integration, the estimates of σ_2 is limited to the interior 48×48 region of the original 72×72 images. A histogram of the estimated σ_2 is computed. The bin size of the histogram was 0.1 (the expected range of σ_2 was from about -10.0 to +10.0 pixels). The histogram was smoothed using a Parzen window of size 5 bins. The mode of the histogram was taken to be the best estimate of σ_2. This value is used to estimate the distance of the object. In autofocusing application, from σ_2, the lens step number which will bring the object to focus is determined. The lens is then moved to this step number to accomplish autofocusing.

In obtaining the object distance or lens step number for focusing from

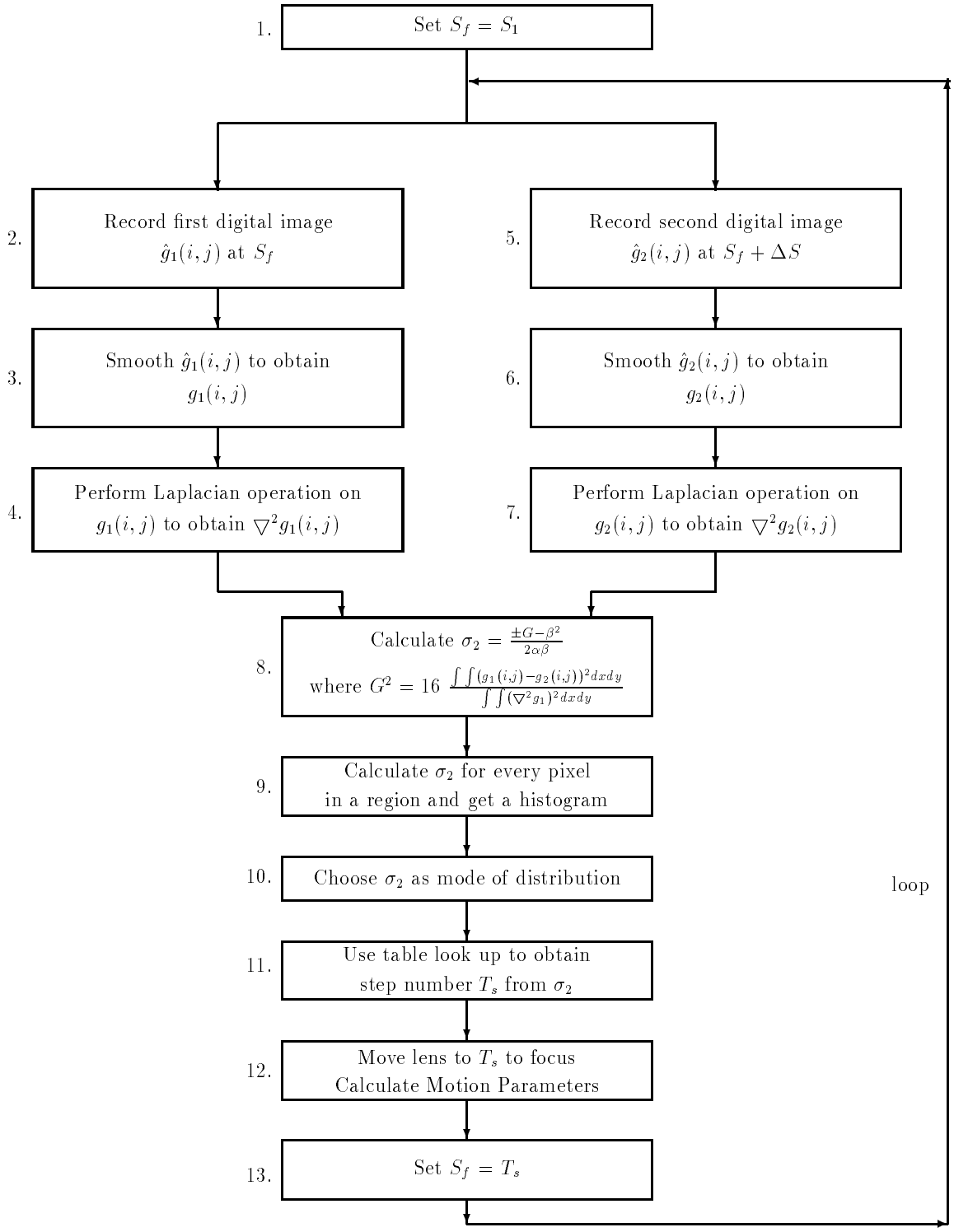


Figure 5-3: Flow chart of CSTM

the computed value of σ_2, a look-up table is used. The look-up table itself is obtained through calibration as described in Chapter 4.

Suppose the result of the first trial is S_f (step number n). After the first iteration, the lens is moved to S_f (step number n) to focus on the object. Since the object would have moved in the mean time, we again take two images at lens positions $S_1 = S_f$ and $S_2 = S_f + \Delta S$ (steps n and $n + 30$ in our implementation). In a camera system as shown in Figure 5.1, it is not necessary to move the lens. Both the images can be obtained simultaneously. Hence the time between two iterations can be very small. The entire procedure is repeated with the two new images. Every time, a new focus position S_f is calculated using the appropriate calibration table (CSTM1) or by using the same calibration table, but shifted versions of it (CSTM2). The amount by which the calibration table has to be shifted depends on the previous focusing position. Everytime a new result is obtained, two new images are taken and the entire procedure is repeated in a loop. This ensures that the object is always in focus (on ID1) whether the object is stationary or moving.

5.3.1 Experiments

We used five different planar objects in our experiments (Figure 5.4). A center region of 72 x 72 pixels, that is usually used for computation is highlighted, in the Tiger image. After filtering, the useful region for computation of σ will be only 48 x 48 pixels. We have tried our experiments with even smaller regions upto 32 x 32 pixels and obtained satisfactory results, as long as there is some contrast in that region.

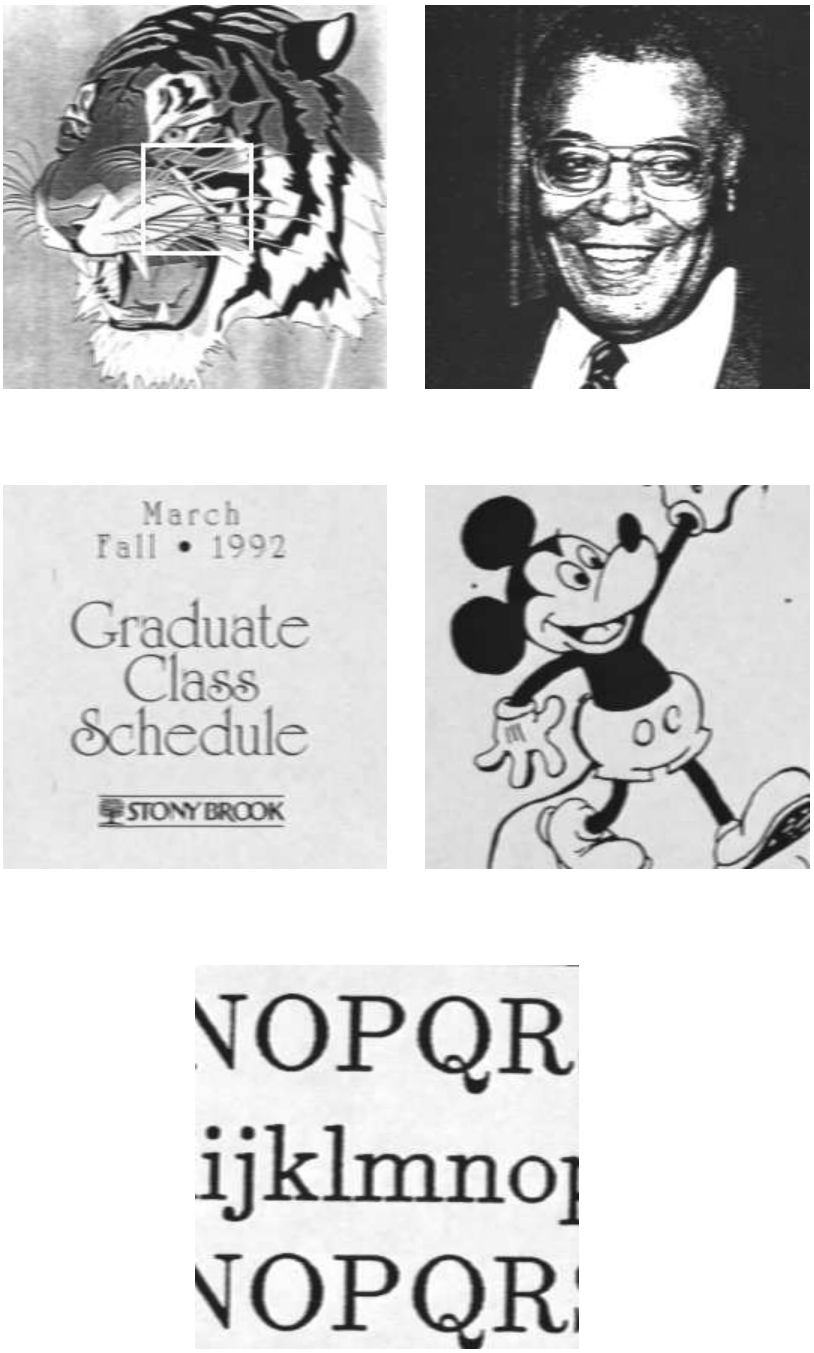


Figure 5-4: Objects used in CSTM experiments

For CSTM1 we calibrated the system at 6 different lens positions, namely steps 40, 50, 60, 70, 80 and 90. The calibration results are plotted in Figure 5.2, where the X-axis is the object distance in step number and Y-axis is the value of sigma. In the Figure, the plot "cal10.dat", was obtained using images recorded at lens steps of 10 and 40, the plot "cal20.dat" was obtained corresponding to lens steps of 20 and 50 and so on. The plots can be seen to be more or less shifted versions of each other. We assumed that the calibration characteristics do not change much in a 10 step interval and hence the choice of these 6 lens positions. Instead of calibrating at all the 96 lens positions we just calibrated at these 6 almost uniformly spaced lens positions and for other lens positions we obtained the calibration data by merely shifting the nearest available calibration data.

The essence of CSTM is that we can take two images from any arbitrary lens positions S_f and $S_f + \Delta S$ to compute distance. To demonstrate this fact, we placed the objects at a known distance (say step 10) and the program was run by specifying a different starting lens position (10, 20, 30, 40, 50 or 60), everytime. It means that the first time the two images were taken at steps 10 and 10+30 = 40 and the program was run without changing the object position. The second time the object position was still the same as before but the program was run with starting lens steps of $S_1 = 20$ and $S_2 = 20+30 = 50$. This procedure was repeated for other lens positions of $S_1 = 30$, 40, 50 and 60.

Suppose we placed an object at step 10 and specified step 60 as the starting lens position. It is equivalent to the case when the object actually moved from

step 60 to 10 (about 2 meters), a relatively high velocity, between two runs of the program. With one single object and for one single object position we did 6 experiments, corresponding to the 6 different starting lens positions mentioned above. The experiments were then repeated for 18 different object distances. Thus, with one single object we performed 6 * 18 = 108 experiments. For five different objects, the total number of experiments becomes 108 * 5 = 540.

During calibration, it was found that the estimated value of σ_2 (in Figure 5.2) was unreliable when both the images g_1 and g_2 (on which the estimation was based) were highly blurred. For this reason, calibration was limited to the case when the lesser blurred image, say g_1, was recorded at a position that was at most 25 lens steps away (corresponding to a radius of blur circle of about 7 pixels) from the focused lens position and the higher blurred image, say g_2, was recorded at a lens position that was at most 25+30 = 55 lens steps away (corresponding to a radius of blur circle of about 14 pixels) from the focused lens position. It is for this reason that the plots in Figure 5.2 do not cover the entire range (0 to 96) of lens positions. For example, "cal10.dat" in Figure 5.2 covers the range from step 10 to step 65. The range 66 to 96 is not covered because in that case the two images would be highly blurred. The range 0 to 10 steps is not covered by any plot because as shown in Table 5.1, lens positions 0 and 5 correspond to placing objects at distances of 9.03 meters and 5.30 meters from the camera. Due to space restrictions in our laboratory, we were not able to place objects farther than 5 meters, and therefore the calibration data for these two points were not obtained.

If both images g_1 and g_2 are highly blurred, then reliable focusing can be

Step	0	5	10	15	20	25	30	35	40	45
FL	36.235	36.086	35.938	35.72	35.646	35.502	35.359	35.217	35.076	34.937
SI	36.180	36.132	36.083	36.035	35.988	35.941	35.894	35.848	35.802	35.757
D0(mm)	9034.0	5300.0	3750.0	2850.0	2500.0	1930.0	1720.0	1465.0	1320.0	1170.0
Step	50	55	60	65	70	75	80	85	90	95
FL	34.738	34.661	34.524	34.389	34.255	34.121	33.989	33.859	33.728	33.380
SI	35.712	35.667	35.622	35.578	35.534	35.491	35.448	35.406	35.363	35.250
D0(mm)	1080.0	965.0	900.0	822.0	770.0	715.0	670.0	628.0	595.0	560.0

Table 5.1. Lens Data

achieved by iterating CSTM twice. The first iteration gives a rough estimation of the focused lens position. The lens is moved to this position and CSTM is applied again. In this case, the images will not be highly blurred as in the first iteration. Therefore good focusing will be achieved. In the experiments, the object was not moved between two iterations. However, modest movement (less than 20 lens steps) will not significantly alter the performance of CSTM.

The mean values (of five objects) of the experimental results for CSTM1 are shown in Table 5.2. In the table, six different sets of results corresponding to six different sets of lens positions (S1,S2) are presented. Also the results of both the first and second iterations are presented. It can be seen from the table that most of the time we get good results in the first iteration itself. But if we look at the row corresponding to object distance step 8, and S1,S2 = 60,90, it can be seen from the table that the first iteration gives a result

Obj. Dist.	Estimated Distance in Step No											
Step No.	S1,S2 = 10,40		S1,S2 = 20,50		S1,S2 = 30,60		S1,S2 = 40,70		S1,S2 = 50,80		S1,S2 = 60,90	
	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2
8	7.8	7.8	7.8	7.8	7.8	7.8	9.0	7.8	17.8	7.8	24.8	7.4
13	12.2	12.2	12.0	12.2	12.0	12.2	12.8	12.2	19.0	12.0	24.4	12.2
17	16.6	16.4	16.8	17.4	16.6	17.4	15.6	17.4	19.4	16.8	24.0	16.6
23	24.0	22.2	22.2	22.2	21.8	22.2	24.0	22.2	23.8	22.2	26.0	21.8
29	29.0	28.8	30.0	28.2	28.2	28.2	28.0	28.2	28.0	28.2	29.4	28.2
34	34.2	33.8	34.8	33.8	33.8	33.8	33.2	33.8	33.6	33.8	33.6	33.8
40	40.6	42.2	40.0	40.0	46.4	40.2	40.0	42.2	39.0	38.0	39.4	41.0
45	43.0	43.0	42.2	40.8	48.2	41.6	41.0	43.0	41.6	41.0	42.8	40.8
51	49.4	48.6	50.0	48.6	50.0	48.6	49.0	48.6	48.6	48.6	49.8	48.6
56	54.8	55.0	55.2	55.0	56.0	55.2	55.0	55.2	55.0	55.0	55.2	55.2
62	60.8	59.6	61.8	59.6	61.8	59.6	62.2	59.6	62.4	59.6	59.6	59.6
68	67.2	67.2	67.8	67.2	67.2	67.2	67.8	67.2	67.0	67.2	67.2	67.2
73	67.8	73.0	72.6	73.0	72.8	73.0	73.0	73.0	72.8	73.0	73.0	73.0
79	69.2	77.0	74.4	77.0	78.8	77.0	78.4	77.0	79.0	77.0	77.0	77.0
85	70.8	84.0	75.4	84.0	84.6	84.0	83.8	84.0	85.0	84.0	84.0	84.0
90	73.8	89.8	77.4	89.8	85.4	89.8	89.8	89.8	89.8	89.8	89.8	89.8
96	74.4	94.4	78.4	94.4	86.4	94.4	94.6	94.4	95.0	94.4	94.4	94.4
102	77.0	100.4	81.4	100.4	87.2	100.4	98.0	100.4	100.6	100.4	100.4	100.4

Table 5.2. Results of CSTM1

of 24.8, which is not very close to the actual value. But the second iteration gives a very close value of 7.4 steps. As explained earlier, this is because the object distance of step 8 is far away from steps 60 and 90 at which the two images are taken.

Some of the results of CSTM1 are also plotted in Figure 5.5. The X-axis indicates the experiment number. Since there are five objects and six lens positions, the number of experiments for each distance is 30. The Y-axis indicates the estimated distance in step number. The plot "step 17" shows the results when the objects are placed at step 17, and images are obtained with different lens positions. The other two plots in Figure 5.5 show the results when the objects are placed at step 56 and step 85 respectively. Ideally these plots should have been straight lines parallel to the X-axis.

The first set of experiments included 440 trials of the case when the two

Obj. Dist.	Estimated Distance in Step No											
Step No.	S1,S2 = 10,40		S1,S2 = 20,50		S1,S2 = 30,60		S1,S2 = 40,70		S1,S2 = 50,80		S1,S2 = 60,90	
	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2	Iter 1	Iter 2
8	6.0	6.0	7.8	6.0	9.2	6.0	9.0	6.0	16.0	7.4	24.8	7.8
13	13.0	13.0	11.2	13.0	13.0	13.0	12.8	13.0	16.8	11.6	24.4	11.6
17	15.8	14.4	14.2	15.8	16.2	14.8	15.6	14.8	17.4	14.2	24.0	14.2
23	20.2	24.0	24.0	24.0	20.8	24.0	24.0	24.0	21.6	24.0	26.0	22.2
29	26.0	26.4	27.4	26.2	26.2	27.0	28.0	26.2	29.4	26.2	29.6	26.2
34	31.2	34.4	32.0	34.4	34.4	33.8	33.2	34.4	35.8	34.2	35.0	33.8
40	36.2	38.8	38.2	40.0	39.2	40.0	40.0	39.0	40.2	40.0	42.6	40.0
45	38.6	41.0	39.8	41.0	39.8	41.0	41.0	40.2	41.8	41.0	44.2	40.2
51	45.2	48.4	47.4	48.4	48.4	48.4	49.0	48.4	48.4	48.4	50.0	48.4
56	51.4	58.0	53.2	58.0	55.0	57.4	55.0	56.8	58.0	56.0	56.0	56.6
62	57.4	66.2	58.0	66.2	61.0	66.2	62.2	66.2	63.4	66.2	66.2	66.2
68	62.6	71.0	64.6	71.0	65.8	71.0	67.8	71.0	68.8	71.0	71.0	71.0
73	64.2	75.8	69.2	75.8	70.0	75.8	73.0	75.8	73.8	75.8	75.8	75.8
79	65.8	79.2	74.2	79.2	77.4	79.2	78.4	79.2	80.0	79.2	79.2	79.2
85	67.6	87.6	75.6	87.6	83.2	87.6	83.8	87.6	86.8	87.6	87.6	87.6
90	70.4	92.0	77.6	92.4	86.2	92.4	89.8	92.4	90.8	92.4	92.4	92.4
96	71.2	98.4	78.4	98.4	87.6	98.4	94.6	98.4	97.2	98.4	98.4	98.4
102	73.0	104.4	81.2	104.4	88.6	104.4	98.0	104.4	103.8	104.4	104.4	104.4

Table 5.3. Results of CSTM2

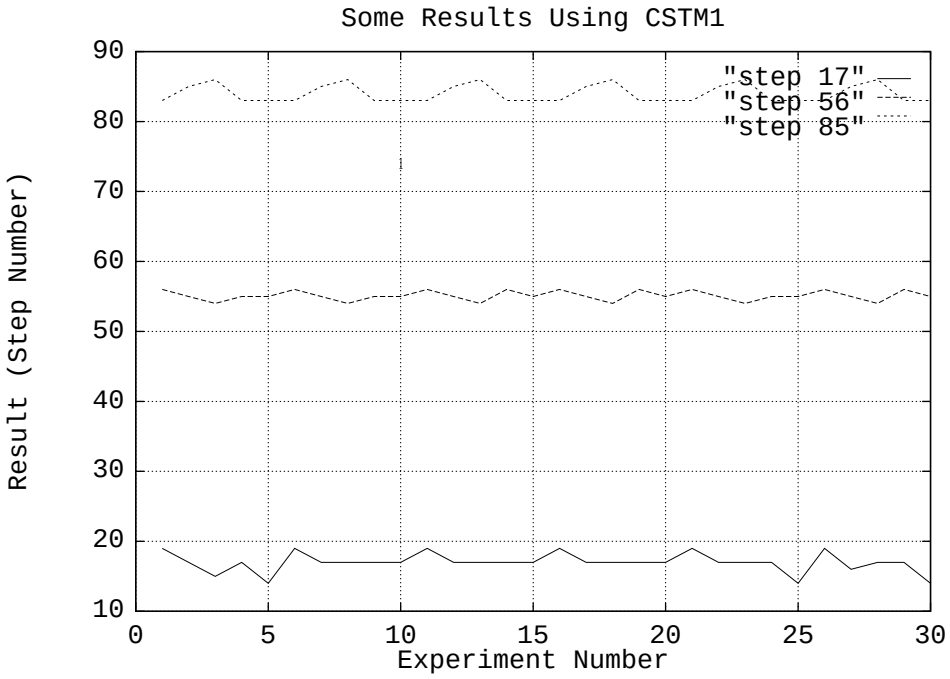


Figure 5-5: Some CSTM results

images were not highly blurred (corresponding to less than 25 steps of blur (7 pixel radius) for one image and less than 55 steps blur (14 pixel radius) for the second image). For these trials, the RMS error in focusing for one iteration of CSTM1 was 2.22 steps out of 97 steps, or about 2.3%. In terms of the radius of blur circle the error is about 0.417 pixel. The second set of experiments included the 440 trials of the first set and an additional 100 trials where both images were highly blurred (according to the criteria explained earlier). In order to perform trials for the highly blurred cases, when necessary, the calibration data in Figure 5.2 was extended through simple linear extrapolation of the plots. Two iterations of CSTM1 were performed for each of the 540 trials in the second set of experiments. The RMS error in focusing for these trials was 2.3 steps out of 97 steps or 2.4%. In terms of the radius of blur circle this error corresponds to about 0.432 pixel.

For experiments on CSTM2, only one set of calibration data corresponding to the plot "cal40.dat" in Figure 5.2 was used. This data set was shifted by appropriate amounts to obtain other required calibration data such as the plots labelled "cal10.dat","cal20.dat" etc. in the figure. Experiments similar to those described earlier for CSTM1 were repeated for CSTM2. The results are tabulated in Table 5.3. The first set of experiments included 440 trials with one iteration, for the case when the two images were not highly blurred. The RMS error in focusing for these experiments was 2.9 steps out of 97 steps or about 3.0%. The second set of experiments included 540 trials which included the 100 trials where the two images were highly blurred. CSTM2 was run for two iterations as before and the RMS error in focusing was 3.05 steps out of

97 steps or about 3.1%.

A focusing error of 3% (corresponding to a radius of blur circle of about 0.56 pixel) is not perceptible by humans. Therefore, the results of CSTM are quite satisfactory. However, further improvement can be obtained by using a Depth-from-Focus (DFF) method and searching only in a small interval near the estimated lens position.

5.4 Conclusions

In this chapter the DFD method based on STM has been extended to continuously focus on moving objects. It has been successfully demonstrated on the SPARCS camera system. Two variations of continuous focusing - CSTM1 and CSTM2- have been investigated. CSTM1 involves straight forward extension of the STM described in the previous chapter and involves extensive camera calibration. The focusing accuracy was 2.3% by calibrating the camera system at 6 different pairs of lens positions. In CSTM2, the camera is calibrated just once corresponding to one lens position. The calibration data corresponding to other positions are obtained by transforming the data obtained for the one single position. A theoretical justification for this has been provided. The focusing accuracy of CSTM2 was found to be about 3% in lens position. The marginal improvement in accuracy of CSTM1 was achieved at the cost of a more cumbersome calibration procedure.

A typical application for CSTM is in autofocusing of video cameras, where it is necessary to quickly focus on objects which keep changing their positions.

CSTM can also be used to continuously obtain a rough depth-map of a dynamic scene. The resolution of the depth-map can then be improved by using stereo vision techniques, if desired.

Chapter 6

Focused Image Recovery

6.1 Introduction

In the previous chapters we discussed methods for recovering the geometric information of a scene. In this chapter we investigate some methods for recovering the focused image of a scene, which is a photometric information. In machine vision, early processing tasks such as edge-detection, image segmentation, stereo matching, etc. are easier for focused images than for defocused images of three-dimensional (3-D) scenes. However, the image of a 3-D scene recorded by a camera is in general defocused due to limited depth-of-field of the camera. Autofocusing can be used to focus the camera onto a desired target object. But, in the resulting image, only the target object and those objects at the same distance as the target object will be focused. All other objects at distances other than that of the target object will be blurred. The objects will be blurred by different degrees depending on their distance from the camera. The amount of blur also depends on camera parameters such as

lens position with respect to the image detector, focal length of the lens, and diameter of the camera aperture. In this chapter, we address the problem of recovering the focused image of a scene from its defocused images.

Earlier we described STM, a spatial domain method of determining distance using defocus information. Also, Subbarao and Wei have proposed a Fourier domain method, called DFD1F, for estimating the distance of objects in a scene [114] using image defocus information. In these methods, two defocused images of the scene are recorded simultaneously with different camera parameter settings. The defocused images are then processed to obtain the distance of objects in the scene in small image regions. In this process, first a blur parameter σ which is a measure of the spread of the camera's Point Spread Function (PSF) is estimated as an intermediate step. In this chapter we investigate two methods for using the same blur parameter σ for recovering the focused images of objects in the scene from their blurred images.

The first method of focused image recovery is based on the Spatial Domain Convolution/ Deconvolution Transform (S-Transform) proposed in [113]. This method uses only the blur parameter σ which is a measure of the spread of the camera's PSF. In particular, the method does not require a knowledge of the the exact form of the camera PSF. The second method, in contrast to the first, requires complete information about the form of the camera PSF. For most practical camera systems, the camera PSF cannot be characterized with adequate accuracy using simple mathematical models such as Gaussian or cylindrical functions. A better model is obtained by measuring experimentally the actual PSF of the camera for different degrees of image blur and using the

measured data. This however requires camera calibration. An alternative but usually a more difficult solution is to derive and use a more accurate mathematical model for the PSF based on diffraction, lens aberrations, and characteristics of the various camera components such as the optical system, image sensor elements, frame grabber, etc. As part of the second method, we describe a camera calibration procedure for measuring the camera PSF for various degrees of image blur. The calibration procedure is based on recording and processing the images of blurred step edges. In the second method, the focused image is obtained through a deconvolution operation in the Fourier domain using the Wiener filter.

For both methods of recovering the focused image, results of experiments on the SPARCS camera system are presented. The results of the first method are compared with the results obtained using two commonly used PSF models– cylindrical based on geometric optics, and a 2-D Gaussian. The results of the second method are compared with simulation results.

6.2 Spatial Domain Approach

In this section we describe the spatial domain method for recovering the focused image of a 3-D scene from a defocused image for which the blur parameter σ has been estimated using either DFD1F or STM. The recovery is done through deconvolution of the defocused image using the Spatial-Domain Convolution/ Deconvolution Transform (S Transform) [113]. The transform itself is general and applicable to n-dimensional continuous and discrete signals

for the case of arbitrary order polynomials.

6.2.1 S - Transform

Let $f(x, y)$ be an image which is a two variable cubic polynomial in a small neighborhood, defined by

$$f(x, y) = \sum_{m=0}^{3} \sum_{n=0}^{3-m} a_{m,n} x^m y^n \tag{6.1}$$

where $a_{m,n}$ are the polynomial coefficients. Let $h(x, y)$ be the PSF of a camera. The moment $h_{m,n}$ of the PSF is defined by

$$h_{m,n} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^m y^n h(x, y)\, dxdy \tag{6.2}$$

Let $g(x, y)$ be the blurred image obtained by convolving the focused image $f(x, y)$ with the PSF $h(x, y)$. The blur parameter σ is a measure of the spread of the camera PSF. For a circularly symmetric PSF denoted by $h(x, y)$ it is defined as

$$\sigma^2 = \int_{\infty}^{\infty} \int_{\infty}^{\infty} (x^2 + y^2)\, h(x, y)\, dx\, dy \tag{6.3}$$

For a PSF model based on paraxial geometric optics, it can be shown that the blur parameter σ is proportional to the blur circle radius. If R is the blur circle radius, then $\sigma = R/\sqrt{2}$. For a PSF model based on a 2D Gaussian function, σ is the standard deviation of the distribution of the 2D Gaussian function.

In Chapter 4 we saw that $f(x, y)$ can be obtained as

$$f(x, y) = g(x, y) - \frac{h_{2,0}}{2} \bigtriangledown^2 g(x, y) \tag{6.4}$$

Equation (6.4) is a deconvolution formula. It expresses the original function (focused image) $f(x, y)$ in terms of the convolved function (blurred image)

$g(x, y)$, its (i.e. g's) derivatives, and the moments of the point spread function $h(x, y)$. In the general case this corresponds to *Inverse S-Transform* [113].

Using the definitions of the moments of h and the definition of the blur parameter σ of h, we have $h_{2,0} = h_{0,2} = \sigma^2/2$, and therefore the above deconvolution formula can be written as

$$f(x, y) = g(x, y) - \frac{\sigma^2}{4} \nabla^2 g(x, y) \tag{6.5}$$

The above equation suggests a method for recovering the focused image $f(x, y)$ from the blurred image $g(x, y)$ and the blur parameter σ. Note that the above equation has been derived under the following assumptions (i) the focused image $f(x, y)$ is modeled by a cubic polynomial (as in Eq. 6.1) in a small (3×3 pixels in our implementation) image neighborhood, and (ii) the PSF $h(x, y)$ is circularly symmetric. These two assumptions are good approximations in practical applications and yield useful results.

Equation (6.5) is similar in form to the previously known result that a sharper image can be obtained from a blurred image by subtracting a constant times the Laplacian of the blurred image from the original blurred image [99]. However that result is valid only for a diffusion model of blurring where the PSF is restricted to be a Gaussian. In comparison, the deconvolution formula of equation (6.5) is valid for all PSFs that are circularly symmetric including a Gaussian. Therefore, the previously known result is a special case of our deconvolution. Further, the restriction on the circular symmetry of the PSF can be removed if desired using a more general version of the S-Transform [113]. Also, the focused image can be generalized to be an arbitrarily high order polynomial although such a generalization does not seem useful in practical

applications.

The main advantages of this method are (i) the quality of the focused image obtained (as we shall see in the discussion on experimental results), (ii) computational complexity, and (iii) the *locality* of the computations. *Simplicity* of the computational algorithm is another characteristic of this method. Given the blur parameter σ, at each pixel, estimation of the focused image involves the following operations (a) estimation of the Laplacian which can be implemented with a few integer addition operations (8 in our implementation), (b) floating point multiplication of the estimated Laplacian with $\sigma^2/4$, and (c) one integer operation corresponding to the subtraction in Eq. (6.5). For comparison purposes in the following sections, let us say that these computations are roughly equivalent to 4 floating point operations. Therefore, for an $N \times N$ image, about $4N^2$ floating point operations are required. All operations are *local* in that only a small image region is involved (3×3 in our implementation). Therefore the method can be easily implemented on a parallel computation hardware.

6.2.2 Experiments

A set of experiments is described later where the blur parameter σ is first estimated from two blurred images and then the focused image is recovered. In this section we describe experiments where σ is assumed to be given.

A poster with printed characters was placed at a distance of step 70 (about 80 cms) from the camera. The focused image is shown in Figure 6.1. The camera lens was moved to different positions (steps 70, 60, 50, 40, 30 and

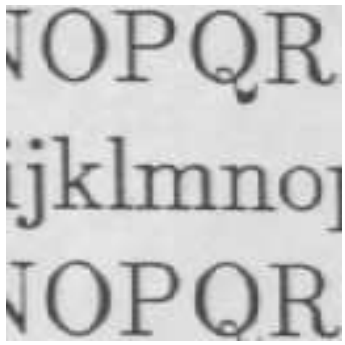


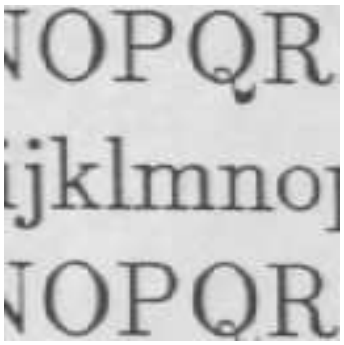
Figure 6-1: Focused character image

20) to obtain images with different degrees of blur. The images are shown in figures 6.2(a) to 6.7(a). The corresponding blur parameters (σs) for these images were roughly 2.2, 2.8, 3.5, 4.7, 6.0 and 7.2 pixels. These images were deblurred using equation (6.5). The results are shown in Figures 6.2(d) - 6.7(d). We see that the results are satisfactory for small to moderate levels of blur corresponding to about $\sigma = 3.5$ pixels. This corresponds to about 20 lens steps or a blur circle radius of about 5 pixels.

In order to evaluate the above results through comparison, two standard techniques were used to obtain focused images. The first technique was to use a two-dimensional Gaussian model for the camera PSF. The spread parameter of the Gaussian function was taken to be equal to the blur parameter σ, and therefore the PSF was:

$$h_b(x, y) \;=\; \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{6.6}$$

The plots of the PSF for two values of σ corresponding to about 3.75 pixels and 7.5 pixels are shown in Figure 6.8.



(a) Blurred Image

(b) Restored by
Geometric PSF Model

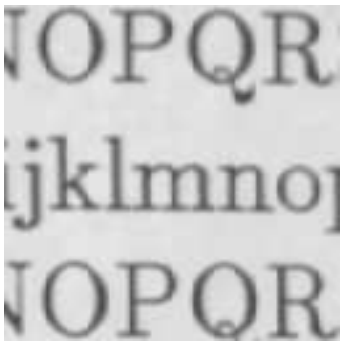
(c) Restored by
Gaussian PSF Model

(d) Restored by
S-Transform

(e) Restored by
Separable MTF Model

(f) Restored using Actual
PSF (Abel Transform)

Figure 6-2: Restoration with 0 steps of blur



(a) Blurred Image

(b) Restored by
Geometric PSF Model

(c) Restored by
Gaussian PSF Model

(d) Restored by
S-Transform

(e) Restored by
Separable MTF Model

(f) Restored using Actual
PSF (Abel Transform)

Figure 6-3: Restoration with 10 steps of blur

(a) Blurred Image

(b) Restored by
Geometric PSF Model

(c) Restored by
Gaussian PSF Model

(d) Restored by
S-Transform

(e) Restored by
Separable MTF Model

(f) Restored using Actual
PSF (Abel Transform)

Figure 6-4: Restoration with 20 steps of blur

(a) Blurred Image

(b) Restored by
Geometric PSF Model

(c) Restored by
Gaussian PSF Model

(d) Restored by
S-Transform

(e) Restored by
Separable MTF Model

(f) Restored using Actual
PSF (Abel Transform)

Figure 6-5: Restoration with 30 steps of blur

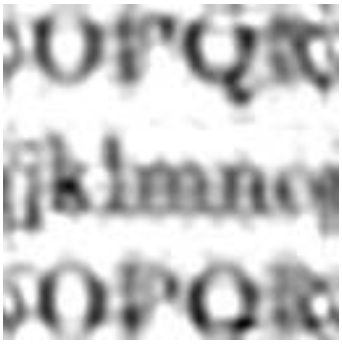
(a) Blurred Image

(b) Restored by
Geometric PSF Model

(c) Restored by
Gaussian PSF Model

(d) Restored by
S-Transform

(e) Restored by
Separable MTF Model



(f) Restored using Actual
PSF (Abel Transform)

Figure 6-6: Restoration with 40 steps of blur

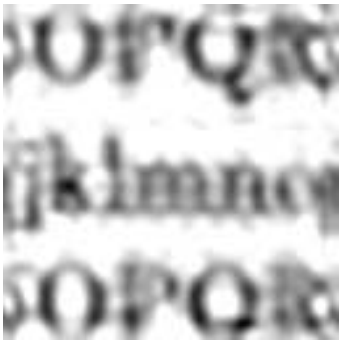
(a) Blurred Image

(b) Restored by
Geometric PSF Model

(c) Restored by
Gaussian PSF Model

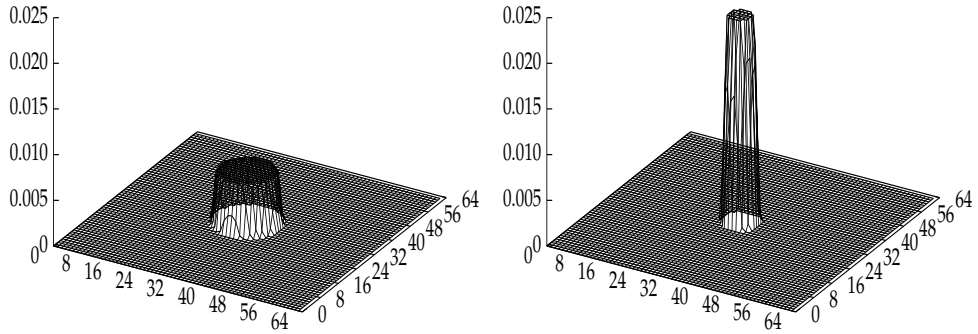
(d) Restored by
S-Transform

(e) Restored by
Separable MTF Model

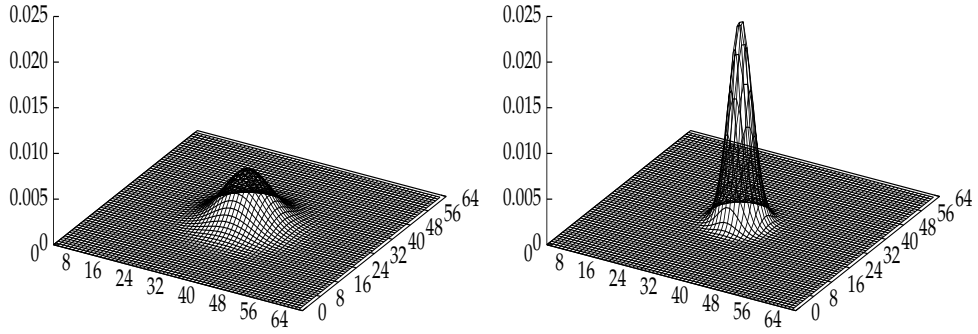


(f) Restored using Actual
PSF (Abel Transform)

Figure 6-7: Restoration with 50 steps of blur



(a) Geometric optics PSF with radius 3.75 and 7.5 pixels



(b) Gaussian PSF with radius 3.75 and 7.5 pixels

Figure 6-8: Cylindrical and Gaussian PSF models

The focused image was obtained using the Wiener filter [99] specified in the Fourier domain by:

$$M(\omega,\nu) = \frac{1}{H(\omega,\nu)} \frac{|H(\omega,\nu)|^2}{|H(\omega,\nu)|^2 + \Gamma} \tag{6.7}$$

where $H(\omega,\nu)$ is the Fourier Transform of the PSF and Γ is the noise-to-signal power density ratio. In our experiments Γ was approximated by a constant. The constant was determined empirically through several trials so as to yield best results. Let $g(x,y)$ be the blurred image, and $\hat{f}(x,y)$ be the restored focused image. Let their corresponding Fourier Transforms be $G(\omega,\nu)$ and $\hat{F}(\omega,\nu)$ respectively. Then the restored image, according to Wiener filtering is

$$\hat{F}(\omega,\nu) = G(\omega,\nu)M(\omega,\nu). \tag{6.8}$$

By taking the inverse Fourier Transform of $\hat{F}(\omega,\nu)$, we can obtain the restored image $\hat{f}(x,y)$.

The results are shown in Figures 6.2(c)-6.7(c). We see that for small values of σ (about 3.5 pixels), the Gaussian model performs well, but not as good as the previous method (Figs. 6.2(d)-6.7(d)). In addition to the quality of the focused image that is obtained, this method has three important disadvantages. The first is computational complexity. For a given σ, first one needs to compute the the OTF $H(\omega,\nu)$, and then the Weiner filter $M(\omega,\nu)$. It is possible to precompute and store $M(\omega,\nu)$ for later usage for different values of σ. But this would require large storage space. After $M(\omega,\nu)$ has been obtained for a given σ, we need to compute $G(\omega,\nu)$ from $g(x,y)$ using FFT algorithm, multiply $M(\omega,\nu)$ with $G(\omega,\nu)$ to obtain $\hat{F}(\omega,\nu)$, and then

compute the inverse Fourier Transform of $\hat{F}(\omega, \nu)$. The complexity of the FFT algorithm is $O(N^2 logN)$ for an $N \times N$ image. Roughly, at least $(2N^2 + 2N^2 log_2 N)$ floating point operations are involved. For $N = 128$ used in our experiments, the number of computations is at least $16N^2$. In comparison, the number of computations in the previous case was $4N^2$. Therefore, this method is at least 4 times slower than the previous method. The second disadvantage of this method is that the computations are not local because of the computation of the Fourier Transform of the entire image. The third disadvantage is the estimation of the noise parameter Γ.

In the second standard technique of focused image recovery, the PSF was modeled by a cylindrical function based on paraxial geometric optics. The plots of the PSF for two values of σ of about 3.75 pixels and 7.5 pixels are shown in Figure 6.8. With a knowledge of the blur parameter σ, it is possible to generate the entire cylindrical PSF. The focused image was again obtained using the Wiener filter mentioned earlier, but this time using the cylindrical PSF.

In computing the Wiener filter, computation of the discrete cylindrical PSF at the border of the corresponding blur circle involves some approximations. The value of a pixel which lies only partially in the blur circle should be proportional to the area of overlap between the pixel and the blur circle. Violation of this rule leads to large errors in the restored image, especially for small blur circles. In our implementation, the areas of partial overlap were computed by resampling the ideal PSF at a higher rate (about 16 times), calculating the PSF by ignoring the pixels whose center did not lie within

the blur circle, and then downsampling by adding the pixel values in 16 x 16 non-overlapping regions.

The results of this case are shown in Figures 6.2(b)-6.7(b) for different degrees of blur. The images exhibit "ripples" around the border between the background and the characters. Once again we see that the results are not as good as for the S Transform method. For low levels of blur (upto about $R = 5$ pixels) Gaussian model gives better results than the cylindrical PSF, and for higher levels of blur (R greater than about 5 pixels) the cylindrical PSF gives better results than the Gaussian PSF.

In addition to the quality of the final result, the relative disadvantages of this method in comparison with the S Transform method are same as those for the Gaussian PSF model.

6.3 Second Method

In the second method of focused image recovery, the blur parameter σ is used to first determine the complete PSF. In practice, the PSF is determined by using σ as an index into a prestored table that specifies the complete PSF for different values of σ. In theory, however, the PSF may be determined by substituting σ into a mathematical expression that models the actual camera PSF. Since it is difficult to obtain a sufficiently accurate mathematical model for the PSF, we use a prestored table to determine the complete PSF. After obtaining the complete PSF, Wiener filtering is used to compute the focused image. First we describe a method of obtaining the prestored table through a

calibration procedure.

6.3.1 Camera calibration for PSF

Theoretically, the PSF of a camera can be obtained from the image of a point light source. However, in practice, it is difficult to create an ideal point light source that is incoherent and polychromatic. Therefore the standard practice in camera design is to estimate the PSF from the image of an edge.

Let $f(x, y)$ be a step edge along the y-axis on the image plane. Let a be the image intensity to the left of the y-axis and b be the height of the step. The image can be expressed as

$$f(x, y) = a + b\ u(x) \tag{6.9}$$

where $u(x)$ is the standard *unit step function*. If $g(x, y)$ is the observed image and $h(x, y)$ is the camera's PSF then we have,

$$g(x, y) = h(x, y) * f(x, y) \tag{6.10}$$

where * denotes the convolution operation.

Now consider the derivative of g along the gradient direction. Since differentiation and convolution commute, we have

$$\frac{\partial g}{\partial x} = h(x, y) * \frac{\partial f}{\partial x} \tag{6.11}$$

$$= h(x, y) * b\ \delta(x) \tag{6.12}$$

where $\delta(x)$ is the *dirac delta* function along the x axis. The above expression can be simplified to obtain

$$\frac{\partial g}{\partial x} = b\ \theta(x) \tag{6.13}$$

where $\theta(x)$ is the *line spread function* of the camera defined by

$$\theta(x) = \int_{-\infty}^{\infty} h(x,y)dy \tag{6.14}$$

For any PSF $h(x,y)$ of a lossless camera, by definition, we have

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\, dx\, dy = 1 \tag{6.15}$$

Therefore we obtain

$$\int_{-\infty}^{\infty} \frac{\partial g(x,y)}{\partial x} dx = b \tag{6.16}$$

Therefore, given the observed image $g(x,y)$ of a blurred step edge, we can obtain the line spread function $\theta(x)$ from the expression

$$\theta(x) = \frac{\frac{\partial g}{\partial x}}{\int_{-\infty}^{\infty} \frac{\partial g}{\partial x} dx} \tag{6.17}$$

After obtaining the line spread function $\theta(x)$, the next step is to obtain the PSF or its Fourier Transform, which is the Optical Transfer Function (OTF). Here we outline two methods of obtaining the OTF, one assuming the separability of the OTF and another using the Inverse Abel Transform.

Separable OTF

Let the Fourier Transforms of the PSF $h(x,y)$ and LSF $\theta(x)$ be $H(\omega,\nu)$ and $\Phi(\omega)$ respectively. Then we have [99]

$$\Phi(\omega) = H(\omega, 0) \tag{6.18}$$

If the camera has a circular aperture then the PSF is circularly symmetric. If the PSF is circularly symmetric (and real), then the OTF is also circularly

symmetric (and real), i.e. $H(\omega, \nu)$ is also circularly symmetric. Therefore we get

$$H(\omega, \nu) = \Phi(\sqrt{\omega^2 + \nu^2}) \tag{6.19}$$

Once we have the Fourier Transform of the LSF, $\Phi(\omega)$, we can calculate $H(\omega, \nu)$ for any values of ω and ν. However, in practice where digital images are involved, $\sqrt{\omega^2 + \nu^2}$ may have non integer values, and we may have to interpolate $\Phi(\omega)$ to obtain $H(\omega, \nu)$. Due to the nature of $\Phi(\omega)$, linear interpolation did not yield good results in our experiments. Therefore interpolation was avoided by assuming the OTF to be separable, i.e. $H(\omega, \nu) = H(\omega, 0)H(0, \nu) = \Phi(\omega)\Phi(\nu)$. A more accurate method, however, is to use to the Inverse Abel Transform.

Inverse Abel Transform

In the case of a circularly symmetric PSF $h_1(r)$, the PSF can be obtained from its LSF $\theta(x)$ directly using the Inverse Abel Transform [55] :

$$h(r) = -\frac{1}{\pi} \int_r^\infty \frac{\theta'(x)}{\sqrt{x^2 - r^2}} dx \tag{6.20}$$

where $\theta'(x)$ is the derivative of LSF $\theta(x)$. Note that $h(x, y) = h_1(r)$ if $r = \sqrt{x^2 + y^2}$. In our implementation the above integral was evaluated using a numerical integration technique.

After obtaining $H(\omega, \nu)$, the final step in restoration is to use equations (6.7) and (6.8) and obtain the restored image.

6.3.2 Calibration Experiments

All experiments were performed using the SPARCS camera system. Black and white stripes of paper were pasted on a cardboard to create a step discontinuity in reflectance along a straight line. The step edge was placed at such a distance (about 80 cms) from the camera that it was in best focus when the lens position was step 70. The lens was then moved to 20 different positions corresponding to step numbers $0, 5, 10 \cdots 90, 95$. At each lens position, the image of the step edge was recorded, thus obtaining a sequence of blurred edges with different degrees of blur. Twelve of these images are shown in Figure 6.9. The difference between the actual lens position and the reference lens position of 70 is a measure of image blur. Therefore, an image blur of +20 steps corresponds to an image recorded at lens position of step 50 and an image blur of -20 steps corresponds to an image recorded at lens position of step 90. The size of each image was 80×200 pixels.

In the experiments, the step edge was placed vertically and therefore the image intensity was almost a constant along columns and the gradient direction was along the rows. To reduce electronic noise, each image was cut into 16 horizontal strips of size 5×200 and in each strip, the image intensity was integrated (summed) along columns. Thus each strip was reduced to just one image row. In each row, the first derivative was computed by simply taking the difference of gray values of adjacent pixels. Then the approximate location of the edge was computed in each row by finding the first moment of the derivative, i.e., if $\bar{i}$ is the column number where the edge is located, and

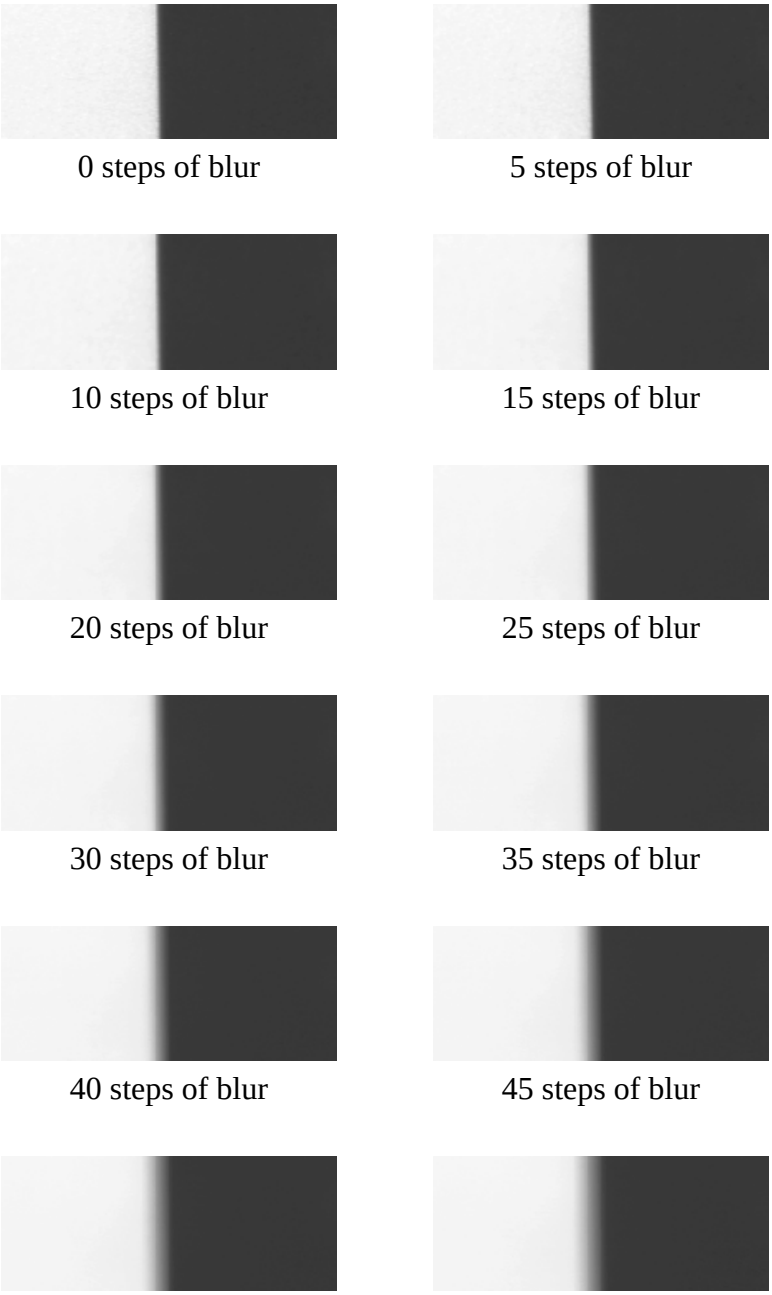


Figure 6-9: Images of blurred step edges

$g_x(i)$ is the image derivative at column i, then

$$\bar{i} = \frac{\sum_{i=1}^{i=200} i g_x(i)}{\sum_{i=1}^{i=200} g_x(i)} \tag{6.21}$$

The following step was included to reduce the effects of noise further. Each row was traversed on either side of position $\bar{i}$ until a pixel was reached where either $g_x(i)$ was zero or its sign changed. All the pixels between this pixel (where for the first time, g_x became zero or its sign changed) and the pixel at the row's end were set to zero. We found this noise cleaning step to be very important in our experiments. A small non-zero value of image derivative caused by noise at pixels far away from the position of the edge affects the estimation of the blur parameter σ considerably.

From the noise-cleaned $g_x(i)$, the LSF was computed as

$$\theta(i) = \frac{g_x(i)}{\sum_{i=1}^{i=200} g_x(i)} \tag{6.22}$$

Eight LSFs corresponding to different degrees of blur are plotted in Figure 6.10. It can be seen that, as the blur increases the LSF function becomes more flat and spread out. The location of the edge $\bar{i}$ was then recomputed using equation (6.21). The spread or second central moment of the LSF, σ_l was computed from

$$\sigma_l = \sqrt{\sum_{i=1}^{200} (i - \bar{i})^2 \theta(i)} \tag{6.23}$$

The computed values of σ_l for adjacent strips were found to differ by only about 2 percent. The average $\overline{\sigma_l}$ was computed over all the strips. It can be shown that σ_l is related to the blur parameter σ by $\sigma = \sqrt{2}\sigma_l$. The effective blur circle radius R is related to σ by $R = \sqrt{2}\sigma$. The values of R computed

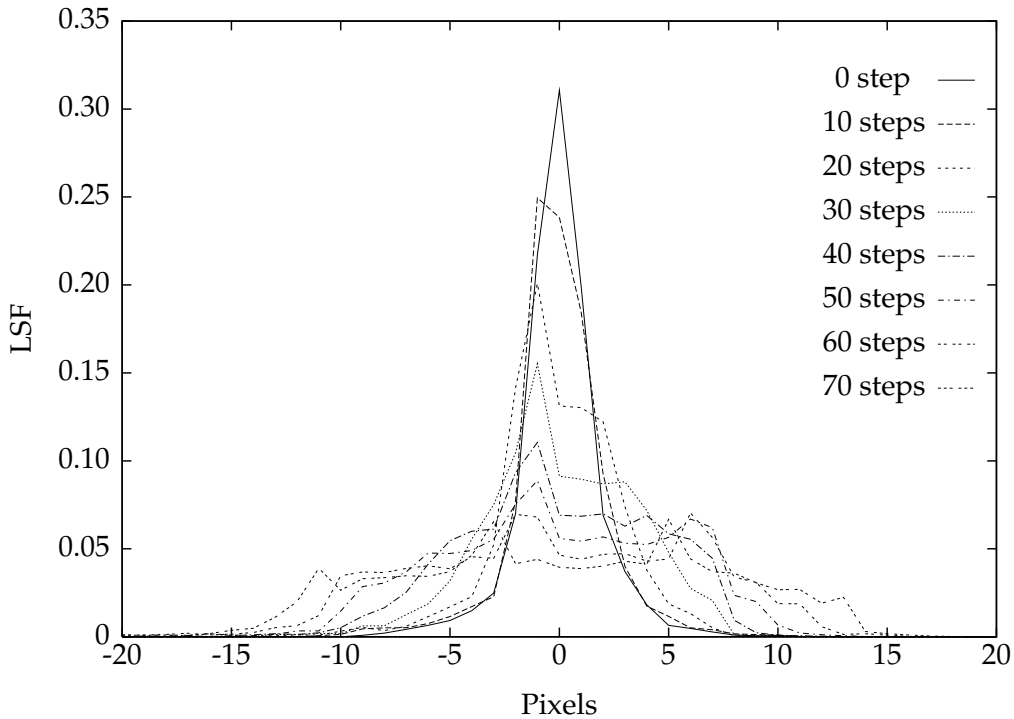


Figure 6-10: LSFs obtained from step edges

using the relation $R = 2\sigma_l$ for different step edges are shown in Figure 6.11. Figure 6.11 also shows the value of R predicted by ideal paraxial geometric optics. The values of R obtained for a horizontal step edge are also plotted in the figure. The values for the vertical and horizontal edges are in close agreement except for very low degrees of blur. This minor discrepancy may be due to the asymmetric (rectangular) shape of the CCD pixels (13×11 microns for our camera).

The PSF's were obtained from the LSFs using the Inverse Abel Transform. Cross sections of the PSFs thus obtained corresponding to the LSFs in Figure 6.10 are shown in Figure 6.12.

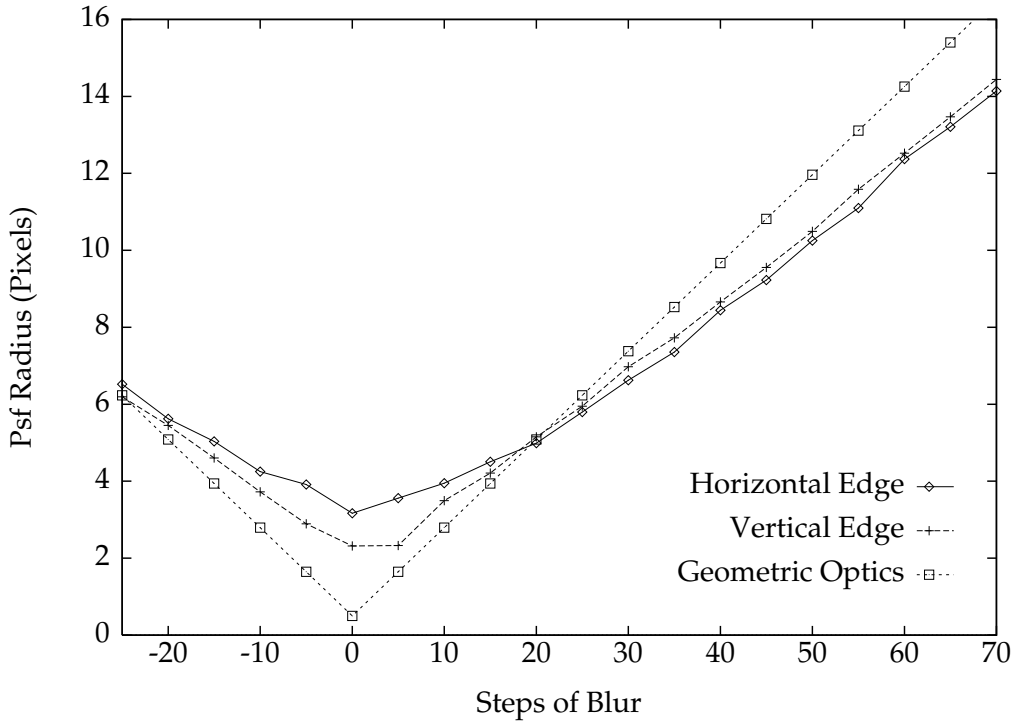


Figure 6-11: PSF radius from step edges

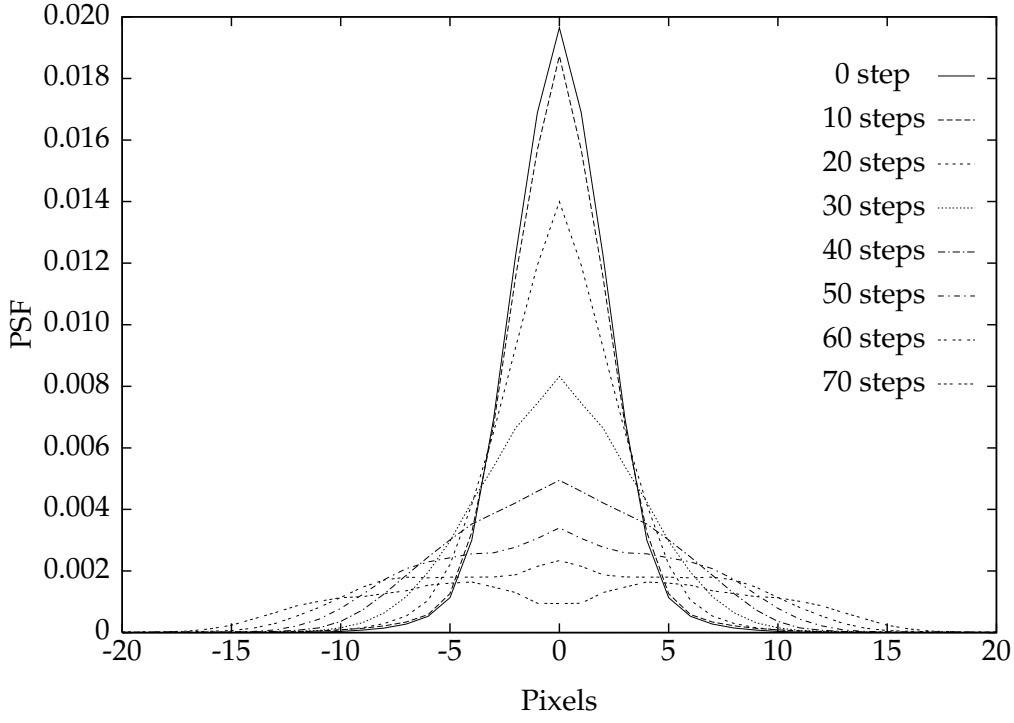


Figure 6-12: PSFs using Inverse Abel Transform

6.3.3 Experimental Results

Using the calibration procedure described in the previous section, the PSFs and the corresponding OTFs were precomputed for different values of the blur parameter σ. These results were prestored in a lookup table indexed by σ. The OTF data $H(\omega, \nu)$ in this table was used to restore blurred images using the Wiener filter $M(\omega, \nu)$. Figures 6.2(e)-6.7(e) show the results of restoration using the separability assumption for the OTF and Figures 6.2(f)-6.7(f) are the results for the case where the Inverse Abel Transform was used to compute the PSF from the LSF. Both these results are better than the other results in Figures 6.2 (b,c,d) - 6.7(b,c,d). The method using the Inverse Abel Transform is better than all the other methods. We find that the results in this case are good even for highly blurred images. For example, the images in Figures 6.6(a) and 6.7(a) are severely blurred corresponding to 40 and 50 steps of blur or σ equal to about 6.0 and 7.2 pixels respectively. It is quite difficult for humans to recognize the characters in these images. However, in the restored images shown in Figures 6.6(f) and 6.7(f) respectively, many of the characters are easily recognizable.

In order to compare the above results with the best obtainable results, the restoration method which uses the Inverse Abel Transform was tested on computer simulated image data. Two sets of blurred images were obtained by convolving an original image with a cylindrical and a Gaussian function. The only noise in the simulated images was quantization noise. The blurred images were then restored using the Wiener filter. The results are shown in Figures 6.13 and 6.14. We see that these results are only somewhat better but

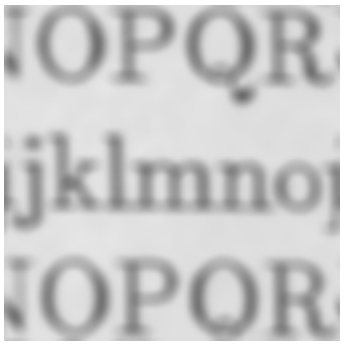
not much better than the results on actual data in Figures 6.2(f)-6.7(f). This indicates that the method of camera calibration for the PSF is reliable.

The main advantage of this method is that the quality of the restored image is the best in comparison with all other methods. It gives good results for even highly blurred images. It has two main disadvantages. First, it requires extensive calibration work as described earlier. Second, the computational complexity is the same as that for the Wiener filter method discussed earlier. For an $N \times N$ image, it requires at least $2N^2 + 2N^2 log_2 N$ floating point operations as compared with $4N^2$ floating point operations for the method based on spatial domain deconvolution. Therefore, for an image of size 128×128, this method is at least 4 times slower than the method based on spatial domain deconvolution. Another disadvantage is that it requires the estimation of the noise parameter Γ for the Wiener filter.

6.4 Experiments with unknown σ and 3-D objects

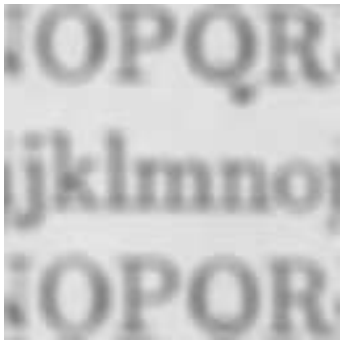
In the experiments described earlier, the blur parameter σ of a blurred image was taken to be known. We now present a set of experiments where σ is unknown. It is first estimated using STM. Then, of the two blurred images, the one that is less blurred is deconvolved to recover the focused image. Results are presented for both the first method based on spatial-domain deconvolution and the second method which uses Inverse Abel Transform.

The results are shown in Figures 6.15(a-d). The first image in Fig. 6.15(a)



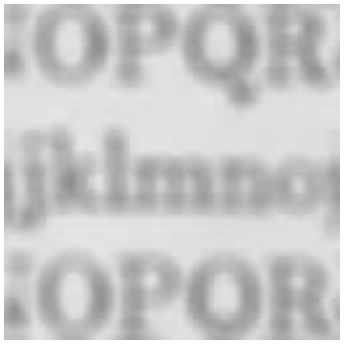
Blurred 0 steps

Restored 0 steps

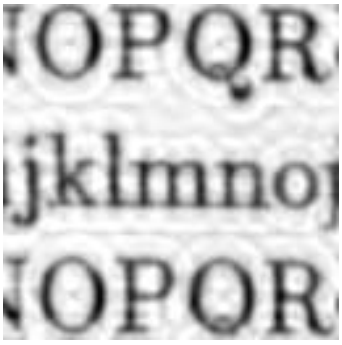


Blurred 10 steps

Restored 10 steps



Blurred 20 steps



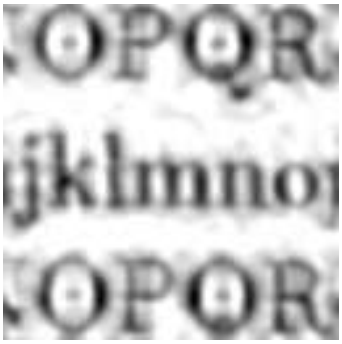
Restored 20 steps

Figure 6.13(a) Simulation with geometric optics PSF

Blurred 30 steps

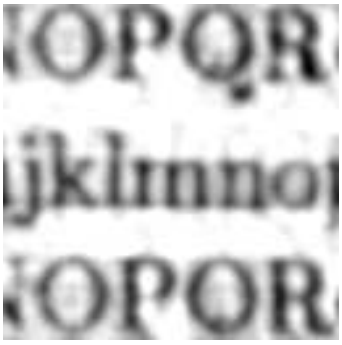
Restored 30 steps

Blurred 40 steps



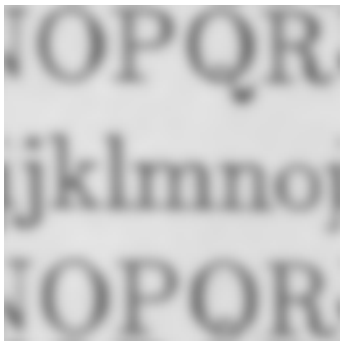
Restored 40 steps

Blurred 50 steps



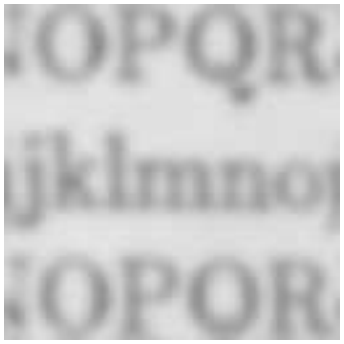
Restored 50 steps

Figure 6-13: (b) Simulation with geometric optics PSF



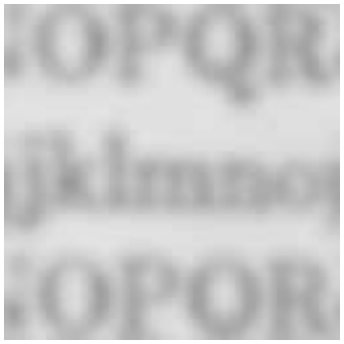
Blurred 0 steps

Restored 0 steps



Blurred 10 steps

Restored 10 steps



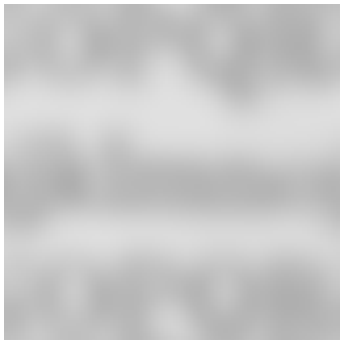
Blurred 20 steps

Restored 20 steps

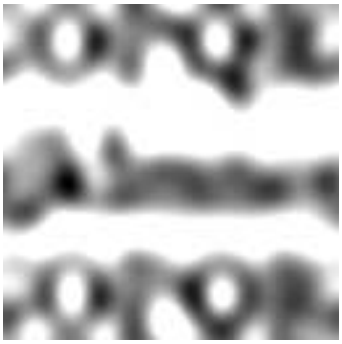
Fig. 6.14(a) Simulation with Gaussian PSF

Blurred 30 steps

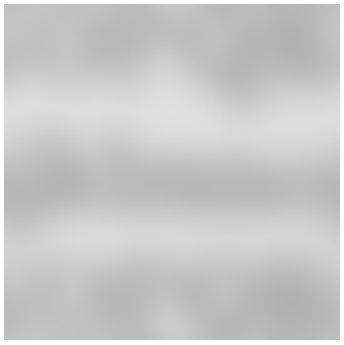
Restored 30 steps



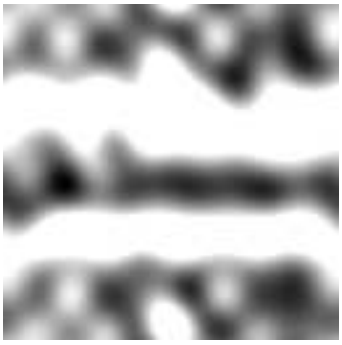
Blurred 40 steps



Restored 40 steps



Blurred 50 steps

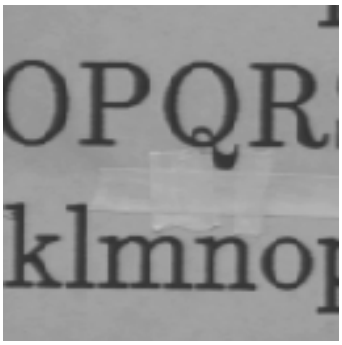


Restored 50 steps

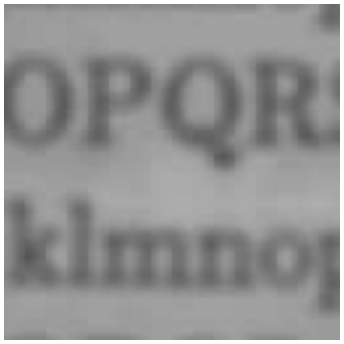
Figure 6-14: (b) Simulation with Gaussian PSF

is the focused image of an object recorded by the camera. The object was placed at a distance of step 14 (about 2.5 meters) from the camera. Two images of the object were recorded with two different lens positions–steps 40 and 70 (see Fig. 6.15(a)). The blur parameter σ was estimated using STM. It was found to be about 5.5 pixels. Using this, the results of restoring the image recorded at lens step 40 is shown in Fig. 6.15(a). Similar experiments were done by placing the object at distances steps 36, 56, and 76 corresponding to 1.31, 0.9 and 0.66 meters from the camera. In each of these cases, the focused image, the two recorded image at steps 40 and 70, and the restored images are shown in Figs. 6.15 (b-d). The blur parameters in the three cases were about 1.79, 1.24, and 2.35 pixels respectively. In the last two cases, the images recorded at lens step 70 was less blurred than the the one recorded at step 40. Therefore the image recorded at lens step 70 was used in the restoration.

In another experiment, a 3-D scene was created by placing three planar objects at three different distances. Two images of the objects were recorded at lens steps 40 and 70. These images are shown in Figure 6.16. It can be seen that different image regions are blurred by different degrees. The image was divided into 9 regions of size 128 x 128 pixels. In each region the blur parameter σ was estimated and the image in the region was restored. The nine different estimated values of σ are 3.84, 4.76, 4.76, 0.054, 0.15, 0.46 (for image with lens step 40) and -2.65, -2.55 and -2.55 (for image with lens step 70) respectively. The different restored regions were combined to yield an image, where the entire image looks focused. Figure 6.16 shows the results using both the first and second methods of restoration. Currently each region can be as



Focused Image
(Focused at step 14)



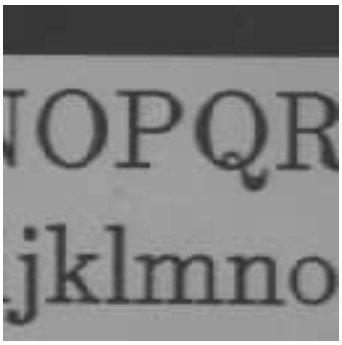
Blurred Image
(Lens at step 40)

Blurred Image
(Lens at step 70)

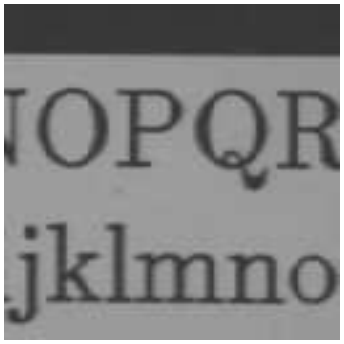
Restored by
S-Transform

Restored using Actual
PSF (Abel Transform)

Fig. 6.15(a) Depth estimation with restoration for step 14



Focused Image
(Focused at step 36)



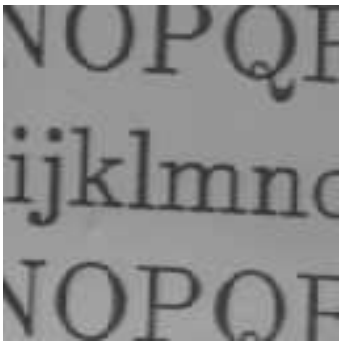
Blurred Image
(Lens at step 40)

Blurred Image
(Lens at step 70)

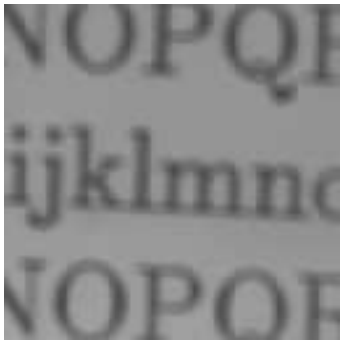
Restored by
S-Transform

Restored using Actual
PSF (Abel Transform)

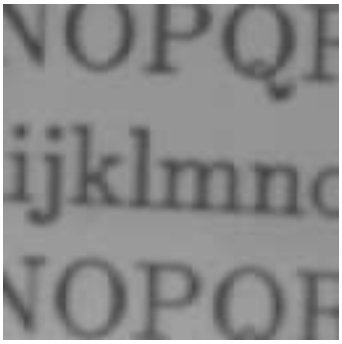
Fig. 6.15(b) Depth estimation with restoration for step 36



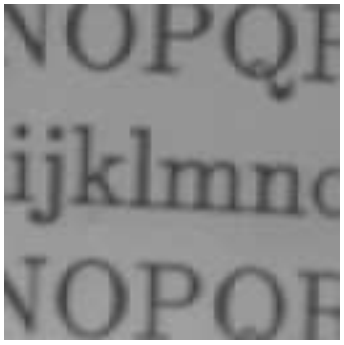
Focused Image
(Focused at step 56)



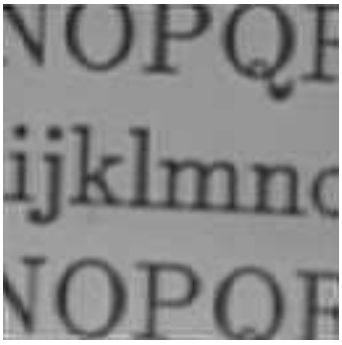
Blurred Image
(Lens at step 40)



Blurred Image
(Lens at step 70)

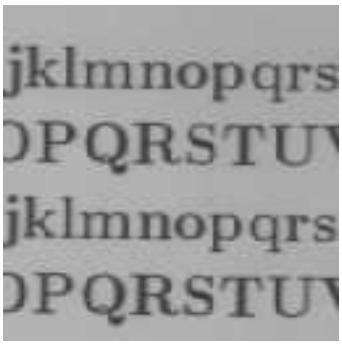


Restored by
S-Transform

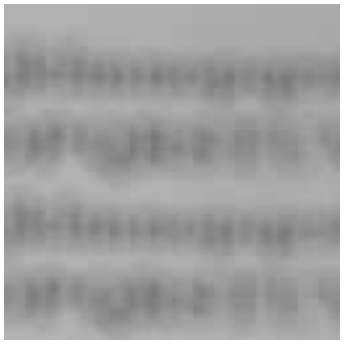


Restored using Actual
PSF (Abel Transform)

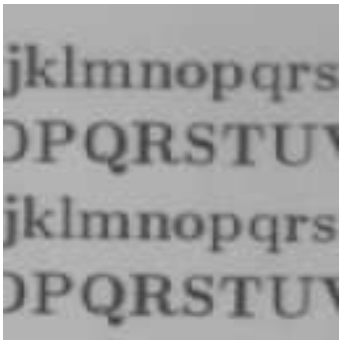
Fig. 6.15(c) Depth estimation with restoration for step 56



Focused Image
(Focused at step 76)

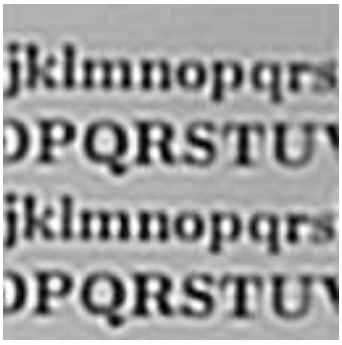


Blurred Image
(Lens at step 40)



Blurred Image
(Lens at step 70)

Restored by
S-Transform



Restored using Actual
PSF (Abel Transform)

Figure 6-15: (d) Depth estimation with restoration for step 76

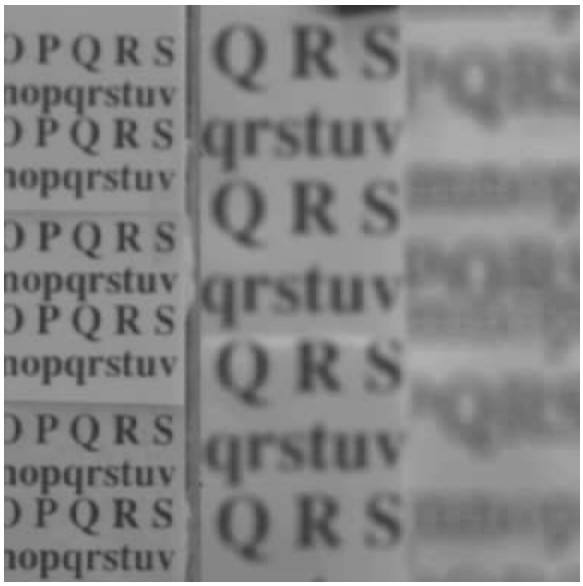
small as 48 x 48 pixels, which is a small region in the entire field of view of 640 x 480 pixels.

In the next experiment, a planar object with posters was placed inclined to the optical axis. The nearest end of the object was about 50 cms from the camera and the farthest end was about 120 cms. The blurred images of the object acquired with lens steps 40 and 70 are shown in Figure 6.17(a) and (b). The images were divided into non-overlapping regions of 64 x 64 pixels and a depth estimate was obtained for each region. The depth estimates obtained in each region are shown in terms of step numbers in Table 6.1. The different regions were then restored separately as before and combined to yield the restored images as shown in Figure 6.17(c) and (d). The restored images appear better than either of the blurred images. However there are some blocking artifacts, which are due to the "wrap around" problem of the FFT algorithm and the finite filter size in the case of the S-Transform method.

6.5 Conclusion

In this chapter we have shown that the focused image of an object can be recovered using two defocused images recorded with different camera parameter settings. The same two images can be used to estimate the depth of the object using STM. For a 3-D scene where the depth variation is small in image regions of size about 64×64, each image region can be processed separately and the results can be combined to obtain both a focused image of the entire scene and a rough depth-map of the scene. If, in each image region, at least

(a) Blurred Image
(Lens Step 40)



(b) Blurred Image
(Lens Step 70)

Fig 6.16 (a),(b) Depth estimation with restoration for 3-D object

(c) Restored by
S-Transform

(d) Restored using Actual
PSF (Abel Transform)

Figure 6-16: (c),(d) Depth estimation with restoration for 3-D object

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
107	107	107	90	87	79	73	71	78	78
93	93	101	91	85	80	78	67	80	80
107	107	96	92	86	81	71	76	71	71
92	93	93	88	83	78	77	72	77	77
95	95	85	93	86	79	73	67	71	71
104	104	95	93	87	79	74	71	77	77
107	107	95	89	85	82	75	72	79	79

Table 6.1. Depth estimates in each region of slanted object

(Step Number)

Figure 6.17 (a) Image of slanted object with lens step 40

Figure 6.17 (b) Image of slanted object with lens step 70

Figure 6.17 (c) Restored using S-Transform

Figure 6-17: (d) Restored using actual PSF (Inv. Abel Transform)

one of the two recorded defocused images is blurred only moderately or less ($\sigma \leq 3.5$ pixels), then the focused image can be recovered very fast (computational complexity of $O(N^2)$ for an $N \times N$ image) using the new spatial domain deconvolution method described here. In most practical applications of machine vision, the camera parameter setting can be arranged so that this condition holds, i.e. in each image region at most only one of the two recorded defocused images is severely blurred ($\sigma > 3.5$ pixels). In those cases where this condition does not hold, the second method which uses the Inverse Abel Transform can be used to recover the focused image. This method requires camera calibration for the PSF and is several times more computationally intensive than the first method above. The methods described in this chapter can be used as part of a 3-D machine vision system to obtain focused images from blurred images for further processing such as edge detection, stereo matching, and image segmentation.

Chapter 7

Experiments with an Off-the-Shelf Camera

7.1 Introduction

In this chapter we describe experiments with an off-the-shelf camera (DELTIS 2000 of Olympus Optical Company). DELTIS is an off-the-shelf camera and no lens data is available. It has a motorized lens which can be controlled from a computer through a serial port. We first describe the implementation of STM and experimental results, on the DELTIS camera. Next we describe the details of a 3-D shape recovery method using DFF. We analyse a few commonly used focus measures and interpolation schemes and provide experimental results. The results indicate that both STM and DFF algorithms can easily be implemented with off-the-shelf cameras.

7.2 S-Transform Method

In Chapter 4, we described a method of calculating the spread parameter σ_2 using two blurred images g_1 and g_2. The final expression was given by

equation (4.48) as

$$\sigma_2 = \frac{G' - \beta^2}{2\beta} \tag{7.1}$$

where G' depends on g_1 and g_2 and their laplacians. It can be seen that the computed value of σ_2 depends linearly on the camera constant β. In the case of DELTIS camera, the constants α and β are not known. However, if an arbitrary value of α and β are used, only the slope and intercept of the calibration plot are changed and hence there is still a one-to-one correspondence between the computed value of σ_2 and the object distance.

As described in Chapter 4, calibration of the camera was done with five different objects. The objects were placed at a known distance and σ was computed for each object. The relation between object distance and step number is plotted in Table 1. The experiments were repeated by placing the objects at different distances. Since there are five objects we obtain five different σ values for each distance. Three σ values around the mean were averaged, to obtain one single calibration table. The calibration table is plotted in Figure 7.1.

After obtaining the calibration plots the distance estimation experiments were conducted on the five objects. The results are tabulated in Table 1, along with the mean and standard deviation. The mean of the standard deviations (overall RMS error) is about 2.07 steps out of about 90 steps. The mean values of the results are also plotted in Figure 7.2. The RMS error seems satisfactory for practical applications. As mentioned earlier, the accuracy can be slightly improved by combining with a DFF method.

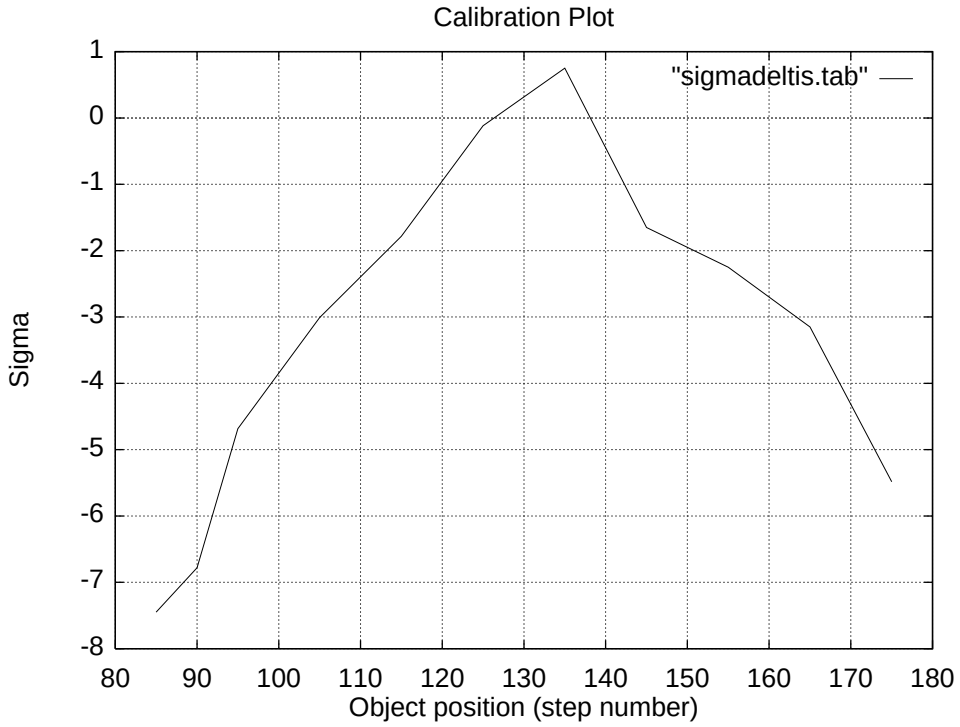


Figure 7-1: Calibration Plot

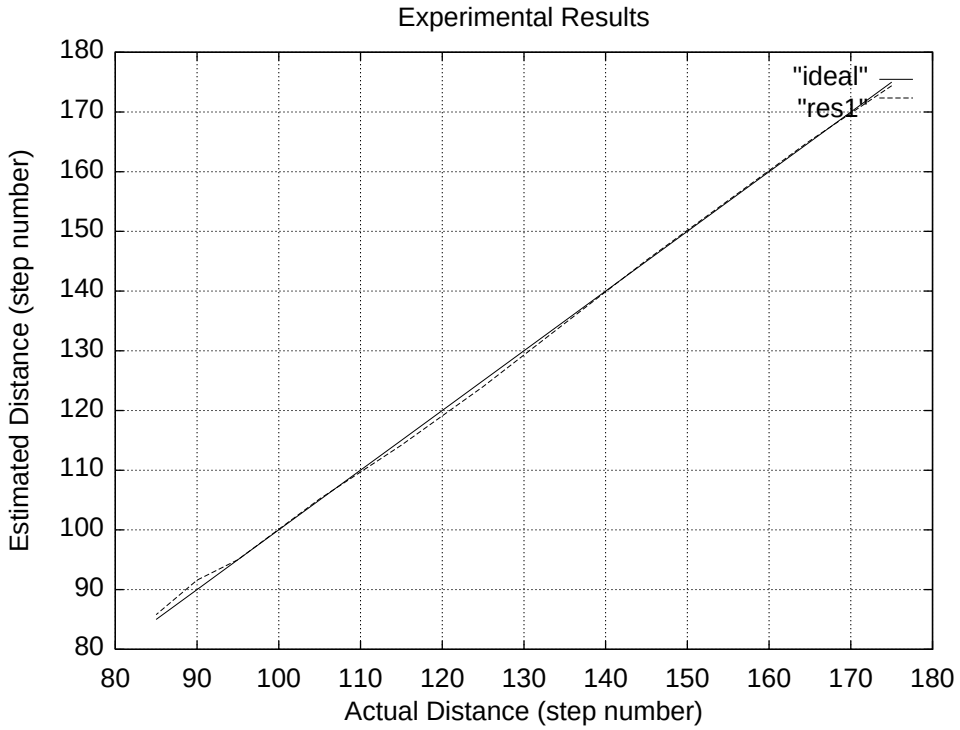


Figure 7-2: Experimental Results using STM

Actual Dist.		Estimated Dist. (step)						
Step	cm	tg	fa	mk	ch	gs	mean	std. dev.
175	19.2	175	169	176	180	172	174.39	3.72
165	21.9	163	166	161	172	164	165.19	3.76
155	25.2	154	155	150	161	156	155.19	3.54
145	29.8	145	143	144	147	147	145.2	1.6
135	36.1	135	133	136	134	135	134.6	1.01
125	46.0	125	125	124	124	122	124.0	1.09
115	60.2	114	114	116	113	114	114.19	0.98
105	91.0	105	104	111	103	103	105.19	2.99
95	184.0	94	94	94	94	99	95.0	2.0
90	331.7	90	91	94	91	92	91.6	1.36
85	530.0	85	85	87	86	86	85.8	0.75

Table 1. Experimental Results

7.3 Thresholds for Focus Measures

In Chapter 3, some of the commonly used focus measures were described. The most important among them are the gradient magnitude squared, laplacian squared and brightness variance. These focus measures have been proved to be sound focus measures in [119]. Depth-from-Focus (DFF) methods require a sequence of images taken with different lens positions. A focus measure is computed for each image and the maximum focus measure is obtained by searching among all the images. The focus measure is computed in a small window (typically 15 x 15 pixels). The computation can be repeated over many image windows and a 3-D depth map of the scene could be obtained. However, when a image window does not have enough contrast information, the focus measures are susceptible to noise leading to erroneous results. In this chapter, we investigate different focus measures and derive the corresponding noise thresholds. We also describe experimental results.

7.3.1 Gradient Magnitude Squared

If $g(i,j)$ is a discrete image, then a focus measure can be computed as

$$FM_1 = \frac{1}{N^2} \sum_{i,j=1}^{N} [g(i,j+1) - g(i,j)]^2 + [g(i+1,j) - g(i,j)]^2 \tag{7.2}$$

For the sake of analysis, let us consider the $g(i,j)$'s to be independent and identically distributed (iid) gaussian random variables with zero mean and standard deviation σ_N (i.e, N(0,σ_N)). Then the expected value of FM_1 is

given by

$$E\left[FM_1\right] = \frac{1}{N^2}\sum_{i,j=1}^{N} E\left[g(i,j+1)-g(i,j)\right]^2 + E\left[g(i+1,j)-g(i,j)\right]^2 \quad (7.3)$$

$$= 4\sigma_N^2. \quad (7.4)$$

The variance of FM_1 can be calculated as

$$\mathrm{Var}\left[FM_1\right] = E\left[FM_1^2\right] - E\left[FM_1\right]^2 \quad (7.5)$$

After simplification, it can be shown that for $N = 1$,

$$\mathrm{Var}\left[FM_1\right] = 20\sigma_N^4 \quad (7.6)$$

Since, we have defined the focus measure as a mean over an N x N region, the actual variance of FM_1 will be divided by a factor of N^2. Hence,

$$\mathrm{Var}\left[FM_1\right] = \frac{20}{N^2}\sigma_N^4 \quad (7.7)$$

In order to estimate the noise variance σ_N^2, we placed objects of constant brightness (white, black and gray posters) at different distances from the camera. The standard deviation of the brightness values was computed in many 15 x 15 non overlapping regions. The mean value of all the standard deviations was found to be 1.37. Hence we used $\sigma_N = 1.37$ in our experiments. Using these values in equations (7.4) and (7.6) the expected value of FM_1 is 7.5 and the variance of FM_1 is $\frac{70.455}{N^2}$. Hence the standard deviation of FM_1 is given by $\sigma_{FM1} = \frac{8.3937}{N}$. A threshold of mean + three times the standard deviation can be chosen for reliable results. The threshold can thus be calculated to be $7.5 + \frac{25.18}{N}$. In our experiments we used a window size of 15 x 15 and hence

Figure 7-3: Objects used for threshold calculations, po115 and pa150

the threshold is 9.18. It can be seen that, by using a larger window size, the threshold can be lowered resulting in more reliable depth estimates. However, a larger window size will reduce the resolution of the depth-map. Hence there is a trade off between window size for focus measure computation and the resolution of the depth map.

Figure 7.3 shows two objects used in our experiments. The object "po115" has high contrast information all over, but the object "pa150" has some regions with almost constant intensity. The size of each image is 150 x 150 pixels. The objects were placed at some known distance from the camera and 20 images were taken with different lens positions. The lens position was varied by 2 steps for every image. The focus measures (gradient magnitude squared) were computed in 15 x 15 regions in the images. For every image there are 10 x 10 = 100 such regions. The focus measures corresponding to some regions are

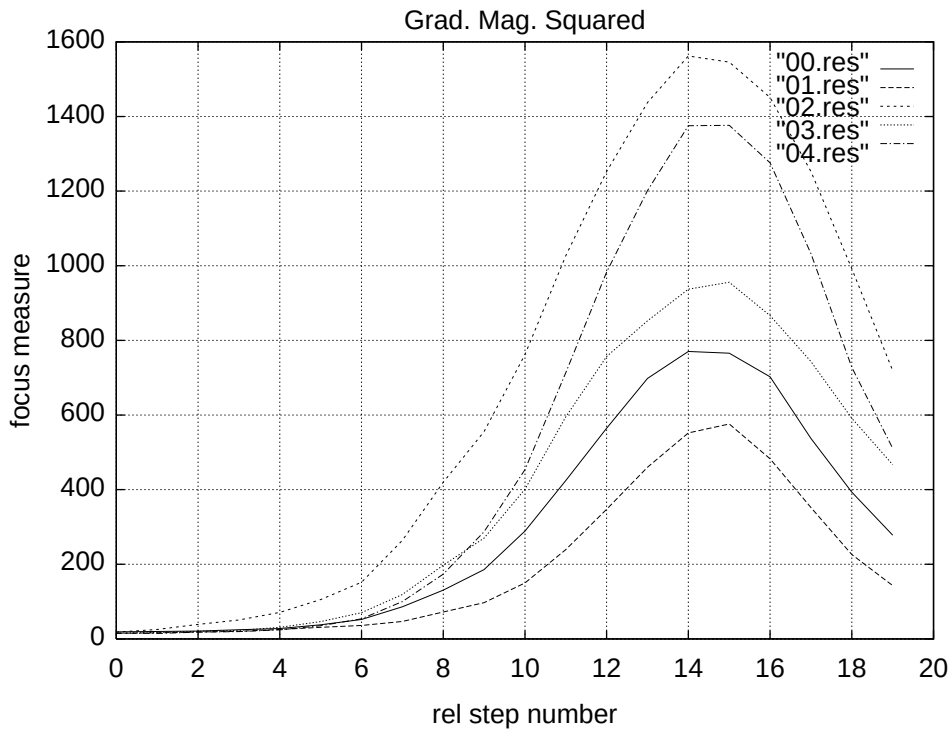


Figure 7-4: Focus Measures with Object po115

plotted in Figures 7.4 and 7.5. It can be seen in Figure 7.4 that all the focus measures are much above the calculated threshold of 9.18. However, in Figure 7.5 it is seen that some of the regions yield very low focus measures (almost constant and less than the threshold). These regions correspond to the almost constant intensity regions in the object "pa150".

7.3.2 Laplacian Squared

The square of the laplacian is a commonly used focus measure. For a discrete image $g(i,j)$ it can be computed as

$$FM_2 = \frac{1}{N^2} \sum_{i,j=1}^{N} \left(g(i+1,j) + g(i-1,j) + g(i,j+1) + g(i,j-1) - 4g(i,j)\right)^2 \tag{7.8}$$

Figure 7-5: Focus Measures with Object pa150

The expected value of FM_2 can be derived as

$$E(FM_2) = \frac{1}{N^2} \sum_{i,j=1}^{N} E\left[\left(g(i+1,j) + g(i-1,j) + g(i,j+1) + g(i,j-1) - 4g(i,j)\right)^2\right] \tag{7.9}$$

As before, if we consider the $g(i,j)$'s to be independent gaussian random variables (i.e., $N(0,\sigma_N)$), it can be shown that

$$E(FM_2) = 20\sigma_N^2 \tag{7.10}$$

Using a similar approach as in the previous case, it can be shown that for N = 1,

$$\mathrm{Var}\left[FM_2\right] = 800\sigma_N^4 \tag{7.11}$$

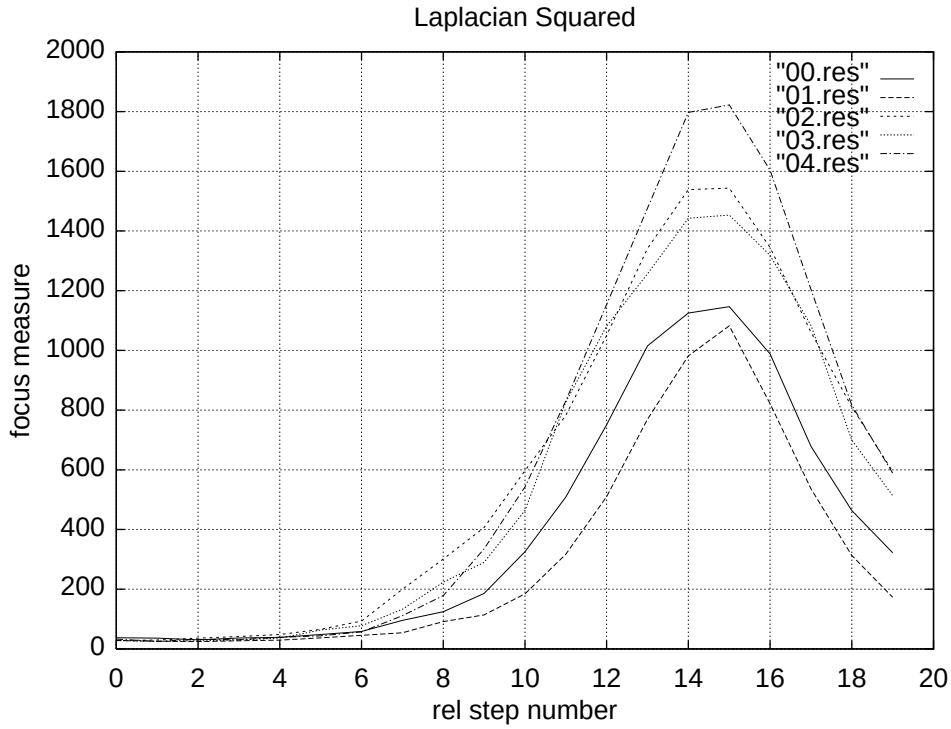


Figure 7-6: Focus Measures with Object po115

and in general for an N x N region,

$$\mathrm{Var}\,[FM_2] = \frac{800}{N^2}\sigma_N^4. \tag{7.12}$$

The standard deviation σ_{FM2} then becomes

$$\sigma_{FM2} = \frac{28.28\sigma_N^2}{N} \tag{7.13}$$

A threshold of mean + three times σ_{FM2} could be chosen as the threshold for the focus measure. For a window size of 15 x 15, the threshold can be calculated to be 37.53 + 10.62 = 48.15. The focus measures were again computed for the objects shown in Figure 7.3. The focus measures corresponding to some regions are plotted in Figures 7.6 and 7.7. It can be seen in Figure 7.6 that all the focus measures are much above the calculated threshold of

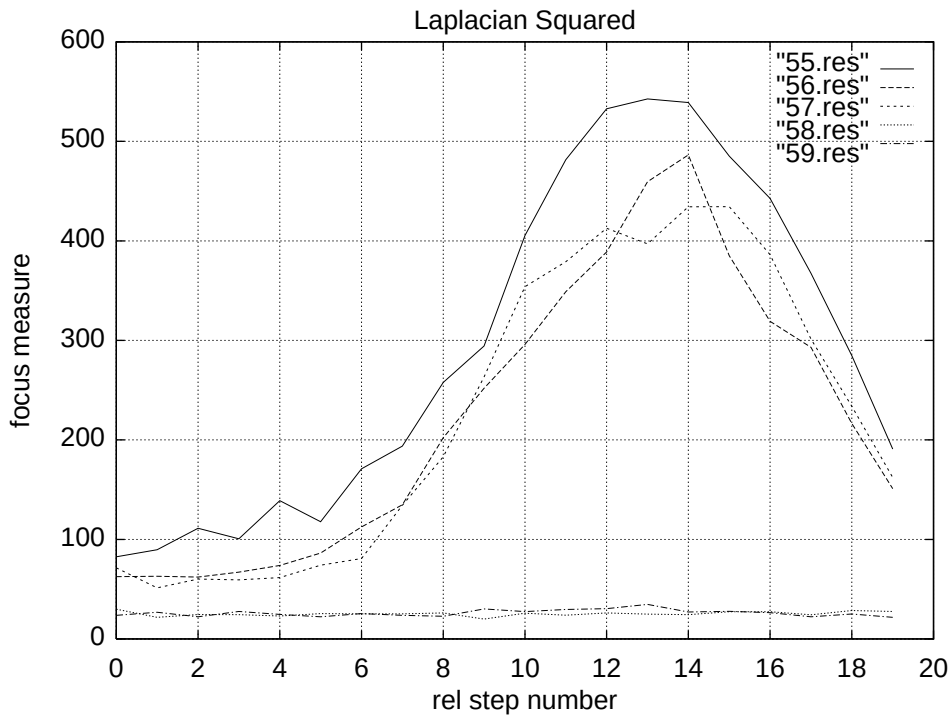


Figure 7-7: Focus Measures with Object pa150

48.15. However, in Figure 7.7 it is seen that some of the regions yield very low focus measures (almost constant and less than the threshold). These regions correspond to the almost constant intensity regions in the object "pa150".

7.3.3 Brightness Variance

A focus measure based on brightness variance can be defined as

$$FM_3 = \frac{1}{N^2} \sum_{i,j=1}^{N} g^2(i,j) \tag{7.14}$$

Assuming $g(i,j)$'s to be gaussian random variables as before, the expected value of FM_3 can be shown to be

$$E\left[FM_3\right] = \sigma_N^2 \tag{7.15}$$

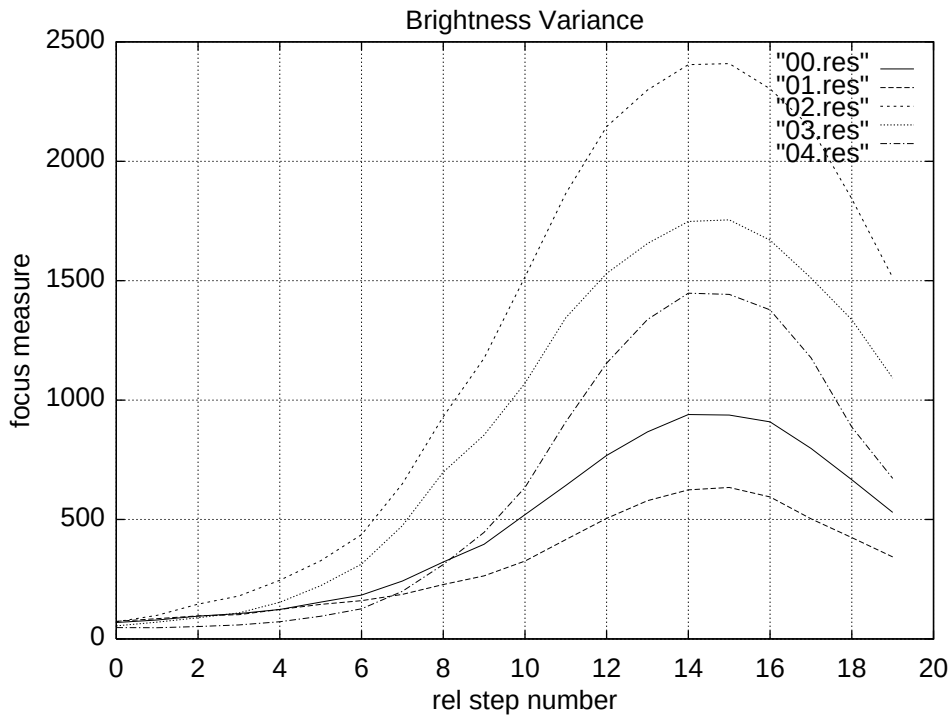


Figure 7-8: Focus Measures with Object po115

and the variance of FM_3 can be shown to be

$$\mathrm{Var}\,[FM_3] = \frac{3}{N^2}\sigma_N^4. \tag{7.16}$$

Calculating the mean + three times the standard deviation, gives 2.5269 as the threshold for a window size of 15 x 15. Some of the values of FM_3 computed on the objects in Figure 7.3 are plotted in Figures 7.8 and 7.9. The plots look identical to the ones obtained with the other two focus measures.

7.4 3-D Shape Recovery

In the previous section, three different focus measures were discussed and their thresholds were derived. In this section we describe experiments where

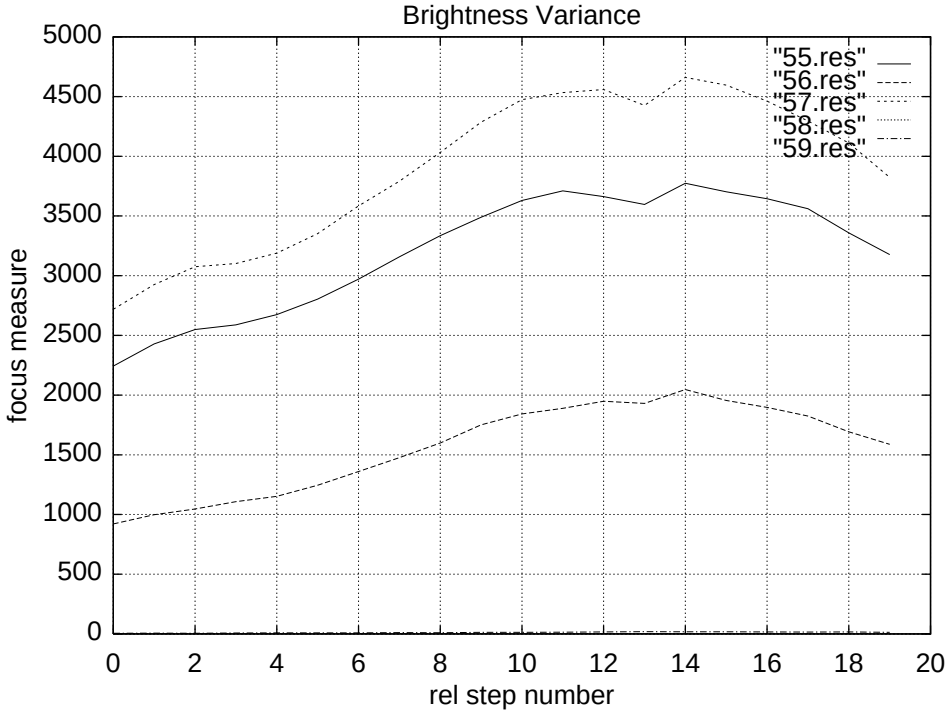


Figure 7-9: Focus Measures with Object pa150

the 3-D shape of an object is recovered using a thresholded focus measure.

7.4.1 Inclined Plane

A poster with characters was placed inclined to the optical axis. The nearest end of the poster was about 20 cms from the camera and the farthest end was about 45 cms. About 10 images of the object were taken with lens position starting at step 175 and decrementing by 6 steps everytime. Two of the images taken with lens steps 121 and 175 are shown in Figure 7.10 (a) and (b).

It can be seen that the right side of Figure 7.10 (a) is focused and the left side is blurred. In Figure 7.10 (b) the left side is focused and the right side is

Figure 7-10: (a) and (b) Images of Character Poster at lens steps 121 and 175

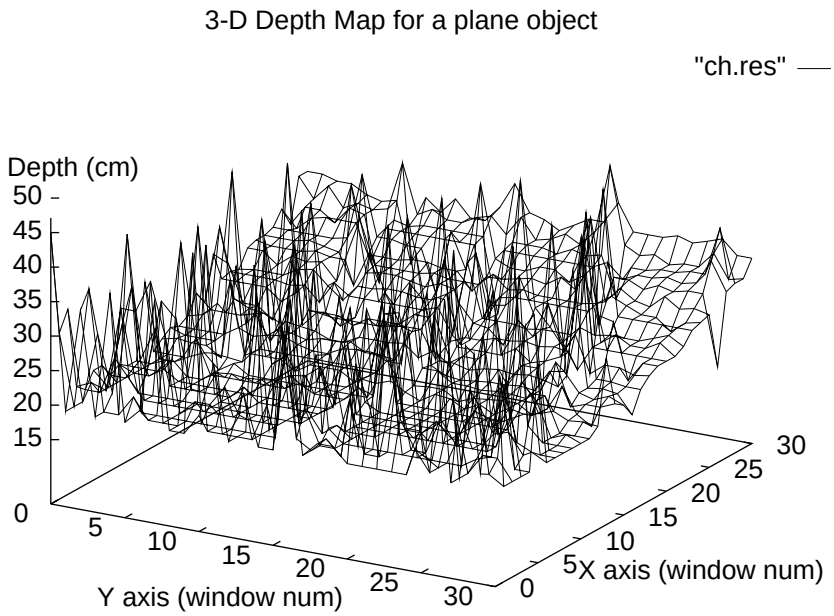


Figure 7-11: Depth Map of Inclined Plane without Thresholding

highly blurred. The images were divided into many non-overlapping regions of size 15 x 15 pixels. The focus measure (laplacian squared) was computed over all the 10 images and the maximum was found for each window. A quadratic interpolation scheme was used with three points around the maximum to obtain the location of the actual peak of the focus measure. The depth map obtained without using any threshold on the focus measure is shown in Figure 7.11. As there are many regions with almost constant intensity in the poster, the depth map looks quite noisy. By using a threshold on the focus measure, most of the noise in the depth map could be eliminated as shown in Figure 7.12.

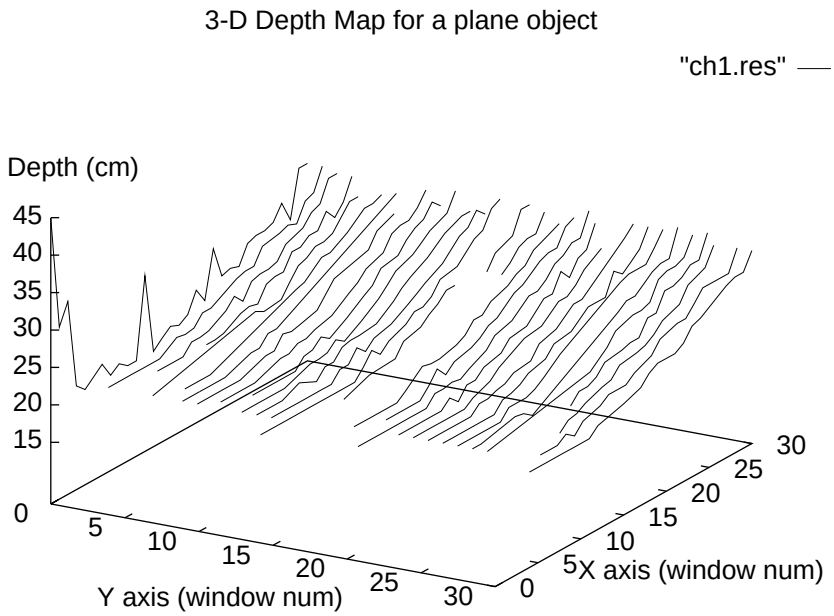


Figure 7-12: Depth Map of Inclined Plane with Thresholding

7.4.2 Cone Object

A cone object was made with a diameter of about 25 cms and height of about 60 cms. A grid pattern was drawn all over the cone. The cone was placed with its axis parallel to the optical axis of the camera and the tip of the cone being about 20 cms from the camera. As in the previous case, 10 images of the cone were obtained and processed. Two of the images corresponding to lens steps 121 and 175 are shown in Figure 7.13 (a) and (b). It can be seen that the center region of Figure 7.13 (a) is highly blurred and the borders of the image are focused. The situation is reversed in Figure 7.13 (b). The resulting depth map is shown in Figure 7.14 and the reconstructed (focused) image is shown in Figure 7.15.

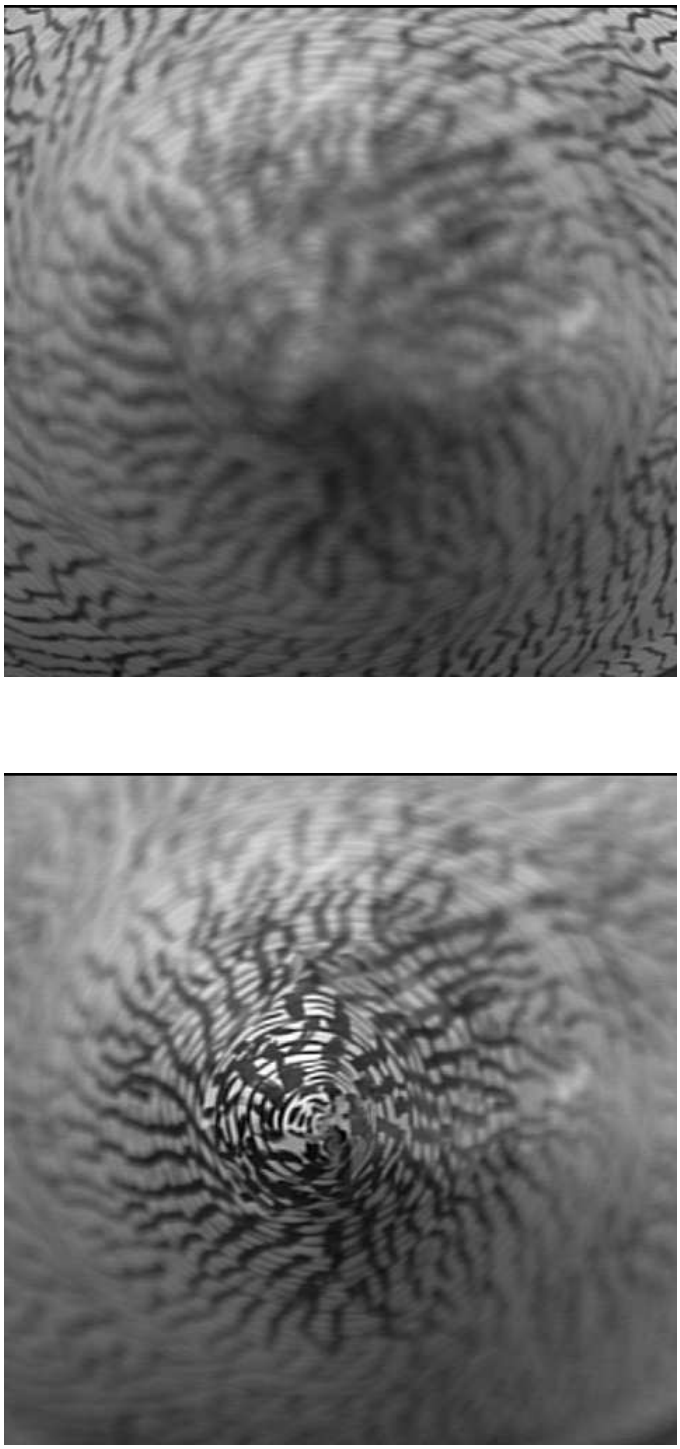


Figure 7-13: (a) and (b) Images of a cone at lens steps 121 and 175

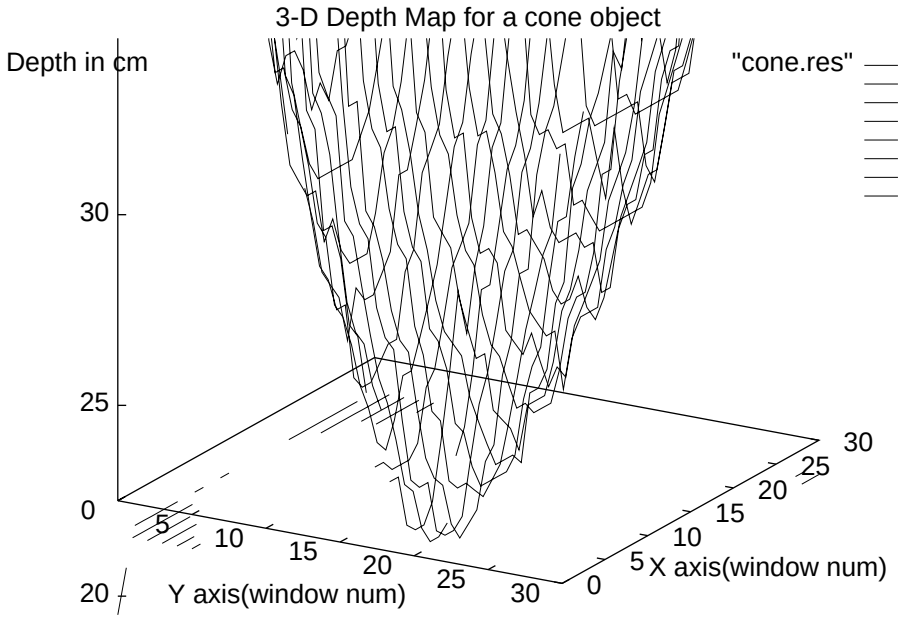


Figure 7-14: Depth Map of a Cone Object

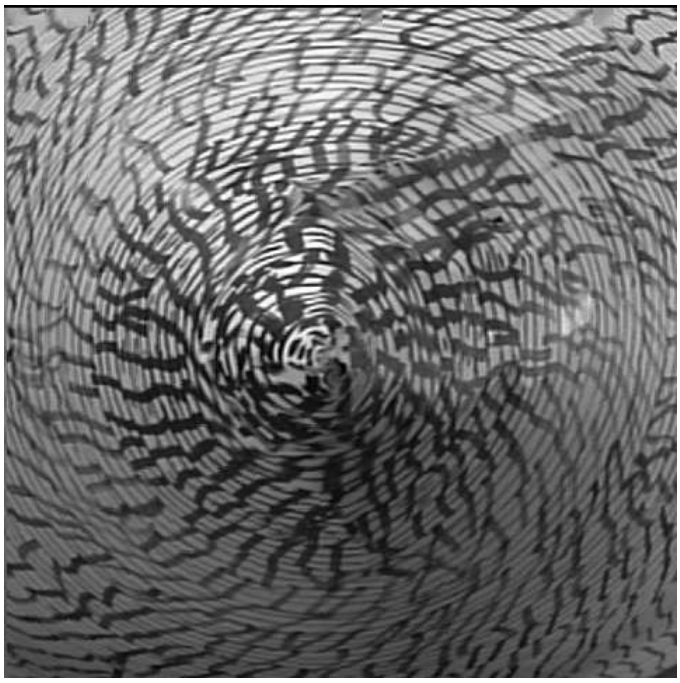


Figure 7-15: Reconstructed Image of the cone

7.4.3 Microscopic Objects

The Depth-from-Focus algorithm was also implemented on a Olympus BH-2 Microscope to recover the shape and focused image of a microscopic object. In the case of a microscope, the different images are obtained by moving the object itself (stage), instead of moving the lens. Ten images of a mustard seed were obtained by moving the object (stage) by 15 microns everytime. Two of the images that were taken 105 microns apart are shown in Figure 7.16 (a) and (b). It can be seen that in Figure 7.16 (a) the borders of the image are focused and in Figure 7.16 (b) the center of the image is focused. The depth map obtained with this object is shown in Figure 7.17 and the reconstructed (focused) image is shown in Figure 7.18.

7.5 Conclusion

In this chapter we first described the implementation of STM on a DELTIS camera, which is an off-the-shelf camera. No lens data was available for the DELTIS camera. However, we successfully implemented STM on the camera and in a large number of experiments we obtained an RMS error of about 2.5% in lens position. The accuracy of the method seems to be quite good for practical applications such as autofocusing video cameras.

Next, we derived some thresholds for three commonly used focus measures. These thresholds were derived based on the assumption that the noise at different pixels are independent and identically distributed gaussian random variables. We have shown experimentals results to confirm our derivations. We

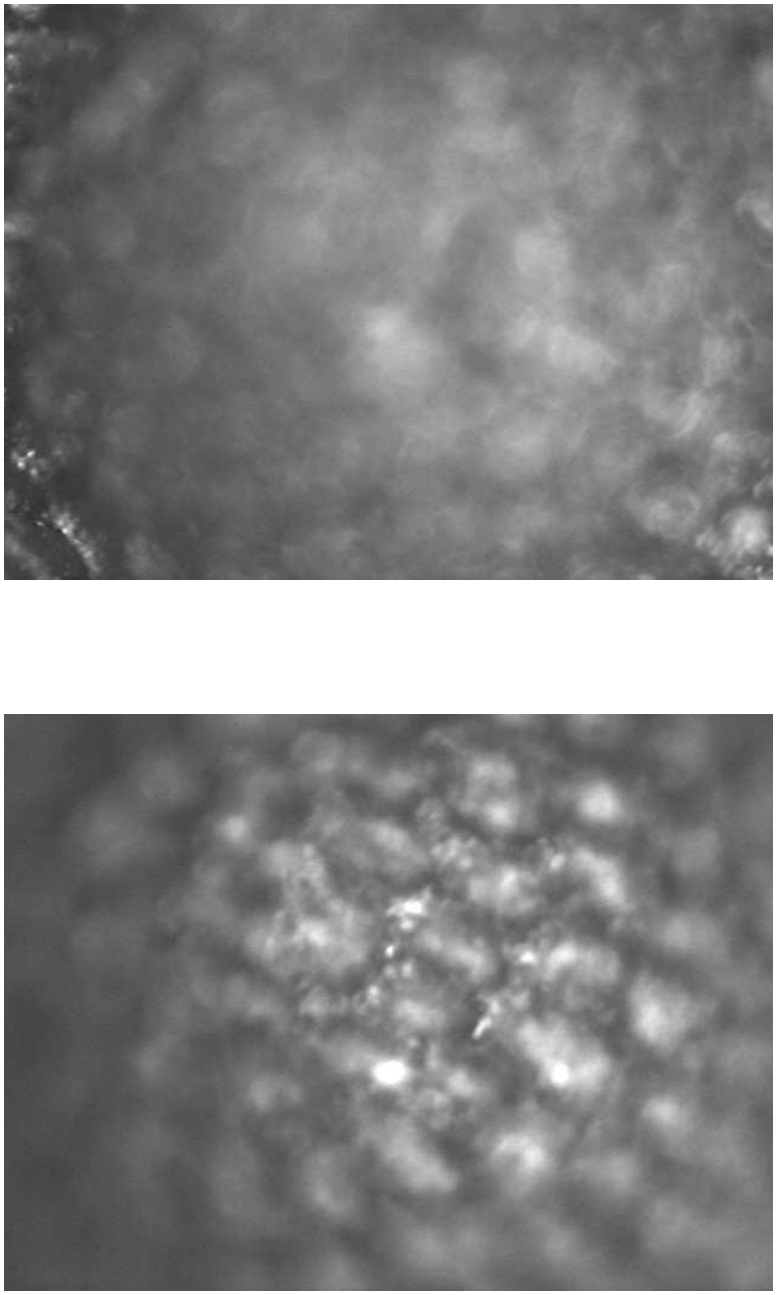


Figure 7-16: (a) and (b) Images of a mustard seed taken 105 microns apart

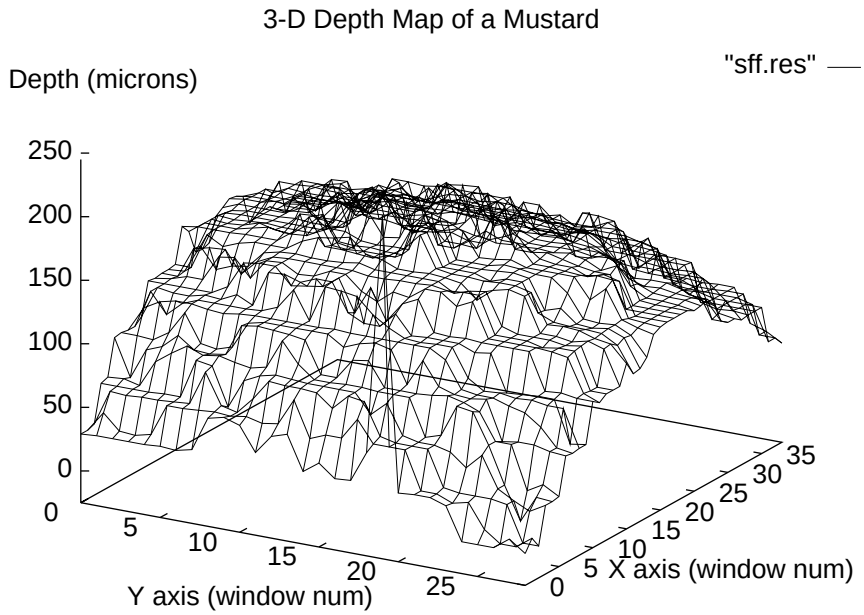


Figure 7-17: Depth Map of a Mustard Seed

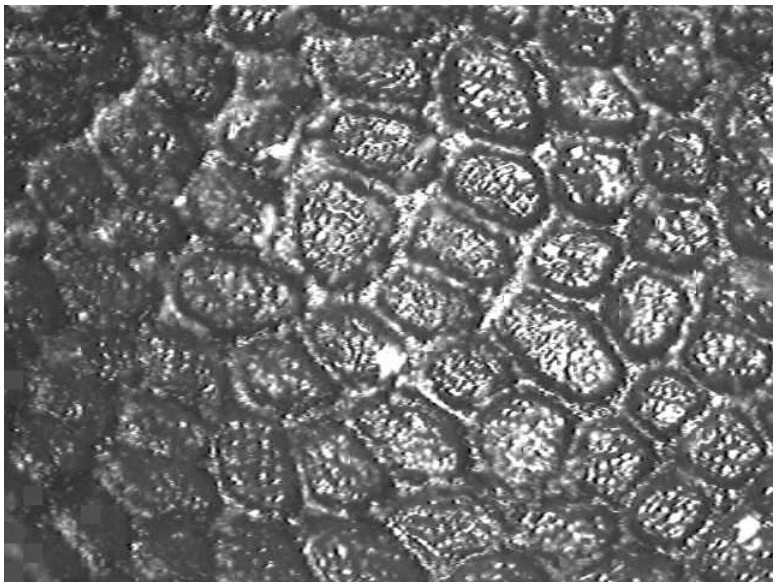


Figure 7-18: Reconstructed Image of the Mustard Seed

have also implemented an algorithm to recover shapes of 3-D objects, using a Depth-from-Focus (DFF) scheme with interpolation. In order to demonstrate the utility of the algorithm, experimental results have been provided both for a camera system and a microscope system.

Chapter 8

Conclusion

8.1 Summary

In this dissertation we have investigated focus as a cue for 3-D depth recovery. Traditionally, Depth-from-Focus (DFF) has been a search method requiring the acquisition of a large number of images with different camera parameters. By searching for that set of camera parameters which gives rise to the most focused image, depth information of a scene can be recovered. DFF has the major advantages that it does not suffer from the *correspondence problem* and the *occlusion problem*. Also, it involves much less computation than stereo. However, DFF has the disadvantage that a long time is required to acquire many images. This is due to the fact that the acquisition of each image involves mechanical motion of the lens system and the scene has to remain stationary during the entire process.

In order to circumvent these problems of DFF, we have investigated a method of recovering depth directly by measuring the amount of blur in an

image. This approach is known as Depth-from-Defocus (DFD) and does not involve any search. The method is based on a new Spatial Domain Convolution/ Deconvolution Transform and is known as the S-Transform Method (STM). STM requires only two images acquired with two different camera parameters to obtain the depth-map of the entire scene. It has been implemented on an actual camera system known as SPARCS and a large number of experiments have yielded an RMS error of 2.3 % in an autofocusing application.

STM is applicable to arbitrary scenes and does not make any assumptions about the underlying focused images, such as the existence of step edges. Since all the computations are done in the spatial domain and are local in nature, STM can be implemented in parallel and the depth-map obtained can be much denser than what can be obtained with Fourier domain methods. We have investigated two variations of STM, one where the two images are acquired with two different lens positions and another where the two images are acquired with different camera apertures. The accuracies of both these methods are about the same.

We have extended STM to continuously focus on moving objects. If the objects in a scene are slowly moving, as in the case of a video camera, it is not possible to acquire two images sequentially. Also it may not be feasible to acquire the two images always from the same two fixed lens positions. With these problems in view, we have extended STM to recover depth from two images taken with any two lens positions. Also, we have suggested a simple camera architecture for the simultaneous acquisition of two images. A large number of experiments have demonstrated the utility of the method with an

accuracy comparable to that obtained with stationary objects.

We have shown that the focused image of an object can be recovered using two defocused images recorded with different camera parameter settings. The same two images can be used to estimate the depth of the object using STM. For a 3-D scene where the depth variation is small in image regions of size about 64×64 pixels, each image region can be processed separately and the results can be combined to obtain both a focused image of the entire scene and a rough depth-map of the scene.

If, in each image region, at least one of the two recorded defocused images is blurred only moderately or less ($\sigma \leq 3.5$ pixels), then the focused image can be recovered very fast (computational complexity of $O(N^2)$ for an $N \times N$ image) using the spatial domain deconvolution method. In most practical applications of machine vision, the camera parameter setting can be arranged so that this condition holds, i.e. in each image region at most only one of the two recorded defocused images is severely blurred ($\sigma > 3.5$ pixels). In those cases where this condition does not hold, the second method which uses the Inverse Abel Transform can be used to recover the focused image. This method requires camera calibration for the PSF and is several times more computationally intensive than the first method above. The methods of focused image recovery that we have investigated can be used as part of a 3-D machine vision system to obtain focused images from blurred images for further processing such as edge detection, stereo matching, and image segmentation.

8.2 Future Research

The fact that DFF is more accurate than DFD while requiring more number of images to be acquired and processed suggests that by effectively combining the two techniques, we can gain the advantages of both speed and accuracy. DFD can first be used to obtain a rough estimate of distance and by acquiring a few more images and searching in the vicinity using DFF techniques, it should be possible to obtain accurate depth estimates. The details of such a scheme for the case of 3-D objects, however, needs further investigation.

Another avenue for future research is the active integration of focus and stereo. We have shown that for a typical configuration, the accuracy of stereo can be about one order higher than that of focus. Some preliminary work in this direction has been done by Krotkov [67] and Abbott [1]. However, a more systematic study of the relative strengths and weaknesses of the two schemes needs to be carried out for the development of robust vision systems.

Parallel hardware and implementation of DFF and DFD algorithms needs to be studied for real-time applications in machine vision, autonomous navigation of vehicles etc. The image overlap problem which limits the accuracy of STM at high levels of blur, if examined in more detail could perhaps lead to the improvement in accuracy of STM. The theoretical properties of the S-Transform may be studied in more detail.

Appendix A

S Transform

In this appendix, we state the Forward and Inverse S Transform expressions for one- and two-dimensional signals. The derivations may be found in [113].

A.1 Forward S Transform

Let P^{N+1} be the space of all real valued polynomial functions of degree less than or equal to N for $N = 0, 1, 2, \cdots$. Let $f \in P^{N+1}$. Then we can write

$$f(t) = \sum_{k=0}^{N} a_k t^k \tag{A.1}$$

where a_k are real valued coefficients and t is a real variable.

Let the nth moment of a real valued function $h(t)$ be denoted by h_n for $n = 0, 1, 2, \cdots$. We have

$$h_n \equiv \int_{-\infty}^{\infty} t^n h(t)\, dt. \tag{A.2}$$

Let M^{N+1} be the space of all real valued functions $h(t)$ such that $h_0 \neq 0$, and all positive integer moments of h upto order N are finite, i.e.,

$$|h_n| < \infty \text{ for } n = 0, 1, 2, \cdots N. \tag{A.3}$$

Definition 1: The *S Transform* of a polynomial $f(t)$ where $f \in P^{N+1}$ with respect to a kernel $h(t)$ where $h \in M^{N+1}$ is denoted by $S_H\{f\}(\tau)$ or $F_H(\tau)$ and is defined by the integral

$$S_H\{f\}(\tau) \equiv F_H(\tau) \equiv \int_{-\infty}^{\infty} h(t)\ f(\tau - t)\ dt, \tag{A.4}$$

where τ is the transform variable and $\tau \in R$ (i.e., τ is real).

The integral in Eq. (A.4) above can be recognized as the *convolution* integral. The S Transform of $f(t)$ where $f \in P^{N+1}$ with respect to a kernel function $h(t)$ where $h \in M^{N+1}$, denoted by $S_H\{f\}(\tau)$ or $F_H(\tau)$ can also be expressed as

$$S_H\{f\}(\tau) \equiv F_H(\tau) \equiv \sum_{k=0}^{N} \frac{(-1)^k}{k!} h_k f^k(\tau). \tag{A.5}$$

A.2 Inverse S Transform

By repeated differentiation and some algebraic manupulations, it is possible to invert the above convolution process and obtain $f(\tau)$ as

$$f(\tau) = \sum_{k=0}^{N} w_k F_H^k(\tau) \tag{A.6}$$

$$\text{where } w_0 = \frac{1}{h_0} \quad \text{and} \tag{A.7}$$

$$w_k = \sum_{p=1}^{k} \sum_{|I_p|=k} \frac{(-1)^{p+k}}{h_0^{p+1}} \prod_{q=1}^{p} \frac{h_{i_q}}{i_q!} \tag{A.8}$$

$$i_q \in \{1, 2, 3, \cdots, k\} \text{ for } q = 1, 2, 3, \cdots, k. \tag{A.9}$$

The right hand side of Eq. (A.8) should be interpreted as summing over all possible permutations of $i_1, i_2, \cdots, i_p$ (for $p = 1, 2, 3, \cdots, k$) such that they sum to k (this condition is specified by $|I_p| = k$). This corresponds to the different ways in which the integer k can be expressed as the summation of positive integers $1, 2, \cdots, k$.

A.3 Two-Dimensional Case

S Transform and Inverse S Transform can be extended to two-dimensional, and in general n-dimensional case. Here we only summarize the results.

First we introduce some notation, and then present the results.

P_2^{N+1}: Space of all real valued two-dimensional polynomial functions of degree less than or equal to N for $N = 0, 1, 2, 3, \cdots$.

$x, y \in R$.

For any function $f(x, y)$,

$$f^{m,n} \equiv \frac{\partial^m}{\partial x^m} \frac{\partial^n}{\partial y^n} f(x, y). \tag{A.10}$$

Two-dimensional N-th order polynomial:

$$f(x, y) = \sum_{m=0}^{N} \sum_{n=0}^{N-m} a_{m,n} x^m y^n \tag{A.11}$$

$$= \sum_{0 \le m+n \le N} a_{m,n} x^m y^n. \tag{A.12}$$

If $f \in P_2^{N+1}$, then

$$f^{m,n} = 0 \quad \text{for} \quad m + n > N. \tag{A.13}$$

Two-dimensional moments:

$$h_{m,n} \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^m \, y^n \, h(x,y) \, dx \, dy \tag{A.14}$$

for $m, n \;=\; 0, 1, 2, 3, \cdots$.

M_2^{N+1}: the space of all real valued functions $h(x,y)$ such that $h_{0,0} \neq 0$ and all moments of h upto order N are finite, i.e.,

$$|h_{m,n}| < \infty \quad \text{for} \quad m+n = 0, 1, 2, \cdots, N. \tag{A.15}$$

Two-dimensional convolution:

$$F_H(x,y) \;=\; h(x,y) \;*\; f(x,y) \tag{A.16}$$

$$= \sum_{0 \leq m+n \leq N} \frac{(-1)^{m+n}}{m!n!} f^{m,n} h_{m,n}. \tag{A.17}$$

Two-dimensional deconvolution:

$$f \;=\; \sum_{k=0}^{N} \sum_{l=0}^{k} w_{k-l,l} \; F_H^{k-l,l} \;=\; \sum_{k=0}^{N} \sum_{0 \leq i+j \leq k} w_{i,j} \; F_H^{i,j} \tag{A.18}$$

$$\text{where } w_{i,j} \;=\; \sum \frac{(-1)^{p+i+j}}{h_{0,0}^{p+1}} \prod_{q=1}^{p} \left(\frac{h_{m_q,n_q}}{m_q! n_q!} \right). \tag{A.19}$$

In the above equation, the summation is done over all possible m_q, n_q, p, for $q = 1, 2, 3, \cdots, p$, subject to the conditions

$$\left[\begin{array}{l} m_1 + m_2 + \cdots + m_p = i \\ \\ n_1 + n_2 + \cdots + n_p = j \\ \\ m_1 + n_1 \geq 1, m_2 + n_2 \geq 1, \cdots, m_p + n_p \geq 1 \end{array} \right]. \tag{A.20}$$

Appendix B

Lens Data and Image Database

This appendix describes the camera data and the image database related to SPARCS (StonyBrook Passive Autofocusing and Ranging Camera System). The image database (SPARCS.DB1) consists of many images taken with different lens positions at different illuminations. These are explained in the following sections.

B.1 Lens Data

The following quantities are defined with respect to Figure B.1.

SI : Distance from Second Principle Plane to Image Detector

r0 : Distance from Object to First Principle Plane

1) SI = PK - HB where PK = 38.95 mm (fixed) HB is obtained from Table B.1.

2) FL = FB - HB where FL, FB and HB are found from Table B.1.

3) Find r0 using lens rule: 1/FL = 1/SI + 1/r0

4) D0 comes from lens data as best focus position.

B.2 Image Database

The image database has a number of images taken with many different lens positions at many different illuminations. The image detector position can be varied from step 0 to step 96 as shown in Figure B.2. The camera constants for any step number are shown in Table B.1. Here we will give the naming conventions used for different databases. All the images are stored in binary row major format and are of size 128 x 128 bytes unless otherwise indicated.

1) Directory: /local2/database/F4Lux200.f35

This image database contains images taken with illumination of 200 Lux. The F Number is 4 and focal length is 35 mm.

There are 5 sub directories namely:

1) 3d_obj

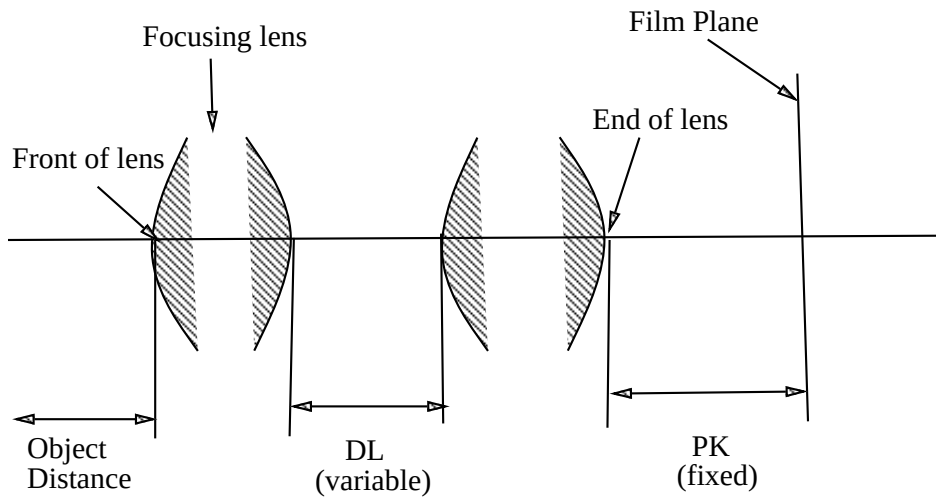
2) edge

3) face

4) grad_sch

5) tiger

In each directory filenames indicate the object position in step number and also the lens position in step number. For example, in the sub directory "edge", there are files like ed2510.img, ed2540.img and ed2570.img. These



DL : 15.677 mm to 21.977 mm

PK : 38.950 mm

Figure B.1

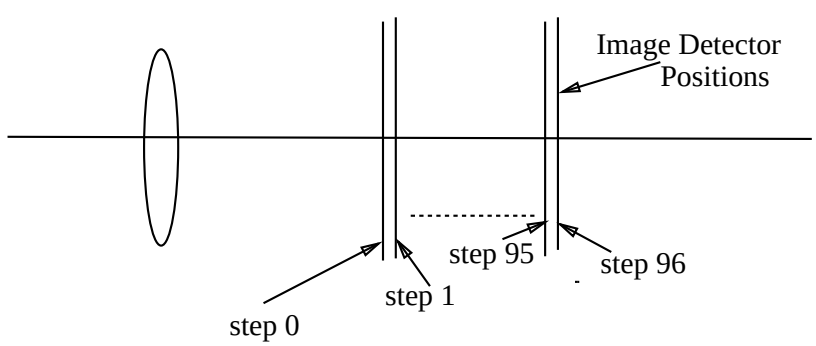


Figure B.2

Step	dx	FB	HB	FL	SI	r0	D0
0	0.0000	39.005	2.770	36.235	36.180	-23835.9	9034.0
5	0.3140	38.905	2.819	36.086	36.132	28973.4	5300.0
10	0.6280	38.805	2.867	35.938	36.083	8943.1	3750.0
15	0.9420	38.707	2.914	35.792	36.035	5296.9	2850.0
20	1.2560	38.608	2.962	35.646	35.988	3751.0	2500.0
25	1.5700	38.511	3.009	35.502	35.941	2906.6	1930.0
30	1.8840	38.414	3.056	35.359	35.894	2372.3	1720.0
35	2.1980	38.319	3.102	35.217	35.848	2000.7	1464.0
40	2.5120	38.224	3.148	35.076	35.802	1729.7	1320.0
45	2.8260	38.130	3.193	34.937	35.757	1523.5	1170.0
50	3.1400	38.036	3.238	34.798	35.712	1359.6	1080.0
55	3.4540	37.944	3.283	34.661	35.667	1228.9	965.0
60	3.7680	37.852	3.328	34.524	35.622	1120.0	900.0
65	4.0820	37.761	3.372	34.389	35.578	1029.0	822.0
70	4.3960	37.670	3.416	34.255	35.534	951.7	770.0
75	4.7100	37.580	3.459	34.121	35.491	884.3	715.0
80	5.0240	37.491	3.502	33.989	35.448	825.8	670.0
85	5.3380	37.403	3.544	33.859	35.406	774.9	628.0
90	5.6520	37.315	3.587	33.728	35.363	729.5	595.0
95	5.9660	37.080	3.700	33.380	35.250	629.3	560.0

Table B.1. Lens Data

three images were taken with an edge object placed at step 25. The 3 images were taken with lens positions 10, 40 and 70 respectively. The object position varies from step 10 to step 95, in steps of 5.

2) Directory: /local2/database/F4Lux400.f35

This image database contains images taken with illumination of 400 Lux. The F Number is 4 and focal length is 35 mm.

There are 11 sub directories namely:

(1) c1 (2) c2 (3) ev (4) fa (5) ft (6) gl (7) gs (8) mk (9) mn (10) sb (11) tg

In each directory filenames indicate the object position in step number and also the lens losition in step number. For example, in the sub directory "tg", there are files like tg3510f4.400, tg3540f4.400, tg3570f4.400. These three images were taken with a tiger object placed at step 35. The 3 images were taken with lens positions 10, 40 and 70 respectively. The object position varies from step 10 to step 95, in steps of 5. The objects used for this database are shown in Figure 4.10.

3) Directory: /local2/database/F8Lux400.f35

This image database contains images taken with illumination of 400 Lux. The F Number is 8 and focal length is 35 mm.

There are 11 sub directories namely:

(1) c1 (2) c2 (3) ev (4) fa (5) ft (6) gl (7) gs (8) mk (9) mn (10) sb (11) tg

In each directory filenames indicate the object position in step number and also the lens position in step number. For example, in the sub directory "tg", there are files like tg3510f8.400, tg3540f8.400, tg3570f8.400. These three images were taken with a tiger object placed at step 35. The 3 images were

taken with lens positions 10, 40 and 70 respectively. The object position varies from step 10 to step 95, in steps of 5.

B.3 3-D Cone Object Database

A cone object was made by Mr. Tae Choi which is 2 metres long and has a black and white stripe pattern all over it. It was placed in front of the camera with the axis of the cone approximately coinciding with the optical axis of the camera. The lens position was varied from step 0 to step 96 in steps of 1. The images thus obtained have been saved as a database for research on 3-D shape recovery. All the files are of size 360 x 360 and are stored in the binary row major format.

Directory : /local2/database/cone360

The image filenames have the format "cone360s[stepno]" where stepno varies from 0 to 96. For example, cone360s7 is the image obtained with lens at step 7 and cone360s35 is the image obtained with lens at step 35.

Bibliography

[1] Abbott, A.L., and Ahuja, N., "Surface Reconstruction by Dynamic Integration of Focus, Camera Vergence and Stereo", *Second Intl. Conf. Computer Vision*, IEEE Computer Society, pp. 532-543 (Dec. 1988).

[2] Allen, P.K., and Bajcsy, R., "Two Sensors are Better Than One: Example of Integration of Vision and Touch", *Technical Report MS-CIS-85-29*, University of Pennsylvania (1985).

[3] Aloimonos, J., "Active Vision", *Proc. ARPA Image Understanding Workshop*, Los Angeles, pp. 552-573 (Feb. 1987).

[4] Alpern, M., "Eye Movements", *Handbook of Sensory Physiology*, ed. D. Jameson and l. Hurvich, Springer-Verlag, New York (1972).

[5] Ayache, N,. and Faugeras, O.D., "Building a Consistent 3D Representation of a Mobile Robot Environment by Combining Multiple Stereo Views", *Proc. Intl. Joint Conf. Artificial Intelligence*, Milan, Italy, pp. 808-810 (Aug. 1987).

[6] Baird, J.W., "The influence of Accomodation and Convergence Upon the Perception of Depth", *Amer. Jr. Psychology 14*, pp. 150-200 (1903).

[7] Bajscy, R., and Allen, P.K., "Sensing Strategies", *Proc. United States-France Robotics Workshop*, University of Pennsylvania (Nov 1984).

[8] Ballard, D., and Brown, C., "Computer Vision", *Prentice-Hall*, New Jersey (1982).

[9] Bassman, H., and Besslich, P.W., "Monocular Computer Vision", *Third Intl. Conf. on Image Processing and its Applications*, pp. 107-111 (July 1989).

[10] Bertero, M., and Poggio, T.A., "Ill-posed Problems in Early Vision", *Proc. IEEE*, 76(8), pp. 869-889 (August 1988).

[11] Besl, J.P., "Active Optical Range Imaging Sensors", *Machine Vision and Applications*, 1(2), pp. 127-152 (1988).

[12] Besl, J.P., "Active Optical Range Imaging Sensors", *Advances in Machine Vision*, Editor Sanz, J.L.C., Springer Verlag, New York, pp. 1-63 (1989).

[13] Besl, J.P., "Range Imaging Sensors", *Research Publication GMR-6090*, Warren, Michigan (March 1988).

[14] Born, M., and Wolf, E., *Principles of Optics*, Pergamon Press, New York (1980).

[15] Bove, V.M. Jr., "Discrete Fourier Transform Based Depth-from-Focus", *Image Understanding and Machine Vision, Technical Digest Series*, Optical Society of America and Air Force Office of Scientific Research (June 1989).

[16] Bova, B., *The Beauty of Light*, John Wiley and Sons, Inc., (1988).

[17] Bracho, R., Schlag, J.F., and Sanderson, A.C., "POPEYE: A Gray-Level Vision System for Robotics Applications", *Tech. Report CMU-RI-TR-83-6*, Carnegie Mellon University (May 1983).

[18] Brooks, R.A., "Visual Map Making for a Mobile Robot", *Proc. IEEE Conf. Robotics and Automation*, St. Louis, pp. 824-829 (March 1985).

[19] Brooks, R.A., "A Robust Layered Control System for a Mobile Robot", *IEEE J. Robotics and Automation*, RA - 2(1), pp. 14-23 (March 1986).

[20] Castore, G., and Crawford, C., "From Solid Model to Robot Vision", *Proc. IEEE Conf. Robotics and Automation*, Atlanta (March 1984).

[21] Campbell, F.W., "Correlation of Accomodation between the two Eyes", *Jour. Opt. Soc. America*, 50(7), pp. 738 (July 1960).

[22] Campbell, F.W., and Westheimer, G., "Dynamics of Accomodation Responses of the Human Eye", *Jour. Physiology*, 151(2), pp. 285-295 (May 1960).

[23] Clark, J.J., and Ferrier, N.J., "Modal Control of an Attentive Vision System", *Second Intl. Conf. Computer Vision*, IEEE Computer Society, pp. 532-543 (Dec. 1988).

[24] Connoly, C.I., "The Determination of Next Best Views", *Proc. IEEE Conf. Robotics and Automation*, St. Louis, pp. 432-435 (March 1985).

[25] Corle, T.R., Fanton, J.T., and Kino, J.S., "Distance Measurements by Differential Confocal Optical Ranging", *Applied Optics*, 26(12), pp. 2416-2420 (June 1987).

[26] Cornsweet, T.N., *Visual Perception*, Academic Press, New York (1970).

[27] Crane, H.D., "A Theoretical Analysis of the Visual Accomodation System in Humans", *Tech. Report NAS 2-2760*, NASA Ames Research Center (1966).

[28] Crowley, J.L., "Navigation for an Intelligent Mobile Robot", *IEEE J. Robotics and Automation*, RA-1(1), pp. 31-41 (March 1985).

[29] Darell, T., and Wohn, K., "Calculating Depth from Focus Using a Pyramid Architecture", *Automated Inspection and High Speed Vision Architectures*, SPIE, Vol. 849, pp. 57-62 (1987).

[30] Darell, T., and Wohn, K., "Pyramid Based Depth from Focus", *IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, pp. 504-509 (1988).

[31] Denstman, H., "State-of-the-art Optics: Automated Image Focusing", *Industrial Photography*, pp. 33-37 (July 1980).

[32] Duda, R. O., and Hart, P.E., *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York (1973).

[33] Durant-White, H.F., "Integration, Coordination and Control of Multi-Sensor Robot Systems", *Ph.D Dissertation MS-CIS-86-67*, University of Pennsylvania (1986).

[34] Engelhardt, K., and Hausler, G., "Acquisition of 3-D data by Focus Sensing", *Applied Optics*, 27(22), pp. 4684-4689 (Nov. 1988).

[35] Ens, J., and Lawrence, P., "A Matrix Based Method for Determining Depth from Focus", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (June 1991).

[36] Ens, J., *An Investigation of Methods for Determining Depth from Focus*, Ph.D Dissertation, Dept. of Electrical Engg., University of British Columbia.

[37] Erteza, A., "Depth of Convergence of a Sharpness Index Autofocus System", *Applied Optics*, 16(8), pp. 2273-2278 (August 1977).

[38] Erteza, A., "Sharpness Index and its Application to Focus Control", *Applied Optics*, 15(4), pp. 877-881 (April 1976).

[39] Ferrie, F.P. and Levine, M.D., "Piecing Together the 3-D Shape of Moving Objects: An Overview", *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp 598-607 (June 1985).

[40] Frisby, J., *Seeing: Illusion, Brain and Mind*, Oxford University Press, New York (1979).

[41] Gaskill, J.D., *Linear Systems, Fourier Transforms, and Optics,* John Wiley & Sons, New York (1978).

[42] Gibson, E.J., and Spelke, E.S., "The Development of Perception", *Handbook of Child Psychology*, Fourth Edition, ed. Paul H. Mussen, John Wiley and Sons, New York (1984).

[43] Gibson, J.J., *The Perception of the Visual World*, Hough-Mifflin, Boston (1950).

[44] Girod, B., and Scherock, S., " Depth from Defocus of Structured Light", *Optics, Illumination and Image Sensing for Machine Vision*, IV, SPIE, Vol. 1194 (1989).

[45] Goldberg, N., "Inside Autofocus: How the Magic Works", *Popular Photography*, pp.77-83 (Feb. 1982).

[46] Goodman, J.W., *Introduction to Fourier Optics*, Physical and Quantum Electronics Series, Mc-Graw Hill, Toronto (1968).

[47] Grimson, W.E.L., *From Images to Surfaces: A Computational Study of the Human Early Vision System*, MIT Press, Cambridge, Massachusetts (1981).

[48] Grossman, P., "Depth from Focus", *Pattern Recognition Letters*, 5, pp. 63-69 (Jan. 1987).

[49] Hanma, K., Masuda, M., Nabeyama, H., and Saito, Y., "Novel Technologies for Automatic Focusing and White Balancing of Solid State Color Video Camera", *IEEE Trans. Consumer Electronics*, CE-29(3), pp. 376-382 (August 1983).

[50] Hecht, E., *Optics*, Addison Wesley, Massachusetts (1987).

[51] Hopkins, H.H., "The Frequency Response of a Defocused Optical System", *Proc. Royal Soc. London*, A 231, pp. 91-103 (1955).

[52] Hopkins, H.H., "The Numerical Evaluation of the Frequency Response of a Optical System", *Proc. Physical Soc. London*, B 70 (1957).

[53] Herman, M., Kanade, T., and Kuroe, S., "Incremental Acquisition of a Three-Dimensional Scene Model from Images", *IEEE Trans. Pattern Analysis and Mach. Intell.*, **PAMI-6**(3), pp. 331-340 (May 1984).

[54] Horn, B.K.P., "Focusing", *Tech. Report AIM 160*, Massachusetts Institute of Technology (May 1968).

[55] Horn, B.K.P., *Robot Vision*, MIT Press, Cambridge, Massachusetts (1986).

[56] Howard, I.P., *Human Visual Orientation*, Wiley, New York (1982).

[57] Hwang, T., Clark, J.J., and Yuille, A.L., "A Depth Recovery Algorithm Using Defocus Information", *IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, pp. 476-481 (1989).

[58] Iberall, T., and Lyons, D., "Towards Perceptual Robotics", *Tech. Report COINS-84-17*, University of Massachusetts (August 1984).

[59] Jarvis, R.A., "Focus Optimization Criteria for Computer Image Processing", *Microscope* **24** (2), pp. 163-180 (1976).

[60] Jarvis, R.A., "A Perspective on Range Finding Techniques for Computer Vision", *IEEE Trans. Pattern Analysis and Machine Intelligence*, **PAMI-5** (2), pp. 122-139 (March 1983).

[61] Jarvis, R.A., "Projection Derived Space Cube Scene Models for Robotic Vision and Collision-Free Trajectory Planning", *Proc. Intl. Symp. Robotic Research*, Kyoto, Japan, pp. 294-301 (1984).

[62] Jarvis, R.A., "Range from Out-of-focus Blur", *Australian Joint Artificial Intelligence Conference* (Nov. 1988).

[63] Johnson, S.M., "Optimal Search for a Maximum is Fibonaccian", *Tech. Report P-856*, RAND, Santa Monica, California (1956).

[64] Kak, A.C., "Depth Perception for Robots", *Handbook of Industrial Robotics*, ed. Hof, S.Y., John Wiley and Sons, pp. 272-319 (1985).

[65] Kent, E.W., Whetley, T. and Nashman, M., "Real-Time Cooperative Interaction Between Structured Light and Reflectance Ranging for Robot Guidance", *Robotica 3*, pp. 7-11 (1985).

[66] Krotkov, E.P., and Martin, J.P., "Range from Focus", *Tech. Report MS-CIS-86-09*, University of Pennsylvania (Jan 1986).

[67] Krotkov, E.P., "Exploratory Visual Sensing for Determining Spatial Layout with an Agile Stereo Camera System", *Ph.D Dissertation MS-CIS-87-29*, University of Pennsylvania (May 1987).

[68] Krotkov, E.P., "Focusing", *Inter. J Computer Vision*, 1(3), pp.223-237 (Oct. 1987).

[69] Krotkov, E.P., Fuma, F., and Summers, J., "An Agile Stereo Camera System for Flexible Image Acquisition", *IEEE J. Robotics and Automation*, 4(1), pp. 108-113 (Feb. 1988).

[70] Krotkov, E.P., "Adaptive Control of Cooperative Sensors: Focus and Stereo Ranging with an Agile Camera System", *IEEE Intl. Conf. Robotics and Automation*, pp. 548-553 (April 1988).

[71] Krotkov, E.P., and Kories, R., "Cooperative Focus and Stereo Ranging", *Proc. Fourth Conf. AI Applications*, pp. 76-81 (March 1988).

[72] Krotkov, E.P., and Kories, R., "Integrating Multiple Uncertain Views of a Static Scene Acquired by an Agile Camera System", *Tech. Report MS-CIS-88-11*, University of Pennsylvania (1988).

[73] Kujoory, M.A., Mayall, B.H., and Mendelsohn, M.L., "Focus-Assist Device for a Flying Spot Microscope", *IEEE Trans. Biomedical Engg.* **20**, pp. 126-132 (March 1973).

[74] Kupchella, C.E., *Sights and Sounds, the very Special Senses*, The Bobbs-Merrill Company Inc,. (1976).

[75] Lai, S., and Fu, C., "A Generalized Depth Estimation Algorithm with a Single Image", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, NO. 4, pp 405-411 (April 1992).

[76] Lee, H., "Review of image-blur models in a photographic system using the principles of optics", *Optical Engineering*, Vol. 29 No. 5 pp. 405-421 (May 1990).

[77] Levi, L., and Austing, R.H., "Tables of the modulation transfer function of a defocused perfect lens", *Applied Optics*, Vol. 7, No. 5, pp. 967-974 (May 1968).

[78] Ligthart, G., and Groen, F.C.A., "A Comparison of different Autofocus Algorithms", *Proc. Intl. Joint Conf. on Pattern Recognition*, IEEE, pp. 597-600 (1982).

[79] Lindberg, D.C., *Theories of vision from Al-kindi to Kepler*, The University of Chicago Press (1976).

[80] Lu, M.C., *Computer Modeling and Simulation Techniques for Computer Vision Problems*, Ph.D Dissertation, SUNY, Stony Brook, New York (1993).

[81] Luo, R.C., and Lin, M., "Issues and Approaches of Automatic Focusing Algorithms for Intelligent Robot Eye-in-hand System", *J. Robotic Systems*, 4(4), pp. 459-476 (1987).

[82] Luo, R.C., and Lin, M., "Design and Implementation of new Autofocusing Algorithms for Intelligent Robot Eye-in-hand Systems", *Proc. Intl. Workshop on Industrial Automation Systems*, Seiken Symposium, pp. 11-16 (Feb. 1987).

[83] Marr, D., *Vision*, W.H. Freeman and Company, New York (1982).

[84] Mason, D.C., and Green, D.K., "Automatic Focusing of a Computer Controlled Microscope", *IEEE Trans. Biomed. Engg.*, BME-22(4), pp. 312-317 (July 1975).

[85] Meer, P., and Weiss, I., "Smoothed Differentiation Filters for Images", *Journal of Visual Communication and Image Representation*, 3, 1 (1992).

[86] Muller, R.A., and Buffington, A., "Real-time Correction of Atmospherically degraded Telescope Images through Image Sharpening", *J. Opt. Soc. America*, 64(9), pp. 1200-1210 (Sep. 1974).

[87] Nakada, Y., "Apparatus for Detecting Distance to an Object", United States Patent #4690549 (Sep. 1987).

[88] Nayar, S.K., "Shape from Focus System" *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, pp. 302-308 (June 1992).

[89] Nishibe, T., "Range Finder", United States Patent #4611910 (Sept 1986).

[90] Pentland, A., "Depth of Scene from Depth of Field", *Proc. Image Understanding Workshop*, pp. 253-259 (April 1982).

[91] Pentland, A., "The Focal Gradient: Optics Ecologically Salient", *Supplement to Investigative Opthalmology and Visual Science*, pp. 243 (April 1985).

[92] Pentland, A., "A New Sense for Depth of Field", *Proc. Intl. Joint Conf. Artificial Intelligence*, Los Angeles, pp. 988-994 (August 1985).

[93] Pentland, A., "A New Sense for Depth of Field", *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-9, pp. 523-531 (July 1987).

[94] Pentland, A., Darell, T., Turk, M., and Huang, W., "A Simple Real-time Range Camera", *IEEE Comp. Soc. Conf. Comp. Vision and Pattern Recognition*, pp. 256-261 (1989).

[95] Potmesil, M., and Chakravarty, I., "Synthetic Image Generation with a Lens and Aperture Camera Model", *ACM Trans. Graphics*, Vol. 1 No. 2, pp. 85-108 (Apr. 1982).

[96] Raskin, E., *Watchers, Pursuers and Masqueraders*, Mc-Graw Hill Book Company (1964).

[97] Rock, I., *The Logic of Perception*, MIT Press, Cambridge, Massachusetts (1983).

[98] Rosenfeld, A., Editor, *Techniques for 3-D Machine Perception*, Vol 3, Machine Intelligence and Pattern Recognition Series, Elsevier Science Publishers, Amsterdam (1986).

[99] Rosenfeld, A., and Kak, A.C., *Digital Picture Processing*, Vol 1, Academic Press, New York (1982).

[100] Schlag, J.F., Sanderson, A.C., Neumann, C.P., and Wimberly, F.C., "Implementation of Automatic Focusing Algorithms for a Computer Vision System with Camera Control", *Tech. Report CMU-RI-TR-83-14*, Carnegie Mellon University (August 1983).

[101] Schreiber, W.F., *Fundamentals of Electronic Imaging Systems*, Vol 15, Springer Series in Information Sciences, Springer-Verlag (1986).

[102] Sobel, I., "Camera Models and Machine Perception", *Tech. Report AIM-121*, Stanford University (May 1971).

[103] Stokseth, P.A., "Properties of a Defocused Optical System", *J. Opt. Soc. America*, Vol. 59 No. 10 pp. 1314-1321 (Oct. 1969).

[104] Strand, T.C., "Optical Three-Dimensional Sensing for Machine Vision", *Optical Engineering*, 24(1), pp.33-40 (Jan. 1985).

[105] Subbarao, M., "Direct Recovery of Depth Map I: Differential Methods", *Proc. IEEE Comp. Soc. Workshop Computer Vision*, pp. 58-65 (Dec. 1987).

[106] Subbarao, M., "Parallel Depth Recovery by Changing Camera Parameters", *Proc. IEEE Sec. Intl. Conf. Computer Vision*, pp. 149-155 (Dec. 1988).

[107] Subbarao, M., and Gurumoorthy, N., "Depth Recovery from Blurred Edges", *Proc. IEEE Conf. CVPR*, pp. 498-503 (1988).

[108] Subbarao, M., "Efficient Depth Recovery through Inverse Optics", Editor: H. Freeman, *Machine Vision for Inspection and Measurement,* Academic press, Boston, pp. 101-126 (1989).

[109] Subbarao, M., "Determining Distance from Defocused Images of Simple Objects", *Tech. Report 89.07.20*, Computer Vision Lab, SUNY, Stony Brook (July 1989).

[110] Subbarao, M., "Computational Methods and Electronic Camera Apparatus for Determining Distance of Objects, Rapid Autofocusing, and obtaining improved Focus Images", U.S. patent application serial number 07/373,996, June 1989.

[111] Subbarao, M., "On the Depth Information in the Point Spread Function of a Defocused Optical System", Tech. Report 90.02.07, Computer Vision

Lab, SUNY, Stony Brook (Feb. 1990).

[112] Subbarao, M., and Nikzad, A., "A Model for Image Sensing and Digitization in Machine Vision", *OE/BOSTON '90*, SPIE conference, Boston, Vol. 1385, pp. 70-84 (Nov. 1990).

[113] Subbarao, M., "Spatial-Domain Convolution/Deconvolution Transform ", *Tech. Report No. 91.07.03*, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY, Stony Brook (1991).

[114] Subbarao, M., and Wei, T., "Depth from Defocus and Rapid Autofocusing : A practical Approach", *Proceedings of the IEEE Computer Society Conference CVPR*, Champaign, Illinois, (June 1992).

[115] Subbarao, M., and Wei, T., "Depth from Defocus and Rapid Autofocusing : A practical Approach", *Tech. Report 92.01.17*, Computer Vision Lab., Dept. of Electrical Engg., SUNY, Stony Brook, New York (Jan 1992).

[116] Subbarao, M., and Lu, M., "Computer Modeling and Simulation of Camera Defocus", *Vol. 1822, Proceedings of SPIE conference*, OE/TECHNOLOGY '92, Boston, pp. 110 - 120 (Nov. 1992).

[117] Subbarao, M., and Surya, G., "Application of Spatial-Domain Convolution/Deconvolution Transform for Determining Distance from Image Defocus", *Vol. 1822, Proceedings of SPIE conference*, OE/TECHNOLOGY '92, Boston, pp. 159 - 167 (Nov. 1992).

[118] Subbarao, M., and Surya, G., "Depth from Defocus: A Spatial Domain Approach", *Tech. Report No. 92.12.03*, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY, Stony Brook, NY 11794-2350. (Revised version to appear in International Journal of Computer Vision).

[119] Subbarao, M., Choi, T.S., and Nikzad, A., "Focusing Techniques", *Vol. 1823 , Proceedings of SPIE conference*, OE/TECHNOLOGY '92, Boston, pp 163 - 174 (Nov. 1992).

[120] Surya, G., and Subbarao, M., "Depth from Defocus by Changing Camera Aperture: A Spatial Domain Approach", *Proceedings of the IEEE Computer Society Conference CVPR*, New York, pp. 61-67 (June 1993).

[121] Tenenbaum, J.M., "Accomodation in Computer Vision", *Ph.D Dissertation*, Stanford University (Nov. 1990).

[122] Tenenbaum, J.M., "A Laboratory for Hand-Eye Research", *IFIPS*, pp.206-210 (1971).

[123] Tolansky, S., *Curiosities of Light Rays and Light Waves*, American Elsevier Publishing Co. Inc., New York (1964).

[124] Weale, R.A., *Focus on Vision*, Harvard University Press (1982).

[125] Wei, T., and Subbarao, M., "Continuous Focusing of Moving Objects Using DFD1F", *Tech. Report No. 93.07.08*, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY Stony Brook, New York (July 1993).