Three Dimensional Machine Vision Using Image Defocus

A Dissertation Presented

by

Tse-Chung Wei

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

Abstract of the Dissertation

Three Dimensional Machine Vision Using Image Defocus

by

Tse-Chung Wei

Doctor of Philosophy

in

Electrical Engineering

State University of New York at Stony Brook

1994

A Fourier domain approach has been investigated for using image defocus information in three-dimensional machine vision. Fast new methods have been developed and implemented for recovering distance and focused image of stationary and moving objects. The methods can also be used in rapid autofocusing of video cameras.

The method for finding distance, named DFD1F, requires only two images acquired at two different camera parameter settings. Therefore it is very fast in comparison with depth-from-focus methods that require a large (10 or more) number of images. The camera parameters include focal length, diameter of camera aperture, and the distance between the image detector and the camera lens. Any one or more of these can be changed to obtain the two different camera settings. Theoretical principles, computational algorithms, and implementation issues are discussed. Experimental results indicate that DFD1F is useful in practical three-dimensional machine vision.

DFD1F has been extended for continuous focusing of moving objects in a video camera. As part of this method, a new camera structure has been proposed, and a new parameterization of the Modulation Transfer Function data of the camera is used.

Four traditional methods and one new method have been investigated for restoring one of the blurred image used by DFD1F to obtain a focused image. The new method is based on a spatial domain approach for deconvolution.

DFD1F does not have the correspondence problem that arises in stereo vision, but, in general, it is less accurate than stereo vision in recovering depth. Therefore DFD1F can be combined with stereo vision to reduce the correspondence problem and obtain accurate depth estimates. This has been demonstrated successfully with experiments on a prototype camera system.

To the memory of my grandfather

Contents

List of Figures

List of Tables

Acknowledgements

My sincere thanks to Prof. M. Subbarao for his guidance and inspiration during the research. Also, my thesis committee, Prof. J. Murray, Prof. P. Djuric, and Prof. R. Surya, for helpful discussions.

I would like to express my appreciation to my colleague Dr. G. Surya for many stimulating discussions and the join work on focused image recovery. Special thanks to Ms. M. Krause for her kindness and help during my graduate studies.

The support of this research by the National Science Foundation (IRI-8821923) and the Olympus Optical Corporation is gratefully acknowledged.

Chapter 1

Introduction

1.1 Motivation

Computational studies and perceptual experiments [?] indicate that in human vision the computational processes that interpret visual information in the early stages are generic ones. As a first approximation, these processes can be considered to be conceptually independent modules that can be studied in isolation. Examples of such early vision processes are stereo, motion, shape-from-texture, image defocus, etc. A second level study of visual perception would involve the investigation of interaction between the early vision processes and how the information from different processes are integrated efficiently to obtain a unique interpretation of the scene.

Early vision processes such as stereo and motion have been studied extensively in the past three decades by many researchers. However, the study of image defocus module was started only a few years ago by a handful of researchers. The potential of image defocus as a source of three-dimensional information had not been recognized until then. There was no prior study and literature that provided a comprehensive treatment of the image defocus module and its possible applications in machine vision. This was the primary motivating factor behind this dissertation.

In this thesis we have studied the image defocus as an independent module in the context of three-dimensional machine vision. Image defocus contains both geometric (distance and shape) and photometric (color and image irradiance) information. Theoretical principles, computational algorithms, and implementation techniques, have been investigated for extracting both geometric and photometric information from image defocus. Image defocus uses ambient illumination and therefore is a passive method of sensing. Therefore it is preferable in many applications when compared to active sensing methods that need sending out beams of energy such as lasers, infrared light, and sonar waves.

1.2 Depth from focus and defocus (DFF & DFD)

In the image formed by an optical system such as a convex lens, objects at a particular distance (or depth) from the lens will be focused whereas objects at other distances will be blurred or defocused by varying degrees depending on their distance. This suggests that the degree of image blur could be a source of distance information. In prior literature, two approaches have been used for obtaining depth information from image defocus. In the first approach, one of the camera parameter such as the image detector position or the focal length is varied until the object of interest is focused. Then the distance of the object is obtained using a lens formula. This approach involves a search of the camera parameter space to find the camera setting that brings a desired object into focus. Therefore this approach requires acquiring and processing many images (about 10 in practice). Most of the passive methods of autofocusing and ranging (finding distance) in computer vision follow this approach [?, ?, ?, ?, ?, ?, ?, ?]. We call this approach to finding distance Depth-from-Focus (DFF) as it involves first focusing the object. A recent work comparing different DFF techniques can be found in [?, ?].

The second approach for finding distance, unlike DFF, does not require focusing the object of interest [?, ?, ?, ?, ?, ?, ?]. In this approach the level of defocus of the object is taken into account in determining distance. Therefore, we will call this approach to be Depth-from-Defocus (DFD). DFD approach does not involve searching for the focused object. Therefore it requires processing only a few images (about two or three) as compared to a large number (about 10) of images in the DFF approach. In addition, only a few images are sufficient to determine the distance of all objects in a scene using the DFD approach, irrespective of whether the objects are focused or not. DFD approach involves less computation than the DFF approach. Also, methods based on the DFD approach are about 5 times faster than those based on the DFF approach due to the reduction in the mechanical movement of camera parts for changing camera parameters. In this thesis we take the DFD approach based on a Fourier domain analysis of image defocus. Both depth-map recovery and restoration of defocused images are investigated.

1.3 Overview

This dissertation is organized as follows. In chapter 2, the image formation process in a camera system is presented. A camera model is presented for a camera system and the parameters of the camera system are defined. Three widely used models for the point spread function (PSF) of the camera are presented. The three models are– PSF based on paraxial geometric optics model of image formation, two-dimensional Gaussian PSF, and the PSF based on wave optics model of image formation. The focused image of a three-dimensional scene is defined and the defocused image of the scene is modeled as the result of convolving the focused image with the camera PSF in small image regions. This chapter forms the basis for the discussion in the remaining chapters.

In chapter 3, we review some commonly used focus measures and Depth-from-Focus methods. We also present a new DFF method and some experimental results. The new method gives more accurate and higher resolution depth-maps at the cost of sensing and processing far more images than the traditional methods of DFF. The DFF methods serve as a benchmark for evaluating the performance of DFD methods.

Chapter 4 contains a literature review of Depth-from-Defocus (DFD) methods. Several methods are summarized and their advantages and disadvantages in comparison with our method are outlined.

In Chapter 5, a new fast method of determining distance of objects and autofocusing a camera using image defocus information is presented. The method, named DFD1F, requires only two images acquired at two different camera parameter settings and therefore is very fast in comparison with depth-from-focus [?] methods which require a large number (10 or more) of images. The camera parameters include focal length, diameter of camera aperture, and the distance between the image detector and the camera lens. Any one or more of these can be changed to obtain the two different camera settings. DFD1F is general in that it is not restricted to any particular model of the point spread function of the camera such as Gaussian or cylindrical. It involves the computation of only a few (about six) one-dimensional Fourier coefficients of a discrete sequence obtained by summing the images along some direction. Therefore DFD1F is very efficient and robust with respect to zero-mean noise.

DFD1F has been successfully implemented on a prototype camera system named SPARCS. It can determine the distance of an object placed in front of the camera in the range 0.6 meter to infinity in less than a second of computation on a personal computer. Based on the computed distance, the camera can autofocus by moving the lens to the correct position. A large number (209) of experiments on natural objects indicate that the method is useful in practical applications such as robotic vision and rapid autofocusing.

In chapter 6, we extend DFD1F to continuous focusing of moving objects. In the case of moving objects, the two images used by DFD1F must be recorded simultaneously in a short time period. A new camera structure is proposed for such recording of the images. In our method for continuous focusing, the requirement of a large memory space has been avoided for storing the MTF data of the camera's optical system. This is achieved by using a parameterization scheme for the MTF data. The method has been implemented on an actual camera system. Experimental results on this system indicate that the method yields an RMS error in focusing of about 4.3% in lens position. The image blur caused by a focusing error of this magnitude is barely noticeable by humans. Therefore, in addition to machine vision, the method has practical applications in video cameras such as camcorders.

Chapter 7 deals with image restoration methods that can be used in conjunction with DFD1F to obtain the focused image without actually moving the lens to focus on the object. Two new methods are presented for recovering the focused image of an object from only two blurred images recorded with different camera parameter settings as in DFD1F. First a blur parameter σ is estimated using DFD1F. Then one of the two blurred images is deconvolved to recover the focused image. The first method is based on a recently proposed Spatial Domain Convolution/Deconvolution Transform. This method requires only the knowledge of σ of the camera's point spread function (PSF). It does not require information about the actual form of the camera's PSF. The second method, in contrast to the first, requires full knowledge of the form of the PSF. As part of the second method, we present a calibration procedure for estimating the camera's PSF for different values of the blur parameter σ. In the second method, the focused image is obtained through deconvolution in the Fourier Domain using the Wiener filter. For both methods, results of experiments on actual defocused images recorded by a CCD camera are given.

The first method requires much less computation than the second method. The first method gives satisfactory results for up to medium levels of blur and the second method gives good results for up to relatively high levels of blur.

Chapter 8 illustrates how DFD1F can be integrated with stereo vision. The integrated method combines the strengths of the two methods but over comes their individual weaknesses. The integrated method is as accurate as stereo vision but much faster than stereo vision alone. Experimental comparison of using only stereo and the integrated approach on a slanted object are presented.

Finally, a summary and possible extensions of this research is presented in Chapter 9.

Chapter 2

Camera and PSF Models

2.1 Introduction

In this chapter we introduce a camera model for a typical camera system used in machine vision. We also present three models for the point spread function (PSF) of the camera system. The following chapters use the models and notations introduced in this chapter.

2.2 Camera Model

In an image forming optical system such as a convex lens, the image of an object formed on an image detector plane will be usually blurred. The degree of blur depends on the focal length f of the lens, the distance u of the object from the camera and the distance s between image detector and the lens (see Fig. 2.1). A well-known relation is the lens formula based on paraxial geometric optics

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

Following this formula, a focused image is obtained under the condition that s equals v. The degree of blur increases as the difference between s and v increases. If f and v are known, then the distance (depth) u of the object can be found using this formula.

A schematic diagram of a camera system with variable camera parameters is shown in Fig. 2.2. It consists of an optical system with two lenses L1 and L2. The effective focal length f is varied by moving one lens with respect to the other. O.A. is the optical axis, P1 and P2 are the principal planes, Q1 and Q2 are the principal points, ID is the image detector, D is the aperture diameter, s is the distance between the second principal plane and the image detector, u is the distance of the object from the first principal plane, and v is the distance of the focused image from the second principal plane.

The distance s, focal length f, and aperture diameter D, will be referred together as camera parameters and denoted by $\mathbf{e}$, i.e.

$$\mathbf{e} \equiv (s, f, D). \tag{2.2}$$

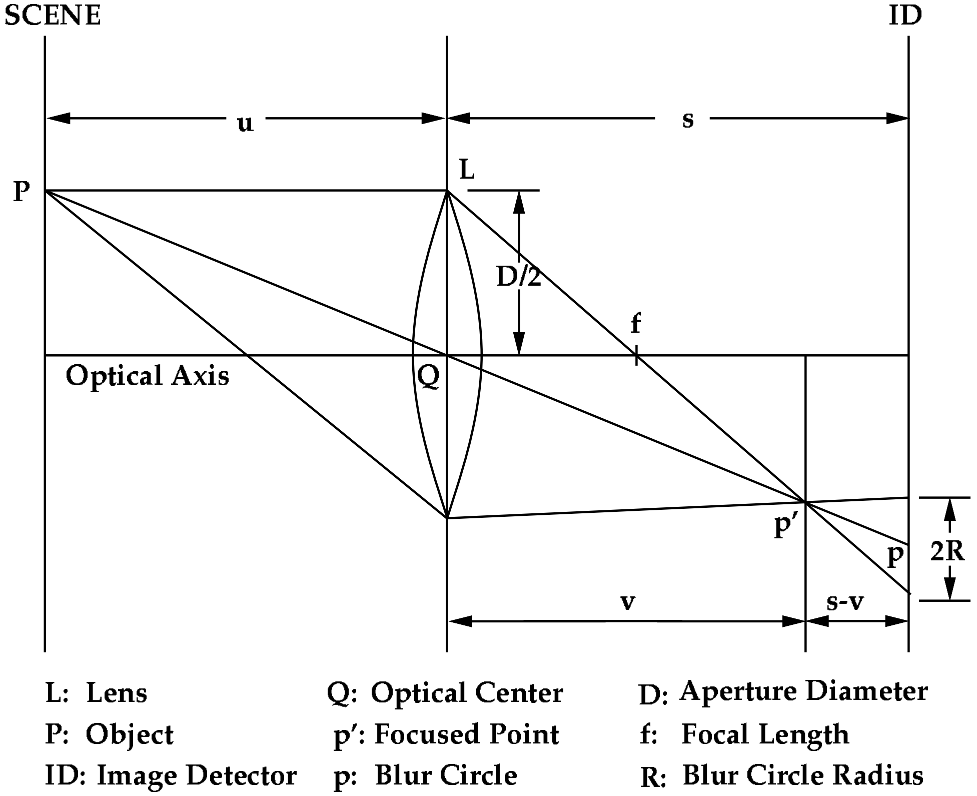


Figure 2.1: Image Formation in a Convax Lens

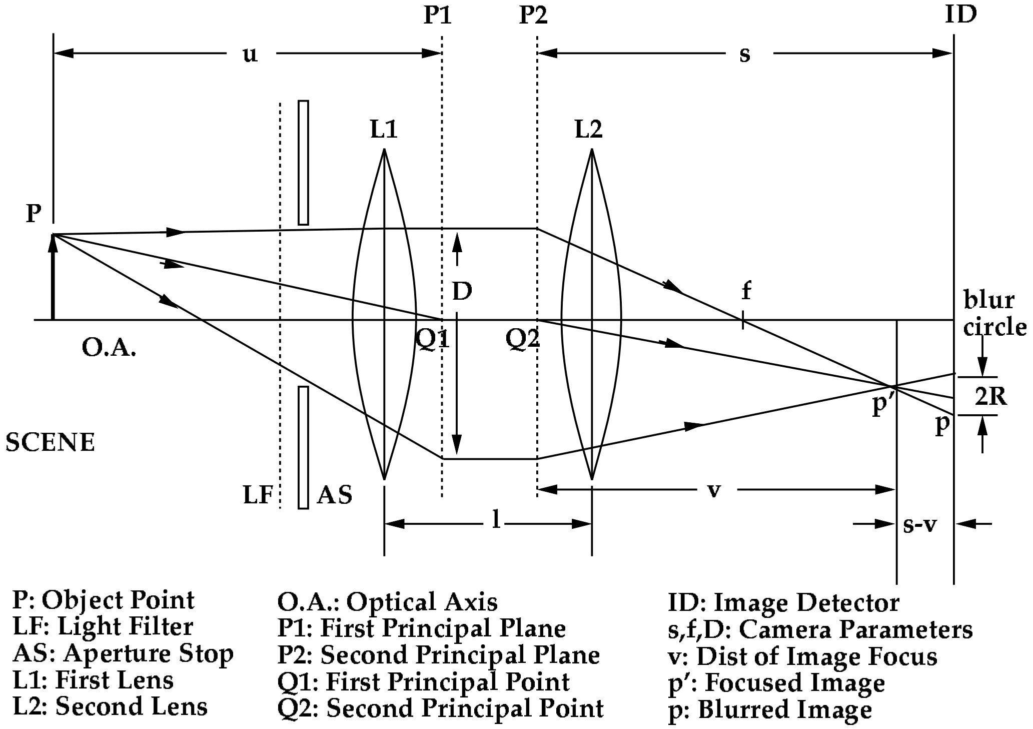


Figure 2.2: Camera Model and Camera Parameters

2.3 Models of Point Spread Function

In the following discussion, we consider the optical system to be circularly symmetric around the optical axis. In Figure 2.2, if a point light source placed at point P is not in focus, then it gives rise to a blurred image p on the image detector ID. Since we have taken the aperture to be circular, the blurred image of P is also a circularly symmetric function. Let the light energy incident on the lens from the point P during one exposure period of the camera be one unit. Then, the blurred image of P is the response of the camera to a unit point source and hence it is the Point Spread Function (PSF) of the camera system. Three forms of Point Spread Function are used often in the literature. We will discuss each of them below.

2.3.1 Geometric Optics PSF

According to paraxial geometric optics model [?] , the blurred image of P has the same shape as the lens aperture but scaled by a factor. This holds irrespective of the position of P on the object plane. The blurred image of P will be a circle with uniform brightness inside the circle and zero outside. This is called a blur circle. The blur circle will be the point spread function $h_a(x, y)$.

Let R be the radius of the blur circle and q be the scaling factor defined as $q = 2R/D$. In Figure 2.2, from similar triangles, we have

$$q = \frac{2R}{D} = \frac{s-v}{v} = s\left[\frac{1}{v} - \frac{1}{s}\right] \tag{2.3}$$

Substituting for $1/v$ from Eq. (??) in the above equation, we obtain

$$q = s\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{2.4}$$

Substituting for $1/v$ from Eq. (??) in the above equation and simplifying, we obtain

$$R = q\frac{D}{2} = s\frac{D}{2}\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{2.5}$$

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.

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 the images g_1 and g_2 correspond to the same field of view [?]. The relation between an original image $g(x, y)$ taken with $s = s_0$ and the corresponding magnification normalized image $g_n(x,y)$ is given by $g_n(x/s_0, y/s_0) = g(x,y)$. However, in most practical applications, the magnification change is less than 3% and can be ignored. It is probably for this reason that most previous literature fails to mention the magnification correction. But this cannot be overlooked from a theoretical point of view.

In the following discussion we assume that the images have been normalized. Without loss of generality, we assume that both the mean brightness and magnification have been normalized to be 1. After magnification normalization, the normalized radius $R' = R/s$ of the blur circle can be expressed as a function of the camera parameter setting e and object distance u as

$$R'(\mathbf{e};u) \;=\; \frac{D}{2}\left(\frac{1}{f}-\frac{1}{u}-\frac{1}{s}\right). \tag{2.6}$$

If we assume the camera to be a lossless system (i.e., no light energy is absorbed by the camera system) then

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h_a(x,y)\;dx\;dy \;=\; 1 \tag{2.7}$$

because the light energy incident on the lens was taken to be one unit. Using this and the fact that the blur circle has uniform brightness inside a circle of radius R' and zero outside, we obtain the PSF to be a cylindrical function:

$$h_a(x,y) = \begin{cases} \frac{1}{\pi R'^2} & \text{if } x^2+y^2 \le R'^2 \\ 0 & \text{otherwise.} \end{cases} \tag{2.8}$$

The Optical Transfer Function (OTF) corresponding to the above PSF

(Eq. ??) is

$$H_a(\omega, \nu; \mathbf{e}, u) \;=\; 2\frac{J_1(R'(\mathbf{e}; u)\; \rho(\omega, \nu))}{R'(\mathbf{e}; u)\; \rho(\omega, \nu)} \tag{2.9}$$

where ω, ν, and ρ are spatial frequencies specified in radians/unit distance, J_1 is the first order Bessel function, and ρ is the radial spatial frequency

$$\rho(\omega, \nu) \;=\; \sqrt{\omega^2 + \nu^2}. \tag{2.10}$$

Eq. (??) explicitly represents the dependence of the OTF on the camera parameter setting $\mathbf{e}$ and the object distance u.

2.3.2 Gaussian PSF Model

In practice, the image of a point object is not a crisp circular patch of constant brightness as suggested by geometric optics. Instead, due to diffraction, poly-chromatic illumination, lens aberrations, etc., it will be a roughly circular blob with the brightness falling off gradually at the border rather than sharply. Therefore, as an alternative to the above cylindrical PSF model, often [?, ?, ?, ?] a two-dimensional Gaussian is suggested which is defined by

$$h_b(x, y) \;=\; \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.11}$$

where σ is a spread parameter corresponding to the standard deviation of the distribution of the PSF. In practice, it is found that [?, ?] σ is proportional to R', i.e.

$$\sigma \;=\; c\,R' \;\text{ for }\; c > 0 \tag{2.12}$$

where c is a constant which is approximately equal to $1/\sqrt{2}$ in practice [?]. Since the blur circle radius R' is a function of $\mathbf{e}$ and u, σ can be written as

$\sigma(\mathbf{e}, u)$. (However, the image of an actual point light source for our camera was closer to a cylindrical function than a Gaussian. The size of the cylindrical function was, however, different from that predicted by paraxial geometric optics.)

The OTF corresponding to the above Gaussian PSF is (ω, ν in radians/unit dist.)

$$H_b(\omega, \nu; \mathbf{e}, u) \; = \; e^{-\frac{1}{2}\rho^2(\omega,\nu)\,\sigma^2(\mathbf{e};u)} \tag{2.13}$$

where

$$\sigma(\mathbf{e}; u) \; = \; c\,\frac{D}{2}\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right). \tag{2.14}$$

Once again, Eq. (??) explicitly represents the dependence of the OTF on the camera parameter setting $\mathbf{e}$ and the object distance u.

2.3.3 Wave Optics

In the wave optics model, light entering and leaving the lens are treated as electro-magnetic fields. The defocusing model used here follows the models used by Hopkins [?], Levi and Austing [?], and Stokseth [?]. According to Levi and Austing [?], the OTF corresponding to a focus defect Δ is given by

$$H_w(\rho, \Delta) = \frac{4}{\pi}\int_\rho^1 \sqrt{1-t^2}\cos\left[2\pi\Delta\rho(t-\rho)\right]dt \tag{2.15}$$

in polar coordinate. Δ is the defocus measure and it can be expressed as [?],

$$\Delta \approx \frac{2R^2}{\lambda}\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{v}\right) \tag{2.16}$$

where λ is the wavelength of the incident light. This gives an approximation for the focus defect Δ in terms of the camera parameters $\mathbf{e}$ and the distance u

of the object. The PSF of wave optics model is given by inverse Fourier-Bessel transform on Eq. (??),

$$h_w(r,\Delta) = 2\pi \int_0^\infty H_w(\rho,\Delta) J_0(2\pi\rho r)\rho d\rho \tag{2.17}$$

where J_0 is the zeroth order Bessel function of the first kind.

For any circularly symmetric PSF $h(r)$, the spread parameter σ is defined as

$$\sigma^2 = 2\pi \int_0^\infty r^3 h(r) dr \tag{2.18}$$

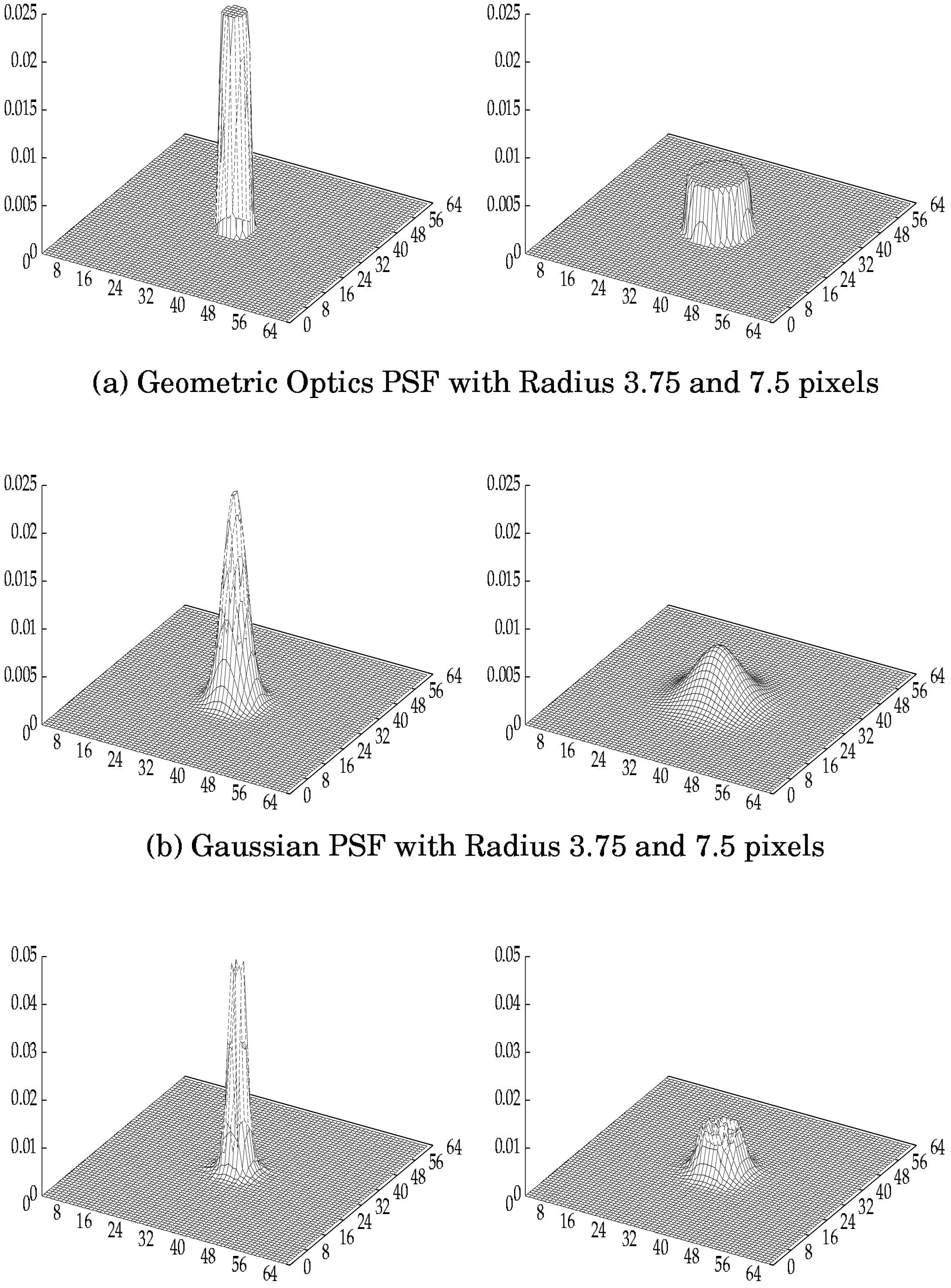
For PSF based on geometric optics the spread parameter is $\sigma_g = \Delta/\sqrt{2}$. It has been shown that for the wave optics PSF model, the spread parameter σ_w has the following relation with σ_g [?]

$$\sigma_w^2 \approx \sigma_g^2 + \sigma_0^2 \tag{2.19}$$

where σ_0 is the spread parameter when the optical system is focused according to geometric optics. As Δ gets larger, i.e. $\sigma_g >> \sigma_0$, the spread parameter for geometric optics model and wave optics model will be almost the same. We conclude that when the blur is moderate or large, for characterizing the blur parameter σ, Gaussian and geometric optics models are good approximations for the wave optics model.

In order to compare the three models of PSF presented in this chapter, 3D plots of the PSFs for the three models are shown in Fig. 2.3. The PSFs correspond to blur circle radius of 3.75 and 7.5 pixels. Fig. 2.4 shows 2D plots of the cross sections of the circularly symmetric PSFs and their Modulation Transfer Functions (MTF) for blur circle radius corresponding to 3.5, 7.5, and

15 pixels. We see that as the blur circle radius increases, geometric optics model becomes closer approximation to the wave optics model.

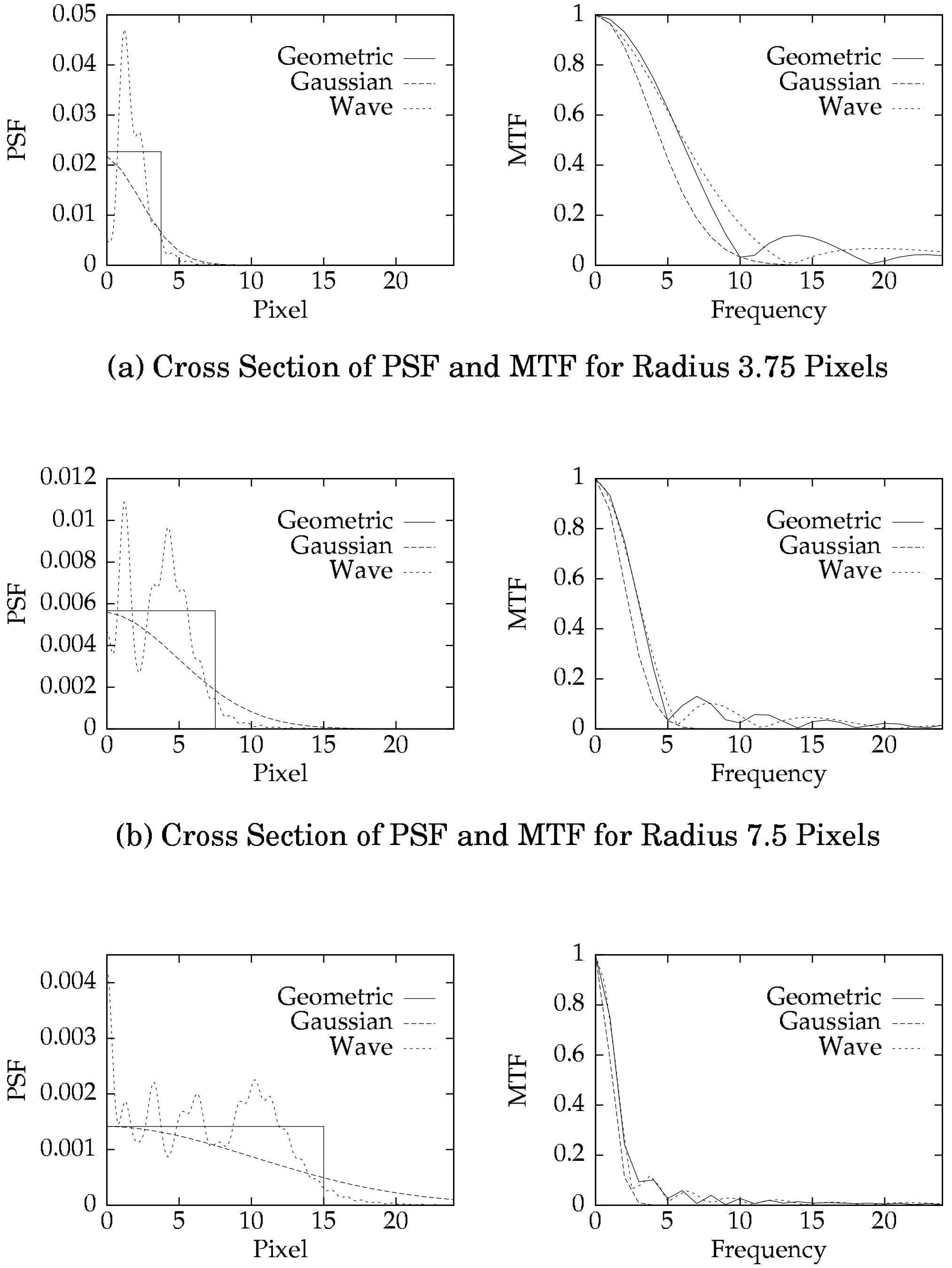


(a) Geometric Optics PSF with Radius 3.75 and 7.5 pixels

(b) Gaussian PSF with Radius 3.75 and 7.5 pixels

(c) Wave Optics PSF with Radius 3.75 and 7.5 pixels

Figure 2.3: PSF



(a) Cross Section of PSF and MTF for Radius 3.75 Pixels

(b) Cross Section of PSF and MTF for Radius 7.5 Pixels

(a) Cross Section of PSF and MTF for Radius 15 Pixels

Figure 2.4: PSF and MTF Cross Sections

2.4 Image Defocusing as Convolution Operation

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) (Subbarao and Nikzad, 1990). We do not know of any previous literature on focusing techniques which gives a precise and correct (we believe) definition of the focused image as we have done here. Such a definition is essential for a sound analysis of DFD methods.

For a planar object perpendicular to the optical axis, the blur circle radius R is a constant over the image of the object. Let $g(x,y)$ be the image of the object recorded by the camera with parameter settings $\mathbf{e} = (s,f,D)$. In this case the camera acts as a linear shift invariant system. Therefore $g(x,y)$ will be equal to the convolution of the focused image $f(x,y)$ with the corresponding point spread function

$$g(x,y) \; = \; h(x,y;\mathbf{e},u) \; * \; f(x,y) \tag{2.20}$$

where $*$ denotes the convolution operator. Convolution in the spatial domain corresponds to multiplication in the Fourier domain. Therefore, if F and G are Fourier transforms of f and g respectively,

$$G(\omega,\nu) \; = \; H(\omega,\nu;\mathbf{e},u) \; F(\omega,\nu) \tag{2.21}$$

The above discussion is valid only for the region close to the optical axis of a thin lens. However, the above equations are still a good approximation for our camera system.

Chapter 3

Review of Depth from Focus Methods

3.1 Introduction

In this chapter we review the literature on depth-from-focus (DFF) methods. The different methods proposed in the literature are summarized and their strengths and weaknesses are outlined.

3.2 Depth from Focus

In depth-from-focus methods, the problem is to find the camera parameter setting $\mathbf{e}$, Eq. (??), that brings a desired object into focus. Once the best focused setting is found, the object distance can be computed by using the lens formula in Eq. (??). The focus setting can be found by recording a number of images at different camera settings and computing a focus measure for each of the recorded images. The image with the highest focus measure is the focused image of the object and the corresponding camera setting is the desired focus setting. Since this method involves focusing the object, it is called

Depth-from-Focus (DFF). Approaches for Depth-from-Focus can be found in [?, ?, ?, ?, ?, ?, ?, ?, ?, ?]. We will discuss some of the common focus measures that have been used in the literature. We will also describe a new DFF method and related experimental results.

3.3 Focus Measures

A good focus measure should be sound in the absence of noise and robust in the presence of noise. Also, another desirable characteristic is that the focus measure should increase monotonically with decreasing defocus and should have a sharp peak at the best focused image. Some widely used focus measures that satisfy the above criteria are summarized below. References on this topic can be found in [?, ?].

- Gray-level variance

  The variance of image gray level can be used as a focus measure. This can be seen intuitively as the more variation in the image suggests that there are more details in the image. The variance of an $N \times N$ image is computed as

  $$\sigma^2 = \frac{1}{N^2} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} (g(x,y) - \mu)^2 \tag{3.1}$$

  where μ is the mean value of gray level within the image.

- Sum of gradient magnitude squared

  In this approach [?, ?] gradient magnitude is computed as

  $$\int_x \int_y |\nabla g|^2 = \int_x \int_y \left[\left(\frac{\partial g}{\partial x} \right)^2 + \left(\frac{\partial g}{\partial y} \right)^2 \right] \tag{3.2}$$

The focus measure is obtained by summing up the gradient magnitude inside a window.

- Sum of Laplacian squared

  Laplacian is a high pass filter. The focus measure using Laplacian is computed as

$$\int_x \int_y \left|\nabla^2 g\right|^2 = \int_x \int_y \left|\frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}\right|^2 \tag{3.3}$$

  Focus measure for a certain point is obtained by applying the above filter over a region with the point of interest at the center.

Some related focus measures are Sum Modulus Difference [?], Sum Modified Laplacian [?], and energy of band-pass filtered images [?]. See [?] for a detailed treatment of the various focus measures.

3.4 Shape and Focused Image Recovery

In a typical DFF method, the 3D shape of an object and its focused image are recovered as follows. The camera setting is changed by moving the lens with respect to the image detector. A sequence of images g_i are recorded for different lens positions s_i for $i = 1, 2, 3, \cdots, n$. Usually the positions are uniformly spaced with a spacing of δs. Focus measures are computed for each image g_i in small regions of size about 5×5 to 20×20. In each image region, the image with the maximum focus measure is determined. The positions of these images are used to compute the 3D shape. The focused images in the different image regions are synthesized to obtain the focused image of the entire object. If the object is flat, the position of the maximum of the focus measure can be found by binary search or Fibonacci search.

In the method outlined above the focus measures are computed at each pixel and summed in small image regions of size about 5×5 to 20×20. The larger the size of the image region the lesser the effect of noise and digitization on the computed focus measure. Also, larger image regions are needed for poor contrast images and for images with low spatial detail. However increasing the size of the image region decreases the spatial resolution of the depth-map recovered. Therefore there is a trade-off between the spatial resolution of the depth-map and signal-to-noise ratio which depends on noise, image contrast, and image detail.

We propose the following improvement to the DFF methods that increases the spatial resolution of the depth-map at the cost of processing more images and additional computation. A sequence of images are recorded with $\delta s \approx p \cdot F$ where p is the linear size of a pixel and F is the F-number of the camera system. The images stacked in order are considered to form an image volume. The focus measure is computed at each pixel as in other DFF methods. Then, instead of summing the focus measure in small image regions as in a typical DFF method, the focus measure is summed in small image volumes. One can consider summing in partially overlapping image volumes. Then the positions of the focus measure maximums are found in small fields of view. From this information, a dense depth-map and a focused image of the 3D object can be reconstructed. Note that, summing in a $3 \times 3 \times 3$ image volume has roughly the same noise smoothing power as summing in a image region of size 5×5, but the spatial resolution of the depth-map in the former case is roughly twice that of the latter case. In the next section we present some experimental results related to this improved DFF method.

3.5 Experiments

In the first experiment, we used a CCD camera to take images of a cone object with the tip of the cone placed at 65 cm and extending to around 2 meters with a base diameter of about 30 cm. A sequence of images of size 360 × 360 was taken by changing camera parameter s. The focal length was 35 mm, F-number was 4, effective pixel size was 0.011 mm and $\delta s = 0.030$ mm. The lens position was changed by a stepper motor with step numbers $0, 1, \cdots 96$. Step 0 corresponded to $s = f$, which focuses objects at distance infinity. Fig. 3.1 shows that for a certain step, only part of the images will be focused. Fig. 3.2 is the step number for each pixel where focus measure is a maximum. Based on the computed step number, a focused image is reconstructed. The reconstructed image is shown in Fig. 3.3. We see that the image is focused everywhere in comparison with images in Fig. 3.1. Fig. 3.4 shows the depth-map of the object. Except for the lower corner of the image where there was very little gray level variation, the results are good.

The second experiment was done using a microscope as the optical system. A mustard seed was placed on a stage under the microscope and 40 images of size 480 × 480 were recorded at 40 different step positions of the stage. The successive step positions of the stage were 0.004 mm apart. Fig. 3.5 shows some of these images. We see that only some part of these images are focused. The reconstructed image that is focused everywhere is shown in Fig. 3.6. Figures 3.7 and 3.8 show the focus step number. They represent the depth-map with a scaling factor of 0.004 mm per step. In Fig. 3.7, a brighter pixel corresponds to a closer point on the object.

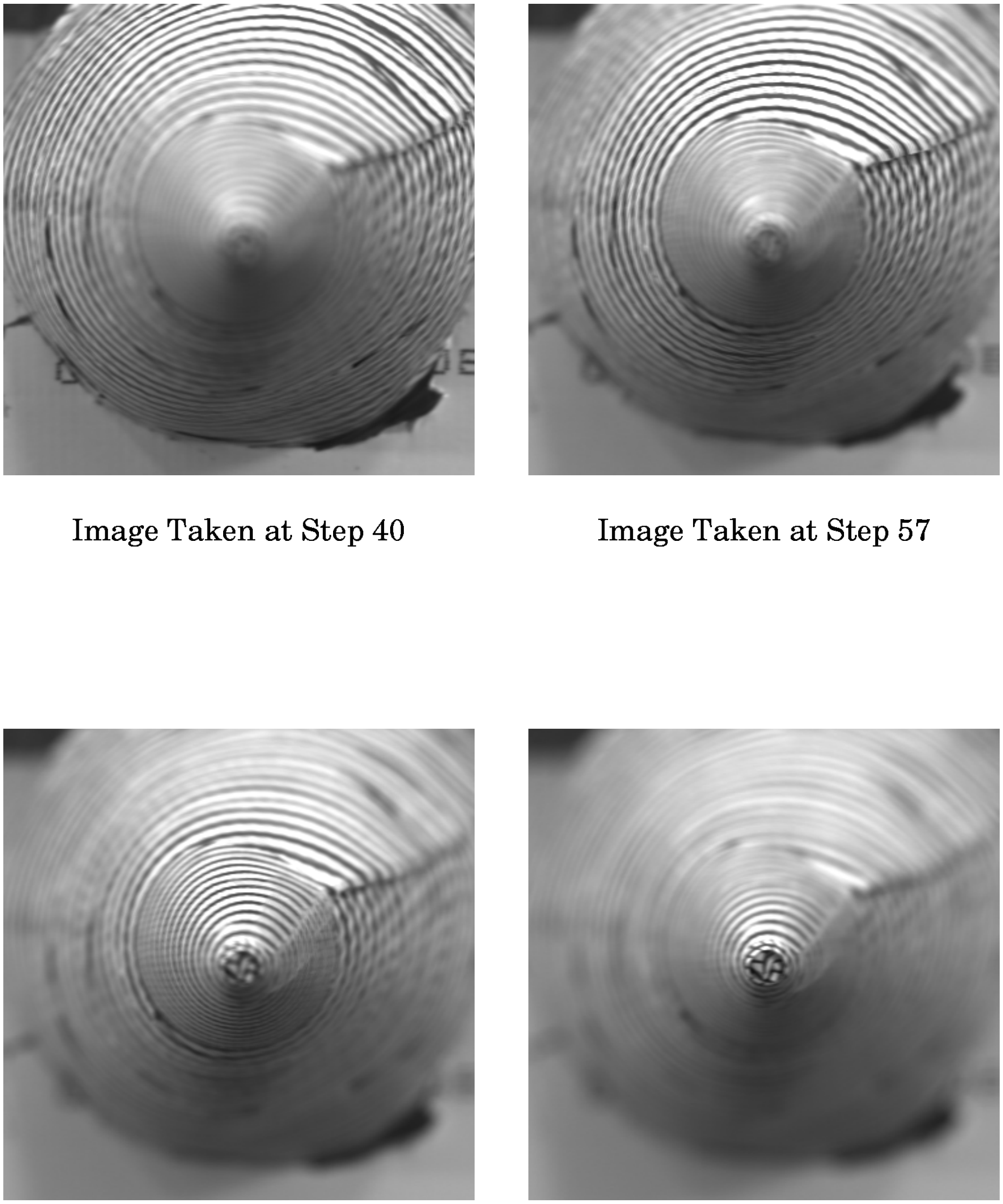


Image Taken at Step 40

Image Taken at Step 57

Image Taken at Step 74

Image Taken at Step 91

Figure 3.1: Images of Cone Object Taken at Different Steps

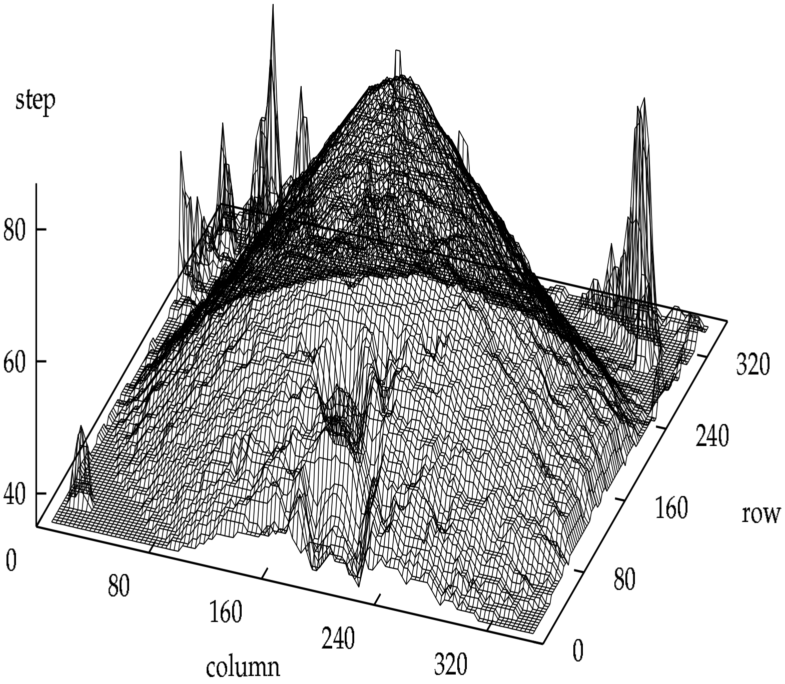


Figure 3.2: Step Map for Cone Object

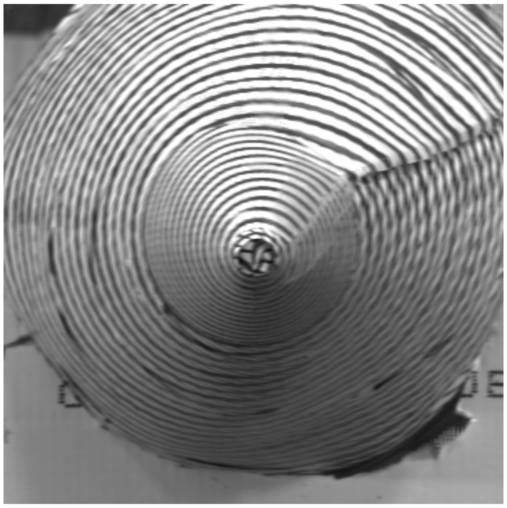


Figure 3.3: Reconstructed Image for Cone Object

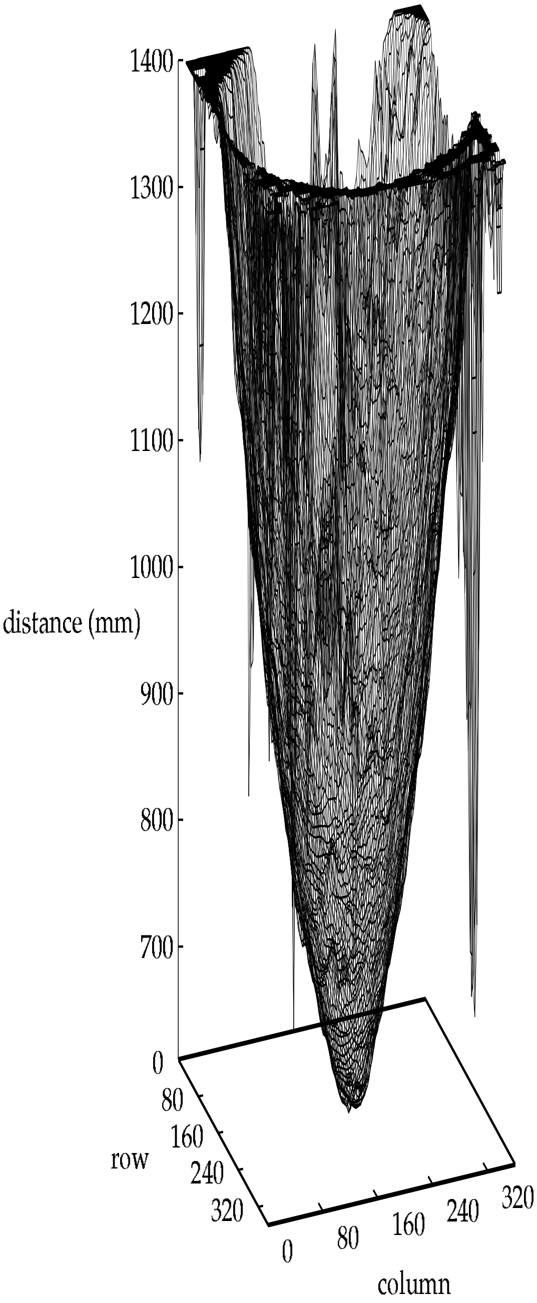


Figure 3.4: Depth Map for Cone Object

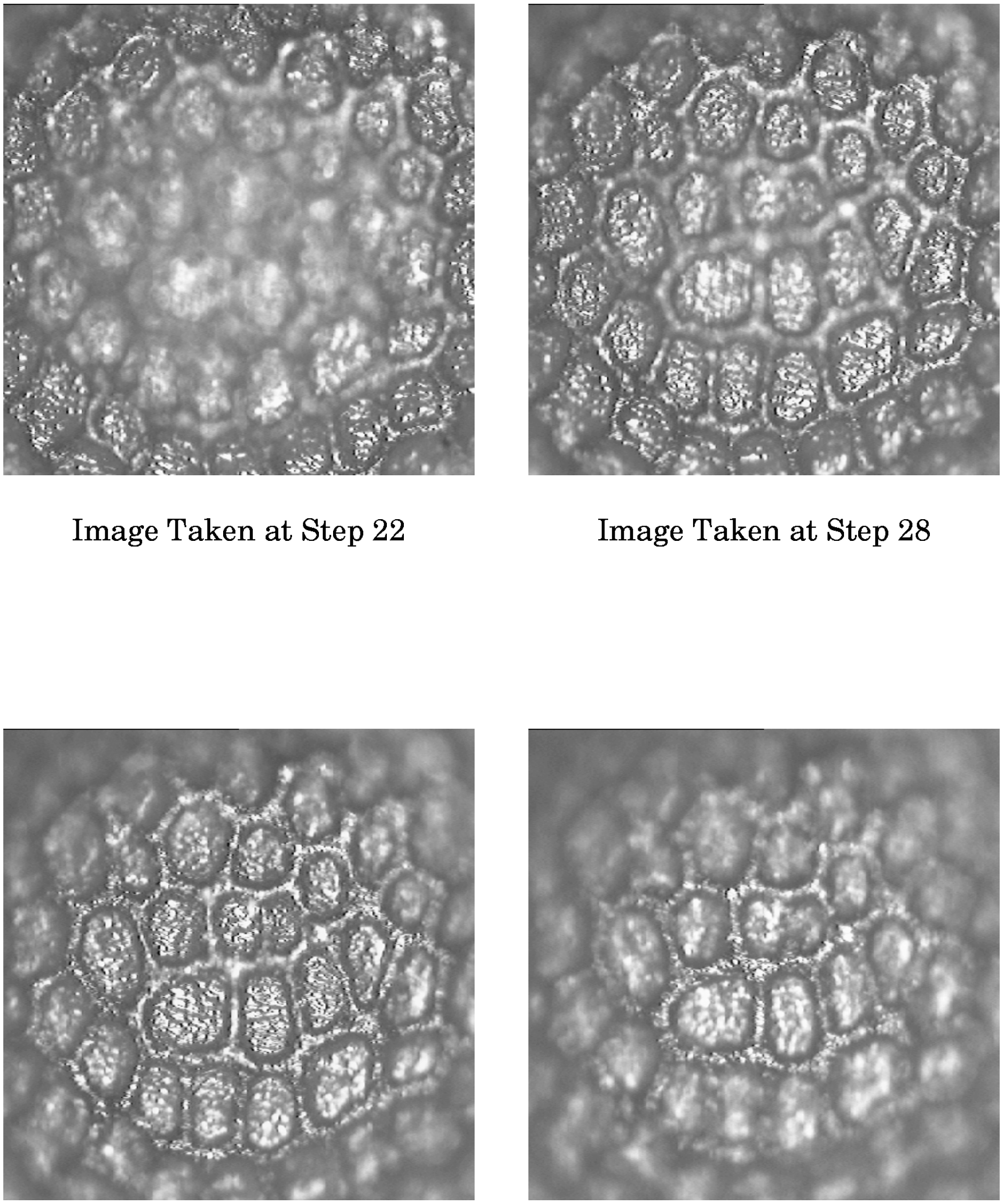


Image Taken at Step 22

Image Taken at Step 28

Image Taken at Step 34

Image Taken at Step 40

Figure 3.5: Images of Mustard Seed Taken at Different Steps

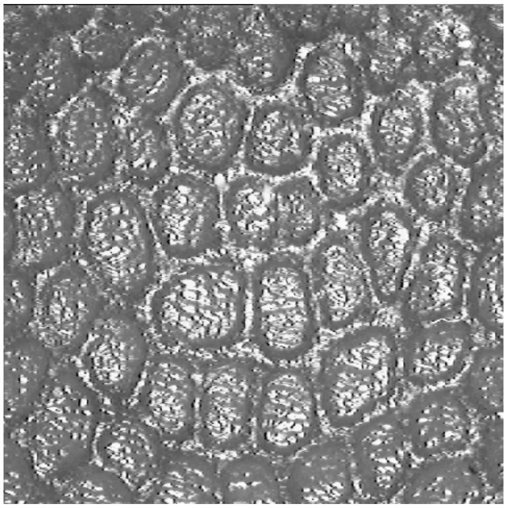


Figure 3.6: Reconstructed Focus Image

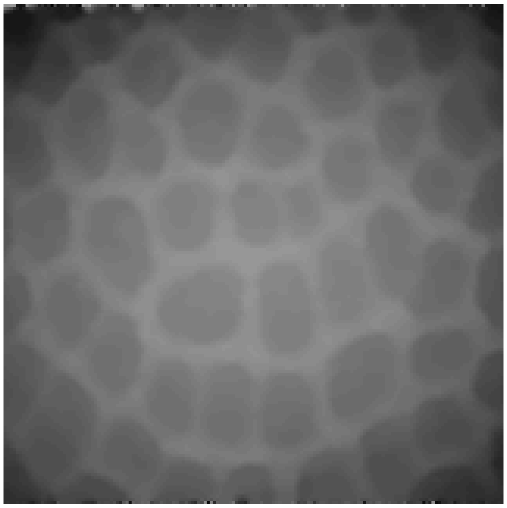


Figure 3.7: Step Map

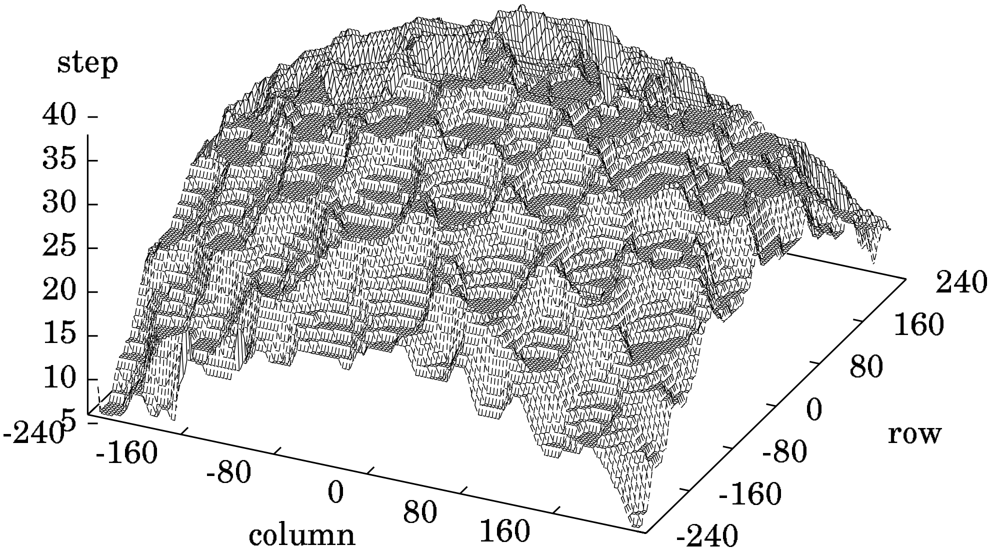


Figure 3.8: 3D Profile of the Mustard Seed

3.6 Conclusion

The advantage of Depth-from-Focus over Depth-from-Defocus is that it gives more accurate results. On the other hand, it takes more images and is slower than DFD methods. It is often the case that the mechanical movement of the camera lens takes up more time than computation itself. Therefore, the more images needed implies the more time wasted in the movement of the lens. However, DFF and DFD share some common advantages over stereo. They don't have to match two images, i.e., no correspondence problem. And the hardware setup is simpler, only a single camera is needed. The accuracy among these three methods can be ranked as, starting with the most accurate one, stereo then Depth-from-Focus and Depth-from-Defocus comes in last.

Chapter 4

Depth From Defocus Review

4.1 Introduction

In Depth-from-Focus (DFF) approach [?, ?, ?, ?, ?], a search is made for the lens position s or/and the focal length f which brings a stationary object to focus. This involves acquiring about 10 images with different s or/and f and finding the image which is in best focus. This approach is slow due to the mechanical motion of camera parts to change s or/and f to record the required 10 or so images.

In Depth-from-Defocus (DFD) approach [?, ?, ?, ?, ?, ?, ?, ?] there is no need to search for s or/and f values which correspond to focusing the object. The level of defocus is used in determining distance. This approach involves processing only a few images (about two or three) as compared to a large number (about 10) of images in the DFF methods. In addition, only a few images are sufficient to determine the distance of all objects in a scene irrespective of whether the objects are focused or not. Therefore this method is much faster than DFF due to the reduction in the mechanical motion of camera parts.

The DFD approach has been successfully applied to high contrast step edges [?, ?, ?, ?, ?]. Application of DFD to arbitrary objects has been investigated by many researchers [?, ?, ?, ?, ?, ?, ?]. In this chapter we will review some of the DFD methods.

4.2 Edge Based Method

Pentland [?, ?, ?] and Grossman [?] both addressed the problem of recovering depth from blurred edges. Pentland modeled the point spread function by the Gaussian model. The blurring process is therefore a convolution of a step edge and a Gaussian function. 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

$$\ln\left[\frac{b}{\sqrt{2\pi}\sigma^3}\right] - \frac{x^2}{2\sigma^2} = \ln\left|\frac{C(x,y)}{x}\right| \tag{4.1}$$

where b is the magnitude of the step edge and the center of the image coordinate system is located on the edge with x axis perpendicular to the edge. The two unknowns b and σ are solved by a regression in x^2. The depth is then computed from σ. The depth of edge is classified into three categories, small, medium and large.

Grossman used edges placed at different depth. The measured width of the edge is used as an index for the focus, therefore the depth. Experimental results are shown in his paper, but he does not provide a theoretical justification for his method.

Subbarao and Gurumoorthy [?] presented a closed-form solution for the problem. In this approach, the point spread function need not be a Gaussian. They only assume the point spread function to be circularly symmetric. For a step edge along the y-axis

$$f(x,y) \; = \; a + bu(x) \tag{4.2}$$

where $u(x)$ is the unit step function and b the height of the step edge. Let $f(x,y)$ be blurred by a PSF $h(x,y)$ that results in the blurred edge $g(x,y)$. It can be expressed as a convolution as in Eq. (??), $g(x,y) = h(x,y) * f(x,y)$. The definition of line spread function $l(x)$ along the y-axis can be written as

$$l(x) \; = \; \int_{-\infty}^{\infty} h(x,y)dy \tag{4.3}$$

The relation between the derivative of $g(x,y)$ along x direction and the line spread function $l(x)$ is

$$\frac{\partial g(x,y)}{\partial x} \; = \; b \, l(x) \tag{4.4}$$

Therefore, one can obtain the line spread function from a blurred edge

$$l(x) \; = \; \frac{\frac{\partial g(x,y)}{\partial x}}{\int_{-\infty}^{\infty} \frac{\partial g(x,y)}{\partial x} dx} \tag{4.5}$$

After the line spread function is obtained, the spread parameter σ_l is computed. They also showed that the spread parameter σ_l is inversely proportional to depth u. Combining Eq. (??) and Eq. (??) they obtain

$$\sigma_l = mu^{-1} + c \tag{4.6}$$

where m and c are some constants that depend on the camera settings. This formula can be used to find the depth from a blurred step edge once the spread parameter of the line spread function is estimated.

Lai, Fu and Chang [?] expressed the spread parameter σ_l in terms of the x and y components as

$$\sigma_l = \frac{\sigma_x \sigma_y}{\sqrt{\sigma_x^2 + \sigma_y^2}} \tag{4.7}$$

In this approach, they first search the image for edges. After the edges are located, σ_x and σ_y are estimated separately using numerical iteration. Then σ_l is used to find the distance.

4.3 General Scene

Previous section described methods based on edges. These methods can not be applied in situations where no edge can be found. Many researchers address this problem with different approaches. We will give a brief description of some approaches in this section.

Pentland [?, ?] used a pin-hole camera to obtain the focused image. The Gaussian point spread function is used in his analysis. If two images of the scene are taken as

$$\begin{aligned} f_1(r,\theta) &= f_0(r,\theta) * G(r,\sigma_1) \qquad (4.8) \\ f_2(r,\theta) &= f_0(r,\theta) * G(r,\sigma_2) \end{aligned}$$

where $G(r,\sigma)$ is a two dimensional Gaussian function with variance σ^2, $f_0(r,\theta)$ is the focused image, $f_1(r,\theta)$ and $f_2(r,\theta)$ are the blurred images. Taking the

Fourier transform and dividing one by the other, and applying the natural log

$$\ln \frac{\sigma_2^2}{\sigma_1^2} + 2\lambda^2\pi^2(\sigma_2^2 - \sigma_1^2) = \ln F_1(\lambda) - \ln F_2(\lambda) \tag{4.9}$$

Since one of the image is taken by a pin-hole camera, $\sigma_1 = \epsilon$ for some small value. The following relation is derived

$$k_1\sigma_2^2 + k_2 \ln \sigma_2 + k_3 = \ln F_1(\lambda) - \ln F_2(\lambda) \tag{4.10}$$

where k_1, k_2 and k_3 are constants. The equation shows that the difference in localized Fourier power is a monotonic increasing function of the blur in the second image.

Implementation of this method involves first convolve the image with an 8×8 Laplacian filter, squaring the values, then convolving again with an 8×8 Gaussian filter. Values from the two images are compared to a lookup table to find the depth information. In his experiment, a measured standard error of 6% is reported at the speed of up to eight frames per second. However, this method changes only one of the camera parameters D and requires two dimensional convolution.

Ens and Lawrence [?] proposed a matrix based regularization method. Consider two blurred images taken with change in camera parameter D. The blurring process is therefore,

$$\begin{aligned} g_1(x,y) &= f(x,y) * h_1(x,y) \\ g_2(x,y) &= f(x,y) * h_2(x,y) \end{aligned} \tag{4.11}$$

He defines a convolution ratio $h_3(x,y)$ of the two defocus operators $h_1(x,y)$ and $h_2(x,y)$, such that

$$g_1(x,y) * h_3(x,y) = g_2(x,y) \tag{4.12}$$

He argued that $h_3(x, y)$ must belong to a family of patterns that can be known a priori and $h_3(x, y)$ gives unique depth indication. Regularization was used to minimize the functional

$$||[g_{1BT}] \cdot h_{3S} - g_{2S}||^2 + \lambda||[C] \cdot h_{3S}||^2 = \text{minimum} \tag{4.13}$$

where $[g_{1BT}]$ is $g_1(x, y)$ in Toeplitz matrix form, h_{3S} is $h_3(x, y)$ in vector form, g_{2S} is the vector form of $g_2(x, y)$, λ is a scale parameter, and $[C]$ is a matrix minimizing the magnitude of the second term if h_{3S} belongs to the expected family of patterns. The Euler equation for Eq. (??) is solved for h_{3S} as

$$h_{3S} = \left([g_{1BT}]^T[g_{1BT}] + \lambda[C]^T[C]\right)^{-1}[g_{1BT}]^T g_{2S} \tag{4.14}$$

However, solving for h_{3S} is computationally expensive and $[C]$ is difficult to find except for simple parametric families. In stead, $\hat{h}_3(x, y)$ is computed iteratively to minimize

$$\sum_{x=0}^{N-k}\sum_{y=0}^{N-k}\left[g_1(x, y)[\otimes]\hat{h}_3(x, y) - g_2(x, y)\right]^2 = \text{minimum} \tag{4.15}$$

The operator $[\otimes]$ designates restricted convolution, where the border of $\hat{h}_3(x, y)$ are not convolved past the borders of $g_1(x, y)$. The size of $g_1(x, y)$ is $N \times N$, size of $\hat{h}_3(x, y)$ is $k \times k$, and size of $g_2(x, y)$ is $(N - k + 1) \times (N - k + 1)$. Implementation of the method is to have all possible $h_3(x, y)$ prestored in a table. For any two images $g_1(x, y)$ and $g_2(x, y)$, Eq. (??) is used to seek out the $h_3(x, y)$ that gives the minimum.

This method is iterative, it is therefore computationally intensive. Choosing of the free parameters can be tricky, and the PSF has to be calibrated a priori. However, his experiments shows a RMS error of 1.3% in terms of distance from the camera for object range 80 cm to 90 cm.

Subbarao and Surya [?, ?] presented a spatial domain approach using S-Transform. According to the third order S-transform [?], it is shown that the blurred image can be expressed as

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

where $h_{m,n}$ are the moments of the point spread function, and $f^{m,n}(x,y)$ are the derivatives of $f(x,y)$.

$$\begin{aligned} h_{m,n} &= \int_{\infty}^{\infty}\int_{\infty}^{\infty} x^m y^n h(x,y)dxdy \qquad (4.17) \\ f^{m,n}(x,y) &= \frac{\partial^m}{\partial x^m}\frac{\partial^n}{\partial y^n} f(x,y) \qquad (4.18) \end{aligned}$$

And the focused image $f(x,y)$ can be found from the blurred image by

$$\begin{aligned} f(x,y) &= g(x,y) - \frac{h_{2,0}}{2} \bigtriangledown^2 g(x,y) \qquad (4.19) \\ &= g(x,y) - \frac{\sigma_h^2}{4} \bigtriangledown^2 g(x,y) \qquad (4.20) \end{aligned}$$

The image is first filtered by a smoothing filter proposed by Meer and Weiss [?]. This filter can be applied separately along x and y directions to fit a cubic polynominal in a small image region. The size for their implementation is 9 × 9. Then third order S-Transform can be used. For two images with blurring kernel spread parameters σ_1 and σ_2, the following relation can be derived

$$g_1(x,y) - g_2(x,y) = \frac{1}{4}(\sigma_1^2 - \sigma_2^2) \bigtriangledown^2 g \tag{4.21}$$

Therefore, the value $(\sigma_1^2 - \sigma_2^2)^2$ can be estimated by

$$(\sigma_1^2 - \sigma_2^2)^2 = 16\frac{\int\int (g_1 - g_2)^2 dxdy}{\int\int (\bigtriangledown^2 g)^2 dxdy} \tag{4.22}$$

This can be pre-stored through camera calibration and then used later as a lookup table for finding the object distance. Experimental results on this method yield an RMS error of 2.3% in the range from 50 centimeter to 5 meter. The drawback of this method is it requires two dimensional operation.

Chapter 5

Depth From Defocus Using One-dimensional Fourier Coefficients (DFD1F)

5.1 Introduction

A new fast method of determining distance of objects and autofocusing a camera using image defocus information is presented in this chapter. The method, named DFD1F, requires only two images acquired at two different camera parameter settings and therefore is very fast in comparison with depth-from-focus [?] methods which require a large number (10 or more) of images.

The major distinguishing features of DFD1F in comparison with prior DFD approaches are- (i) it requires the computation of only a few (about 6) and that too only one-dimensional Fourier coefficients (hence the suffix 1F in DFD1F), (ii) it is general in that it is not restricted to any particular model of the point spread function of the camera system, (iii) only a few (two or three) images acquired with different camera parameter settings are needed, (iv) there is no restriction on the camera parameter settings such as pin-hole aperture, etc., and (v) the method has been demonstrated on a very large database of planar images.

DFD1F has been implemented on a prototype camera system named SPARCS. It can determine the distance of an object placed in front of the camera in the range 0.6 meter to infinity in less than a second of computation on a personal computer. Based on the computed distance, the camera can autofocus by moving the lens to the correct position. Experiments indicate that the method is useful in practical applications such as robotic vision and rapid autofocusing.

5.2 Theoretical Basis

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) (Subbarao and Nikzad, 1990).

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 [?].

For a planar object perpendicular to the optical axis, the blur circle radius R' is a constant over the image of the object (this may not be obvious at first sight, but it can be proved easily). 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. Convolution in the spatial domain corresponds to multiplication in the Fourier domain. Therefore, if F and G_i are Fourier transforms of f and g_i respectively,

$$G_1(\omega,\nu) \;=\; H_a(\omega,\nu;\mathbf{e_1},u)\;F(\omega,\nu) \qquad \text{and} \tag{5.2}$$

$$G_2(\omega,\nu) \;=\; H_a(\omega,\nu;\mathbf{e_2},u)F(\omega,\nu). \tag{5.3}$$

The effect of the focused image F is eliminated by dividing G_1 by G_2 to obtain

$$\frac{G_1(\omega,\nu)}{G_2(\omega,\nu)} \;=\; \frac{H_a(\omega,\nu;\mathbf{e_1},u)}{H_a(\omega,\nu;\mathbf{e_2},u)}\,. \tag{5.4}$$

with $H_a(\omega,\nu;\mathbf{e},u)$ defined in Eq. (??).

In the above equation, the distance u is the only unknown quantity. Therefore we can solve the above equation to find the distance. Indeed, we can obtain an equation similar to the above irrespective of the PSF model. It is not restricted to any particular model of PSF such as the one based on paraxial geometric optics, or on a Gaussian PSF. We will next discuss methods for solving this equation. First we consider two special cases, and then the general case.

5.2.1 Paraxial Geometric Optics Model for PSF

Substituting for the right hand side of Eq. (??) from Eq. (??), we obtain

$$\frac{G_1(\omega,\nu)}{G_2(\omega,\nu)} = \frac{J_1(R'(\mathbf{e_1};u)\ \rho(\omega,\nu))\ R'(\mathbf{e_2};u)}{J_1(R'(\mathbf{e_2};u)\ \rho(\omega,\nu))\ R'(\mathbf{e_1};u)}. \tag{5.5}$$

The left hand side of the above equation is computed from the recorded images. Explicit closed-form solution for u in the above equation is difficult to obtain because of the presence of the Bessel function. But it can be solved easily using numerical techniques.

5.2.2 Gaussian PSF Model

Interestingly, a closed-form solution for u can be obtained in the case of a Gaussian PSF model. In this case we have

$$\frac{G_1(\omega,\nu)}{G_2(\omega,\nu)} = e^{-\frac{1}{2}(\omega^2+\nu^2)(\sigma_1^2-\sigma_2^2)} \tag{5.6}$$

Taking logarithm on either side and rearranging terms, we get

$$\sigma_1^2 - \sigma_2^2 = \frac{-2}{\omega^2+\nu^2}\ \ln\left(\frac{G_1(\omega,\nu)}{G_2(\omega,\nu)}\right). \tag{5.7}$$

For some (ω,ν), the right hand side of equation (??) can be computed from the given image pair. Therefore equation (??) can be used to estimate $\sigma_1^2-\sigma_2^2$ from the observed images. Measuring the Fourier transform at a single point (ω,ν) is, in principle, sufficient to obtain the value of $\sigma_1^2-\sigma_2^2$, but a more robust estimate can be obtained by taking the average over some domain in the frequency space. Let the estimated average be C given by

$$C = \frac{1}{A}\int\int_B \frac{-2}{\omega^2+\nu^2}\ \ln\left(\frac{G_1(\omega,\nu)}{G_2(\omega,\nu)}\right)\ d\omega\ d\nu \tag{5.8}$$

where B is a region in the (ω, ν) space not containing points where $G_1(\omega, \nu) = G_2(\omega, \nu)$, and A is the area of B. Therefore, from the observed images we get the following constraint between σ_1 and σ_2:

$$\sigma_1^2 - \sigma_2^2 \; = \; C \, . \tag{5.9}$$

Equation (??) (and therefore Eq. (??) is a single equation in two unknowns σ_1 and σ_2. Therefore, this equation can not be solved without using an additional constraint. Pentland [?] solved this equation by forcing σ_2 to be zero (or nearly zero). Forcing σ_2 to zero corresponds to requiring the knowledge of the focused image of the scene. In his experiments, Pentland obtained this information by setting the camera aperture to be very small (pin-hole dimensions). However, a very small aperture has two main problems: (i) it increases the camera exposure period to a larger duration, and (ii) it increases diffraction effects which distort the image.

One of the important contribution of this work is the recognition that an additional constraint exists between σ_1 and σ_2 in terms of the camera parameters $\mathbf{e_1}$ and $\mathbf{e_2}$. It is the use of this constraint that removes the requirement of a focused image or other information and makes our method sound and useful in real-time applications such as autofocusing. As an alternative to this constraint, Ens and Lawrence [?] adopted a heuristic approach where the PSF for an inverse filter was forced to be smooth (regularization approach). The following linear relation between σ_1 and σ_2 can be obtained in terms of the known camera parameters using Eq. (??) and eliminating $1/u$:

$$\sigma_1 \; = \; \alpha \sigma_2 + \beta \tag{5.10}$$

where

$$\alpha = \frac{D_1}{D_2} \quad \text{and} \quad \beta = \frac{cD_1}{2}\left(\frac{1}{f_1} - \frac{1}{f_2} + \frac{1}{s_2} - \frac{1}{s_1}\right). \tag{5.11}$$

Equations (??,??) together constitute two equations in two unknowns. From these equations we obtain

$$(\alpha^2 - 1)\,\sigma_2^2 \,+\, 2\alpha\beta\,\sigma_2 \,+\, \beta^2 \,=\, C. \tag{5.12}$$

Above we have a quadratic equation in σ_2 which is easily solved. In general there will be two solutions. However a unique solution is obtained if $D_1 = D_2$. We can also derive other special cases where a unique solution is obtained (e.g.: $D_1 \neq D_2$, $s_1 = s_2 = f_1 = f_2$; in this case only the negative solution of σ is acceptable which is unique). Having solved for σ_2 we obtain the distance u from equation (??). Thus, the distance is determined from only two images obtained with different camera parameter settings. This should be compared to the DFF methods [?] which require recording and processing a large number of images. Note that the camera parameter setting could differ in any one, any two, or all three of the parameters: s, f, D.

Equation (??) plays a central role in determining distance. As noted earlier, it is not restricted to any particular model of PSF (not even the assumption of circular symmetry of the PSF is required). The left hand side of Eq. (??) can be computed from the observed images g_1 and g_2. The right hand side can be expressed as an analytic function as in Eq. (??) or Eq. (??), if one uses an analytic model for the PSF. In this case, standard numerical techniques can be used to solve the equation. However, for practical camera systems, the function on the right hand side is usually very complicated. In such cases, the right hand side can be represented by a table of values obtained by initial camera calibration. Then, solving the equation corresponds to searching the table to find a position where the stored value is nearly the same as the computed left hand side value. The position found gives the distance of the object.

5.3 Engineering the Implementation

The DFD method described above was implemented by us on our camera system. However, the method was found to be unreliable for several reasons. These reasons prompted us to make some important improvements to the original method. We will discuss the reasons and the consequent modifications next.

5.3.1 Choice of Fourier Coefficients

The noise characteristic of our camera is apparent from Fig. 5.1. It shows the picture of a uniform planar surface having constant reflectance and illumination. The noise introduced by our camera appears to have two components. One is a regular vertical sine-wave pattern with a period of about 8 pixels, and another is a zero-mean random component. The presence of these noise components degrades the performance of our original method very significantly.

In order to overcome the above difficulty, we decided to use only the most robust Fourier coefficients in determining distance. It should be noted that, in principle, measurement of a single Fourier coefficient may be sufficient to determine the distance. However, in practice, one needs to measure several (about 6 in our experiments) Fourier coefficients for a robust estimation of distance. For the noise characteristics of our camera, it can be shown that the Fourier coefficients along the vertical axis in the Fourier domain are the most reliable (see Fig. 5.1).

Using only the Fourier coefficients along the vertical axis also results in significant computational advantages. It is shown here that these coefficients can be computed by first summing the pixels along rows, and then computing one-dimensional Fourier coefficients. The application of a one-dimensional Fourier transform instead of a two-dimensional

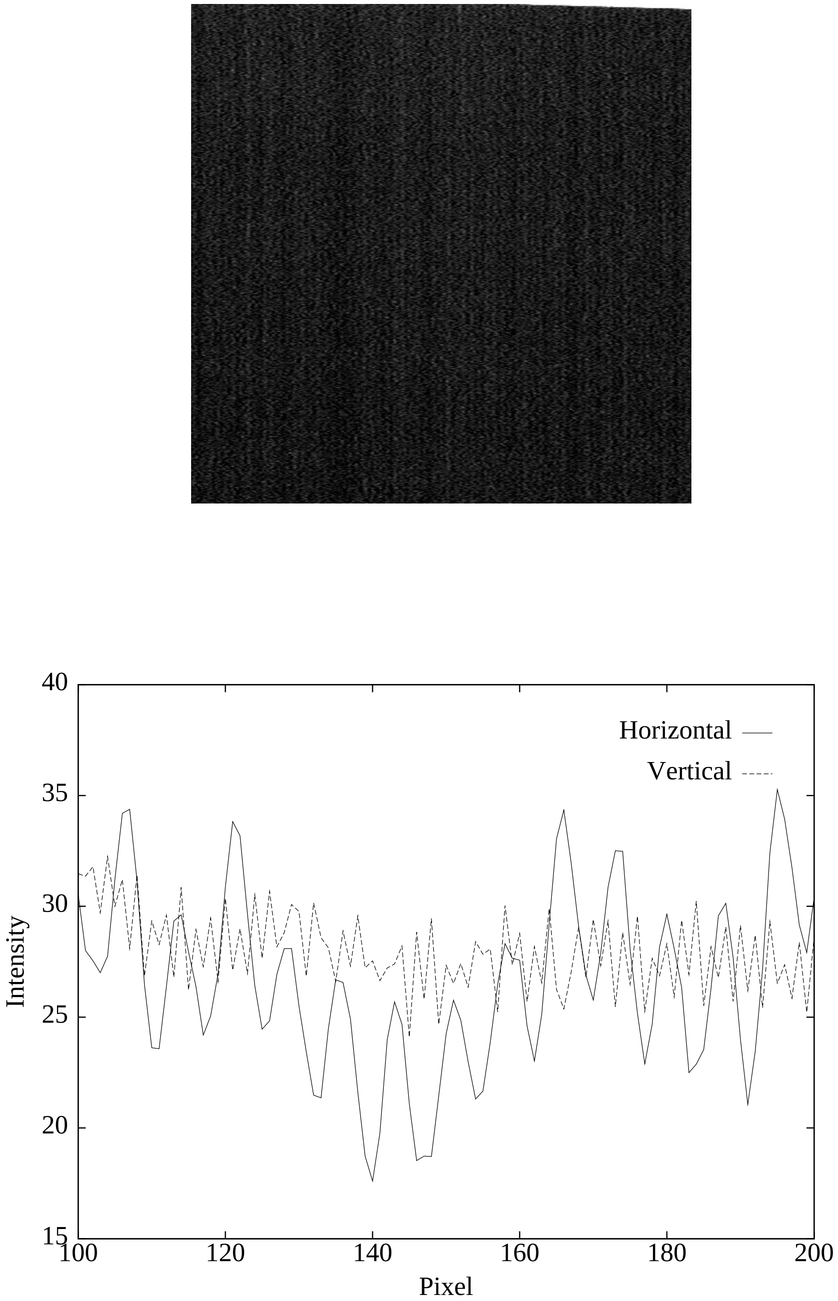


Figure 5.1: Noise Characteristics

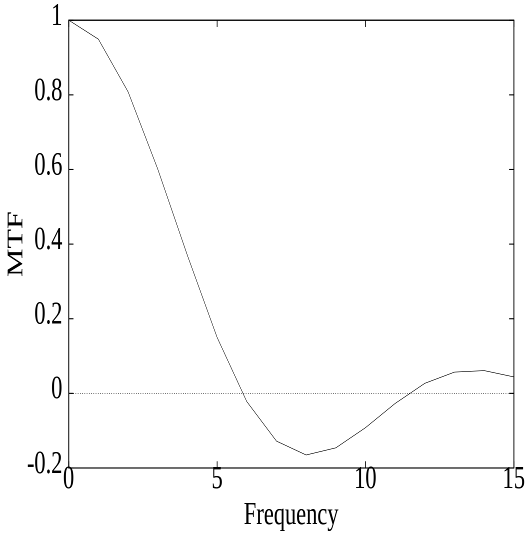


Figure 5.2: Example of Actual OTF

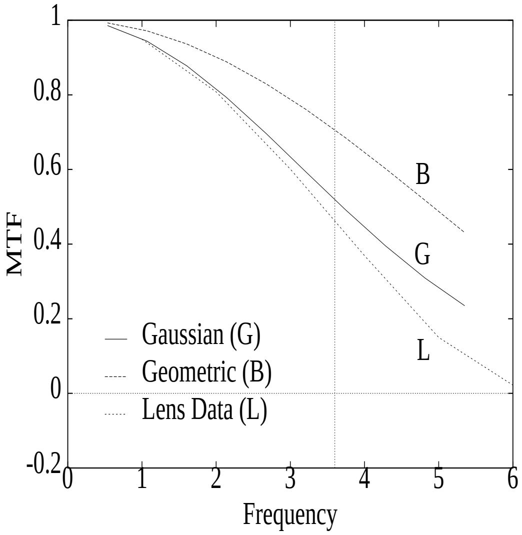


Figure 5.3: MTF

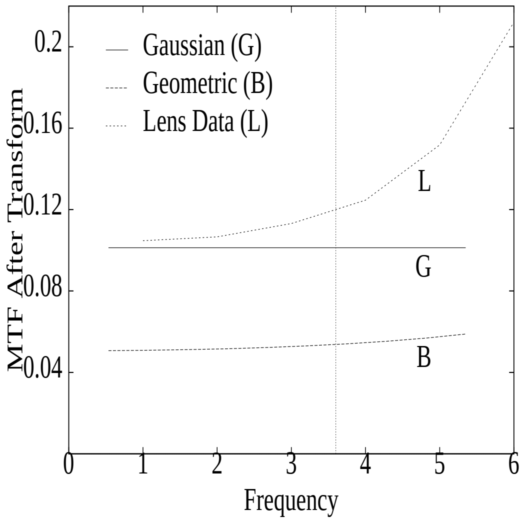


Figure 5.4: MTF after log/ρ^2 Transform

Fourier transform results in tremendous computational savings.

For any image $g(x,y)$, let $g_l(y)$ be the result of integrating $g(x,y)$ with respect to x, i.e.

$$g_l(y) = \int_{-\infty}^{\infty} g(x,y)dx \tag{5.13}$$

Let $G(\omega,\nu)$ and $G_l(\nu)$ be the Fourier transforms of $g(x,y)$ and $g_l(y)$ respectively. Then it can be shown that $G(0,\nu) = G_l(\nu)$. Therefore, the two-dimensional Fourier coefficients along the vertical (ν) axis defined by $G(0,\nu)$ can be computed by first summing the image $g(x,y)$ along the x-axis to obtain $g_l(y)$ and then computing the one-dimensional Fourier coefficients $G_l(\nu)$. A similar result can be derived for the case of Discrete Fourier Transform which is the one used in our implementation. It is also possible to use the Fourier coefficients along the horizontal axis, however, this requires a filtering step to remove the vertical periodic noise in our camera.

If the spatial frequency spectrum of an image is localized in a small region in the two-dimensional Fourier space, then, instead of a fixed set of axes (e.g. vertical and/or horizontal) one has to compute Fourier coefficients along an axis which passes through the region. In the worst case, finding such a region may involve the computation of all 2D Fourier coefficients of the image. With this extra computation, DFD1F can be used for all images with some spatial frequency content.

Video images are scanned by rows rather than columns. Therefore, instead of summing of row pixels after digitization, one can first integrate the analog video signal rowwise using an analog integrator and then digitize the result to obtain the required one-dimensional sequence. This saves both digitization hardware and digital summation hardware. From this point of view also, using vertical Fourier coefficients is better than using horizontal Fourier coefficients.

5.3.2 Table Searching

The next important improvement to the original DFD method was in the table searching step. Here is the problem. Suppose that a table of values corresponding to the right hand side of Equation (??) is given, and a set of 6 values (each one computed at a specific spatial frequency along the vertical axis) corresponding to the left hand side of the same equation are given. Now, given that the set of 6 values is noisy, what is the best strategy for searching the table to find the distance u? We tried many approaches to this problem which failed to perform satisfactorily on our camera system. Through much trial and error, we finally engineered the following method which has been implemented and is found to perform well. This method, to our satisfaction, has a theoretical justification.

Let ρ be the spatial frequency (along the vertical axis). Now a table corresponding to the right hand side of Eq. (??) specified by $T_s(\rho, u)$ is given ($\mathbf{e}_1$ and $\mathbf{e}_2$ are fixed). Also, corresponding to the left hand side of Eq. (??), 6 computed values specified by $T_c(\rho)$, $\rho = 1, 2, \cdots, 6$ are given. The problem is to find the value of the index u in the table $T_s(\rho, u)$ where all the six computed values are nearly equal to the corresponding values in the table $T_c(\rho)$. The first idea was to use a simple minimum-mean square error (MSE) method. But this did not succeed because of the nature of the Optical Transfer Function (OTF) of the camera whose cross-section looks like a sinc function (see Fig. 5.2).

The MSE method relied too much on information in high frequencies which were more noisy than the low frequencies. Therefore our next idea was to try a weighted minimum-mean square error (WMSE) approach which would almost equally emphasize information in both low frequencies and high frequencies. Many weighting schemes were tried, but none of them performed satisfactorily.

Finally, we devised a scheme where $T_s(\rho, u)$ and $T_c(\rho)$ were defined as

$$T_s(\rho, u) = \frac{-2}{\rho^2} \ln \frac{|H(\rho; \mathbf{e_1}, u)|}{|H(\rho; \mathbf{e_2}, u)|} = \frac{-2}{\rho^2} \ln |H(\rho; \mathbf{e_1}, u)| + \frac{2}{\rho^2} \ln |H(\rho; \mathbf{e_2}, u)| \tag{5.14}$$

$$T_c(\rho) = \frac{-2}{\rho^2} \ln \frac{|G_1(\rho)|}{|G_2(\rho)|} = \frac{-2}{\rho^2} \ln |G_1(\rho)| + \frac{2}{\rho^2} \ln |G_2(\rho)|. \tag{5.15}$$

MSE was computed as

$$\mathrm{MSE}(u) = \sum_{\rho} [T_s(\rho, u) - T_c(\rho)]^2 \tag{5.16}$$

The value of u for which the MSE was a minimum was taken as an estimation of the distance of the object. This method worked well. In the subsequent discussion, we will refer to this method as DFD1F.MSE.

DFD1F.MSE was derived through the following arguments. Suppose the PSF is a Gaussian, then the MTF is also a Gaussian as in Eq. (??). In this case, the variance of the PSF distribution is

$$\sigma^2 = \frac{-2}{\rho^2} \ln |H_b(\rho, u)|. \tag{5.17}$$

The variance σ^2 depends only on the camera parameters $\mathbf{e}$ and distance u as is clear from Eq. (??). In particular, it does not depend on the spatial frequency ρ. Therefore, the right hand side of Eq. (??) should be a constant (with respect to ρ because the left hand side is a constant). However, this is exactly true only for a Gaussian MTF. But if the MTF of a camera is approximately Gaussian, then there is reason to believe that the right hand side of Eq. (??) will also be approximately constant; in fact, for our purposes, it does not have to be exactly constant. It would be sufficient if the quantity does not change "too much", say by a factor of more than about 2.

It is found that for relatively low frequencies (e.g. frequencies which are less than half of the first zero-crossing) a Gaussian is generally close to the MTF of practical cameras including our camera. Fig. 5.3 shows a comparison of a Gaussian MTF, an MTF based on paraxial geometric optics, and the MTF of our camera, all corresponding to the same variance. Fig. 5.4 shows these same MTFs after the transformation specified by the right hand side of Eq. (??). We call this the $\log/\rho^2$ (log-by-rho-squared) transform as it involves first taking the log of the MTF and then dividing the result by ρ^2. In some sense, the above scheme gives approximately equal emphasis to information in the values computed at different spatial frequencies during the computation of mean square error.

The $\log/\rho^2$ transform is also very useful in another respect. In our implementation, we need to interpolate $T_s(\rho, u)$ with respect to ρ and u. Equations (??) and (??) can be used to do this. Applying the $\log/\rho^2$ transform to $H(\rho; \mathbf{e}_1, u)$ in Eq. (??), we get an expression similar to Eq. (??). As discussed above, the right hand side of this expression is nearly a constant with respect to ρ. Therefore a simple linear interpolation with respect to ρ can be used on this expression to obtain good results. More interestingly, the square root of this expression varies almost linearly with $1/u$. This behavior is predicted by Eq. (??). Therefore once again a linear interpolation can be used on the square root of the expression with respect to $1/u$. This interpolated data can be transformed to obtain the table $T_s(\rho, u)$ at small intervals of ρ and $1/u$ from data given at coarse intervals.

DFD1F.MN: In another variation of the implementation, mean of $T_c(\rho)$ and $T_s(\rho, u)$ were computed over ρ as

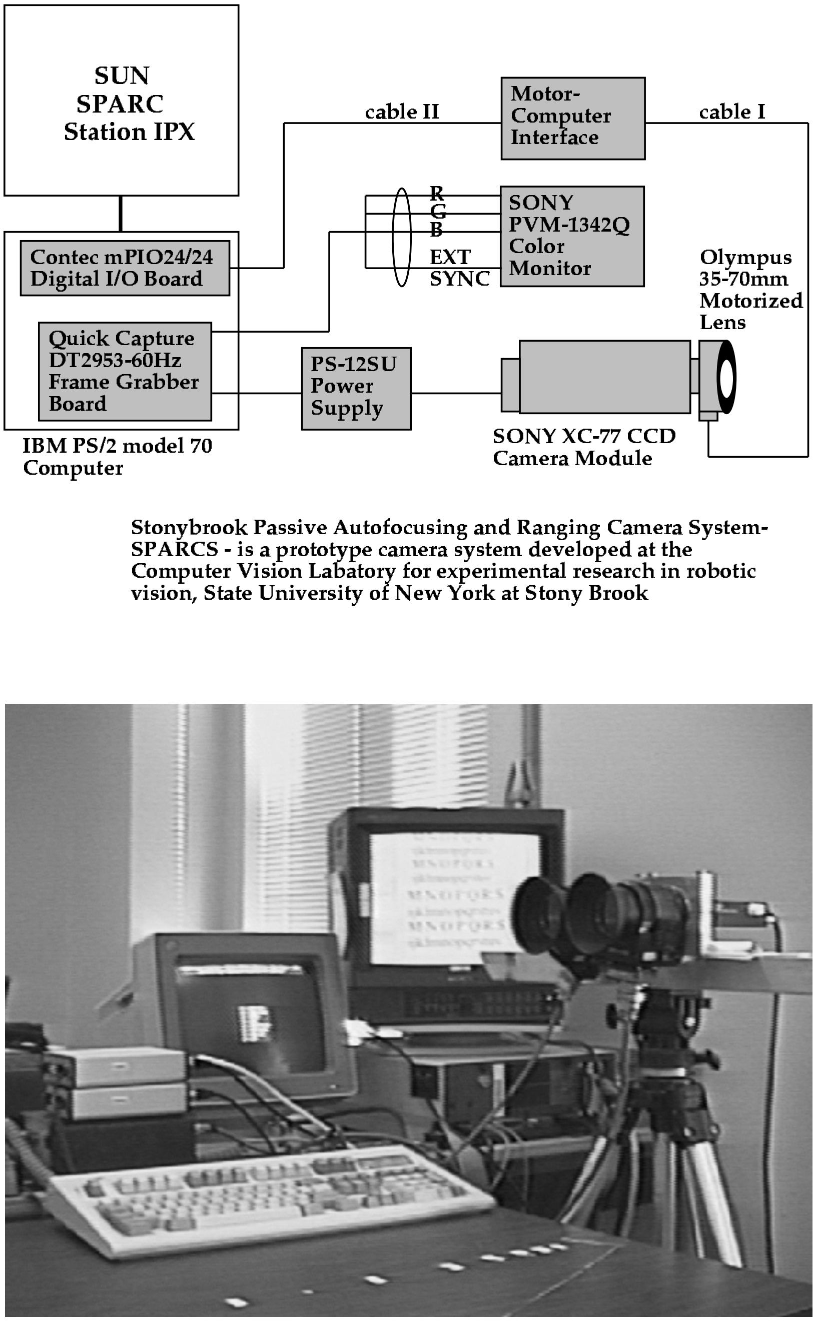
$$\overline{T}_s(u) = \frac{1}{n} \sum_{i=1}^{n} T_s(\rho_i, u) \quad \text{and} \quad \overline{T}_c = \frac{1}{n} \sum_{i=1}^{n} T_c(\rho_i) \tag{5.18}$$

Then the distance u was estimated to be that u for which $|\overline{T}_s(u) - \overline{T}_c|$ was a minimum. This method will be referred to as DFD1F.MN. Since $T_s(\rho_i, u)$ and $T_c(\rho_i)$ are almost constant with respect to ρ, this method works nearly as good as the DFD1F.MSE method.

5.4 Implementation

Our implementation does not depend on any particular model of PSF, but it is strongly guided by the PSF based on paraxial geometric optics, and the Gaussian PSF. DFD1F is implemented on a system named Stonybrook Passive Autofocusing and Ranging Camera System (SPARCS). This system was built during the last few years by our research group in the Computer Vision Laboratory, Department of Electrical Engineering, State University of New York at Stony Brook. Fig. 5.5 shows a schematic diagram and a picture of SPARCS.

SPARCS has a SONY XC-77 black/white CCD camera with a Olympus 35-70 mm motorized lens. Images from this camera are captured by a frame grabber (QuickCapture DT2953 of DATATRANSLATION). The captured images are processed by an IBM PS/2 Model 70 personal computer. The focal length of the lens can be varied manually from about 35 mm to 70 mm. The F-number (ratio of focal length f to aperture diameter D) also can be varied manually to 4, 8, 22, etc. The lens system consists of multiple lenses, and focusing is done by moving the front lens forward and backward. This lens motion can be done both manually and under computer control. The motor is a stepper motor with 97 steps numbered from 0 to 96. Step number 0 corresponds to focusing an object at distance infinity, and step number 96 corresponds to focusing a close object at around 50 cm distance. The motor is controlled by a microprocessor which can communicate with the IBM PS/2 computer. In effect, the system is set up such that, a C program running on the PS/2 can



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

Figure 5.5: SPARCS

move the lens to any desired step number and take pictures and process them. The communication between the microprocessor and the computer takes place through a digital I/O board (Contec mPIO24/24) and a motor-computer interface. The pictures from the camera can be displayed on a color monitor (SONY PVM-1342Q) in real-time. Also, pictures stored in the PS/2 computer can be displayed on the monitor.

In our experiments, the F-number was manually set to 4, and the focal length was manually set to 35 mm. These settings were not changed during the experiments. However, according to the lens data provided by the lens manufacturer to us, due to the complex nature of the optical system, the focal length changes by a small amount when the front lens is moved from one end to the other. We also believe that the effective diameter D of the entrance pupil also changes by a small amount when the front lens is moved, but we do not have lens specification data related to this.

For F-number = 4, and focal length $f = 35$ mm camera setting, Table 1 shows the distance of an object which is in best focus as a function of the lens step position. This data was obtained using a lens simulation program by the lens manufacturer. Because the F-number is small, the aperture is relatively large, and therefore this table can be shown to be different from the one predicted by the paraxial geometric optics model. However, since DFD1F uses the actual MTF of the lens instead of the one predicted by paraxial geometric optics, it performs well.

Fig. 5.6 shows a plot corresponding to Table 1 where the vertical axis is the reciprocal of distance and the horizontal axis is the lens position in motor

Lens Step	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
Distance(m)	∞	5.300	3.750	2.850	2.500	1.930	1.720	1.465	1.320	1.170	1.080	0.965	0.900	0.822	0.770	0.715	0.670	0.628	0.595	0.560

Table 5.1: Lens Step vs Best Focused Distance (Simulated Data)

step number. In Table 1 and Fig. 5.6 we observe that there is a one-to-one monotonic relation between lens position specified by a step number and the distance of an object which would be in best focus when the lens is at that position. Therefore, if an object is known to be in best focus, then its distance can be found from the table (or Fig. 5.6) using the step number of the lens position as an index into the table. Conversely, if the distance of an object is known, the lens position for which it will be focused can be found from the table (or Fig. 5.6). Since there are only 97 steps for the lens position in our camera, only about 97 distinct distances can be measured. Further, we observe in Table 1 that the distance of a best focused object decreases rapidly in the beginning and more slowly later as a function of lens position specified by the step number. In fact the two are approximately related by a reciprocal linear relation. This is in fact predicted by the lens formula (??). For these reasons, we will

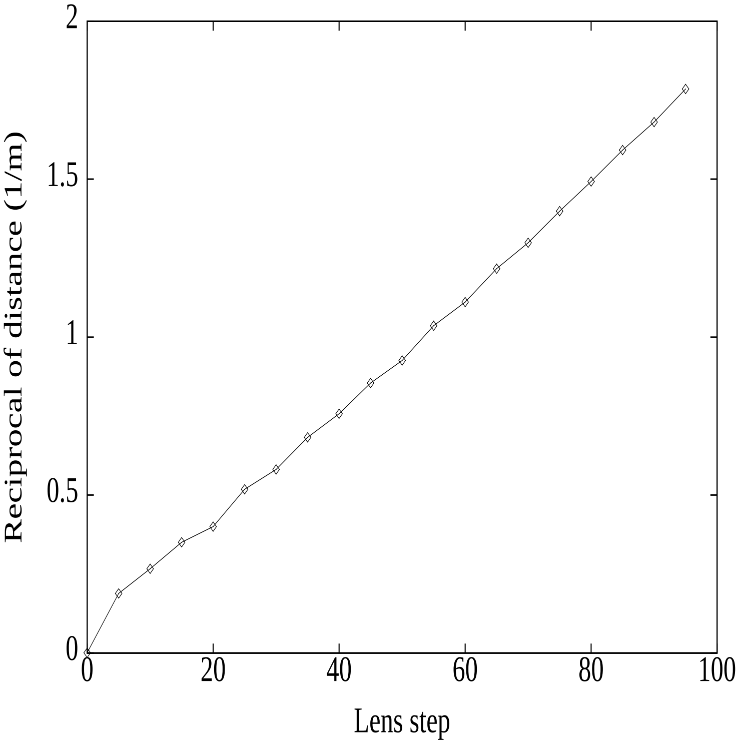


Figure 5.6: 1/u vs Lens Step

use the step numbers as units of distance in specifying the distance of an object. For example, if the distance of an object is said to be step number 35, it means that the object's distance is such that the object would be in best focus if the lens is moved to step number 35. If we assume that the distance of an object of interest has approximately a uniform probability distribution in the interval $(0, \infty)$, then the object is more likely to be focused for a lens position near step 0 than near step 96. Based on this argument, SPARCS is implemented such that the initial position of the lens is always step 0.

The overall operation of SPARCS for finding the distance and autofocusing of an object is summarized in a flow chart in Fig. 5.7. The stepwise operation is also explained below with comments. The lens is first moved to step 10 and a first image g_{10} is recorded. 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. Bright incandescent lamps are highly recommended for this reason. Under low or flicker illumination, one image frame may be substantially different from another. This was clearly evident from a number of tests on SPARCS.

The lens is then moved to step 40, and a second image g_{40} of the object is recorded. Again, optionally, several frames may be recorded and averaged.

The object to be focused is specified by specifying a region on the

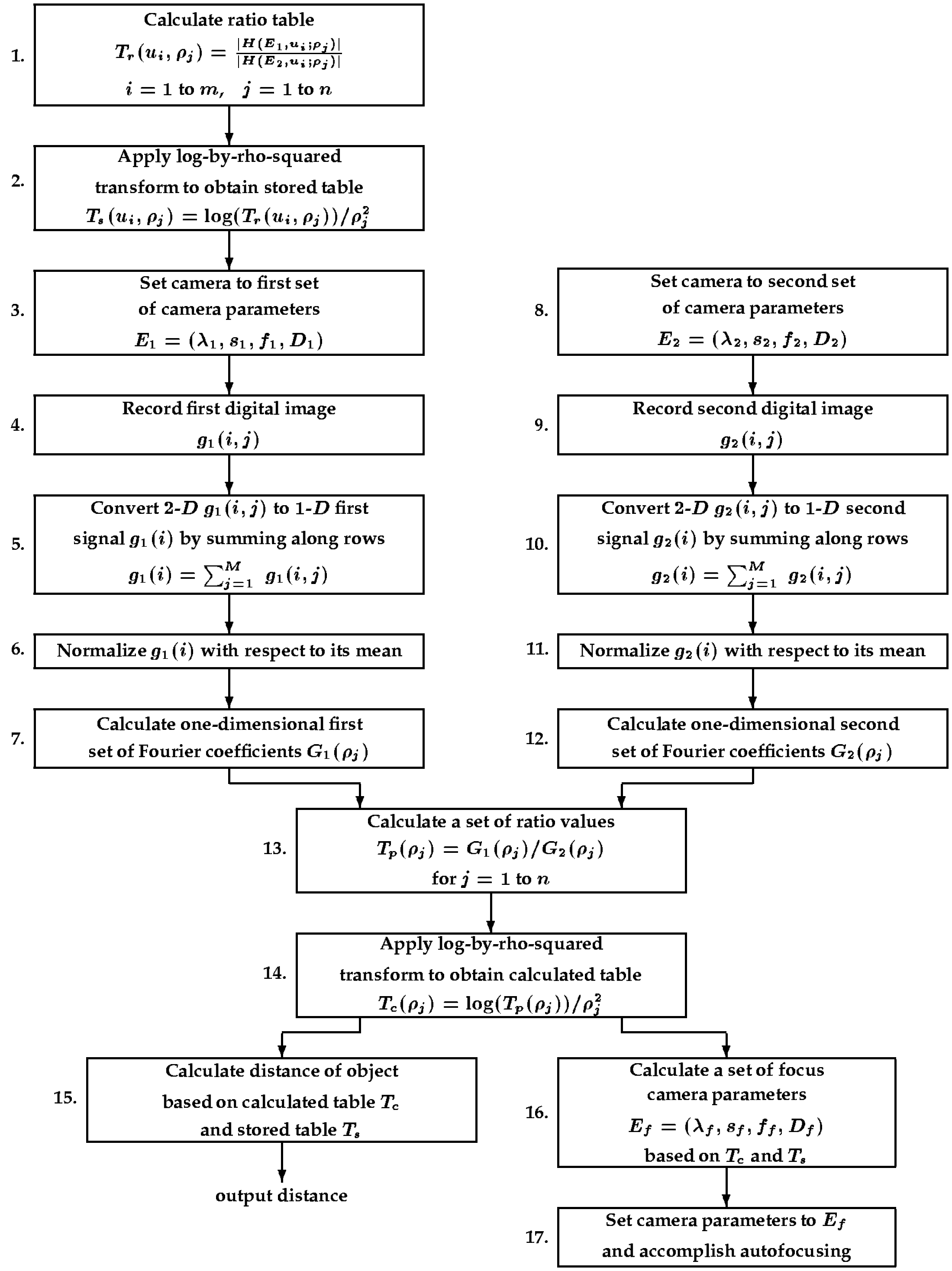

1.
Calculate ratio table
$T_r(u_i, \rho_j) = \frac{|H(E_1, u_i; \rho_j)|}{|H(E_2, u_i; \rho_j)|}$
$i = 1$ to m, $j = 1$ to n
2.
Apply log-by-rho-squared
transform to obtain stored table
$T_s(u_i, \rho_j) = \log(T_r(u_i, \rho_j))/\rho_j^2$
3.
Set camera to first set
of camera parameters
$E_1 = (\lambda_1, s_1, f_1, D_1)$
4.
Record first digital image
$g_1(i, j)$
5.
Convert 2-D $g_1(i, j)$ to 1-D first
signal $g_1(i)$ by summing along rows
$g_1(i) = \sum_{j=1}^{M} g_1(i, j)$
6.
Normalize $g_1(i)$ with respect to its mean
7.
Calculate one-dimensional first
set of Fourier coefficients $G_1(\rho_j)$
8.
Set camera to second set
of camera parameters
$E_2 = (\lambda_2, s_2, f_2, D_2)$
9.
Record second digital image
$g_2(i, j)$
10.
Convert 2-D $g_2(i, j)$ to 1-D second
signal $g_2(i)$ by summing along rows
$g_2(i) = \sum_{j=1}^{M} g_2(i, j)$
11.
Normalize $g_2(i)$ with respect to its mean
12.
Calculate one-dimensional second
set of Fourier coefficients $G_2(\rho_j)$
13.
Calculate a set of ratio values
$T_p(\rho_j) = G_1(\rho_j)/G_2(\rho_j)$
for $j = 1$ to n
14.
Apply log-by-rho-squared
transform to obtain calculated table
$T_c(\rho_j) = \log(T_p(\rho_j))/\rho_j^2$
15.
Calculate distance of object
based on calculated table T_c
and stored table T_s
output distance
16.
Calculate a set of focus
camera parameters
$E_f = (\lambda_f, s_f, f_f, D_f)$
based on T_c and T_s
17.
Set camera parameters to E_f
and accomplish autofocusing

Figure 5.7: Flow Chart for DFD1F

image. The default region is the center of the image. The size of the region is also an option and the default size is 128 × 128.

The two recorded 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. One normalization we have deliberately ignored is the magnification normalization. For our camera system, the change in the magnification due to change in lens position was found to be negligible (about 2%). But if it is not negligible, magnification normalization should be done.

Due to blurring and spreading of light from point objects, the grey levels at the border of an image region are affected by light from points immediately outside the image region. The light distributions produced by points inside and outside the border overlap. We call this the image overlap problem [?]. In order to reduce the errors due to the image overlap problem, the image is weighted (i.e. multiplied) by a two-dimensional Gaussian centered at the center of the image and having a spread parameter σ equal to about 1/3rd of the image size (i.e. about 40 for a 128 × 128 image). The weighting function is

$$w(i,j) = \exp\left(-\frac{(i-\bar{\imath})^2 + (j-\bar{\jmath})^2}{2\sigma^2}\right) \tag{5.19}$$

with $\bar{\text{i}} = \mathit{image\_height}/2$, $\bar{\text{j}} = \mathit{image\_width}/2$, and i and j are the row and column indices. This function has a weight of 1 at the center and gradually reduces to about 0.325 at the border.

The two images g_1 and g_2 are then summed rowwise to obtain two one-dimensional sequences, say $g_{10}[i]$ and $g_{40}[i]$.

The MTF of the lens system as a function of object distance u and spatial frequency ρ was provided to us by the lens manufacturer. Plots of MTF for lens step positions 10, 40, and 70 are shown in Figures 5.8, 5.9 and 5.10. The manufacturer obtained this data using a computer simulation of the lens system. The same data could be obtained through direct measurements on the lens system using special equipment. The manufacturer provided the data at intervals of 1 cycle/mm spatial frequencies starting from 1 cycle/mm to 15 cycles/mm. However, for our camera, each pixel corresponds to 0.6 cycles/mm. This is calculated as follows:

Vertical sampling interval = 0.013 mm. Therefore, sampling frequency = 1 sample / 0.013 mm = 76.92 samples/mm. Therefore the maximum spatial frequency which can be present in the image without aliasing distortion arising =76.92/2 =38.46 cycles/mm. Let the image size be 128×128. In the discrete Fourier transform, the highest frequency corresponds to the discrete index $128/2 = 64$. Therefore, 64 corresponds to 38.46 cycles/mm, and one discrete frequency index corresponds to $38.46/64 = 0.601$ cycles/mm.

Therefore, the MTF data provided by the manufacturer is coarser (1 cycle/mm) than what we would like (0.601 cycle/mm). In order to obtain the MTF data at intervals of 0.601 cycles/mm from the data given at

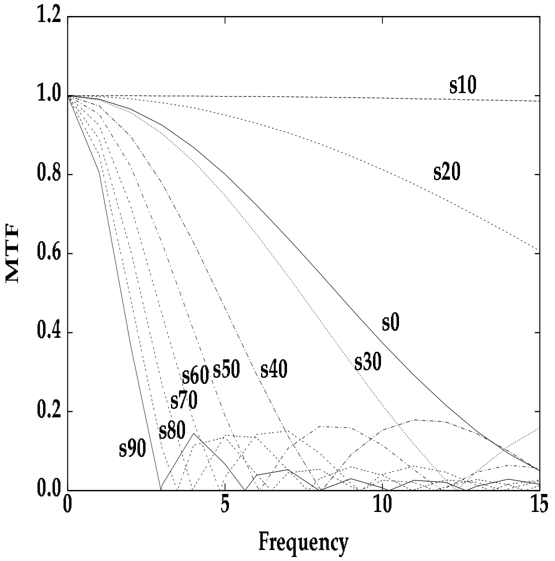


Figure 5.8: MTF for Lens Step #10

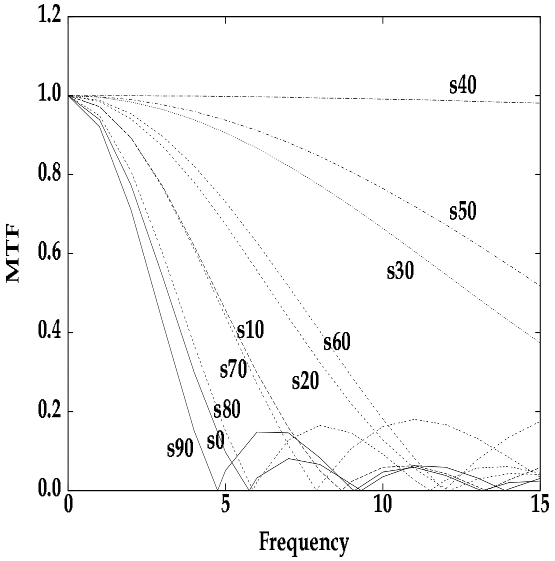


Figure 5.9: MTF for Lens Step #40

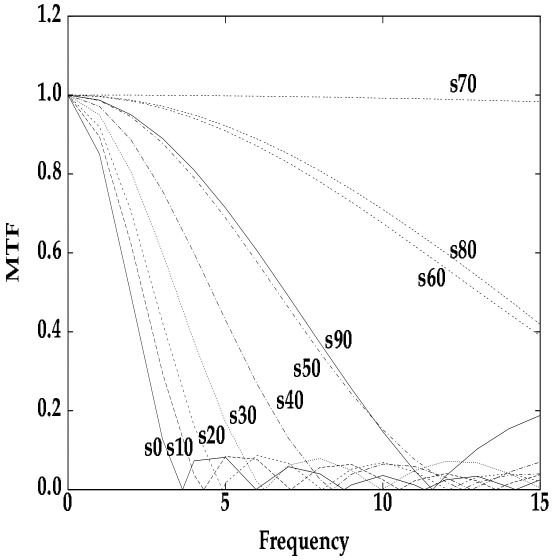


Figure 5.10: MTF for Lens Step #70

intervals of 1 cycle/mm, we used an interpolation scheme. This interpolation is more easily carried out after applying the $\log/\rho^2$ transformation discussed earlier. After the transformation, a linear interpolation gives more accurate results than a higher order interpolation directly on the MTF. Only for the first point at $\rho = 0.6$ cycles/mm, we need to do either extrapolation after the transformation, or an interpolation before the transformation.

For robustness against noise, we did not use data at points where the MTF value was low. This threshold was arbitrarily chosen to be around 30% of the maximum magnitude. This precludes usage of all data points after the first zero crossing of the main lobe of the MTF. Also, it precludes the usage of some data points to the left of the first zero crossing belonging to the main lobe. One effect of this restriction on limiting the data points used is that it restricts the maximum allowable blur in an image. The MTF is circularly symmetric and its cross section has the general shape of a sinc function. A plot of the MTF data of our camera is shown in Fig. 5.2.

Due to the restriction on the minimum MTF magnitude for using the corresponding data, we cannot for example use an image acquired at step 10 of an object which is very close because the object will be very highly blurred. Generally, for reliable estimates of distance, about 6 data points are needed.

After the interpolation, the Table corresponding to

$$T_s(\rho; u) = \frac{-2}{\rho^2} \ln \frac{H_{10}(\rho)}{H_{40}(\rho)} \qquad (5.20)$$

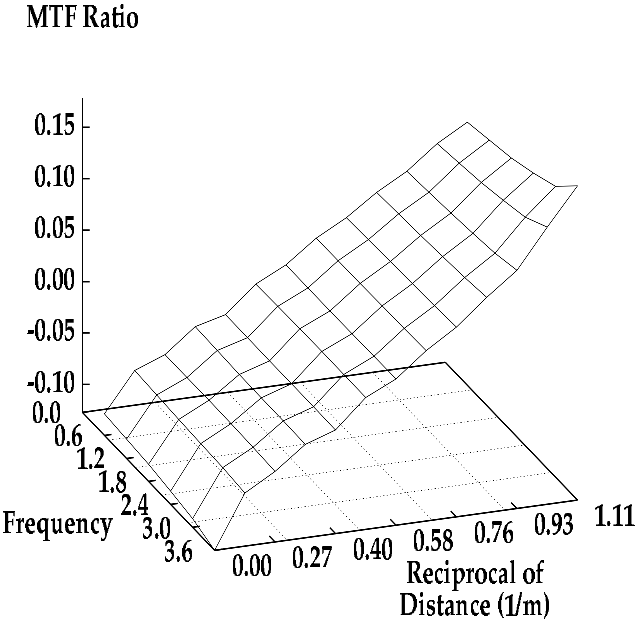


Figure 5.11: T_s for Lens Step #10 and #40

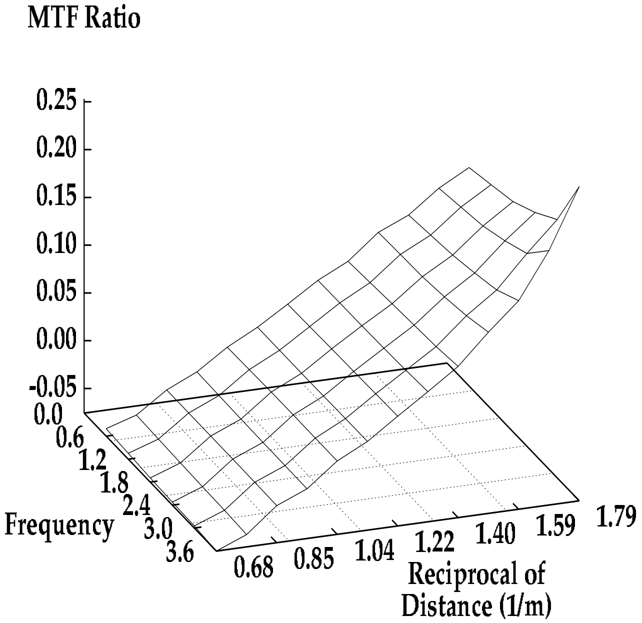


Figure 5.12: T_s for Lens Step #40 and #70

is obtained, where H_{10} and H_{40} represent the MTFs for lens positions 10 and 40 respectively. A plot for this table is shown in Fig. 5.11. The table size is fixed to be 6 because we use 6 points in the estimation of distance.

Next, the first 6 discrete Fourier coefficients of g_{10} and g_{40} are computed. Let these be $G_{10}(\rho)$ and $G_{40}(\rho)$ for $\rho = 1, 2, \cdots, 6$. The computed table T_c is obtained by

$$T_c(\rho) \;=\; \frac{-2}{\rho^2} \ln \frac{G_{10}(\rho)}{G_{40}(\rho)}. \tag{5.21}$$

Mean-square error is computed between T_c and T_s for different values of u. The value of u for which the mean-square error is a minimum is taken to be the estimated distance of the object. However, if the minimum error occurs for a distance corresponding to higher than step 60 then it is considered to be too far from step position 10 for reliable results. Therefore, in this case, a third image is taken at step position 70, and the images taken at steps 40 and 70 are used in estimating distance. A plot of $T_s(\rho, u)$ for step positions 40 and 70 is shown in Fig. 5.12. Again, MSE is computed for this table. The value of u for which MSE is a minimum is taken to be the distance of the object. The distance of the object is printed on the computer terminal, and the lens is moved to the corresponding position to focus the object.

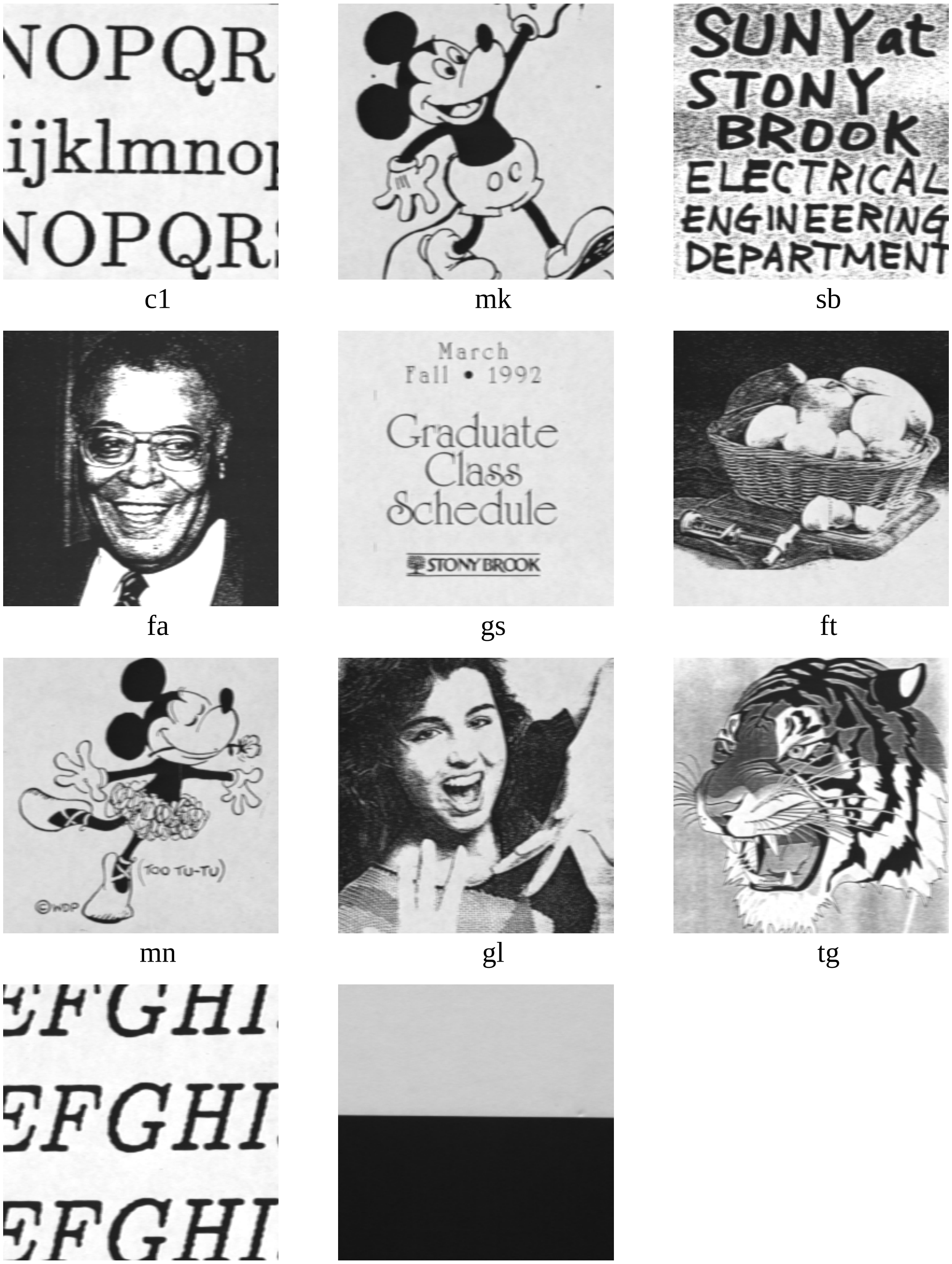
5.5 Experiments

Three types of experiments were conducted under the following conditions. Camera setting: focal length = 35 mm, F-number = 4, camera gain control +6dB, White balance = off, Gamma compensation = off.

5.5.1 Experiment 1

This experiment was designed to test the accuracy of DFD1F. The illumination was kept constant at 400 lux (about the ambient illumination in an office environment). A fixed image size of 128 × 128 was used. Four image frames were time-averaged to reduce noise. Eleven pictures shown in Fig. 5.13 were used as test objects: a step edge (ev), two human faces (fa, gl), four text based objects (c1, c2, gs, sb), two cartoon pictures (mk, mn), a fruit bucket (ft), and a tiger head (tg). Each object was placed at 19 different distances shown in Table 5.1 (except ∞ corresponding to step 0). The distances will be denoted by u_i where i is the step number of the lens position which would focus an object at distance u_i according to Table 5.1.

Three different programs corresponding to three methods of determining distance were run. Two of them are DFD1F.MSE and DFD1F.MN. The third is a depth-from-focus program [?] based on maximizing a focus measure. The focus measure is defined as the energy of low-pass filtered image gradient [?]. It performs a binary search by moving the



c1 mk sb

fa gs ft

mn gl tg

c2 ev

Figure 5.13: Test Objects for Experiment 1

lens to find a position where the focus measure is a global maximum. The lens position is then used as an index into a table to find object distance. This method uses as many images as needed (about 10 or more as compared to a maximum of 3 in DFD1F) to find the best focused position. The depth accuracy obtained by this method is better or close to the best accuracy obtained by other known DFF methods (depending on image content and noise) [?]. Therefore it serves as a good DFF method and a benchmark against which DFD methods can be compared. We will refer to this DFF method as DFF.BST.

The results of the experiments DFF.BST, DFD1F.MSE, and DFD1F.MN, are shown in Table 5.2. In Table 5.2, the first column shows object distance. Columns 2 to 4 correspond to the results of DFF.BST, DFD1F.MSE, and DFD1F.MN. The entries there show the mean focus position ± the standard deviation for the 11 test objects as determined by the three programs. The last row shows the overall root-mean-square (RMS) error for each program. We see that, as expected, the mean focus position in each of the rows are more or less a constant whereas the entries along a column increase monotonically. The mean focus values are different from Table 5.1 because of a assembly error between the lens and the CCD camera. The error corresponds to a constant shift of about 12 lens steps. Plots of the reciprocal of distance as a function of the these mean values is shown in Fig. 5.14. This Figure shows that the plots are almost linear, except for a small glitch at the beginning for DFD1F.MSE and DFD1F.MN. One can use these plots to find the distance of objects for

Distance (m)	DFF.BST	DFD1F.MSE	DFD1F.MN
5.300	16.09 ± 0.79	27.18 ± 6.62	26.73 ± 6.63
3.750	21.82 ± 0.83	21.82 ± 2.33	21.09 ± 2.44
2.850	26.09 ± 0.67	25.36 ± 2.71	25.00 ± 2.73
2.500	30.82 ± 1.70	30.82 ± 1.75	30.00 ± 1.93
1.930	35.55 ± 0.78	37.36 ± 2.27	36.82 ± 2.33
1.720	42.09 ± 0.79	41.55 ± 3.20	40.64 ± 3.33
1.465	47.73 ± 0.96	47.82 ± 2.82	46.55 ± 3.10
1.320	51.45 ± 1.50	50.27 ± 2.63	49.91 ± 2.66
1.170	56.18 ± 1.34	53.18 ± 3.13	52.45 ± 3.21
1.080	60.00 ± 2.80	59.36 ± 4.58	58.55 ± 4.65
0.965	68.18 ± 1.47	67.36 ± 5.09	65.82 ± 5.32
0.900	71.91 ± 0.90	72.64 ± 2.67	70.82 ± 3.23
0.822	77.09 ± 2.81	75.91 ± 4.56	74.55 ± 4.76
0.770	85.00 ± 1.71	80.09 ± 3.18	78.91 ± 3.39
0.715	89.27 ± 2.00	82.00 ± 3.95	80.64 ± 4.18
0.670	93.00 ± 0.74	85.27 ± 3.14	84.18 ± 3.32
0.628	94.36 ± 1.67	88.55 ± 3.58	86.91 ± 3.93
0.595	95.09 ± 1.56	90.09 ± 3.63	89.45 ± 3.68
0.560	95.09 ± 1.31	92.64 ± 3.47	92.00 ± 3.53
RMS Error	1.52	3.61	3.76

Table 5.2: Results of Experiment 1

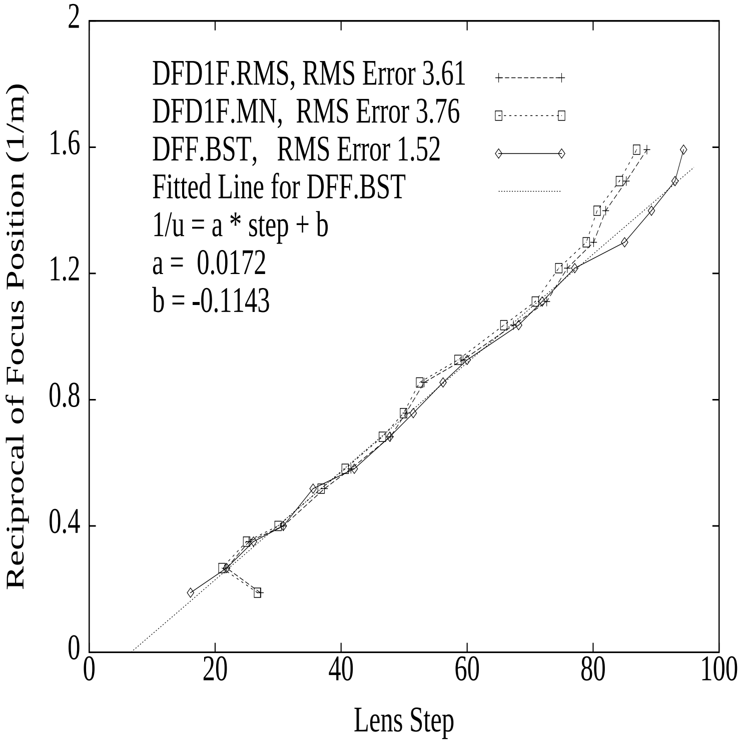


Figure 5.14: Results of Experiment 1

any of the three programs.

The RMS error for DFF.BST is 1.52 steps out of 97 steps. This corresponds to about 1.6% RMS error in the lens position for autofocusing. The RMS error for DFD1F.MSE is 3.61 steps out of 97 steps. This corresponds to about 3.7% RMS error in autofocusing. Blurring due to this small error is not easily noticeable by humans. Therefore, it gives satisfactory results in autofocusing applications. We see that this (3.7%) compares well with the performance of DFF.BST method (1.6%) which is close to the best achievable by any known DFD method. For DFD1F.MN, the RMS error in this case is 3.76 steps out of 97 steps. This is only a little worse than the DFD1F.MSE method. This corresponds to about 3.8% RMS error in autofocusing. Once again, it (3.8%) compares well with the DFF.BST method (1.6%).

5.5.2 Error Analysis

In the plot of Table 5.2 shown in Figure 5.14, we see that the reciprocal of object distance $1/u$ is linearly related to lens position. This relation can be specified by

$$1/u = ax + b \tag{5.22}$$

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 0.03 mm. The RMS errors mentioned above are for lens position. This gives a good indication of the performance of the method for application in rapid autofocusing of cameras. As for error in determining object distance, it can be estimated by taking error differentials in Eq. (??):

$$
\begin{aligned}
|\delta(1/u)| &= a\,|\delta x| && (5.23)\\
\rightarrow\ |\frac{\delta u}{u}| &= a\,|\delta x| u && (5.24)\\
\rightarrow\ |\delta u| &= a\,|\delta x| u^2 && (5.25)
\end{aligned}
$$

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 system, $a \approx 0.0172$. Setting $|\delta x|$ to be the RMS error of 3.6 steps, a plot of the relative error $|\frac{\delta u}{u}|$ and absolute error $|\delta u|$ are shown in Figures 5.15 and 5.16. In Figure 5.15 we see that the percentage error in distance at 0.6 meter is about 4% and increases linearly to about 30% at 5 meter distance. This compares well with the error obtained by the DFF.BST approach of about 1.6% at 0.6 meter increasing linearly to about 12.5% at 5 meter for DFF.BST. Figure 5.16 shows that absolute error increases quadratically from about 2.5 centimeter at 0.6 meter to about 1.5 meter at 5 meter distance. The corresponding numbers for DFF.BST are about 1 cm at 0.6 meter and about 0.6 meter at 5 meter.

The total number of experiments conducted on DFD1F.MSE is 209 (=11 × 19). In each experiment, a maximum of 3 images were used. All the images used in the experiments have been collected and stored in an image database named SPARCS.DB1. This database which contains a

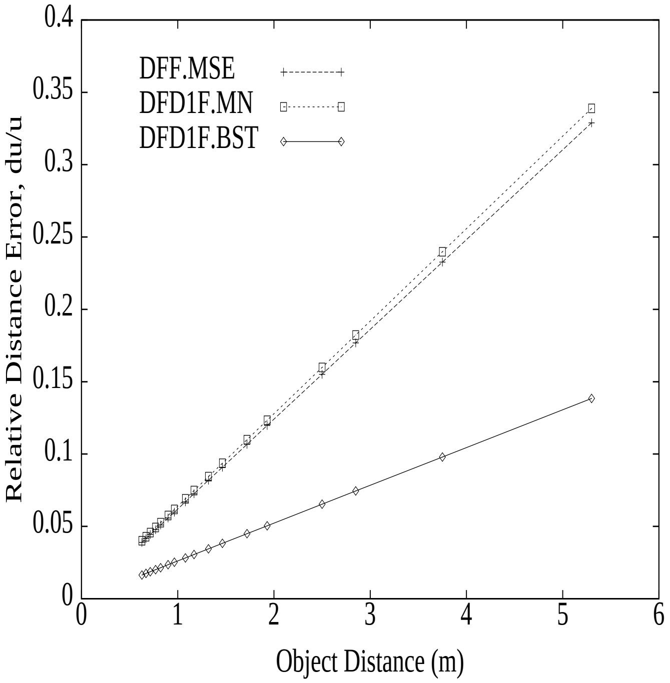


Figure 5.15: Error in Relative Distance

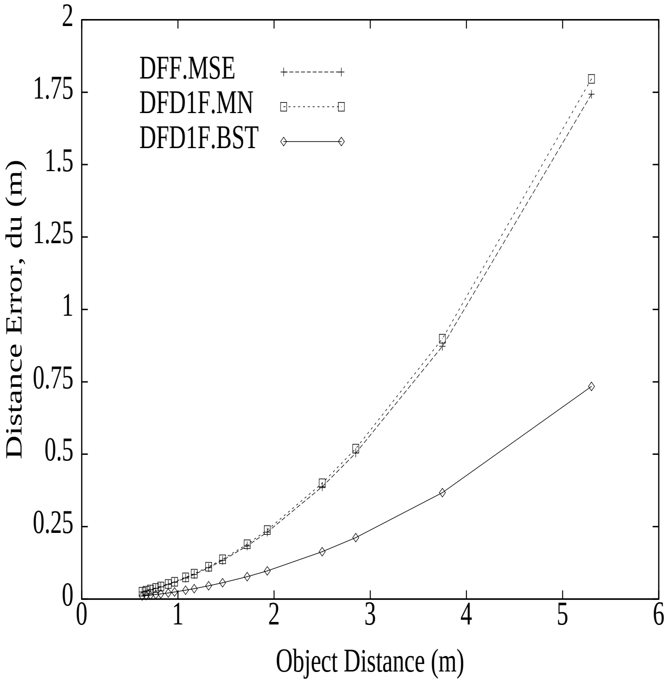


Figure 5.16: Error in Absolute Distance

total of 627 (=3 × 209) images will be made available to other interested researchers for investigating different DFD methods.

5.5.3 Experiment 2

In this experiment, the object distance was fixed to be 2 meter, but test object and illumination were changed. The camera setting was same as before, except that no image frame averaging was done (i.e. only one image frame was used). A set of seven different standard charts were provided as test objects to us by our lens manufacturer to test our method. These objects, titled A,B,...,G, are shown in Fig. 5.17. The objects are: b/w edge, gradually changing gray level pattern, light gray background with one thin white line at the center, white background with one black thin line at center, edge with a small change in gray level, and black background with one thin white line at the center. The image size was fixed to be 64×128. The results are shown in Table 5.3. We find that the method performs very well for high illumination and high contrast objects and the performance degrades gradually with decreasing illumination and contrast. Overall the results indicate that DFD1F is useful in practical situations.

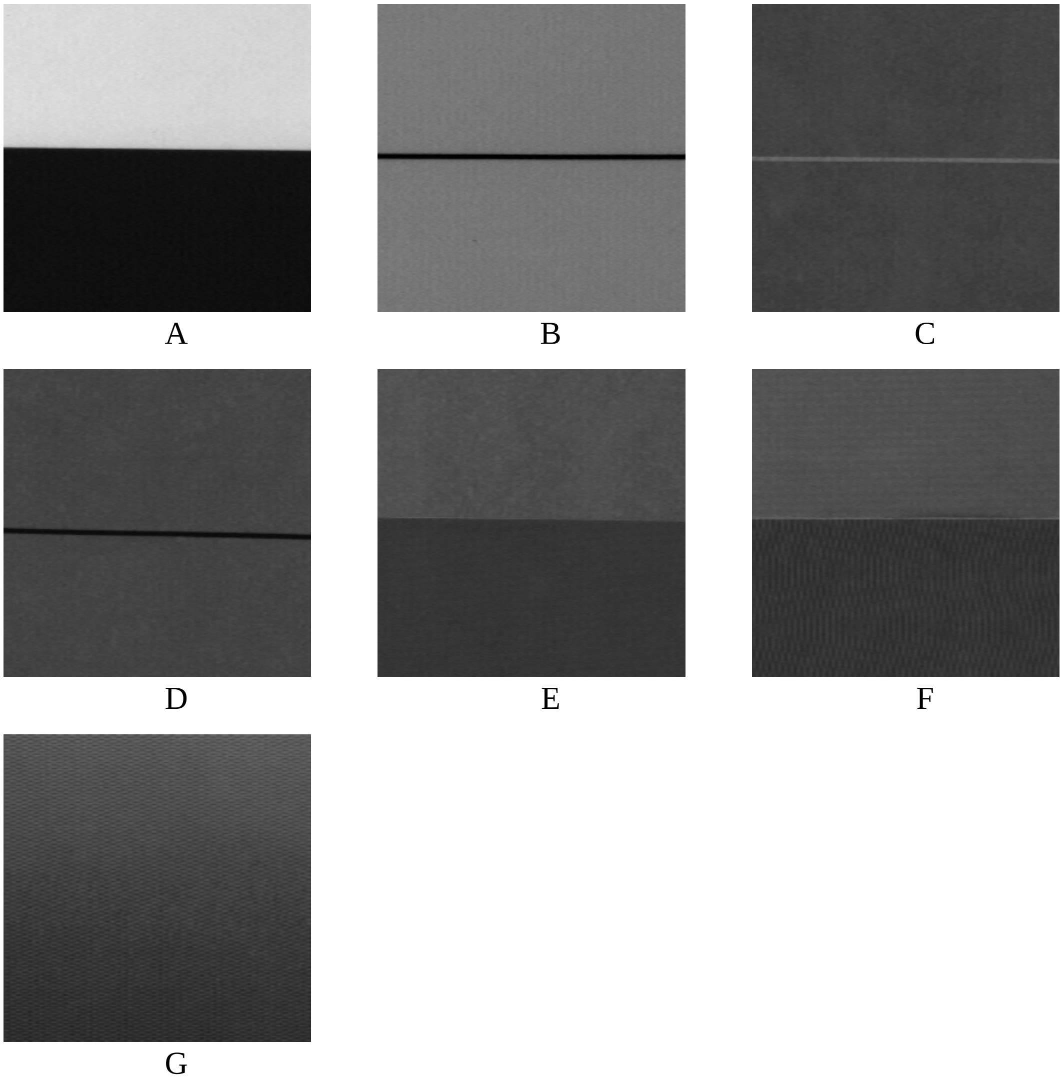


Figure 5.17: Test Objects for Experiment 2

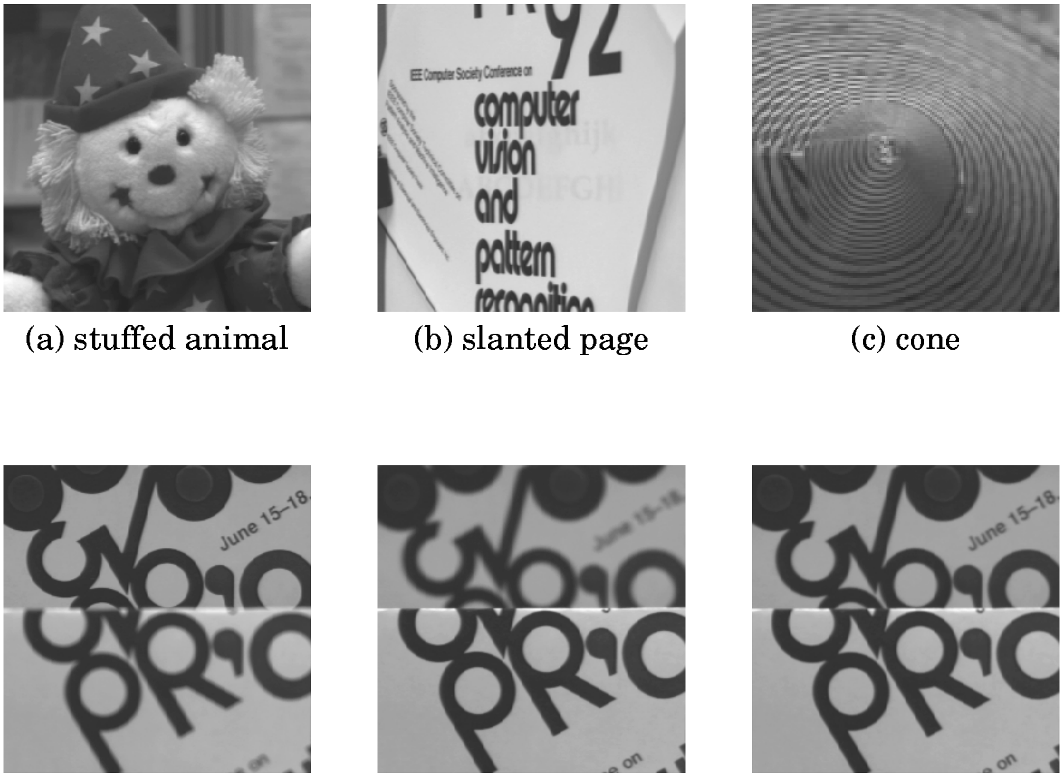
		A	B	C	D	E	F	G
800	Mean	32.2	31.0	26.2	29.9	32.5	31.1	24.9
(lux)	Std	0.40	0.45	1.17	1.22	0.67	0.94	1.14
200	Mean	31.7	29.6	26.9	28.3	30.8	28.3	24.7
(lux)	Std	2.10	3.26	3.11	2.93	2.23	4.24	2.37
50	Mean	28.9	32.4	26.5	22.7	31.5	37.6	31.6
(lux)	Std	4.87	15.07	23.46	22.42	16.10	17.51	24.06

Table 5.3: Experimental Results for Different Illuminations

5.5.4 Experiment 3

In experiments 1 and 2, the objects were planar posters placed normal to the optical axis. Those experiments were useful in doing a rigorous performance and error analysis of DFD1F. Here we report the results of finding the distance of 3D objects.

For a 3D object, the radius of the blur circle changes from one point to the other on its image. Therefore, the observed image cannot be modeled as the result of convolving the focused image with a single PSF. Therefore, theoretically, DFD methods cannot be used. Even the DFF approach has this weakness although to a lesser degree. However, in practice, both DFD and DFF methods can be used if the depth variation in the image window being processed is not large. Some kind of “average”



(a) stuffed animal

(b) slanted page

(c) cone

(d) occlusion edge
(far portion focused)

(e) occlusion edge
(near portion focused)

(f) occlusion edge
(focused by DFD1F)

Figure 5.18: Test Objects for Experiment 3

depth estimate in the image window is obtained.

Figure 5.18a-f shows the 3D objects used in our experiments. The experiments were performed under room illumination (350-400 lux). The first object (5.18a) is a stuffed animal which has a depth variation of about 2 to 3 cm. The second (5.18b) is a slanted page with printed characters which has a depth variation of about 7 to 20 cm. These two objects were placed such that their nearest points were located at 1, 2, and 3 meters. For each of the three distances, DFF.BST, DFD1F.MSE, and DFD1F.MN programs were run 10 times to obtain the mean and standard deviation of the focused position. The results are shown in Table 5.4. We see that all three methods give nearly the same results.

The third test object (5.18c) is a cone with its tip at 2 meters and its axis approximately along the optical axis. In the 128 × 128 image window processed by the three methods, the cone extended from 2 meters to 3.4 meters. Again the programs were run 10 times and mean and standard deviations were obtained (see Table 5.4). DFF.BST gave the mean focus position as lens step 24 corresponding to a distance of 3.2 meters from the camera. DFD1F.MSE and DFD1F.MN gave lens steps 25.1 and 25.6 respectively corresponding to a distance of about 2.9 meters. The results are closer to the distance of the bottom of the cone, possibly because the fraction of the image area occupied by the bottom half is much larger than the top half of the cone.

The last object (5.18d-f) was an occlusion edge created by two sheets of paper placed 30 cm apart along the optical axis. The near portion (bottom half) was placed at 1, 2, and 3 meters respectively. The results are shown in

Object	Distance	DFF.BST	DFF1F.MSE	DFF1F.MN
Stuffed Animal	1.00m–1.02m	65.0 ± 1.00	65.5 ± 0.50	64.1 ± 0.30
Stuffed Animal	2.00m–2.02m	28.0 ± 0.45	28.0 ± 0.00	27.2 ± 0.40
Stuffed Animal	3.00m–3.02m	17.8 ± 0.60	22.3 ± 0.78	22.1 ± 0.94
Slanted Page	1.00m–1.07m	55.6 ± 0.80	54.0 ± 0.00	52.2 ± 0.40
Slanted Page	2.00m–2.12m	27.0 ± 1.00	29.5 ± 0.67	28.6 ± 0.49
Slanted Page	3.00m–3.20m	18.0 ± 0.00	20.3 ± 0.90	20.2 ± 0.87
Occlusion Edge	1.00m, 1.30m	56.8 ± 0.98	53.0 ± 0.00	52.5 ± 0.50
Occlusion Edge	2.00m, 2.30m	26.4 ± 0.80	24.5 ± 0.50	24.9 ± 0.30
Occlusion Edge	3.00m, 3.30m	17.0 ± 1.00	19.9 ± 0.54	19.4 ± 0.49
Cone	2.00m–3.40m	24.0 ± 0.00	25.1 ± 0.54	25.6 ± 0.49

Table 5.4: Results of Experiment 3

Table 5.4. Figure 5.18 shows three images, one with the top portion focused (5.18d), one with the bottom portion focused (5.18e), and the last one is the focused image of the scene according to DFD1F.MSE (5.18f). We see that in the last image, in comparison with the first two images, both the near and the far portions are nearly focused. This is because DFD1F.MSE (as does DFF.BST) gives a distance which is somewhere in between the near and the far portions. These experiments show that DFD1F is useful in practical applications.

5.6 Conclusion

We have described the theory and implementation of a new DFD method named DFD1F for determining depth from image defocus information. The method has been demonstrated successfully on an actual camera system built by us. Experimental results here indicate that DFD1F is useful for passive ranging and rapid autofocusing. The ranging accuracy is high for nearby objects and it decreases with increasing distance. DFD1F can be combined with a DFF method such as the DFF.BST to reduce the percentage error by a factor of about 2 to 3 at the additional cost of acquiring and processing a few (about 3) more images. This combination represents a good trade-off between speed and accuracy.

In comparison with stereo method of ranging, DFD1F method does not suffer from the correspondence problem, but it is in general less accurate than stereo vision. Therefore DFD1F could be used to get a rough estimate of distance which can then be used by a stereo algorithm to determine more accurate distance. The computation associated with establishing correspondence is reduced due to the availability of a rough estimate of distance. We will show a simple implementation of combining DFD1F with stereo in chapter 8.

DFD1F can be used to obtain a rough and coarse depth-map of a scene very fast in parallel from only two images of the scene irrespective of whether any object is focused or not. The entire field of view of the camera can be divided into many smaller subfields of view, and an “average” depth estimate of the scene can be obtained in each subfield of view using DFD1F. The images in the subfields of view can be processed in parallel.

DFD1F can be used to estimate the focused image of an object from only two observed images of the object, both of which are blurred. First the distance of the object is determined. Then the PSF corresponding to one of the blurred image is computed from a knowledge of the camera parameters. The focused image is then estimated by deconvolving the blurred image with the computed PSF. We will discuss this topic in chapter 7.

Distance of "plain" objects such as white walls which do no exhibit reflectance variation under uniform illumination cannot be determined by DFD1F. However, a random illumination pattern can be projected onto such an object to make it "textured". DFD1F can then be used to determine its distance.

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 of DFD1F since maximizing depth-of-field reduces the difference in blur between objects at different distances. Therefore, DFD1F can be made much more accurate by designing cameras with small depth-of-field for the purpose of ranging.

Finally, it is interesting to investigate the relevance of DFD methods to human vision. DFD1F suggests that two images of a scene observed by human eyes with different focal lengths can be used to extract a rough depth map of the scene. There is evidence that the human eye deliberately exhibits small fluctuations in its focal length to obtain two images. The following paragraph is quoted from Weale [?] (page 18):

> "... the state of accommodation of the un stimulated eye is not stationary, but exhibits micro fluctuations with an amplitude of approximately 0.1 D (diopter: a unit of lens power given by the reciprocal of focal length expressed in meters) and a temporal frequency of 0.5 cycles/second. He (Cambell, [?]) demonstrated convincingly that these were not a manifestation of instrumental noise, since they occurred synchronously in both eyes. It follows that their origin is central."

DFD1F implies that such fluctuations could be used to perceive depth in the entire scene simultaneously.

Chapter 6

Continuous Focusing of Moving Objects

6.1 Introduction

The problem of continuous focusing of a moving object using DFD approach has not been investigated in the previous literature. In this chapter we present a method for continuous focusing of moving objects. It is based on DFD1F described in the previous chapter.

Finding the distance of a moving object at a given time instant using a DFD method requires the simultaneous recording of two images of the object at the given time instant with different degrees of blur. The change in blur is caused by varying camera parameters such as lens position, focal length, and aperture diameter of the recording camera. A new camera structure is proposed for such recording of the images.

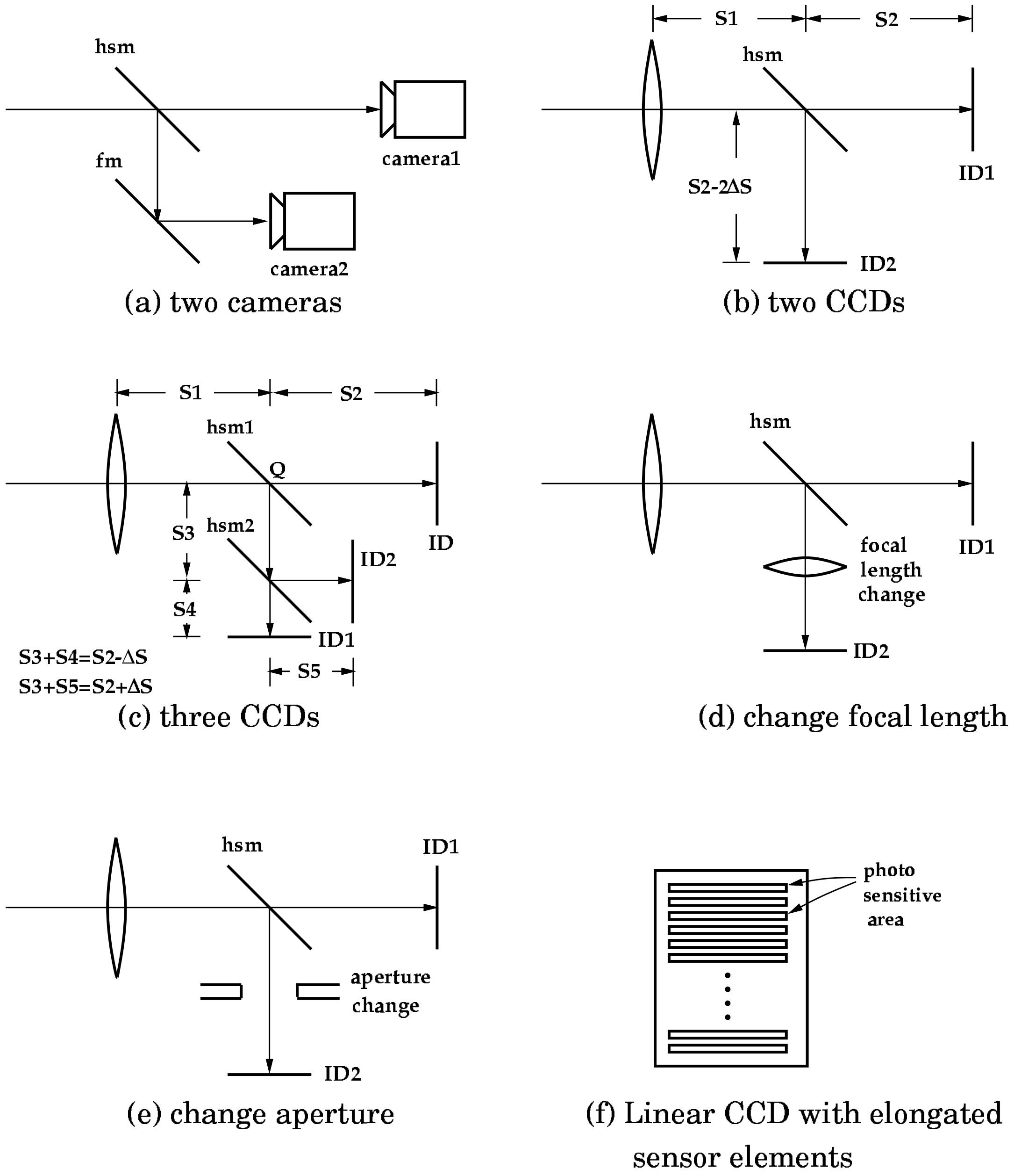
A straightforward adaptation of DFD1F for moving objects would require a large amount of memory space. The memory space is mainly needed to store a large look-up table representing the low-frequency MTF data of the optical system of the camera. A parameterization scheme is proposed for reducing the memory requirement.

The method proposed here has been implemented on an actual camera system. The results of experiments on this system are reported here. The results indicate that the method has a root-mean-square (RMS) error of about 4.3% in the lens position for focusing. The image blur caused by a focusing error of this magnitude is barely noticeable by humans. Therefore, in addition to machine vision, the method has practical applications in video cameras such as camcorders.

6.2 Camera Structures

In DFD1F, the camera parameters have to be changed after recording the first image g_1 but before recording the second image g_2. In most camera systems, this takes a few seconds of time since some mechanical parts (e.g. lens, aperture, etc.) have to be moved. In the case of moving objects this time delay is unacceptable because the object would have changed its position during the delay period. The images g_1 and g_2 must correspond to the same position of the object. Therefore the two images have to be recorded simultaneously in a short period of time. Fig. 6.1 shows camera structures for accomplishing this.

In Fig. 6.1(a), two identical cameras and a beam splitter with a



(a) two cameras

(b) two CCDs

(c) three CCDs

(d) change focal length

(e) change aperture

(f) Linear CCD with elongated sensor elements

Figure 6.1: Camera Structures for Focusing on Moving Object

hsm: half silvered mirror

fm: full mirror

mirror are used. The cameras are identical in all respects except that their camera settings are different.

In Fig. 6.1(b), a single lens with two image detectors (ID1 & ID2) and a beam splitter are used. There is a displacement of $2\triangle S$ between the two image detectors. This will serve as the change in lens position for the camera settings $\mathbf{e}_1$ and $\mathbf{e}_2$. Since the purpose of ID2 is to obtain one of the two images used by DFD1F, it can be a smaller CCD array as long as it is large enough to provide the image block used in focusing. In our experimental setting, a 128×128 CCD will be sufficient. A linear CCD with elongated sensor elements (fig. 6.1(f)) can also be used. The wider pixel size will automatically sum up the image along one direction to obtain the one-dimensional signal we need. The other image needed for DFD1F will be a subimage of the larger CCD. In this case, it will be a subimage from ID1.

Figure 6.1(c) shows a minor variation of Fig. 6.1(b) where two beam splitters and an additional image detector are used. The image detectors marked ID1 and ID2 can be smaller, for their purpose is obtain the two images used by DFD1F. In this case, image summation along one direction can be done by using a regular linear CCD (with square sensor elements) by rotating the half silvered mirror hsm1 (about the point Q) by a small angle.

In Fig. 6.1(d), a beam splitter and a lens are used for varying the focal length. The two images used in focusing will be the image on ID2 and a subimage from ID1. Therefore, ID2 can be made smaller.

In Fig. 6.1(e), a beam splitter and an aperture are used for changing the aperture. This setup is similar to Fig. 6.1(d), except that the variation is in aperture diameter instead of focal length.

It is possible to combine the features of Fig. 6.1(b, d & e) in a single camera where $\mathbf{e_1}$ and $\mathbf{e_2}$ are different in more than one camera parameter.

6.3 Parameterization of MTF

In this chapter we consider only the case of changing lens position s for continuous focusing. An alternative to this (which is employed by the human vision system) is to change the focal length f. In order to continuously change s for continuous focusing, it is necessary to store the MTF data for every possible value of s for each possible object distance. This would require a large memory space. For example, in our camera system, this would require storing about 6x100x100 floating point numbers. Here we propose a scheme for parameterizing the MTF data so that the memory requirement is drastically reduced (to about 100 floating point numbers for our camera).

The problem of storing MTF data arises because, the two PSF models based on paraxial geometric optics (Eq. ??) and and a Gaussian (Eq. ??) are not sufficiently accurate for actual camera systems. The two PSF models are good approximations but not adequate. This is particularly true for small F-number (less than 8) cameras.

The motivation for our parameterization scheme is based on the observation that the MTF of our camera is roughly if not exactly a Gaussian for low frequencies. This is clear from Figure 6.2. It shows a plot of a typical MTF (marked Lens Data) for the lens used in our experiments. Figure 6.3 shows a plot of σ obtained by applying the $\log/\rho^2$ transform to the lens MTF and then taking square root:

$$\sigma(\rho, \mathbf{e}, u) \;=\; \sqrt{\frac{-2}{\rho^2} \ln H_c(\rho; \mathbf{e}, u)} \qquad (6.1)$$

where H_c represents the lens MTF. In this plot we see that σ is almost a constant with respect to ρ for low frequencies. The average value of σ in the range $\rho = 1$ to $\rho = 3.6$ is taken as a measure of blur in one version of DFD1F. It can be used as a spread parameter of the PSF. The plot of a Gaussian MTF with this average σ is shown in Figure 6.2 for comparison. We see that the Gaussian is close to the lens MTF for low frequencies (up to $\rho = 3.6$). As expected, applying the $\log/\rho^2$ transform to the Gaussian MTF and taking the square root gives a function which is constant with respect to ρ for all frequencies (see Fig. 6.3). Figures 6.2 and 6.3 also show the plots of the MTF as predicted by paraxial geometric optics model and the square root of the $\log/\rho^2$ transform of the MTF. In this case the radius of the blur circle is taken to be $\sqrt{2}$ times the σ of the Gaussian.

From the above discussion we conclude that in practical applications a blur parameter σ can be defined as in Eq. (??) which is almost a constant with respect to low frequencies. This can be used to characterize the MTF data of a lens system.

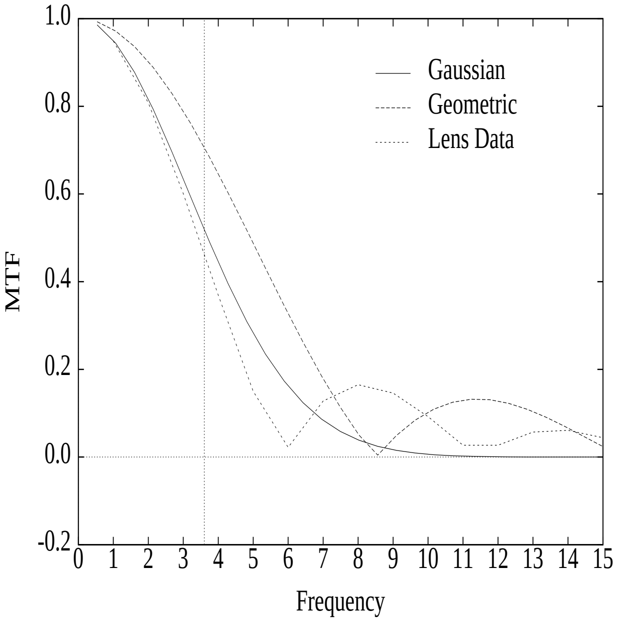


Figure 6.2: Gaussian, Geometric and Lens MTF

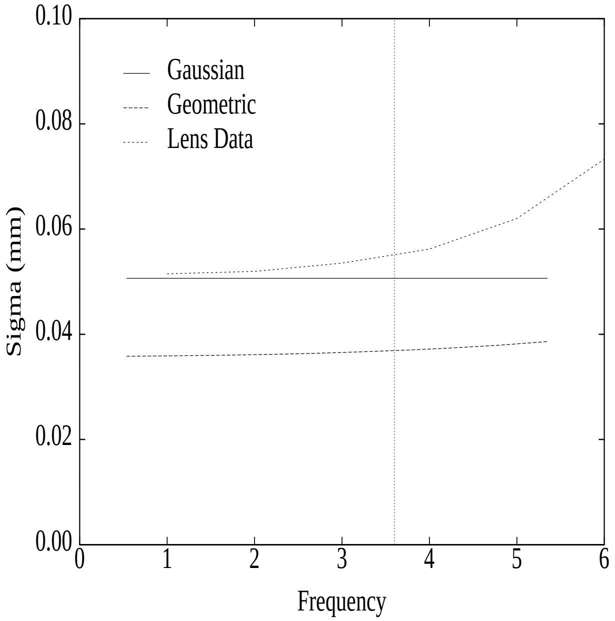


Figure 6.3: Estimated σ from MTF

If the lens MTF is exactly a Gaussian, then from Eq. (??) and $\sigma = R/\sqrt{2}$ we have:

$$\sigma = \frac{Ds}{2\sqrt{2}}\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right) \tag{6.2}$$

The minimum value of s is equal to f because objects at infinity come to focus at $v = f$ (see lens formula) and all other objects come to focus at $v > f$. Normalizing the magnitude of σ corresponding to $s = s_0$ [?] and using the lens formula, we obtain

$$\sigma = \frac{Ds_0}{2\sqrt{2}}\left(\frac{1}{v} - \frac{1}{s}\right) \tag{6.3}$$

Assuming $s_0 = f$ and using the approximation $vs \approx f^2$ which holds for most cameras, we obtain

$$\sigma = \frac{s - v}{2\sqrt{2}F\#} \tag{6.4}$$

where $F\#$ denotes the F-number (equal to f/D) of the camera. From the above expression we see that σ is directly proportional to the difference $s - v$ and inversely proportional to the F-number of the camera.

In the camera used in our experiments, the parameter s is changed by moving the lens with respect to the image detector. The lens motion is effected by a stepper motor with 97 steps numbered 0 to 96. When the lens is at step 0, except for assembly error, $s = f$ and therefore an object at infinity will be focused on the image detector. When the lens is moved to the other end (lens step 96), an object at a distance of about 0.55 meter will be focused. Each lens step corresponds to a relative displacement of the lens with respect to the image detector of about 0.025mm. For each lens position there corresponds a unique object distance for which the object will be in best focus on the image detector. For example, except for a constant due to assembly error between the lens and the camera, Table 1 shows the relation between the lens position and the distance of an object in best focus. It is therefore convenient to specify object distances in terms of lens position. For example, according to Table 1, if the distance of an object is said to be step 30, it means that the object is at distance 1.320 meters from the camera.

The relation between the lens position s (in mm) and the corresponding step number i is given by $s = f + i\ \delta s$ where δs is the lens displacement in mm per step of the motor. Similarly, if j specifies object distance in lens step, then we have $v = f + j\ \delta s$. Using these relations and Eq. ?? we can write

$$\sigma = \frac{\delta s}{2\sqrt{2}F\#}(i-j) \tag{6.5}$$

The above relation implies that the blur parameter σ which characterizes the lens MTF varies linearly with respect to both lens position i and object distance j with the same proportionality constant or slope. It is found that, after a minor modification, this model holds well for the actual lens MTF. The modification that is needed is the addition of a constant σ_{min}. This can be justified as follows. When an object is in best focus we have $i = j$ in the above expression. Therefore σ is zero according to this model. However, in practice, even when a point light source is in best focus, its' image is not a point which is dimensionless but the wellknown Airy pattern (bright and dark rings due to diffraction). Further, when the aperture is large (F-number of less than 8), the paraxial assumption (i.e. the light rays incident on the lens are almost parallel to the optical axis) does not hold. Therefore, instead of a single point where all rays from a point source are focused, there is only what is called a circle of least confusion which has finite dimensions. For these reasons, we propose the following model for the blur parameter σ

$$\sigma = \sigma_{min} + K|i - j| \tag{6.6}$$

(Another alternative model is $\sigma^2 = \sigma_{min}^2 + K^2(i-j)^2$.) The parameters σ_{min} and K are both approximately inversely proportional to the F-number of the camera system.

We next show some plots for the MTF data of our camera which indicate that the above model for σ is a good approximation to our camera.

Figure 6.4 shows a plot of the MTF data for the lens system used in our experiments. In this plot, one axis corresponds to spatial frequency ρ and the other axis corresponds to distance of the object being imaged. (The distance of the object is specified in terms of the lens position or lens step number.)

Figure 6.5 shows a plot of the MTF data in Figure 6.4 after the log/ρ^2 transform and taking square root. Here we see that, as before, the value of σ is almost a constant with respect to low frequencies for all object distances (lens steps 0 to 90). Therefore, σ has been averaged with respect to low frequencies and the resulting plot is shown in Figure 6.6. Here we see that σ is almost linear with respect to object distance specified in lens step. σ has a minimum value σ_{min} at step 40 corresponding to the

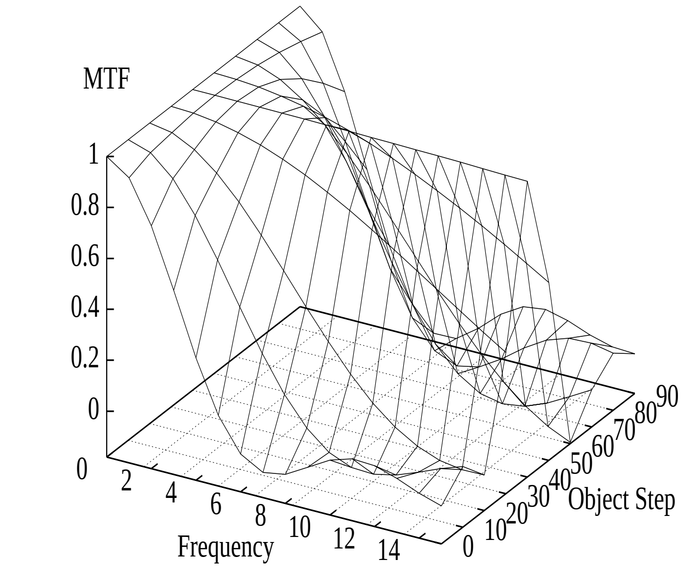


Figure 6.4: Lens MTF for Lens Step 40

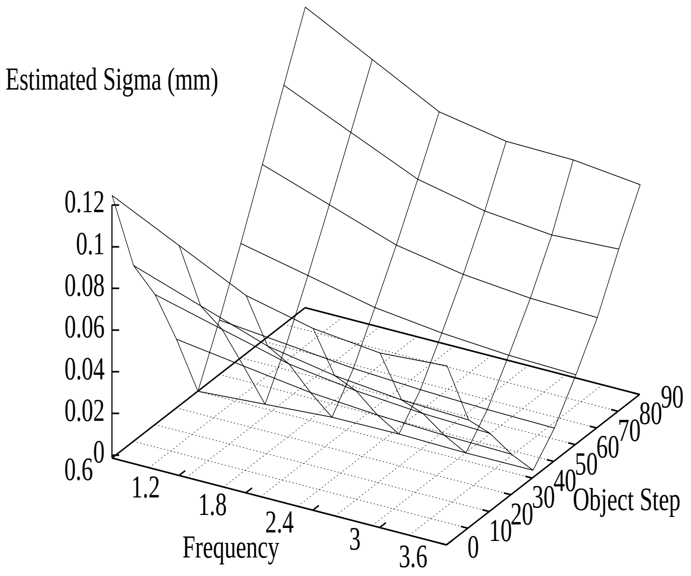


Figure 6.5: Estimated σ from Figure 6.4

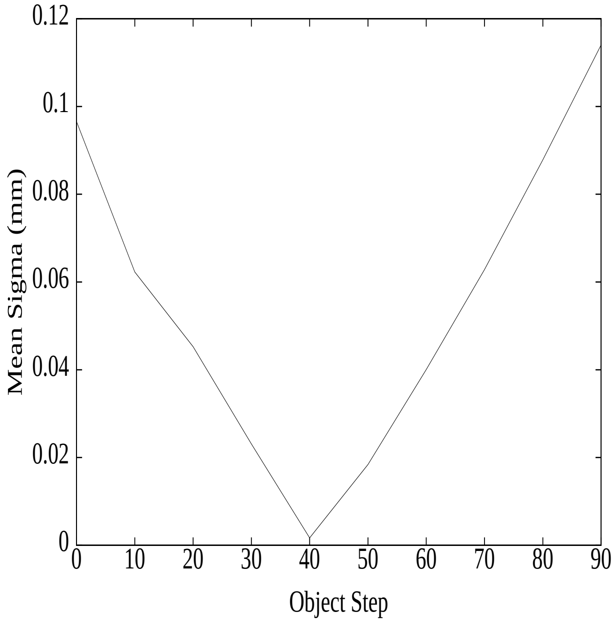


Figure 6.6: Mean σ from Figure 6.5

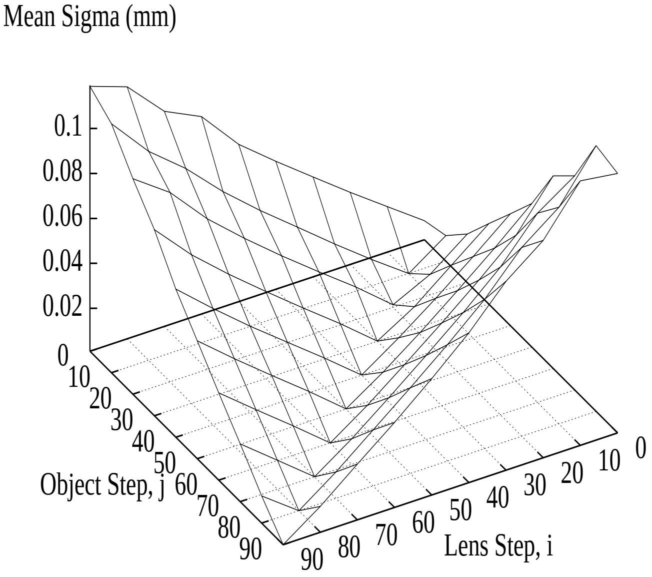


Figure 6.7: Estimated σ from Lens MTF

distance of the best focused object. On either side of the minimum, the slope is almost the same.

Figure 6.6 corresponds to an object distance of step 40. Similar plots have been obtained for object distances 0,10,20,...,90, and are shown as a 3D plot in Figure 6.7. We see that σ is a minimum along the diagonal and varies linearly on either side of the diagonal. The minimum value σ_{min} along the diagonal is almost a constant. The axes in this plot are lens position and object distance specified in step numbers. The slope on either side of the diagonal are almost the same. This implies that σ depends only on the difference between lens step i and object step j. These plots indicate that our proposed model (Eq. ??) can be used for practical camera systems. For the plot data in Figure 6.7, $\sigma_{min} = 8.904 \times 10^{-4}$ and $K = 1.343 \times 10^{-3}$mm.

6.4 Computational Steps

A flow chart of the algorithm for continuous focusing is shown in Fig. 6.8. Initially the lens is moved to step 15 which corresponds to focusing an object at about 3 meters distance. The variable Lens_Step in the flow chart corresponds to the position of the lens at any given instant. The stored table $\overline{T}_s[i]$ is computed next for two camera settings $\mathbf{e}_1$ and $\mathbf{e}_2$. In our experiments, the only camera parameter that was different for the two camera settings was the lens position. The first one was

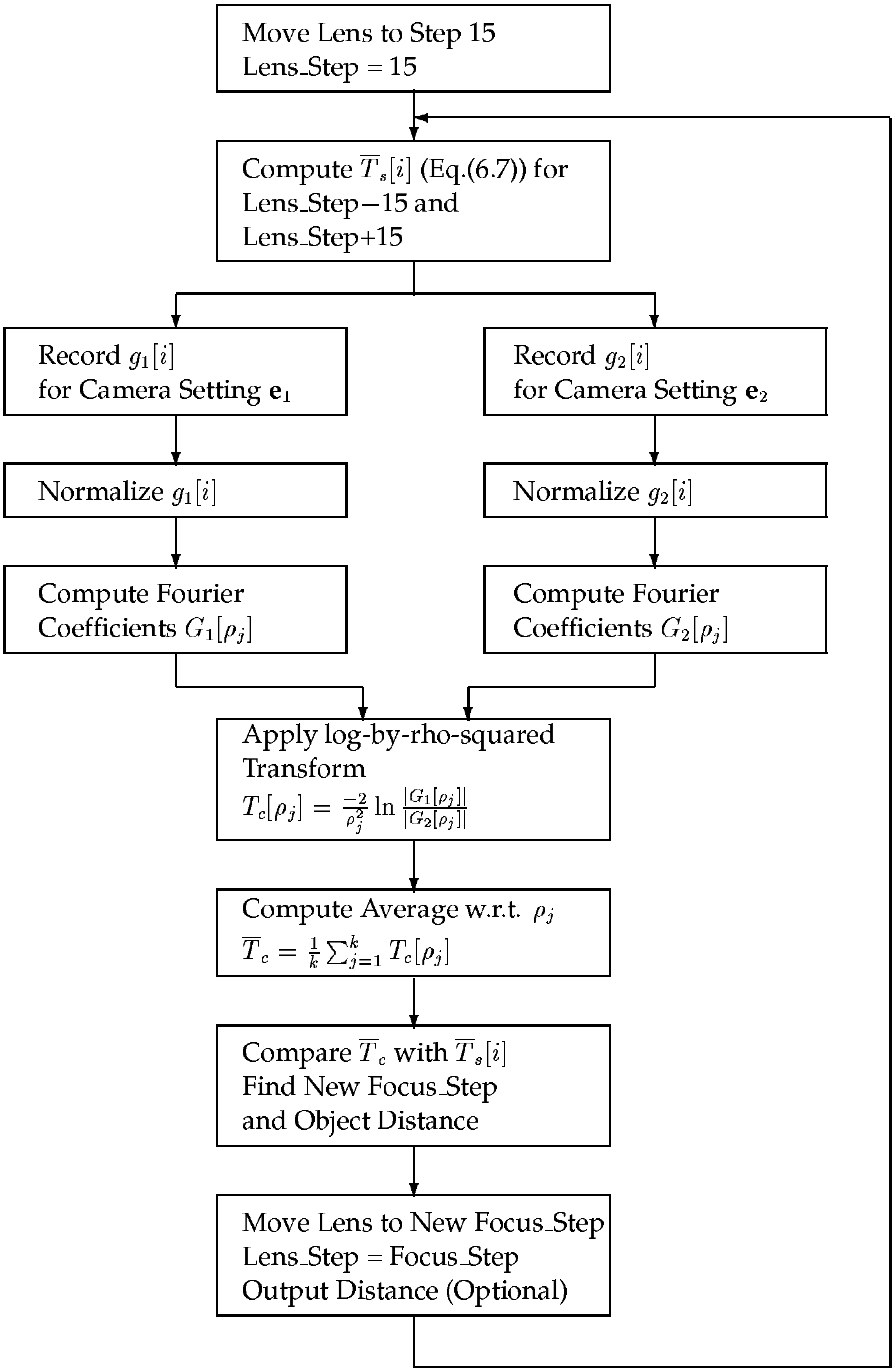

Move Lens to Step 15
Lens_Step = 15
Compute $\overline{T}_s[i]$ (Eq.(6.7)) for
Lens_Step$-$15 and
Lens_Step+15
Record $g_1[i]$
for Camera Setting $\mathbf{e}_1$
Record $g_2[i]$
for Camera Setting $\mathbf{e}_2$
Normalize $g_1[i]$
Normalize $g_2[i]$
Compute Fourier
Coefficients $G_1[\rho_j]$
Compute Fourier
Coefficients $G_2[\rho_j]$
Apply log-by-rho-squared
Transform
$T_c[\rho_j] = \frac{-2}{\rho_j^2} \ln \frac{|G_1[\rho_j]|}{|G_2[\rho_j]|}$
Compute Average w.r.t. ρ_j
$\overline{T}_c = \frac{1}{k} \sum_{j=1}^{k} T_c[\rho_j]$
Compare $\overline{T}_c$ with $\overline{T}_s[i]$
Find New Focus_Step
and Object Distance
Move Lens to New Focus_Step
Lens_Step = Focus_Step
Output Distance (Optional)

Figure 6.8: Flow Chart for Focusing on Moving Object

$Lens\_Step - 15$ and the second one was $Lens\_Step + 15$. In principle, other parameters such as focal length and aperture diameter, could also be varied. The parameters could be varied either one at a time or, two or more simultaneously. In our experiments, the stored table was computed as

$$\overline{T}_s[i] \;=\; [\sigma_{min} + K\;|i - (Lens\_Step - 15)|]^2 - [\sigma_{min} + K\;|i - (Lens\_Step + 15)|]^2 \tag{6.7}$$

Two images g_1 and g_2 are recorded corresponding to the camera settings $\mathbf{e}_1$ and $\mathbf{e}_2$. The image size in our experiments was 128×128. Both were summed along rows to obtain one-dimensional signals.

The two images g_1 and g_2 are normalized as in DFD1F with respect to mean brightness. In our implementation, normalization with respect to magnification was not done as the change in magnification was small (about 2-3%).

A few low frequency Fourier coefficients of g_1 and g_2 are computed. In the experiments, the first 6 coefficients were computed. The table $T_c[j]$ is then computed using the $\log/\rho^2$ transform as

$$T_c[j] = \frac{-2}{\rho_j^2} \ln \left| \frac{G_1(\rho_j)}{G_2(\rho_j)} \right| \tag{6.8}$$

Next the mean value of $\overline{T}_c$ is computed as

$$\overline{T}_c = \frac{1}{k} \sum_{j=1}^{k} T_c[j] \tag{6.9}$$

The mean $\overline{T}_c$ is compared with the stored table values $\overline{T}_s[i]$ and the index i for which the two values are closest is found. This index gives the lens step position for focusing the object. The index is also used to find the actual distance of the object through another table lookup. The lens is moved to the focusing step position and the variable Lens_Step is set to this new position.

Next the above algorithm is repeated beginning from the computation of $\overline{T}_s[i]$. The algorithm terminates when the camera power is turned off.

6.5 Experiments

Three poster pictures–FACE, NAVY, and OPTCON– shown in Figures 6.9, 6.10, and 6.11 respectively, were used as test objects. The reason for using planar objects is that it simplifies error analysis in the image window being processed. The estimated distance of the object can be compared with the actual distance to compute RMS errors. For 3D objects with depth variation in the image window of interest, the estimated distance will be some kind of "average" of different points in the window as discussed in chapter 5.

Experiments were done under the following camera settings: focal length: 35 mm, F#: 4, White balance: off, Gamma compensation: off, camera gain control: +6dB, illumination: 300 Lux. Each picture was placed at 24 different positions in sequence at time instants 1 to 24. The initial position was about 1 meter from the camera. The object was then

Figure 6.9: Test Image, FACE

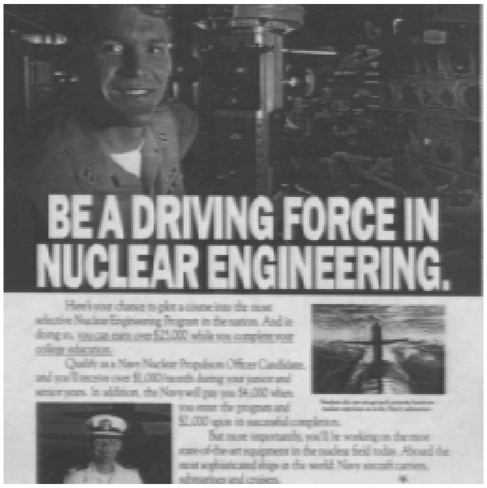


Figure 6.10: Test Image, NAVY

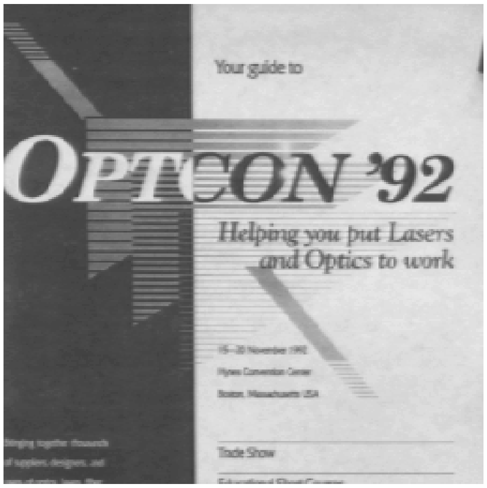


Figure 6.11: Test Image, OPTCON

moved gradually closer to the camera to a distance of about 0.6 meter. Next the object was moved gradually away up to a distance of about 5 meters. Then the object was moved back in steps to about 1 meter from the camera.

In Figure 6.12, the plot labeled 'Actual' shows the actual distance (in lens step number) of the "moving" object at different time instants. The estimated distance at each time instant for the three objects are plotted in Figure 6.12. We see that at the beginning there is a kind of "warm-up" period when the errors are relatively large. This is because, at the beginning the lens position (at step 15) was very far from the focused lens position (around step 55). Therefore the recorded images were highly blurred resulting in more error. After a few time instants, the camera "locks" onto the "moving" object and continuously focuses onto the object. During this "locked" period, focusing error is small because the lens position is not too far from the focused position and therefore the recorded images are less blurred.

In the beginning, no matter where the object is, the initial lens position will be at step 15. At each time instant, the camera records two images, one at 15 steps behind and another at 15 steps ahead of the current lens position. Using these images, it estimates the distance of the object and moves the lens to focus it. After moving the lens, it again records two more images and repeats the process. There are $24 \times 3 = 72$ data points in Fig. 6.12. The RMS error based on these 72 focusing results is about 4.2 lens steps out of 97 steps, or about 4.3%. The image

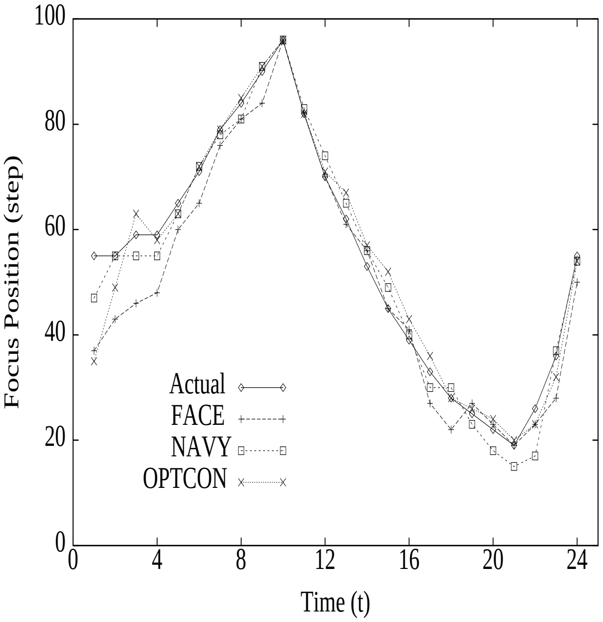


Figure 6.12: Experimental Results

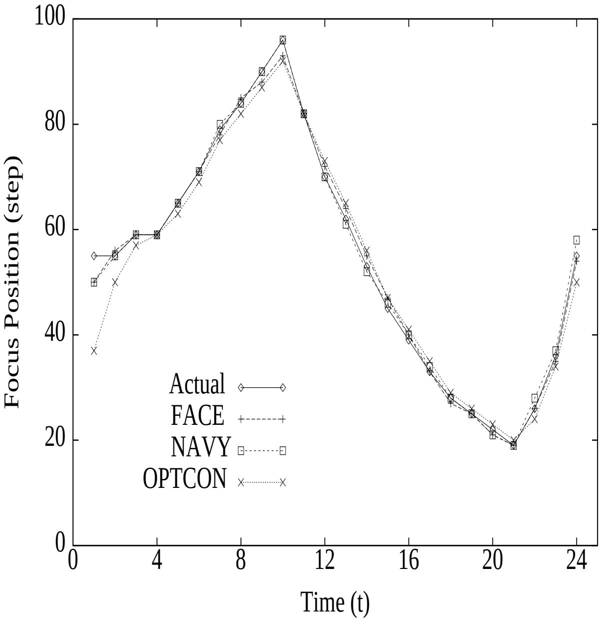


Figure 6.13: Simulation Results

blur due to a lens position error of this magnitude is small and is not easily noticeable by humans. Therefore, in addition to machine vision, the method is useful in camcorders.

Figure 6.13 shows the results of experiments on simulated image data. Paraxial geometric optics model of image formation was used to compute the blurred images corresponding to the three images in Figures 6.9, 6.10, and 6.11. We see that the focusing results are very good as expected.

6.6 Conclusion

In this chapter, we have extended the DFD1F method for continuous focusing of moving objects. Practical camera structures for the implementation are also presented. Experimental results show that the method can be applied to consumer video cameras such as camcorders. The method can also be applied to robotic vision for continuous tracking of moving objects.

Chapter 7

Focused Image Recovery

7.1 Introduction

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 (3D) scenes. However, the image of a 3D 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.

A spatial domain approach, named STM, comparable to the DFD1F was proposed by Subbarao and Surya [?, ?, ?]. STM also uses image defocus information to estimate distances. In STM and DFD1F, 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 present 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 a new spatial domain convolution/deconvolution transform (S transform) proposed in [?]. 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 present 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 2D Gaussian. The results of the second method are compared with simulation results. A subjective evaluation of the results leads to the following conclusions. The first method performs better and is much faster than the methods based on simple PSF models. The focused image recovery is good for up to medium levels of image blur (upto an effective blur circle radius of about 5 pixels). The performance of the second method is comparable to the simulation results. The simulation results represent the best attainable when all noise, except quantization noise, is absent. The second method gives good results upto relatively high levels of blur (upto an effective blur circle radius of about 10 pixels). Overall the second method gives better results than the first, but it requires estimation of the camera's PSF through calibration and is computationally several times (about 4 in practice) more expensive.

7.2 Estimation of Blur Parameter σ

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

$$\sigma^2 = \int_{\infty}^{\infty} \int_{\infty}^{\infty} (x^2 + y^2)\, h(x, y)\, dx\, dy \tag{7.1}$$

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 $\sigma = R/\sqrt{2}$, Eq. (??). For a PSF model based on a 2D Gaussian function, σ is the standard deviation of the distribution of the 2D Gaussian function.

In both STM and DFD1F methods, the blur parameter σ is first estimated and then the object distance is estimated based on σ. In addition to object distance, the blur parameter depends on other camera parameters as defined in Eq. (??). The parameters include– the distance between the lens and the image detector denoted by s, the focal length f of the lens, and the diameter D of the camera aperture. Both STM and DFD1F require at least two images, say $g_1(x, y)$ and $g_2(x, y)$, recorded with different camera parameter settings, say $\mathbf{e_1} = (s_1, f_1, D_1)$ and $\mathbf{e_2} = (s_2, f_2, D_2)$ respectively, such that at least one, but possibly two or all three, of the camera parameters are different, i.e. $s_1 \neq s_2$ or $f_1 \neq f_2$, or $D_1 \neq D_2$. DFD1F and STM also require a knowledge of the values of the camera parameters $\mathbf{e_1}$ and $\mathbf{e_2}$ (or a related camera constant which can be determined through calibration). Using the two blurred images g_1, g_2, the camera settings (or related camera constants) $\mathbf{e_1}$ and $\mathbf{e_2}$, and some camera calibration data related to the camera PSF, both STM and DFD1F methods estimate the blur parameter σ. A Fourier domain method is used in DFD1F whereas a spatial domain method is used in STM. The methods are general in that no specific model is used for the camera PSF, such as a 2D Gaussian or a cylindrical function.

Both STM and DFD1F have been successfully implemented on a prototype camera system named SPARCS. Refer to chapter 5 for the description of SPARCS. Experimental results on estimating σ have yielded a root-mean-square (RMS) error of about 2.7% for STM and about 3.7% for DFD1F. One estimate of σ can be obtained in each image region of size as small as 48×48 pixels. By estimating σ in small overlapping image regions, the scene depth-map can be obtained.

In the following sections we describe two methods for using the blur parameter σ thus estimated (using STM or DFD1F) to recover the focused image of the scene.

7.3 Spatial Domain Approach

In this section we describe the spatial domain method for recovering the focused image of a 3D 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 a new Spatial-Domain Convolution/Deconvolution Transform (S Transform) [?]. The transform itself is general and applicable to n-dimensional continuous and discrete signals for the case of arbitrary order polynomials. However, a special case of the general transform will be used in this section. First we summarize the S-Transform Convolution and Deconvolution formulas that are applicable here and then discuss their application for recovering the focused image.

7.3.1 S-Transform

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

$$f(x,y) = \sum_{m=0}^{3} \sum_{n=0}^{3-m} a_{m,n} x^m y^n \tag{7.2}$$

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

$$h_{m,n} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^m y^n h(x,y)\ dxdy \tag{7.3}$$

Let $g(x, y)$ be the blurred image obtained by convolving the focused image $f(x, y)$ with the PSF $h(x, y)$. Then we have

$$g(x,y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x-\zeta, y-\eta) h(\zeta,\eta)\ d\zeta d\eta \tag{7.4}$$

By substituting the Taylor series expansion of f in the above relation and simplifying, the following relation can be obtained:

$$g(x,y) = \sum_{0 \leq m+n \leq 3} \frac{(-1)^{m+n}}{m!n!} f^{m,n}(x,y) h_{m,n} \tag{7.5}$$

Equation (??) 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. If the PSF

$h(x, y)$ is circularly symmetric (which is largely true for most camera systems) then it can be shown that

$$h_{0,1} = h_{1,0} = h_{1,1} = h_{0,3} = h_{3,0} = h_{2,1} = h_{1,2} = 0 \text{ and } h_{2,0} = h_{0,2} \tag{7.6}$$

Also, by definition in Eq. (??), for the PSF of a camera,

$$h_{0,0} = 1 \tag{7.7}$$

Using these results, equation (??) can be expressed as

$$g(x, y) = f(x, y) + \frac{h_{2,0}}{2} \bigtriangledown^2 f(x, y) \tag{7.8}$$

where $\bigtriangledown^2$ is the Laplacian operator. Taking the Laplacian on both sides of the above equation and noting that 4-th and higher order derivatives of f are zero as f is a cubic polynomial, we obtain

$$\bigtriangledown^2 g(x, y) = \bigtriangledown^2 f(x, y) \tag{7.9}$$

Substituting the above equation in Equation (??) and rearranging terms we obtain

$$f(x, y) = g(x, y) - \frac{h_{2,0}}{2} \bigtriangledown^2 g(x, y) \tag{7.10}$$

Equation (??) 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 [?].

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

$$f(x,y) = g(x,y) - \frac{\sigma^2}{4} \nabla^2 g(x,y) \tag{7.11}$$

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. ??) 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 (??) 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 [?]. However that result is valid only for a diffusion model of blurring where the PSF is restricted to be a Gaussian. In comparison, our deconvolution formula 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 in our method of deconvolution using a more general version of the S-Transform [?]. Such generalization is not possible for the previously known result. In our deconvolution method, the focused image can be generalized to be an arbitrarily high order polynomial although such a generalization does not seem useful in practical applications that we know.

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. (??). 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.

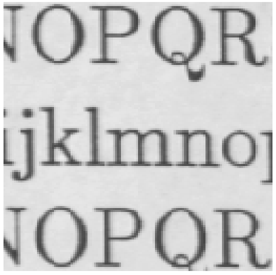
7.3.2 Experiments

A set of experiments is described in Section 7.5 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 7.1. The camera lens was moved to different positions (steps 70, 60, 50, 40, 30 and

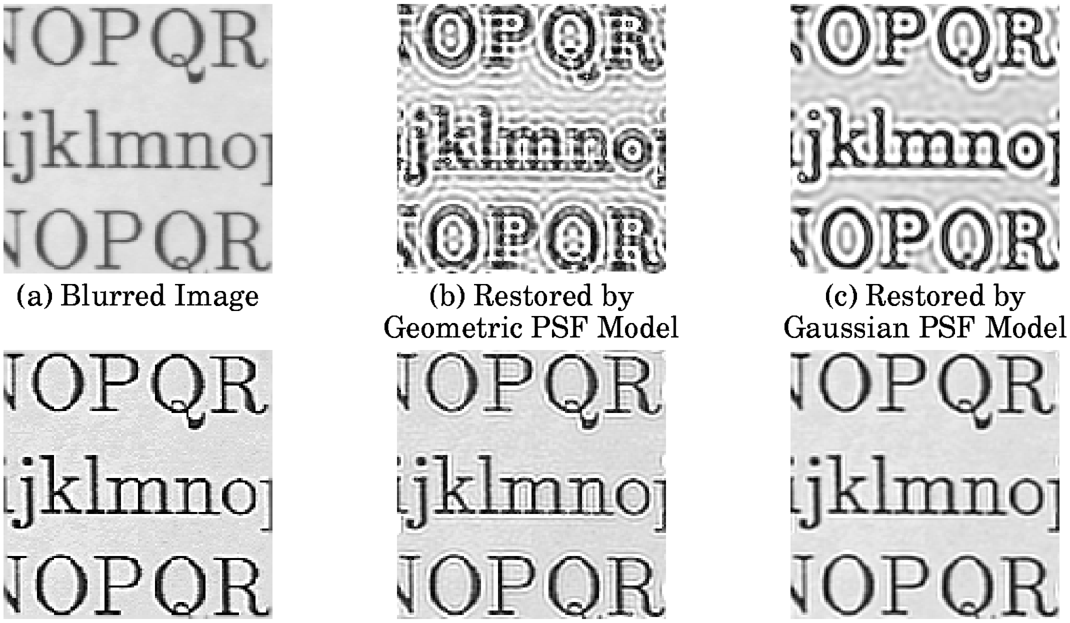
20) to obtain images with different degrees of blur. The images are shown in figures 7.2a-7.7a. The corresponding blur parameters (σs) for these images were roughly 2.2, 2.8, 3.5, 4.7, 6.0 and 7.2 pixels. These images were deblurred using equation (??). The results are shown in Figures 7.2d-7.7d. We 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 (Eq. ??). The spread parameter of the Gaussian function was taken to be equal to the blur parameter σ.



OPQR
ijklmno
OPQR

Fig. 7.1 Focused Image for Character



(a) Blurred Image

(b) Restored by Geometric PSF Model

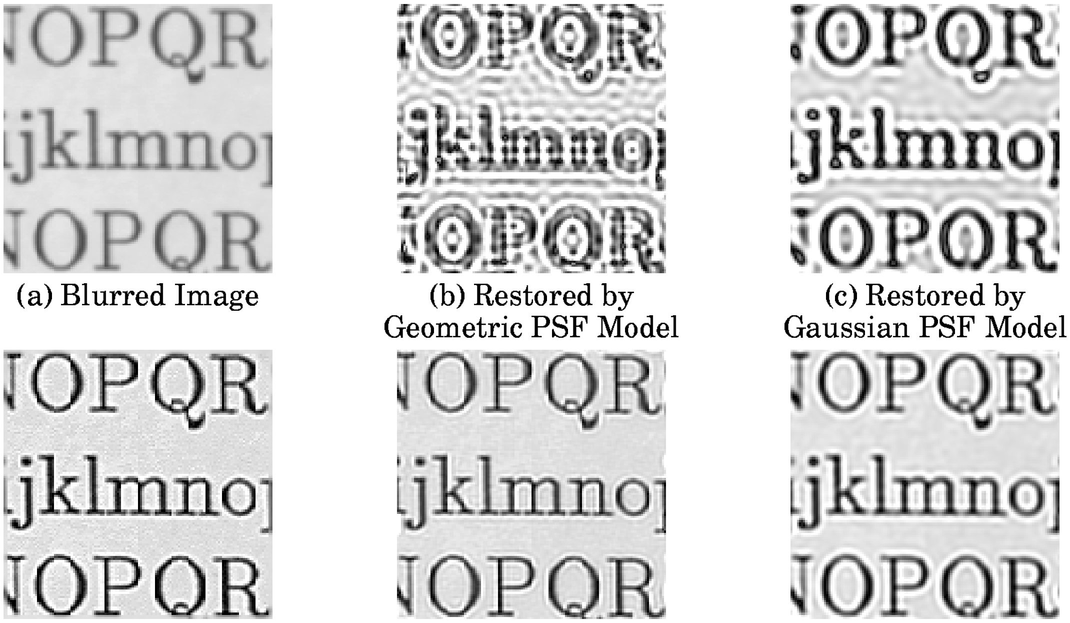
(c) Restored by Gaussian PSF Model

(d) Restored by S-Transform

(e) Restored by Separable MTF Model

(f) Restored by Actual PSF (Abel Transform)

Fig. 7.2 Restoration with 0 Step of Blur



(a) Blurred Image

(b) Restored by Geometric PSF Model

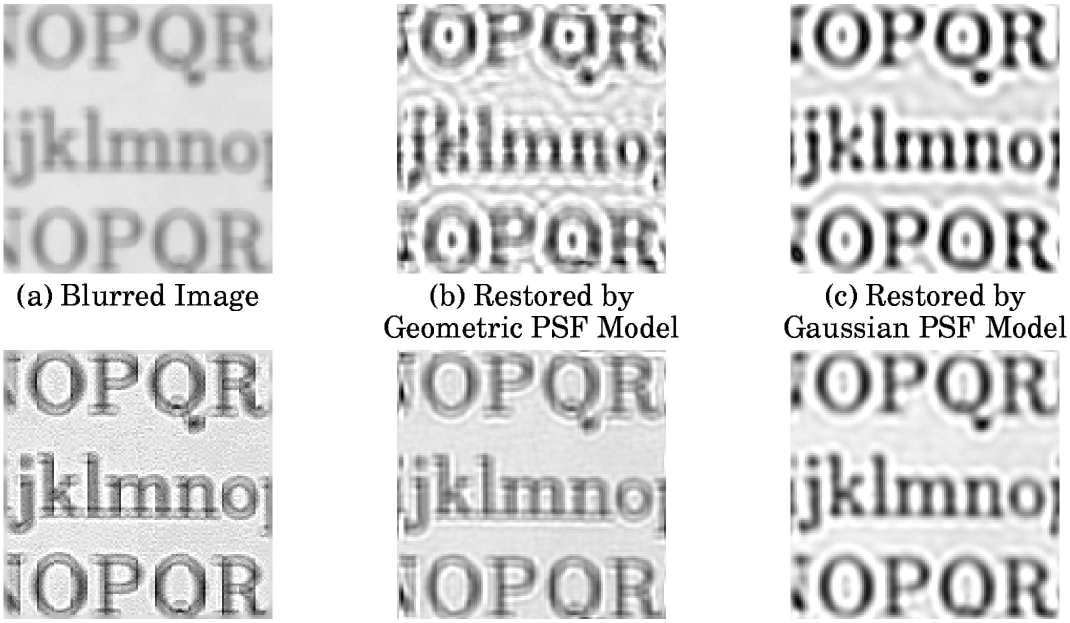
(c) Restored by Gaussian PSF Model

(d) Restored by S-Transform

(e) Restored by Separable MTF Model

(f) Restored by Actual PSF (Abel Transform)

Fig. 7.3 Restoration with 10 Steps of Blur



(a) Blurred Image

(b) Restored by Geometric PSF Model

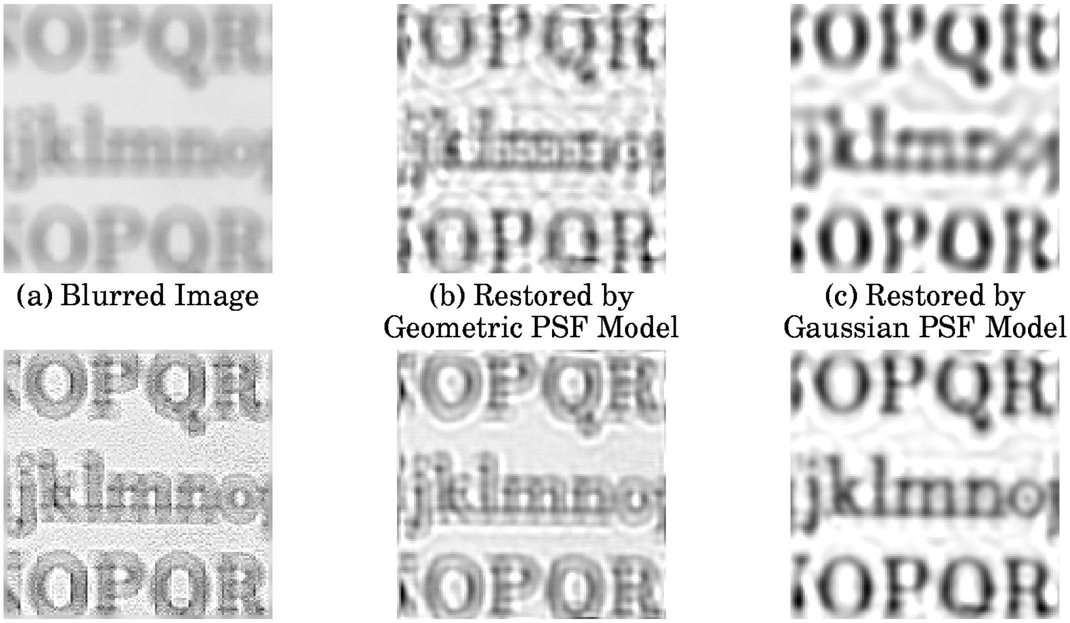
(c) Restored by Gaussian PSF Model

(d) Restored by S-Transform

(e) Restored by Separable MTF Model

(f) Restored by Actual PSF (Abel Transform)

Fig. 7.4 Restoration with 20 Steps of Blur



(a) Blurred Image

(b) Restored by Geometric PSF Model

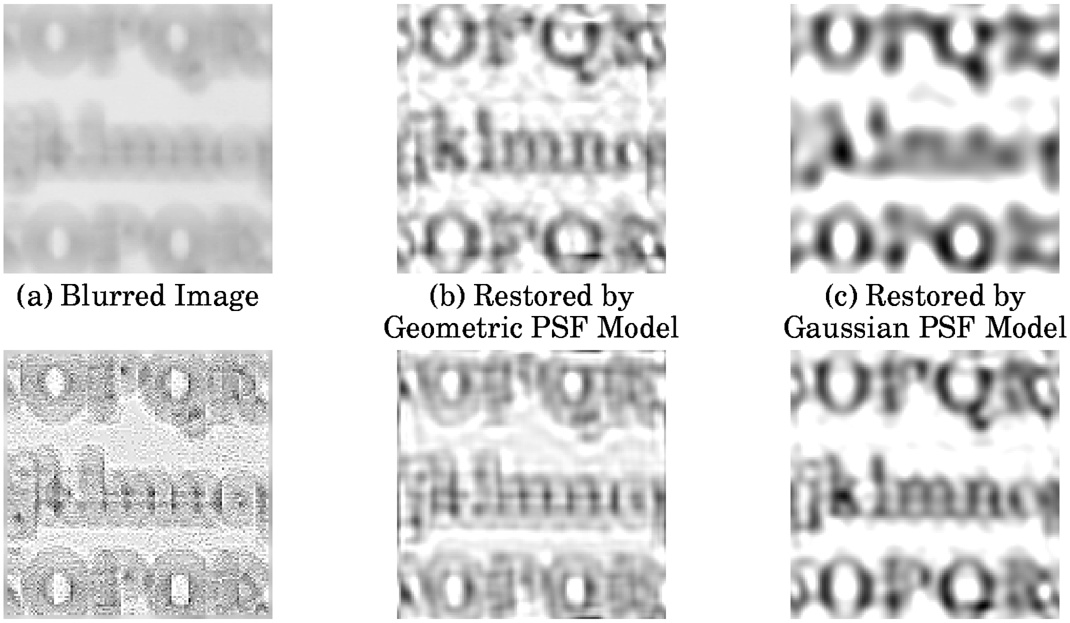
(c) Restored by Gaussian PSF Model

(d) Restored by S-Transform

(e) Restored by Separable MTF Model

(f) Restored by Actual PSF (Abel Transform)

Fig. 7.5 Restoration with 30 Steps of Blur



(a) Blurred Image

(b) Restored by Geometric PSF Model

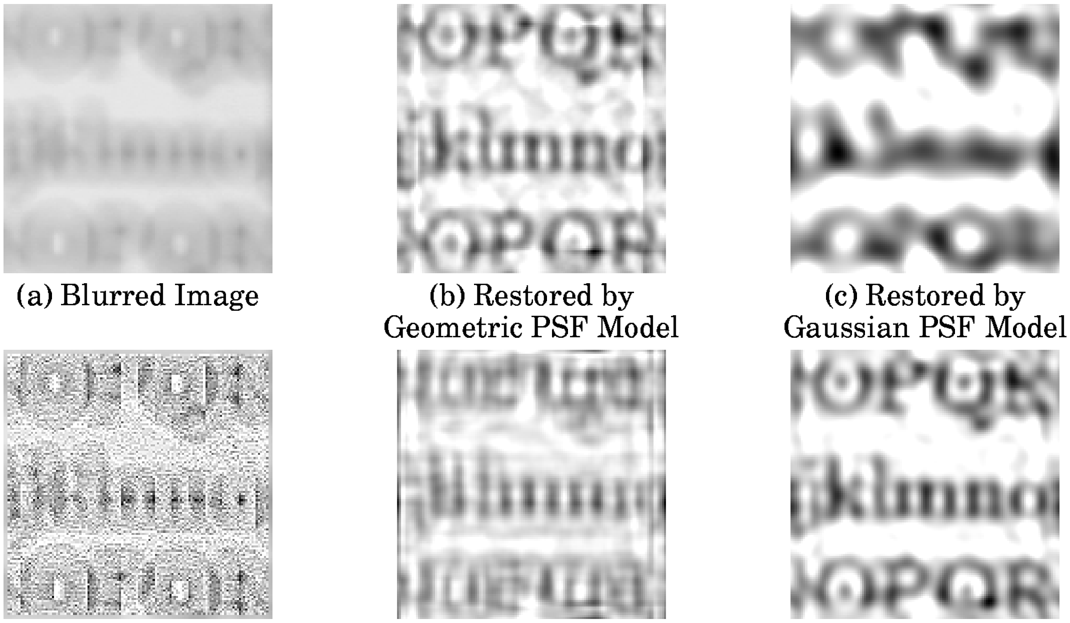
(c) Restored by Gaussian PSF Model

(d) Restored by S-Transform

(e) Restored by Separable MTF Model

(f) Restored by Actual PSF (Abel Transform)

Fig. 7.6 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 by Actual PSF (Abel Transform)

Fig. 7.7 Restoration with 50 Steps of Blur

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

$$M(\omega,\nu) = \frac{1}{H(\omega,\nu)} \frac{|H(\omega,\nu)|^2}{|H(\omega,\nu)|^2 + \Gamma} \tag{7.12}$$

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

$$\hat{F}(\omega,\nu) = G(\omega,\nu)M(\omega,\nu). \tag{7.13}$$

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 7.2c-7.7c. 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. 7.2d-7.7d). 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 Wiener 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 (Eq. ??). The relation between blur circle radius and spread parameter σ are taken to be $R = \sqrt{2}\sigma$. With a knowledge of the blur parameter σ, it is thus possible to use equation (??) and 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 7.2b-7.7b 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.

7.4 Second Method

In the second method, 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 filter is used to compute the focused image. First we describe a method of obtaining the prestored table through a calibration procedure.

7.4.1 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

$$f(x,y) = a + b\ u(x) \tag{7.14}$$

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,

$$g(x,y) = h(x,y) * f(x,y) \tag{7.15}$$

where * denotes the convolution operation.

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

$$\begin{aligned} \frac{\partial g}{\partial x} &= h(x,y) * \frac{\partial f}{\partial x} && (7.16) \\ &= h(x,y) * b\ \delta(x) && (7.17) \end{aligned}$$

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

$$\frac{\partial g}{\partial x} = b\,\theta(x) \tag{7.18}$$

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

$$\theta(x) = \int_{-\infty}^{\infty} h(x,y)dy \tag{7.19}$$

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

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h(x,y)\;dx\;dy = 1 \tag{7.20}$$

Therefore we obtain

$$\int_{-\infty}^{\infty} \frac{\partial g(x,y)}{\partial x} dx = b \tag{7.21}$$

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

$$\theta(x) = \frac{\frac{\partial g}{\partial x}}{\int_{-\infty}^{\infty}\frac{\partial g}{\partial x}dx} \tag{7.22}$$

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

$$\Phi(\omega) = H(\omega,0) \tag{7.23}$$

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

$$H(\omega, \nu) = \Phi(\sqrt{\omega^2 + \nu^2}) \tag{7.24}$$

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 that 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 [?] :

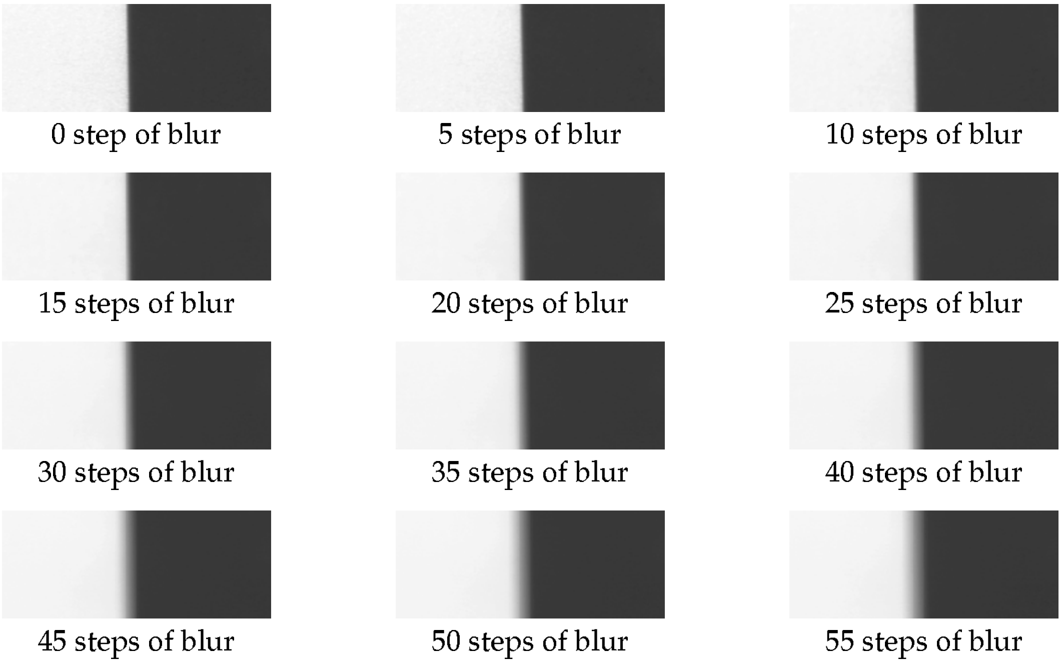
$$h(r) = -\frac{1}{\pi} \int_r^\infty \frac{\theta'(x)}{\sqrt{x^2 - r^2}} dx \tag{7.25}$$

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 (??) and (??) and obtain the restored image.

7.4.2 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 7.8. 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.



0 step of blur

5 steps of blur

10 steps of blur

15 steps of blur

20 steps of blur

25 steps of blur

30 steps of blur

35 steps of blur

40 steps of blur

45 steps of blur

50 steps of blur

55 steps of blur

Fig. 7.8 Step Edges for Calibration

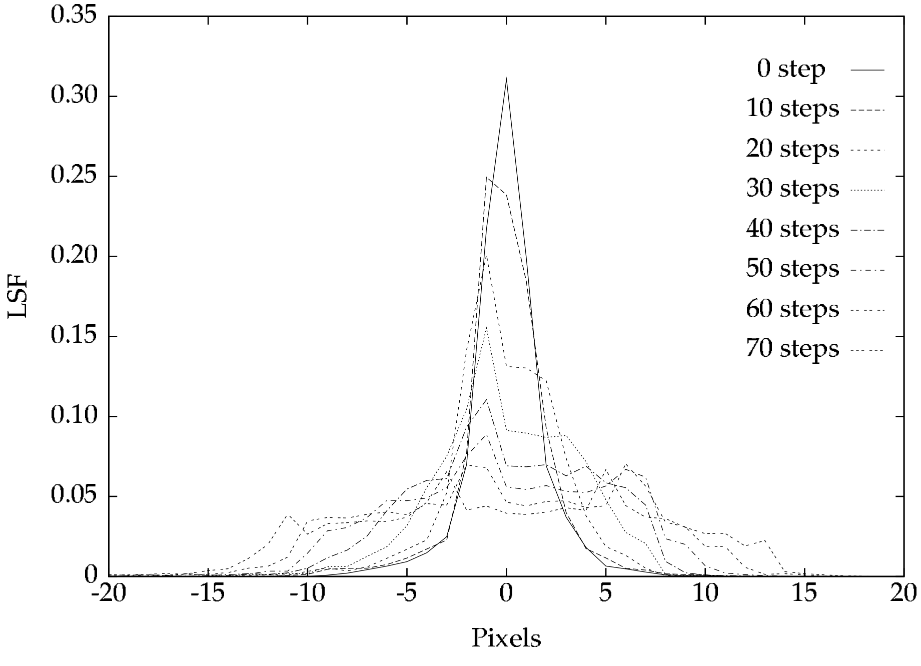


Fig. 7.9 LSF from Step Edges

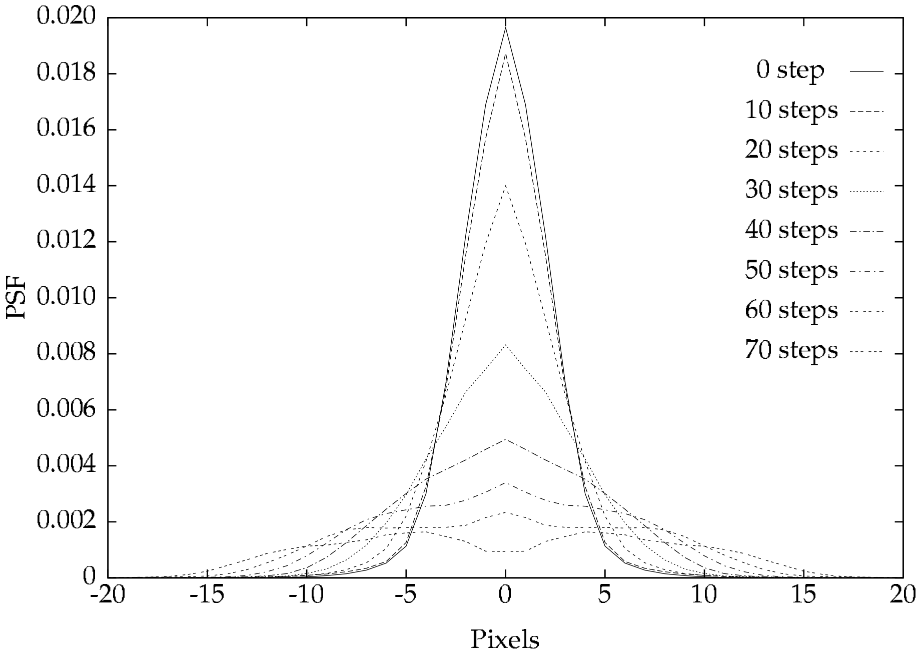


Figure 7.10: PSF by Inverse Abel Transform

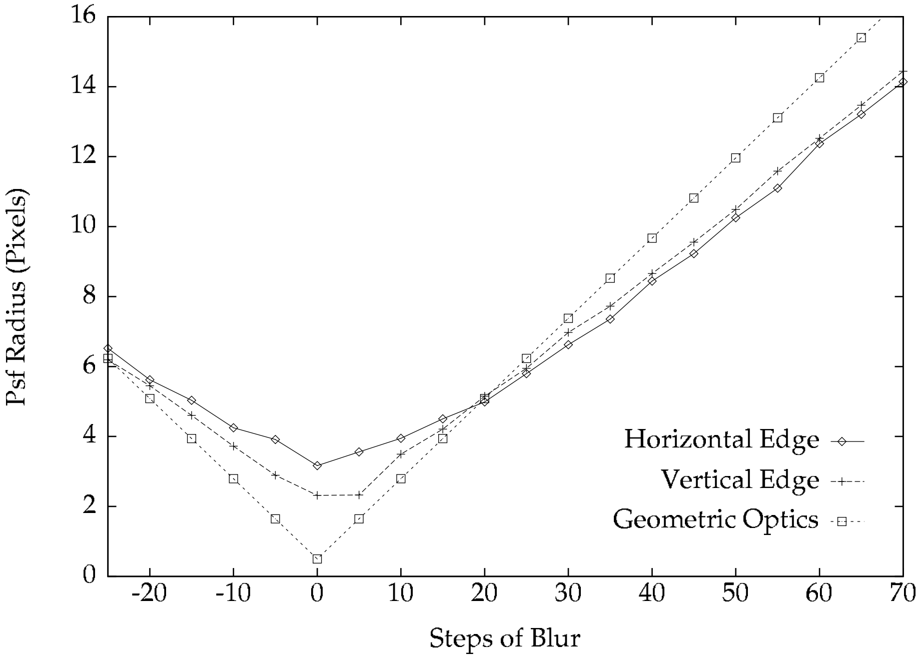


Figure 7.11: PSF Radius from Step Edges

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

$$\bar{i} = \frac{\sum_{i=1}^{i=200} i g_x(i)}{\sum_{i=1}^{i=200} g_x(i)} \tag{7.26}$$

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 line spread function was computed as

$$\theta(i) = \frac{g_x(i)}{\sum_{i=1}^{i=200} g_x(i)} \tag{7.27}$$

Eight LSFs corresponding to different degrees of blur are plotted in Figure 7.9. 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 (??). The spread or second central moment of the LSF, σ_l was computed from

$$\sigma_l = \sqrt{\sum_{i=1}^{200} (i - \bar{i})^2 \theta(i)} \tag{7.28}$$

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 using the relation $R = 2\sigma_l$ for different step edges are shown in Figure 7.11. Figure 7.11 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 7.9 are shown in Figure 7.10.

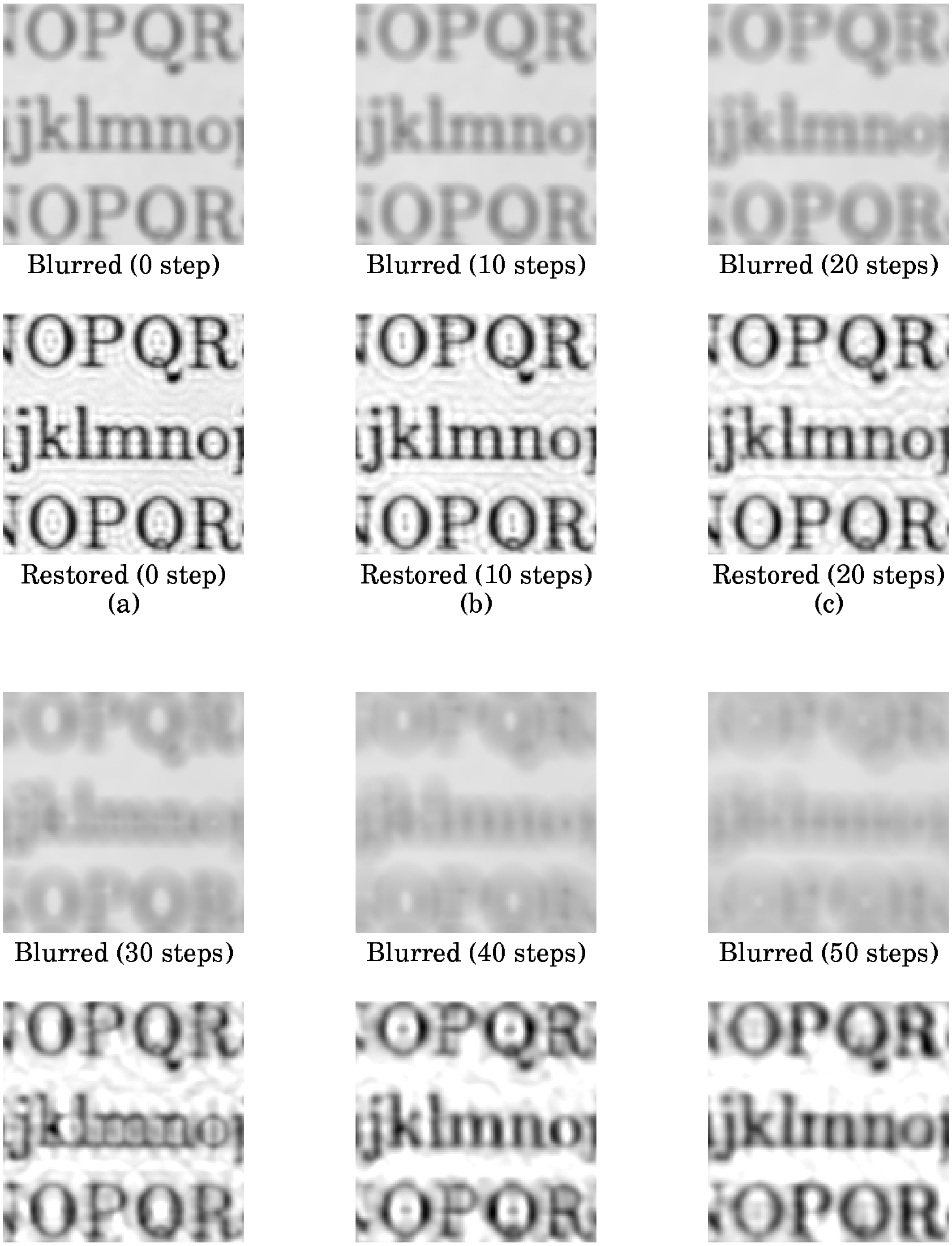
7.4.3 Experimental Results

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 7.2e-7.7e show the results of restoration using the separability assumption for the OTF and Figures 7.2f-7.7f 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 7.2(b,c,d)-7.7(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 7.6a and 7.7a 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 impossible for humans to recognize the characters in these images. However, in the restored images shown in Figures 7.6f and 7.7f 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 functions. 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 7.12 and 7.13. We see that these results are only somewhat better but not much better than the results on actual data in Figures 7.2f-7.7f. This indicates that our 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.



Blurred (0 step)

Blurred (10 steps)

Blurred (20 steps)

Restored (0 step)
(a)

Restored (10 steps)
(b)

Restored (20 steps)
(c)

Blurred (30 steps)

Blurred (40 steps)

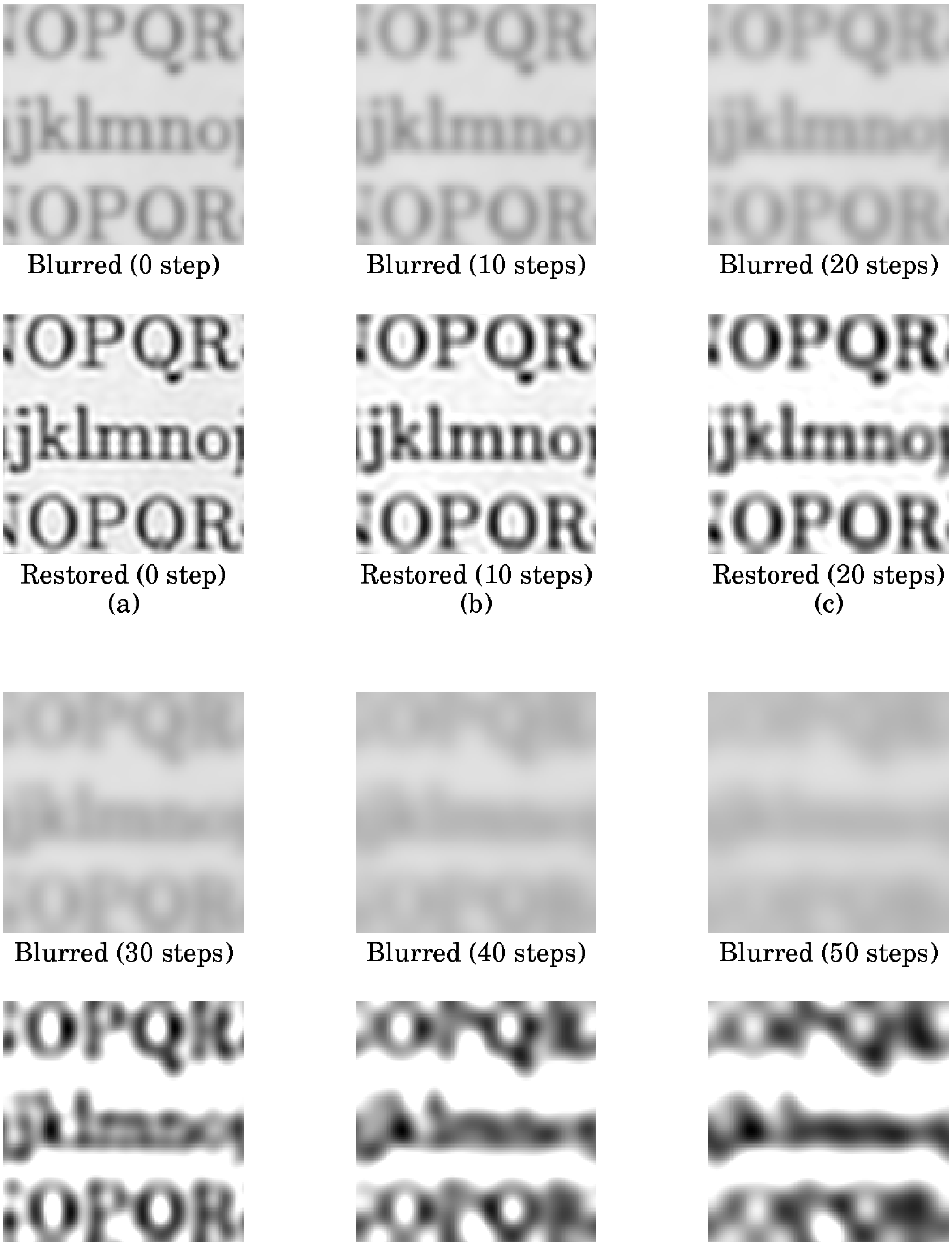
Blurred (50 steps)

Restored (30 steps)
(d)

Restored (40 steps)
(e)

Restored (50 steps)
(f)

Fig. 7.12 Simulation with Geometric Optics PSF



Blurred (0 step)
Blurred (10 steps)
Blurred (20 steps)

Restored (0 step)
(a)
Restored (10 steps)
(b)
Restored (20 steps)
(c)

Blurred (30 steps)
Blurred (40 steps)
Blurred (50 steps)

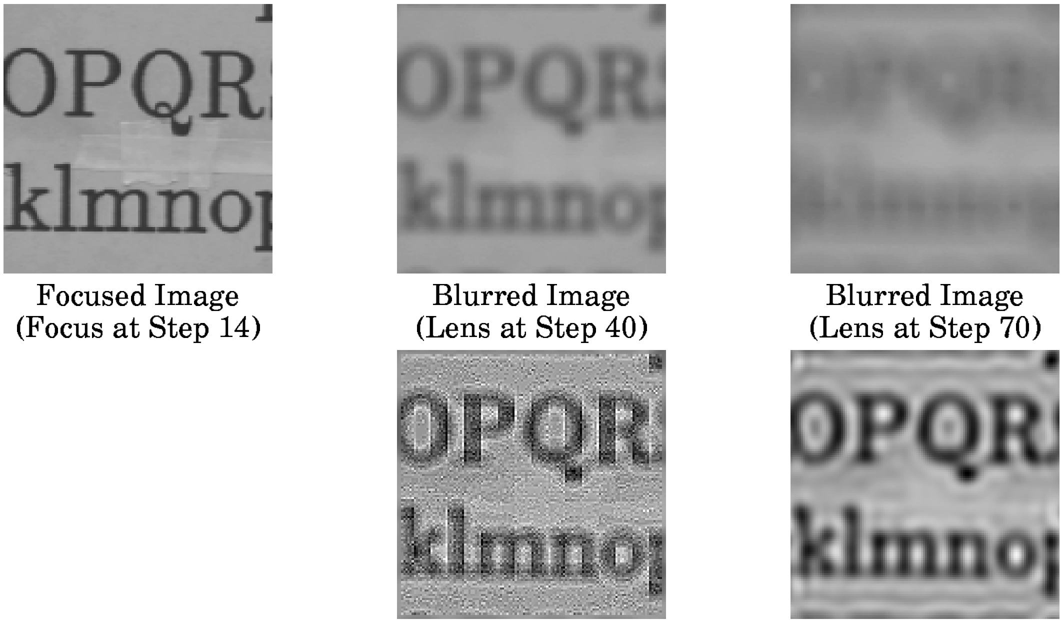
Restored (30 steps)
(d)
Restored (40 steps)
(e)
Restored (50 steps)
(f)

Fig. 7.13 Simulation with Gaussian PSF

7.5 Experiments with Unknown σ and 3D Object

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 one of the two depth-from-defocus methods proposed by us recently (DFD1F or 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 7.14(a-d). The first image in Fig. 7.14a 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. 7.14a). The blur parameter σ was estimated using the depth-from-defocus method proposed in [?]. 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. 7.14a. 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



Focused Image (Focus at Step 14)

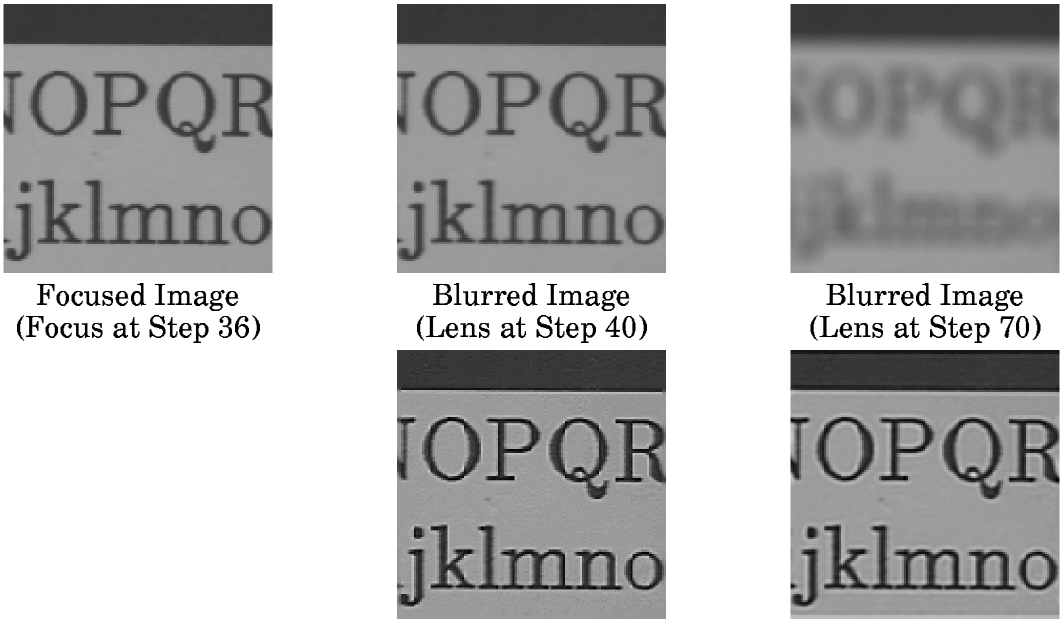
Blurred Image (Lens at Step 40)

Blurred Image (Lens at Step 70)

Restored by S-Transform

Restored by Actual PSF (Abel Transform)

Fig. 7.14(a) Depth Estimation with Restoration for Step 14



Focused Image (Focus at Step 36)

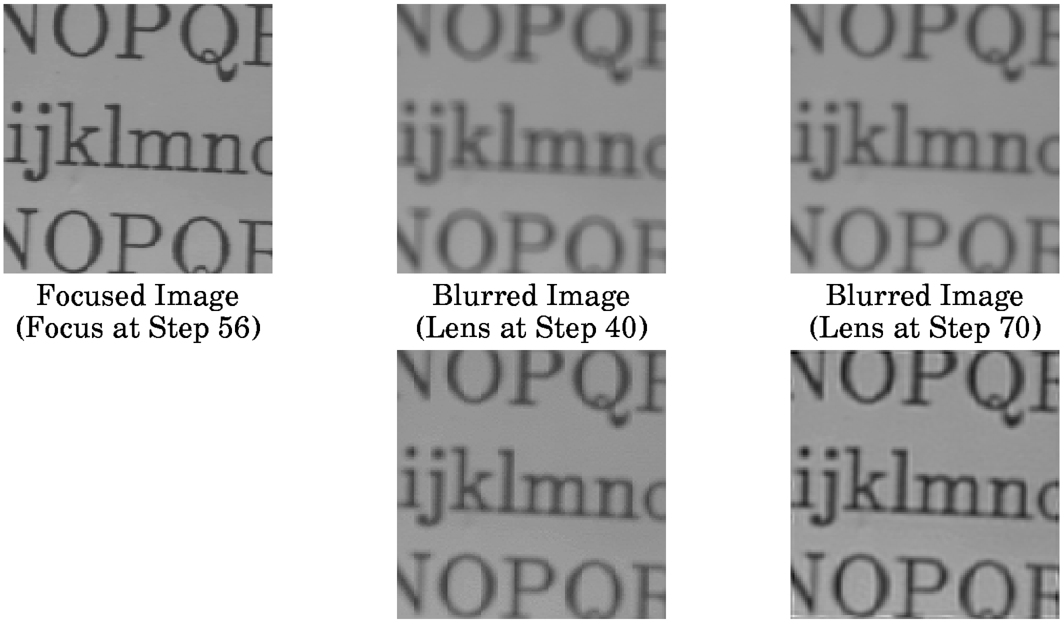
Blurred Image (Lens at Step 40)

Blurred Image (Lens at Step 70)

Restored by S-Transform

Restored by Actual PSF (Abel Transform)

Fig. 7.14(b) Depth Estimation with Restoration for Step 36



Focused Image
(Focus at Step 56)

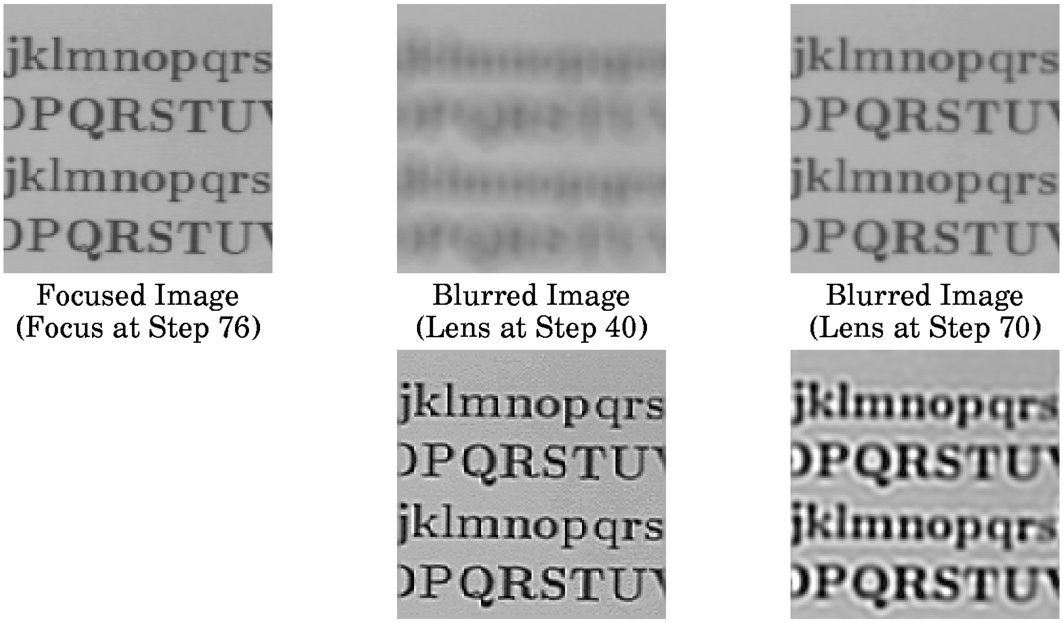
Blurred Image
(Lens at Step 40)

Blurred Image
(Lens at Step 70)

Restored by
S-Transform

Restored by Actual
PSF (Abel Transform)

Fig. 7.14(c) Depth Estimation with Restoration for Step 54



Focused Image
(Focus at Step 76)

Blurred Image
(Lens at Step 40)

Blurred Image
(Lens at Step 70)

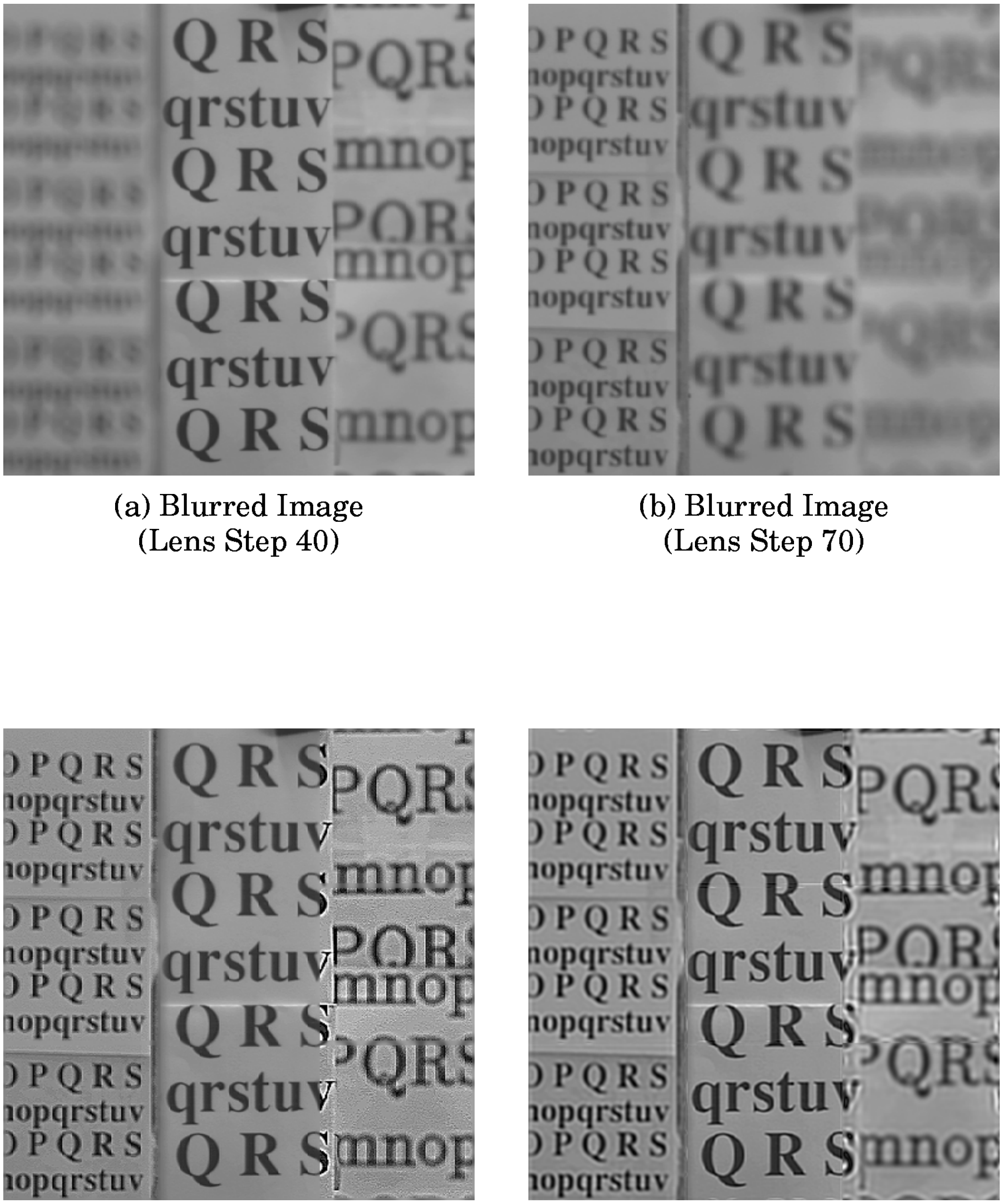
Restored by
S-Transform

Restored by Actual
PSF (Abel Transform)

Fig. 7.14(d) Depth Estimation with Restoration for Step 76 and 70, and the restored images are shown in Figs. 7.14b-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 3D 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 7.15. 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 7.15 shows the results using both the first and second methods of restoration. Currently each region can be as small as 48 x 48 pixels, which is a small region in the entire field of view of 640 x 480 pixels.

In the last experiment, a planar object 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 FIg. 7.16(a,b). The images were divided into non-overlapping regions of 64×64



(a) Blurred Image
(Lens Step 40)

(b) Blurred Image
(Lens Step 70)

(c) Restored by
S-Transform

(d) Restored using Actual
PSF (Abel Transform)

Fig. 7.15 Depth Estimation with Restoration for 3-D Object

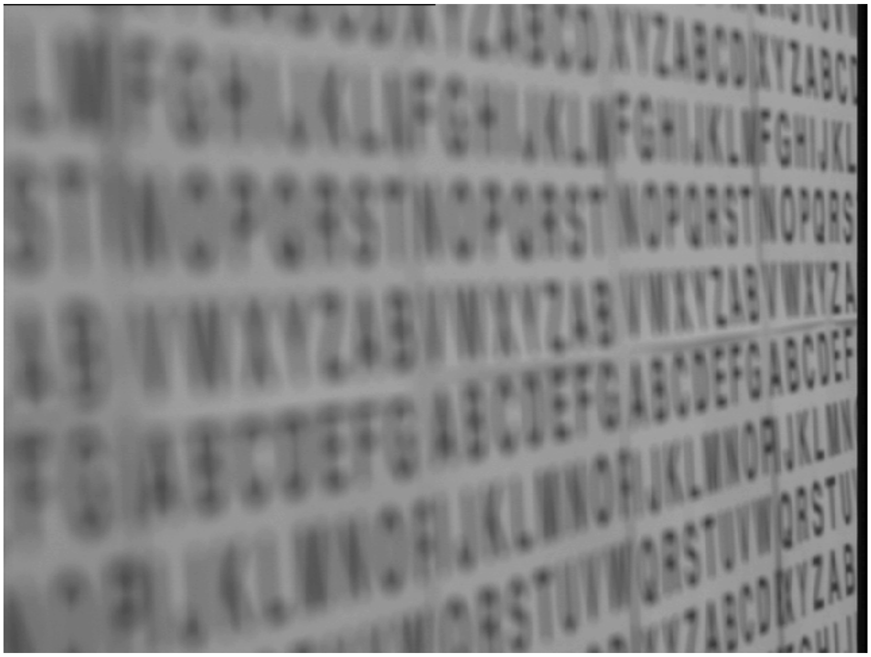


Figure 7.16(a): Image of Slanted Object with Lens Step 40

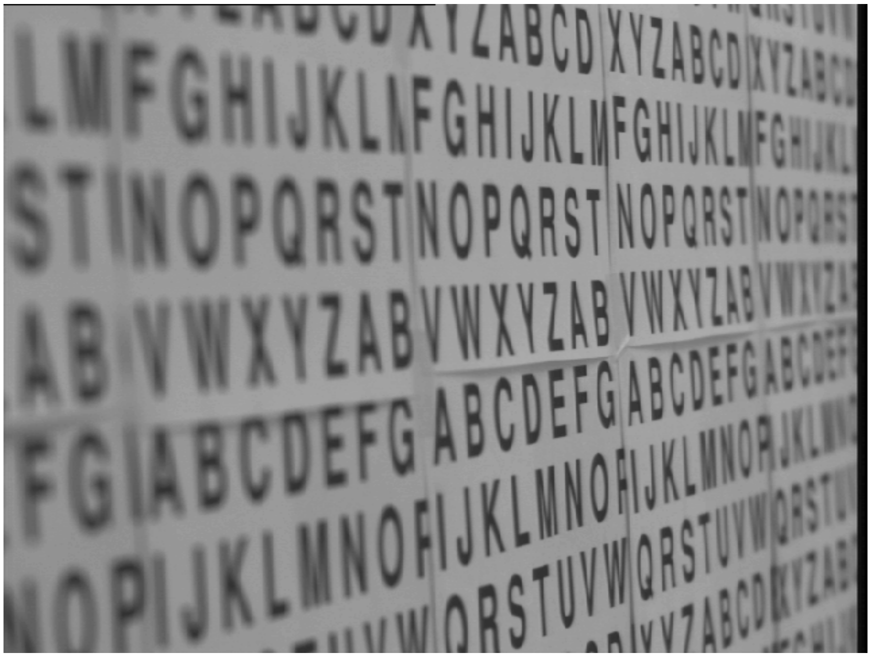


Figure 7.16(b): Image of Slanted Object with Lens Step 70

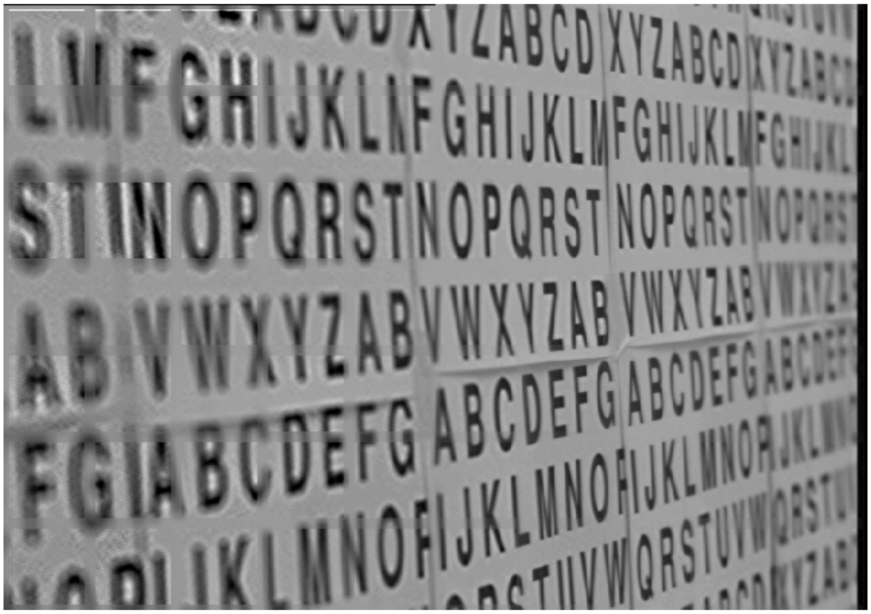


Figure 7.16(c): Restored with S-Transform

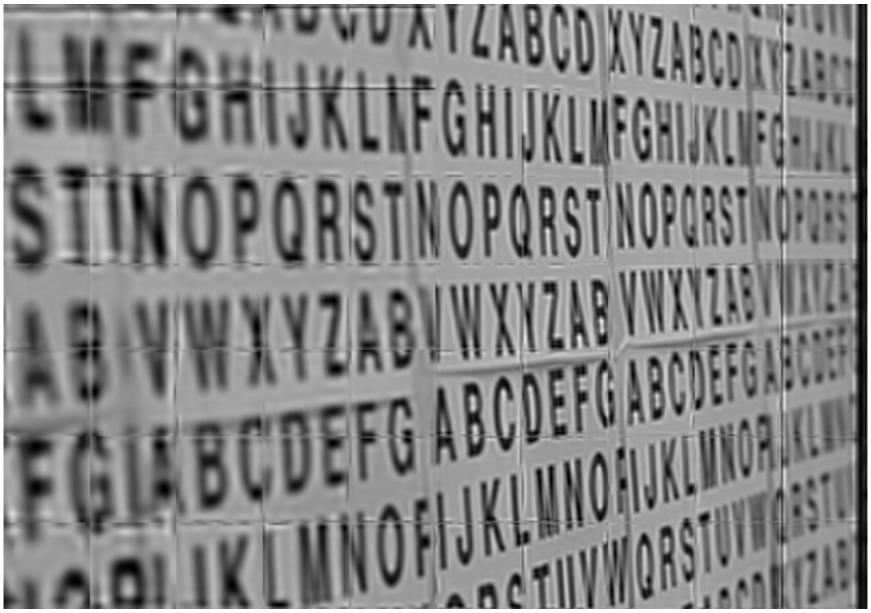


Figure 7.16(d): Restored Using Actual PSF (Abel Transform)

pixels and a depth estimate was obtained for each region. The different regions were then restored separately as before and combined to yield the restored image as shown in Fig. 7.16(c,d). The restored images appear better than either of the blurred images. However, there are some blocking artifacts, which are due to the "warp around" problem of the FFT algorithm and the finite filter size in the case of the S-Transform method.

7.6 Conclusion

The focused image of an object can be recovered using two defocused images recorded with different camera parameter settings. The same two images can used to estimate the depth of the object using a depth-from-defocus method. For a 3D 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 one of the two recorded defocused images is blurred only moderately or less ($\sigma <= 3.5$ 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 in this chapter can be used as part of a 3D machine vision system to obtain focused images from blurred images for further processing such as edge detection, stereo matching, and image segmentation.

Chapter 8

Integration of Stereo and DFD1F

8.1 Introduction

In human vision, the primary source of depth information is stereo disparity. The images formed in the left eye and and the right eye are first processed to establish correspondence between image points in the two images. Then the distance of objects in the 3D scene are determined through triangulation. A similar method is used for depth recovery in machine vision using two cameras. A review of stereo ranging methods can be found in [?].

The primary computational task in stereo is establishing correspondence between points in the left and right images. This problem is known as the correspondence problem. It is complicated by occlusion, i.e. some object points visible in one image may not be visible in the other image. The computational time required for solving the correspondence problem can be reduced by using a rough depth-map information obtained from other methods such as motion, shading, focus, and defocus. A number of approaches of combining different methods with stereo have been studied by researchers [?, ?, ?, ?, ?, ?, ?, ?]. In this chapter we consider the integration of Depth-from-Defocus and Stereo.

Stereo ranging is much more accurate than depth-from-defocus method but stereo is computationally much slower than depth-from-defocus method. By combining the two methods, the correspondence problem associated with stereo can be simplified and the coarse accuracy associated with depth-from-defocus can be improved. In the combined method, first a rough depth-map is obtained using depth-from-defocus method. Then the rough depth-map is used to solve the correspondence problem. Next the stereo triangulation principle is used to recover an accurate depth-map of the scene.

In this chapter we illustrate the integration of DFD1F and stereo with an example. We use an area matching algorithm, Sum of Squared Differences (SSD) [?, ?], to solve the correspondence problem. The SSD method is simple and it serves our purpose well. It should be noted that DFD1F can also be used with other correspondence matching algorithms. In our experiment, we were able to speed up the matching process by approximately a factor of three by integrating DFD1F and stereo. Further improvement can be achieved by a better calibration of the stereo camera system.

8.2 Stereo Camera System

A stereo camera system with two cameras is shown in Fig. 8.1. Two cameras with parallel optical axes are displaced by a distance b. The displacement b is often referred to as the baseline. Origin is taken at the center of the baseline and the x and z axes are taken to be parallel to the baseline and the optical axes respectively. Image coordinate systems (x_l, y_l) and (x_r, y_r) are defined on the left and the right image planes with origin on the optical axis and the axes parallel to the (x, y) axes.

An object point (x, y, z) will form two image points on the left and right images as points (x_l', y_l') and (x_r', y_r'). The following relation can be found by triangulation[?]

$$\frac{x_l'}{f} = \frac{x + b/2}{z} \qquad \text{and} \qquad \frac{x_r'}{f} = \frac{x - b/2}{z} \tag{8.1}$$

with

$$\frac{y_l'}{f} = \frac{y_r'}{f} = \frac{y}{z} \qquad \text{and} \qquad \frac{x_l' - x_r'}{f} = \frac{b}{z} \tag{8.2}$$

Solving for the unknowns

$$x = b\frac{(x_l' + x_r')/2}{x_l' - x_r'}, \qquad y = b\frac{(y_l' + y_r')/2}{x_l' - x_r'}, \qquad z = b\frac{f}{x_l' - x_r'} \tag{8.3}$$

The difference $x_l' - x_r'$ is called disparity. We will denote disparity by d. Two immediate observations can be seen from these equations. First, disparity is inversely proportional to object distance z. This indicates that the accuracy of distance measured through stereo is more accurate for closer objects than for far away objects. Secondly, disparity d is proportional to b, i.e., a larger baseline will have a better accuracy in

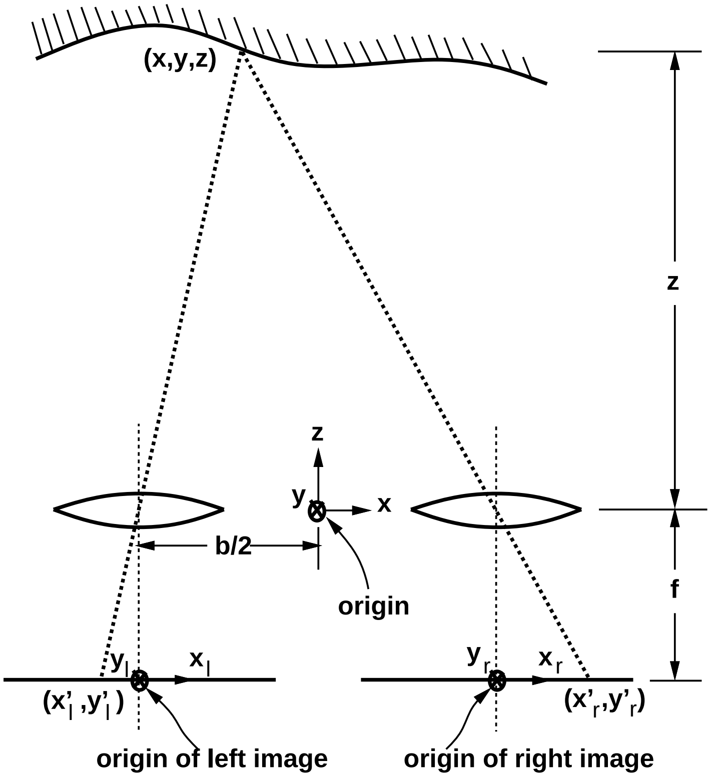


Figure 8.1: Schematic of Stereo Camera System

object distance recovery. However, the trade-off in this case will be a larger matching space and objects are more likely to be seen in one image only but not both.

8.3 Experiments

First the stereo system was calibrated to establish the relationship between 3D world coordinates and their corresponding 2D image coordinates. Once this relationship is established, 3D information can be inferred from 2D information and vice versa. The calibration was done as follows. A step edge was used to align the two cameras in the horizontal direction to make the two optical axes nearly parallel to each other. Then a set of fixed spacing dot patterns were used to find the disparity at various distances. Two images of the dot patterns taken by the left and right cameras at distance 90 cm are shown in Fig. 8.2. We have highlighted the areas that will be used to find the disparity for this particular distance. Within the area, the centroid of each dot was computed and matched to the corresponding point in the other image. The disparity for this distance was computed as the average disparity of all the points within the area. In this case, the average of 64 points was used. Table 8.1 shows the lens step number verses distance and disparity obtained by this calibration method. The disparity at distances ∞ and 530 cm were obtained through interpolation using (??).

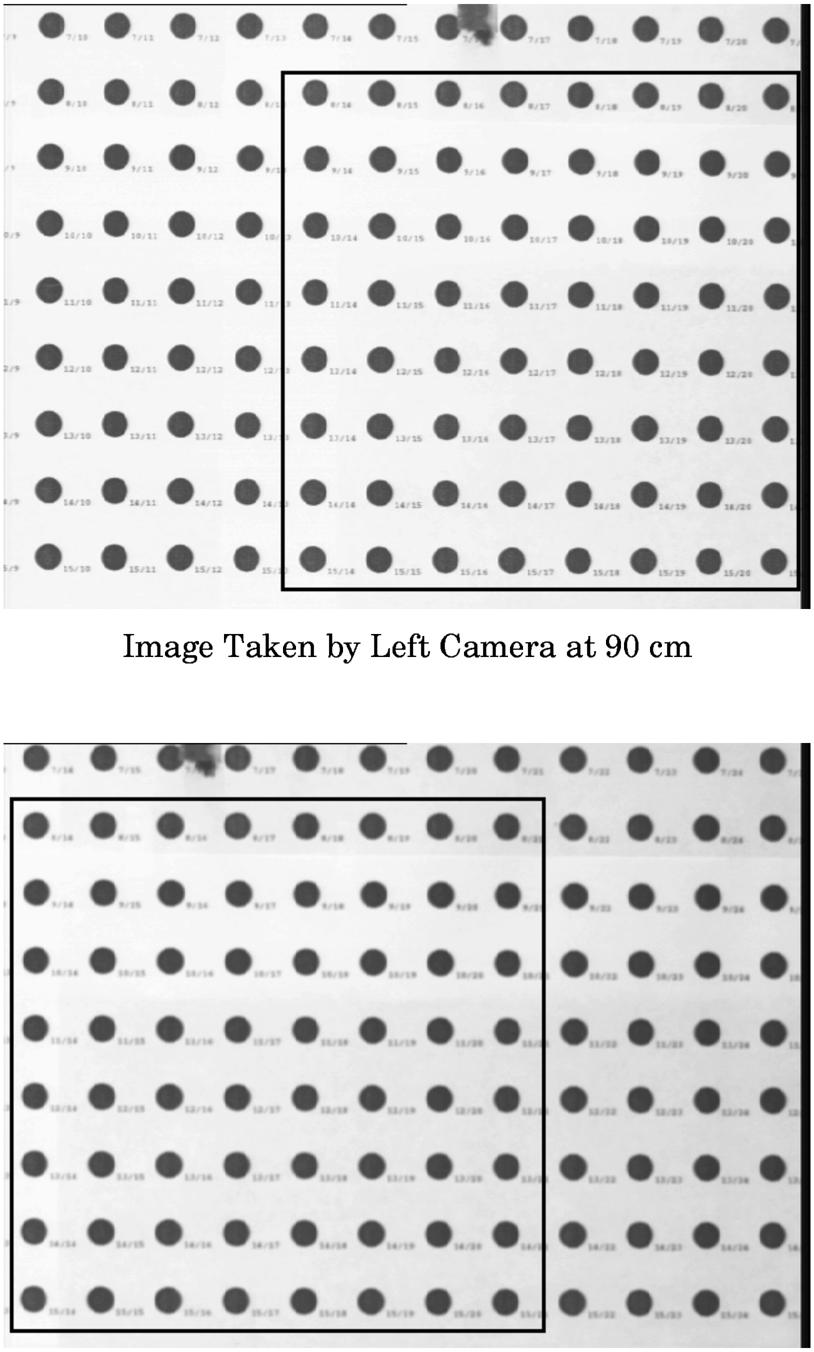


Image Taken by Left Camera at 90 cm

Image Taken by Right Camera at 90 cm

Figure 8.2: Equally Spaced Dotted Patten Used in Calibration

Lens Step	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95
Distance(m)	∞	5.30	3.75	2.85	2.50	1.93	1.72	1.465	1.32	1.17	1.08	0.965	0.90	0.822	0.77	0.715	0.67	0.628	0.595	0.56
Disparity(pixel)	-48	-17	-5	20	31	65	80	110	126	155	172	201	218	246	267	291	314	337	359	388

Table 8.1: Lens Step vs Best Focused Distance and Disparity

We have earlier expressed object distances in terms of the focused lens position specified as step number of the stepper motor of the lens. We have also seen that this lens step number is linearly related to inverse distance. Therefore, we can expect a linear relationship between disparity and object distance specified in lens steps since disparity is also linearly related to inverse distance. This can be verified from the plot of disparity versus lens steps in Fig. 8.3. A straight line was fitted to represent this relationship. For our stereo camera system, the relation between disparity d and lens step l can be expressed as

$$d = a_0 + a_1\, l \qquad \text{with} \qquad a_0 = -54, \;\text{ and }\; a_1 = 4.616. \tag{8.4}$$

We will be using this equation to find the matching space in our

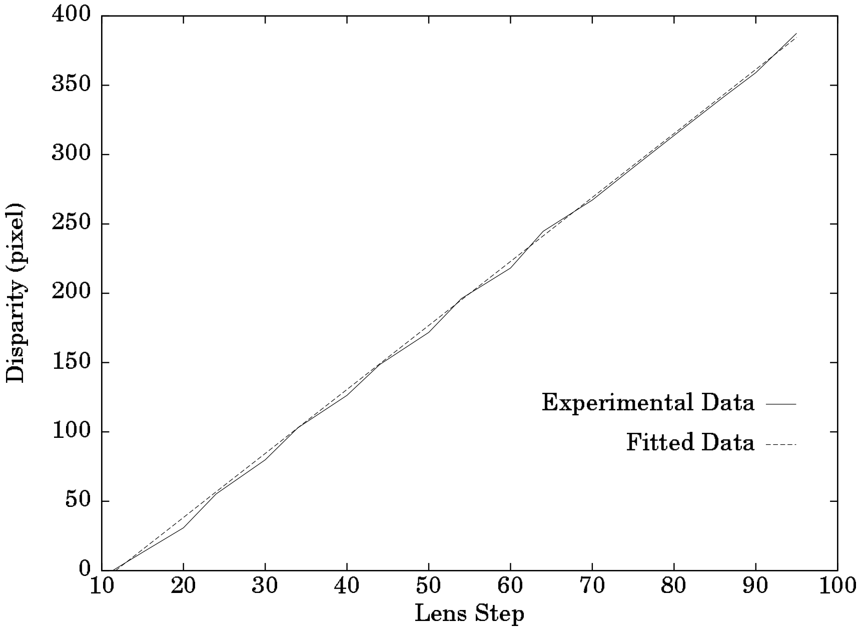


Figure 8.3: Disparity vs Lens Step

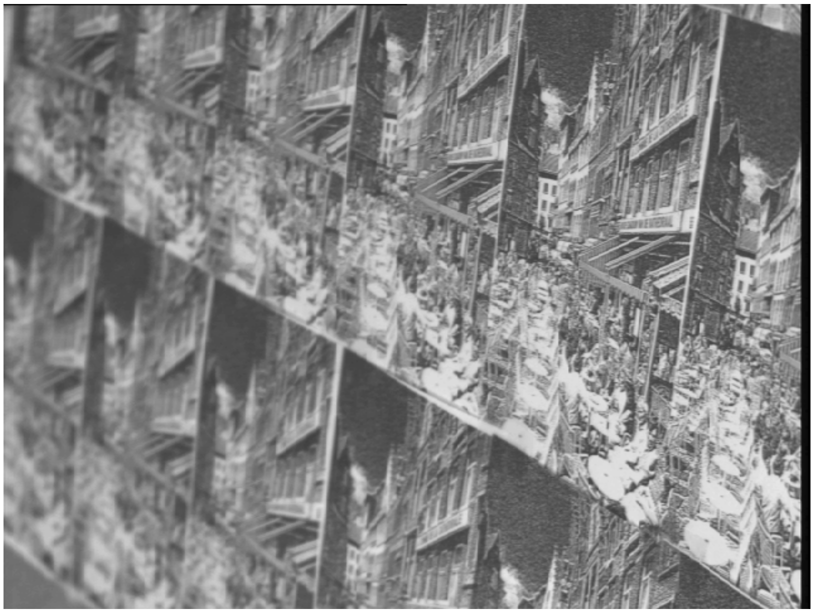


Image Taken by Left Camera

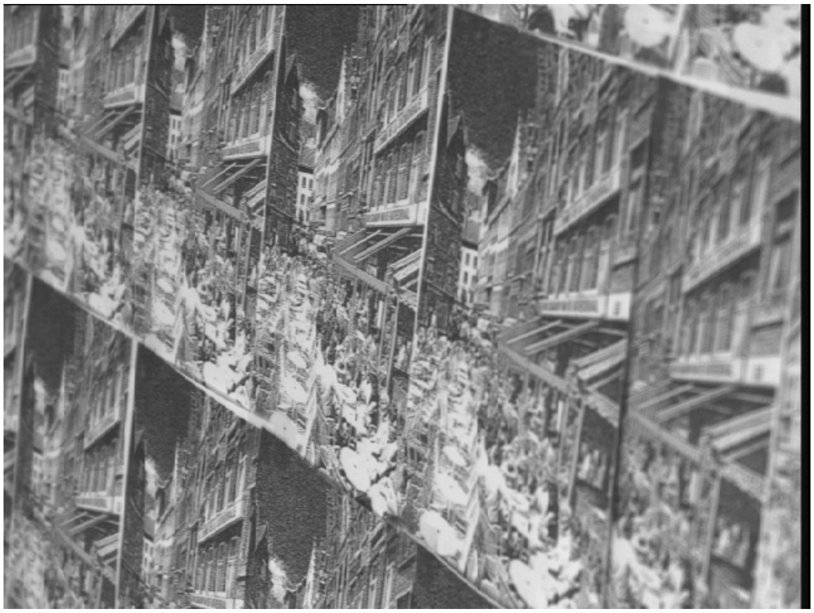


Image Taken by Right Camera

Figure 8.4: Slanted Object Used in Experiment

experiment.

A slanted object was used in our experiment. The depth of the object extended from 60 cm to 150 cm but the part visible in both cameras was from about 70 cm to 120 cm. The two images of size 640 × 480 taken by the two cameras are shown in Fig. 8.4. In the SSD method, a point in the left image was matched to each point on the corresponding epipolar line in the right image. In the actual implementation, since the initial calibration to determine the epipolar line was not accurate, the search for the best match was made in a narrow image region that included 3 rows above and 3 rows below the expected epipolar line. A window size of 11 × 11 was used to compute the sum of squared differences. The point that gave the minimum SSD was taken to be the match point. Our implementation was not computationally optimal and therefore the computation time on SUN SPARCstation IPX was about 150 minutes.

In Fig. 8.5, an object point moving along line L will be recorded on the left image at point p'_l. The same point will form an image somewhere on the epipolar line, depending on the distance of the object. For object distance P, a rough depth estimation can be obtained, (P_- to P_+), by applying DFD1F on the left camera. Therefore, matching space can be reduced to in between p'_+ and p'_- instead of the whole epipolar line.

In the combined method, a rough depth-map in terms of camera lens steps is first computed for the left image using DFD1F. This information is then used to reduce the number of possible matching points on the right image. From our earlier discussion, the RMS error for DFD1F was

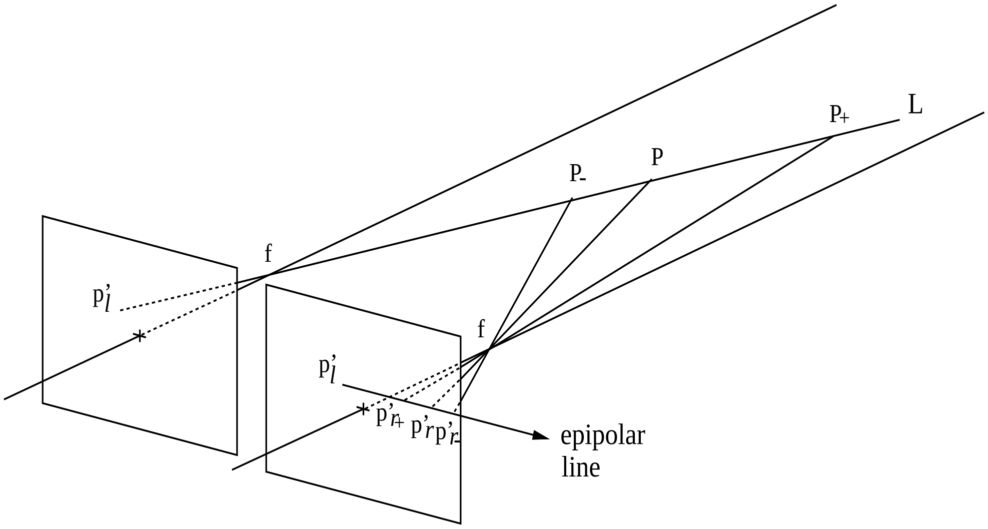


Figure 8.5: Epipolar Line

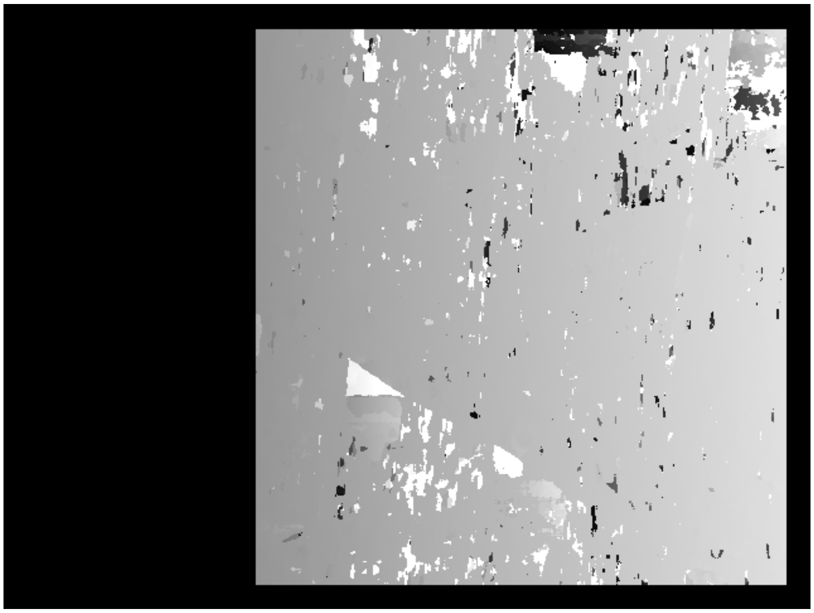
3.6 steps. Therefore, assuming an error of roughly three times the RMS error, the worst case error was taken to be 10 steps. Substituting ±10 in Eq. ??, we obtain the search space for matching to be 92 pixels. Therefore, instead of searching the entire epipolar line of size 640 pixels, only 92 pixels were searched. The implementation of the SSD matching algorithm in this case was similar to the previous case. The computational time in this case was roughly 50 minutes, about 1/3 of the time in the previous case.

In general, let DFD1F indicate the focused step number to be k for a point (x_l', y_l') in the left image. Also, let $\pm m$ steps be the confidence interval for the distance of the point in lens steps. One will then need to search for match points on the right image within the range

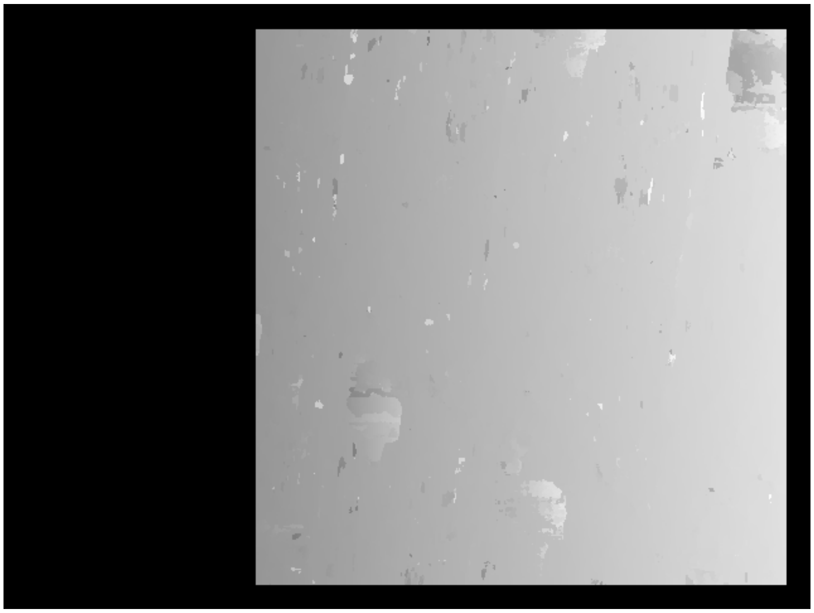
$$(x_l' + a_0 + a_1\ (k - m), y') \longleftrightarrow (x_l' + a_0 + a_1\ (k + m), y') \tag{8.5}$$

provided that the potential match point is not outside of the right image. With this approach, the matching space depends on a portion of the epipolar line predicted by DFD1F instead of the whole epipolar line.

Experimental results are shown in Fig. 8.6 and Fig. 8.7. Both SSD method and combination of DFD1F and SSD method were implemented for the sake of comparison. In Fig. 8.6, the disparity at each pixel is displayed on the image as gray levels– the brighter pixel the larger the disparity. In Fig. 8.7, the distance obtained by both methods are plotted. It shows that by combining SSD with DFD1F, the number of mismatch points can be reduced and a better result can be obtained with less computation time.

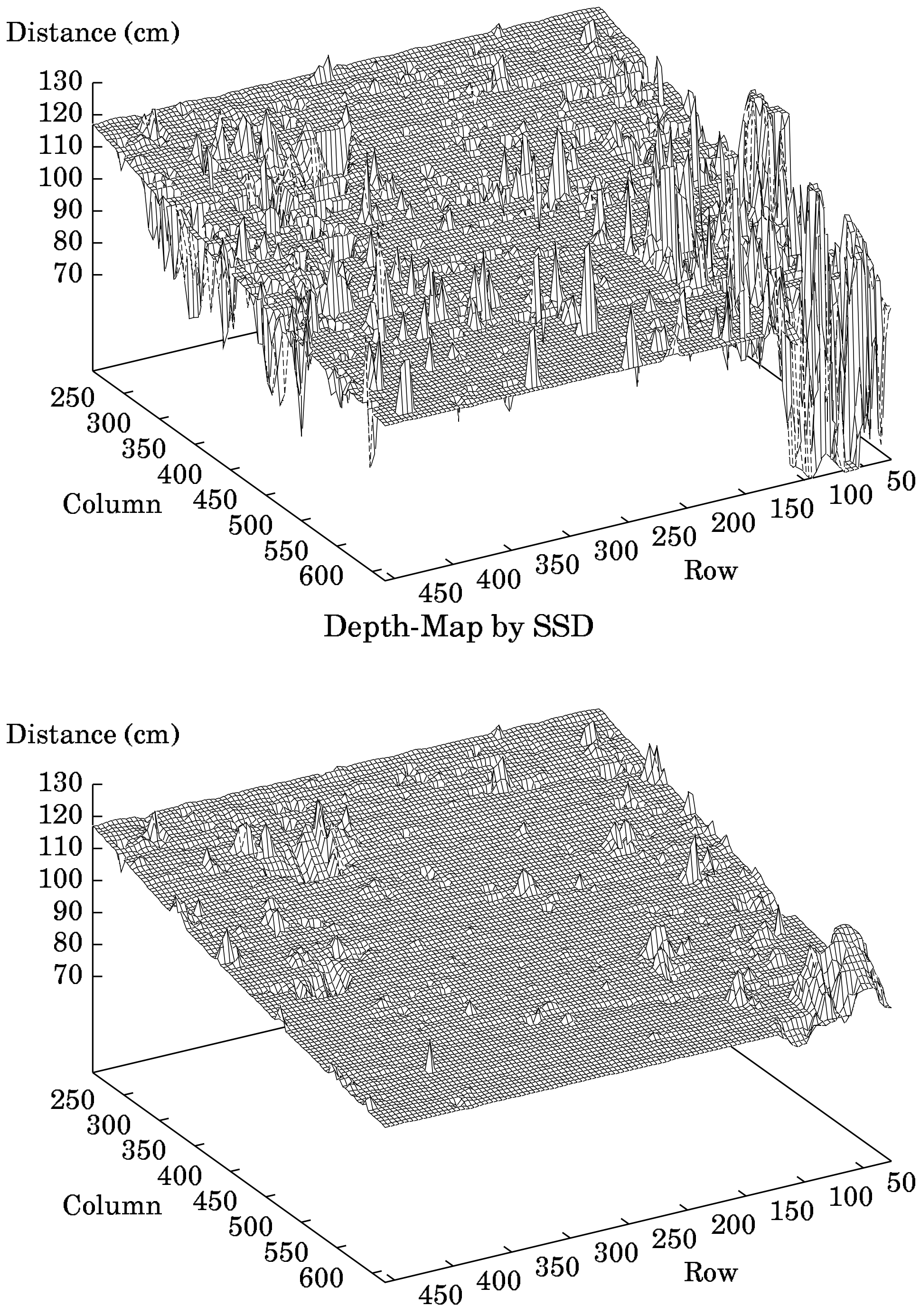


Disparity by SSD



Disparity by Combining DFD with SSD

Figure 8.6: Experimental Results, Disparity Map



Depth-Map by SSD

Depth-Map by Combining DFD and SSD

Figure 8.7: Experimental Results, Depth Map

Chapter 9

Conclusion

In this dissertation, we started with the investigation of Depth-from-Focus (DFF) method. This method is essentially a search method that requires a large number of images. Based on the focus measures computed on each of these images, the 3D shape of the scene is reconstructed. An implementation of DFF is tested on images from a video camera system and a microscope. Due to the large number of images required and the mechanical movements of the camera system, DFF method is slow when compared with the Depth-from-Defocus (DFD) method. However, the accuracy of DFF is usually better than DFD method.

The main subject of this dissertation is a new DFD method– DFD1F. The method is based on computing the Fourier coefficients of the images. By processing two images of the scene with different degrees of blur, DFD1F can find the depth information with a root-mean-square error of 3.7% for stationary objects. This figure was obtained using a prototype camera system based on a large number of experiments. Although the theoretical basis of DFD1F is relatively simple, much creative engineering is needed to make the method work on a real camera system. Complete details of our implementation of DFD1F are presented in this dissertation.

We extended DFD1F to continuous focusing of moving objects. Focusing of moving objects requires simultaneous acquisition of two images with different camera parameter settings. We have proposed some camera structures to accomplish this task. The other problem in making DFD1F work on moving objects is the large memory space needed to store a lookup table. This problem has been overcome by a parameterization scheme for the camera's MTF data. Experimental results show that the method has an RMS error of 4.3% in terms of lens position. In a commercial camera system, this amount of error is hardly noticeable by human eyes due to the large depth-of-field.

The two images used by DFD1F can also be used to find the focused image. This is done by first estimating the spread parameter of the camera point spread function. Two methods are presented here for focused image restoration. The trade off between the two methods are speed and performance. For a small blur, the first method based on S-Transform can be used to restore the focused image. It requires only local operations and no camera calibration is necessary. For a larger blur, the second method based on calibrated PSF can be used to produce better results. But the second method requires the computation of 2D FFT. Hence, it is slower.

It is possible to find a coarse depth map of an entire scene by dividing the scene image into many small subimages and applying DFD1F on these subimages. The coarse depth-map thus obtained can be used by a stereo ranging camera system to speed up correspondence matching. Experiments on a slanted object using DFD1F and stereo have been presented to demonstrate the advantages of combining DFD1F and stereo.

The research reported in this thesis can be further extended in many ways. We think that the accuracy and performance of DFD1F can be further improved using new techniques and more than the 6 Fourier coefficients used in our implementation. It is also of interest to investigate the integration of DFF with DFD and stereo, and possibly other approaches such as motion and shading. Designing a special-purpose hardware using DSP chips for real-time implementation of DFD1F is important in real-time 3D machine vision. Techniques for calibrating and implementing DFD methods on a microscope are useful in medical and biological image analysis.

Appendix A

Error Senstivity Analysis of DFD1F

In this appendix, we will analyze the error in depth estimation due to some small error in camera parameters. A numerical example will be provided. From the discussion in chapter 2 and 5, the stored table used by DFD1F is a function of object distance u, camera parameters $\mathbf{e_1}$ and $\mathbf{e_2}$. Therefore, the stored table T_s in Eq. (??) is a function of s_1, D_1, f_1, s_2, D_2, f_2, and u. The error sensetivity can be expressed as

$$\frac{\delta T_s}{T_s} = P_{s_1}\frac{\delta s_1}{s_1} + P_{f_1}\frac{\delta f_1}{f_1} + P_{D_1}\frac{\delta D_1}{D_1} + P_{s_2}\frac{\delta s_2}{s_2} + P_{f_2}\frac{\delta f_2}{f_2} + P_{D_2}\frac{\delta D_2}{D_2} + P_u\frac{\delta u}{u} \quad \text{(A.1)}$$

The parameters P_{s_1}, P_{f_1}, P_{D_1}, P_{s_2}, P_{f_2}, P_{D_2}, and P_u are list as follows

Geometric Optics PSF Model:

$$P_{s_1} = \left(\frac{J_0(R_1\rho)}{J_1(R_1\rho)}R_1\rho - 2\right) \Bigg/ \left[s_1\left(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1}\right)\ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right|\right]$$

$$P_{f_1} = \left(-\frac{J_0(R_1\rho)}{J_1(R_1\rho)}R_1\rho + 2\right) \Bigg/ \left[f_1\left(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1}\right)\ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right|\right]$$

$$\begin{aligned}
P_{D_1} &= \left(\frac{J_0(R_1\rho)}{J_1(R_1\rho)}R_1\rho - 2\right) \Big/ \ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right| \\
P_{s_2} &= \left(-\frac{J_0(R_2\rho)}{J_1(R_2\rho)}R_2\rho + 2\right) \Big/ \left[s_2\left(\frac{1}{f_2} - \frac{1}{u} - \frac{1}{s_2}\right)\ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right|\right] \\
P_{f_2} &= \left(\frac{J_0(R_2\rho)}{J_1(R_2\rho)}R_2\rho - 2\right) \Big/ \left[f_2\left(\frac{1}{f_2} - \frac{1}{u} - \frac{1}{s_2}\right)\ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right|\right] \\
P_{D_2} &= \left(-\frac{J_0(R_2\rho)}{J_1(R_2\rho)}R_2\rho + 2\right) \Big/ \ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right| \\
P_u &= \left[\left(\frac{J_0(R_1\rho)}{J_1(R_1\rho)}R_1\rho - 2\right)\left(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1}\right)^{-1} - \left(\frac{J_0(R_2\rho)}{J_1(R_2\rho)}R_2\rho - 2\right)\left(\frac{1}{f_2} - \frac{1}{u} - \frac{1}{s_2}\right)^{-1}\right] \\
&\quad \Big/ \left(u \ln\left|\frac{J_1(R_1\rho)R_2}{J_1(R_2\rho)R_1}\right|\right)
\end{aligned}$$

Gaussian PSF Model:

$$\begin{aligned}
P_{s_1} &= \frac{\partial T_s}{\partial s_1}\frac{s_1}{T_s} = \frac{2\sigma_1^2}{s_1(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1})(\sigma_1^2 - \sigma_2^2)} \\
P_{f_1} &= \frac{\partial T_s}{\partial f_1}\frac{f_1}{T_s} = \frac{2\sigma_1^2}{f_1(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1})(\sigma_2^2 - \sigma_1^2)} \\
P_{D_1} &= \frac{\partial T_s}{\partial D_1}\frac{D_1}{T_s} = \frac{2\sigma_1^2}{\sigma_1^2 - \sigma_2^2} \\
P_{s_2} &= \frac{\partial T_s}{\partial s_2}\frac{s_2}{T_s} = \frac{2\sigma_2^2}{s_2(\frac{1}{f_2} - \frac{1}{u} - \frac{1}{s_2})(\sigma_2^2 - \sigma_1^2)}
\end{aligned}$$

$$P_{f_2} = \frac{\partial T_s}{\partial f_2}\frac{f_2}{T_s} = \frac{2\sigma_2^2}{f_2(\frac{1}{f_2}-\frac{1}{u}-\frac{1}{s_2})(\sigma_1^2-\sigma_2^2)}$$

$$P_{D_2} = \frac{\partial T_s}{\partial D_2}\frac{D_2}{T_s} = \frac{2\sigma_2^2}{\sigma_2^2-\sigma_1^2}$$

$$P_u = \frac{\partial T_s}{\partial u}\frac{u}{T_s} = \frac{2\sigma_2^2/(\frac{1}{f_2}-\frac{1}{u}-\frac{1}{s_2})-2\sigma_1^2/(\frac{1}{f_1}-\frac{1}{u}-\frac{1}{s_1})}{u(\sigma_2^2-\sigma_1^2)}$$

Numerical Example:

For camera settings: focal length $f = 35$ mm, F# = 4 ($D = f/F\# = 8.75$ mm), taking two images at steps 10 and 40, object distance $u = 1.5$ meter. Using the geometric optics PSF model with $\rho = 2\pi$ 0.6. The following quantity can be obtained.

$s_1 =$	35.248 mm	$f_1 =$	35.000 mm	$D_1 =$	8.750 mm
$s_2 =$	35.992 mm	$f_2 =$	35.000 mm	$D_2 =$	8.750 mm
$\rho =$	$0.6 \times 2\ \pi$	$u =$	1500.0 mm		
$R_1 =$	5.52 pixels	$R_2 =$	1.90 pixels		

Using the equations above, the parameters is computed as

$P_{s_1} =$	-131.053	$P_{f_1} =$	131.982	$P_{D_1} =$	2.151
$P_{s_2} =$	-34.719	$P_{f_2} =$	35.703	$P_{D_2} =$	-0.1510
$P_u =$	-3.913				

Suppose there was an error in the lens movement when taking the first image. i.e, lens step was 11 instead of 10. Therefore, s_1 changes from 35.248 mm to 35.2728 mm.

$$\frac{\delta s_1}{s_1} = \frac{35.2728 - 35.248}{35.248} = 0.0007036 \tag{A.2}$$

$$\frac{\delta T_s}{T_s} = P_{s_1} \frac{\delta s_1}{s_1} = -0.09221 \tag{A.3}$$

This error will contribute to $\delta u/u$ (assuming there is no other source of error),

$$\frac{\delta u}{u} = \frac{\delta T_s}{T_s} / P_u = -0.09221/-3.913 = 0.02357 \tag{A.4}$$

A 2.4% error in distance will be introduced by this error in lens movement, or a distance estimation error of 35.4 mm.

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

[3] Born M. and Wolf, E., Principles of Optics, Pergamon Press, Oxford, Sixth Edition (1980).

[4] Cambell, F.W., "Correlation of accommodation between the two eyes", Journal of the Optical Society of America, 50, p. 738 (1960).

[5] Cryer, J.E. Tsai, P.-S. and Shah, M., "Integration of Shape from X Modules: Combining Stereo and Shading", IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition, pp. 720–721, New York City, New York, (June 1993).

[6] Dhond, U.R. and Aggarwal, J.K., "Structure from Stereo – A Review", IEEE Trans. Systems, Man and Cybernetics, Vol 19, No 6, pp. 1489–1509 (Nov./Dec. 1989).

[7] Ens, J. and Lawrence, P., "An Investigation of Methods for Determining Depth from Focus", IEEE Trans. Pattern Analysis and Mach. Intell., PAMI-15, No. 2, pp. 97–108 (Feb. 1993).

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

[9] Grossman, P., "Depth from focus", Pattern Recognition Letters 5, pp. 63–69, (Jan. 1987).

[10] Hoff, W. and Ahuja N., "Surfaces from Stereo: Integrating Feature Matching, Disparity Estimation, and Contour Detection" IEEE Trans. Pattern Analysis and Mach. Intell., Vol 11, No 2, pp. 121-136 (Feb. 1989).

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

[12] Horn, B.K.P., "Focusing", Artificial Intelligence Memo, No 160, MIT (1968).

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

[14] Hougen, D. and Ahuja N., "Integration of Stereo and Shape from Shading Using Color", Proc. 2nd Intl. Conf. Automation, Robotics and Computer Vision, Singapore (Sept. 1992).

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

[16] Jarvis, R.A., “A perspective on range finding techniques for computer vision”, IEEE Trans. Pattern Analysis and Mach. Intell., PAMI-5, No. 2, pp. 122–139 (March 1983).

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

[18] Krotkov, E., “Focusing”, International Journal of Computer Vision, 1, pp. 223-237 (1987).

[19] 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).

[20] Lai, S.H., Fu, C.W. and Chang, S., “A Generalized Depth Estimation Algorithm with a Single Image” IEEE Trans. Pattern Analysis and Mach. Intell., PAMI-14, No. 4, pp. 405–411 (April 1992).

[21] Leu, J.J., Tsai, C.J., Hung, Y.P. and Chen, C.H., “Depth Recovery by Integrating Depth-from-defocus with Stereo”, International Conf. on Automation, Robotics, and Computer Vision, Singapore, pp. CV-6.7.1 - CV-6.7.5 (Sep. 1992).

[22] 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).

[23] Ligthart, G. and Groen, F., “A Comparison of Different Autofocus Algorithms”, International Conference on Pattern Recognition,

pp. 597–600 (1982).

[24] Marr, D. and Poggio, T., “A Computational Theory of Human Stereo Vision”, Proc. of the Royal Society of London, B 204, pp. 301–328 (1979).

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

[26] Moerdler, M.L. and Boult, T.E., “The Integration of Information from Stereo and Multiple Shape-From-Texture Cues”, IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition, pp. 524–529 (1988).

[27] Nayar, S.K., “Shape from Focus System for Rough Surfaces”, IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition, Champaign, Illinois, pp. 302–308 (June 1992).

[28] Okutomi, M. and Kanade, T., “A Multiple-Baseline Stereo”, IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition, (1991).

[29] Pentland, A.P., “Depth of Scene from Depth of Field”, Proc. Image Understanding Workshop, pp. 253–259 (April 1982).

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

[31] Pentland, A.P., “A New Sense for Depth of Field”, IEEE Trans. Pattern Analysis and Mach. Intell., Vol. PAMI-9, No. 4, pp. 523–531 (1987).

[32] 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).

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

[34] Ross, B., "A Pratical Stereo System", IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition, pp. 148–153 (1993).

[35] Schlag, J.F., Sanderson, A.C., Neuman, C.P. and Wimberly, F.C., "Implementation of automatic focusing algorithms for a computer vision system with camera control", CMU-RI-TR-83-14, Robotics Institute, Carnegie-Mellon University (1983).

[36] Schreiber, W.F., Fundamentals of Electronic Imaging Systems, Springer-Verlag, Section 2.5.2. (1986).

[37] Stewart, C.V. and Dyer, C.R., "Local Constraint Integration in a Connectionist Model of Stereo Vision", IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition, pp. 165-170 (1988).

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

[39] Subbarao, M., "Parallel depth recovery by changing camera parameters", Second International Conference on Computer Vision, Florida, USA, pp. 149–155 (Dec. 1988).

[40] 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.

[41] Subbarao, M., “Determining distance from defocused images of simple objects”, Tech. Report No. 89.07.20, Computer Vision Laboratory, Dept. of Electrical Engineering, State University of New York, Stony Brook, NY 11794-2350.

[42] 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).

[43] Subbarao, M., “Spatial-Domain Convolution/Deconvolution Transform ”, Tech. Report No. 91.07.03, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY, Stony Brook (1991).

[44] Subbarao, M. and Choi, T., “Focusing Techniques”, Proceedings SPIE, Boston, Massachusetts, Vol 1823, pp. 163–174, (Nov. 1992).

[45] Subbarao, M. and Surya, G., “Application of Spatial-Domain Convolution/Deconvolution Transform for Determining Distance from Image Defocus”, Proceedings SPIE, Boston, Massachusetts, Vol 1822, pp. 159–167 (Nov. 1992).

[46] 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).

[47] 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 (1993).

[48] Subbarao, M. and Lu, M.-C., "Computer Modeling and Simulation of Camera Defocus", Proceedings SPIE, Boston, Massachusetts, Vol 1822, pp. 110–120 (Nov. 1992).

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

[50] 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).

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

[52] Tenenbaum, J.M., Accommodation in Computer Vision, Ph.D. Dissertation, Stanford University, (Nov. 1970).

[53] Weale, R.A., Focus on Vision, Harvard University Press, Cambridge, Massachusetts (1982).