A Unified Approach to Image Focus and Defocus Analysis

A Dissertation Presented

by

Yen-Fu Liu

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

June 1998

State University of New York
at Stony Brook

The Graduate School

Yen-Fu Liu

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

Murali Subbarao, Advisor, Associate Professor
Department of Electrical Engineering

Petar M. Djurić, Chairman, Associate Professor
Department of Electrical Engineering

Wendy Tang , Associate Professor
Department of Electrical Engineering

Amitabh Varsheny, Assistant Professor
Department of Computer Science

This dissertation is accepted by the Graduate School.

Graduate School

Abstract of the Dissertation

A Unified Approach to Image Focus and Defocus Analysis

by

Yen-Fu Liu

Doctor of Philosophy

in

Electrical Engineering

State University of New York at Stony Brook

1998

Advisor: Dr. Muralidhara Subbarao

Abstract

Recovering the three-dimensional (3D) information lost due to the projection of a 3D scene onto a two-dimensional (2D) image plane is an important research area in computer vision. In this thesis we present a new approach to reconstruct a highly accurate 3D shape and focused image of an object from a sequence of noisy defocused images. This new approach – *Unified Focus and Defocus Analysis* (UFDA) – unifies the two approaches– *Image Focus Analysis* (IFA) and *Image Defocus Analysis* (IDA) – which have been treated separately in the research literature so far. UFDA is based on modeling the sensing of defocused images in a camera system. The concept of a "Three-Dimensional Point Spread Function" (3D PSF) in the (x, y, d) space is introduced, where x and y are the image spatial coordinates and d is a parameter representing the level of defocus. The importance of the choice of this parameterization is that it facilitates the derivation of a 3D convolution equation for image formation under certain weak conditions. The problem of 3D shape and focused image reconstruction is formulated as an optimization problem where the difference (mean-square error) between the observed image data and the estimated image data is minimized by an optimization approach. The estimated image data is obtained from the image sensing model and the current best known solutions to the 3D shape and focused image. Depending on the number of images in the sequence, an initial estimation of solution can be obtained through IFA or IDA methods.

Three optimization techniques have been applied to UFDA – a classical gradient descent approach, a local search method and a regularization technique. Based on these techniques, an efficient computational algorithm has been developed to use variable number of images.

A parallel implementation of UFDA on the Parallel Virtual Machine (PVM) is also investigated. Since one of the most computational intensive part of the UFDA approach is the estimation of image data that would be recorded by a camera for a given solution for 3D shape and focused image . This computational step has to be repeated once during each iteration of the optimization algorithm. Therefore this step has been sped up by using the PVM. PVM speeds up the computation process by utilizing the power of multiple machines on a network. Experimental results show significant speedup is obtained through parallelization.

UFDA shows the trade-off between (i) the amoumt of image data, (ii) computation cost and (iii) accuracy. Experimental results indicate UFDA to be useful in prcatical applications.

To my parents, wife and daughter

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my sincere gratitude to my academic advisor Professor Muralidhara Subbarao, for his intelligent guidance and constant encouragement during the research. I am indebted to Professor Petar M. Djurić, the committee chairman, Professor Wendy Tang and Professor Ambitah Varshney for serving on my thesis committee and for all the valuable advices.

I would like to thank my brothers, friends and colleagues: Yen-Po Liu, Yen-Hong Liu, Hsiang-Tsun Li, Jong-Kae Fwu, Jenn-Kwei Tyan, Ta Yuan, Nai-Wei Lo, Gopal Surya and all my other friends at 202, 204 and 244. I would especially like to thank two of my friends, Hsiang-Tsun Li and Jong-Kae Fwu, for their generous help and encouragement during my stay at Stony Brook. I also thank Mrs. Maria Krause and Mrs. Judy Eimer, for their kindness and help during my graduate studies.

One person that I am indebted to and without him I would not have completed this dissertation is, my advisor, Dr. Muralidhara Subbarao. His patience, intelligence and excellence in research has inspired me to be an better researcher than I was.

My deepest appreication and love go to my parents, Guang-Ruei Liu, Pei-Yu Yu, my beloved wife Hsiao-Lei Su and my daughter In-Hsiaun Liu. I thank them for their sacrifice and patience.

The support of this research in part by the Olympus Optical Cooperation is gratefully acknowledged.

Chapter 1

Introduction

1.1 Background

The 3D scene information that can be sensed by a camera are of two types– *geometric* or depth-map and *photometric* or "brightness". The 3D shapes and distances of objects in the scene constitute the geometric information while the image irradiance constitutes the photometric information. There are many methods to recover these two types of scene information in 3D machine vision. They can be classified into two categories: active methods and passive methods. Active methods project energy onto the object in the scene and detect the reflected energy. Examples of this approach are sonar ranging based on time-of-flight of sound, and laser ranging and structured-light method based on triangulation. Due to the limitation of energy source, these methods are impractical when the conditions of the object and environment exceed certain range. Passive methods, on the other hand, use ambient illumination during data acquisition. Some well-known examples of passive methods are– stereo vision, motion parallax, shape-from-shading, depth-from-focus (image focus analysis or IFA) and depth-from-defocus (image defocus analysis or IDA). Stereo vision takes two views from two different cameras simultaneously and finds correspondence between points in the two images to compute the disparity and then depth map. Motion parallax uses the dynamic properties of objects and relative movement between the camera and environment as cues to determine 3D information. Shape-from-shading is a photometric method to determine surface orientation from image brightness. The image of a three-dimensional object depends on its shape, its reflectance properties, and distribution of light source. Horn[4] showed how to determine shape from shading by modeling the image formation process with the image irradiance equation.

Among these methods, stereo vision is the most popular method. However, the major computational problems associated with stereo are the *correspondence problem* and detection of *occlusion*. Recently, *Image Focus Analysis* (IFA) [19, 67, 5, 31, 38, 48, 54] and *Image Defocus analysis* (IDA) methods [14, 34, 41, 51] have attracted the attention of researchers as they do not suffer from the problems associated with stereo.

IFA is a search method which searches the camera parameters that correspond to focusing the object. Here, a large number of images is needed as input to compute a focus measure (FM) in order to determine the focused image and 3D shape. It needs mechanical motion of camera parts to change camera parameters (e.g. lens position) in order to acquire images and hence slow. Many IFA methods have been proposed in the literature. Subbarao and Tae-Choi [48] proposed a Shape from Image Focus method which is based on finding the best FM on a Focused Image Surface instead of over image frames sensed by a planar image detector. In any IFA method the problem of how to choose the best focus measure has also been investigated. Some examples of focus measures are image energy, energy of Laplacian, and energy of image gradient[49]. Recently, Subbarao and Tyan [52] described two metrics named AUM (*Autofocusing Uncertainty Measure*) and ARMS error (*Autofocusing Root-Mean-Square Error*) for noise sensitivity analysis of focus measures to select the the best one from a set of focus measures.

IDA, unlike IFA, is not a search method. It measures the amount of blur in an image and needs as low as only two images to obtain the depth map of an entire scene. It is faster than IFA but less accurate. In the literature [34, 35] Pentland proposed a method that based on modeling a blurred step edge as the result of convolving a focused image with a Gaussian PSF. He solved for the blur parameter and the height of the edge by a linear regression method to obtain the depth of the edge. He also proposed an algorithm for arbitrary scene using two images. One of the images is a focused image formed by a pin-hole camera, and the other is a defocused image obtained from a wide aperture camera. This image is defocused by different amounts at different positions depending on the distance of object points. He then obtained the 3D information by comparing the corresponding points in these two images and measuring the change in focus. Enns and Lawrence [71] proposed a matrix based regularization method. They used an iterative regularization method to solve the inverse filtering problem for two blurred images to determine depth.

In comparison with other IDA methods, their method is computationally expensive and needs to calibrate the camera PSF.

Subbarao and Wei [68] have proposed a Fourier domain approach, called DFD1F for obtaining depth using defocusing information. It computes six one-dimensional Fourier coefficients from images to get depth map. Another method named STM [51] (*Spatial domain Convolution/Deconvolution Transform Method*) by Subbarao and Suyra uses spatial domain analysis approach. In this method, two images recorded with two different camera parameter settings and a local cubic polynomial model of image brightness are used to obtain depth.

The method of our interest in this research falls into the category of focus and defocus analysis. But unlike those IFA and IDA methods, the method we propose here has the flexibility to deal with different conditions such as the amount of image data, the knowledge of camera characteristics, the degree of accuracy and computation cost in the applications.

In the following section, we will addressed the motivation for developing this method.

1.2 Motivation

The purpose of this research is to find a method that can reconstruct a better 3D shape and focused image in any circumstance compared to IFA and/or IDA. To achieve this goal, the characteristics of IFA and IDA are exploited. The IFA and IDA methods form two extremes of a range of methods useful in 3D shape and focused image recovery. At one end of this range of methods is the IFA method which uses a large amount of image data but minimal information about the camera characteristics (e.g. the camera's point spread function). At the other end is the IDA method which uses minimal image data but much information about the camera characteristics. Therefore, a new theory that unified IFA and IDA is proposed in this work to tackle the problem of 3D scene recovery. This new theory results in a unified approach that suggests new methods that lie between the two extremes of the IFA and IDA methods. The unified approach will be referred to as the *Unified Focus and Defocus Analysis* or UFDA.

In this dissertation, the theory of UFDA, optimization techniques, computational algorithms, together with both sequential and parallel implementation of UFDA are investigated to demonstrated the usefulness of UFDA.

1.3 Dissertation Overview

This dissertation is organized as follows. Chapter 2 provides the concepts of Image Focus Analysis (IFA). Here, the basic principles, the definition of Focused Image Surface (FIS), the advantages and disadvantages of IFA are addressed. Following that, in chapter 3, we introduce Image Defocus Analysis (IDA). In contrast to the IFA methods that involve the search of sharpest focus measure in a large number of images, IDA methods require only a few images for recovering the 3D scene of an object whether the images are focused or not. Two methods of IDA are summarized in this chapter, one is a Spatial

Domain approach using S-Transform, the other is a Fourier Domain method using one-dimensional Fourier coefficient. Theories of these two approaches are briefly described in this chapter.

In chapter 4, the image formation model for the Unified Focus and Defocus Analysis (UFDA) is presented. A Three-dimensional Point Spread Function (3D PSF) is introduced here for the formation of images where the image data is treated as a function sampled in the 3D space (x, y, d). The x and y are image spatial coordinates and d is defined as blur parameter that represents the level of defocus. The importance of this image formation model is the way we parameterized the 3D PSF and the function of the image data in terms of blur parameter d. For this choice of parameterization under certain conditions, we can derive a 3D convolution expression for image formation. In this chapter, both principles and derivations of this image formation model are discussed. This model is later applied for the estimation of image data in the UFDA optimization process. Our experiments show that this model produces very accurate image data. We conclude this chapter with a discussion of two other cases of this image formation model under different conditions which enable us to apply UFDA.

In chapter 5, we propose a unified approach to image focus and defocus analysis (UFDA). IFA and IDA can be viewed as two extremes in a range of methods useful in 3D shape and focused image recovery. UFDA is investigated by first exploiting the characteristics of IFA and IDA and their approaches to the focus and defocus information in the image data. The theory of UFDA is then established by linking IFA and IDA with the number of unknowns and the number of constraints that are embedded in the image data. Therefore, based on the image sensing model in chapter 4, the problem of 3D shape and focused image reconstruction is formulated as an optimization problem where the difference (mean-square error) between the observed image data and estimated image data is minimized iteratively by an optimization technique. Here, the estimated image data is obtained from the image sensing model and the current best known solution to the 3D shape and focused image. An initial solution can be obtained either from an IFA and/or an IDA method. From our experiments, this UFDA approach shown that it reduces the error in shape and focused image introduced by the image-overlap problem and the non-smoothness of the object's 3D shape. The theory and some practical considerations of UFDA are discussed here. This chapter is concluded by the discussion of how UFDA alleviate this image-overlap problem.

Chapter 6 shows the derivation of the 3D PSF based on geometric optics in a single lens camera system. In practice, the point spread function of each camera is different and can be obtained either from the manufacturer or through a calibration procedure. The 3D PSF that we have proposed here is a good approximation to the real camera point spread function in a certain range of blur. Also, unlike two-dimensional shift-invariant type point spread function, this 3D PSF states the response for each object point depending on their distance to the lens.

In chapter 7, three optimization techniques that were applied to the UFDA optimization process are presented. These techniques include a gradient descent approach, a local search method and a regularization method.

This chapter begins with the discussion of gradient descent approach. The gradient descent method is an iterative method that based on going downhill with respect the error function to find the lowest point. In our case, the focused image surface in a certain image region is approximated by a piecewise planar surface patch with three parameters - slope with respect to x-axis, slope with respect to y-axis and z-axis intercept. Three search methods of this gradient descent type have been tried in the experiments. They are a sequential parameter search (SPS), a parallel parameter search (PPS) and a sequential followed by a parallel parameter search (SPPS). The theory and implementation techniques of these methods are presented. Following the gradient descent method, a local search method that is similar to a brute force search method is introduced. This search method searches a limited range of depth around the initial solution of each point of the focused image surface for the object to find a local error minimum in each iteration. This method can obtain very accurate result but is highly computation intensive. So far, both the gradient descent and local search method are applied to minimize the difference between the observed and estimated image data directly. In particular, if a perfect solution is found, then the difference of observed and estimated data should converge to zero. On the other hand, for the third optimization technique, the problem of 3D shape and focused image recovery is formulated as an ill-posed inverse optics problem in the sense of Hadamard and solve through a regularization method. The regularization method that we applied to the UFDA uses minimization the integral of the Laplacian Squared of the blur parameter d as the smoothness constraint for the intended solution. Therefore, the cost functional to be minimized in this approach is a weighted sum of a similarity functional– difference between observed and estimated image data and a stabilizing functional– smoothness constraint with a regularization parameter λ. In addition to the smoothness constraint, three constraints are also used in the error minimization. They are - positivity of image brightness and depth, and a local error control constraint. The first two capture the physical reality and restrict the space of possible solution. The last prevents overshoots and helps in fast converge of the solution during iteration. The theory of this regularization, how to apply the regularization to the UFDA, derivation of Euler-Lagrange equation, implementation techniques and how the value of λ affects the minimization are discussed.

Chapter 8 describes a computational algorithm specifically for the UFDA. The importance of this algorithm is that it deals with variable number of input images in an optimal fashion. This algorithm consists of two parts. First, how to obtain an initial solution that is based on the number of images and available methods. The second part is to solve the deconvolution problem for the focused image in the UFDA optimization process. Details of this algorithm are covered in this chapter and several experiments of UFDA are carried out here based on this algorithm. These experiments are divided into two groups. One group is to recover the 3D shape and focused image with many image frames while the other one uses only a few image frames. The optimization techniques that are described in chapter 7 are applied to these experiments and we conclude this chapter by the performance comparison of these techniques in each group.

In chapter 9, a parallel implementation of UFDA by Parallel Virtual Machine (PVM) is presented. This part of research is a join work by Mr. Naiwei Lo, Dr. Muralidhara Subbarao, Dr. Bradely S. Carlson and author. The motivation of this work is to show that the computation cost of UFDA can be easily reduced by the power of parallel processing. Also, this work demonstrates an application of PVM to the problem of 3D scene recovery. In this chapter, the identify of parallelizable part of UFDA, what is PVM, implementation techniques, experimental configurations are described. The performance of different configuration of network machine are also analyzed here.

Finally, in chapter 10, we conclude this dissertation by a summary of our research and possible extension of this work for future investigation.

Chapter 2

Image Focus Analysis

2.1 Image Focus Analysis (IFA)

In IFA, a large sequence of image frames of a 3D scene is recorded with different camera parameters (e.g. focal length or/and lens to image detector distance). In each image frame, different objects in the scene will be blurred by different degrees depending on their distance from the camera lens. Each object will be in best focus in only one image frame in the image sequence. The entire image sequence is processed to find the best focused image of each object in the 3D scene. The distance of each object in the scene is then found from the camera parameters that correspond to the image frame that contains the best focused image of the object. The IFA methods are based on the fact that for an aberration-free convex lens, (i) the radiance at a point in the scene is proportional to the irradiance at its *focused image* [4] (photometric constraint), and (ii) the position of the point in the scene and the position of

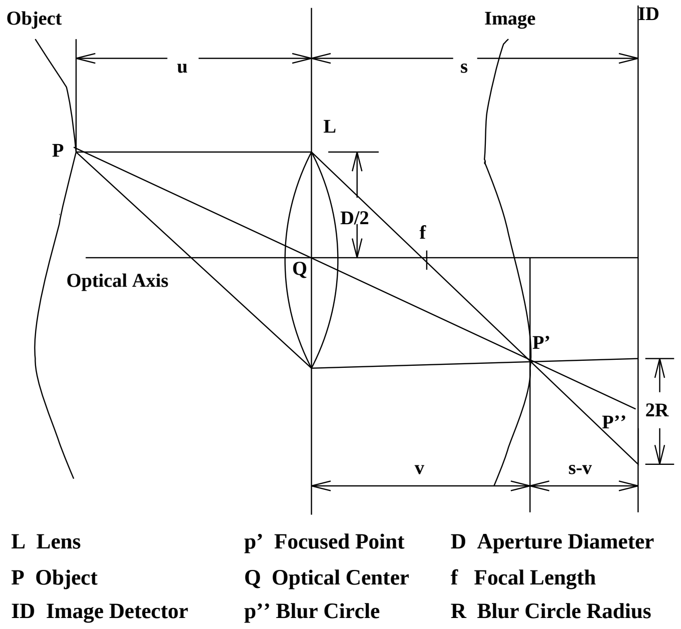

Figure 2.1: Image formation in a convex lens

its focused image are related by the *lens formula* (geometric constraint)

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

where f is the focal length, u is the distance of the object from the lens plane, and v is the distance of the focused image from the lens plane (see Fig. 2.1). Given the irradiance and the position of the focused image of a point, its radiance and position in the scene are uniquely determined. In a sense, the positions of a point-object and its image are *interchangeable,* i.e. the image of the image is the object itself. Now, if we think of an object surface in front of the lens to be comprised of a set of points, then the focused images of these points define another surface behind the lens (see Fig. 2.1). This surface is defined to be the *Focused Image Surface* (FIS) and the image irradiance on

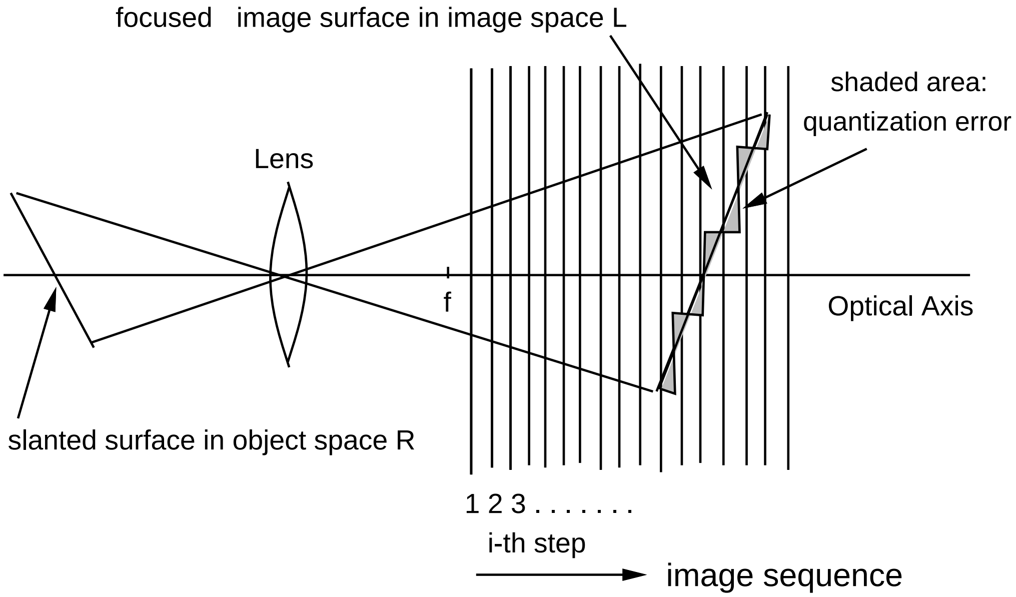

Figure 2.2: Focused Image Surface

this surface to be the *focused image.* There is a *one to one correspondence* between the FIS and the object surface. The geometry (i.e. the 3D shape information) and the radiance distribution (i.e. the photometric information) of the object surface are uniquely determined by the FIS and the focused image. In traditional IFA methods (e.g. [5, 30, 48, 49]) a sequence of images are obtained by continuously varying the distance s between the lens and the image detector or/and the focal length f (see Fig. 2.2 and Fig. 2.3).

For each image in the sequence, a focus measure is computed at each pixel (i.e. each direction of view) in a small (about 15×15) image neighborhood around the pixel. At each pixel, that image frame among the image sequence which has the maximum focus measure is found by a search procedure. The grey level (which is proportional to image irradiance) of the pixel in the image

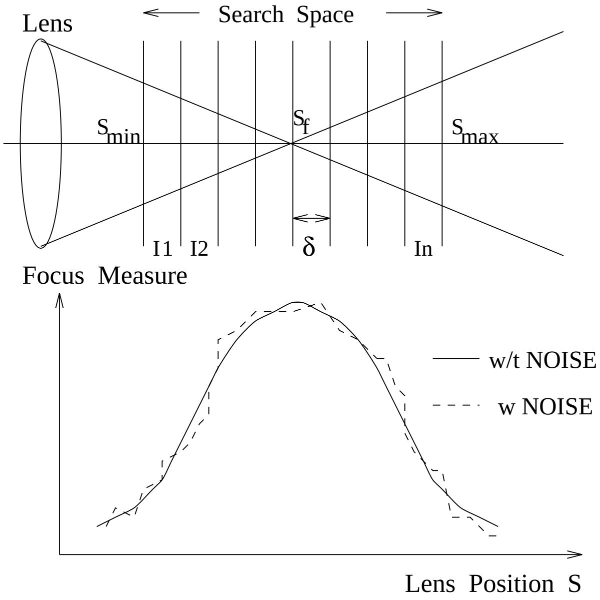

Figure 2.3: Image Focus Analysis

frame thus found gives the grey level of the focused image for that pixel. The values of s and f for this image frame are used to compute the distance of the object point corresponding to the pixel. An example of a focus measure is the grey level variance. IFA methods involve a search for the values of s or/and f that results in a maximum focus measure and these methods require the acquisition and processing of a large number of images. The disadvantage of acquiring a large number of images is the amount of time required to adjust the camera parameters (lens position s or/and focal length f) before recording each image. This involves mechanical motion of camera parts which is often slower than electronic computation. During the entire period of adjusting camera parameters, the scene must remain stationary.

Chapter 3

Image Defocus Analysis

3.1 Introduction

Recently some researchers [71, 34, 68, 41, 51] have proposed methods for finding distance and focused image of an object which do not require focusing the object.(see Fig. 3.1) They take the level of defocus of the object and the corresponding camera parameter values into account in determining distance and focused image. Therefore this approach is called *Image Defocus Analysis* (IDA). IDA methods do not involve searching for f and s values which correspond to focusing the object. Therefore these methods require processing only a few images (about 2-3) as compared to a large number of images in the IFA methods. In addition, only a few images are sufficient to determine the distance of all objects in a scene using the IDA methods, irrespective of whether the objects are focused or not. The two main disadvantages of the IDA methods are (i) they require accurate camera calibration for the cam-

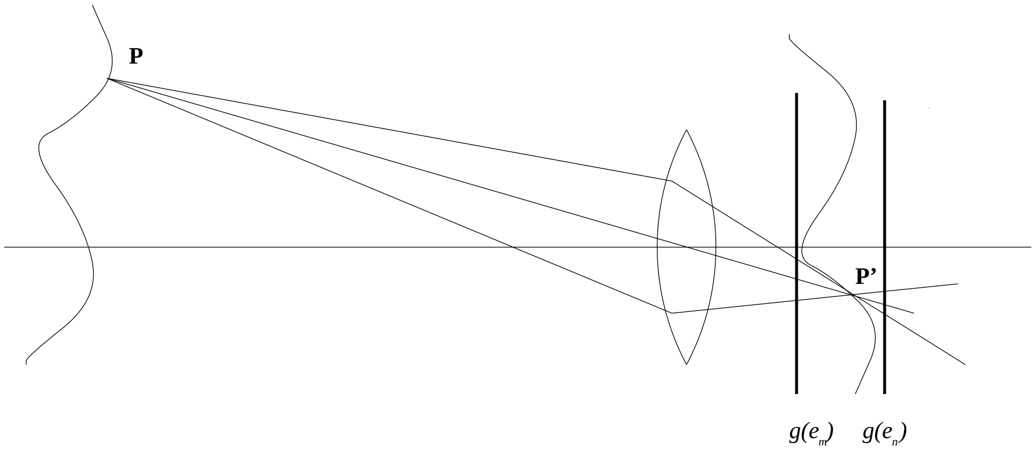

Figure 3.1: Image Defocus Analysis

era characteristics (point spread function as a function of different camera parameters), and (ii) they are less accurate than IFA methods. Here we summarize two main approaches – Fourier Domain [56] and Spatial Domain [53] approaches to image defocus analysis in the following sections.

3.2 Fourier Domain Approach

The image formation in a camera with variable camera parameters (s, f, D) is shown in Fig. 2.1 Here s specifies the lens position, f the focal length, and D the aperture diameter. The camera parameter setting is denoted by $\mathbf{e_i}$ where $\mathbf{e_i} = (s_i, f_i, D_i)$ is a vector representing the $i - th$ camera parameter setting . According to paraxial geometric optics [13] the normalized radius of the blur circle is

$$R = \frac{D}{2}\left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right) \tag{3.1}$$

where u is the object distance. For the sake of conceptual simplicity, let the Point Spread Function (PSF) of the camera be a two-dimensional Gaussian:

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{3.2}$$

where the spread parameter σ is proportional to the blur circle radius R. Therefore we can write

$$\sigma = cR \tag{3.3}$$

where c is a camera constant. In this camera model, a blurred image g of a planar object at distance u having focused image f is given by the convolution of the PSF h and the focused image f, i.e. $g = h * f$. In the Fourier domain, this relation becomes $G(\omega,\nu) = H(\omega,\nu)F(\omega,\nu)$ where $H(\omega,\nu) = e^{-\frac{1}{2}(\omega^2+\nu^2)\sigma^2}$. For two blurred pictures g_1, g_2, taken with two different camera settings $\mathbf{e}_1, \mathbf{e}_2$, we obtain $G_1(\omega,\nu)/G_2(\omega,\nu) = e^{-\frac{1}{2}(\omega^2+\nu^2)(\sigma_1^2-\sigma_2^2)}$ or,

$$\sigma_1^2 - \sigma_2^2 = \frac{-2}{\omega^2+\nu^2}\ln\frac{|G_1(\omega,\nu)|}{|G_2(\omega,\nu)|} \tag{3.4}$$

Further, from Eqs. (3.1), (3.3) we have

$$\sigma_1 = c\frac{D_1}{2}\left(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1}\right) \quad , \quad \sigma_2 = c\frac{D_2}{2}\left(\frac{1}{f_2} - \frac{1}{u} - \frac{1}{s_2}\right) \quad , \tag{3.5}$$

Eliminating $1/u$ from the above two relations we obtain

$$\sigma_1 = \alpha\sigma_2 + \beta \text{ where } \alpha = \frac{D_1}{D_2} \quad , \quad \beta = cD_1\left(\frac{1}{f_1} - \frac{1}{f_2} + \frac{1}{s_2} - \frac{1}{s_1}\right) \tag{3.6}$$

Eqs. (3.4), (3.6) together yield

$$(\alpha^2-1)\sigma_2^2 + 2\alpha\beta\sigma_2 + \beta^2 = \frac{-2}{\omega^2+\nu^2}\ln\frac{|G_1(\omega,\nu)|}{|G_2(\omega,\nu)|} \tag{3.7}$$

In the above equation σ_2 is the only unknown. The equation is quadratic and therefore σ_2 is easily obtained by solving it. The two solutions result in a two-fold ambiguity. Methods for resolving this ambiguity are discussed in [68, 51, 53, 56]. From the solution for σ_2, the distance u of the object is obtained from Eq. (3.5).

The above discussion illustrates the conceptual feasibility of determining distance from two defocused images. Repeating the above procedure in all image neighborhoods, the depth-map of an entire scene can be obtained from only two blurred images in parallel. The Gaussian PSF model results in closed-form solution. However, in general, a closed-form solution cannot be obtained for the actual PSF of a camera. A numerical method will have to be used. Details for the case of arbitrary PSFs are presented in [56].

3.3 Spatial Domain Approach

A new spatial-domain convolution/deconvolution transform (S Transform) is defined in [43]. The definition of the transform for the general case is quite complicated. For two-dimensional images, it is even more involved. However, a special case of the transform suffices for image defocus analysis. This case turns out to be particularly simple.

For two-dimensional images, under a local cubic polynomial model, the transform is defined as follows. If a focused image f is blurred by convolution with a circularly symmetric PSF h to result in the blurred image g, then g is the *forward S transform* of f with respect to the kernel function h, and it is given by:

$$g(x,y) = f(x,y) + \frac{h_2}{2} \bigtriangledown^2 f(x,y) \tag{3.8}$$

where h_2 is the second moment of h with respect to x or y, i.e.

$$h_2 \; = \; \int\!\!\int x^2 \; h(x,y) \; dx \; dy \; = \; \int\!\!\int y^2 \; h(x,y) \; dx \; dy \tag{3.9}$$

and $\bigtriangledown^2$ is the *Laplacian* operator.

The *inverse S transform* of g with respect to the moment vector $(1, h_2)$ is equal to f and it is defined as

$$f(x,y) \; = \; g(x,y) - \frac{h_2}{2} \bigtriangledown^2 g(x,y) \tag{3.10}$$

For a Gaussian PSF model used in the previous subsection, it can be shown that $h_2 = \sigma^2/2$. Therefore, if two blurred images g_1 and g_2 are acquired with different camera settings $\mathbf{e}_1$ and $\mathbf{e}_2$ corresponding to blur parameters σ_1 and σ_2 we obtain

$$f \; = \; g_1 - \frac{\sigma_1^2}{4} \bigtriangledown^2 g_1 \quad , \quad f \; = \; g_2 - \frac{\sigma_2^2}{4} \bigtriangledown^2 g_2 \tag{3.11}$$

From the above two relations, Eq. (3.6), and the fact that $\bigtriangledown^2 g_1 = \bigtriangledown^2 g_2$ we obtain

$$(\alpha^2 - 1)\sigma_2^2 + 2\alpha\beta\sigma_2 + \beta^2 = \frac{4(g_1 - g_2)}{\bigtriangledown^2 g_1} \tag{3.12}$$

where α and β are as defined in the previous section. Except for the right hand side, the above relation is similar to Eq. (3.7). Therefore we have a quadratic equation in σ_2 which can be easily solved. The distance u can be obtained from σ_2 from Eq. (3.5). Implementation details and experimental results for this approach are described in [51, 53].

Chapter 4

Image Formation Model

Consider the image formation in Fig. 2.1. The image irradiance or brightness distribution produced on the image detector plane by a point light source depends on its position in the object space and the camera parameters $\mathbf{e}$. The position of the point source can be specified in terms of a distance u and a direction (x, y) where u is the distance of the point light source from the lens along the optical axis and (x, y) is the point on the image detector where the line passing through the point source and the optical center intersects the image detector plane. Let $h(x, y, x', y', \mathbf{e}, u)$ represent the image brightness at a point (x', y') on the image detector plane produced by a point light source of unit brightness whose position is specified by distance u and the direction (x, y), and when the camera parameter setting is $\mathbf{e}$. The term unit brightness is defined such that

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h(x, y, x', y', \mathbf{e}, u)\; dx'\; dy' \;=\; 1. \tag{4.1}$$

The function h is indeed the PSF of the camera system. Let a 3D scene consist of only opaque objects so that along each direction of view (x, y) there is only one visible object point at a distance $u(x,y)$ with a brightness of $F(x,y)$ units. In this case, the visible surfaces can be considered to be made up of a collection of point light sources lying on a surface defined by $u(x,y)$ and the image brightness $g(x',y')$ observed at a point (x',y') on the image detector is given by the sum of the image irradiances produced by each of the point sources in the scene. Therefore we have

$$g(x',y') \; = \; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x,y)\; h(x,y,x',y',\mathbf{e},u(x,y))\; dx\; dy. \qquad (4.2)$$

For the sake of computational efficiency we will specify the dependence of the PSF h in terms of two new scalar parameters d' and d instead of the vector parameter $\mathbf{e}=(D,f,s)$ and $u(x,y)$. d' and d are defined as the *blur parameter* of point light sources at distances u' and u respectively where u' is a known reference distance. The *blur parameter* d is defined as the normalized blur circle diameter of a point source at distance u. In Fig. 2.1, the diameter of the blur circle can be shown to be

$$d'' \; = \; Ds\left(\frac{1}{f}-\frac{1}{s}-\frac{1}{u}\right) \qquad (4.3)$$

The image magnification changes with s. In order to simplify our analysis and the computational algorithms, all images recorded with different s are normalized to have a fixed magnification. For example, an image recorded with the image detector at s can be scaled to unit magnification by reducing the dimensions of the image by a factor of s. The blur circle diameter in a magnification normalized image is defined as the *blur parameter* and is given by

$$d \; = \; D\left(\frac{1}{f}-\frac{1}{s}-\frac{1}{u}\right) \qquad (4.4)$$

Similarly, d' is defined as

$$d' \;=\; D\left(\frac{1}{f}-\frac{1}{s}-\frac{1}{u'}\right) \tag{4.5}$$

Note that

$$d \;=\; d'+D\left(\frac{1}{u'}-\frac{1}{u}\right) \tag{4.6}$$

d' will essentially replace the vector parameter $\mathbf{e}$ thus reducing the dimensionality of the PSF by two (since the 3 component vector $\mathbf{e}$ is replaced by the scalar d'). The reference distance u' can be arbitrary, but it is convenient to choose it as

$$\frac{1}{u'} \;=\; \frac{1}{2}\left(\frac{1}{u_{min}}+\frac{1}{u_{max}}\right) \tag{4.7}$$

where u_{min} and u_{max} are the minimum and maximum distances at which objects may be present. Since d' depends on $\mathbf{e}$ and d depends on both $\mathbf{e}$ and $u(x,y)$, we may denote them as $d'(\mathbf{e})$ and $d(\mathbf{e},u(x,y))$. Now, if a sequence of images $g_i(x',y')$ are recorded with different camera parameter settings $\mathbf{e}_i$ for $i=1,2,3,\cdots$, then Eq. (4.2) may be written as

$$g_i(x',y') \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x,y)\; h(x,y,x',y',d(\mathbf{e_i},u(x,y)),d'(\mathbf{e_i}))\; dx\; dy. \tag{4.8}$$

In changing $\mathbf{e_i} = (D_i, f_i, s_i)$ from frame to frame, one may change any one parameter at a time, any two at a time, or all three simultaneously. The special cases where only one or two parameters are varied at a time while keeping the other parameters constant are of interest in practical applications. Consider if changing the camera parameter settings involves changing f, or s, or both s and f, but does not involve changing D. In this case, the image $g_i(x',y')$ are arranged so that d' is monotonic as i increases. This image sequence can then be denoted by $g(x',y',d')$ instead of $g_i(x',y')$ and we obtain

$$g(x',y',d') \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x,y)\; h(x,y,x',y',d,d')\; dx\; dy. \qquad (4.9)$$

If $\frac{1}{f}$ is changed by δ, or $\frac{1}{s}$ is changed by δ, or if both $\frac{1}{f}$ and $\frac{1}{s}$ are changed so that the change in $\frac{1}{f}-\frac{1}{s}$ is δ, then both d and d' change by the same amount given by $D\delta$. For this reason we can write

$$g(x',y',d') \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x,y)\; h(x,x',y,y',d'-d)\; dx\; dy. \qquad (4.10)$$

In the case of paraxial optics model of image formation, the above expression can be simplified to the following three-dimensional convolution expression:

$$g(x',y',d') \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F'(x,y,d)\; h(x'-x,y'-y,d'-d)\; dx\; dy\; dd. \quad (4.11)$$

where

$$F'(x,y,d) = \begin{cases} F(x,y) & \text{if } d(\mathbf{e},u(x,y)) = 0 \text{ or } d' = -D\left(\frac{1}{u'}-\frac{1}{u}\right) \text{ and} \\ & \\ 0 & \text{otherwise.} \end{cases} \qquad (4.12)$$

The above convolution expression can be abbreviated as

$$g(x',y',d') \;=\; F'(x',y',d') \;\star\; h(x',y',d') \qquad (4.13)$$

where $\star$ denotes the convolution operator. It will be found that our choice of parameterization of the PSF in terms of the normalized blur circle diameter facilitates the derivation of the convolution expression. Other choices (that are not linearly related to the blur circle diameter) will not lead to the derivation of the convolution expression. In particular, parameterising in terms of s or f directly will not lead to the convolution expression.

In deriving the convolution expression above, we have assumed that D is not changed in order to change $\mathbf{e}$. However, if D is changed, the above result will not hold because the value of $d'-d$ is not a constant with respect to object point locations $(x, y, u(x, y))$ but varies depending on $u(x, y)$ since

$$d' - d \;=\; -D\left(\frac{1}{u'} - \frac{1}{u}\right) \tag{4.14}$$

In this case we obtain

$$g(x', y', d') \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F'(x, y, d)\; h(x'-x, y'-y, d', d)\; dx\; dy\; dd \tag{4.15}$$

where $F'(x, y, d)$ is as defined earlier.

In the (x', y', d') space or the $(x', y', \mathbf{e}, u)$ space, the condition d($\mathbf{e}$,u(x',y'))=0 or

$$D\left(\frac{1}{f} - \frac{1}{s} - \frac{1}{u}\right) \;=\; 0 \tag{4.16}$$

specifies a hyper-surface that corresponds to the Focused Image Surface (FIS) defined earlier. The brightness distribution on this surface is the focused image.

In microscopy, images are acquired by moving the object specimen away from the lens. In this case, the camera parameters $\mathbf{e}$ is not changed, but $u(x', y')$ is changed in equal increments. In this case we can still use the (x', y', d') space where d' is changed by changing u' in equal but known values. Relative motion of the object with respect to the lens along the optical axis shifts the image position of points on the object by unknown amounts. This

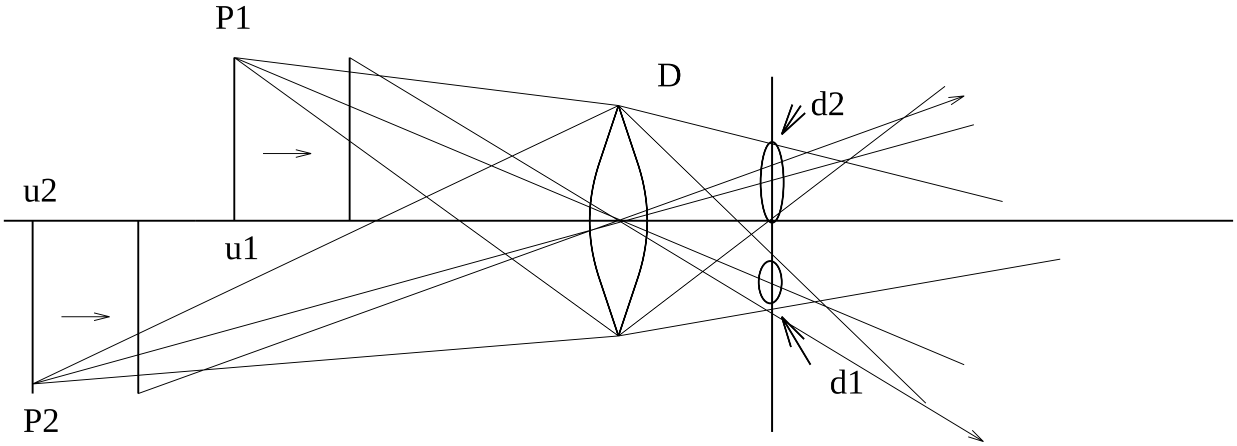

Figure 4.1: Correspondence problem in microscopy

gives rise to the *correspondence problem* encountered in the analysis of *optical flow* in 3D computer vision (see Fig. 4.1.)

But in microscopes, the object will be close to the distance of focal length and focal length is very small. Also object motion (microscope stage) will be small. Under such conditions we may be able to ignore the problem of correspondence. However, since the 3D PSF changes with object distance, we cannot derive a convolution relationship. Suppose the change in object movement is δ, then

$$d \;=\; D\left(\frac{1}{f} - \frac{1}{s} - \frac{1}{\delta + u(x,y)}\right) \tag{4.17}$$

Now $d' - d$ once again changes from point to point depending on $u(x,y)$ and therefore only Eq. (4.15) remains applicable but not the convolution expression.

Recording a sequence of images by changing the camera parameters (or the object distance as in microscopy) should be treated as sampling the function $g(x', y', d')$. The sampling period δ along the d' dimension should be roughly the same as in the (x', y') dimension (i.e. one pixel). Therefore in acquiring a sequence of images, $\mathbf{e}$ should be changed from one image frame to the other (resulting in a sampling period of δ) such that d' changes by about 1 pixel. The sampling period could be larger than this (δ), but choosing it smaller would increase redundancy between two successive image frames.

The analysis in this section dealt with scenes with opaque surfaces. However, this analysis could be extended to 3D transparent objects as in confocal microscopy. In this case we can use 3D Fourier analysis on the 3D convolution expression Eq. (4.11). In comparison with the previous literature [21] our use of (x', y', d') space is new compared to other camera parameters in place of d'.

The analysis in this section in terms of continuous functions can be easily extended to the case of discrete sampled functions. The results in this section are directly applied to the discrete domain in the following sections.

Chapter 5

Unified Focus and Defocus Analysis (UFDA)

5.1 Introduction

The image focus and defocus analysis problem is briefly described in chapter 2 and 3. We discuss this further to lay the foundation for UFDA. First, for simplicity, consider a particular case of the IFA problem where the distance s between the lens and the image detector is changed for focusing (see Figs. 2.1, 2.3, and 6.1). But the analysis here can be extended to the general case where both s and f are varied for focusing. A sequence of images $g_i(j,k)$ are recorded (see Figs. 2.2 and 5.1) with $s = s_i$ for $i = 0, 1, 2, \cdots, I-1$, $j = 0, 1, 2, \cdots, J-1$, $k = 0, 1, 2, \cdots, K-1$, $s_0 = s_{min} < s_1 < s_2 < \cdots s_{I-1}$. Usually, $s_{min} = f$. J and K are the number of rows and columns respectively in each image frame and I is the number of image frames (see Fig. 2.2 and Fig. 5.1). We can think of this image sequence as sampled data of an image volume and denote the image data as $g(i,j,k)$ (Fig. 5.1). In this image volume, the problem is to find the set of pixels which lie on the focused image surface (FIS) of the object. For

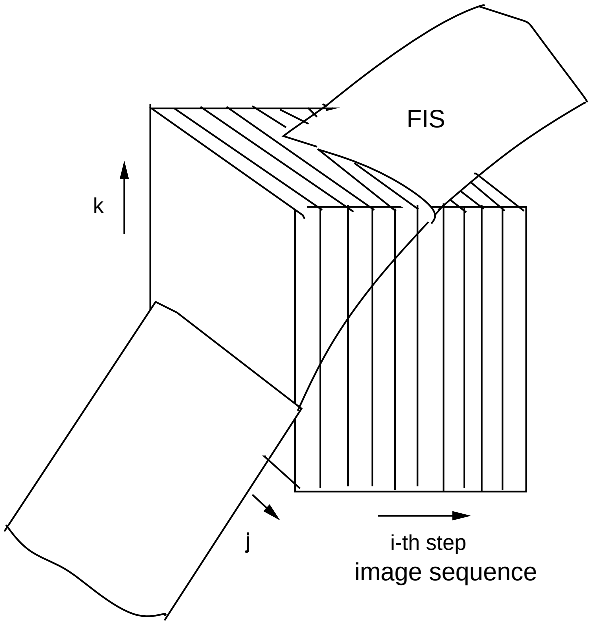

Figure 5.1: Image Volume

surfaces with a slope of up to δ/d where δ is the smallest distance between two image frames and d is the pixel size (Fig. 2.3), for any given row j and column k, there is only one pixel which lies on the FIS. If the slope exceeds this limit, then there will be more than one pixel through which the FIS passes for a given row j and column k. This problem can be avoided by either increasing δ or decreasing d. We shall assume that the slope of FIS is less than or equal to δ/d.

The image frame number i_v to which this pixel belongs depends on (j,k) and therefore it can be expressed as a function $i_v(j,k)$. This function $i_v(j,k)$ represents the FIS and the grey level on the FIS given by

$$F(j,k) = g(i_v(j,k), j, k) \tag{5.1}$$

represents the focused image. Now the image focus analysis problem can be stated as– given the image sequence $g_i(j,k)$ and the camera parameters s_i and f, find $i_v(j,k)$ and $F(j,k)$ in some image region or over the entire image. In order to find the function $i_v(j,k)$ which specifies the FIS, we use the fact that the focus measure computed over $F(j,k)$ (or $g(i_v(j,k),j,k)$) is a maximum over all possible functions i_v. A search for the function i_v is computationally expensive, and typically the FIS is assumed to be a smooth surface in solving for i_v. In the autofocusing problem, only some of the images in the image sequence are used to find the lens position s_i that results in focusing the image in a narrow region. The object imaged in the narrow image region is assumed to be approximately planar and normal to the optical axis. A detailed treatment of the IFA methods can be found in [49, 48]. A comparison of some of the features of IFA and IDA methods is useful in relating them to the unified approach above. The IDA or image defocus analysis problem can be stated as– given two of the images $g_m(j,k)$ and $g_n(j,k)$ in the image sequence $g_i(j,k)$, the camera parameters $\mathbf{e}_m$ and $\mathbf{e}_n$ corresponding to these two images, and the the camera's point spread function as a function of the camera parameters, find $i_v(j,k)$ and $F(j,k)$ in some image region or over the entire image. The values of $i_v(j,k)$ and $F(j,k)$ are both unknown at each of the $J \times K$ pixels. Therefore the total number of unknowns is $2JK$. Given the two images $g_m(j,k)$ and $g_n(j,k)$, we can write Eq. (5.3) – (which is a discrete version of Eq. (4.11).) at each of the $J \times K$ pixels in the two images. This results in $2JK$ constraints, i.e. the solution for $i_v(j,k)$ and $F(j,k)$ must be consistent with the observed data $g_m(j,k)$ and $g_n(j,k)$. Therefore (ignoring border effects due to blurring) the number of unknowns and the number of constraints match. It is known that the spatial domain method of image defocus analysis STM [51] can be used to solve for both $i_v(j,k)$ and $F(j,k)$ at each pixel (under some weak assumptions). If the camera aperture diameter remains the same while recording $g_m(j,k)$ and $g_n(j,k)$, then the solutions for $i_v(j,k)$ and $F(j,k)$ are unique. Otherwise a two-fold ambiguity may result which can be resolved by recording a third image $g_p(j,k)$ with different camera parameter settings. Theoretically only two or three images are sufficient to solve for $i_v(j,k)$ and $F(j,k)$ under some weak assumptions. However, as is clear from the experiments reported in [51], due to noise and lack of sufficient contrast information in very small image regions, a few (about 1 or 2) image frames more than the theoretical minimum would be needed in practical applications. Further $i_v(j,k)$ may need to be smooth or piecewise constant in small image regions depending on image contrast.

In summary, IDA methods use the minimum amount of image data (2 or 3 image frames) needed, but require full information about the camera point spread function $h(\mathbf{e}_i, i, j, k)$. In contrast with IDA methods, IFA methods require a large number of image frames to be recorded and processed, but they need little information about the camera point spread function. The IFA methods do not need calibration of the camera system to determine the PSF (h) as a function of the camera parameters ($\mathbf{e}_i$) but the IDA methods need the calibration. The computational algorithm in the IFA methods is simple relative to that in the IDA methods and the accuracy is better (by a factor of 2 in the experiments reported in [51]).

Let the image volume data $g(i,j,k)$ with I image frames be used in an IFA method. Using the same data, in the unified approach we can write Eq.

(5.3) at each of the $I \times J \times K$ pixels. This is an over constrained system (for $I > 2$). However, in the traditional IFA methods, information about the PSF h is not used. Therefore the image data yields just about sufficient constraints to obtain an approximate solution to $i_v(j,k)$ and $F(j,k)$.

5.2 UFDA

The unknowns in both the IFA and IDA methods are $i_v(j,k)$ and $F(j,k)$. If the image frame size is J rows and K columns, then the number of unknowns for $i_v(j,k)$ and $F(j,k)$ are both JK. Therefore the total number of unknowns is $2JK$. In order to determine these unknowns, a number of image frames $g_i(j,k)$ are recorded at different camera parameter settings $\mathbf{e}_i = (s_i, f_i, D_i)$. The image frame data can be represented as 3D image volume data $g(i,j,k)$ where i is the frame number (Fig. 5.1). This recorded image volume data depends on the camera parameters $\mathbf{e}_i$, the FIS $i_v(j,k)$, and the focused image $F(j,k)$. This dependence is specified by the camera characteristics (e.g. point spread function) [47]. In order to illustrate the concepts, the paraxial geometric optics model of image formation will be used. Under this model, we introduce the concept of a 3D point spread function $h(\mathbf{e}_i, i, j, k)$ defined as the image volume data $g(i,j,k)$ recorded by the camera with camera parameter setting of $\mathbf{e}_i$ when the scene in front of the camera contains only a single point light source such that when it is perfectly focused onto a single pixel, the brightness recorded by the pixel is 1 unit (grey level). In this case, for an arbitrary scene with focused image $F(j,k)$ and FIS $i_v(j,k)$, the observed volume image data

$g(i,j,k)$ can be expressed as

$$g(i,j,k) = \sum_m \sum_n F(m,n)\ h(\mathbf{e}_i, i, j, k, m, n) \tag{5.2}$$

Under certain conditions (see chapter 4), $g(i,j,k)$ above can be approximated by $g'(i,j,k)$ where

$$g'(i,j,k) = \sum_m \sum_n F(m,n)\ h(\mathbf{e}_i, i - i_v(m,n), j - m, k - n) \tag{5.3}$$

Then the problem of 3D shape and focused image reconstruction is formulated as an optimization problem where the difference or mean-square error E between the observed image data $g(i,j,k)$ and the estimated image data $g'(i,j,k)$ is minimized where

$$E = \sum_i \sum_j \sum_k (g(i,j,k) - g'(i,j,k))^2 \tag{5.4}$$

The estimated image data $g'(i,j,k)$ is obtained from Eq. (5.3) using the PSF h and the current best known solutions to the 3D shape $i_v(j,k)$ and focused image $F(j,k)$. An initial estimation of the solutions is obtained through traditional IDA and IFA methods. This solution is improved iteratively by an optimization technique.

This approach reduces the errors in shape and focused image introduced by the image-overlap problem and the non-smoothness of the object's 3D shape. Efficient computational techniques for obtaining a unique solution for $i_v(j,k)$ and $F(j,k)$ need to be developed. For example, a smoothness constraint may be added (as in the regularization technique) to the error term as

$$E = \sum_i \sum_j \sum_k (g(i,j,k) - g'(i,j,k))^2 + \lambda(\bigtriangledown^2 i_v(j,k))^2 \tag{5.5}$$

where λ is a weighting parameter.

The above approach unifies the traditional IDA and IFA techniques. This becomes clear from the previous discussion. Since in terms of the amount of image data used, and the amount of information used about the camera PSF h, the IFA and IDA methods may be viewed as two extremes in a range of possible methods. One can devise methods that are "in between" these two extremes in the sense that they use less amount of image data than the IFA methods but more than the IDA methods, and possibly more information about the camera PSF than the IFA methods but less than that of the IDA methods. One can also introduce the camera parameters $\mathbf{e}_i$ as additional unknowns into the problem.

In practical applications, using only 2 or 3 images in the IDA methods do not yield reliable results in the presence of noise. In this case, one might like to obtain reliable results using more image data, but not as much data as in traditional IFA methods. In this case, there is a need to device a method that is in a sense a combination of IDA and IFA methods. It is necessary to avoid acquiring and processing of excessive number of image frames. There is need for an algorithm that helps in guiding the acquisition of the most useful image data while avoiding image data that would contribute only marginally to improving the accuracy of estimated FIS and focused image. In a sense, the solution to $i_v(j,k)$ and $F(j,k)$ should progressively improve in accuracy with the acquisition and processing of additional image data.

One algorithm that incorporates these features is outlined here. Suppose that a 3D scene contains many objects of different shapes at different distances.

First a few image frames (about 3 to 5) placed far apart in the image volume data are acquired. IDA methods are applied using these images to get a rough estimate of $i_v(j,k)$. Then additional image frames are recorded only in regions surrounding the values taken by estimated $i_v(j,k)$. No image frames need to be recorded outside the range of estimated $i_v(j,k)$ because the objects would be more blurred in those image frames than in the image frames recorded near the estimated values of $i_v(j,k)$. More blurring increases border effects and the image-overlap problem both of which lead to larger errors in the estimation of $i_v(j,k)$. Then the image focus analysis and interpolation are used to obtain improved estimate of $i_v(j,k)$. This solution is further refined using the unified approach described earlier. One can also use the unified approach directly on as few as only two images (as in traditional IDA) methods, but, in the absence of good starting solutions, the method will become computationally intensive. If more than two images are given, then one may use the unified approach to minimize the effects of noise by solving an over-constrained system or introduce additional unknowns in the form of camera parameters.

Blurring has no effect on uniformly bright image regions and it's effect increases with rapid variations in image brightness. Therefore, in the presence of noise, image regions with slowly varying brightness provide less reliable information than those regions with rapidly varying (e.g. near edges) brightness. Therefore, one may compute a weighted least square error where the weight at a point can be a function that increases with increasing image gradient. Solutions for the FIS and the focused images may be obtained by minimizing this weighted mean-square error.

One can apply derivatives (e.g Laplacian) or other linear shift invariant (LSI) filters (e.g. Laplacian of Gaussian) on either side of Eqs. (5.2,5.3). Then we may solve for the FIS by minimizing the least square error between the derivative (or filtered) observed data and the predicted data.

5.3 Image Overlap Problem

It is pointed out earlier that UFDA reduces the error caused by image-overlap problem. In this section, we explain the image-overlap problem and how UFDA deals with it. Image-overlap occurs when two or more near-by point sources in the scene are defocused on the image detector plane and their blurred images overlap (Fig. 5.2). In particular, the brightness near the border of an image region is affected by the defocused objects in the surrounding image regions. This phenomenon introduces error in 3D shape recovery using IFA and IDA methods when an image is divided into small regions and each region is processed independently. This error may be reduced by selective illumination or by multiplying the image intensity by center-weighted masks, but these methods are not always satisfactory. In UFDA the image sensing model takes into account the contribution from all object point sources in estimating the observed image brightness at a given point. In particular, the image brightness near the border of an image region is computed by taking into account all possible object point sources– both those inside the image region and those outside but close to the border. Therefore the error in 3D shape

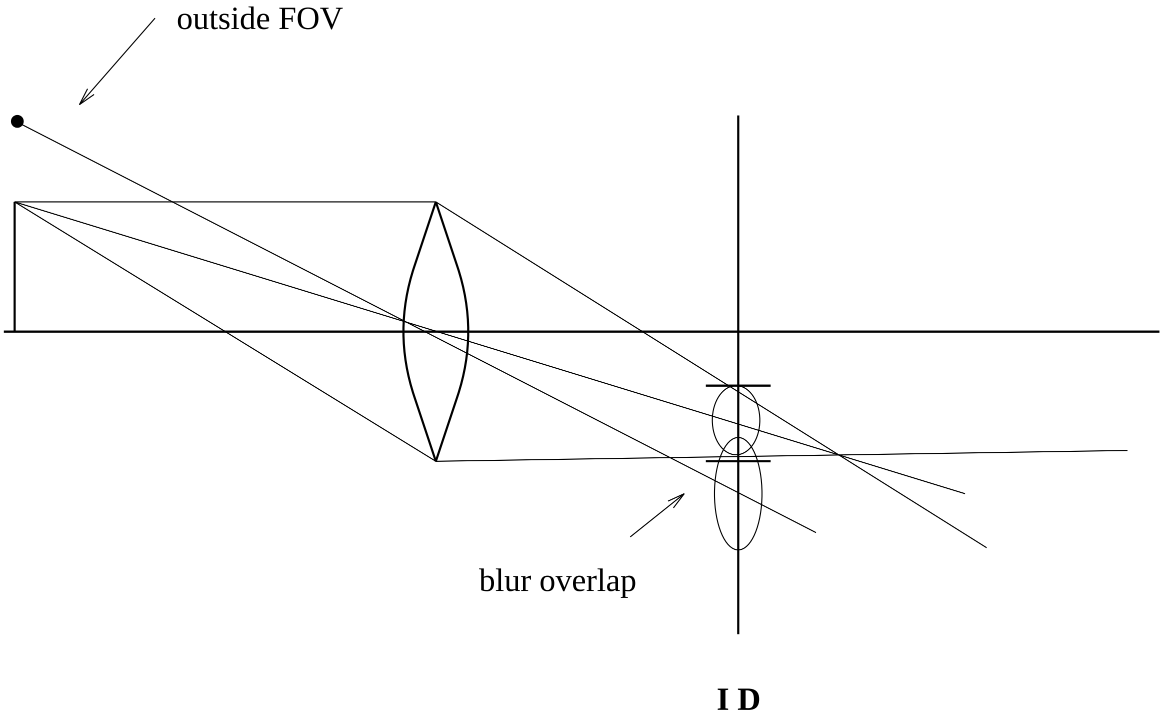

Figure 5.2: Image overlap problem

recovery is minimized. However, near the border of the whole image (field stop boundary) the error due to the image-overlap problem persists since image data outside the whole image is not available. Techniques for minimizing this error will be investigated in the future research.

Chapter 6

Three-dimensional Point Spread Function (3D PSF)

In this chapter we first discuss the point spread function for geometric optics model of image formation and derive an expression for the 3D PSF. Figure 2.1 shows the optical system described in chapter 2. According to paraxial geometric optics the blurred image of a point has the same shape as the lens aperture but scaled by some factor. This holds irrespective of the position of the point in the scene. The blurred image of a circular aperture lens will then have a circular shape referred as the "blur circle" with uniform brightness inside the circle and zero outside. This blur circle is the response of the optical system to a point source in the object space. Therefore it is the Point Spread Function (PSF) of the optical system and is denoted by $h(x, y)$. We define q to be the scaling factor where $q = 2R/D$.

In Fig. 2.1, from similar triangles we have

$$q = \frac{2R}{D} = \frac{s-v}{v} = s\left[\frac{1}{v} - \frac{1}{s}\right] \tag{6.1}$$

Substituting for $\frac{1}{v}$ from the lens formula (1) we obtain,

$$q = s\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{6.2}$$

Therefore, we obtain the radius of blur circle r''

$$r'' = q\frac{D}{2} = s\frac{D}{2}\left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s}\right] \tag{6.3}$$

Note that, from Eq.(4.3), $r'' = d''/2$. The magnification normalized radius of the blur circle is $r = d/2$. If the camera is a lossless system (i.e. no light is absorbed) then the PSF satisfies the condition

$$\int\int h(x,y)\ dx\ dy \ = \ 1 \tag{6.4}$$

The PSF is a given by cylindrical function:

$$h_1(x,y) = \begin{cases} \frac{1}{\pi r^2} & \text{if } x^2+y^2 \leq r^2 \\ \\ 0 & \text{otherwise.} \end{cases} \tag{6.5}$$

In IFA and IDA, researchers approximate the depth of a 3D scene to be a constant in small image regions in order to model the defocus process as a Linear-Shift-Invariant (LSI) process. Under such assumption the results obtained are approximate because the shape of a 3D object could vary within an image region. In UFDA, this approximation is avoided by considering a 3D PSF that takes depth variation into account in modeling the image formation.

Unlike the PSF of shift-invariant case, in UFDA the blur circle in 3D image space is varied according to the position of point in the object space.

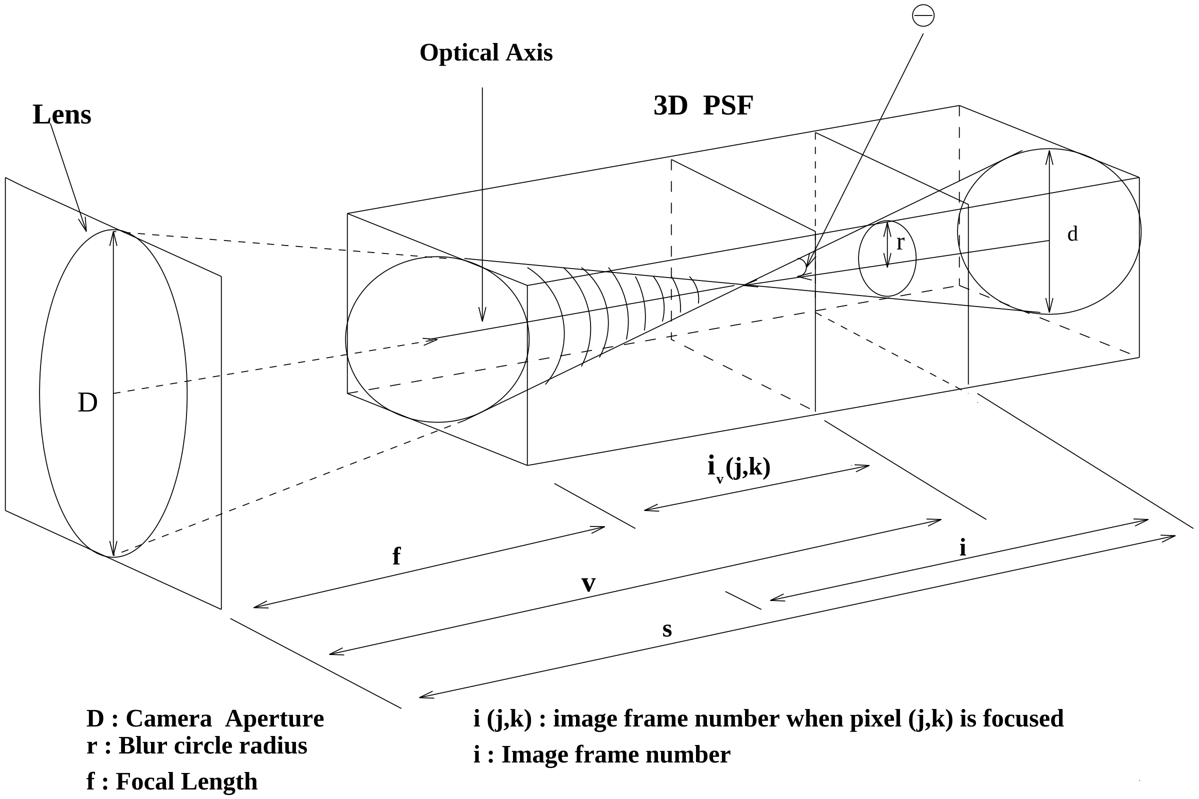

Figure 6.1: 3D PSF

Figure 6.1 illustrates the 3D PSF in the 3D image volume space where the lens position s is varied. It is a double cone with its axis parallel to the optical axis. The tips of the two cones meet at the focused point. The image brightness outside the cone is zero and inside the cone it is $1/(\pi r^2)$ where r is the radius of the cone cross-section. This radius r is also the radius of the blur circle which is the image of the point light source. In Fig. 6.1, the radius is given by

$$r = (i - i_v(j,k)) \cdot \delta \cdot \tan\theta \tag{6.6}$$

where i is image frame number on which the blur circle is observed, $i_v(j,k)$ is the frame number of the focused image point at pixel (j,k), δ is the distance between two successive frames, D is the diameter of lens aperture, θ is the half-cone angle given by

$$\tan\theta = \frac{D}{2(f + \delta \cdot i_v(j,k))} \approx \frac{D}{2f} \tag{6.7}$$

This angle θ depends on the focused position $i_v(j,k)$ but we approximate it to be a constant. We normalize the image magnification so that all image frames have the same magnification as that of the frame at $s = f$. In these magnification normalized images, the blur circle radius becomes:

$$r(i,j,k) = (i - i_v(j,k)) \cdot \delta \cdot \frac{D}{2s} \tag{6.8}$$

The 3D PSF is then expressed in terms of the blur circle radius derived above as

$$h(i,j,k) = \begin{cases} \frac{1}{\pi \cdot r^2(i,j,k)} & \text{inside the cone, i.e. } j^2 + k^2 < r^2 \\ 0 & \text{outside the cone} \end{cases} \tag{6.9}$$

For a circularly symmetric optical system as in this case, $h(i,j,k)$ can be represented by a two-variable function $h(i,r)$ were $r = \sqrt{j^2+k^2}$. By taking the reference distance u' to be infinity, the 3D PSF can be expressed in terms of the blur parameter $d = 2r$ as $h(x,y,d)$. The image sensing model developed earlier for this case is used to generate defocused image data and to recover 3D information.

Chapter 7

Optimization Techniques

The 3D shape and focused image reconstruction is formulated as an optimization problem where the difference between the sensed image data and the estimated image data is minimized. To achieve this minimization, three optimization techniques are proposed in this chapter. They are a gradient descent approach, a local search method and a regularization approach. The first two methods perform parameter searches for the local error minimization to this optimization problem. But for the third method, this problem is further defined as an ill-posed inverse problem. In order to solve this particular problem, principles from the calculus of variations are applied. The objective here is to develop a method that can take advantage of the structure of UFDA and obtain a more accurate solution with less computation. Theories and implementation techniques of these three methods are presented in the following sections.

7.1 Gradient Descent Approach

This methods is based on the iterative gradient descent with respect to the error function to find the lowest point. The direction of gradient downhill is defined as the direction of a negative gradient vector. In our applications, a piecewise planar approximation is applied for the 3D shape where the focused image surface in a small image region (of about 8x8 size) is approximated by a plane equation of

$$i_v(j,k) \; = \; i_0 + \tan\theta_x \; j + \tan\theta_y \; k. \tag{7.1}$$

In the above expression i_0 is the distance where the plane containing the planar patch intersects the optical axis, $\tan\theta_x$ is the slope with respect to the x axis and $\tan\theta_y$ is the slope with respect to the y axis. The gradient vector is represented by the partial derivatives of the error function with respect to the parameters i_0, θ_x and θ_y in this case. Optimization is carried out with respect to the three parameters i_0, tan θ_x and tan θ_y. The initial solution for 3D shape is obtained by an IFA and/or IDA method. In each iteration, an improved estimate of the parameters is obtained by adjusting their values based on the error gradient. The iterative search for the parameters that minimize the error stops when the error stops decreasing. The final error value depends on the shape and focused image of the object. Three variations of gradient descent approach have been used in our research. In the first method, a search for the parameters that minimize the error is made by considering one parameter at a time. We call this the Sequential Parameter Search (SPS) method. In the second method, the parameter space is searched considering all parameters(gradient vector) simultaneously. We call this the Parallel Parameter Search (PPS) method. In the third method, first SPS is applied and then PPS is applied. This is called the Sequential and Parallel Parameter Search (SPPS) method. More details on these methods are provided below.

7.1.1 Sequential Parameter Search (SPS)

The shape and focused image obtained using an IFA and/or IDA method in small (8x8) image regions is read in as the initial solution. Then the following steps of parameter search are carried out for each image region in a sequential manner for the SPS.

1. *Searching for i_0*

   Adjust i_0 by one depth unit to near or far side in the 3D image volume space and convolve the adjusted shape with its corresponding 3D PSF to generate a new sequence image data. Compute the error between this generated image data and sensed image data. Iterate this step until no further improvement is possible in minimizing the error.

2. *Searching for θ_x*

   Take the solution obtained in the previous step as initial solution. Change θ_x by tilting the planar surface patch in the 8x8 image region about the vertical axis in steps of 15° in the range -60° to $+60^\circ$. Find the value of θ_x that minimizes the error between the sensed image data and image data generated corresponding to the current solution. This results in an improved estimation for shape. The reason for searching in 15° intervals is because of the proportion of pixel size (0.013 mm) to depth unit (inter frame distance :0.03mm).

3. *Searching for θ_y*

   This step is similar to the previous step except that the planar patch is tilted about the horizontal axis instead of the vertical axis. This step improves the solution for θ_y. The result of this step is the solution for shape and focused image.

4. Repeat above steps for another image region (8x8) until the whole image has been processed.

7.1.2 Parallel Parameter Search (PPS)

In this method, the gradient vector which contains partial derivatives of error with respect to all three parameters is used. Partial derivatives are obtained by computing the change in error when the current solution is incremented by a small amount. A solution that results in a local minimum for the error is obtained by searching along the direction of the negative gradient vector. At the local minimum, a new negative gradient vector is computed and a new search is then started. This process is iterated until certain criteria is met.

7.1.3 Sequential and parallel Parameter Search (SPPS)

In the Sequential and Parallel Parameter Search (SPPS) method, first SPS method is applied. Then, using the result of SPS as initial solution, PPS is applied. Experimental results indicate that a somewhat better solution can be obtained by this method at the cost of increased computation. However, the percentage decrease in error is marginal compared to applying only SPS or only PPS.

7.2 Local Search Method

The local search method is an iterative error minimization method similar to a brute force search method. The difference is that this approach searches for the best 3D shape of FIS in a narrow range around the initial solution at each pixel. In our implementation, the initial solution is obtained from the SPS method. This solution is refined by adjusting the shape of FIS at each pixel by one unit at a time iteratively along the direction which decreases the error. The estimated 3D shape is continually updated until no further improvement is possible. This method provides a very accurate result but it needs a lot of computation.

7.3 Regularization Method

7.3.1 Introduction

In this section we extend UFDA by proposing a regularization technique to recover focused image and 3D depth-map of scenes. This approach provides substantial improvement in performance in comparison with the gradient descent approach used earlier by us for optimization. The problem of 3D shape and focused image recovery from defocused image data is formulated as an ill-posed inverse optics problem. The definition of a well-posed problem (by Hadamard) is that its solution exists, is unique, and depends continuously on the initial data. In the inverse optics problem considered here, presence of noise in defocused image data and effects of defocusing at borders makes the solution non-existent. Even if noise were absent, the solution will not be unique in those image regions where there is lack of sufficient image contrast. Many ill-posed inverse problems in vision have been solved using the regularization approach so far [62, 65]. A matrix based regularization technique was used by Ens [3] for image defocus analysis. This paper presents the application of regularization to UFDA.

The cost functional to be minimized in our approach is a weighted sum of a similarity functional E_i and a stabilizing functional E_s, given by $E_i + \lambda \cdot E_s$. E_i is the difference between the observed image data and estimated image data computed based on an initial solution for depth-map and focused image. E_s is a smoothness constraint based on the Laplacian squared of a local blur parameter, and λ is the regularization parameter. Our choice of the smoothness term based on the Laplacian squared of the blur parameter instead of surface curvature simplifies the derivation of the Euler-Lagrange equation. It also ensures the smoothness of surface structure. Three important constraints are used in minimizing the cost functional. They are– positivity of image brightness and depth, and a local error-control constraint. The first two capture physical reality and restrict the space of possible solutions. The last prevents overshoots and helps in fast convergence of the solution during iteration.

The performance of the regularization approach was compared with that of two other approaches. One is a gradient descent type approach and the other is a local search approach. The regularization approach performed better than the other two approaches in our experiments.

7.3.2 A Regularization Approach to UFDA

Form the image formation model of UFDA, the observed defocused image data is denoted by $g_o(x, y, d)$. It can be thought of as an image volume in the (x, y, d) space. The focused image of the scene will be denoted by $F_o(x', y')$ and the FIS of the scene will be denoted by $d_o(x', y')$. The observed image data is related to the focused image and FIS by Eqs. 4.11, 4.12. Given an initial estimate of the solution for focused image to be $F_e(x', y')$ and for FIS to be $d_e(x', y')$, the estimated image data $g_e(x, y, d)$ corresponding to the estimated solution can be computed using Eqs. 4.11,4.12. Now we can define the similarity functional E_i as

$$E_i \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(g_o(x,y,d) - g_e(x,y,d))^2 \; dx \; dy \; dd \qquad (7.2)$$

The similarity functional is a quadratic error term that represents the mean-square difference between the observed and estimated image data.

We formulate the stabilizing functional E_s as a function of Laplacian square of the estimated FIS $d_e(x', y')$:

$$E_s = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\nabla^2 d_e(x, y))^2 \; dx \; dy \tag{7.3}$$

The stabilizing term reflects the physical property of the 3D scene that surfaces are usually smooth. Our choice of the stabilizing term based on the FIS expressed in terms of the blur parameter $d_e(x', y')$ instead of surface curvature simplifies the derivation of the Euler-Lagrange equation. In addition, it offers computational advantages in the UFDA framework.

The cost functional E to be minimized is obtained by combining the similarity functional E_i and the stabilizing functional E_s as

$$E = E_i + \lambda E_s \tag{7.4}$$

The regularization parameter λ is a weighting factor which controls the closeness of the solution to the data with the degree of regularization. This formulation transforms the ill-posed problem into a well-posed problem.

The solution to the above minimization problem is subjected to additional constraints –positivity of depth and image brightness (for 8 bits/pixel grey-level):

$$0 \; \leq \; u'_e(x', y') \;\; , \;\; \text{and} \;\; 0 \; \leq \; g_e(x, y, d) \; \leq \; 255. \tag{7.5}$$

Another constraint which was found to be useful is a local error-control constraint. We require that a measure of local error decrease from one iteration to the next. If this is not satisfied, then the solution is not updated in the current iteration. However, it may be updated in the subsequent iterations. This is expressed by

$$e_i^n(x, y) \; \leq e_i^{n-1}(x, y) \tag{7.6}$$

where n denotes the iteration number and

$$e_i(x,y) \;=\; \int_{-\infty}^{\infty} (g_o(x,y,d) - g_e(x,y,d))^2 \; dd \tag{7.7}$$

The local error-control constraint is used to constrain the solution when the regularization process overestimates or underestimates the solution. This situation arises when the difference between the estimated and observed image data change dramatically from point to point.

According to calculus of variations[1, 4, 65], minimizing the functional Eq. (7.4) is equivalent to solving the associated Euler-Lagrange equation. In order to derive the desired Euler-Lagrange equation, we need to first express E in Eq. (7.4) more explicitly in terms of the dependent variables and functions. This is made possible by reexpressing the image formation equations 4.11, 4.12 in an unusual way. We define a new function $F_e''(x',y',d') = F_e(x',y')$ where $F_e(x',y')$ is the estimated focused image. Note that F_e'' is constant with respect to d'. The structure of the estimated FIS in the (x',y',d') space is specified by $\delta(x',y',d'-d_e(x',y'))$ where $\delta(\cdot)$ is the Dirac delta function. Now the estimated image data $g_e(x,y,d)$ can be expressed as:

$$g_e(x,y,d) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_e''(x',y',d')\cdot\delta(x',y',d'-d_e(x',y'))\cdot\; h(x-x',y-y',d-d')\; dx'\; dy'\; dd' \tag{7.8}$$

Now, by the sifting property [58] of the $\delta(\cdot)$ function, the above equation becomes

$$g_e(x,y,d) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_e''(x',y',d_e(x,y))\cdot\;\; h(x-x',y-y',d-d_e(x,y))\; dx'\; dy' \tag{7.9}$$

In the above equation, unlike Eq. 4.11, integration with respect to d' does not appear. Since the focused image is located only at $d_e(x, y)$, the above equation can be written as

$$g_e(x, y, d) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F_e(x', y')\cdot\; h(x - x', y - y', d - d_e(x, y))\; dx'\; dy'. \quad (7.10)$$

Derivation of the above equation which does not involve integration with respect to d' facilitates derivation of the Euler-Lagrange equations for optimization.

Now the functional E can be expressed as

$$E = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{F}(d_e, \bigtriangledown^2 d_e(x, y))\; dx\; dy \quad (7.11)$$

where

$$\mathcal{F}(d_e, \bigtriangledown^2 d_e(x, y)) \;=\; \int_{-\infty}^{\infty} (g_o - \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(x', y')\cdot \quad (7.12)$$
$$h(x - x', y - y', d - d_e(x, y))\; dx'\; dy')^2\; dd + \lambda\cdot(\bigtriangledown^2 d_e(x, y))^2$$

Hence, the necessary condition to minimize E is the following Euler-Lagrange equation [1]:

$$\mathcal{F}_{d_e} + \frac{\partial^2}{\partial x^2}\mathcal{F}_{d_{exx}} + \frac{\partial^2}{\partial y^2}\mathcal{F}_{d_{eyy}} \;=\; 0 \quad (7.13)$$

where $\mathcal{F}_{d_e}$, $\mathcal{F}_{d_{exx}}$ and $\mathcal{F}_{d_{eyy}}$ are partial derivatives of $\mathcal{F}$ with respect to d_e, d_{exx} and d_{eyy}. Here d_{exx} and d_{eyy} are the second order partial derivatives of d_e with respect to x and y. The derivation of the Euler-Lagrange equation is included in Appendix A.

From Eq. 7.12 we obtain

$$\mathcal{F}_{d_e} \;=\; \frac{\partial}{\partial d_e}\{\int_{-\infty}^{\infty}(g_o - g_e)^2 dd\} \;\text{, and }\; \frac{\partial^2}{\partial x^2}\mathcal{F}_{d_{exx}} + \frac{\partial^2}{\partial y^2}\mathcal{F}_{d_{eyy}} \;=\; 2\lambda\cdot\bigtriangledown^2(\bigtriangledown^2 d_e) \quad (7.14)$$

Therefore the Euler-Lagrange equation becomes

$$\frac{\partial}{\partial d_e}\{\int_{-\infty}^{\infty}(g_o - g_e)^2 dd\} + 2\lambda \cdot \bigtriangledown^2(\bigtriangledown^2 d_e) \;=\; 0 \tag{7.15}$$

The results in this section so far in terms of continuous functions can be easily extended to the case of discrete sampled functions. Therefore we do not present explicitly the discrete domain equations in order to be concise. In the subsequent parts of this paper, sometimes we refer to the continuous domain equations as if they were discrete domain equations. The context should make our intention clear.

The discrete version of Eq.(7.15) is obtained by the following procedure. First, the discrete version of Laplacian of Laplacian of blur parameter term $\bigtriangledown^2(\bigtriangledown^2 d_e)$ is expressed in the form [4] $\kappa \cdot (d_{j,k}$ - $\overline{d}_{j,k})$ where (j,k) are the discrete pixel coordinates corresponding to image coordinates (x,y). $d_{j,k}$ is the blur parameter at (j,k) which represents the level of defocus there. $\overline{d}_{j,k}$, and κ can be obtained by convolving $d_{j,k}$ with a computational molecule derived from a molecule that is appropriate for the biharmonic operator(see Fig. 7.1).

It can be shown that

$$\begin{aligned}\overline{d}_{j,k} \;=\; & \frac{1}{20}[8 \cdot (d_{j+1,k} + d_{j,k+1} + d_{j-1,k} + d_{j,k-1}) \\ & -2 \cdot (d_{j+1,k+1} + d_{j+1,k-1} + d_{j-1,k+1} + d_{j-1,k-1}) \\ & -(d_{j,k+2} + d_{j+2,k} + d_{j-2,k} + d_{j,k-2})]\end{aligned} \tag{7.16}$$

and $\kappa \;=\; 20$. The first term in Eq. 7.15 is $\frac{\partial}{\partial d_e}\mathcal{F}$, and it is estimated by a finite difference method. The discrete version of the Euler-Lagrange equation is

$$\frac{\partial}{\partial d_e}\sum_d (g_o - g_e)^2 + 2\lambda\kappa \cdot (d_{j,k} - \overline{d}_{j,k}) \;=\; 0 \tag{7.17}$$

		1		
	2	- 8	2	
1	- 8	20 (j,k)	- 8	1
	2	- 8	2	
		1		

Figure 7.1: Biharmonic operation mask

This leads to an iterative formula of the form

$$d_{j,k}^{(n+1)} = \overline{d}_{j,k}^{n} - \frac{1}{2\lambda\kappa} \cdot \frac{\partial}{\partial d_e} \sum_{d} (g_o - g_e^n)^2 \qquad (7.18)$$

Using this equation, the estimated solution of the 3D shape is updated iteratively.

7.3.3 Iterative Algorithm and Different Implementations

The following iterative algorithm was implemented for the regularization approach.

1. Record the observed image data $g_o(x, y, d)$ of a given 3D scene with focused image $F_o(x', y')$ and depth-map $u'_o(x', y')$. Let the FIS corresponding to this depth-map be $d_o(x', y')$.

2. Obtain an initial estimate of the solution for focused image and FIS by applying IFA and/or IDA to the observed image data. Let the estimated focused image be $F_e(x', y')$ and FIS be $d_e(x', y')$.

3. Update the initial solution iteratively using Eq. 7.18. In each iteration, the updated solution is checked for the depth and brightness positivity constraints and the local error-control constraint. If necessary, the solution is modified appropriately to satisfy the constraints.

4. Stop iterating when certain criterion is met (e.g. error stops decreasing or a certain maximum number of iterations has been completed).

It is necessary to select a good λ in order to attain a balance between the data similarity error (E_i) and smoothness of FIS (E_s). Depending on the initial solution, the derivative of the quadratic error term changes from point to point. Locally this variation may be small, but it can change substantially from one image region to another. This phenomenon makes the choice of the regularization parameter λ difficult. After a few trials, a value of 50 for λ was found to give satisfactory results. Further, the second term that gives the adjustment value at any iteration was limited to a magnitude of $0, \pm 1\epsilon, \pm 2\epsilon, \pm 3\epsilon$ where ϵ was the sampling interval (roughly equal to the size of one pixel) along the d dimension. In one variation of our algorithm the maximum adjustment value was taken to be 3ϵ in the first two iterations, 2ϵ in the next two iterations, and 1ϵ in the subsequent iterations. However there was no significant difference in its performance. The results of the regularization approach were compared with two other approaches explained later.

7.4 Conclusion

The principles of a gradient descent approach, a local search method and the theory of a regularization approach to UFDA are presented. Among these methods, the local search is most straight forward followed by the gradient descent method. The regualrization method is much more complicated in terms of mathematics and implementation. But for a smooth object, the regularization approach is much faster and gives better result compared to the other two type of methods. One disadvantage of this approach is that it involves the selection of the regularization parameter λ where a different choice of λ will result in noisy solution(under smoothing), or in the loss of relevant curvature information(over-smoothing)[80]. The gradient descent approach also performs well and is useful in the case of rough (non-smooth) objects for which the regularization approach is unsuitable. In particular, SPS offers a good balance between accuracy and computational time. The pixel-by-pixel search improves upon the accuracy of SPS at the cost of high computation.

Chapter 8

A Computation Algorithm to UFDA

In this chapter we present a computational algorithm for UFDA which uses variable number of images in an optimal fashion. The proposed computational algorithm consists of two main steps. In the first step, an initial solution is obtained by a combination of IFA, IDA, and interpolation. The initial solution is refined by minimizing the error between the observed image data and the image data estimated using a given solution and the image formation model. Optimization techniques introduced in chapter 7 are used for error minimization. Our experiments indicate that the most difficult part of the algorithm is to obtain a reasonable solution for the focused image when only a few image frames are available. We employ several methods to address this part of the problem. The algorithm has been implemented and experimental results are presented.

8.1 Introduction

A new theory named Unified Focus and Defocus Analysis (UFDA) [76, 77] is proposed in chapter 5 which provides a unified theoretical framework for IFA and IDA. For UFDA to deal with variable number of input images, we need to devise a systematic scheme to use the available image data effciently in order to maximize to power of UFDA. For instance, we can use either IFA, IDA or both IFA and IDA adaptively to find the depth and focused image as the initial solution to the UFDA optimization process to search a better solution. Four examples of how to applied IFA and IDA to obtain an initial solution adaptively are described as follows.

1. If the number of input images are sufficient, an IFA or a multiresolution IFA can be applied to obtain a fast and accurate result, or

2. IDA can be applied to two of the blurred image in the image sequence to find a rough depth map first and then a number of images can be taken around this preliminary solution to apply IFA to get a refined estimation.

3. If there are few input images ($3 \sim 5$), IFA can be applied to these images and determines which one among them is in better focus and then IDA is used to this image with one of its nearby neighboors to obtain new estimation.

4. If only two image are available, an IDA approach can be applied to estimate the depth and focused image.

In this chapter, we propose a computational algorithm for UFDA to deal with variable number of images. Our computational algorithm is demonstrated with several experiments on simulated image data. Depending on the number of image frames, these experiments are separated into two groups. In the first experiments group, UFDA is used to reconstruct the 3D shape and focused image from many image frames. Here, three optimization techniques - the gradient descent(SPS, PPS, SPPS), the local search, and the regularization methods are applied to a spherical object for error minimization. In the second experiments group, UFDA is applied to the case when few images are available. In this group, two experiments are carried out. One experiment uses three image frames to solve for the distance and focused image for a planar object. The gradient descent(SPS) is applied here for optimization. The other experiment uses five images frames and a regularization approach for a spherical object. One critical problem need to be addressed here is how to obtain good estimation on the focused image when only few image data is known. In the case of many images are available, regardless of what method is used to get depth information, the estimated focused image can be obtained by retrieving the image data directly from those image frames or a quadratic interpolation among the image frames is good enough to do the estimation. However, when small number of images are known, above methods will not valid. Therefore, how to get the focused image is indeed a deconvolution problem in such case. Several methods have been tried here to solve this problem for comparison. These methods include a Spatial Domain Convolution/Deconvolution method by Subbarao and Gopal[51], a Fourier domain method with wiener filter[56] and a cubic spline interpolation method[2]. For the case of many images are available, experimental results show that UFDA can attain a very accuracy result with the cost of processing of more images. On the other hand, for the few image case, experimental results show that obtaining a reasonable solution for focused image from only a few image frames is difficult but important in obtaining a good solution for the 3D depth-map of a scene.

8.2 Computational Algorithm

The computational algorithm we propose consists of two main steps. The first is the estimation of initial solution for 3D shape and focused image, and the second is the iterative improvement of the initial solution through error minimization. This algorithm extends the research on UFDA in two respects. The first extension is a systematic way to deal with variable number of images, from minimum needed to maximum possible. The second extension is an investigation of different techniques for estimating the focused image for the case of variable number of images.

We present the computational algorithm for a specific case based on an actual camera that we use in our laboratory, but it can be generalized easily. Our camera has the following specifications: focal length f=35 mm, aperture diameter D=9 mm, width of square pixels p=0.013 mm, and lens displacement for each step of a stepper motor is approximately 0.030 mm. Let the observed image data be represented by $g(x, y, d)$ where x and y are pixel indices and d is the blur parameter index. The image data will be represented with 8 bits per pixel and therefore g will be assumed to take integer values: $0, 1, 2, \cdots 255$. We choose the blur parameter d to be the blur circle diameter of a point light source at infinity. It is given by $d = Ds(1/f - 1/s)$ where s is the distance between the lens and the image detector. Based on the actual camera in our laboratory, d will be specified in units of 0.6 times the width of a pixel (e.g.. $d = 2$ means the blur circle diameter is equal to 1.2 times the width of one square pixel). Large images are divided into many smaller subimages and processed. We take the subimages to be of size 32×32. Therefore x and y will take integer values: $0, 1, 2, \cdots 31$. Based on our camera, we will assume that d takes integer values: $0, 1, 2, \cdots 99$. The focused image surface $d_v(x, y)$ which represents the 3D shape of the scene will also take integer values in the range 0 to 99.

The image data $g(x, y, d)$ can be thought of as an "image volume" in the (x, y, d) space. The FIS $d_v(x, y)$ is embedded in this volume. The value of the image volume data on the FIS gives the focused image $F(x, y)$ of the scene, i.e.

$$F(x, y) = g(x, y, d_v(x, y)) \tag{8.1}$$

An image volume corresponds to a small and fixed field-of-view of the camera. In each image volume (or field-of-view) FIS $d_v(x, y)$ can be approximated by a piecewise constant or a planar or a smooth curved surface. The 3D shape parameters of this surface (i.e. FIS $d_v(x, y)$) and the focused image $F(x, y)$ are obtained by processing all or some of the image volume data.

A sequence of image frames consists of cross sections of the image volume $g(x, y, d)$ taken at different values of d. Let the values of d where the cross sections are taken be $d_1, d_2, \cdots d_n$.

IFA: Application of IFA to the image sequence consists of computing a focus measure in small image regions, say 8×8, for all images in the sequence and finding the value $d = d_i$ for which the focus measure is a maximum. The FIS in the 8×8 region is given by $d_v(x, y) = d_i$ and the focused image in the 8×8 region is given by $F(x, y) = g(x, y, d_i)$.

Combining IFA and Interpolation: The solution provided by IFA can be improved through interpolation by fitting smooth curves. For example, let $M(d)$ be the focus measure function of the 8×8 image region around the position $d = d_i$ estimated by IFA described above. Then a smooth curve can be fitted to the points $M(d_{i-1})$, $M(d_i)$, and $M(d_{i+1})$, and the position $d = d_f$ where $M(d)$ is a maximum can be obtained [22]. Once we find $d_v(x, y) = d_f$, the focused image $F(x, y) = g(x, y, d_f)$ can be obtained by interpolating the points $g(x, y, d_{i-1})$, $g(x, y, d_i)$, and $g(x, y, d_{i+1})$ and perhaps other nearby points.

IDA: In contrast to IFA which requires many image frames, IDA can be applied to only two image frames, say for $d = d_j$ and $d = d_k$. It can provide an estimate of $d_v(x, y)$ and the focused image $F(x, y)$.

Combining IFA and IDA: Given a sequence of images as above, IFA and IDA can be combined as follows. First IFA is applied as above and the position $d = d_i$ where the 8×8 image region is in best focus is found. This will be $g(x, y, d_i)$. Also the second best focused image close to d_i is found (at d_{i-1} or d_{i+1}) based on the computed focus measures. Let this be $g(x, y, d_{i+1})$. Then IDA is applied to these two most focused images to get a solution for shape and focused image. If the distance measured along the d dimension between successive image frames in an image sequence is high (about 5 or more), then combining IFA and IDA is better than combining IFA and interpolation. IDA gives better results than interpolation when the image frames are far apart.

The solutions for shape and focused image in 8×8 image regions are synthesized to obtain a solution for the entire 32×32 image. The solution thus obtained can be taken as initial solution in UFDA and it is further improved through error minimization as discussed in the previous section.

If n image frames are given, we assume that these frames are roughly uniformly placed along the d dimension and cover the entire range of values taken by the FIS $d_v(x, y)$. An algorithm for obtaining the initial solution can be summarized as follows.

If the distance between image frames is only one along the d dimension, then use IFA as it will give very good results. If the inter-frame distance is about 2 to 4, then IFA combined with interpolation should be used to get good results. If the inter-frame distance is 5 or more, then the method that combines IFA and IDA described earlier should be used. In all cases, if the number of image frames given is only two–the minimum required, then only IDA can be applied.

If the images can be recorded dynamically, then first only two image frames are recorded far apart (say at $d = 20$ and $d = 50$ for our camera specified earlier). Then IDA is applied to find a rough depth map. Based on this the approximate minimum d_{min} and maximum d_{max} of the FIS $d_v(x, y)$ are found. After this, further image frames are recorded for d only in the range

d_{min} to d_{max}. This avoids recording and processing of unnecessary image frames.

In the error minimization step, in each image region, it is sufficient to consider a few image frames (about 5) around the most focused image frame in that image region.

In the algorithm outlined above, the problem of estimating the focused image given an estimation of 3D shape is a very difficult one when the number of available image frames is limited. The focused image will have to be estimated from a few image frames which are in best focus through deconvolution. If a piecewise constant approach is used for 3D shape, then a Fourier domain deconvolution can be applied (e.g. Weiner Filter).In other cases a spatial domain method will have to be used. We have found that the STM method based on IDA in [46] gives good results. Even then, deconvolution in the presence of quantization and noise is found to be a very difficult problem but satisfactory results can still be obtained.

8.3 3D shape and Focused Image Recovery from many image frames

8.3.1 Initial Solution

For this case the initial solution can be obtained by an IFA method or IFA with interpolation approach. A traditional shape from focus method is applied here to find the initial solution. First the defocused image data is recorded by the camera system. Then the image frames are divided into 8x8 subimages and in each such subimage a focus measure is computed at every two or four image frame interval. We used the energy of the Laplacian as the focus measure. In each 32x8x8 image space (for 32 image frames), the subimage which has the maximum focus measure is determined. The position of such subimages 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. In this method, since all pixels in an image region are assigned the same depth, a piecewise constant approximation is used for the object shape. The procedure for obtaining the initial solution is summarized below.

1. Partition each observed (sensed) image into 8x8 subimages.

2. Compute a focus measure by summing square of image laplacian in each 8x8 window at every two or four image frame interval. If the focus position is neither the nearest position nor the farthest position of the sensed image frame then apply quadratic interpolation to the initial focus distance and its neighboring positions to find the focus position. After obtaining the focus position of this subimage we assign this focus position (depth) to every point in this subimage. Then use this focus position in quadratic interpolation on three neighboring image frames (centerd at this position) to find the focused image.

3. Repeat (2) until the whole image region has been processed.

The above steps yield an initial estimation for the 3D shape and focused image of the scene. This will be used as the starting solution in the optimization process. Three optimization techniques are used here for the experiments. The implementation procedure and experimental results are presented next.

8.3.2 Experiment with Gradient Descent Approach

A particular case of the UFDA approach that has been implemented by us is described in this section. In this case, among the three camera parameters (D, f, s), only s is changed. Also, paraxial-geometric optics model of image formation is used in deriving the PSF. Fig. 6.1 shows the PSF $h(\mathbf{e}, i, j, k)$ of Eq.(4.4) in the 3D image volume space. From the derivation of 3D PSF in chapter 6 we obtain Eq.(6.9). This PSF was used in the following experiments. A $32 \times 32 \times 32$ size image volume data was synthesized where the FIS $i_v(j,k)$ was a hemispherical object (with radius 24) and the focused image $F(j,k)$ was a checker board (see Figs. 8.1 and 8.2).

The observed image data $g(i,j,k)$ was synthesized using Eq.(5.3). Then an initial solution for the focused image $F'(j,k)$ and the FIS $i'_v(j,k)$ was obtained using an IFA method. In the IFA method, a focus measure (energy of image Laplacian) was computed in 8×8 non-overlapping regions. A piecewise constant approximation to the FIS in each 8×8 image region was obtained by finding the position $i'_v(j,k)$ where the focus measure was a maximum. These estimated solutions and the 3D PSF were used to compute the estimated image data $g'(i,j,k)$ using Eq.(5.3). Then the error E between the observed image data $g(i,j,k)$ and the estimated image data $g'(i,j,k)$ was computed using Eq.(5.4). This error was minimized using a gradient descent approach. In order to apply a gradient descent approach for error minimization, the FIS was approximated by a piecewise planar surface patch in 8×8 image regions where each surface patch was expressed by Eq.(45). The error E was minimized in $8 \times 8 \times 8$ image volume regions surrounding the initial estimates of the solutions with respect to the three parameters i_0, θ_x, and θ_y.

Three variations of gradient descent appraoches– SPS, PPS, and SPPS, were tried. These three methods were applied for two sets of data, one set with only quantization noise and another with both quantization noise and zero-mean Gaussian random noise with standard deviation 1.0.

Experimental results on accuracy and computation time are presented here for each of the three optimization methods. The initial solutions obtained by IFA method are shown in Figs. 8.3 and 8.14. The improved solutions obtained by SPS, PPS and SPPS are shown in Figs. 8.5, 8.6, and. Figs. 8.7, 8.8and. Figs. 8.9, 8.10 respectively. The results show that SPPS obtains the most accurate data among the three methods. The processing times for SPS, PPS, and SPPS, are approximately 32, 40, and 55 minutes respectively on a SUN Sparcstation 1. SPPS attains a small improvement in accuracy at the cost of much more computation than SPS or PPS. Also the rate of convergence for SPS and PPS is faster than SPPS. From these comparisons we feel that use of SPS alone will be adequate in practical applications in terms of both accuracy and computational efficiency. Further, these results show that, as expected, the improvement obtained by the optimization method is small in image regions where the slope of the object surface is small, but the

Table 8.1: Error percentage of gray level per pixel for each method

	IFA Method	SPS	PPS	SPPS
Quantization Noise	8.75%	5.25%	7.25%	4.75%
Quantization and Electronic Noise	8.5%	5.8%	7.8%	5.26%

improvement is significant in regions where the slope is large. The percentage error in grey level between the observed image and the images estimated from the best solutions are shown in Table 8.1.

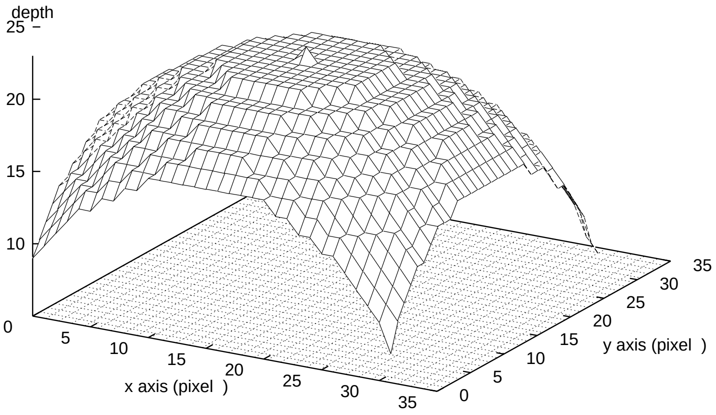

Figure 8.1: Original FIS of a 32x32 Hemisphere

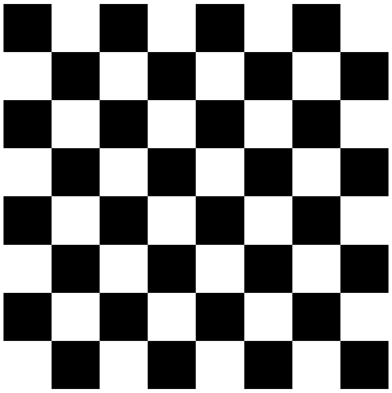

Figure 8.2: Original Focused Image (32x32)

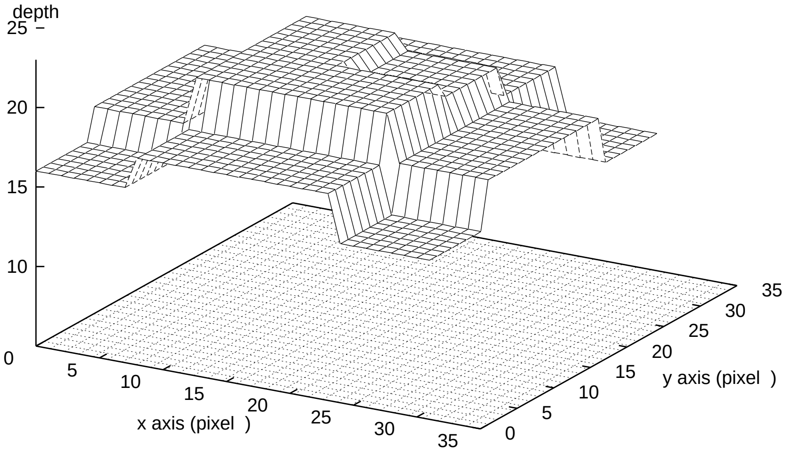

Figure 8.3: Initial Solution for FIS from an IFA method

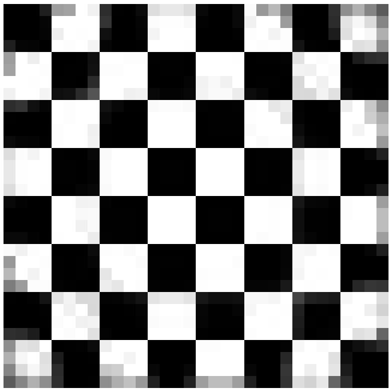

Figure 8.4: Initial Solution for Focused Image from an IFA method

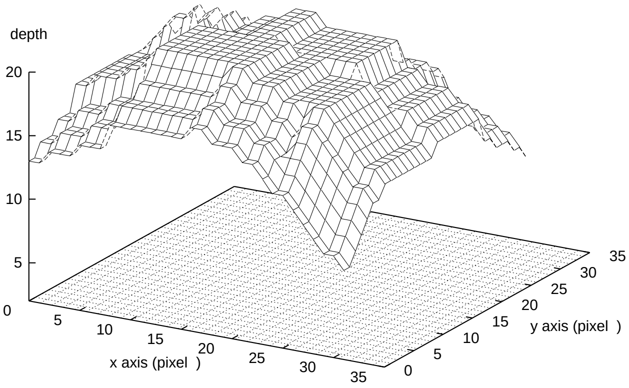

Figure 8.5: FIS by SPS

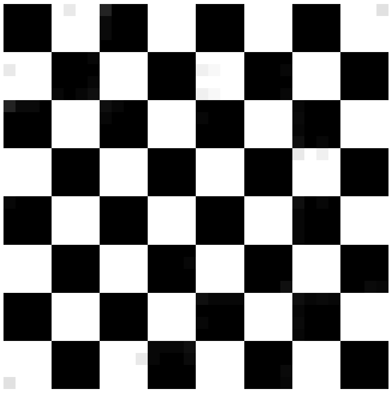

Figure 8.6: Focused Image by SPS

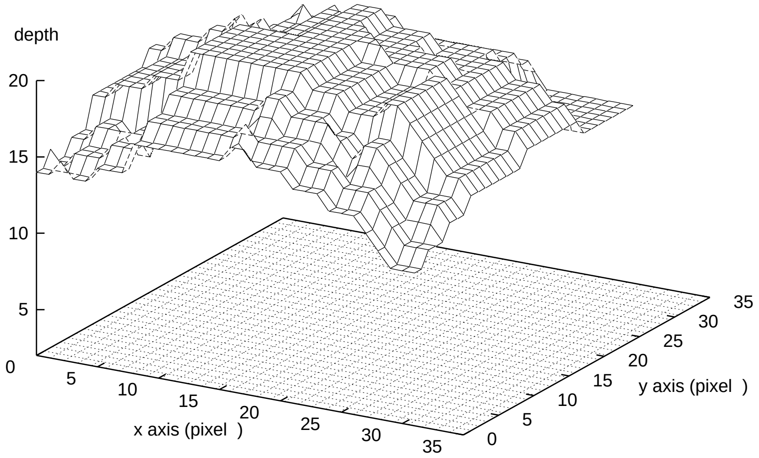

Figure 8.7: FIS by PPS

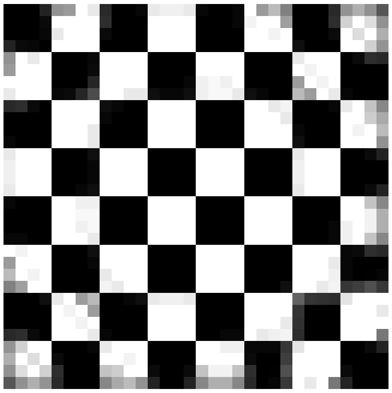

Figure 8.8: Focused Image by PPS

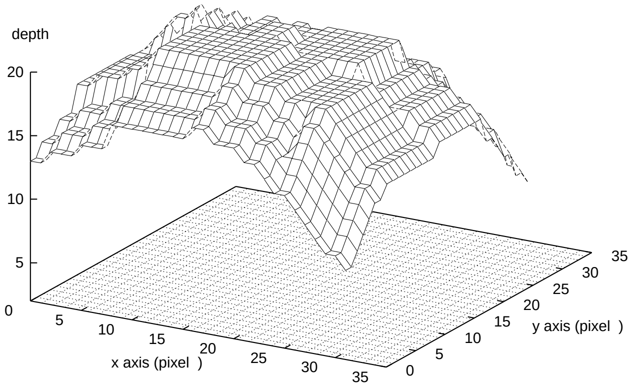

Figure 8.9: FIS by SPPS

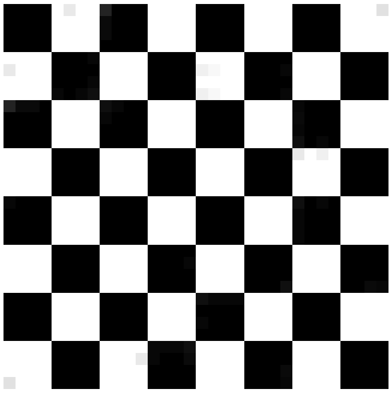

Figure 8.10: Focused Image by SPPS

8.3.3 Experiment with Regularization Approach

The same image model of a $32 \times 32 \times 32$ size image volume data was applied here. (see Figs. 8.1, 8.2) . Another similar but larger $64 \times 64 \times 32$ size image volume data was generated where the FIS was a hemispherical object (with radius 24) and the focused image was a checker board the same as the $32 \times 32 \times 32$ model. From UFDA, the observed image data $g(i,j,k)$ was synthesized using Eq.(4.11) with camera parameters (D=9mm, f=35mm, s= 35mm to 35.9mm). Then an initial solution for the focused image and the estimated 3D shape was obtained using same procedure as in the gradient descent method experiment. These estimated solutions and the 3D PSF were used to compute the estimated image data $g'(i,j,k)$ using Eq.(4.11). The error E_i between the observed image data $g(i,j,k)$ and the estimated image data $g'(i,j,k)$ was computed using Eq.(7.2). The regularization method that we derived earlier in chapter 7 was applied to the image volume data.

The original focused image surface and focused image for the $32 \times 32 \times 32$ and $64 \times 64 \times 32$ image model are shown in Figs.(8.11, 8.12, 8.15) . The initial solution for these two images are shown in Figs.(8.13, 8.14, 8.16). The results for the regularization method (7 iterations) are presented in Figs.(8.17, 8.18). For $64 \times 64 \times 32$ image data, the results of the regularization method for 2 to 7 iterations are shown in Figs.(8.19, 8.20, 8.21, 8.22, 8.23, 8.24). The error percentage of gray level per pixel for $32 \times 32 \times 32$ and $64 \times 64 \times 32$ image models at different number of iteration are shown in Table 8.2.

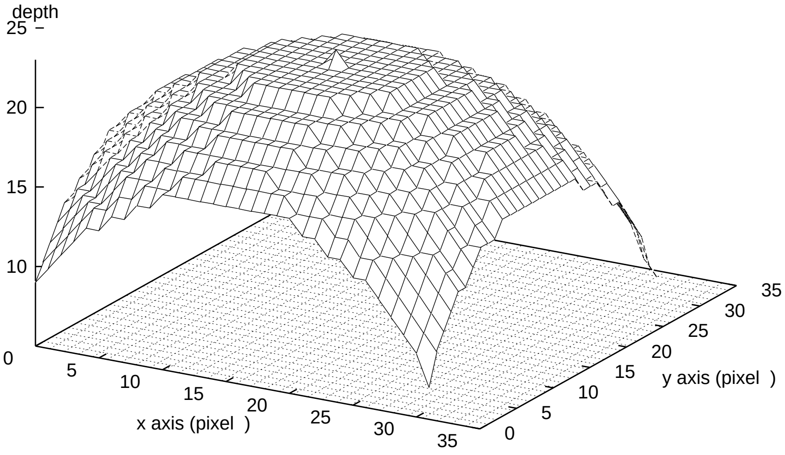

Figure 8.11: Original FIS of a 32x32 Hemisphere

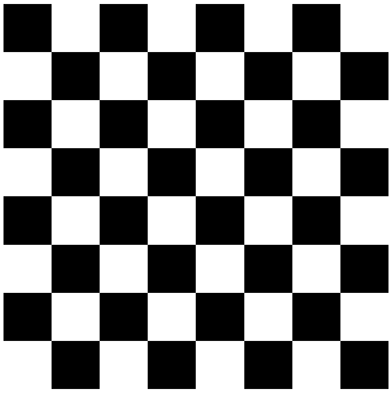

Figure 8.12: Original Focused Image

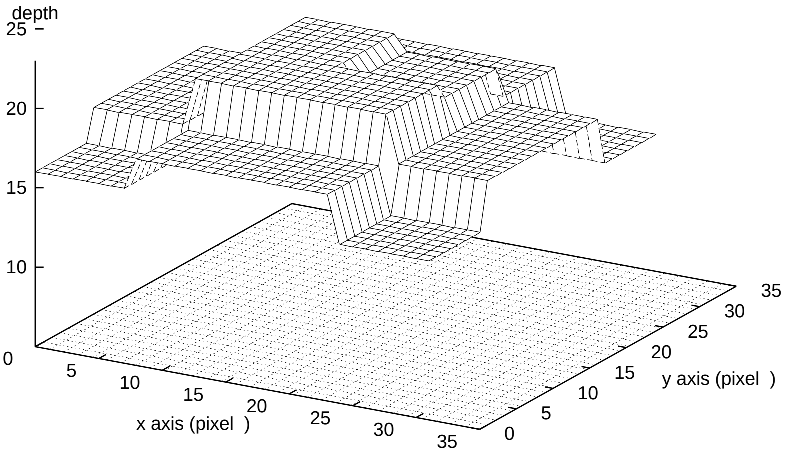

Figure 8.13: Initial Solution for FIS(32x32)

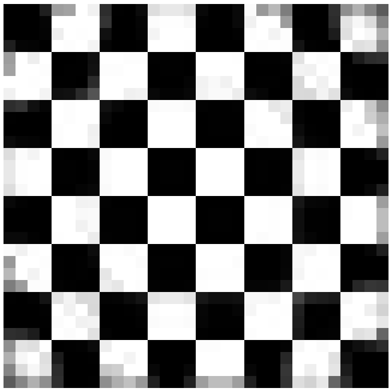

Figure 8.14: Initial Solution for Focused Image from an IFA method

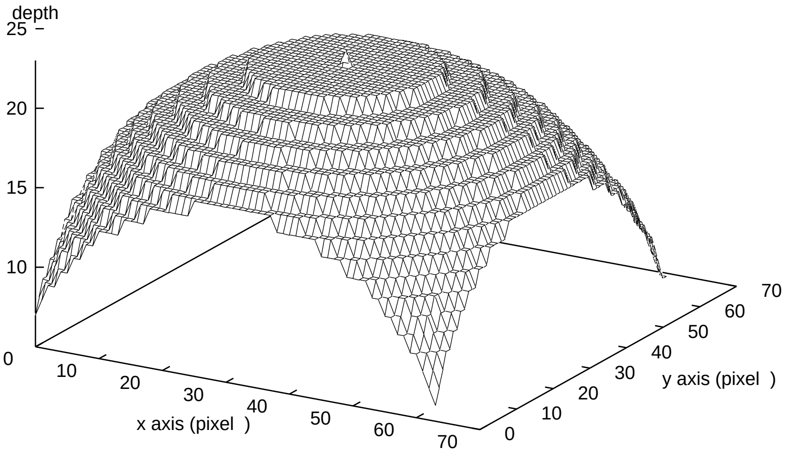

Figure 8.15: Original FIS of a 64x64 Hemisphere

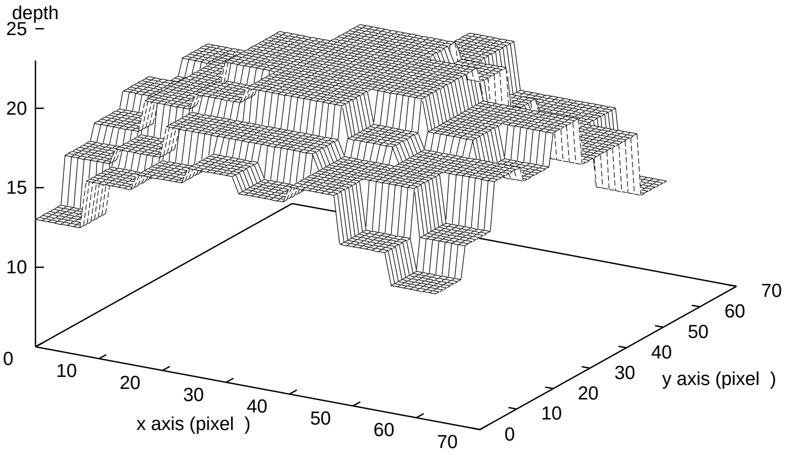

Figure 8.16: Initial Solution for FIS (64x64)

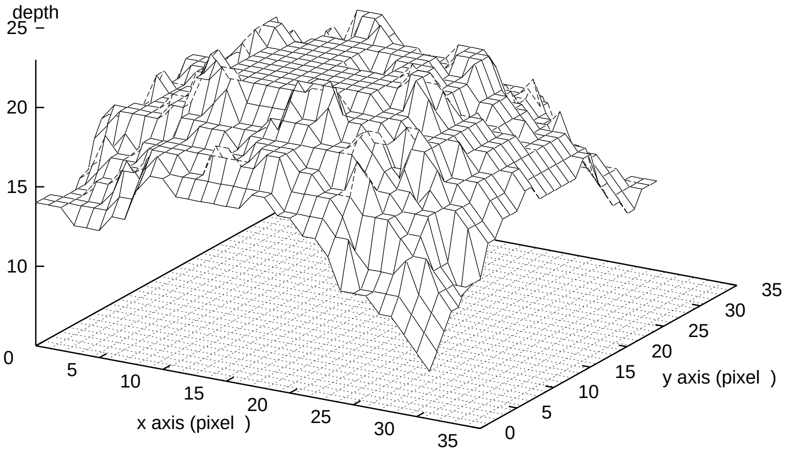

Figure 8.17: FIS by regularization (7iter.)

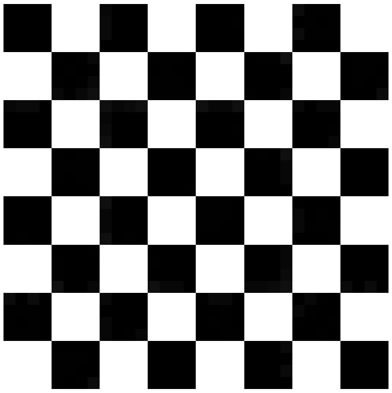

Figure 8.18: Focused Image by regularization (7iter.)

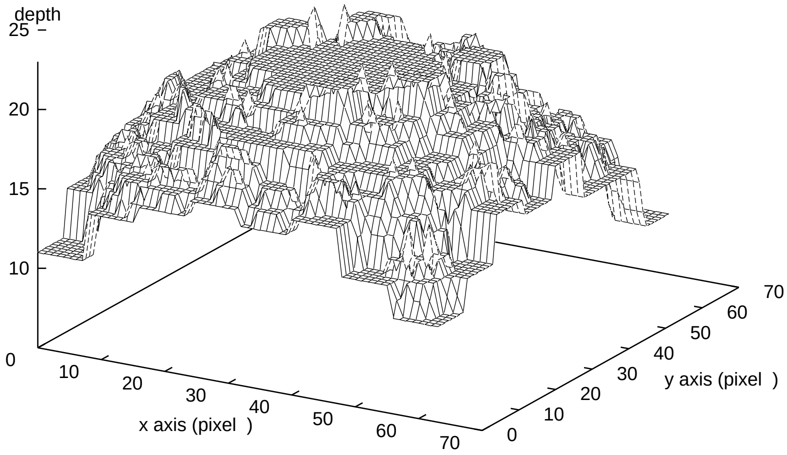

Figure 8.19: FIS(64x64) recovered (2 iters.)

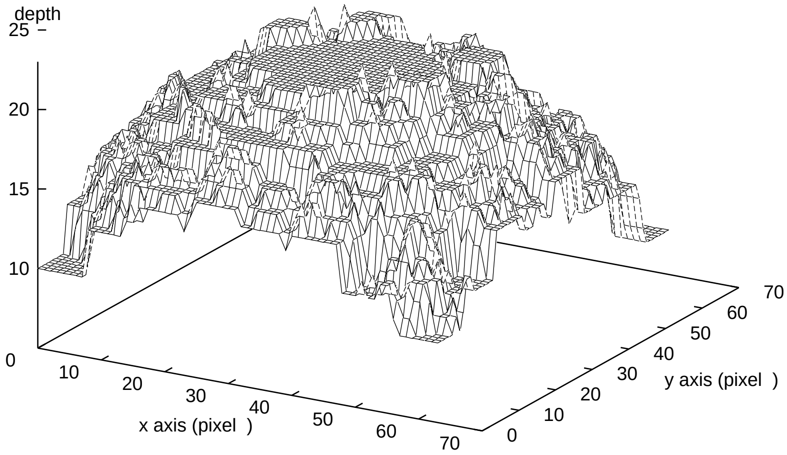

Figure 8.20: FIS(64x64) recovered (3 iters.)

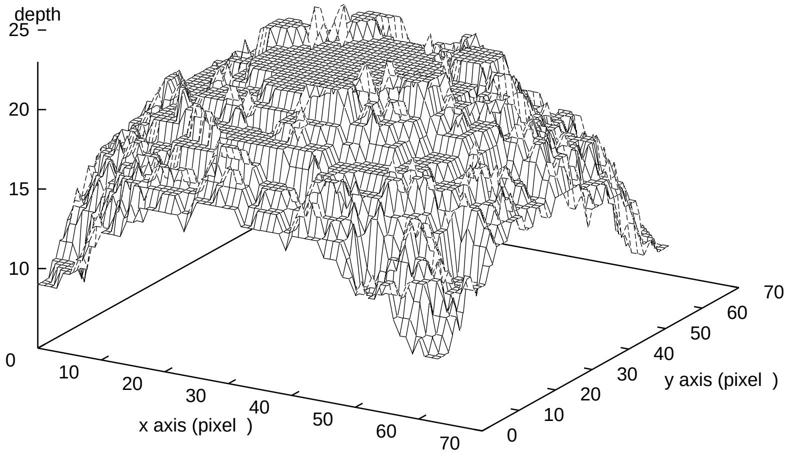

Figure 8.21: FIS(64x64) recovered (4 iters.)

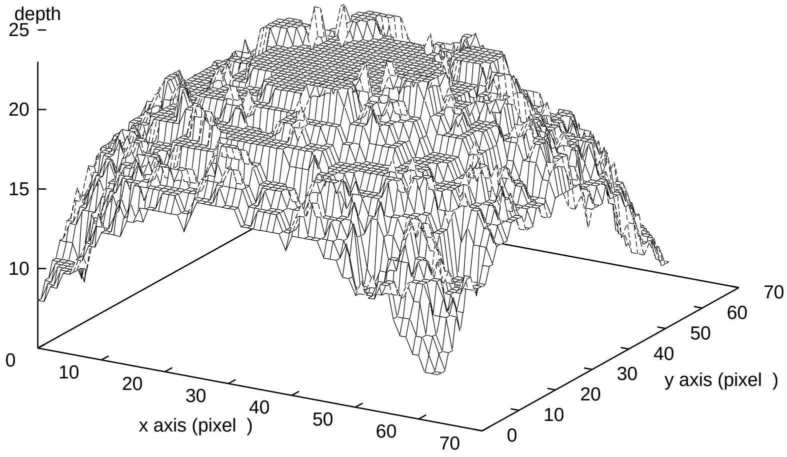

Figure 8.22: FIS(64x64) recovered (5 iters.)

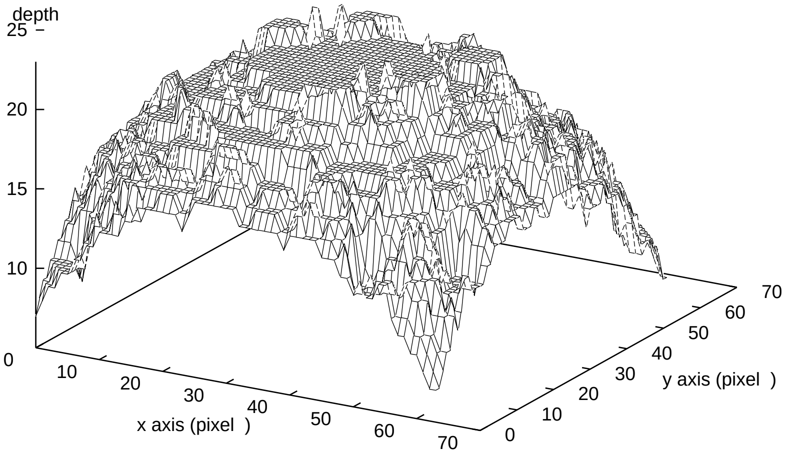

Figure 8.23: FIS(64x64) recovered (6 iters.)

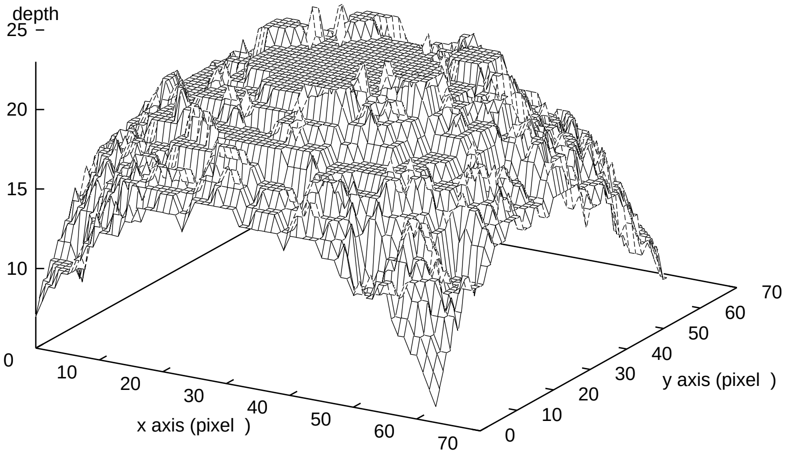

Figure 8.24: FIS(64x64) recovered (7 iters.)

Table 8.2: Error percentage of gray level per pixel for regularization method at different iteration.

Number of iteration	Image model: 32x32x32	Image model: 64x64x32
1	6.00%	4.53%
2	4.43%	3.59%
3	3.14%	3.15%
4	2.58%	2.90%
5	2.32%	2.72%
6	2.13%	2.69%
7	2.02%	2.67%

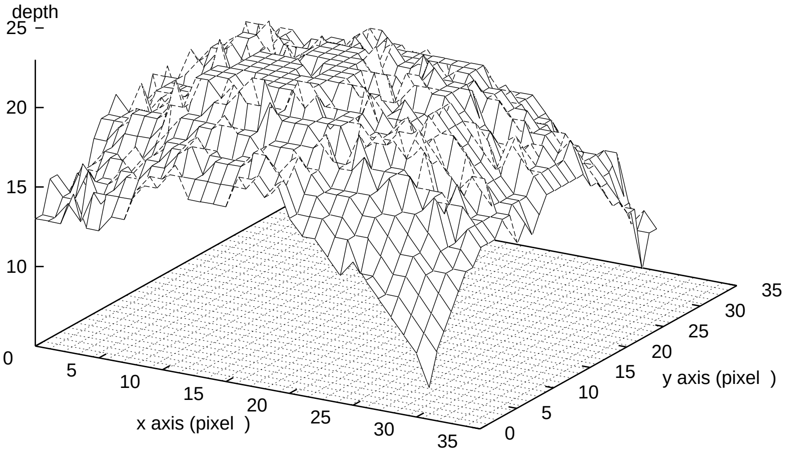

Figure 8.25: FIS (32x32) by local search method

8.3.4 Experiment with a local search method

The same $32 \times 32 \times 32$ image model is used here for the experiment. The initial solution also obtained from the procedures described in section 8.3.1. The results for local search method are shown in Figs.(8.25, 8.26).

8.3.5 Performance comparison

The performance of these three optimization methods in above experiments are compared to see which one is most useful in practical applications. A percentage error in gray level per pixel between the observed image and the images estimated from the best solutions for the $32 \times 32 \times 32$ image volume of these experiments and a comparison of the computation time for each method

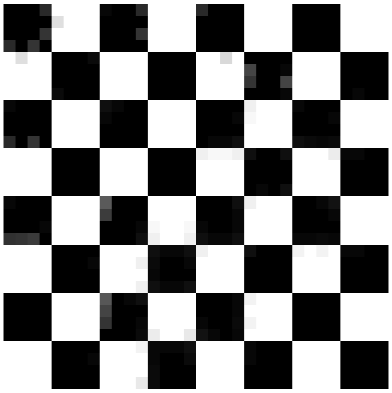

Figure 8.26: Focused Image by local search method

are shown in Table 8.3. All the simulation time is measured on a Pentium 166 MHz personal computer. From Table 8.3 and those reconstructed 3D shapes and focused images we conclude that for this test image model ,the regularization approach performed better than other approaches in terms of accuracy and computational time. Although the local search method also can obtain a good estimated solution but it involves to much computation. Also the gradient descent approach only attained moderate accuracy. Therefore, we recommend this approach for smooth objects in practical applications.

Table 8.3: Performance comparison of different methods

	SPS	PPS	SPPS	Local Seaech	Regu. (7 iters.)
% of gray level	5.25%	7.25%	4.25%	2.89%	2.02%
Computation time	4	5	7	14	5

8.4 3D shape and Focused Image Recovery from few image frames

In the following sections, we present e the results of two experiments where only a small number of image frames (3 and 5 respectively) are used.

8.4.1 Planar Object

We generated simulated image data for a CCD camera with the same parameters specified earlier: focal length f=35 mm, aperture diameter D=9 mm, square pixel width p=0.013 mm, and unit of image frame distance 0.6 pixel (i.e. d=1 corresponds to 0.6 pixel). In this experiment, an image volume of 32×32×32 was considered and three defocused image frames at position $d = 5, 16, 27$ were synthesized using Eq. (9.1). The FIS of the scene was a planar object given by $d_v(x, y) = 13$ and the the focused image is a checker board pattern (see Figs.(8.27, 8.31)). According to the proposed computational algorithm, an IFA method is first applied to these three images. Focus measures were computed over the entire 32×32 image frames without dividing them into smaller subimages. The two best focused image frames were determined based on the focus measures. These two frames were used in an IDA method (STM proposed in [51]) to find an initial solution for distance of the planar object and the focused image. This STM method uses a blur parameter σ to obtain 3D depth map and focused image. This parameter is related to the blur circle diameter d of point light sources by [42, 53] $\sigma = d/(2\sqrt{2})$. The IDA method (STM) was implemented exactly as described in [51]. This involves constructing a histogram of σ obtained at each pixel in the 32×32 image and taking the mode of the histogram as the solution for FIS $d_v(x, y)$.

After estimating the FIS as above, the focused image was estimated using three different methods to investigate and evaluate their relative performance. The first was based on a spatial domain deconvolution formula (inverse S transform) derived under some weak assumptions in [51]:

$$F(x, y) = g(x, y) - \frac{\sigma^2}{4} \bigtriangledown^2 g(x, y) \tag{8.2}$$

where $g(x, y)$ is the least defocused image available. The second method was based on Wiener filter in the Fourier domain as described in [46]. The third was using cubic spline interpolation [2] method on the three given image frames.

The solution obtained for FIS and focused image (from each of the three methods) was used as initial solution in UFDA based on a special case of gradient descent approach presented in [76]. It is a sequential parameter search (SPS) method with only one parameter– the distance of the planar object.

The estimated solutions and the 3D PSF were used to compute the estimated image data $g_e(x, y, d)$ using Eq. (9.1). Three images frames were estimated corresponding to the three frames of observed data $g_o(x, y, d)$. The error E between the observed image data $g_o(x, y, d)$ and the estimated image data $g_e(x, y, d)$ is computed using Eq. (9.4). The error was minimized iteratively using the gradient descent method for estimating FIS and the three different methods of focused image estimation.

The initial solution from the IFA method are shown in Figs.(8.28, 8.31). The position of the planar object is estimated to be at position 16 and the focused image is the image data at this position. The initial solution from the IFA method followed by the IDA (STM) method are shown in Figs.(8.31, 8.30). The estimated position of the planar object is 14 and the focused image is obtained from Eq. (8.2). The reconstructed 3D shape from the UFDA optimization with STM, Fourier method, and cubic interpolation are positions 13, 14, 14 respectively (Fig. 8.31). The focused image obtained using SPS and STM is shown in Fig. 3. The results from these three methods are also presented in terms of the percentage error in gray level per pixel between observed image data and estimated image data in Table 8.1.

These results show that the STM method is better than the other two. This can be explained as follows. The checker board test image that we used will introduce error into the Fourier method because of the periodicity property of the discrete Fourier transform. As for the cubic spline interpolation method, since there are only three observed images that are far apart, interpolation gives very poor results as expected.

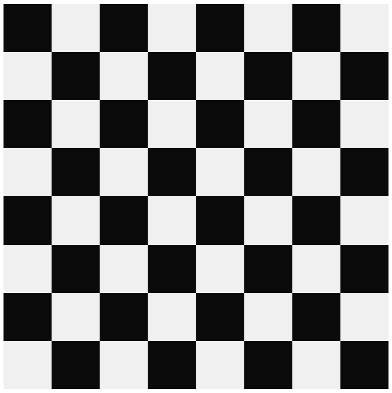

Figure 8.27: Original Focused Image

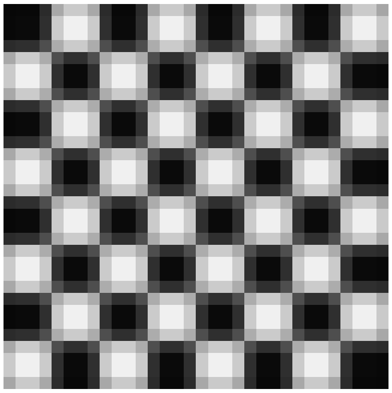

Figure 8.28: Focused Image from an IFA method

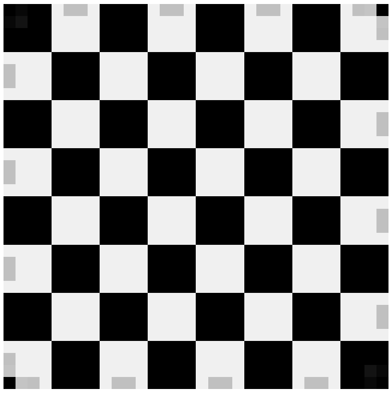

Figure 8.29: Focused Image by an STM method

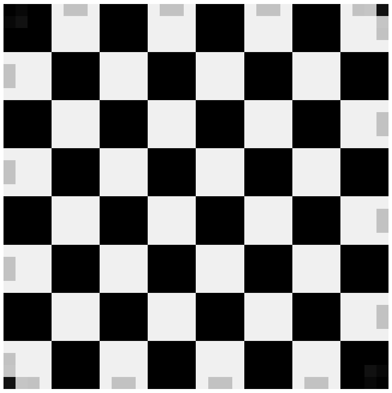

Figure 8.30: Focused Image by SPS with STM

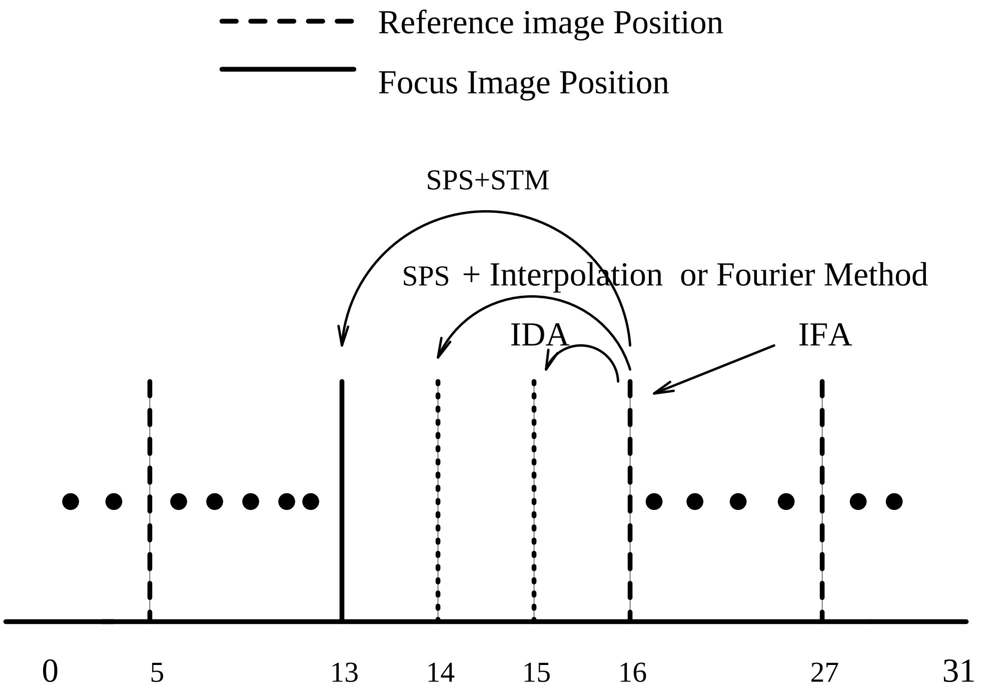

Figure 8.31: Focused position form IFA, IDA and SPS

8.4.2 Spherical Object

The second experiment is for an object whose FIS has a spherical shape shown in Fig. 8.32. In a 32×32×32 image volume similar to the first experiment, the FIS has a radius of 24 (Fig. 8.32). The focused image is the same checker board pattern shown in Fig. 8.33. Five image frames at positions $d = 4, 10, 16, 22, 28$ are used. These images are processed first by an IFA method followed by an IDA method (STM) as before for the planar object. In this case, due to the curved shape of the object, the 32×32 image frames are divided into 4×4 subimages and processed separately. A combination of IFA and IDA are applied separately in 4×4 image regions separately. In applying IDA (STM), the smoothing and differentiation filters proposed by Meer and Weiss[28] were choosen to be of size 3×3. One estimate of the blur parameter σ (and hence $d_v(x, y)$) is obtained at each pixel by integrating over the 3×3 region around the pixel.

This initial solution is used in the UFDA optimization method based on a regularization approach proposed in [77]. The five observed image frames and the corresponding five estimated image frames are used in computing the error measure. The error was minimized iteratively subject to the smoothness of the FIS as in [77]. In each iteration, after a solution for FIS was obtained, the solution for the focused image was estimated by two different approaches. One was the STM method described earlier and the other was the cubic spline interpolation method.

The initial solution from an IFA method and an IFA followed by an IDA method are shown in Figs.(8.34, 8.35, 8.36, 8.37). In these figures, we see that

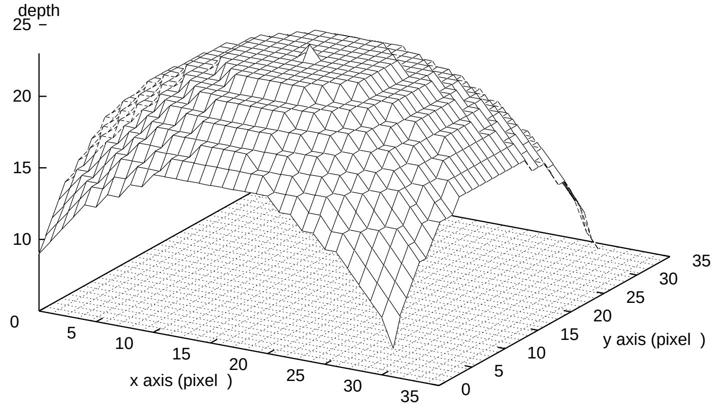

Figure 8.32: Original FIS of a 32x32 Hemisphere

the focused image from STM is much better than that from a traditional IFA method but the 3D shape has only limited improvement. This is because the IFA and IDA methods operate on very small image regions (4×4) and STM is very sensitive to noise if the window size is too small. The solution for the regularization method after five iterations is shown in Figs.(8.38, 8.39). The same results are presented in terms of percentage gray level error in Table 8.2. The results for the interpolation method were poor and therefore are not included here. The improvement of initial solution by UFDA optimization based on regularization is limited. The main reason for this is the difficulty in obtaining good estimates of the focused image for curved objects which result in very poor initial solutions for FIS as in our experiment.

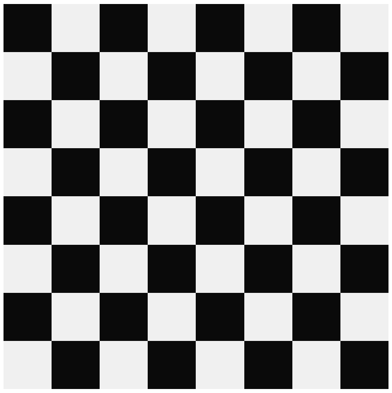

Figure 8.33: Original Focused Image

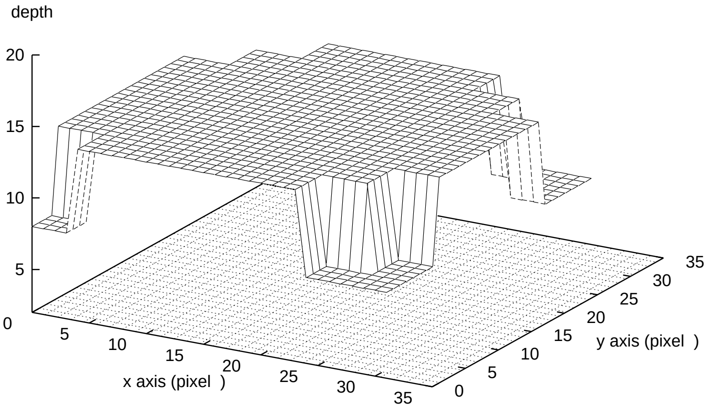

Figure 8.34: FIS by an IFA method

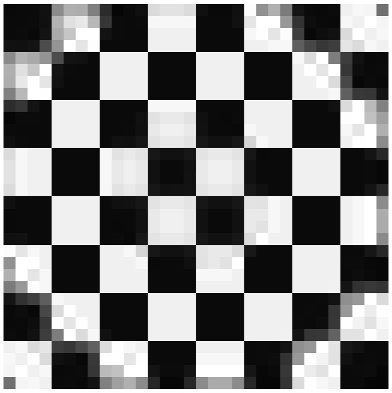

Figure 8.35: Focused Image from an IFA method

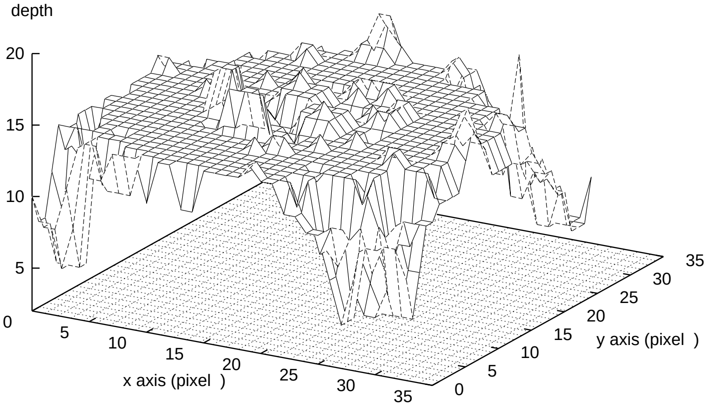

Figure 8.36: FIS by an IDA method (STM)

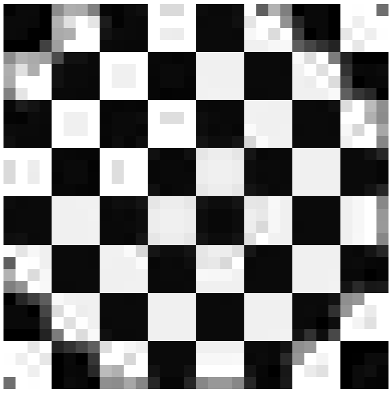

Figure 8.37: Focused Image from an IDA method (STM)

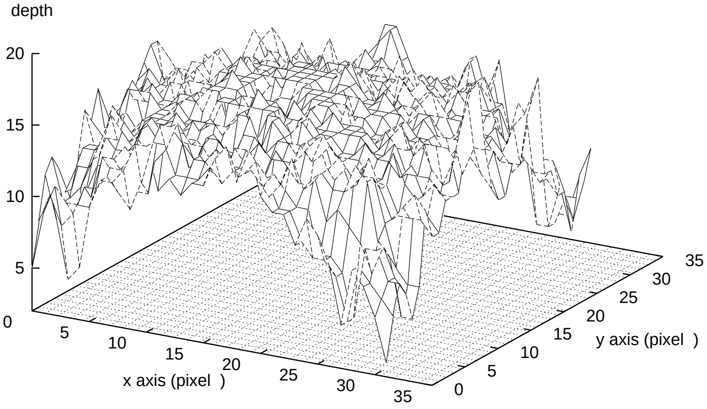

Figure 8.38: FIS by regularization(5 iters)

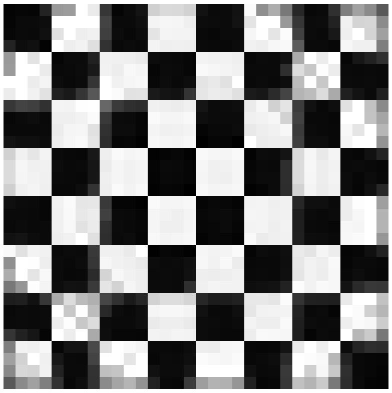

Figure 8.39: Focused Image by regularization(5 iters.)

Table 8.4: Error percentage of gray level per pixel for planar object

IFA	IFA and IDA	SPS + STM	SPS + Interpolation	SPS + Fourier method
13.8%	7.6%	6.6%	8.8%	12.1%

Table 8.5: Error percentage of gray level per pixel for Sphere object

IFA	IFA and IDA	Regu. + STM (5 iter.)	Regu. + Interpolation (5 iter.)
7.3%	6.7%	5.3%	11.5%

8.5 Conclusion

A computational algorithm for the Unified Focus and Defocus Analysis has been demonstrated with several experiments. This algorithm deals with variable number of input images in an optimal fashion. With this algorithm, UFDA is more efficient because it uses the image data in a better way and avoid unnecessary data acquisition and computation. Also, it can attain more accurate result since it process the most useful data and obtain the solution from the most suitable method for the situation. Experimental results show that estimating focused images from a small number of defocused images (the "deconvolution" problem) is difficult. Further research is needed to obtain a good solution to this problem. This will facilitate further improvements in the performance of UFDA. However, if the number of available images is not few but many, then UFDA provides good results useful in practical applications.

Chapter 9

Parallel Implementation of UFDA by Parallel Virtual Machine (PVM)

A unified approach to image focus and defocus analysis (UFDA) is proposed us for three-dimensional shape and focused image recovery of objects. One version of this approach which yields very accurate results is highly computationally intensive. In this chapter we present a parallel implementation of this version of UFDA on the Parallel Virtual Machine (PVM)[1]. One of the most computationally intensive parts of the UFDA approach is the estimation of image data that would be recorded by a camera for a given solution for 3D shape and focused image. This computational step has to be repeated once during each iteration of the optimization algorithm. Therefore this step has been sped up by using the Parallel Virtual Machine (PVM). PVM is a software package that allows a heterogeneous network of parallel and serial computers

[1] This research is a join work by Mr. Naiwei Lo, Dr. Muralidhara Subbarao, Dr. Bradely S. Carlson and author.

to appear as a single concurrent computational resource. In our experimental environment PVM is installed on four UNIX workstations communicating over Ethernet to exploit parallel processing capability. Experimental results show that the communication overhead in this case is relatively low. An average of 1.92 speedup is attained by the parallel UFDA algorithm running on 2 PVM connected computers compared to the execution time of sequential processing. By applying the UFDA algorithm on 4 PVM connected machines an average of 3.44 speedup is reached. This demonstrates a practical application of PVM to 3D machine vision.

9.1 Introduction

The recovery of three-dimensional (3D) scene information of objects from images is a problem of inverse optics. Methods to solve this problem normally involves intensive computation in modeling the image formation process. Developing a highly accurate model for the image formation process that requires a reasonable amount of computation is an important problem. One way to handle this problem is to use a parallel algorithm implemented on a parallel computer. In this paper we present a parallel implementation of the Unified Focus and Defocus Analysis (UFDA) [76] for 3D information recovery on a local area network (LAN). The communication interface between different machines is provided by the Parallel Virtual Machine (PVM) [73, 79]. PVM is a portable message-passing programming system designed to link separate host machines to form a "virtual machine".

UFDA was proposed by us recently [76] for accurate reconstruction of 3D shape and focused image from a sequence of defocused images. This technique unified two distinct approaches – Image Focus Analysis (IFA)[5, 30, 48] and Image Defocus Analysis (IDA)[3, 51]. The unification of IFA and IDA to UFDA is achieved by exploiting the relationship between the number of constraints embodied in the defocused images of these two approaches to the number of unknowns in the 3D shape and focused image. The framework of UFDA is based on modeling the sensing of defocused images in a camera system. A "Three-Dimensional Point Spread Function" (3D PSF) in the (x, y, d) space for this image formation model is introduced. Here x and y are the image spatial coordinates and d is a parameter representing the level of defocus. The problem of 3D shape and focused image reconstruction is formulated as an optimization problem where the difference (mean-square error) between the observed image data and the estimated image data is minimized. The estimated image data is obtained from the image sensing model and the current best known solutions to the 3D shape and focused image. An initial estimation to the solutions is obtained through traditional shape-from-focus methods. This solution is improved iteratively by a gradient descent approach. As we showed in [76] this is a very computationally intensive process, especially the part of estimating image data for a given solution to the 3D shape and focused image. This part has to be repeated to update the estimated image data during each iteration of the gradient descent search. One efficient optimization technique to UFDA has been proposed in [77]. This paper presents a parallel version of UFDA implemented on PVM. Our experiments show that

PVM provides a means to distribute heavy computation load of practical image recovery application on a LAN and improve the performance with respect to the total execution time.

9.2 3D Shape and Focused Image Recovery by UFDA

9.2.1 UFDA

UFDA [76] is a new theory that unifies IFA and IDA. To briefly illustrate IFA and IDA in terms of the definition of the problem for the 3D shape and focused image recovery, consider the following. The image coordinates of a point light source are denoted by (x', y') and the image coordinates of a point where the brightness is measured on the image detector are denoted by (x, y). The camera parameters are $\mathbf{e} = (D, f, s)$ where D is the diameter of camera aperture, f is the focal length of the lens, and s is the distance between the lens and the image detector. An image sequence can be thought of as sampled data of an image volume and denote the image data as $g(x, y, d)$ for $x = 0, 1, 2, \cdots, J-1$, $y = 0, 1, 2, \cdots, K-1$, $d = 0, 1, 2, \cdots, I-1$, where J and K are the number of rows and columns respectively in each image frame and I is the number of image frames. Whereas IFA uses many image frames, IDA uses only 2 or 3 images from this image sequence. $d_v(x', y')$ represents the 3D shape and $F(x', y')$ is the focused image. The problem for IFA can be stated as– given the image sequence $g(x, y, d)$ and the camera parameters s and f, find $d_v(x', y')$ and $F(x', y')$ in some image region or over the entire image. Similarly, the problem for IDA can be stated as– given two of the images $g_m(x, y)$ and $g_n(x, y)$ in the image sequence $g(x, y, d)$, the camera parameters $\mathbf{e}_m$ and $\mathbf{e}_n$ corresponding to these two images, and the camera's point spread function as a function of the camera parameters, find $d_v(x', y')$ and $F(x', y')$ in some image region or over the entire image.

The IFA and IDA methods can be viewed as two extremes of a range of methods useful in 3D shape and focused image recovery. At one end of this range of methods is the IFA method which uses a large amount of image data but minimal information about the camera characteristics (e.g. the camera's point spread function). At the other end is the IDA method which uses minimal image data but much information about the camera characteristics. The theory of UFDA results in a unified approach that suggests new methods that lie between the two extremes of the IFA and IDA methods. The unknowns in both the IFA and IDA methods are $d_v(x', y')$ and $F(x', y')$. If the image frame size is J rows and K columns, then the number of unknowns for $d_v(x', y')$ and $F(x', y')$ are both $J \times K$. Therefore the total number of unknowns is $2JK$. In UFDA, to determine these unknowns, a number of image frames $g(x, y, d)$ are recorded at different camera parameter settings. This recorded image volume data depends on the camera parameters $\mathbf{e}$, the 3D shape $d_v(x', y')$, and the focused image $F(x', y')$. This dependence is specified by the camera characteristics (e.g. point spread function) [47]. A three-dimensional point spread function (3D PSF) $h(x, y, \mathbf{e}, d)$ that is based on paraxial geometric optics is developed in [76].

The problem of 3D shape and focused image recovery in UFDA can be stated as – given a sequence of images obtained by sampling $g(x,y,d)$ and the image formation model, find $d_v(x',y')$ and $F(x',y')$.

Under certain weak conditions we can derive a three-dimensional convolution expression[76]:

$$g(x,y,d) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F'(x',y',d')\; h(x-x',y-y',d-d')\; dx'\; dy'\; dd'. \quad (9.1)$$

where

$$F'(x',y',d') = \begin{cases} F'(x',y') & \text{if } d' = d_v(x',y') \text{ and} \\ \\ 0 & \text{otherwise.} \end{cases} \quad (9.2)$$

The above convolution expression can be abbreviated as

$$g(x,y,d) \;=\; F'(x',y',d') \;\star\; h(x',y',d'), \quad (9.3)$$

where $\star$ denotes the convolution operator. Details on the 3D PSF and the derivation of the 3D convolution equation are reported in [76].

Here the problem of 3D shape and focused image reconstruction is formulated as an optimization problem where the difference or mean-square error E between the observed image data $g_o(x,y,d)$ and the estimated image data $g_e(x,y,d)$ is minimized.

$$E \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (g_o(x,y,d) - g_e(x,y,d))^2 \; dx \; dy \; dd \quad (9.4)$$

The estimated image data $g_e(x,y,d)$ is obtained from Eq. (9.1) using the 3D PSF h and the current best known solutions to the 3D shape $d_v(x',y')$

and focused image $F(x', y')$. An initial estimation of the solutions is obtained through traditional IDA and/or IFA methods. This solution is improved iteratively by an optimization technique.

9.2.2 Optimization

In the optimization a gradient descent approach is used to minimize the error E in Eq. (9.4). It is based on an iterative approach of going downhill with respect to the error function to find the lowest point. The focused image surface in a small image region is approximated by a piecewise planar surface patch with three parameters– slope with respect to x-axis, slope with respect to y-axis, and z-axis intercept. Error gradient with respect to these three parameters are used in the gradient descent error minimization. A sequential followed by a parallel parameter search (SPPS) is used for this method. First, the optimization is done with respect to one parameter at a time. After that, three parameters are searched simultaneously.

9.2.3 Sequential Implementation

Three image models ($16 \times 16 \times 16$, $32 \times 32 \times 32$ and $64 \times 64 \times 32$ size image volume data) are synthesized where the focused image surface of each model is a hemispherical object (with radius 12, 24, 24) and the focused image is a checker board. From UFDA, the observed image data $g(x, y, d)$ is synthesized using Eq. (9.1) with camera parameters (D=9mm, f=35mm, s= 35mm to 36.5mm). Then an initial solution for the focused image and the estimated

3D shape of these image models is obtained using an IFA method. In the IFA method a focus measure (energy of image Laplacian) is computed in a small window of size 4×4 (for $16 \times 16 \times 16$) or 8×8 (for $32 \times 32 \times 32$ and $64 \times 64 \times 32$) non-overlapping regions. A piecewise constant approximation to the focused image surface in each of the above window regions is obtained by finding the position where the focus measure is a maximum. These estimated solutions and the 3D PSF are used to compute the estimated image data $g_e(x, y, d)$ using Eq. (9.1). The error E between the observed image data $g_o(x, y, d)$ and the estimated image data $g_e(x, y, d)$ is computed using Eq. (9.4). The SPPS gradient descent method is applied to improve the estimated solution iteratively.

This UFDA approach reduces the errors in shape and focused image introduced by the image-overlap problem and the non-smoothness of the object's 3D shape. Experimental results are reported in [76] where accurate reconstructed 3D shape and focused image are obtained at the cost of computation.

9.3 Parallelizable Portion of UFDA

The highly intensive computation load of the UFDA approach is mainly due to the iterative estimation of image data in the image formation process using Eq. (9.1). The algorithm for convolving an estimated 3D object with 3D PSF is described below.

```
for $i = 1$ to $N$ do            /* $i$: frame index */
```

```
        for $j$ = 1 to $N$ do          /* $j$: row index */
          for $k$ = 1 to $N$ do          / * $k$: column index */
            $sum$= 0;
            for $m$ = 1 to $N$ do          /* $m$: row index */
              for $n$ = 1 to $N$ do          /* $n$: column index */
                $sum = sum +$
          $F(m,n)\cdot\ h(i - d_v(m,n), j - m + \frac{N}{2}, k - n + \frac{N}{2})$;
              end          /* $n$ loop */
            end          /* $m$ loop */
            if ($sum$>255)
              then $sum$ = 255;
            $g(i,j,k)$ =$sum$;
          end                    /* $k$ loop */
        end               /* $j$ loop */
      end          /* $i$ loop */
```

Here $g(i, j, k)$ (i.e. sum) denotes the gray level in the image volume, $h(i, j, k)$ is the 3D PSF, $d_v(m, n)$ is the focused image surface, $F(m, n)$ is the gray level on the $d_v(m, n)$, N represents the total number of image frames and image size for the corresponding index respectively. This algorithm explains explicitly the way we do convolution. From an accuracy point of view, this image sensing model takes into account the contribution of all object point sources in estimating the observed image brightness at a given point. In particular, the image brightness near the border of an image region is computed by taking into account all possible object point sources– both those inside the image region and those outside but close to the border. Therefore the error in 3D shape recovery is minimized. However, the price paid to minimize the error is a heavy computation workload. The computational complexity of this algorithm is $O(n^5)$ in each iteration. The solution is refined iteratively to improve accuracy. In the next section we consider data parallelization of the above algorithm.

9.4 Parallel Implementation of UFDA

In this section we describe the structure of Parallel Virtual Machine, the PVM environment in our laboratory, and implementation of the parallel UFDA algorithm.

Parallel Virtual Machine (PVM) [78], created in the summer of 1989 at Oak Ridge National Laboratory (ORNL), is a software system that permits a network of heterogeneous UNIX computers to be used as a single large memory-distributed parallel computer. Therefore, the computation power of a cluster of workstations/computers can be aggregated to solve large computational problems. In [72], the viability of network computing on a cluster of workstations for many large scientific applications has been established. In this paper we are interested in the viability of network computing for practical applications such as UFDA.

PVM provides the functions to start up tasks on the virtual machine and allows the tasks to communicate and synchronize with one another. A computation unit in PVM is defined as a *task* usually analogous to a UNIX process. Applications written in Fortran77 or C can be parallelized by using message-passing constructs. Thus, under careful design an application can run multiple tasks in parallel by sending and receiving messages between tasks (processes). In order to function correctly in a heterogeneous computer environment PVM handles all data conversion between computers with different data formats as necessary. PVM also allows application tasks to exploit the architecture best suited to their solution and provides heterogeneity at the application, machine, and network level.

The PVM system contains two parts – a daemon and an interface library. Any user with a valid login can install the daemon on a machine. When a user wants to run a PVM application, she/he first creates a virtual machine by starting up PVM. The PVM application can then be executed from a UNIX prompt on any of the hosts registered in the virtual machine environment. Multiusers and multitasking are supported in the PVM system. The PVM library includes user callable routines for message passing, spawning processes, coordinating tasks, and modifying the virtual machine. In order to use the PVM system all application programs must be linked with this library.

The PVM system adopts a two-process setting technique. The details are shown in Fig. 9.1. The PVM daemon process is responsible only for commu-

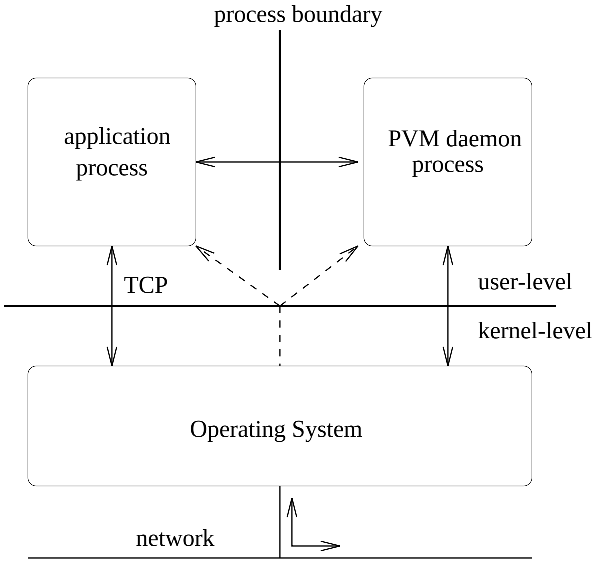

Figure 9.1: The structure of PVM system (data paths are indicated by solid arrow lines and scheduling control paths are marked by broken arrow lines).

nication. Scheduling of the application process and communication process is done by the operating system's scheduler. To exchange messages between two application processes, the data is copied across the process boundary into and out of the PVM daemon process. The messages are communicated across the network by means of the **UDP/IP** protocol. The PVM daemon process is responsible for retransmitting lost **UDP** packets and resolving reordering of packets to provide reliable communication. In recent versions of the PVM system, a direct **TCP** connection between two communicating application processes is established for message passing. This implementation reduces the total communication latency by a factor of 3-4. In [74, 75] new designs using multithread implementation to further hide the communication latency are discussed.

In our experiments two PVM environments are set for collecting related information. The first PVM environment is composed of two SUN SPARC IPX workstations with 16 and 32 megabyte main (RAM) memory on the two machines respectively. The second PVM environment connects four SUN SPARC 5 workstations with 64 megabyte RAM on each machine. Every computer has its own hard disk drive and both environments use 10 megabits/sec Ethernet to connect machines.

In the first environment the parallel programs include a master part and a slave part. The master program is implemented on the SUN SPARC IPX with 32 megabyte memory on board and the slave program is installed on the other computer. Once the PVM daemon processes on both machines are activated by executing the PVM console program from the master machine, the master parallel program is executed and its corresponding slave parallel program is invoked through the network automatically. All data communication work is handled by PVM connection function calls. For example, **pvm_send()** and **pvm_recv()** are the functions to send and receive data between two application processes respectively. To execute the parallel programs on a larger network, we apply a polling scheme to distribute equal amounts of workload from a master program/machine to other slave programs/machines.

To utilize the computation power of the local area network efficiently data communication overhead among computers must be reduced to a minimum extent. In our implementation of the parallel estimation algorithm, initial data is transmitted over the network from the master machine to the slave machine at the first iteration of estimation. Initial data includes estimated focus image, observed image volume, estimated 3D shape, and precomputed 3D PSF data. For the regular data communication only the current estimated 3D shape and the new estimated image volume are transmitted back and forth at each iteration of estimation.

To get significant performance improvement from parallel execution the computation complexity of parallelized algorithm (process) should be at least two orders of magnitude greater than the complexity of communication overhead. From section 9.3 we know the computation complexity of the sequential estimation process is $O(n^5)$ where n is the corresponding image size. The complexity of communication overhead, $O(n^3)$, is bounded by the transmission of new estimated image volume at each iteration. Therefore, we predict that the overall execution speed of our application can be improved by using a parallel implementation.

9.5 Experimental Results

The first important issue for a parallel application/experiment is to generate correct results as its sequential counterpart does. In our UFDA experiments the output results from both parallel and sequential versions are identical. In the $32 \times 32 \times 32$ image volume case the original focused image surface (FIS), the initial solution obtained from the IFA method, and the recovered focused image surface are shown in Figs. 9.2, 9.3 and 9.4, respectively.

In order to eliminate the performance inconsistency and degradation of both parallel and sequential UFDA applications caused by the dynamic net-

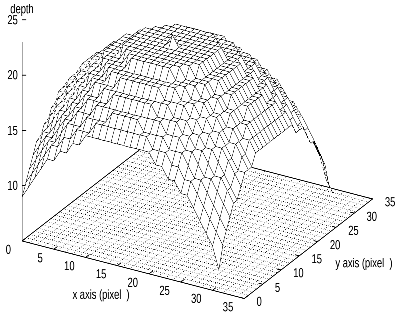

Figure 9.2: The original 32 × 32 × 32 focused image surface (FIS).

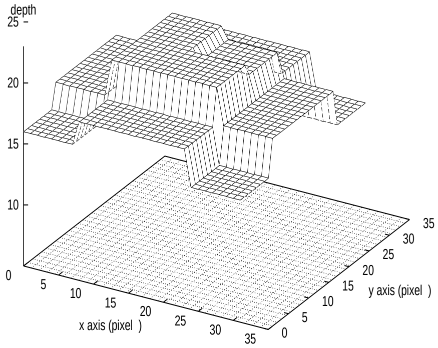

Figure 9.3: The initial solution of $32 \times 32 \times 32$ focused image surface generated by the IFA method.

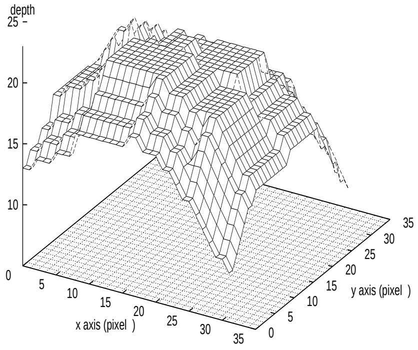

Figure 9.4: The recovered 32 × 32 × 32 focused image surface generated by parallelized UFDA.

work workload, all performance measurements in our experiments are recorded with only the UFDA user process(es) and essential system daemon processes running on the workstation(s) of the network.

In Table 1 the performance details of sequential UFDA under different sizes of image volume are presented. Both SPARC IPX and SPARC 5 computer environments are used to execute the sequential UFDA program to get the performance information as reference values for the parallel version. Notice that the total execution time is equal to the user time plus the system time. For the parallel UFDA the corresponding results measured on the master machine are shown in Tables 2, 3 and 4. In parallel UFDA the total execution time is composed of user time, system time, and communication overhead. The user time is defined as the execution time spent by the user process. The definition of system time is the execution time spent by the system (UNIX kernel). The communication overhead is caused by the internal data processing delay of ethernet cards (data copy from kernel to card memory) and the transmission latency of the local network. Because the shared data between machines has to be processed before sending and after receiving it, a portion of the measured user time of the parallel UFDA algorithm is contributed by the added data preprocessing and postprocessing programs. Measured entries in all tables are reported in seconds.

Compared to Table 1 the values of user time in Table 2 are lower because the computation workload is shared by two workstations. On the contrary the values of system time in Table 2 are greater than the ones in Table 1. The reason is that PVM procedure calls invoke some kernel functions. Therefore, the system time increases when more PVM procedure calls are executed.

To describe the speed advantage of a parallel algorithm compared to a serial reference algorithm the speedup ratio must be defined. Let m represent the problem size (that is, the size of image volume in our case). Suppose that we have a parallel algorithm that uses r processors that terminates in time $T_r(m)$. Let $T(m)$ be the time required by the serial (uniprocessor) reference algorithm for this problem. The speedup of the parallel algorithm is defined as $S_r(m) = \frac{T(m)}{T_r(m)}$. The speedup ratios of the parallel UFDA algorithm varied from 1.49 to 1.97 for 2 networked computers, and from 2.17 to 3.56 for 4 networked computers as shown in Fig. 9.5. The speedup diagram indicates that if the image volume is large enough the communication overhead is not the dominant factor of the total execution time. The percentage of total ex-

Table 9.1: Measured execution times for sequential UFDA.

Computer Model	Image Volume	Total Execution Time	User Time	System Time
SPARC IPX	$16 \times 16 \times 16$	70.6	70.4	0.2
SPARC IPX	$32 \times 32 \times 32$	2345.2	2344.4	0.8
SPARC IPX	$64 \times 64 \times 32$	35421.5	35406.0	15.5
SPARC 5	$16 \times 16 \times 16$	26.5	26.5	0.0
SPARC 5	$32 \times 32 \times 32$	895.3	895.1	0.2
SPARC 5	$64 \times 64 \times 32$	13661.5	13661.0	0.5

Table 9.2: Measured execution times for parallel Implementation using Two SPARC IPXs.

Image Volume	Total Exe. Time	User Time	System Time	Comm. Overhead
$16 \times 16 \times 16$	41.5	36.6	0.7	4.2
$32 \times 32 \times 32$	1242.0	1168.8	2.9	70.3
$64 \times 64 \times 32$	18009.2	17784.8	21.1	203.3

Table 9.3: Measured execution times for parallel Implementation using Two SPARC 5s.

Image Volume	Total Exe. Time	User Time	System Time	Comm. Overhead
$16 \times 16 \times 16$	17.8	15.0	0.3	2.5
$32 \times 32 \times 32$	479.5	441.3	1.2	37.0
$64 \times 64 \times 32$	6956.8	6787.7	7.3	161.8

Table 9.4: Measured execution times for parallel Implementation using Four SPARC 5s.

Image Volume	Total Exe. Time	User Time	System Time	Comm. Overhead
$16 \times 16 \times 16$	12.2	9.0	1.1	2.1
$32 \times 32 \times 32$	270.1	242.4	2.3	25.4
$64 \times 64 \times 32$	3835.2	3698	18.6	118.6

ecution time spent on communication overhead under different image models presented in Fig. 9.6 also reveals the same result. The communication overhead is significant in $16\times16\times16$ image model case of parallel UFDA algorithm because the total computation workload is not large enough to hide the communication latency caused by sending initial data from the master machine to the slave machine. However, since the computation complexity of the parallel UFDA algorithm ($O(n^5)$) is two orders of magnitude greater than its corresponding communication complexity ($O(n^3)$), the communication overhead of initial data transmission between two workstations is shared and hidden by iterative computation for large image volumes.

In Fig. 9.5 we notice that the speedup performance with 2 SPARC IPXs is higher than the one with 2 SPARC 5 machines. The reason is that the

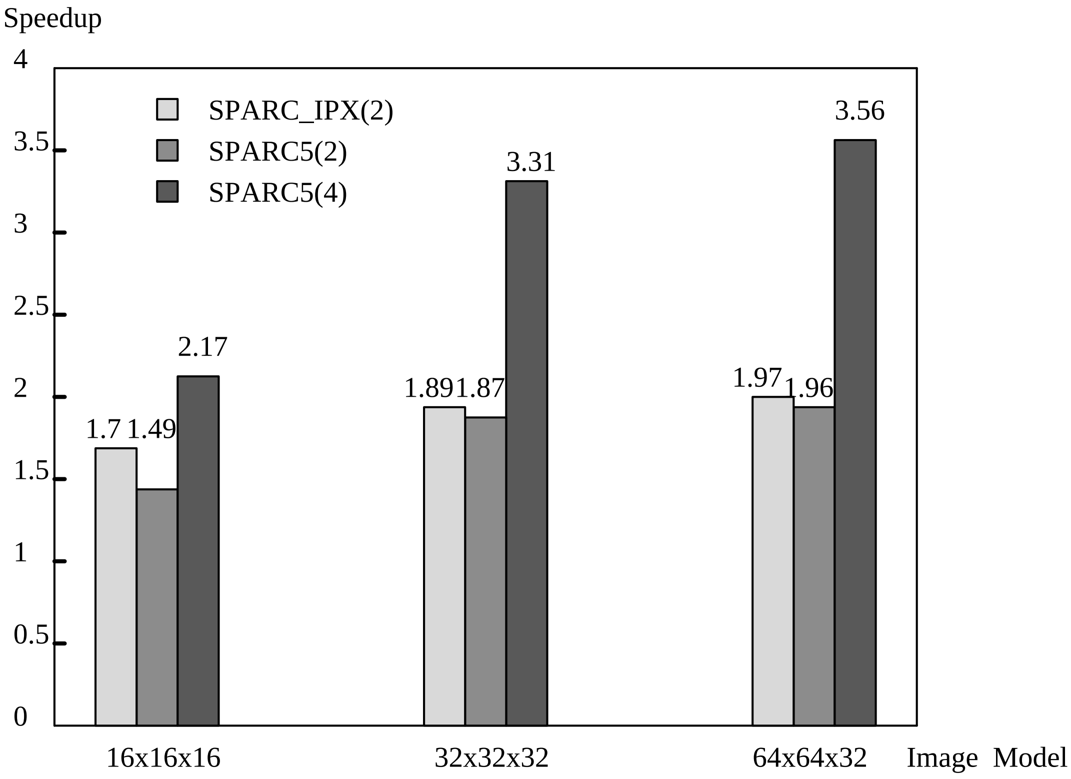

Figure 9.5: Speedup analysis for different image models.

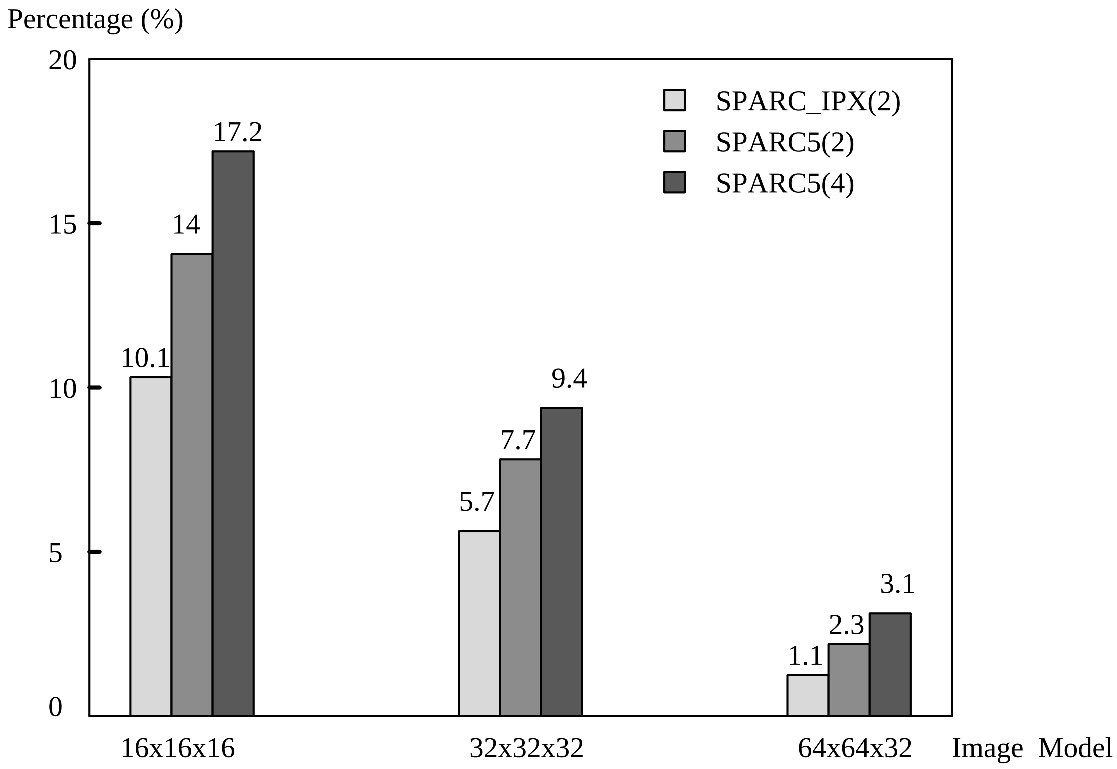

Figure 9.6: The percentage of total execution time spent on communication overhead in different image models.

computation capability of SPARC 5 is better than SPARC IPX. Therefore, for a given amount of data to be processed and a given network environment (10 megabits/sec ethernet) SPARC 5 spends less time to complete the data computation but about the same amount of time as SPARC IPX to transmit data between machines. Therefore, the speedup ratio of SPARC 5 model is lower than the SPARC IPX model. As the image size increases, the time spent on data computation grows faster than the communication overhead, and therefore the speedup of SPARC 5 model catches up with the speedup of SPARC IPX model.

From Tables 2 and 3 we observe that the communication overhead of SPARC 5 model is less than the one for SPARC IPX model. Again, this is because the SPARC 5 model is faster and has more main memory. Therefore the data copying from kernel memory to ethernet card takes less time in SPARC 5 than in SPARC IPX. Since the data transmission time is the same for both models, the communication overhead of SPARC 5 model must be less than for SPARC IPX.

From Tables 3 and 4 we observe that the communication overhead of 4 SPARC 5s is less than that of 2 SPARC 5s. This result is counter-intuitive, but correct. Recall that we define the communication overhead as the internal data processing delay of ethernet cards and the transmission latency of the local network. Assume that the data transmitted between the master and the slave have the same size. Let T_c be the data processing delay of ethernet card, T_t be the transmission latency of the local network, and T_p be the data computation time of the master machine in a PVM environment with 4 computers networked. Because T_c, T_t and T_p are all proportional to the amount of data (image size), we can easily estimate their values when only one or two processors are used.

Shown in Fig. 9.7 is the time-line comparison of an application execution with different numbers of processors. In our PVM environment the data distribution from the master machine to the slave machines uses a polling scheme. In this scheme the master machine first copies data from its kernel memory to the ethernet card buffer which is then transmitted to the slave machine over the network. While the ethernet card is sending data to one of the slave machines from its buffer over the network, the master machine can simultaneously copy the data into the ethernet card buffer for another slave machine for transmission. A similar overlap of copying and transmission occurs when the master machine receives data back from the slave machine. The communication overhead of 4 processors is $4T_c + 2T_t$ and the communication overhead of 2 processors is $4T_c + 4T_t$. Therefore, the communication overhead of 4 SPARC 5s is less than that of 2 SPARC 5s. If we assume that the sizes of the sent and received data are different, then we have two different data processing delays, T_{c1} and T_{c2} and two different transmission latencies, T_{t1} and T_{t2}. We can derive that if $T_{c2} + T_{t1} + T_{t2} > T_{c1}$, then the conclusion is the same. In our experiments the size of shared data at each iteration is small (16k bytes or less). T_{c1} is about the same value as T_{c2}, and $T_{t1} \approx T_{t2}$.

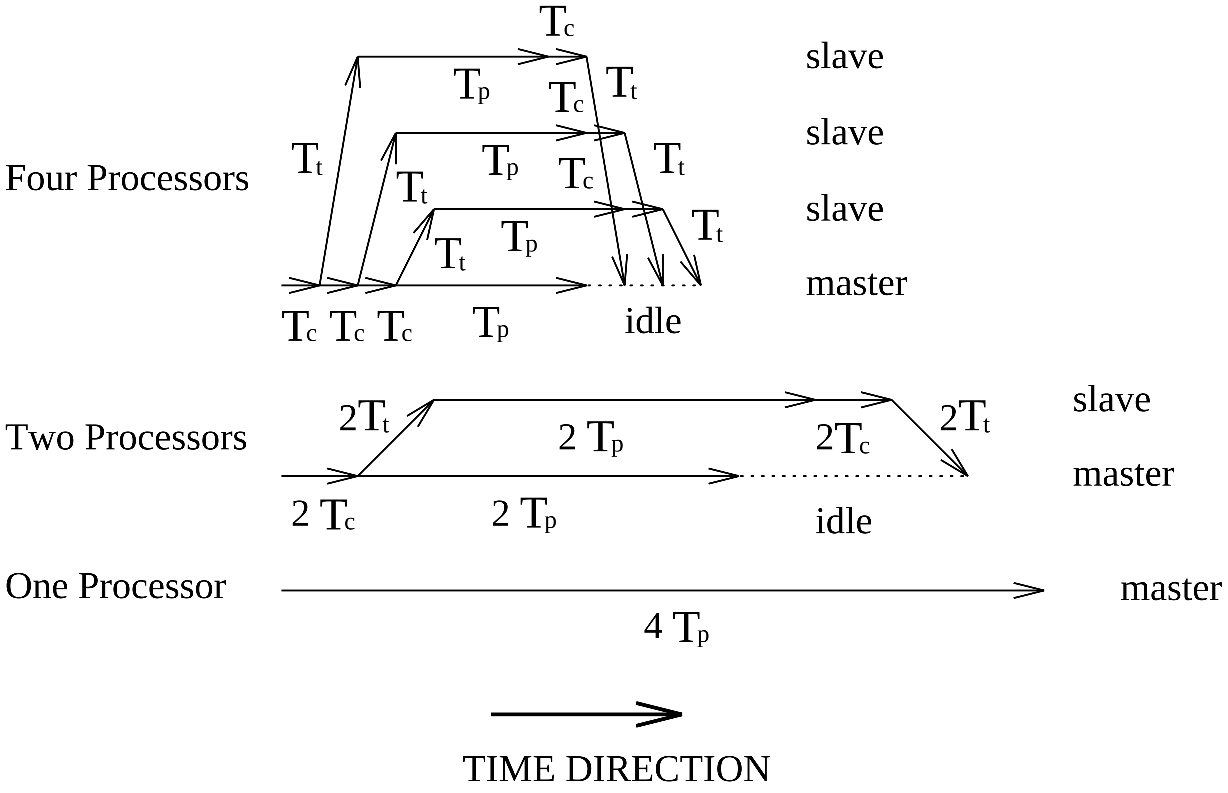

Figure 9.7: The time-line comparison of an application execution with different number of processors.

9.6 Conclusion

In this paper a parallel implementation of the UFDA using PVM is presented. One important characteristic of UFDA is the trade-off between the accuracy of 3D shape and focused image recovery and the computation load. Higher accuracy is obtained at the cost of additional image acquisition and processing. PVM helps to speed up the most computationally intensive part of UFDA– the computation of estimated image data for a given solution.

Parallization of UFDA through PVM offers a solution to reduce the total execution time of the UFDA application and provides an example of utilizing the distributed computation power of a local area network. In our experiments with two and four computers networked the parallel UFDA application depending on the size of image data achieves a speedup in the range of 1.49 to 1.97 and 2.17 to 3.56, respectively. Our experiments demonstrate the advantage of using PVM to solve computation intensive problems in machine vision applications such as UFDA.

Chapter 10

Conclusion

10.1 Summary

In this dissertation, we present a new theory that unifies Image Focus Analysis (IFA) and Image Defocus Analysis (IDA). IFA methods are based on computing focus measure to search the most focused image and depth information . During this search, no information about camera's point spread function is required. One of the advantages of IFA is that it does not suffer from the correspondence and occlusion problem associated with methods like stereo. Also, it can be implemented on a single image system which facilitates its applications to a variety of systems. On the other hand, for this type of method to get a good result, it has to process a large number of images in order to cover the range of focus. This leads to a long time in the acquisition of image data due to the mechanical motion of camera. In addition, due to the specs of some applications, it is possible that there are not enough image data for IFA to obtain a reasonable result.

IDA methods, in contrast to IFA, are based on computing Fourier or Spatial domain convolution/deconvolution transform. They can use as little as two image at different camera parameter settings to find the relations between the 3D scene and the degree of blur in determine the depth and focused image for entire scene. This process involves the knowledge of very accurate camera's point spread function to make this method feasible. Since IDA needs only two images and the object are not necessary in focus. It alleviates the limitations due to the available images and data acquisition for the IFA. However, the IDA method requires accurate camera calibration for the information of point spread function.

Based on these distinctions of IFA and IDA, it is necessary to devise a method that incorporates the advantages from both methods and then able to make an adjustment by itself depends on the requirements of applications. This consideration leads us to the new theory and an unified approach to the image focus and defocus analysis (UFDA). The theory of UFDA is formed by linking the number of unknowns and constraints in the image data for IFA and IDA respectively. The unified approach is then built up based on modeling the sensing of defocused in camera system. In the proposed image formation model, a 3D PSF is used here for accurately imitate the actual response of a real camera system. The goal here is to obtain a better solution compared to IFA or IDA methods. This is attained by using a computational algorithm for the UFDA which processes different number of input images in an optimal fashion to improve accuracy and computational efficiency. Besides, the 3D shape and focused image recovery problems formulated as an optimization problem where the difference between the acquired image data and the estimated image data is minimized. Here, several optimization techniques that aim on this particular problem have been investigated. The analysis of these approaches for UFDA in different aspects depending on their characteristics and the smoothness of the object are presented. Also, a parallel implementation of UFDA by PVM has be conducted. This experiments give us a indication of when the computing resource can be fully utilized then the UFDA can obtain a very accurate solution in a desired time.

Our work so far on UFDA provides a reasonably complete theoretical and implementation framework for application of UFDA in practical machine vision. Experimental results shown UFDA offers trade-off between (i) the number of image frames used, (ii) the accuracy of results, (iii) the amount of computation used, and (iv) the knowledge of known camera characteristics.

10.2 Future Research

Analysis of defocused image data for 3D shape and focused image recovery is our primary goal. A major topic of our future research is the development of computationally efficient and accurate techniques for solving the optimization problem in UFDA. For this purpose, this research can be further in following ways.

First, a dynamic local error minimization scheme can be applied to obtain a accurate solution while reduce the computation time. To perform this scheme, the observed image data are only acquired at a certain limited blur range with respect to the estimated solution at a small image region. Since the image data that acquired at large blur area will only make marginal effect for the total local error minimization. Hence, using such scheme, we should able to improve the computation efficiency dramatically without sacrifice the accuracy. This method can be applied to practical application like microscopy. In microscopy, 3D shape and focused image are recovered for very small object like Ball Grid Array (BGA). In this application, the desired image is first divide into small regions to perform IFA. When IFA is applied to the image frames to produce a initial solution, UFDA with the dynamic local error minimization scheme can then be applied to reconstruct the shape and focused image.

Another method that possibly to try in the future is the multiresolution IFA method or a combination of multiresolution IFA with above dynamic local error minimization. The multiresolution is achieved by applying the IFA method at a certain range based on previous estimated solution to obtain the initial solution progressively. The advantage of this method is that it can reduce the computation cost with the processing of most useful data.

The other direction for future research is the investigation of advanced optimization techniques for UFDA to further improve the accuracy and reduce computation. Methods using probability principles like Bayesian estimation or simulation annealing can probably make improvement using different aspects of the structure of UFDA compared to the regularization that we have tried.

And finally, a possible exploration of 3D Fourier Analysis of defocused image data as in 3D microscopy. Since the traditional approach in 3D microscopy differs from ours in terms of problem formulation and parameterization of defocus.

Appendix A

Derivation of Euler-Lagrange Equation

proof the Euler-Lagrange equation of

$$\mathcal{I} \; = \; \int_{x_1}^{x_2} \int_{y_1}^{y2} \mathcal{F}(x, y, f, f_{xx}, f_{yy}) \; dx \; dy \tag{A.1}$$

is

$$\mathcal{F}_f + \frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}} + \frac{\partial^2}{\partial y^2}\mathcal{F}_{f_{yy}} \; = \; 0 \tag{A.2}$$

(Proof :)

Functional $\mathcal{F}$ depends on the unknow functions f, f_{xx} and f_{yy}.

Assume the surface pass through (x_1, y_1) , (x_1, y_2) , (x_2, y_1) , (x_2, y_2).

$\eta(x, y)$ is a test function and ϵ be a small value.

By definition [1], function f has continuous derivative up to order 1. So we can write following boundary conditions.

- $f(x_1, y) = \vec{z_1}(x_1, y)$; $\vec{z_1}(x_1, y)$ is a set of the values for each (x, y) along the line from (x_1, y_1) to (x_1, y_2) for $y_1 \leq y \leq y_2$.

- $f(x_2, y) = \vec{z_2}(x_2, y)$; $\vec{z_2}(x_2, y)$ is a set of the values for each (x, y) along the line from (x_2, y_1) to (x_2, y_2) for $y_1 \leq y \leq y_2$.

- $f(x, y_1) = \vec{z_3}(x, y_1)$; $\vec{z_3}(x, y_1)$ is a set of the values for each (x, y) along the line from (x_1, y_1) to (x_2, y_1) for $x_1 \leq x \leq x_2$.
- $f(x, y_2) = \vec{z_4}(x, y_2)$; $\vec{z_4}(x, y_2)$ is a set of the values for each (x, y) along the line from (x_1, y_2) to (x_2, y_2) for $x_1 \leq x \leq x_2$.
- $f_x(x_1, y) = \vec{z_5}(x_1, y)$; $\vec{z_5}(x_1, y)$ is a set of the values for each (x, y) along the line from (x_1, y_1) to (x_1, y_2) for $y_1 \leq y \leq y_2$.
- $f_x(x_2, y) = \vec{z_6}(x_2, y)$; $\vec{z_6}(x_2, y)$ is a set of the values for each (x, y) along the line from (x_2, y_1) to (x_2, y_2) for $y_1 \leq y \leq y_2$.
- $f_x(x, y_1) = \vec{z_7}(x, y_1)$; $\vec{z_7}(x, y_1)$ is a set of the values for each (x, y) along the line from (x_1, y_1) to (x_2, y_1) for $x_1 \leq x \leq x_2$.
- $f_x(x, y_2) = \vec{z_8}(x, y_2)$; $\vec{z_8}(x, y_2)$ is a set of the values for each (x, y) along the line from (x_1, y_2) to (x_2, y_2) for $x_1 \leq x \leq x_2$.
- $f_y(x_1, y) = \vec{z_5}(x_1, y)$; $\vec{z_5}(x_1, y)$ is a set of the values for each (x, y) along the line from (x_1, y_1) to (x_1, y_2) for $y_1 \leq y \leq y_2$.
- $f_y(x_2, y) = \vec{z_6}(x_2, y)$; $\vec{z_6}(x_2, y)$ is a set of the values for each (x, y) along the line from (x_2, y_1) to (x_2, y_2) for $y_1 \leq y \leq y_2$.
- $f_y(x, y_1) = \vec{z_7}(x, y_1)$; $\vec{z_7}(x, y_1)$ is a set of the values for each (x, y) along the line from (x_1, y_1) to (x_2, y_1) for $x_1 \leq x \leq x_2$.
- $f_y(x, y_2) = \vec{z_8}(x, y_2)$; $\vec{z_8}(x, y_2)$ is a set of the values for each (x, y) along the line from (x_1, y_2) to (x_2, y_2) for $x_1 \leq x \leq x_2$.

By above B.C.s , we can write

$$\begin{cases} \eta(x_1, y) = 0, & y_1 \leq y \leq y_2. \\ \eta(x_2, y) = 0, & y_1 \leq y \leq y_2. \\ \eta(x, y_1) = 0, & x_1 \leq x \leq x_2. \\ \eta(x, y_2) = 0, & x_1 \leq x \leq x_2. \end{cases}$$

$$\begin{cases} \eta_x(x_1, y) = 0, & y_1 \leq y \leq y_2. \\ \eta_x(x_2, y) = 0, & y_1 \leq y \leq y_2. \\ \eta_x(x, y_1) = 0, & x_1 \leq x \leq x_2. \\ \eta_x(x, y_2) = 0, & x_1 \leq x \leq x_2. \end{cases}$$

$$\begin{cases} \eta_y(x_1, y) = 0, & y_1 \leq y \leq y_2. \\ \eta_y(x_2, y) = 0, & y_1 \leq y \leq y_2. \\ \eta_y(x, y_1) = 0, & x_1 \leq x \leq x_2. \\ \eta_y(x, y_2) = 0, & x_1 \leq x \leq x_2. \end{cases}$$

From Taylor series theorem

$$\begin{aligned} \mathcal{F}(x, y, f + \epsilon\eta(x, y), f_{xx} + \epsilon\eta_{xx}(x, y), f_{yy} + \epsilon\eta_{yy}(x, y)) \;&=\; \mathcal{F}(x, y, f, f_{xx}, f_{yy}) + \\ \epsilon\eta(x, y)\frac{\partial\mathcal{F}(x, y, f, f_{xx}, f_{yy})}{\partial f} + \epsilon\eta_{xx}(x, y)&\frac{\partial\mathcal{F}(x, y, f, f_{xx}, f_{yy})}{\partial f_{xx}} + \epsilon\eta_{yy}(x, y)\frac{\partial\mathcal{F}(x, y, f, f_{xx}, f_{yy})}{\partial f_{yy}} \\ +e(higher\ \ order\ \ terms) \end{aligned} \tag{A.3}$$

Plug Eqn.25 into Eqn.23 , $\mathcal{I}$ can be rewritten as

$$\mathcal{I} \;=\; \int_{x_1}^{x_2}\int_{y_1}^{y2}\{\mathcal{F}(x, y, f, f_{xx}, f_{yy}) + \epsilon\eta(x, y)\mathcal{F}_f + \epsilon\eta_{xx}(x, y)\mathcal{F}_{f_{xx}} + \epsilon\eta_{yy}(x, y)\mathcal{F}_{f_{yy}} + e\}\ dx\ dy \tag{A.4}$$

By

$$\frac{\partial\mathcal{I}}{\partial\epsilon}\ |_{\epsilon=0} = 0 \tag{A.5}$$

we get

$$\int_{x_1}^{x_2}\int_{y_1}^{y2}\{\underbrace{\eta(x, y)\mathcal{F}_f}_{=x.1} + \underbrace{\eta_{xx}(x, y)\mathcal{F}_{f_{xx}}}_{=x.2} + \underbrace{\eta_{yy}(x, y)\mathcal{F}_{f_{yy}}}_{=x.3}\}\ dx\ dy \;=\; 0 \tag{A.6}$$

Use integration by parts

x.2 ⇒

$$\begin{aligned}
&\int_{x_1}^{x_2}\int_{y_1}^{y2}\eta_{xx}(x,y)\mathcal{F}_{f_{xx}}\ dx\ dy \qquad \text{(A.7)}\\
&= \int_{y_1}^{y2}\{\eta_x\mathcal{F}_{f_{xx}}\ |_{x_1}^{x_2} - \eta\frac{\partial}{\partial x}\mathcal{F}_{f_{xx}}\ |_{x_1}^{x_2} + \int_{x_1}^{x_2}\eta\frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}}\ dx\}\ dy\\
&= \int_{y_1}^{y2}\{\eta_x(x_2,y)\mathcal{F}_{f_{xx}} - \eta_x(x_1,y)\mathcal{F}_{f_{xx}}\}\ dy - \int_{y_1}^{y2}\{\eta(x_2,y)\frac{\partial}{\partial x}\mathcal{F}_{f_{xx}} - \eta(x_1,y)\frac{\partial}{\partial x}\mathcal{F}_{f_{xx}}\}\ dy +\\
&\quad \int_{y_1}^{y2}\int_{x_1}^{x_2}\eta(x,y)\frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}}\ dx\ dy
\end{aligned}$$

Further expand Eqn. x.2 and applied B.C. ⇒

$$\begin{aligned}
&= \underbrace{\int_{y_1}^{y2}\eta_x(x_2,y)\mathcal{F}_{f_{xx}}dy}_{=0} - \underbrace{\int_{y_1}^{y2}\eta_x(x_1,y)\mathcal{F}_{f_{xx}}dy}_{=0} - \underbrace{\int_{y_1}^{y2}\eta(x_2,y)\frac{\partial}{\partial x}\mathcal{F}_{f_{xx}}\ dy}_{=0} + \qquad \text{(A.8)}\\
&\quad \underbrace{\int_{y_1}^{y2}\eta(x_1,y)\frac{\partial}{\partial x}\mathcal{F}_{f_{xx}}\ dy}_{=0} + \int_{y_1}^{y2}\int_{x_1}^{x_2}\eta(x,y)\frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}}\ dx\ dy\\
&= \int_{y_1}^{y2}\int_{x_1}^{x_2}\eta(x,y)\frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}}\ dx\ dy
\end{aligned}$$

Similarly to Eqn. x.3 ⇒ we get

$$\begin{aligned}
&\int_{y_1}^{y2}\int_{x_1}^{x_2}\eta_{yy}(x,y)\mathcal{F}_{f_{yy}}\ dy\ dx \qquad \text{(A.9)}\\
&= \int_{x_1}^{x_2}\int_{y_1}^{y2}\eta(x,y)\frac{\partial^2}{\partial y^2}\mathcal{F}_{f_{yy}}\ dx\ dy
\end{aligned}$$

So Eqn 28 becomes

$$\int_{x_1}^{x_2}\int_{y_1}^{y2}\{\eta(x,y)\mathcal{F}_f + \eta(x,y)\frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}} + \eta(x,y)\frac{\partial^2}{\partial y^2}\mathcal{F}_{f_{yy}}\}\ dy\ dx = 0 \qquad \text{(A.10)}$$

Reorganize it, we get

$$\int_{x_1}^{x_2}\int_{y_1}^{y2}\eta(x,y)\{\mathcal{F}_f + \frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}} + \frac{\partial^2}{\partial y^2}\mathcal{F}_{f_{yy}}\}\ dy\ dx = 0 \qquad \text{(A.11)}$$

Since it must satisfy all the test function $\eta(x, y)$, so the desired Euler-Lagrange equation is

$$\mathcal{F}_f + \frac{\partial^2}{\partial x^2}\mathcal{F}_{f_{xx}} + \frac{\partial^2}{\partial y^2}\mathcal{F}_{f_{yy}} \;=\; 0 \qquad (A.12)$$

Bibliography

[1] N.I. Akhiezer, *The Calculus of Variations,* Blaisdell Publishing Company, pp104-109, First Edition , 1962.

[2] J.L. Buchanan and P.R. Turner. *Numerical Methods and Analysis*, McGraw-Hill, 1992

[3] J. Enns and P. Lawrence, "A Matrix Based Method for Determining Depth from Focus," *Proceedings of the IEEE Computer Society Conference on CVPR*, June 1991.

[4] B. K. P. Horn, *Robot Vision,* McGraw-Hill Book Company, 1986.

[5] E. Krotkov, "Focusing," *International Journal of Computer Vision*, 1, 223-237, 1987.

[6] S. K. Nayar and Y. Nakagawa, "Shape from Focus: An Effective Approach for Rough Surfaces," *IEEE Trans. on PAMI*, 16 (8):824-831, Aug. 1994.

[7] 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).

[8] M. Born and E. Wolf, *Principles of Optics*, Pergamon Press, Oxford, Sixth Edition, 1980.

[9] V. M. Bove, Jr., Entropy-based depth from focus. *Journal of Optical Society of America A,* 10:561-566, April 1993.

[10] C. H. Chen, “The SparcSupport Dual Motor Controller User Manual”, Dec. 1992, Report submitted as part of a undergraduate research project, Computer Vision Laboratory, Dept. of Electrical Engg., SUNY, Stony Brook, NY 11794-2350.

[11] T. S. Choi, *Shape and Image Reconstruction from Focus* , Ph.D. Thesis, Dept. of Electrical Engineering, State University of New York, Stony Brook, Dec. 1993.

[12] J. Ens and P. Lawrence, “A Matrix Based Method for Determining Depth from Focus”, *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1991.

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

[14] P. Grossman, “Depth from Focus”, *Pattern Recognition Letters* 5, pp. 63–69, Jan. 1987.

[15] Grimson, W.E.L. *From image to surfaces: A Computational Study of the Human Early Vision System,* MIT, Cambridge, 1981

[16] R.M. Haralick and L.G. Shapiro,“Computer and Robot Vision”, Addison-Wesley Publishing Co., Inc, 1992

[17] R.W. Hamming , “Introduction to Applied Numerical Analysis”, Mcgraw-Hill book company, 1971

[18] E. Hecht, *Optics,* Addison-Wesley Publishing Co., 1987.

[19] B. K. P. Horn, “Focusing”, Artificial Intelligence Memo No. 160, MIT, 1968.

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

[21] B. Jahne, “Digital Image Processing“ Springer-Verlag.

[22] J.K. Tyan, *Analysis and Application of Autofocusing and Three-Dimensional Shape Recovery Techniques based on Image Focus and Defocus* , Ph.D. Thesis, Dept. of Electrical Engineering, State University of New York, Stony Brook, Dec. 1997.

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

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

[25] S. H. Lai, C. W. Fu, and S. Chang, “A generalized depth estimation algorithm with a single image”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-14, No. 4, pp. 405–411, April 1992.

[26] G. Ligthart, and F. C. A. Groen, *A comparison of different autofocus algorithms,* Proceedings of the International Conference on Pattern Recognition, 1982.

[27] M. C. Lu, *Computer Modeling and Simulation for Computer Vision Problems*, Ph.D. Thesis, Dept. of Electrical Engineering, State University of New York, Stony Brook, May 1993.

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

[29] S. K. Nayar, M. Watanabe, and M. Noguchi, "Real-time focus range sensor", *Proc. of Int. Conf. on Computer Vision,* pp. 995-1001, June 1995.

[30] S. K. Nayar and Y. Nakagawa, "Shape from Focus: An Effective Approach for Rough Surfaces," *IEEE Trans. on PAMI*, 16 (8):824-831, Aug. 1994.

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

[32] A. Papoulis, *The Fourier Integral and its Applications*, McGraw-Hill, 1962.

[33] A. Papoulis, *Systems and Transforms with Applications in Optics*, McGraw-Hill, 1968, page 30.

[34] A. P. Pentland, "A New Sense for Depth of Field", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 4, pp. 523–531.

[35] A. P. Pentland, "A Simple Real-time Range Camera", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Diego, California, June 1989.

[36] T. Poggio, V. Torre, and C. Koch, *Computational Vision and Regularization Theory,* Nature, Vol. 317, No. 6035, pp. 314-319. 1985 (September).

[37] A. Rosenfeld, and A. C. Kak, *Digital Picture Processing,* Vol. I . Academic Press, 1982.

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

[39] J. Stoer and R. Bulirsch, "Introduction to Numerical Analysis", Springer-Verlag., New York , 1986

[40] M. Subbarao, *Direct recovery of depth-map I: differential methods,* IEEE computer society workshop on computer vision, Miami Beach, Dec. 1987, pp. 58–65.

[41] M. Subbarao, "Parallel Depth Recovery by Changing Camera Parameters", *Second International Conference on Computer Vision,* Florida, USA, pp. 149-155, December 1988.

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

[43] M. Subbarao, "Spatial-Domain Convolution/Deconvolution Transform ", Tech. Report No. 91.07.03, Computer Vision Laboratory, Dept. of Electrical Engineering, State University of New York, Stony Brook, NY 11794-2350.

[44] M. Subbarao, and G. Natarajan, "Depth Recovery from Blurred Edges", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, pp. 498-503, June 1988.

[45] M. Subbarao, and T. Wei, "Depth from Defocus and Rapid Autofocusing : A practical Approach", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, June 1992, pp. 773-776.

[46] M. Subbarao, T. Wei, and G. Surya, "Focused Image Recovery from Two Defocused Images Recorded with Different Camera Settings", *IEEE Transactions on Image Processing*, Vol. 4, No. 12, 16 pages, Dec. 1995.

[47] M. Subbarao, and M. C. Lu, "Computer Modeling and Simulation of Camera Defocus", *Machine Vision and Applications*, (1994) 7, pp. 277-289.

[48] M. Subbarao and T. S. Choi, "Accurate Recovery of Three-Dimensional Shape from Image Focus", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* March 1995, pp. 266-274.

[49] M. Subbarao, T. S. Choi, and A. Nikzad, "Focusing Techniques", *Journal of Optical Engineering*, Nov. 1993, pp. 2824-2836.

[50] M. Subbarao, and G. Surya, "Depth from Defocus by Changing Camera Aperture: A Spatial Domain Approach", Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, June, 1993, New York, pp. 61-67.

[51] M. Subbarao and G. Surya, "Depth from Defocus: A Spatial Domain Approach," *International Journal of Computer Vision*, 13, 3, pp. 271-294 (1994).

[52] M. Subbarao and J. K. Tyan, "The Optimal Focus Measure for Passive Autofocusing and Depth-from-Focus", Proceedings of SPIE's international symposium, Videometrics IV, Vol. 2598, pp. 89-99, Oct. 1995.

[53] G. Surya, *Three-Dimensional Scene Recovery from Image Defocus* , Ph.D. Thesis, Dept. of Electrical Engineering, State University of New York, Stony Brook, Dec. 1994.

[54] J. M. Tenenbaum, *Accommodation in Computer Vision,* Ph.D. Dissertation, Stanford University, Nov. 1970.

[55] B. Topielski, "A Computer Controlled Lens Positioning System for Autofocusing", Report submitted as part of a undergraduate senior design

project, Nov. 1990, Computer Vision Laboratory, Dept. of Electrical Engg., SUNY, Stony Brook, NY 11794-2350.

[56] T. Wei, *Three-Dimensional Machine Vision using Image Defocus* , Ph.D. Thesis, Dept. of Electrical Engineering, State University of New York, Stony Brook, Dec. 1994.

[57] Y. Xiong and S. A. Shafer, "Depth from focusing and defocusing," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition,* pages 68-73, June 1993.

[58] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Vol. 1, Academic Press, 1982.

[59] M. Subbarao, and G. Natarajan, "Depth Recovery from Blurred Edges", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, pp. 498-503, June 1988.

[60] M. Subbarao, "On the Depth Information in the Point Spread Function of a Defocused Optical System", Tech. Report No. 90.02.07, Feb. 1990, Computer Vision Laboratory, Dept. of Electrical Engineering, State University of New York, Stony Brook, NY 11794-2350.

[61] T. Poggio, V. Torre, C. Koch *Reading in Computer Vision* , Morgan Kaufmann Publishers, 1987.

[62] T. Poggio, V. Torre, C. Koch " Computational vision and regularization theory", Nature, Vol.317,No.6053, pp314-319, Sep 1985

[63] T. Poggio, V. Torre, A. Yuille, " A Regularization sulotion to edge detection", MIT, AI Memo 883 A.I. lab 1985

[64] D. Terzopoulos, *Regularization of Inverse Visual Problems Involving Discontinuities* IEEE Trans. PAMI, Vol. PAMI-8, No.4, pp413-424, July 1986.

[65] Tikhonov, A.N. , Arsenin, V.Y. *Solutions of ill-posed Problems*, Winston, Washington, DC , 1977

[66] J. W. Goodman, *Introduction to Fourier Optics,* McGraw-Hill, Inc., 1968.

[67] R. A. Jarvis, "A Perspective on Range Rinding Techniques for Computer Vision", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, No. 2, pp. 122–139, March 1983.

[68] M. Subbarao, and T. Wei, "Depth from Defocus and Rapid Autofocusing : A practical Approach", *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Champaign, Illinois, June 1992,pp. 773-776

[69] M. Subbarao, "Determining Distance from Defocused Images of Simple Objects", Tech. Report No. 89.07.20, Computer Vision Laboratory, Dept. of Electrical Engineering,

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

[71] J. Enns and P. Lawrence, “A Matrix Based Method for Determining Depth from Focus”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1991.

[72] G.A. Geist, B.W. Peyton, W.A. Shelton and G.M. Stocks, “Modeling High-temperature Superconductors and Metallic Alloys on the Intel iPSC/860,” *Proc. Fifth Distributed Memory Computing Conference, ed D. Walker and Q. Stout, IEEE Computer Society Press*, pp 504-512, April 1990.

[73] A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek and V. Sunderam, *PVM: Parallel Virtual Machine - A Users Guide and Tutorial for Network Parallel Computing,* MIT Press, 1994.

[74] V. Strumpen and T. L. Casavant, “Implementing Communication Latency Hiding in High-Latency Computer Networks,” *High-Performance Computing and Networking*, LNCS 919, pp. 86-93, Springer-Verlag, Milano Italy, May 1995.

[75] V. Strumpen, “Software-Based Communication Latency Hiding for Commodity Workstation Networks,” *IEEE International Conference on Parallel Processing*, pp. I-146 - I-153, 1996.

[76] M. Subbarao and Y.F. Liu, “Accurate Reconstruction of Three-dimensional Shape and Focused Image from a Sequence of Noisy Defocused Images,” SPIE Vol. 2909 pp. 178-191, Boston Mass., Nov. 1996

[77] M. Subbarao and Y.F. Liu, “Analysis of Defocused Image Data for 3D Shape Recovery using a Regularization Technique,” SPIE Vol. 3204, ISAM’97, Pittsburgh, Oct. 1997

[78] V. Sunderam, “PVM: A Framework for Parallel Distributed Computing,” *Concurrency: Practice and Experience*, Vol. 2 No. 4, December 1990.

[79] V. Sunderam, A. Geist, J. Dongarra and R. Manchek, “The PVM Concurrent Computing System: Evolution, Experience, and Trends,” *Parallel Computing*, Vol. 20 (4), 1993.

[80] D.C. Vargas, E.J. Rodríguez, M. Flickner, and J.L.C.Sanz “ Splines and Spline Fitting Revisited,” *Image Technology*, pp. 403-437, Springer 1996.