Integration of Focus and Defocus Analysis with Color Stereo for Three-Dimensional Shape Recovery

A Dissertation Presented

by

Ta Yuan

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

State University of New York

at Stony Brook

December 1999

Copyright © by
Ta Yuan
1999

State University of New York
at Stony Brook

The Graduate School

Ta Yuan

We, the dissertation committee for the above candidate for the

Doctor of Philosophy degree,

hereby recommend acceptance of this dissertation.

Muralidhara Subbarao, Advisor, Professor
Department of Electrical Engineering

Nam Phamdo, Chairman, Associate Professor
Department of Electrical Engineering

Ridha Kamoua, Associate Professor
Department of Electrical Engineering

Amitabh Varshney, Assistant Professor
Department of Computer Science

This dissertation is accepted by the Graduate School

Dean of the Graduate School

Abstract of the Dissertation

Integration of Focus and Defocus Analysis with Color Stereo for Three-Dimensional Shape Recovery

by

Ta Yuan

Doctor of Philosophy

in

Electrical Engineering

State University of New York

at Stony Brook

1999

Recovering the 3D (three-dimensional) shape of objects in a scene is an important topic in computer vision. Several methods including passive and active methods are developed to investigate this topic. Two kinds of information are recovered during the 3D shape recovery. One is the geometric information about the shape of the object. The other is the photometric information about the light energy distribution on the object surface. One effective method is developed in this research and its algorithms and results are presented in this dissertation.

The method developed in this research integrates three important techniques in machine vision — IFA (Image Focus Analysis), IDA (Image Defocus Analysis), and SIA

(Stereo Image Analysis). IDA is used to fast determine the spatial range occupied by the object and passes the result to IFA. With higher accuracy IFA uses the information from IDA to cut down the image acquisition and refines the estimated shape of the object. At last the refined shape information is used by SIA to limit the search range for finding conjugate points as well as to avoid the correspondence problem associated with SIA. A shape of better accuracy is obtained. Meanwhile the focused image of the object is obtained. This integrated method finds the optimal performance among the used image data, the computational cost, and the accuracy in the recovered shape of the object.

Color information is used and other techniques including multiple baseline stereo and multiple resolution stereo are integrated together to increase the accuracy and speed. IDA and IFA are incorporated with SIA to deal with the correspondence and occlusion problems. Results are improved and are to be further improved by match in blur stereo. Calibrations and experiments are done on a practical digital stereo vision system. A 360-degree 3D model with both geometrical and photometrical information is recovered by integrating multiple partial 3D models of an object recovered with the integrated method developed in this research. Results are presented and the effectiveness and usefulness are validated.

To Jesus,

My parents,
My wife, and my son

Table of Contents

Table of Figures

List of Tables

Acknowledgements

I thank God for His love and kindness for surrounding me with so many nice people and good things.

He blesses me with my loving parents who care for me more than themselves and my brothers who support me in any way they can. He also bestows on me my beloved wife and son. They give me joy beyond anything. I receive His love through the love from brothers and sisters in His church. None of this is done by me but solely given by Him.

I thank God for giving me a great advisor with lots of wisdom and patience. Prof. Murali Subbarao's inspiring advice and kind personality make my research work a pleasant experience. I would like to express my respect and cordial thanks to him. I also thank Prof. Wendy Tang for her kind help with my research. And I thank Prof. Nam Phamdo, Prof. Ridha Kamoua, and Prof. Amitabh Varshney for their advice and being in my dissertation committee.

I would like to thank Yen-fu Liu, Jenn-kwei Tyan, Xiangdong Qin, Huei-yung Lin and other members in the vision lab. for their help to me . And I thank Mrs. Maria Krause, Mrs. Judy Eimer, and Mrs. Deborah Kloppenburg for their help during my study at Stony Brook.

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

Chapter 1 Introduction

1.1 3D shape recovery

Recovering the 3D (three-dimensional) shape of objects in a scene is an important topic in computer vision. Two kinds of information about the 3D shape are to be recovered. One is the *photometric* information, which is related to the light energy casting upon the objects and is often referred to as *brightness*. The other is the *geometric* information, which is related to the spatial distribution that the objects occupy and is often referred to as the *depth map*.

Many computational methods are proposed to recover these two kinds of information of a scene. And generally they can be categorized into two types — *active* and *passive* — by their active sources used for sensing the objects of a scene. Active methods such as sonar ranging and laser ranging use sound wave and laser beam to sense and recover the shape of the objects in a 3D scene. On the other hand, passive methods use ambient lighting and sometimes, structured light, to acquire data for recovering the shape of the objects. Both types of methods have their own advantages and disadvantages. For example, using laser beam needs more complicated setup and might be harmful to living creatures, but its accuracy could be higher and more reliable. On the contrary, passive ranging methods such as shape from shading, shape from focus (image focus analysis, IFA), and stereo matching do not need extra equipment for active energy sources (which might be expensive in cost.) And they are not limited by the range of the energy sources like active ones do. But the accuracy may not be as good as that of some active methods in some aspects. There are many trade-off's such as cost, speed, accuracy, and so on, in applying ranging methods. And with the progress of both hardware and software those trade-off's are varying in significance with respect to each other.

This research focused on the category of *passive* ranging methods. Its goal is to develop a prototype of a low-cost ranging system that can recover the shape of objects in a scene fast and accurately.

1.2 Passive ranging

Passive ranging methods such as stereo vision — that is well known, motion parallax, and shape from shading have all been investigated extensively. Stereo vision and motion parallax utilize local correspondence and/or relative movement between two or more images to find out the shape of objects in the scene. Shape from shading use different lighting setups to recover the orientation and the depth map of objects. There is an inherent problem with stereo vision in finding the correspondence in the acquired images. This problem is referred to as the *correspondence problem*. This problem has two common causes. One is due to insufficient data during data acquisition (for example, the bordering area in one image cannot find its correspondence in others because its correspondence is outside the clipping window of the data acquisition equipment such as a camera.) The other is more complicated. It is the *occlusion* problem. Some part of the object is occluded by another part of the object while acquiring the corresponding data set. This problem will be dealt with in this research. Shape from shading does not have correspondence problem but it needs to change the lighting condition during the data acquisition. In the past decade, two new passive methods have been introduced and have shown advantages such as no correspondence problem and the feasibility of integration with other methods.

The Image Focus Analysis (IFA)[22,41,52,62,60] acquires lots of images with different parameters of the imaging system (such as a camera system.) Then it searches for the best focusing parameters along the acquired image volume by a *focus measure* (FM) in every direction in the imaging system. The camera parameters include *focal length*, *aperture diameter*, and *lens position*. Both the geometric (depth map) and photometric (focused image) information could be recovered by IFA. Several focus measures have been investigated. For example, *image energy*, *energy of image gradient*, and *energy of image Laplacian*.[52] Several IFA methods have been proposed by scientists. Subbarao and Choi[43] recovered 3D shape from Focused Image Surface instead of planar image frames by a method "Shape from Image Focus." Later, Subbarao and Tyan[41] derived AUM (*Autofocusing Uncertainty Measure*) and ARMS (*Autofocusing Root-Mean-square Error*) metrics to determine the best focus measure at hand. Recently Subbarao and Liu[44,45] developed another new IFA method which unified IFA and another passive method IDA (*Image Defocus Analysis*), which will be discussed next, to solve *Image Overlap Problem* and to recover the 3D shape by 3D PSF (*Point Spread Function*.)

Image Defocus Analysis (IDA)[9,26,28,40,49,58,70] is different from IFA in many aspects. First it does not search the imaging system parameters to find out the best focusing ones. Instead it computes the degree of the blur of the acquired data (i.e., images) and recovers the depth information of objects in a scene. Second, it does not need to acquire a lot of image frames; two or three are enough. Third, it needs careful and precise calibration data of the imaging system about its blurring characteristics. Or it would not be able to recover the shape well. Fourth, its computational cost is much lower than that of IFA. It is possible for IDA to perform real-time autofocusing, distance measurement, and shape recovery. The last difference is about the accuracy. IFA is better than IDA in accuracy. One major job for utilizing IDA is the calibration of blurring characteristics of the imaging system. Several models have been brought forth both in mathematics and experiments. For example, geometric optics model, wave optics model, and Gaussian model all provide ways of defining the PSF (*Point Spread Function*) and computing the *blur parameter* in order to recover the depth information about a scene. In some previous work of Subbarao and Surya,[40] they developed a method STM (Spatial domain convolution/deconvolution Transform Method) to do the shape recovery in spatial domain. In this method they modeled the image with a local cubic polynomial and the PSF as an arbitrary circularly symmetric function. On the other hand, Subbarao and Wei[70] proposed a Fourier domain approach to utilize the defocusing information and obtain the depth information of an object. Recently, Subbarao and Liu[44,45] introduced the concept of 3D PSF in their new method UFDA (*Unified Focus and Defocus Analysis*), in which they showed the flexibility to fine-tune between the IFA and IDA extremes to obtain better 3D shape and focused image.

Among all the shape recovering methods, stereopsis is probably most popular. It has attracted scientists' attention for many decades.[74] It uses a well known method called "*triangulation*" to recover the depth information about an object through computing the "*disparity*", that is, the *correspondence* of an object point between a pair of stereo images. It is often defined as the difference of the coordinates of the two corresponding image points, each of which lies in one of the stereo images respectively and belongs to the same spatial object point. Or it is defined as the shift of one image point in one of the stereo images from that of the same object point in the other image. This method is actually a human eye system analogy. Hundreds of papers have been brought forth discussing the exploration into this regime. This method's inherent problem is how to find out the "*correspondence*" (disparity) between stereo images. The problem is mentioned earlier as the "*Correspondence Problem*." And its computational task is sometimes referred to as "*stereo matching*."

Many papers propose different approaches of doing the stereo matching in order to reduce the false matches and increase the precision of the match. They could be categorized by different matching strategies.[13,74] One common categorization is "*area-based*" matching versus "*feature-based*" matching. Area-based matching such as correlation methods uses correlation between the corresponding brightness patterns in the stereo images to find out the match. It has two usual assumptions. One is the "*smoothness constraint*" in the local matching neighborhood. That is, inside the window of the local supportive neighborhood of matching, the depth is assumed to be constant. The other is the "*global consistency constraint*," or specifically, the "*figural continuity*." The "*figural continuity*" introduces the smoothness and continuity of the surface patches and contours in the images (of objects in the scene.) Using these two constraints we can determine the correct matches and/or detect and eliminate the false matches. On the other hand, feature-based matching such as edge matching uses symbolic features derived from intensity images to do the matching. The symbolic features include features such as orientation, end points of edges, edge points, length, contrast normal to the orientation, and so on. There are several strong and weak points associated with each of them. Here I am going to mention just one. The output of the area-based matching is usually a "*dense depth map*." And the result of the feature-based matching is a "*sparse depth map*." It is faster to do feature-based matching because its computation is less than that of the other. But in 3D shape recovery dense depth maps are preferred. And computational cost is reduced by several techniques such as integration of IDA/IFA with SIA (*Stereo Image Analysis*)[54,71], which is going to be discussed in this dissertation.

There are other ways to characterize the stereo vision. By the property of the input data acquired we have *gray-level stereo*[37,11] and *color stereo*.[5,75,4,38,19,71] By the geometrical setup of the stereo cameras we have *parallel-axis stereo*[37,54,11,1,20] and *non-parallel-axis stereo*.[55,63,56,75] By the number and geometry of stereo pairs acquired we have *binocular stereo*,[54,5,75] *trinocular stereo*,[1,20,55,56] and *multinocular stereo*.[37,11,63] And similarly we have single-baseline stereo[54,5] and *multiple-baseline stereo*.[37,71,11] Moreover by hierarchical structure of matching we have *single resolution stereo*[37,54] and *multiple resolution stereo (multiple scales stereo)*[5,71,13], which is a coarse-to-fine process. These different kinds of stereo could be combined and integrated together to facilitate different applications. For example, a camera array can be of parallel-axis, gray-level, and multinocular (multiple-baseline) stereo.[77,69] In this dissertation, integration of several different stereo methods is presented.

In order to solve the correspondence problem there are a few topics to discuss about. First is where to find the *conjugate point* (the correspondence) in the other image?

Trying to search the entire image increases computational cost and also the probability of false matches. From the geometric optics[21,59] a notion of "*epipolar line*" is introduced. Then the search for conjugate points lies solely on the epipolar line instead of in the entire image. In turn finding the epipolar line itself becomes a problem. In parallel-axis stereo it is much easier to determine epipolar lines. But in non-parallel-axis stereo the computation for determining epipolar lines are much more. Sometimes it is necessary to do extensive calibrations for the camera setup. In order to further reduce the search on the epipolar line the "*ordering constraints*" is introduced. It states that the order of object points is preserved in finding the correspondence along the epipolar line through geometric optics. But in the case that the occlusion is present the order may not preserve. That is, if there is concave part in an object the ordering may fail due to occlusion.[15,73] Many authors present their own methods to determine the search range during matching on the epipolar lines. In this dissertation an effective method of determining the search range on the epipolar lines is brought forth. It integrates the IDA and IFA to find out a good estimate of the object shape and thus limits the range of search in SIA.

The second topic in solving the correspondence problem is the "foreshortening effect" of the appearance of an object in the stereo images.[13] This effect seriously affects the area-based correlation type methods. Some papers present methods such as "*warping*" (or *reprojection*)[13,74] of the images to compensate this effect. One possible way to deal with this effect is to use multiple-baseline stereo. With shorter baseline and hence less foreshortening effect we do the matching. Then the result could be used to better guide the matching in the longer baseline stereo with more foreshortening effect but higher accuracy. Not only the area-based matching is affected by the foreshortening effect but also does the feature-based matching. The orientation, length, and spacing of adjacent edge segments will change after the foreshortening.

The next topic is the "*occlusion problem*." Stereo matching fails when the correspondence can not be found or is wrongly found. Occlusion itself is a big topic in stereo vision. Many researchers try to solve or alleviate the problem.[14,18,77,15,73,6] Occlusion usually associates with abrupt depth change. This will violate the consistency in the two assumptions in area-based methods. Choosing the size of the correlation window becomes a significant issue while matching in non-smooth surfaces. An adaptive window approach has been proposed to choose the best size of the window.[68] Usually edges will appear at places of abrupt depth changes. Although featured-based methods can detect the existence of edges (brightness changes) but some features will still be hidden if occlusion is there and thus cripple the matching process. So occlusion detection is necessary in solving the correspondence problem. In this research one useful occlusion detection algorithm is proposed and experimented. It combines the IDA/IFA output and the SIA results to find out where the possible occlusion happens.

The last one is "*image details*." For area-based methods there must be enough image details (contrast) to result in a successful matching. For featured-base methods there also must be enough image details for matching. Or even interpolation is applied we still cannot obtain satisfactory results. On the other hand, if the details are too much the preprocessing time for extracting features increases. And also the probability of false matches increases. One effective way is to work on multiple resolutions or scales to speed up matching and to reduce false matches. In this dissertation a high contrast pattern is projected onto the object that lacks image details to facilitate the matching used in this research.

1.3 Integration of multiple methods

Since each method has its own strong and weak points. It is a good idea to strengthen the good points and cancel the weak points by integrating more than one method into one. It provides for the users with more choices and fits the need of different applications. Before integrating any method with another it is necessary to do extensive investigation into the methods that are going to be integrated. Without knowing the capacities and the limits of each method the integration would be trivial and irrelevant.

Some researchers have tried integrating two or more methods into one. For example, combining *depth from focus* (DFF) with stereo ranging[67], or *depth from defocus* (DFD) with stereo vision[70], integration of focus, camera vergence, and stereo[8], unification of IDA and IFA[79], and integration of DFF and DFD[33]. In this dissertation a new approach is developed. This new method integrates IDA/IFA and SIA. Besides that, color stereo, multiple-baseline stereo, multiple-resolution stereo, and occlusion detection have all been integrated into one method in different degrees to improve 3D shape recovery.

The new algorithm could be briefed as follows: By IDA a rough depth map of the scene is obtained. IDA takes two images at two fixed camera lens positions, say, at step number 120 and 155. Usually the lens position is referred to as "*lens step number*" of the motorized camera lens. After processing the defocus information stored in the two blurred images a rough depth map can be recovered. This rough depth map provides for us information about the range that the objects of the scene occupy. It is also used as an input to the next stage, IFA stage, for obtaining a more accurate estimate of the 3D scene. Using IDA output as input, IFA could limit the range that it is going to search for the focus information of the 3D scene and hence reduce the number of images that it is going to take. IFA takes more images than IDA does but its result is more accurate than IDA's. So a refined depth map could be obtained after a successful IFA processing. Those depth maps are not as accurate as that obtained from stereo (SIA) but do not suffer the correspondence problem. It provides for the SIA a good initial estimate of the scene. SIA can then estimate the search range of stereo matching and thus reduces both false matches and the processing time. Sometimes due to the full knowledge of the bounding range (the possible working range) of objects in some cases, IDA stage could be simply omitted by just passing the known range into IFA stage. This could serve as a variation of IDA/IFA method. But for a more general case IDA does provide a useful way to estimate the working range of the scene.

In the last stage, SIA stage, stereo matching is performed. In this research the matching falls into parallel-axis, area-based stereo. A dense depth map is generated. To the result from IDA/IFA an estimated error is added to form a range in which the correct depth resides. Then the search range of disparity is formed from the range of depth estimated from IDA/IFA. Finally a depth map with the accuracy same as that of the stereo vision (better than IFA) is obtained. This is the backbone of the integration of IDA/IFA/SIA. Besides this, color stereo is integrated into the backbone. Color stereo allows data acquisition to be in color. Color data improves the accuracy of stereo matching.[4] And color focused images obtained from the *color shape from focus* provide color textures for mapping after the 3D shape is recovered. Multiple baseline stereo is also integrated into the backbone. By multiple baseline stereo an intermediate depth map is obtained from a shorter baseline. This depth map has higher accuracy than that of IDA/IFA and has lower accuracy than that in the case of longer baseline. With this estimate the search range for the case of longer baseline is further reduced. Therefore the total time in stereo matching is reduced. And the probability of false matches is also reduced because the search range is reduced in both cases. Multiple resolution stereo is another way of speeding up the matching. In multiple resolution stereo the size of the original stereo images is reduced. The resulted depth map is then enlarged to guide the matching in the higher resolution stereo images. The total time for matching is reduced. With different multiple baseline setup or different levels of resolution the speedup of matching could be adjusted, depending on the need of the applications. Finally the occlusion detection is implemented into the backbone. This part consists of two parts. One is "*background removal*" that separates the objects from the background. After the removal of background the matching time is reduced and also the false matches are reduced. The other part is "*object occlusion detection*." "Object occlusion" means some part of the object is occluded by other parts of the object. It's not like the background to be removed. By using the IDA/IFA results the candidate match is "*back-matched*" to the original image. The result shows where the possible occlusion occurs. The details will be shown in later passages of this dissertation.

1.4 Organization of this dissertation

This dissertation is organized as follows: Chapter 1: This introduction. It discusses about the 3D shape recovery, passive and active ranging methods, famous passive ranging methods, IDA, IFA, SIA and its common problems, integration of passive methods, the integration of IDA, IFA, SIA in this dissertation, and a brief overview of this dissertation. Chapter 2 discuses the theoretical background of IDA, IFA, SIA, and all the different kinds of stereo, such as multiple-baseline stereo, multiple-resolution stereo, color stereo, and so on. Then I will discuss the theory of integration of IDA, IFA, and SIA. Finally some new theories for improving SIA, such as occlusion detection and "*match in blur*" are discussed. Chapter 3 shows the camera calibrations and experiments on algorithms for integrating IDA, IFA, and SIA. Results of these experiments are presented. Chapter 4 displays the results of color stereo, multiple baseline stereo, multiple resolution stereo, and "match in blur". Chapter 5 shows the practical application of the integrated IDA/IFA/SIA algorithm — the recovery of a 360 degree shape of an object. Multiple 3D views of an object are recovered and stitched into one 3D shape of 360 degree. The results are very impressive. It not only shows the feasibility of practical use of this algorithm on real objects but also shows the validation of the algorithm. Chapter 6 concludes this dissertation and discusses the future work.

Chapter 2 Theoretical Background

2.1 Introduction

In this chapter I am going to discuss the theoretical background of the methods used in this research. IFA, IDA, SIA, and their integration. Other stereo methods like color stereo, multiple baseline stereo, multiple resolution stereo, occlusion detection, and match in blur are also discussed. The goal is to build up a solid theory for these methods. The experimental part is presented in later chapters.

In machine vision, three of the important techniques for ranging and three-dimensional (3D) shape recovery are — image defocus analysis (IDA),[9,28,49,58,47,64,39,48,46] image focus analysis (IFA),[12,22,29,30,60,25] and stereo image analysis (SIA).[27,66,76,10,17,16,32,8,13,74]

In IFA[41,43,44,45,52,62,79,33,78] 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.

In IDA[26,40,44,45,70,51,50,65,53,42] only a few (2 or 3) image frames of a scene are recorded with different camera parameters. The degree of defocus of each object in these few image frames along with the corresponding camera parameters are analyzed to find the focused image and distance of every object in the scene. In comparison with IFA, IDA requires (i) less number of images, (ii) less computation, but (iii) more information about the defocusing characteristics of the camera. In addition, IDA is less accurate than IFA. Methods in this research integrate the two techniques to obtain accuracy equivalent to IFA but with less number of images and computation than IFA. This is accomplished by first using IDA to obtain a rough estimate of depth-map, and improving the accuracy of the estimate using IFA in a narrow range around the estimated depth-map.

In SIA[37,54,71,69,5,57] two or more images are recorded from two or more spatial locations by displacing the camera. Then distances of objects in the scene are found through "*triangulation*." Depending on camera parameter values, in some vision systems (e.g. the human vision system), IDA and IFA methods are less accurate than stereo vision in providing the depth-map of a scene. However, unlike stereo vision, IDA and IFA do not suffer from the *correspondence* and *occlusion* problems. In this dissertation, I describe a technique for integrating IFA and IDA with stereo vision. This technique has the potential to result in a fast, reliable, and accurate method for ranging and 3D shape recovery. The rough depth-map provided by IDA and IFA is used to simplify the stereo correspondence and occlusion detection problems. The rough depth-map essentially reduces the range of stereo disparity, which is searched for stereo matching. In addition, false matches due to occlusion are also reduced. Therefore, stereopsis yields a more accurate 3D shape of objects. If the object whose 3D shape is to be measured does not have sufficient contrast information, then contrast is introduced by projecting a light pattern onto the object. This facilitates the application of IFA, IDA, and SIA.

The technique described in this dissertation is implemented on a camera system named Stonybrook VIsion System (SVIS) and also a newer system, DVS (Digital Vision System.) Methods for calibrating the camera system and results of experiments on SVIS and DVS are presented in later chapters.

2.2 Review of Image Focus Analysis, Image Defocus Analysis, and Stereo Image Analysis

2.2.1 Image Focus Analysis

The depth-map and the focused image of a scene are respectively the geometric and photometric information that are of interest in machine vision. IFA and IDA are useful in recovering both these types of information. IFA methods (see Fig. 1) 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 (photometric constraint), and (ii) the position of the point in the scene and the position of its focused image are related by the lens formula (geometric constraint)[21,34,31,35,59]

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \qquad \text{Equation 2-1}$$

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 Equation 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 Equation 2-1). This surface is defined to be the Focused Image Surface (FIS) and the image irradiance on this surface to be the focused image. There is a one to one correspondence between 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.

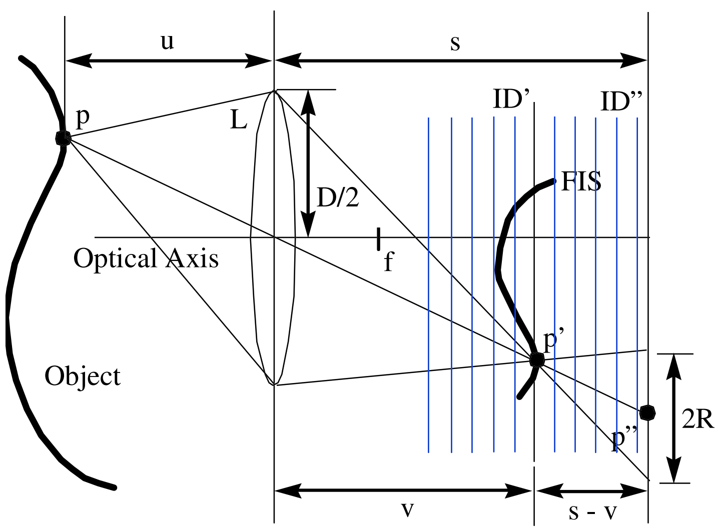


Figure 2-1 Image focus analysis

In traditional IFA methods a sequence of images is obtained by continuously varying the distance s between the lens and the image detector or/and the focal length f (see Figure 1). 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, the image frame among the image sequence, which has the maximum focus measure, is found by a search procedure. The gray level (which is proportional to image irradiance) of the pixel in the image frame thus found gives the gray 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 gray 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.

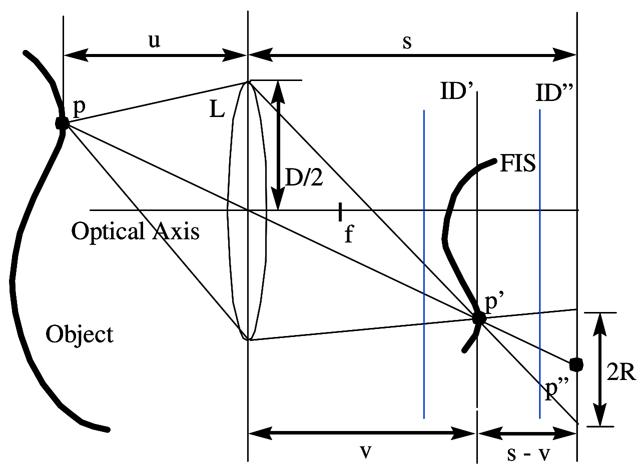


Figure 2-2 IDA (Image Defocus Analysis)

2.2.2 Image Defocus Analysis

IDA methods do not require focusing the object. They take the level of defocus of the object into account in determining distance and focused image. IDA methods do not involve searching for *f* and *s* values that correspond to focusing the object. Therefore these methods require processing only a few images (about 2-3, see Figure 2-2) 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 camera characteristics (a blur parameter as a function of camera parameters), and (ii) they are less accurate than IFA methods. Here we summarize two main approaches — (1) Fourier Domain approach [70] and Spatial Domain approach [26] in the following subsections.

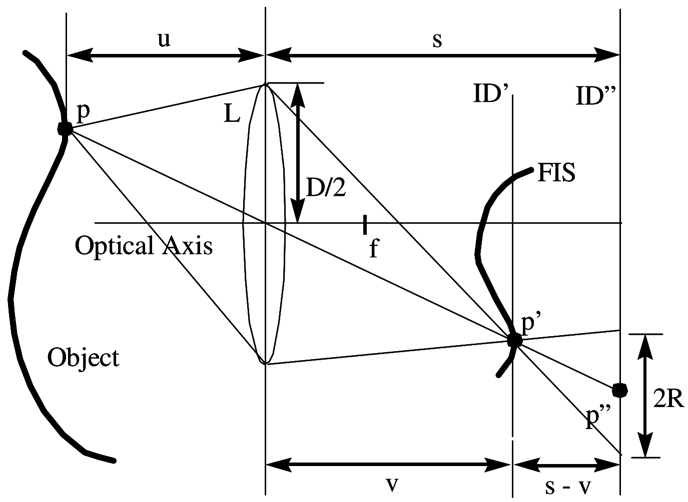


Figure 2-3 Image formation model

Fourier Domain Approach

The camera setting in the image formation model (See Figure 2-3) can be denoted as $e = (s, f, D)$. Where s is the distance from the lens to the Image Detector (ID), f is the focal length, and D is the aperture diameter. In paraxial geometric optics [31] the normalized radius of the blur circle R, which is a function of camera parameters e and object distance u,

$$R(e;u)=\frac{D}{2}\left(\frac{1}{f}-\frac{1}{u}-\frac{1}{s}\right), \qquad \text{Equation 2-2}$$

is a constant over the image detector for fixed e and u. In this case the camera acts as a linear shift invariant system. Therefore the image $g(x,y)$ (often blurred) becomes the convolution of the focused image $f(x,y)$ with the corresponding *point spread function*

(PSF), that is, $g(x,y) = h(x,y) * f(x,y)$. For a Gaussian PSF model the point spread function is a two-dimension Gaussian function :

$$h(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad \text{Equation 2-3}$$

where σ is the spread parameter and is proportional to the normalized radius of the blur circle R, that is, $\sigma = cR$, where c is a camera constant. In the Fourier domain, the convolution becomes multiplication and so $G(u,v)$ is equal to $H(u,v)$ times $F(u,v)$ where $H(u,v)$ equals $e^{-\frac{1}{2}(u^2+v^2)\sigma^2}$. For two blurred images g_1 and g_2, taken with two different camera settings e_1 and e_2, we obtain $G1(u, v)/G2(u,v) = e^{-\frac{1}{2}(u^2+v^2)(\sigma_1^2-\sigma_2^2)}$ or,

$$\sigma_1^2 - \sigma_2^2 = \frac{-2}{u^2+v^2} \ln \frac{|G_1(u,v)|}{|G_2(u,v)|} \qquad \text{Equation 2-4}$$

Besides,

$$\sigma_1 = cR_1 = c\frac{D_1}{2}\left(\frac{1}{f_1} - \frac{1}{u} - \frac{1}{s_1}\right), \quad \sigma_2 = cR_2 = c\frac{D_2}{2}\left(\frac{1}{f_2} - \frac{1}{u} - \frac{1}{s_2}\right) \qquad \text{Equation 2-5}$$

By eliminating *1/u* from the above two equations we have

$$\sigma_1 = \alpha\sigma_2 + \beta \quad , where \;\; \alpha = \frac{D_1}{D_2}, \;\; \beta = cD_1\left(\frac{1}{f_1} - \frac{1}{f_2} + \frac{1}{s_2} - \frac{1}{s_1}\right) \qquad \text{Equation 2-6}$$

Equation 2-4, Equation 2-6 together yield

$$(\alpha^2 - 1)\sigma_2^2 + 2\alpha\beta\sigma_2 + \beta^2 = \frac{-2}{u^2+v^2} \ln \frac{|G_1(u,v)|}{|G_2(u,v)|} \qquad \text{Equation 2-7}$$

The only unknown left is s_2 that can be easily obtained by solving the above quadratic equation. The methods of solving the two-fold ambiguity of the solved s_2 are discussed in [42,40,26,70]. After s_2 is determined the object distance u can be easily found.

The above discussion depicts the idea of finding the object distance with two defocused images. Applying the same procedure on the whole image area the 3D information of the scene can be recovered. Not like the Gaussian PSF model the general PSF model of a camera cannot obtain a close-form solution. A numerical approach should be used. Detailed discussion of the case of arbitrary PSFs is presented in [70].

Spatial Domain Approach

A spatial domain convolution/deconvolution transform (*S-transform*) is defined in [50] and can be used in image defocus analysis. The general form of the S-transform deals with images and n-dimensional signals for the case of arbitrary order polynomials and is quite complicated. A special case of it under a local cubic polynomial model, however, turns out to be simple and suffices for image defocus analysis.

If a focused image f is blurred by convolution with a circularly symmetric PSF h to result in a blurred image g, then g is the forward S transform of f with respect to the kernel function h, and it is given by :

$$g(x, y) = f(x, y) + \frac{h_2}{2} \nabla^2 f(x, y) \qquad \text{Equation 2-8}$$

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

$$h_2 = \iint x^2 h(x, y) dxdy = \iint y^2 h(x, y) dxdy \qquad \text{Equation 2-9}$$

and ∇^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

$$f(x, y) = g(x, y) - \frac{h_2}{2} \nabla^2 g(x, y) \qquad \text{Equation 2-10}$$

Using the definition of the moment of h and the definition of the blur parameter σ of h, we have $h_2 = \sigma_2/2$. for a Gaussian PSF model used in the previous subsection.

For two blurred image g_1 and g_2, taken with different camera settings e_1 and e_2 corresponding to blur parameters σ_1 and σ_2 we have

$$f = g_1 - \frac{\sigma_1^2}{4}\nabla^2 g_1, \quad f = g_2 - \frac{\sigma_2^2}{4}\nabla^2 g_2 \qquad \text{Equation 2-11}$$

From the above two equations, Equation 2-6, and the fact that $\nabla^2 g_1 = \nabla^2 g_2$ we obtain

$$(\alpha^2 - 1)\sigma_2^2 + 2\alpha\beta\sigma_2 + \beta^2 = \frac{4(g_1 - g_2)}{\nabla^2 g_1} \qquad \text{Equation 2-12}$$

where α and β are as defined in the previous subsection. Therefore we have a quadratic equation in σ_2 that can be found easily. The object distance u can be obtained by means of σ_2 and the camera setting in Equation 2-5. Implementation details and experimental results for this approach are described in [40,26]. In this dissertation the IDA method belongs to the spatial domain approach.

An IDA method can be combined with an IFA method to reduce the number of images acquired and processed but attain the same accuracy as IFA. First IDA method is used to obtain a rough depth-map. This requires acquiring and processing only 2 or 3 images. Then IFA is applied to a short sequence of images that are acquired with camera parameters so that only objects in the rough depth-map range estimated by IDA are focused. Acquiring and processing of image frames that correspond to focusing objects that are far away from the depth-map estimated by IDA is avoided. This saves image acquisition and processing time of unnecessary image frames. The accuracy of the depth-map obtained will be the same as that of IFA. In comparison with a bare IFA, the combined IDA/IFA may save much time in the best case when all objects in a scene are at the same distance, but in the worst case when objects in scene are at all possible distances, the combined IDA/IFA will take a bit more time than the bare IFA. In a typical application, the combined method can be expected to save some modest time.

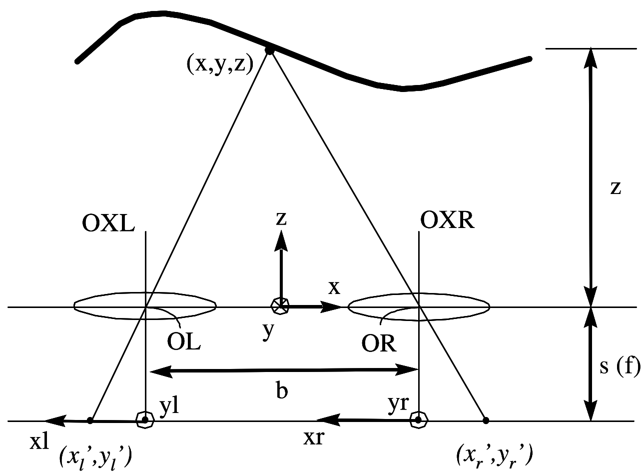


Figure 2-4 Triangulation method

2.2.3 Stereo Image Analysis

In SIA,[13] depth-map is recovered by comparing two images (a right image and a left image) acquired from different camera positions (a right position and a left position). (See Figure 2-4.) The line connecting the two camera positions is called the *baseline*. In this disseration we consider only the simplest case where the baseline is perpendicular to the optical axis of the camera. In this case, the image displacement d (also called *disparity*) of a point in the left image with respect to its position in the right image is $d = bs/z$ where b is the baseline distance, s is the ("*focal length*") distance between the optical center and the image detector, and z is the distance of the point from the camera along the optical axis. Let the image coordinates in the left image be (x_l', y_l') and that in the right image be (x_r', y_r'). From the similarity of the triangles we have

$$\frac{x_l'}{f} = \frac{x + b/2}{z} \quad and \quad \frac{x_r'}{f} = \frac{x - b/2}{z}, \qquad \text{Equation 2-13}$$

$$\frac{y_l'}{f} = \frac{y_r'}{f} = \frac{y}{z} \quad . \qquad \text{Equation 2-14}$$

Rearrange Equation 2-13 and Equation 2-14 we obtain

$$\frac{x_l' - x_r'}{f} = \frac{b}{z} \quad . \qquad \text{Equation 2-15}$$

The amount *(x_l'-x_r')* is the *disparity*. And we can solve for *x*, *y*, and *z* :

$$x = b\frac{(x_l' + x_r')/2}{x_l' - x_r'} \quad , \quad y = b\frac{(y_l' + y_r')/2}{x_l' - x_r'} \quad , \quad z = b\frac{f}{x_l' - x_r'} \quad . \qquad \text{Equation 2-16}$$

From the above equations we note that distance is inversely proportional to disparity, and disparity is directly proportional to baseline, and is also proportional to the effective focal distance. We can think that given a fixed error in determining the disparity we may increase the accuracy of the depth determination by increasing the baseline. However the two images become less similar when the two cameras separate farther and hence not suitable for stereo matching anymore.

In a general case, the two cameras will not be aligned exactly, as we have assumed in our case, the simplest one. It would be very difficult to arrange for the optical axes to be parallel to each other and the baseline exactly perpendicular to the optical axes. We need to do camera calibrations on behalf of these issues. Assume we have a stereo camera system in such a general case. In order to measure object distance we have to calibrate the system to find out the relation between the two cameras' relative positions and orientations.

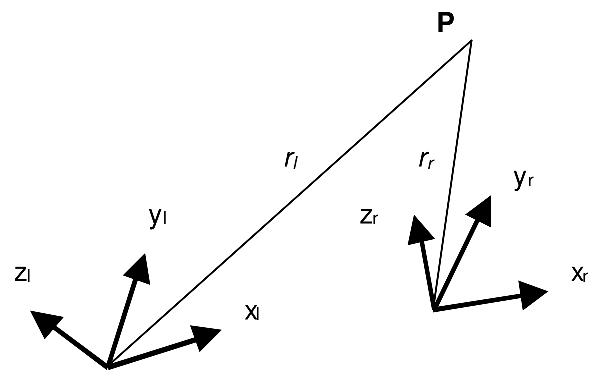


Figure 2-5 Transformation between cameras

The transformation between two camera stations can be treated as a rigid body motion [13] and can be decomposed into a rotation and a translation (see Figure 2-5). If $r_l = (x_l, y_l, z_l)'$ and $r_r = (x_r, y_r, z_r)'$ is the coordinates of the same point, which measured in the left and right camera stations, respectively, then

$$r_r = \mathbf{R} r_l + r_0 \,, \qquad \text{Equation 2-17}$$

where R is a 3 × 3 orthonormal matrix representing the rotation and r_0 is an offset vector corresponding to the translation. For an orthonormal matrix the following condition holds:

$$\mathbf{R}'\mathbf{R} = \mathbf{I} \,, \qquad \text{Equation 2-18}$$

where " ′ " means the matrix transposition and I represents the 3 × 3 identity matrix. Once we know (R, r_0) from calibration, we can compute the position of a point with known left and right image coordinates. If (x_l', y_l') and (x_r', y_r') are these coordinates, then

$$\left(r_{11}\frac{x_l^{'}}{f}+r_{12}\frac{y_l^{'}}{f}+r_{13}\right)z_l+r_{14}=\frac{x_r^{'}}{f}z_r\ ,$$
$$\left(r_{21}\frac{x_l^{'}}{f}+r_{22}\frac{y_l^{'}}{f}+r_{23}\right)z_l+r_{24}=\frac{y_r^{'}}{f}z_r\ , \qquad \text{Equation 2-19}$$
$$\left(r_{31}\frac{x_l^{'}}{f}+r_{32}\frac{y_l^{'}}{f}+r_{33}\right)z_l+r_{34}=z_r\ .$$

From these equations we can solve z_l and z_r and then compute r_l and r_r. In this research it is a simple two parallel camera system (simulated by moving a camera along a positioning stage.) Evolution into a more general one is possible in the future research.

The key problem in stereopsis is how to find the pair of corresponding points (the conjugate points) in the left and right images? That is, if we have (x_l', y_l') in the left image how could we find out (x_r', y_r') in the right image, or vice versa? In the general case, assuming (R, r_0) is known and distance from lens to image plane (image detector) are the same for both cameras, say f, an image point (x_l', y_l') in the left image corresponds to a ray through the origin of the coordinate system :

$$x_l = x_l's, \qquad y_l = y_l's, \qquad z_l = fs\ . \qquad \text{Equation 2-20}$$

In the right coordinate system the coordinates of points on this ray are

$$x_r=\left(r_{11}x_l^{'}+r_{12}y_l^{'}+r_{13}f\right)s+r_{14}\ ,$$
$$y_r=\left(r_{21}x_l^{'}+r_{22}y_l^{'}+r_{23}f\right)s+r_{24}\ , \qquad \text{Equation 2-21}$$
$$z_r=\left(r_{31}x_l^{'}+r_{32}y_l^{'}+r_{33}f\right)s+r_{34}\ ,$$

and their projections onto the right image plane are

$$\frac{x_r^{'}}{f}=\frac{x_r}{z_r} \quad and \quad \frac{y_r^{'}}{f}=\frac{y_r}{z_r}\ . \qquad \text{Equation 2-22}$$

Combining Equation 2-21, Equation 2-22, and let $x_r = as + u$, $y_r = bs + v$, $z_r = cs + w$, we have :

$$\frac{x_r'}{f} = \frac{a}{c} + \frac{cu - aw}{c} \frac{1}{cs + w} \quad ,$$
$$\frac{y_r'}{f} = \frac{b}{c} + \frac{cv - bw}{c} \frac{1}{cs + w} \quad . \qquad \text{Equation 2-23}$$

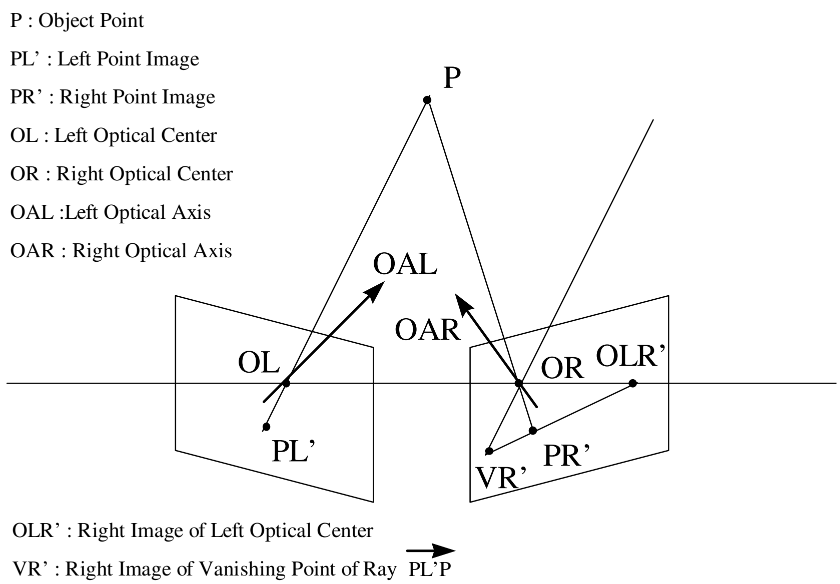


Figure 2-6 Epipolar Line

This becomes a straight line starting from *(u/w, v/w)*, for *s = 0*, to *(a/c, b/c)*, as *s* goes to infinity. The former is the right image of the origin of the left camera coordinates. The latter is the right image of the vanishing point of the ray (see Figure 2-6). All the matching for (x_l', y_l') is done along the line on the right image; the line is called epipolar line. The stereo camera system in this research is assumed to have parallel optical axes to which the baseline is perpendicular. In this configuration the epipolar lines are all horizontal and parallel to one another (see Figure 2-7). This simplifies the matching algorithm used in this dissertation. The disparity (image displacement of points in the right and left images) is obtained by matching the point in the right image to the corresponding point in the left image. The matching process is computationally intensive since all permissible values of d need to be searched. In addition, sometimes, due to occlusion, no matching point will be present in the left image. However, if we know the approximate distance *z′* of a point at actual distance *z*, then the search for the matching point can be limited to a small range around the expected displacement of $d'=bs/z'$ along the epipolar line. An estimate of *d′* can be obtained by the combined IDA/IFA method for depth-map recovery. Thus, integrating IDA, IFA, and SIA, the search interval for the matching points can be reduced dramatically and computational time can be saved. Also, the depth-map from IDA/IFA can be used to detect occlusion and avoid false matches. The accuracy of the depth-map obtained by the integrated method will be the same as that of SIA.

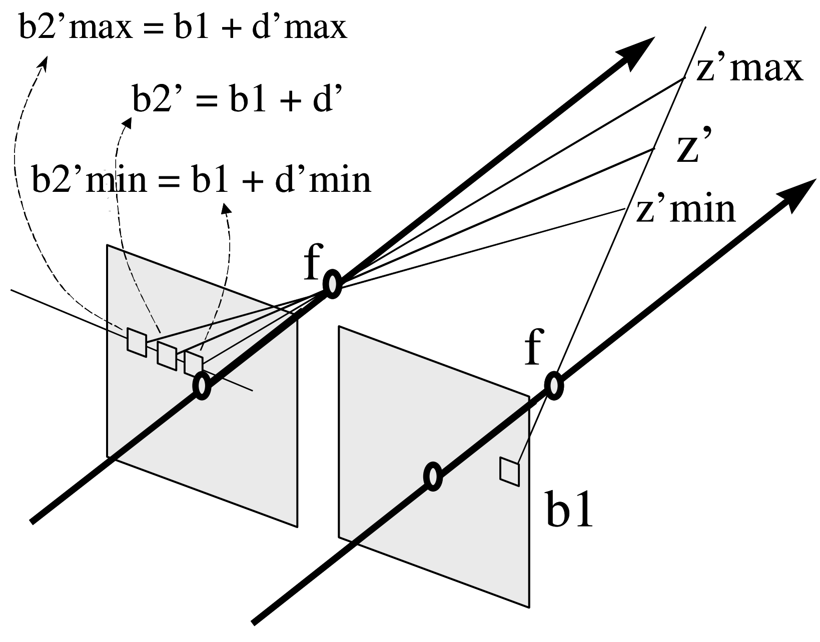


Figure 2-7 Parallel optical axes

There are many approaches for locating (matching) the conjugate points. These methods are all dealing with the correspondence problem, which states of identifying features in two images that are projections of the same entity in the three-dimensional world.[13] Once this is done, we can compute the distance to this entity. If the point on the object surface appears in both images, a matching algorithm could be applied and the two image points must lie on corresponding epipolar lines as previously mentioned.

To identify conjugate points we ought to first analyze the image and extract features out of it. These features locate where to do the matching. Distinctive gray level patterns such as an edge is a good example for matching. Here we are going to brief three types of matching approaches in the following subsections.

Gray Level Matching

For a portion of a surface that is smooth and not tilted too much with respect to lines connecting it to each camera station. Suppose neighboring points are imaged in both camera stations. And we think about matching the gray levels of these neighboring points. It is like matching gray level waveforms on corresponding epipolar lines. Due to differences in the foreshortening of surface intervals as viewed from two camera stations, these waveforms are compressed and/or expanded. Consider the simple geometry of camera positions:

$$\frac{x_l^{'}}{f} = \frac{x+b/2}{z} \qquad and \qquad \frac{x_r^{'}}{f} = \frac{x-b/2}{z} \quad .$$

We try to find a function $z(x,y)$ such that

$$E_l(x_l^{'}, y_l^{'}) = E_r(x_r^{'}, y_r^{'}),$$

or

$$E_l\left(f\,\frac{x+b/2}{z(x,y)}, y^{'} \right) = E_r\left(f\,\frac{x-b/2}{z(x,y)}, y^{'} \right) \quad . \qquad \text{Equation 2-24}$$

Let $$\frac{x'}{f} = \frac{x}{z} \quad and \quad d(x',y') = \frac{bf}{z} \quad . \qquad \text{Equation 2-25}$$

Equation 2-24 becomes

$$E_l\left(x'+\frac{1}{2}d(x',y'),y' \right)= E_r\left(x'-\frac{1}{2}d(x',y'),y' \right) .$$ Equation 2-26

Now the question becomes finding out a disparity function *d(x',y')* such that the above equation hold. We want *z*, and hence *d*, to vary smoothly. Thus we might look for a solution that minimizes some measure of departure from smoothness, such as

$$e_s = \iint (\nabla^2 d)^2 dx'dy' .$$ Equation 2-27

On the other hand we want to minimize one term associated with the inexact image brightness measurements:

$$e_i = \iint (E_l - E_r)^2 dx'dy' .$$ Equation 2-28

Overall we are minimizing $e_s + \lambda e_i$, where λ is a weighting factor that is large if brightness measurements are accurate and small if they are not. Now the Euler equation is

$$F_d - \frac{\partial}{\partial x'} F_{d'_x} - \frac{\partial}{\partial y'} F_{d'_y} = 0 ,$$ Equation 2-29

where

$$F = (\nabla^2 d)^2 + \lambda\left(E_l\left(x'+\frac{1}{2}d(x',y'),y' \right) - E_r\left(x'-\frac{1}{2}d(x',y'),y' \right)\right)^2 ,$$ Equation 2-30

so that

$$\nabla^2(\nabla^2 d) = \lambda\left(E_l - E_r\right)\frac{1}{2}\left(\frac{\partial E_l}{\partial x^2} + \frac{\partial E_r}{\partial x^2} \right) .$$ Equation 2-31

Here E_l and E_l/x' are measured at the point *(x'+(1/2)d(x',y'),y')* in the left image and E_r and E_r/x' are measured at the point *(x'-(1/2)d(x',y'),y')* in the right image. The operator

$$\nabla^2(\nabla^2) = \frac{\partial^4}{\partial x'^4} + 2\frac{\partial^4}{\partial x'^2 y'^2} + \frac{\partial^4}{\partial y'^4}$$ Equation 2-32

is called the biharmonic operator.

Replace the biharmonic term $\nabla^2(\nabla^2 d)$ by $\kappa(d_{ij} - \underline{d}_{ij})$ where d is the result of convolving d with a computational molecule derived from a molecule that is appropriate for the biharmonic operator. Thus we have an iterative scheme of the form

$$d_{i,j}^{n+1} = \underline{d}_{i,j}^{n} - \frac{\lambda}{\kappa}\left(E_l - E_r\right)\frac{1}{2}\left(\frac{\partial E_l}{\partial x^2} + \frac{\partial E_r}{\partial x^2}\right) . \qquad \text{Equation 2-33}$$

where the partial derivatives of E_l and E_r are estimated using first differences. This iteration will tend to reduce the difference in brightness. It needs smooth disparity change and good initial values. Smoothing or blurring the image at different degrees and working in a manner of multi-scaling may help this approach work well. The gray level of conjugate points are in general not exactly the same since the surface is being observed from two different directions. Some specular surface and recording error may also change the gray levels.

Correlation Methods

The idea of these methods is that if conjugate patches of two images have similar brightness then we will have a correlation peak after we correlate the patches along the epipolar lines. But choosing the size of the patches is a big question, though. It cannot be too small or the brightness pattern will not be distinctive enough and false matches occur. It cannot be too large or the resolution will be lost. Even worse these two patches cannot match well since patches with different disparity are combined together. Multiple resolution scheme matches on reduced images first then on higher resolution ones with the help of some estimated information obtained from previous low-resolution results. Their major shortcomings include i) the less sensitivity to differences in foreshortening, and 2) the inability to determine disparity in regions that lack image detail. A modification is warping the images (before they are sent to the correlator) to some fictitious intermediate image (the third image) or by some initial value of disparity and then iterating with improved disparity. But the convergence is not guaranteed. And also edge-enhancing may help the correlation methods succeed.

Edge-Matching Methods

It is hard to find conjugate points if the gray level is more or less constant. The correlation methods cannot find clear maximum if the size of patch is less than the one of the uniform region. The gray level matching methods obtain the interpolated disparity from neighboring areas with suitable gray level variations. Matching at places of rapid change of brightness is more reasonable. Such places are most likely to be edges in the images. Matching is carried out between crude symbolic descriptions of the images rather than between images themselves. False matches are possible because there could be many edges along a given epipolar line. Auxiliary information such as the cross-edge gray level difference will also be recorded with the edge itself. Some useful constraint like order-preservation on epipolar line (see Figure 2-7) and some assumption such that all the edges are visible in the two images still cannot prevent ambiguities. Multiple scaling can be adopted and is also proved useful and acceptable. Some edge detection techniques can be used to find out the edges on the images.

In this dissertation the matching techniques include the following major parts :

1. Combining IDA/IFA to obtain a rough depth-map, avoiding the correspondence problem in stereo matching.

2. Using the rough depth-map with the camera calibration table (steps vs. distance) to confine the matching in an estimated range of disparity and reduce large amount of computation.

3. Matching image blocks (patches) by the criterion of finding the minimum *SSD* (sum of squared differences).[37]

There are several ways to improve the above matching techniques. For example, utilization of the techniques described in previous subsections such as the multiple scaling, and so on, could help the matching. The details are discussed in the following sections.

2.3 Improvement in Stereo Image Analysis

2.3.1 Color SIA

Gray level images have facilitated research in areas like machine vision for a long time. From its names we can see the various applications it has been adopted. Those names such as "*black and white*," "*monochrome*," "*gray scale*," and so on. Sometimes it is necessary to indicate that the image is a pure "binary image" or one of 256 gray levels to avoid confusion. Not just mention it by a name such as "black and white." And the definitions of some of these names are vague and application dependent. Nonetheless gray level images have served as the main trunk in data acquisition in the history of machine vision, computer graphics, image processing, and so on. In this dissertation, gray level images mean images consisted of pixels of one byte each (8 bits per pixel); which means 256 gray levels.

Along with the progress of hardware and software people are eager to pond the door of color information. They try to fully use the information that color brings to them. Photos, images, graphics, videos, and so on, are all in color now. It is inevitable to encounter the need for color information processing. In stereo matching, algorithms that work on gray level images are extended to color images. Color stereo is more accurate than gray level one with more computational cost.[19,4,38] For a simple example, if images are full of colors that yields same gray level values, the stereo will fail in gray level matching but succeed in color matching. But this effect mainly depends on the objects in use.

Several color models are developed for specifying individual color.[61] RGB (red, green, blue) model is useful for hardware such as color monitors and color video cameras. CMY (cyan, magenta, yellow) model fits the need of color printers. YIQ (luminance, inphase, quadrature) model is color TV broadcast standard. HSI (hue, saturation, intensity) model is suitable for image processing algorithms based on the color sensing properties of the human visual system. In this research the RGB model is adopted everywhere. It could be depicted as a cube with one corner at the origin of a 3D coordinate system with three axes x, y, and z; and each of which corresponds to the R, G, and B, respectively. The values of R, G, and B range from 0 to 255. Each image pixel is represented with 3 bytes. Each byte (8 bits) stands for R, G, and B, respectively. So the value is from 0 (black) to 255 (white) in gray level. The gray level image could be obtained from the color image represented in this format by the following formula:

Grey-level value = (Red + Green + Blue)/3 . Equation 2-34

Some researcher also suggest using different color models could enhance the matching accuracy.[19,3] Color channels have their independence from one another but also have relationship with one another. A model that can exploit those properties could possibly yield better results in stereo matching. This needs more research in the human vision system.

For area-based matching methods like SSD (Sum of Squared Difference) methods, some measure has to be computed and minimized. A new measure has been defined to utilize color information to exercise matching.[71,38]

$$SSD(x, y, d_x, d_y) = \sum_x \sum_y \left| f_r(x, y) - f_l(x + d_x, y + d_y) \right|^2 \qquad \text{Equation 2-35}$$

In the above equation, f_r, f_l are image gray levels of the right and left images, respectively; and *(x, y)* is the index in the matching window of a pre-determined size. d_x, d_y are the horizontal and the vertical disparity estimated in the matching. In color stereo matching, a modified measure — *Color SSD* — is defined as the sum of the *SSD*s computed for each of the three color bands:

$$\textbf{\textit{Color SSD}} = \textbf{\textit{SSD}}_{\textbf{\textit{red}}} + \textbf{\textit{SSD}}_{\textbf{\textit{green}}} + \textbf{\textit{SSD}}_{\textbf{\textit{blue}}} \ . \qquad \text{Equation 2-36}$$

(d_x, d_y) that minimizes the color SSD measure determines the best match.

Some people use other kinds of criteria such as mean square error (MSE) to distinguish the best match from the candidate matches:[4,5]

$$MSE(x, y, d_x, d_y) = \frac{1}{n \cdot m} \sum_x \sum_y \left| f_r(x, y) - f_l(x + d_x, y + d_y) \right|^2 \ . \qquad \text{Equation 2-37}$$

Where *n* and *m* are the dimensions of the pre-determined matching window. The difference between color images could be defined as the following equation:

$$f_{r,color}(x, y) - f_{l,color}(x, y) = \sqrt{(R_r - R_l)^2 + (G_r - G_l)^2 + (B_r - B_l)^2} \ . \qquad \text{Equation 2-38}$$

$f_{r,color}(x,y) = (R_r,G_r,B_r)$, and $f_{l,color}(x,y) = (R_l,G_l,B_l)$ are two different color triples. Similarly, (d_x, d_y) that minimizes the color MSE measure determines the best match.

2.3.2 Multiple baseline SIA

From the geometry and the mathematical relationship between the baseline and the disparity (see Figure 2-4 and Equation 2-16) one can see that the longer the baseline is, the more precise the recovered depth is (when the depth and the focal length of the camera are fixed.) The better precision is due to the larger disparity with a longer baseline. On the contrary, the longer the baseline is, the larger the search range associated to the same depth change in stereo matching is, and hence the larger the probability of false matches is. The precision of the matching is derived as follows:

From the triangulation method we have:

$$z = \frac{bf}{d} \quad , \qquad \text{Equation 2-39}$$

where z is the depth, b is the baseline, f is the focal length, and d is the disparity. Suppose a longer baseline $(b+\Delta b)$ is used, and the corresponding disparity is $(d+\Delta d)$. Then we have:

$$z = \frac{bf}{d} = \frac{(b+\Delta b)f}{(d+\Delta d)} \quad , \qquad \text{Equation 2-40}$$

and hence

$$\frac{\Delta b}{\Delta d} = \frac{b}{d} \quad , \qquad \text{and also} \qquad \Delta b = \frac{b\Delta d}{d} \quad . \qquad \text{Equation 2-41}$$

Assume that there is an error δd in disparity obtained from stereo matching. And the recovered depth values of the short and the long baselines are:

$$\hat{z}_s = \frac{bf}{d + \delta d} \quad , \qquad \text{Equation 2-42}$$

and

$$\hat{z}_l = \frac{(b + \Delta b) f}{d + \Delta d + \delta d} \quad . \qquad \text{Equation 2-43}$$

Their errors in depth are:

$$\Delta z_s = \left| \hat{z}_s - z \right|$$

$$= \left| \frac{bf}{d + \delta d} - \frac{bf}{d} \right| = \left| bf \left(\frac{1}{d + \delta d} - \frac{1}{d} \right) \right| = \left| bf \left(\frac{d - d - \delta d}{d(d + \delta d)} \right) \right|$$

$$= \left| \frac{bf\delta d}{d(d + \delta d)} \right| \quad . \qquad \text{Equation 2-44}$$

$$\Delta z_l = \left| \hat{z}_l - z \right|$$

$$= \left| \frac{(b + \Delta b) f}{d + \Delta d + \delta d} - \frac{bf}{d} \right| = \left| bf \left(\frac{1 + \frac{\Delta d}{d}}{d + \Delta d + \delta d} - \frac{1}{d} \right) \right| = \left| bf \left(\frac{d + \Delta d - d - \Delta d - \delta d}{d(d + \Delta d + \delta d)} \right) \right|$$

$$= \left| \frac{bf\delta d}{d(d + \Delta d + \delta d)} \right| \quad . \qquad \text{Equation 2-45}$$

We can see that $\Delta z_l < \Delta z_s$, when $\Delta d > 0$. δd is usually a few pixels and is small compared to the baseline. So the Equation 2-44 and Equation 2-45 could be approximated as:

$$\Delta z_s \cong \left| \frac{bf\delta d}{d^2} \right| \qquad \text{Equation 2-46}$$

and

$$\Delta z_l \cong \left| \frac{bf\delta d}{d(d + \Delta d)} \right| \quad . \qquad \text{Equation 2-47}$$

Divide Equation 2-47 by Equation 2-46 along with Equation 2-41, we can obtain:

$$\frac{\Delta z_l}{\Delta z_s} = \frac{d}{d + \Delta d} = \frac{b}{b + \Delta b} \quad < \ 1 \ \ when \ \ \Delta b \ (\ and \ \Delta d) > \ 0 \quad . \qquad \text{Equation 2-48}$$

This means that a longer baseline yields a more precise result than a shorter one.

The accuracy of matching is kind of a different concept. Suppose we have a repetitive pattern with a period of p, see Figure 2-8.

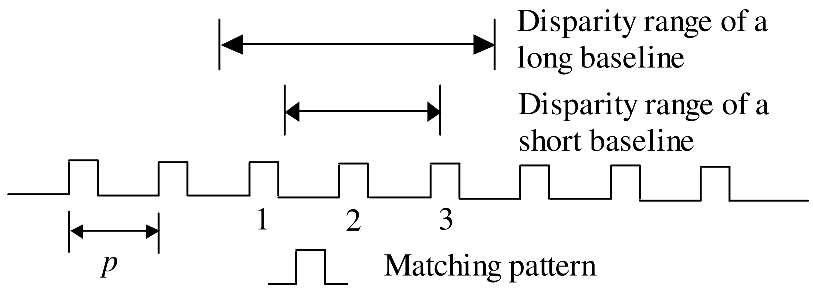


Figure 2-8 Accuracy of matching in a multiple baseline system

Suppose the matching pattern shown in Figure 2-8 lies in a depth range from z_{near} to z_{far}, both measured from the lens. The corresponding range of disparity of a long baseline is D_{long} and the one of a short baseline is D_{short}. I am going to prove that the D_{long} is greater than D_{short} .

Assume that the disparity of a long baseline at z_{near} and z_{far} are d_{ln} and d_{lf}. Similarly the ones of a short baseline are d_{sn} and d_{sf}.

$$d_{\ln} = \frac{b_l f}{z_{near}} \quad , \quad d_{lf} = \frac{b_l f}{z_{far}} \quad , \qquad \text{Equation 2-49}$$

$$d_{sn} = \frac{b_s f}{z_{near}} \quad , \quad d_{sf} = \frac{b_s f}{z_{far}} \quad , \qquad \text{Equation 2-50}$$

where b_l is the long baseline and b_s the short. f is the focal length of the camera lens. Then we have:

$$D_{long} = d_{\ln} - d_{lf}$$

$$= \frac{b_l f}{z_{near}} - \frac{b_l f}{z_{far}} \quad , \qquad \text{Equation 2-51}$$

$$D_{short} = d_{sn} - d_{sf}$$

$$= \frac{b_s f}{z_{near}} - \frac{b_s f}{z_{far}} \quad . \qquad \text{Equation 2-52}$$

D_{long} - D_{short}

$$= \frac{b_l f}{z_{near}} - \frac{b_l f}{z_{far}} - \frac{b_s f}{z_{near}} + \frac{b_s f}{z_{far}}$$

$$= \frac{f(b_l - b_s)}{z_{near}} - \frac{f(b_l - b_s)}{z_{far}}$$

$$= f(b_l - b_s)\left(\frac{1}{z_{near}} - \frac{1}{z_{far}}\right)$$

$$= f(b_l - b_s)\left(\frac{z_{far} - z_{near}}{z_{far} z_{near}}\right) > 0 \quad . \qquad \text{Equation 2-53}$$

So $D_{long} > D_{short}$. In Figure 2-8 we can see that due to the fact that the disparity range (the range for searching the best match) for a longer baseline is larger, it might cover more repetitive patterns and thus generate more ambiguities or false matches. But a short baseline can limit the disparity range into a smaller one so that the matching errors can be less. So the longer the baseline is, the higher the probability of false matches is. There is a tradeoff between precision and accuracy due to different lengths of baselines. Moreover, the larger the disparity range is, the more the computational cost is in stereo matching. And also the depth range — z_{near} to z_{far}, is not usually known. A large range should be assumed to cover the pattern. Therefore the computation is much more even for a short baseline.[37] Note that in Figure 2-8 some factors such as the foreshortening effect is simplified. In this research this effect is a minor one and is not taken into consideration. But there is a clue that the foreshortening effect is less in the case of a short baseline than that in the case of a long baseline. It might be a potential way of alleviating this effect in stereo matching by the multiple baseline approach.

If we would like to achieve the same accuracy as the long baseline but not to spend as much as it does, a multiple baseline system is a good choice. First a rough estimate of depth is obtained by a faster matching through a short baseline setup. Then one can use the estimated depth map as a guide to limit the search range in a long baseline setup. The final precision is the same as that of the long baseline but the accuracy is higher due to the short baseline matching and the computational cost is also decreased in this way. The problem of accuracy (false matches) could still remain even the search range is narrowed in the short baseline case. Its search range could still possibly cover more than one ambiguity. And the initial range of depth (an unknown) could still be too large (a guess or estimate) and cost more than want. One excellent way of giving a good estimate of the depth is apply the IDA/IFA before the stereo matching. Then in the short baseline case one can always limit the search range even more and thus secure the accuracy to the best. Integration of IDA/IFA with multiple baseline SIA and its result are presented in this dissertation.

2.3.3 Multiple resolution SIA

Another improvement in SIA is to use multiple resolution (multiple scale) SIA. Multiple resolution SIA is a coarse-to-fine approach. “Coarse” means lower spatial resolution or lower spatial frequency.[13,7,61] It often relates to blurring or smoothing the pictures or filtering with low-pass filters. In feature-based matching some details of high frequency are filtered out and thus “coarse” images are obtained. Matching in these coarse images yields sparser depth map and reduces the computational time. With the help of the sparser depth map as a guide one can match images of higher spatial resolution, say, “finer” images, more efficiently.

One obtains smoothed images by filtering the images with filters such as mean filters. In area-based matching the computation of the matching will remain the same if smoothing is the only process applied to the images. One can obtain a lower-resolution depth map by enlarging the size of the matching block. But this way will not improve the speed much. Another way to do is to reduce the size of the images with the size of the matching block remains the same. Resampling (reducing the size of) the images by a factor of 2,4, or 8 yields images of a smaller size. With the matching window remaining unchanged, the stereo matching is done on a pair of smaller-sized images. The computation is reduced proportional to the reduction of the size of the images. In this research, the data is resampled on a grid. Its effect is just to extract one pixel from two (assuming that the resampling factor is 2.) One pixel is simply skipped without processing. In this research the result is good enough as an initial estimate of the depth map even without smoothing the images first. But in the case of larger resampling factors, say, 4,8, or larger, it might cause problems due to undersampling. One way of making use of all the data while generating lower resolution images is to combine filtering with size-reduction. For example, for each block in the resampling grid, all the pixels are first filtered by filters such as a mean filter, a median filter, or a gaussian filter, and are then reduced to be one pixel. This way will use more of the information in the higher resolution images.

The lower resolution depth map obtained from matching the lower resolution images is then expanded by the same factor used in resampling and is used as a guide in matching in higher resolution images. This guide further reduces the search range in the higher resolution matching and hence reduces the computation even more. The result shown in later chapters proves the usefulness of the multiple resolution SIA.

2.3.4 Background removal and occlusion detection in SIA

In traditional matching algorithms all the pixels or blocks in the image are matched. Every part of the image is treated the same and is matched. But in some cases the objects that we are interested in can be distinguished from some background scene (which is not as interesting as the objects.) For example, in order to recover the shape of a cup, a cube, or a ball, the object is placed before the camera and a vertical board is placed behind it as a background. The object is the foreground part whose shape is our interest but the background plane is not what we want. It will be a waste trying to match the irrelevant background such as a vertical flat plane placed behind the objects in interest or just the whatever scene behind them.) Moreover, matching errors will occur due to occlusion of the background scene from the foreground objects. And these errors are usually severe and should be removed in order to obtain good recovered shape. In this research a new method for removing the background from matching is developed and the computational time and the probability of false matches due to occlusion is reduced.

Not only the background should be removed from matching but also the occlusion caused by object itself. If one part of the object is hidden from the viewer by another part of the object after the camera is moved to the other stereo position, say, the left position, then the matching will fail and errors occur due to the occlusion. Before one can remove the occlusion errors the occlusion should be detected first in the matching. The occlusion detection should tell the user where there might be occlusion and let the user do some correction to it. Detecting the occlusion and posting solution to it are two different steps. The user could do interpolation, extrapolation, and other techniques to solve the occlusion problem once it is identified.

Background removal

Occlusion is usually due to abrupt depth changes among objects (usually happens at borders of objects.) Abrupt depth changes violate the *global and local smoothness constraints* adopted by most of the stereo matching algorithms. If abrupt depth change happens, some part of the objects cannot be seen in both stereo images (or will be hidden from each other.) This usually causes severe problem in stereo matching. For example, the background (assumed to be an upright flat planar object behind the foreground object,) will have a depth discrepancy between itself and the foreground object, say, a dummy head object. This abrupt depth change will cause occlusion at the border of the head object and the background. It is necessary to separate the background from the foreground to avoid the false matches due to the occlusion at the border of the object.

The IDA and IFA methods recover the shape of the scene without encountering the correspondence problem and also no associated occlusion problem exists. Generally speaking, one can extract any slice of data of any given depth by intersecting a plane of that given depth with the recovered depth map (viewed as an 3D object in space.) The plane is assumed to be perpendicular to the optical axis of the camera. The intersection is the part of the object at the given depth. This depth-extraction operation could be formulated as:

$$If\ D(m,n) = depth\ then\ DE(m,n) = 1 \quad else \quad DE(m,n) = 0 \quad \forall\, m, n \quad . \qquad \text{Equation 2-54}$$

Where $D(m,n)$ is the recovered depth map from IDA/IFA, $DE(m,n)$ is the result of the depth extraction, and (m,n) is the indices of the depth map. Due to the noise in data acquisition and camera electronics, the extracted data might be erroneous compared to the true data of a given depth. In order to cover all the true data, depth extraction in a given range is preferred. Therefore Equation 2-54 is rewritten as:

$$If\ D(m,n) \in dept\_\ range\ \ then\ DE(m,n) = 1 \quad else \quad DE(m,n) = 0 \quad \forall\, m, n \quad .$$

Equation 2-55

The extracted depth data is then processed for other purpose.

In the same way one can extract any part of the depth data from the depth map obtained from IDA/IFA. If we know two groups (or more than two groups) of objects with distinguishable sets of depth information, we can then separate them from each other by the depth extraction method discussed above. The sets of depth are supposed to have distinct displacement in depth from each other so that depth extraction with some allowable range of error will be applied without any overlapping problem. Or ambiguities will occur in object group separation.

In such a way the background could be separated from the foreground. Detailed algorithm, implementation, and results of the integration of the background removal method into SIA will be given in chapter 5.

Occlusion detection

Since the background is removed from the stereo matching. All the matching is now done on the true object. But the occlusion is still existent because object could be occluded by itself. Moreover, the depth change of the foreground object around the border near the background is still high and can still cause occlusion-like problem. One way should be found to detect these problematic places. The user can then choose how to deal with these occlusion errors once they are identified.

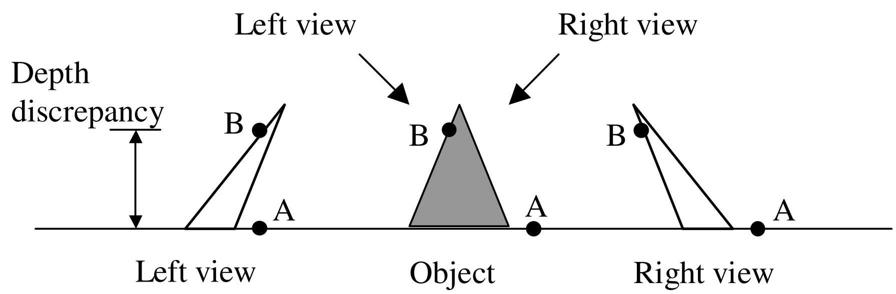


Figure 2-9 Depth discrepancy in occlusion

Suppose that when occlusion happens, one object A in the right image is occluded by another object B in the left image (see Figure 2-9.) It tries to match object A to object B according to the search range in stereo matching process. And one false match around object B is found. Since object A is occluded by object B, object B should be closer to the camera than object A. There will be depth discrepancy between object A and object B. The false match lies around the object B so its depth is different from that of object A. With IDA/IFA one can find the estimated depth of the false match in the left image. It will be different from that of object A in the right image. Using the estimated depth of the false match to find the estimated search range and match the false match in the left image back to the right image, the match found there should drift from the original block (object A) in the right image because of the different depth values used for setting the search ranges in each of the two stereo images. If the center of the back-matched block is shifted from that of the original block over a predetermined *occlusion threshold*, then the match around object B is considered as occlusion. If the shift is under the occlusion threshold, it is considered as a minor "false match."

The shift between the original matching block and the back-matched block is denoted as S_{bk}. How to choose the occlusion threshold O_{th} is a question in occlusion detection. One reasonable value is half of the matching block size. The reason for choosing this value is that once the back-matched block is determined, this block is represented by its center. The coordinates of this center are used to compare with the center of the original matching block and the shift is calculated. If the shift is larger than half the matching window size, the center of the back-matched block falls into another block other than the original matching block. It is assumed here that an off-grid block's depth is determined by the depth of the on-grid block in which the center of the off-grid block lies. According to the local smoothness constraint, the depth inside one block is assumed to be constant and its neighboring blocks will possibly have different depth from the one that it has. Now that the center of the back-matched block has fallen into another block instead of the original matching block, the depth is considered as being changed from the original one. The larger the shift is, the more severe the depth change is. So it is a reasonable choice to set the O_{th} to be half of the matching block size. Once the shift is larger than the O_{th}, it means a depth change and therefore possible occlusion; if the shift is smaller than the O_{th}, then the depth is not changed and no occlusion is found.

Details about the algorithms, implementation, and results of the occlusion detection are presented in chapter 5.

2.3.5 Match In Blur SIA (MIB)

In traditional area-based stereo matching, two stereo images are taken by a stereo camera system. The matching is then applied on these two images. These two images are usually focused at their centers of the images just like ordinary pictures. Off-center part of the image is defocused in different degrees depending on its depth. This kind of matching could be called "*blur match in blur* (BMIB)." It means that the two stereo images both have blurred part (of course there is focused part in them.) Depending on the lens capability of the camera the blur part varies in size and degree of defocus (related to the *depth of field* of the lens.) In an image taken for a cone object (the tip of the cone is set towards the camera,) one can see a ring-like depth of field (the part in focus) in a convex lens camera. This effect could be easily seen in a microscopy system.[33] Longer focal length and larger aperture size decrease the depth of field.

Another type of stereo matching, which is used in this research, is called "*focus match in focus* (FMIF)." After applying IDA/IFA process, focused stereo images are recovered and used in stereo matching. Every part of the images is in focus and there is no blurred part in either of the two stereo images. This type of matching excludes the effect of the depth of field. No matter how large the aperture is and how long the focal length is, stereo matching is always done on focused images. Of these two types of stereo matching, can one tell which one is better? I am going to discuss this problem in the later sections and then introduce a new type of stereo matching.

We remember the need of enough image details in area-based (correlation-based) methods of stereo matching. Blurred images have less image details than focused images. With same object pattern, focused images preserve the best contrast of the image texture. So matching in focus prevails matching in blur at this point. Next, focused images have more high frequency details than blurred ones. These high frequency details usually correspond to edges and sharp details in an image. These can also be viewed as parts that are fast-varying in image brightness or contrast. Suppose we have two image pairs, one pair is focused and the other is defocused. We want to find out the minimum SSD value in stereo matching on these two pairs of images. Assume the brightness pattern at some location on the right focused image and the pattern on the right defocused image at the same location are as those depicted in Figure 2-10. Note that the brightness pattern is simplified in one dimensional for conceptual demonstration but it could be extended to represent a two dimensional pattern within a small patch of the image. The slope of the brightness pattern of the right focused image is assumed to be larger than that of the defocused one. It represents the fast-varying property of the fine image details of high spatial frequency.

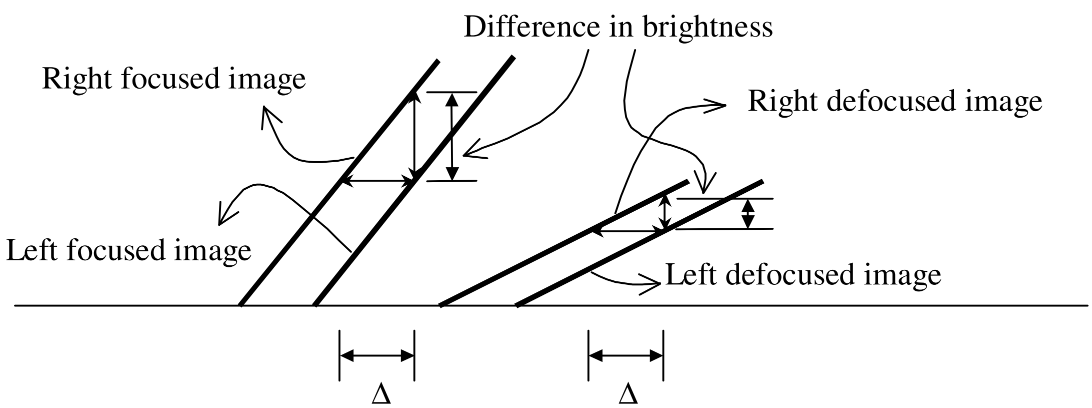


Figure 2-10 Difference in matching in focused and defocused images

While matching on the focused images, the right pattern is used to find its match along the epipolar line in the left image. Ideally speaking, if the match is found and is correct, the difference between them is zero (calculated at each pixel location in the pattern block.) Due to noise and the foreshortening effect they will not be exactly identical and the difference will not be zero; at most close to zero. But the difference should be the minimum supposed that there is no repetition of the right pattern in the search range along the epipolar line in the left image. While the right pattern moving away from the correct match, the difference at each pixel location will deviate from zero (or the minimum value; generally speaking, not a fixed one.) If the difference at each pixel location is taken as an absolute value or is squared up, their sum along the pattern will increase as the pattern slides away from its true match. They become less similar to each other. The match with the minimum *SSD* is selected as the best match.

What happened to the defocused images? They can be analyzed in just the same way as the above. The difference between these two image pairs is that one pair has more image details than the other. We can see again from Figure 2-10 that while moving the right pattern away from its match in both cases, suppose that a displacement Δ is introduced, the difference thus produced in the focused pair is larger than that in the defocused pair. This property will make the SSD value obtained in the focused images larger than that in the defocused ones. Although at the correct matches the SSD values in both cases are ideally the same, i.e., zero. We have a *sharper* SSD curve (with a minimum at or close to zero) along the searching path in the left focused image. This implies a more precise result at finding the match.[37] And apparently it is also less sensitive to noise.

The above discussion compares FMIF and BMIB. And some advantages of FMIF over BMIB are presented. In BMIB only a small part of the images are in focus. So the overall performance is expected not to be as good as that of FMIF. But it still has some advantage. Consider the focused part in a typical center-focused image. While finding its match in the left image, in its search range only around its match are focused. The SSD value will be large when it tries to match the blurred part along its search range. Only its correct match will give a smaller SSD value. In this way one can see that the blurred part somewhat helps the stereo matching by excluding itself from the candidacy. But this happens only in the focused part of an image. And that part is usually small in most cases. Based on this idea one new matching type is developed in this dissertation. It is called "*focus match in blur* (FMIB or just MIB.)"

In IDA/IFA the focused images at both camera positions are constructed. Instead of matching in the left focused image, one will match in a blurred image taken at the same lens position (represented as a step number) as that of the matching block in the right focused image. Different matching blocks in the right image have different focusing step numbers. Their correct matches found in the corresponding left images picked in the above way will also be in focus as they are because they both lie at the same distance (have the same step number) from the camera. But other part in their search range will be out of focus. So the matching is improved in this way. And the advantages of the FMIF are preserved. Unlike the BMIB, all the matching blocks in the right image are in focus. Better matching performance is anticipated.

Some experiments and results are presented in chapter 4. Some more discussion and comments are also provided.

Chapter 3 Integration of Defocus and Focus Analysis with Stereo for 3D Shape Recovery

3.1 Introduction

In the previous chapter the theories of IFA, IDA, and SIA are presented. In this chapter I am going to present the implementation, experiments, and results of the backbone of the integration of the IDA, IFA, and SIA. Details about system setup, camera calibration, procedures of integration, experiments, and results are elaborated in this chapter. Other implementations of SIA such as multiple baseline SIA, and so on, that are integrated into the backbone discussed in this chapter will be presented in later chapters. Hardware and software updating is also included in later chapters.

The method for integrating IDA, IFA, and SIA is implemented on a vision system named SVIS. Several experiments have been done on this system and good results are generated, which will be displayed in this chapter.

3.2 Camera System

The integration of IDA, IFA and SIA was implemented on a camera system named Stonybrook VIsion System or SVIS (see Figure 3-1.) SVIS is a vision system built in the Computer Vision Laboratory, State University of New York at Stony Brook. SVIS consists of a digital still camera (DELTIS VC 1000 of Olympus Co.). S-Video signal from the camera is digitized by a frame grabber board (Matrox Meteor Standard board). All processing is done on a PC (Intel Pentium, 200 MHz.). The camera is mounted on a linear motion stage driven by a stepper motor (X-9 stage and MD-2 stepper motor system of Arrick Robotics Inc.). The camera is mounted such that its optical axis is perpendicular to linear motion of the stage. The right and left images for stereo disparity analysis are obtained by moving the camera to different positions and recording images. The stepper motor that moves the camera is controlled through a parallel printer port on the PC. Focusing of the camera is done by a motorized lens system inside the camera. The lens motor is controlled from a serial (RS-232) communication port on the PC. The Matrox frame grabber installed in the PC is used to record 480×640 size monochrome images with 8 bits/pixel. A user friendly windows software interface has been developed to control the whole system under MS Windows 95 OS. It includes convenient controls for manipulating (i) the lens system, (ii) digitizer board, (iii) linear motion stage on which the camera is mounted, and (iv) all the application programs. In addition, an overhead projector is used to project a high contrast pattern onto 3D objects that have low contrast.

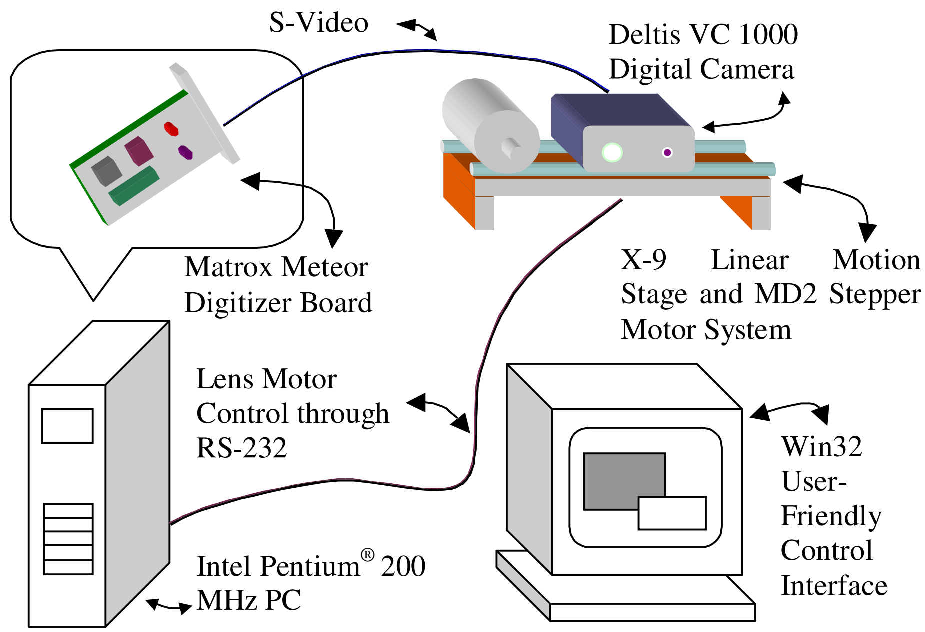


Figure 3-1 SVIS

The camera lens system has separate controls for zooming and focusing. Zooming can be varied from a focal length of 10.2 mm (WIDE mode) to 19.6 mm. (TELE mode). The experimental results reported in this dissertation were carried out with a zoom focal length of 19.6 mm (TELE mode). Focusing is done by driving a stepper motor that controls lens position with respect to the image sensing CCD in the camera. The stepper motor has step positions ranging from 70 to 170. Objects at infinity are focused when the lens stepper motor is at step number 70, and objects close by (about 25 cm) are focused when the lens position is at step 170. Since each lens step number corresponds to focusing objects at some unique distance, we often use this corresponding step number to specify the distance of objects. If an object is said to be at a distance of step X, it means that the distance of the object is such that the object would be in best focus if the lens is moved to step number X. Specifying distance of objects in terms of lens step numbers is particularly convenient in IDA and IFA.

3.3 Camera Calibration

The internal parameters of SVIS such as focal length, lens to CCD distance, aperture diameter, baseline distance, etc. were not known accurately. Therefore SVIS had to be calibrated with respect to four important factors. The first was change in image magnification as the lens moved from step number 70 to 170 for autofocusing. This is important because correspondence between different image regions is needed between image frames recorded with different lens positions. This facilitates comparison of focus measures computed in different image frames to find the image frame in which a given image region is in best focus. The second factor needing calibration was the relation between the distance of objects from the camera and the corresponding lens step number at which the objects would be in best focus. The third factor needing calibration was a blur parameter needed for image defocus analysis. Finally, calibration was needed to establish a relation between object distance and the corresponding stereo disparity. Calibrations with respect to each of these four factors were carried out and four tables were created to represent the calibration data. The calibration procedures are described below.

3.3.1 Magnification Factor

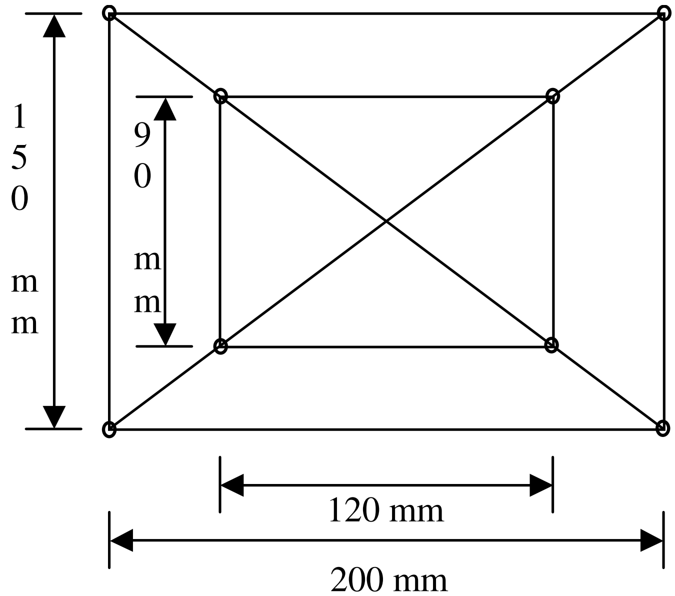


Figure 3-2 Calibration pattern

The magnification at lens position of step 70 was taken to be 1 unit and the magnification at other lens positions were estimated as follows. A planar pattern (see Figure 3-2) was placed normal to the optical axis at about 600 mm from the camera. The pattern consisted of two rectangles of sizes — 200 mm width × 150 mm height, and 120 mm width × 90 mm height — with a common center. The image pixel coordinates of the corner points of the rectangles, the center, and the width and height (in pixels) of the rectangles were recorded as a function of lens position in step number. This was made possible by the interactive Windows interface. The image pixel coordinates of the corner points were recorded by visually pointing the mouse pointer to the perceived image point. When the images were blurred, the corner points spread out into large circular patches. In these cases, the location of the image points were found by visually estimating the center of the circular patches on the computer monitor and pointing the mouse pointer there. The pixel coordinates noted by us had an error of about ± *1* pixel. The lengths of the diagonals of the rectangles were computed from the pixel coordinates of the corner points. The length of the diagonal was a maximum at step 70 and decreased monotonically.

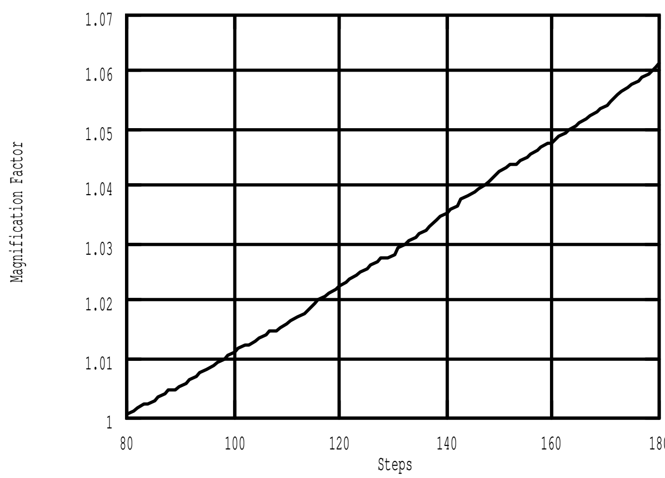


Figure 3-3 Magnification Factor vs. Steps

The magnification factor at a given step number was obtained by dividing the length of a diagonal at step 70 and the length of the same diagonal at the given step number. These ratios obtained for different diagonals were averaged to obtain the final estimate of the magnification factor. This mean ratio was recorded at intervals of roughly 10 steps each. Between these intervals, the magnification factor was estimated by linear interpolation. A plot of the magnification factor as a function of step number is shown in Figure 3-3.

Any image recorded at a given step *s* was magnified by the magnification factor corresponding to that step. Thus, images recorded at different lens step positions had the same magnification as that for step number 70.

3.3.2 Lens step vs. focusing distance

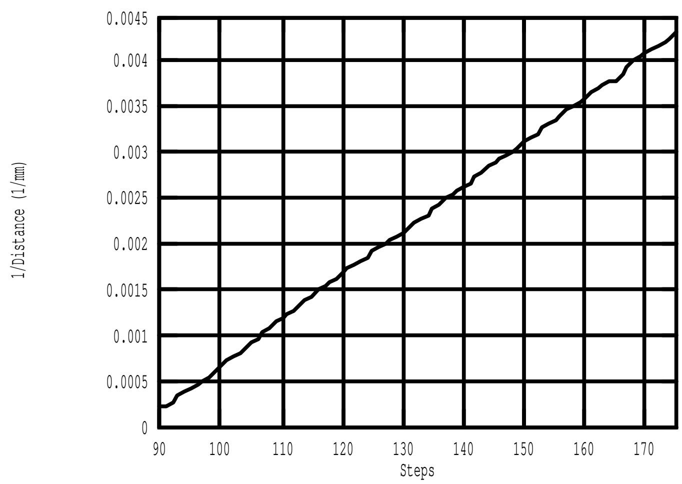


Figure 3-4 1/Distance vs. Steps

The lens position in step number and the corresponding focused distance was obtained using an autofocusing algorithm as follows. A large planar high contrast object was placed normal to the camera's optical axis at a known distance from the camera. The camera was then autofocused by maximizing a focus measure. The lens position in step number that resulted in a maximum focus measure was found by a binary search type of algorithm. The image used was the central 128 × 128. The focus measure was the sum of square of Laplacian of image gray-level. Then the focused step position and the distance of the object from the camera were recorded. This procedure was repeated for many different distances of the object corresponding to roughly 5 step intervals for the focused lens position. Then the gaps were filled by linear interpolation with respect to step number and reciprocal of object distance. This calibration data was used in finding the distance of object points given the focused lens step number for the object points (see Figure 3-4).

3.3.3 Blur parameter vs. focused lens step

The IDA used in the implementation in this chapter is based on a spatial domain approach proposed in [40]. In IDA, only one camera parameter, the lens position (step number) was varied in acquiring the two needed images. All other parameters (focal length and aperture diameter) were nearly constant. In this case we find that a blur parameter σ_2 (which is proportional to the diameter of blur circle) is related to a quantity G' that can be computed from the two recorded images by:

$$\sigma_2 = \frac{G' - \beta^2}{2\beta} \qquad \text{Equation 3-1}$$

The camera constant β in the above equation is a function of the two camera parameter settings at which the two images are recorded. It can be computed if the camera parameters are known. Since they were not known, it was determined experimentally as follows.

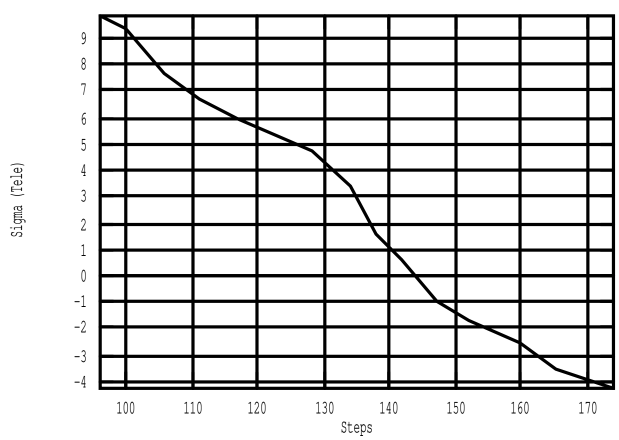


Figure 3-5 Blur Parameter vs. Steps

The IDA[40] was implemented with two images recorded at lens positions of step 120 and 155. An object was placed at such a distance that it was focused when the lens was at step 120. In this case the blur parameter for the image recorded at step 120 is zero, but the blur parameter σ_2 for the image recorded at step 155 is $-\beta$. Therefore β is obtained directly by computing the square root of $-G'$ (a quantity that can be computed from the two observed images) and the sign of β is negative (note that in this case the sign of G' will be negative). This method yielded $\beta = -2.0$ for focal length at WIDE mode end and $\beta = -4.0$ at TELE mode end. β can also be computed by placing an object at a distance corresponding to the focused lens position 155. In this case, the blur parameter for the second image is zero, but that for the first image is equal to β. Therefore β can be estimated as the negative value of the square root of G'.

Another calibration table is needed that relates the blur parameter of an object in the second image (recorded at step 155) to the focused lens step number of the object. This table is created as follows. First an object is placed at a given distance and the IFA method is used to autofocus the object by a binary search for the maximum of a focus measure. The lens step number that autofocuses the object is recorded. Then the IDA method is applied and the blur parameter σ_2 is calculated. This procedure was repeated for several different objects at the same distance and the average σ_2 and the average focus step number were recorded. This gives one entry of the calibration table. This procedure was repeated for several different object distances at roughly regular intervals in terms of focused lens step number (about 5). The gaps in the table were filled by linear interpolation with respect to lens step number and blur parameter. The resulting data is shown in Figure 3-5.

3.3.4 Disparity vs. Distance

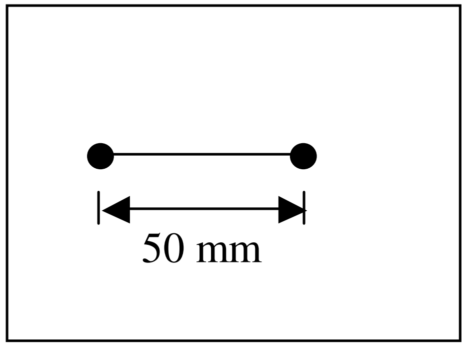


Figure 3-6 Horizontal alignment for CCD array

First the optical axis of the camera was set perpendicular to the stereo baseline by trial and error as follows. The length of the baseline was known to be *50.0 ± 0.2* mm. A line segment of the same length as the baseline (50.0mm) was placed at about 500 mm from the camera such that the line segment was parallel to image rows (i.e. the end points had the same row index coordinates but different column index coordinates). This ensured that the line segment was roughly parallel to the camera baseline. See Figure 3-6 for details. There were two assumptions made in this step. The first was that the line segment itself was placed nearly horizontal in the world. The second was that the image row was nearly parallel to the baseline. The latter was to be verified in the next step.

Figure 3-7 Alignment for CCD array with the baseline

Next the camera was moved to the other position to simulate the two-camera stereo system. If the row index of any point in the right image was shifted up or down in the left image. Then the alignment of the CCD array with the baseline was not exact (see Figure 3-7.) In this experiment the shift was about 3 pixels up in the case of a baseline of 50 mm. This minor deviation of alignment was noted down and compensated later in the software without making any change in the hardware. Also the line segment remained nearly horizontal with respect to the image row after the camera was moved.

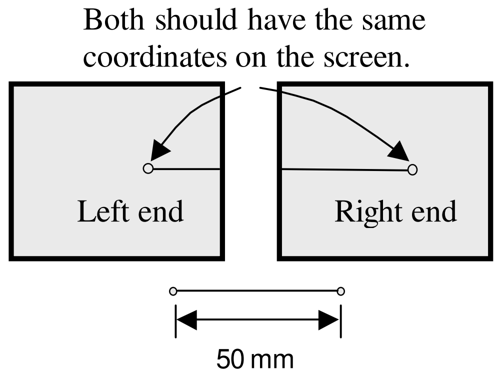


Figure 3-8 Calibration of the Optical Axis and the Baseline

At the last step the same line segment was used as before and the camera was moved from right to left along the baseline. If the coordinates of the right end of the line segment in the right image are the same as the coordinates of the left end of the line segment in the left image, then the optical axis of the camera is perpendicular to the baseline. This fact was used to adjust the optical axis to be perpendicular to the baseline. We believe that an accuracy of ± *2* degree was achieved. Note that the optical axis was assumed to be perpendicular to the line segment in advance.

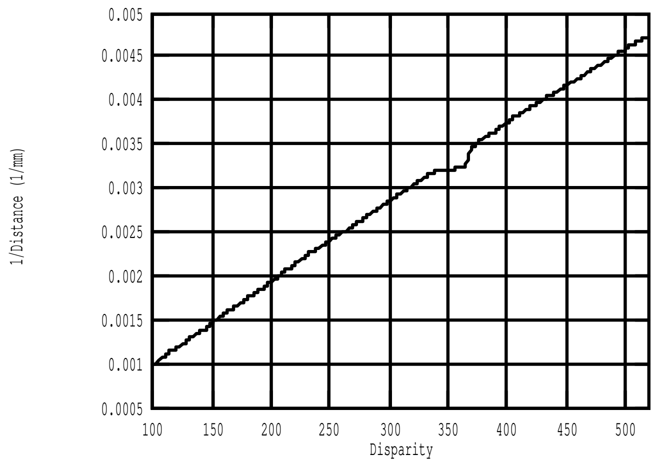


Figure 3-9 1/Distance vs. Disparity

The relation between disparity and distance was obtained as follows. A planar object was placed normal to the optical axis at different distances from the camera. At each distance, the object was autofocused using the IFA method. Then the disparity of a set of points on the planar object was recorded. The points were the corners in a checker pattern. The coordinates of corresponding points in the left and right images were found visually by pointing the mouse to their images on the computer monitor. The average disparity for the points was computed. This disparity was then normalized corresponding to the magnification of the camera for the autofocused lens step position. This is necessary since stereo matching is done on focused images that are reconstructed from magnification-normalized images. The resulting disparity was recorded for many different object distances (in the range 250mm to 1000 mm) at roughly regular intervals of disparity. The gaps in these intervals were filled by linearly interpolating the disparity with respect to reciprocal of distance. The resulting calibration data is depicted in Figure 3-9.

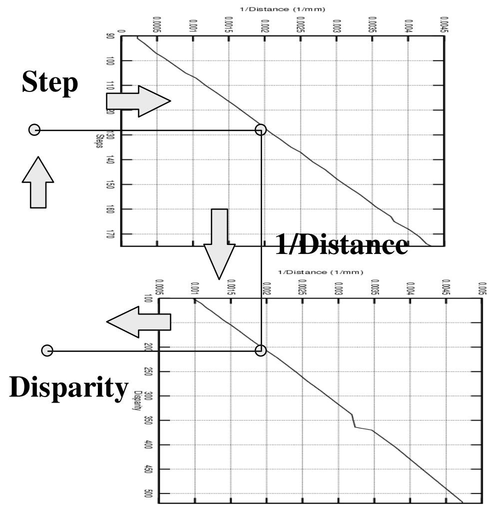


Figure 3-10 Generation of the Disparity vs. Step Table

The calibration data relating the lens step number for autofocusing with object distance, and the data relating disparity with object distance were combined to obtain a table relating lens step number for autofocusing with disparity (see Figure 3-10). The two calibration data were combined using the following method. For each lens step position, the corresponding autofocused object distance was found. Then the reciprocal of this distance was compared to the reciprocal of distance in the disparity vs. distance data for each possible disparity. The disparity corresponding to the minimum absolute difference in reciprocal distances was taken for creating the lens step vs. disparity table (Figure 3-11.)

This table is useful in combining IFA and SIA. If the focused step number of a point is determined using IFA, then rough disparity of the point can be obtained by looking up this table. This rough disparity is then improved through a stereo matching method.

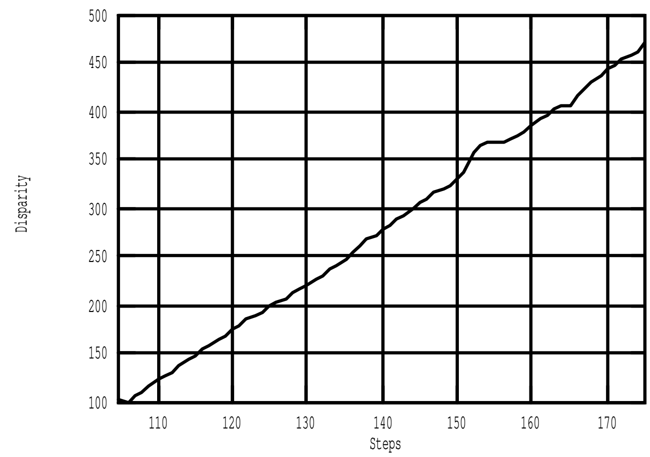


Figure 3-11 Disparity vs. Steps

3.4 Integration

3.4.1 Combining IDA and IFA

First a rough depth-map is obtained using IDA.[40] One estimate of depth is obtained in each image region of size 48×48. The two needed images were recorded at lens steps 120 and 155. Four image frames were time-averaged to reduce noise. The magnification of the images was normalized using the magnification table. The IDA[40] was applied to images of size 432×432. Thus a coarse depth-map array of size 9×9 was obtained. At this stage, the depths were expressed in terms of the lens step number that focuses objects at that depth. In each image region, the actual depth-map at higher resolutions was assumed to be within ± 10 steps of the estimated depth-map step number (see Figure 3-12.) Using this initial depth-map, a higher resolution depth-map of size 27×27 (one estimate in 16×16 image region) was obtained using IFA. The lens step numbers for which image frames needed to be recorded and processed in IFA were determined using the following algorithm. The purpose of this algorithm is to acquire and process only those image frames near the estimated depth-map values. This avoids processing unnecessary image frames in which all image regions are highly blurred (see Figure 3-13).

The estimated depth values (in step numbers) is first quantized to multiples of DEL (=3) steps. Then, for each quantized value s that occurs in the depth-map, $2*numdel+1$ ($numdel=3$) lens step positions at $s+i*DEL$ for $i=0, \pm 1, \pm 2, \ldots, \pm numdel$ are marked. If any of these steps are outside the range of minimum and maximum step positions, they are discarded. Then image frames are recorded at each marked lens position. All images are normalized with respect to the magnification corresponding to the step position at which they are recorded. Then, in the resulting image sequence, focus measures are computed in image regions of size 16×16. The step number where the focus measure is a maximum in each image region is determined. These image regions with maximum focus measures are synthesized to obtain a focused image of the entire scene. Further, the maximum focus measure and the two focus measures in the preceding image frame (DEL steps below) and the succeeding image frame (DEL steps above) are taken. A local quadratic curve is fitted to the three focus measures (the center one being the maximum) and the position of the maximum of the curve is computed. This position is taken as an improved estimate of the depth-map. If the focus measure for the preceding or succeeding image frame is not available, then this last step is not performed (see Figure 3-13).

3.4.2 Integration of IDA and IFA with Stereo

The depth-map obtained above is improved using SIA as follows. The combined IDA/IFA is applied at the right camera position, and a focused right image and right depth-map are obtained. The camera is moved by 50 mm to the left position. Then the combined IDA/IFA is applied at the left camera position, and a focused left image and left depth-map are obtained. Then SIA is applied to the right and left focused images. For each image block in the right focused image, the expected disparity is obtained using the step number vs. disparity table. The range of disparity to be searched for establishing correspondence is obtained by finding the disparity for DELST (=10) steps below and DELST steps above the focused step number. For each image block of size *16* $\times$ *16* in the right focused image, the best matching position in the left focused image is found using *Sum-of-Squared-Differences (SSD)*.[36] *SSD* is computed by finding the gray-level difference between corresponding pixels of the right image block and the left image block, squaring the differences, and summing them (see Figure 3-14). The *SSD* is defined as :

$$SSD = \sum_{x} \sum_{y} \left| f_r(x, y) - f_l(x, y) \right|^2 , \qquad \text{Equation 3-2}$$

where f_r, f_l are image gray-levels of the right and left images, respectively. In this chapter the system is configured to be a two-camera type (actually it is simulated by one moveable camera.) There is no difficulty to extend the stereo system to be one with three or more cameras. In our case, we just need to move the camera to another (the third one) or other more positions along the linear positioning stage. The basic idea of doing this is that the larger the amount of data we have, the easier the matching process will be. Some papers proposed a *trinocular stereo* system in which three cameras were used instead of two.[11] It worked as a simple case of the *multiple baseline stereo* system [36,37] in which *SSSD* was used. *SSSD* stands for "sum of sum of squared differences." It is an extension of *SSD*, in which the *SSD* results from several conjugate image pairs (obtained by trinocular stereo or multiple baseline stereo) are summed up and yield a single *SSSD* result. The *SSD* (or *SSSD*) method is simple and yields good result. And due to its regularity it is easy to implement it on a parallel machine. The extension of the current system to be a multiple baseline stereo system will be discussed in later chapters.

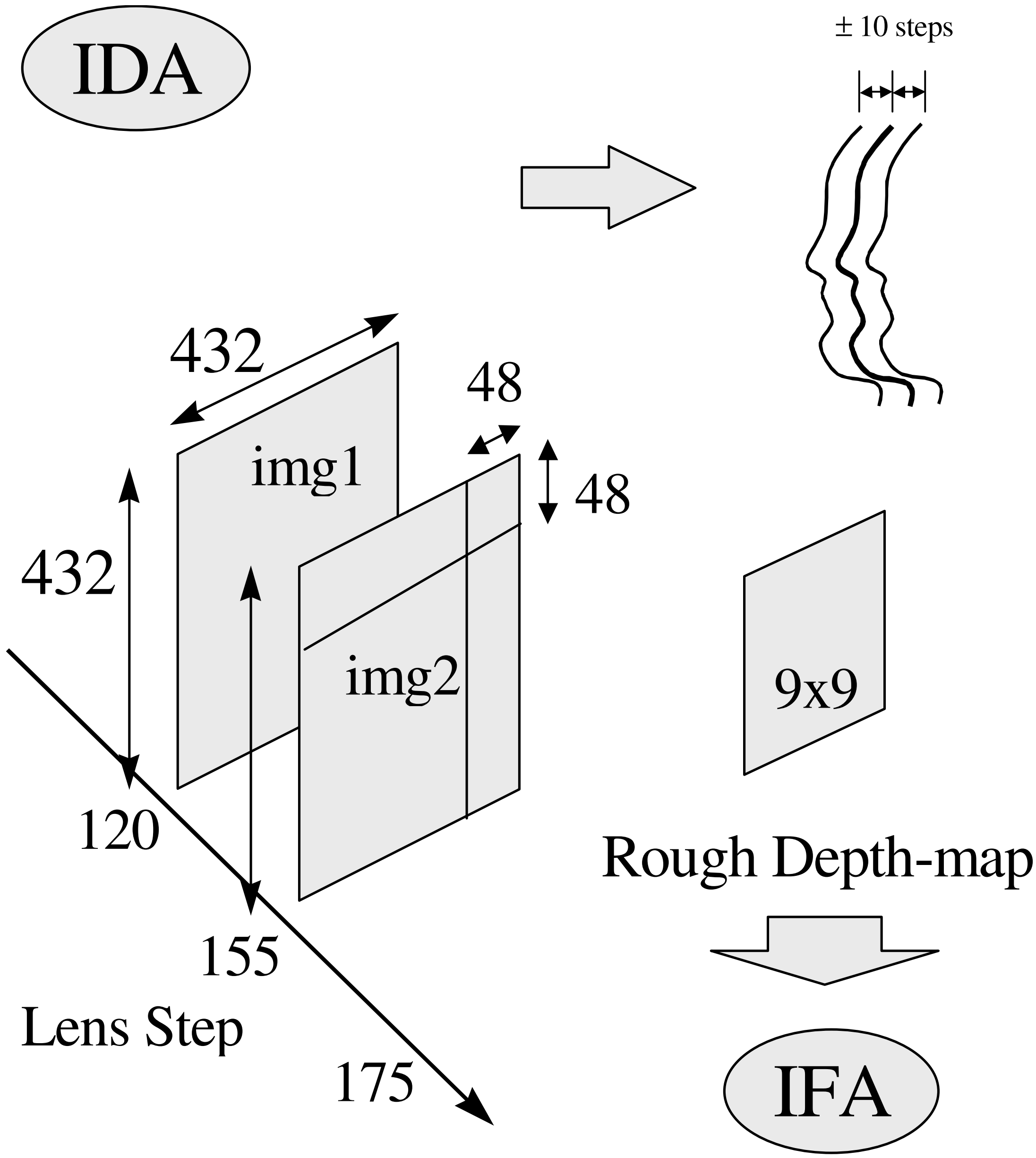


Figure 3-12 Applying IDA

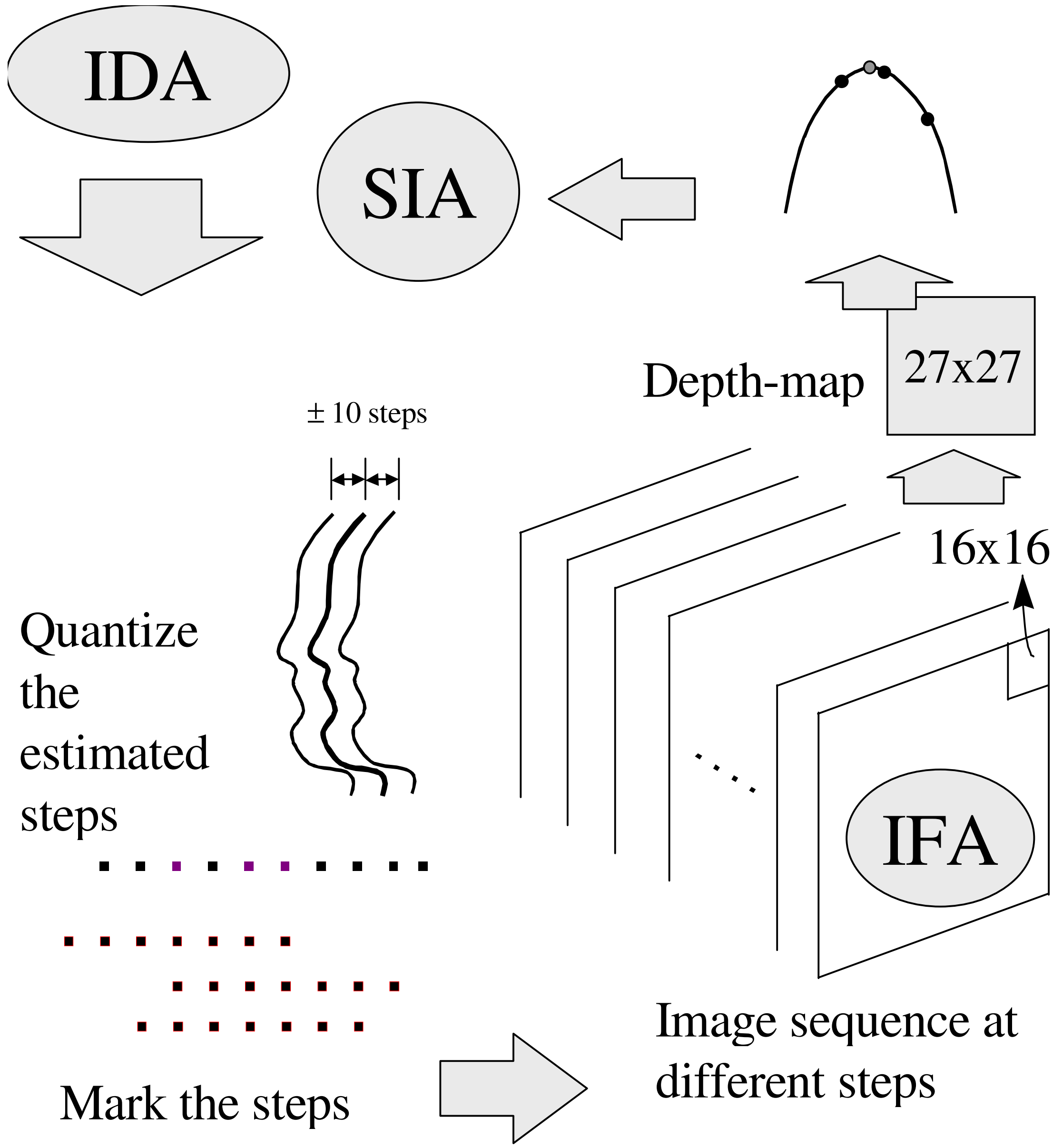


Figure 3-13 Combining IDA/IFA

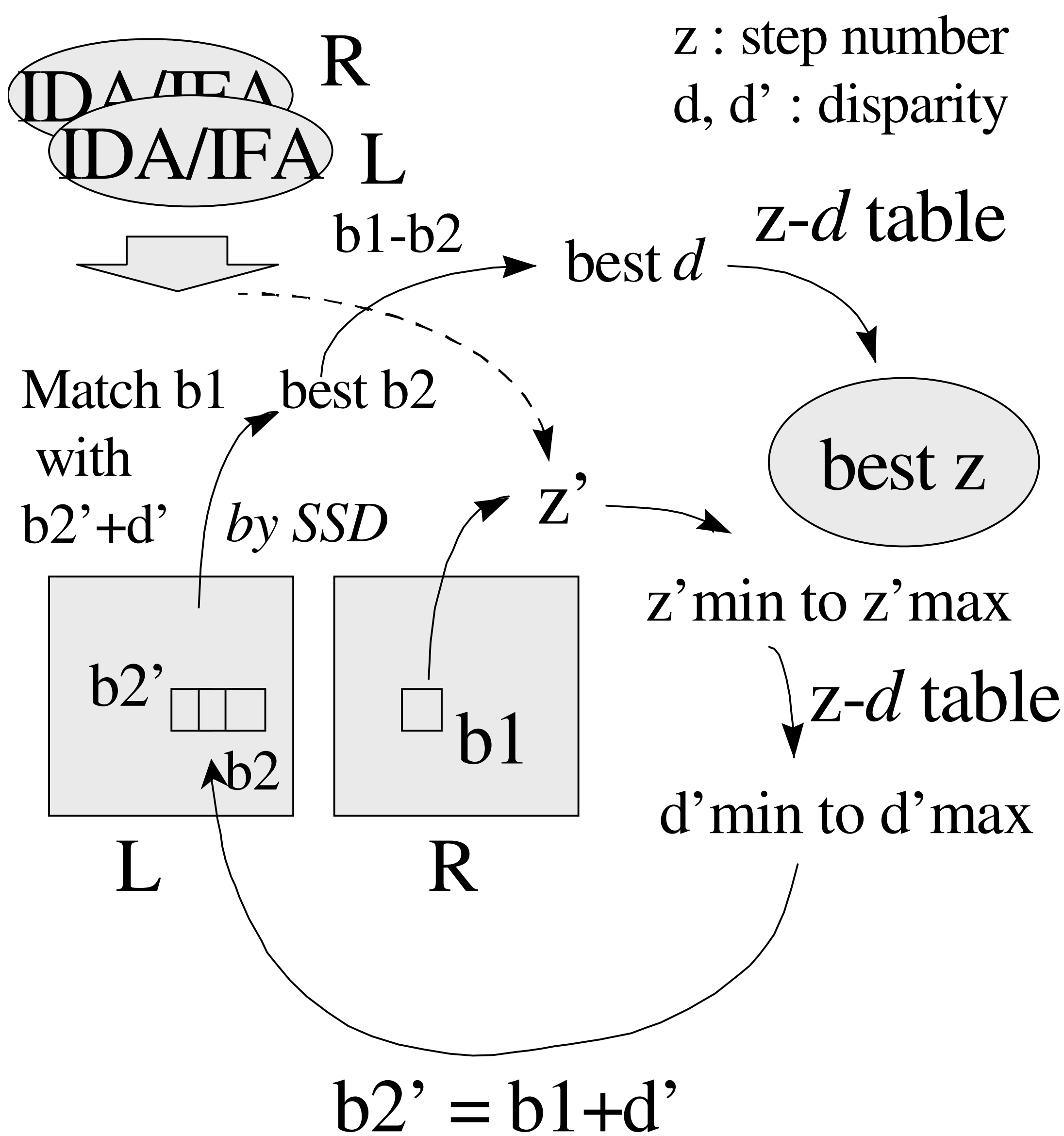


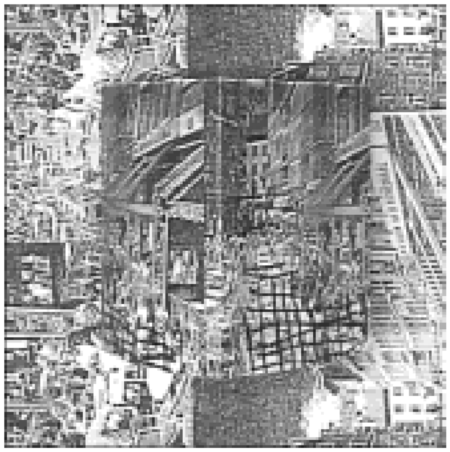
Figure 3-14 Integrating IDA/IFA and SIA

3.5 Experiments and results

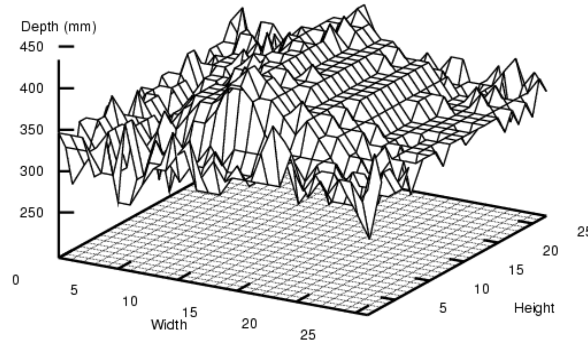
The algorithm for integrating IDA, IFA, and SIA was named DEFOST (DEfocus FOcus and STereo). The implementation of DEFOST was tested on SVIS. The algorithm parameters were set to operate well for objects in the depth range of 30cm to

80 cm from the camera. The results of three of the objects — a prism object, a cone object and a face object — are presented here. The objects were placed about 60 cm distance from the camera. In general, the magnification correction data obtained by calibration was sufficiently accurate for IFA and SIA, but not for IDA. The results of IDA were satisfactory at the center of the image where the effect of magnification change was minimum, but the results were not satisfactory away from the center. This was compensated by allowing for error in IDA and applying IFA in a larger range than necessary, thus increasing the number of images and computing used. It should be noted that shape of objects is obtained by looking for changes in scene depth-map. Therefore the percentage error in depth-map will be much less than that in shape. The focused image and 3D shape of a prism recovered from IDA/IFA are shown at the top of Figure 3-15. The final 3D shape recovered from DEFOST after stereo disparity analysis is shown at the bottom in Figure 3-15. (In all plots of 3D shape in the Figures, X-Y plane (Z= 0) is at 1000 mm from the camera, and Z-axis points towards the camera). These results were obtained by using 2 image frames for IDA and about 10 image frames for IFA for each of the right and left camera positions. The 3D plots of shape are somewhat distorted by the plotting software used in the experiments because the depth-map data obtained by SIA was not corrected for perspective distortion in imaging. The perspective distortion makes a constant size surface patch appear smaller (i.e. projects onto smaller image area on the CCD array) as it moves farther. The DEFOST software implementation was not optimized to minimize computation since the goal was to demonstrate the conceptual feasibility of integrating IDA, IFA, and SIA. Therefore, for each object, shape recovery takes about 3 minutes on the Pentium PC (200 MHz.) We

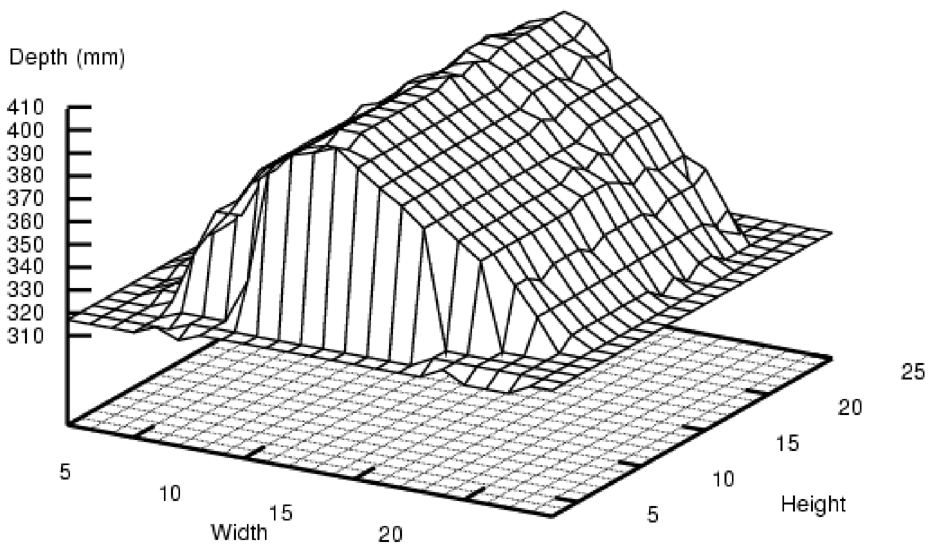
believe that this time can be reduced by a factor of 3 or more by optimizing the software design. Results for a cone object are shown in Figure 3-16, and for a face object are shown in Figure 3-17. In general the results are found to be satisfactory. A precise quantitative analysis of the accuracy of shape is yet to be carried out. Significant further improvements in accuracy can be obtained by using more intelligent stereo matching techniques than the simple *SSD* technique.



Reconstructed Focused Image

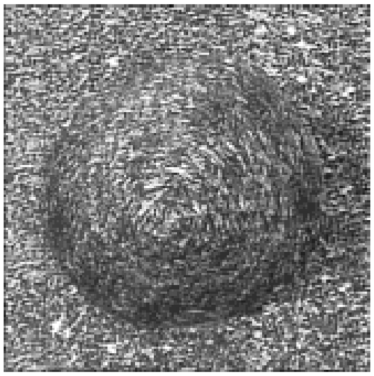


Intermediate Depth-map

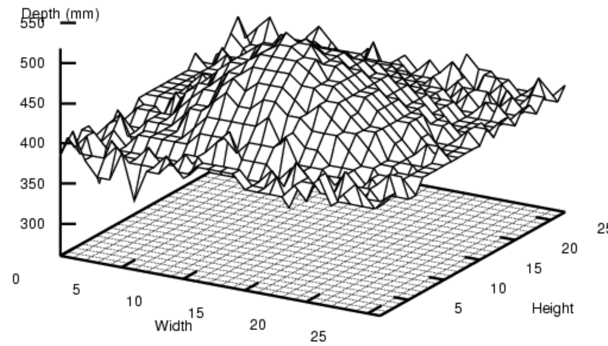


Final Depth-map

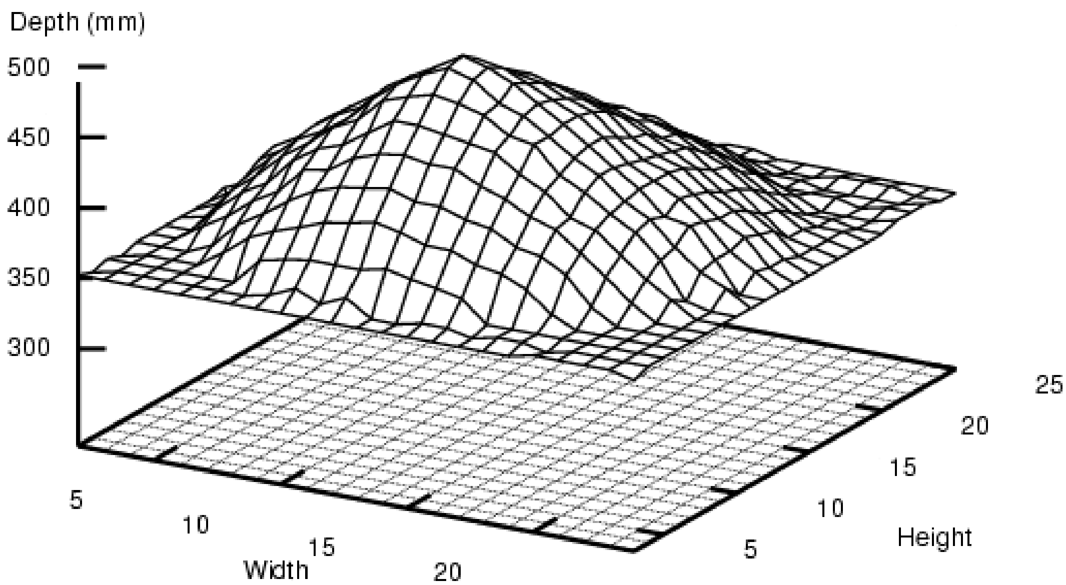
Figure 3-15 Experimental Result for Prism Object



Reconstructed Focused Image



Intermediate Depth-map



Final Depth-map

Figure 3-16 Experimental Result of Cone Object

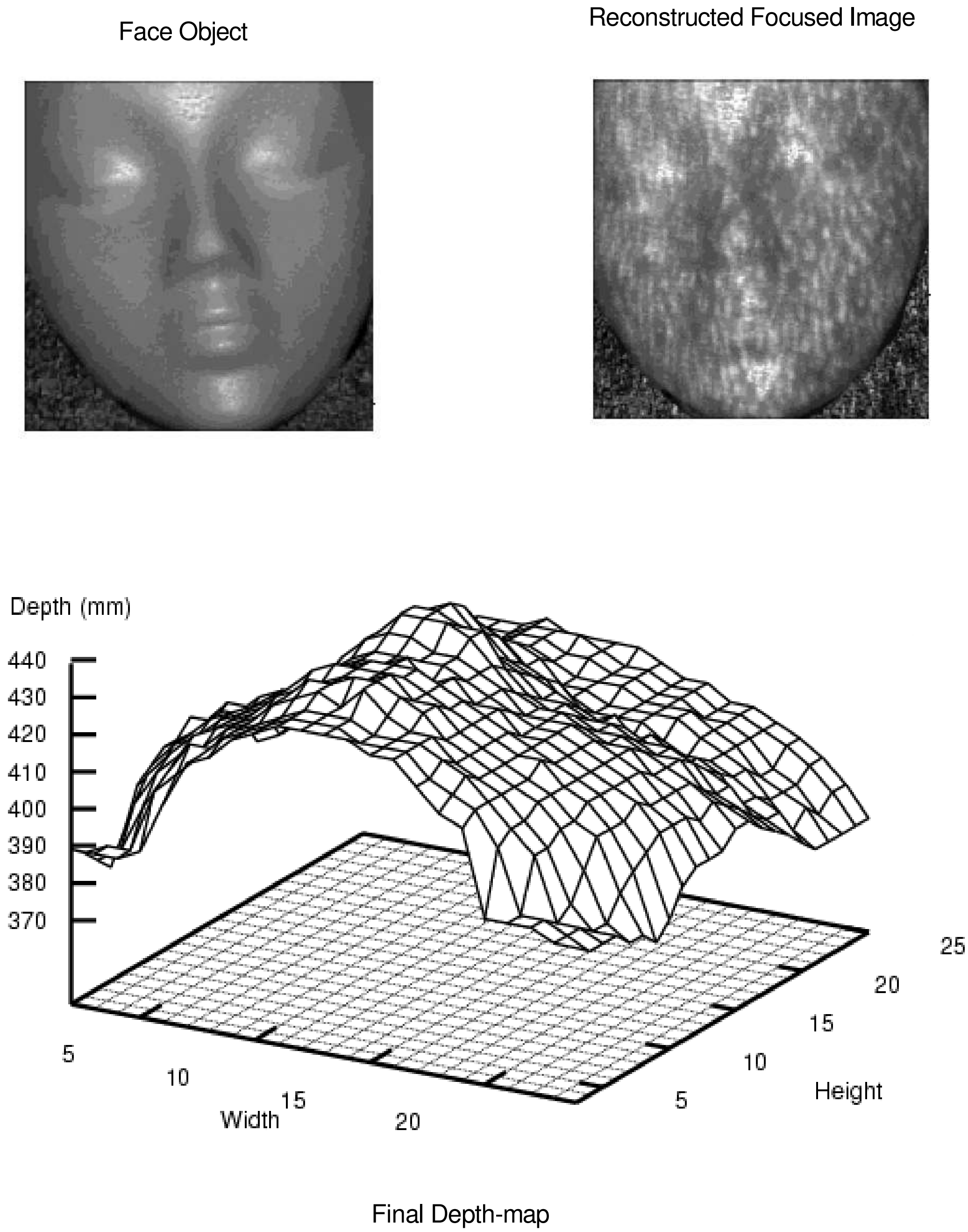


Figure 3-17 Experimental Result of Face Object

3.6 Conclusion

This chapter presents the integration of defocus and focus analysis with stereo image analysis for recovering the 3D shape of objects. The integrated method has been implemented on a vision system — SVIS — and the effectiveness of the method has been demonstrated. The main advantage of the method is the reduction in computation and errors in shape caused by stereo correspondence matching. A rough depth-map provided by defocus and focus analysis decreases the search space for correspondence matching and avoids false matches due to occlusion. It is found that careful calibration of the camera system with respect to several factors are very important in building a successful vision system that integrates defocus, focus, and stereo. Details on these calibration methods are presented. Experimental results are also presented.

Chapter 4 Integration of Multiple-baseline Color Stereo Vision with Focus and Defocus Analysis for 3D Shape Measurement

4.1 Introduction

In the previous chapter the integration of IDA, IFA, and SIA on a stereo vision system (SVIS) is introduced. In this chapter the 3D vision system SVIS for three-dimensional shape measurement is extended to be one that integrates three methods: (i) multiple-baseline, multiple-resolution Stereo Image Analysis (SIA) that uses color image data, (ii) Image Defocus Analysis (IDA), and (iii) Image Focus Analysis (IFA). Besides what is discussed above, a new way of improving SIA is also discussed. It is named MIB (match in blur.) Some experimental results and discussion will be presented in this chapter.

In machine vision, three-dimensional (3D) shape measurement techniques have varying requirements in terms of amount of image data, computational resources, and camera hardware. The techniques yield varying levels of performance in terms of accuracy and speed. The performance of many of these techniques can be improved if an initial rough estimate of shape is available in addition to the required image data and camera parameters. The initial rough estimate of shape can be obtained using another technique with lesser requirements (e.g. less computation or image data). This suggests integrating two or more compatible techniques to optimize the overall performance. In the previous chapter[54] a method for three such techniques — Image Defocus Analysis (IDA),[40,70,28,26] Image Focus Analysis (IFA)[22,62,52,41], and Stereo Image Analysis (SIA)[13,74,37,54]is presented. In this chapter the previous method is enhanced in three respects — (i) using color (RGB) image data instead of monochrome image data in Stereo Image Analysis, (ii) using stereo images recorded at multiple baselines, and (iii) using a multiple resolution stereo matching algorithm.

Color images provide more information than gray-level images and yield more accurate estimates of disparity than gray-level. But color image matching requires about 3 times more computation. In the multiple-baseline SIA, three or more images are recorded at positions along the same baseline. The matching results obtained using shorter baseline images can be used to match longer baseline images with less computation and also less error. In this chapter, implementation of multiple-resolution matching approach in the single-baseline SIA is reported and its effectiveness is shown. And the overall 3D shape measurement technique integrates IDA, IFA, and SIA. It is implemented on the camera system — SVIS (depicted in chapter 3). Details of algorithms and results of experiments on SVIS are presented in later sections.

As introduced in chapter 2, the (F)MIB ((focus) match in blur) has advantages over FMIF (focus match in focus) and BMIB (blur match in blur) due to the following reasons:

1. More image details, higher contrast, and so on.
2. Sharper SSD curve and therefore better accuracy and precision.
3. Less sensitive to noise.

4. Extra limit on the search range.

Some experimental results are presented in this chapter.

4.2 Color stereo matching (Color DEFOST)

The 3D shape measurement algorithm presented in the previous chapter[54] is named DEFOST and it integrates Defocus analysis (IDA), Focus Analysis (IFA), and Stereo Analysis (SIA). Color image data could be used easily in IDA and IFA. For example, at each pixel, the color band with the highest contrast (or highest Laplacian) is determined from among the RGB band images. Then the IDA/IFA is applied to that band to get an estimate of depth at the corresponding pixel. However color data is not used in IDA and IFA in this chapter since the resulting improvement in performance was thought to be marginal.

In DEFOST, single-baseline stereo was implemented using gray-level images. For each small image region in the right image, the best match in the left image was found by minimizing the Sum-of-Squared-Difference (*SSD* [36]) measure defined by:

$$SSD = \sum_{x} \sum_{y} \left| f_r(x, y) - f_l(x, y) \right|^2 . \qquad \text{Equation 4-1}$$

In the above equation, f_r, f_l are image gray-levels of the right and left images, respectively; and *(x, y)* is the index in the matching window of a pre-determined size.

In color stereo matching, *Color SSD* is used. Color SSD is defined as the sum of the *SSD*s computed for each of the three color bands:

$$\textbf{\textit{Color SSD}} = SSD_{red} + SSD_{green} + SSD_{blue} . \qquad \text{Equation 4-2}$$

The matching computation for color images is 3 times that of gray-level images.

4.2.1 Experiments

The Stonybrook VIsion System (SVIS) described in the previous chapter[54] was used in the experiments. All parameters of the camera system were the same as those used in the previous chapter. The DEFOST algorithm for integrating IDA, IFA, and SIA, presented in the previous chapter was modified to use color image data for stereo matching. IDA and IFA were applied to gray-level images computed from color images as

$$\textbf{\textit{Grey-level value}} = \textbf{\textit{(Red + Green + Blue)/3}} \ . \qquad \text{Equation 4-3}$$

The 3D shape of a prism placed about 0.65 meter away from the camera was measured using the DEFOST algorithm with both gray-level and color image data. Objects used in the experiments are shown in Figure 4-1, Figure 4-2, Figure 4-3. The results are shown in Figure 4-4 and Figure 4-5 for comparison. A color random dot pattern was pasted on the prism to create a high contrast image. Instead of pasting a pattern, high contrast images can also be created by projecting a light pattern with color random dots. For the prism object used in the experiments, the improvement in the accuracy of 3D shape obtained using color image data instead of gray-level image data is small.

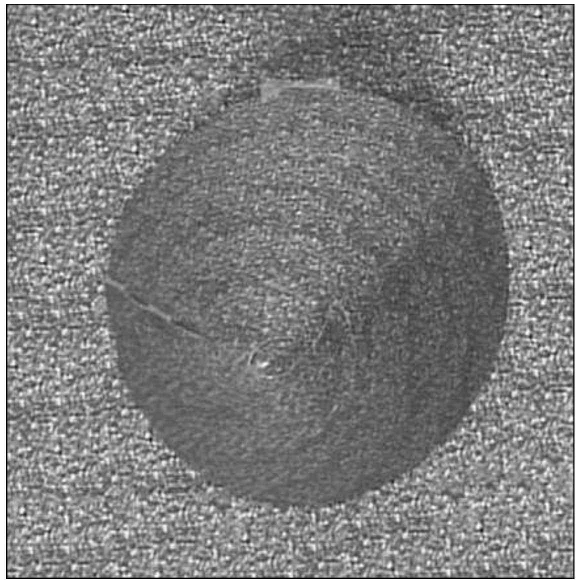


Figure 4-1 Picture of the cone object

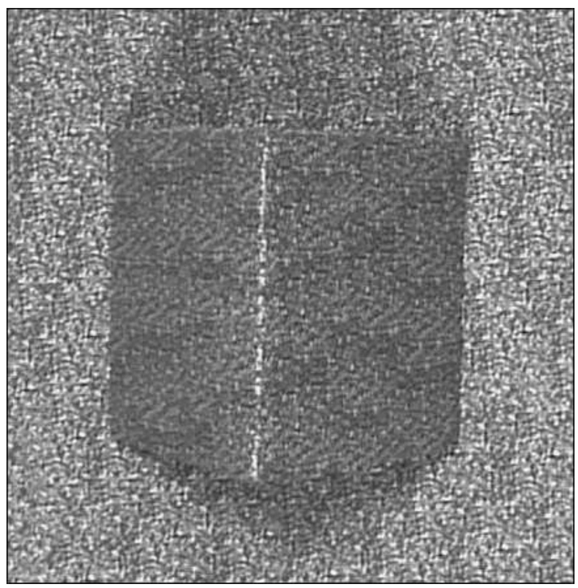


Figure 4-2 Picture of the prism object

Figure 4-3 Picture of the face object

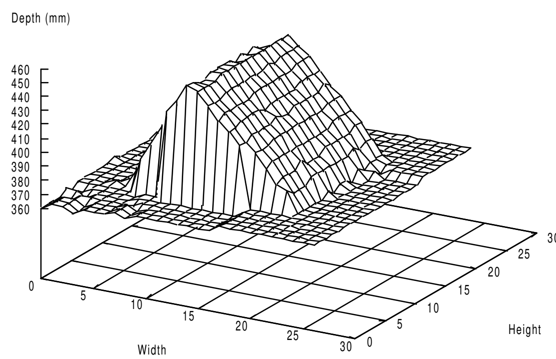


Figure 4-4 Result of DEFOST for the prism object

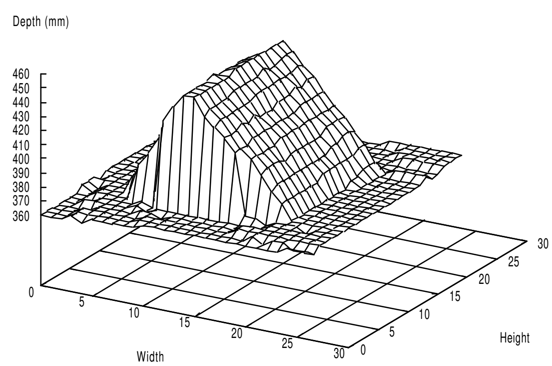


Figure 4-5 Result of Color DEFOST for the color prism object

4.3 Multiple-resolution SIA

Computation time in stereo image matching can be reduced by a coarse-to-fine searching strategy. First a coarse search is made for the best match under low spatial resolution. Lower resolution images are obtained by down sampling the original images. Spatial resolution is typically reduced by a factor of 2 to 4. The disparity-map obtained under low-resolution matching is scaled to original high resolution and used as initial estimates for high resolution matching. In high resolution matching, the search for the best match is limited to a small space along the epipolar line around the initial estimate. The size of the search space is determined by the resolution scaling factor and the expected error in the low resolution matching.

Multiple-resolution matching technique could be applied to color image data by multiple-resolution processing of each band image. However, in this chapter, multiple-resolution matching is applied only to gray-level images. Multiple-resolution image analysis may also be applied to IFA, but it is not considered in this chapter.

4.3.1 Experiments

Multiple-resolution matching algorithm was implemented as follows. First, the IDA and IFA part of the DEFOST algorithm was implemented using the camera parameters and the algorithm as described in the previous chapter.[54] Only the image size was chosen to be 448×448 instead of 432×432 to make the lower resolution image size be a multiple of 16. After applying IDA and IFA, one can obtain a 28×28 depth-map where each depth estimate corresponds to one 16×16 image block. The depth values are expressed in terms of the focusing step number of the stepper motor that moves the camera lens forward and backward. In addition to depth-map, IDA and IFA also yield left and right focused images of size 448×448. These images are shrunk to 224×224 by down sampling the original images by a factor of 2. The depth-map was averaged (in units of focusing step number) in 2×2 blocks to obtain a 14×14 low resolution depth-map. The maximum error in the depth-map was taken to be ±6 steps as in DEFOST. This maximum error was used to look up a calibration table to determine the range of disparities over which a search for best match should be made. This calibration table was obtained by scaling the corresponding table used in the previous chapter. Scaling of the table by a factor of 2 accounts for image shrinking by the same factor.

Stereo matching of the 224×224 low resolution left and right focused images was carried out using 16×16 image blocks and *SSD* as a measure of similarity. The range of search for the best match was determined from disparity range as mentioned above. The result of this step was a 14×14 depth-map array (expressed in pixels of disparity) with improved accuracy. This depth-map was magnified to a 28×28 array by expanding each depth estimate to a 2×2 array and the disparities were scaled by 2. The resulting 28×28 depth-map array was taken to be the initial disparity estimate for the 448×448 high resolution (original) images. The maximum error in this disparity estimate was taken to be ±10 (pixels). These estimates were refined through stereo matching on 16×16 image blocks on the high resolution images using *SSD* as before. The resulting disparities were used to obtain actual depth estimates (expressed in mm) using a calibration table as in the previous chapter.

The results thus obtained (multiple-resolution matching) compare well with the ones obtained using the original DEFOST (only high-resolution matching). Further, as expected, the multiple-resolution approach was found to be faster by a factor of nearly 2. The results for two objects — a prism and a face — are presented (see Figure 4-6, Figure 4-7, Figure 4-8, and Figure 4-9; Figure 4-7 is the low-resolution result and Figure 4-9 is the result of the DEFOST.) The computation times are tabulated below.

Stereo matching	Multiple-resolution		High resolution only
Resolution	Low resolution	High resolution	
Computation time (sec)	6	16	40
Total time (sec)	6+16=22		40

Table 4-1 Comparison of computation time for multiple-resolution and single resolution stereo matching

The speedup is almost 2 (40/(6+16)) in the above case. In the multiple-resolution matching approach, lower the resolution, higher the matching uncertainty and therefore larger the search space at the higher resolution. Also, if the resolution is too low, gross matching errors are possible. Accurate estimation of the matching uncertainty is also difficult.

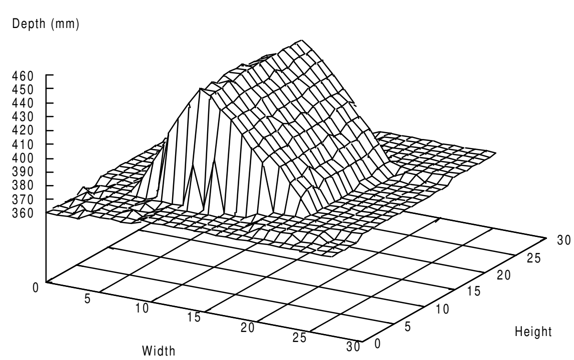


Figure 4-6 Result of multiple-resolution SIA for the prism object

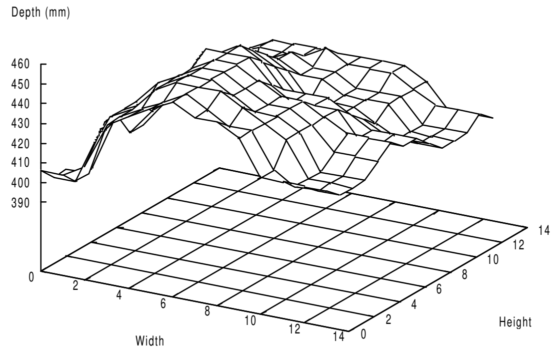


Figure 4-7 Low resolution result of multiple-resolution SIA for the face object

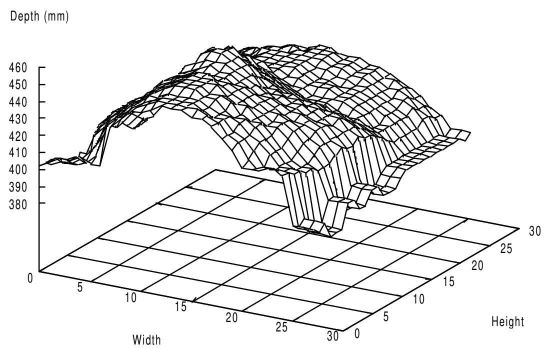


Figure 4-8 Result of multiple-resolution SIA for the face object

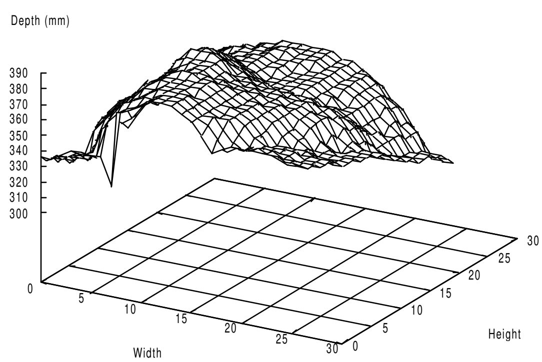


Figure 4-9 Result of DEFOST for the face object

4.4 Multiple-baseline SIA

The two problems associated with multiple-baseline stereo are accuracy and precision.[37] Longer baseline yields better precision due to wider triangulation. But longer baseline makes the range of search for the best match larger and leads to a higher possibility of false matches. In [36,37] M. Okutomi, et al, propose that using multiple-baseline stereo can determine a unique and clear minimum of sum of sum of squared differences (*SSSD*)-in-inverse-depth at the correct matching position and can also eliminate false matches and increase precision. But they have to search the whole possible depth range to compute the *SSSD*. In this research it shows that one can have a good initial estimate of the depth map from IDA/IFA. This estimate is used to limit the search range and reduce false matches. In the best case the unique minimum of *SSD* will be found in a range around the estimated depth map[54] obtained by IDA/IFA.

4.4.1 Experiments

In DEFOST two images are taken, say, right and left, and in the multiple-baseline configuration in this chapter a third image (the middle one) is taken and two pairs of stereo images are formed. The right and the middle ones form the first pair and the right and the left form the other. Applying SIA on the first pair of stereo images refines the result of the IDA/IFA output. Because of the shorter baseline of the first pair the precision is less than the second pair but the search range in matching will also be smaller. This yields a less precise result but faster matching. This result will be used to obtain an initial estimate of the disparity map for matching the second longer-baseline pair of stereo images. The second pair of images is matched by searching in a narrow disparity range around the initial estimates. Thus the computation is reduced and the accuracy of the final matching results is improved.

The results of 3D shape recovery by multiple-baseline stereo using gray-level image data is presented for a cone object here (see Figure 4-10). This result compares well with the result of the original DEFOST. All camera parameters for both methods are the same. The computation time of multiple-baseline stereo matching for several objects are averaged and tabulated below (see Table 4-2) for several baseline factors (the factor by which the full baseline is divided). For full baseline (50 mm) the baseline factor is 1 and for half baseline (25mm) the factor is 2. The disparity vs. depth calibration data for different baselines was obtained by scaling the calibration data for full baseline with the baseline factor. The last row of the table lists the computation speed-ups obtained.

Baseline factor	8	4	2	4/3	1
Time for matching the first pair	6	14	24	34	40
Time for matching the second pair	7	7	7	7	N/A
Total matching time	13	21	31	41	40
Speedup in time	3.08	1.90	1.29	0.98	1

Table 4-2 Speedup in time for multiple-baseline SIA

In the case of 1/8 baseline the speedup is high but the final result is less accurate than the rest because the precision error may overshadow the accuracy in the second pair matching. If one increases the search range for the second pair then the computational time will also increase. There is a tradeoff between speed and accuracy. On the other hand obtaining the third focused image takes time. In SVIS camera system it takes about 2 seconds to acquire each image. If the depth values of adjacent image blocks have large difference then there will be a large amount of distortion in a stereo image pair due to foreshortening effect. The longer the baseline is, the larger the effect is. For an object of large depth variations and a long baseline, the matching will not be good due to foreshortening effect. It is found that adding a third image in the middle and using the shorter-baseline stereo image pair to get an initial result will help the matching on the longer-baseline stereo image pair. This could reduce false matches and increase accuracy of matching.

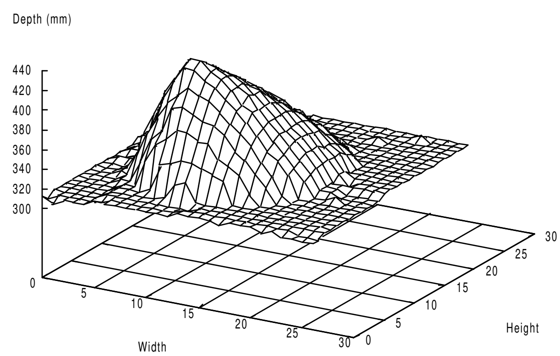


Figure 4-10 Result of multiple-baseline SIA for the cone object

4.4.2 Multiple-baseline SIA with color image data

Multiple-baseline SIA with color image matching is also implemented. Results for two objects, a cone object and a face object, are presented for a baseline factor of 2. The intermediate results of the first stereo image pairs (for half baseline) are also shown (Figure 4-11 – Figure 4-14). These results compare well to DEFOST and color DEFOST. It is also tabulate below the computation times for various baseline factors. The times are obtained by averaging the results of several experiments.

Baseline factor	8	4	2	4/3	1
Time for matching the first pair	27	41	78	106	127
Time for matching the second pair	25	25	25	25	N/A
Total matching time	52	66	103	131	127
Speedup in time	2.44	1.92	1.23	0.97	1

Table 4-3 Speedup in time in color multiple-baseline SIA

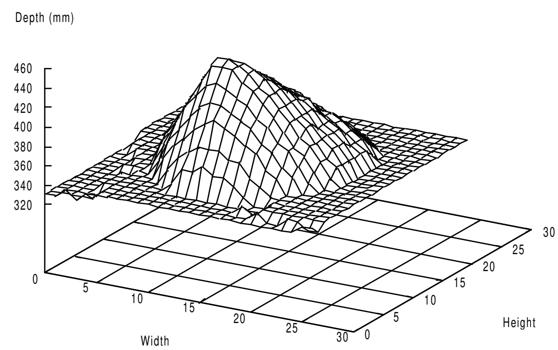


Figure 4-11 Result of the first stereo pair of color multiple-baseline SIA for the color cone object

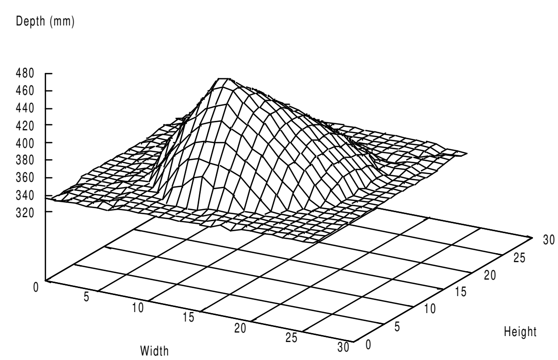


Figure 4-12 Result of color multiple-baseline stereo for the color cone object

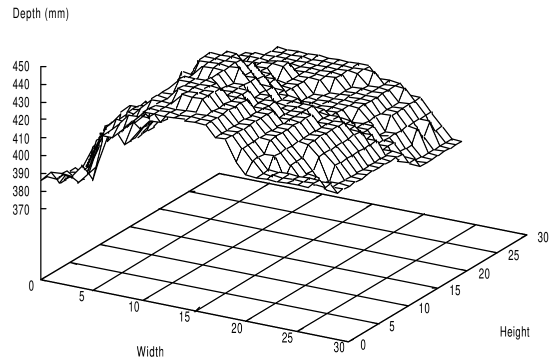


Figure 4-13 Result of the first stereo pair of color multiple-baseline SIA for the color face object

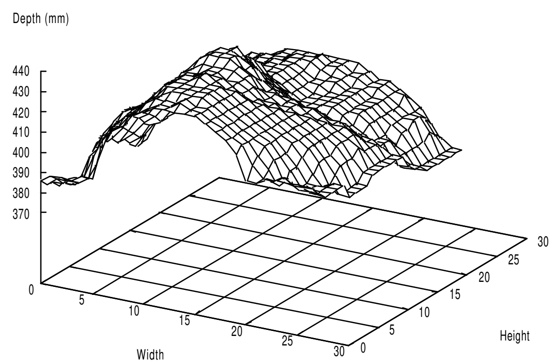


Figure 4-14 Result of color multiple-baseline SIA for the color face object

4.5 Match in blur

In this section I am going to present some experimental results and bring out some discussion and questions about this new concept — "match in blur (MIB.)" In all the experiments, the results obtained by FMIF (focus match in focus) with background removal are used for contrast purpose (Color FOST with background removal.)

First, in the experiment of BMIB (blur match in blur) the matching is done on two center-focused images as in normal camera shooting. A cone-like object is used. It is placed at a distance of 300 mm to 700 mm from the camera (Figure 4-15.) The other parameters of the camera system are similar to those in Color FOST.

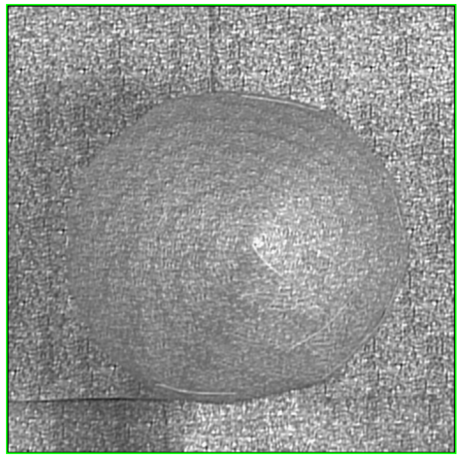


Figure 4-15 Color cone object

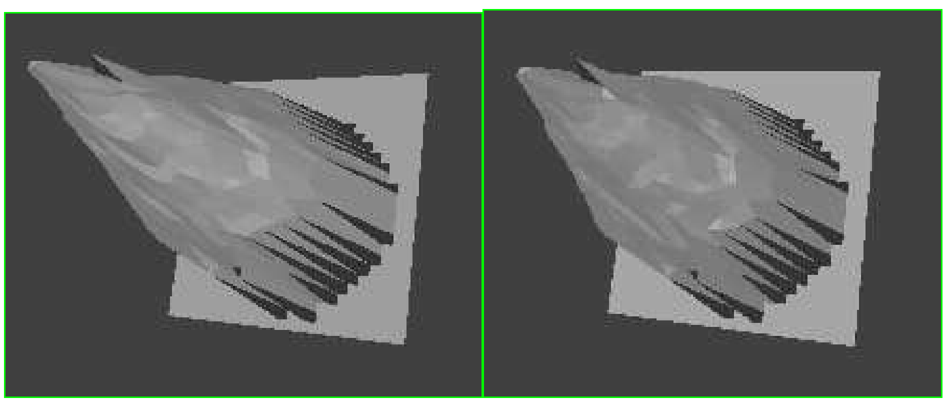


Figure 4-16 BMIB test (left: BMIB, right: FMIF)

Form Figure 4-16 we cannot see much difference between BMIB and FMIF. But they are not exactly the same if examined carefully.

The next set of results shows the MIB (match in blur) and FMIF of a cone object (Figure 4-17.) MIB matches a focused block in the right focused image to a matched focused block among blurred blocks on the left image taken at the same focusing lens position as the right focused block. Similarly, the results are both good and cannot tell which is better although in small details they are different.

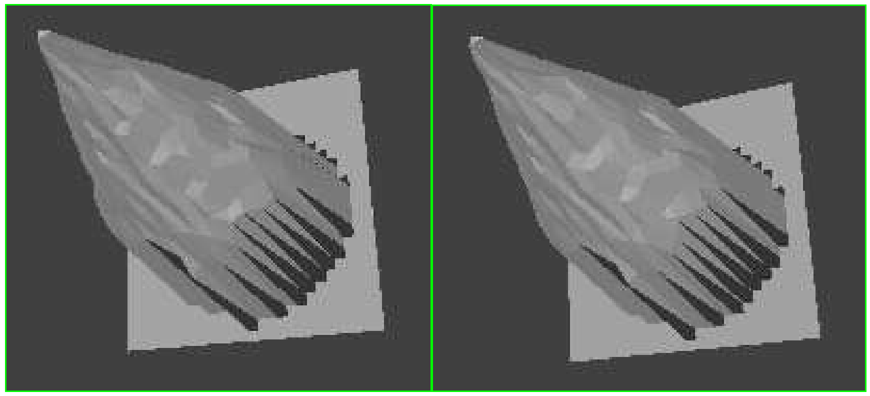


Figure 4-17 MIB test (left: MIB, right: FMIF)

The following set displays the effect of IDA/IFA assistance on both BMIB and MIB of a cone object. Without the estimated depth map from IDA/IFA, BMIB and MIB cannot work well (Figure 4-18, Figure 4-19.) Even the FMIF itself cannot work well without applying IDA/IFA (Figure 4-19.)

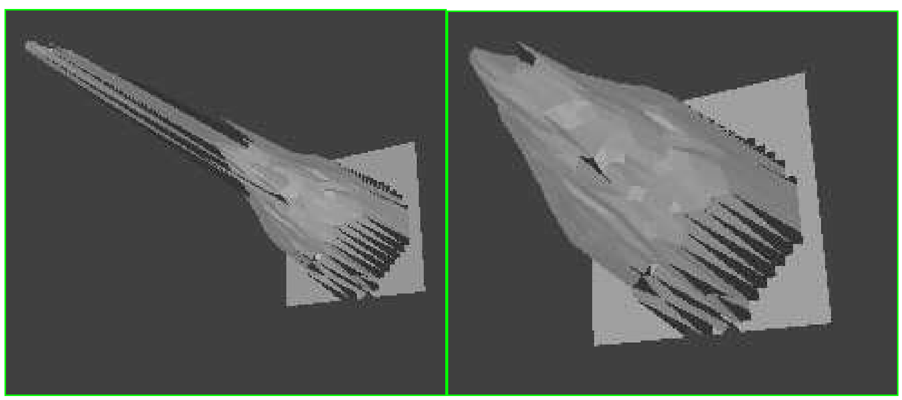


Figure 4-18 BMIB without integrating IDA/IFA (left: BMIB/no IDA/IFA, right: FMIF/IDA/IFA)

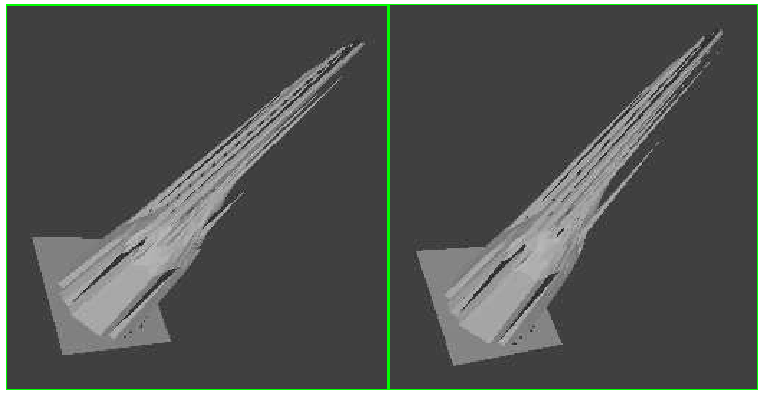


Figure 4-19 MIB without integrating IDA/IFA (left: BMIB/no IDA/IFA, right: FMIF/no IDA/IFA)

The last two sets of results that I am going to present are BMIB, MIB of a face object placed at a distance from 600 mm to 900 mm from the camera (Figure 4-20 - Figure 4-22.)

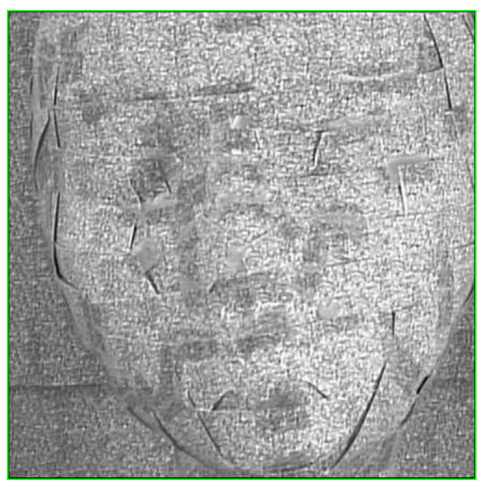


Figure 4-20 Color face object

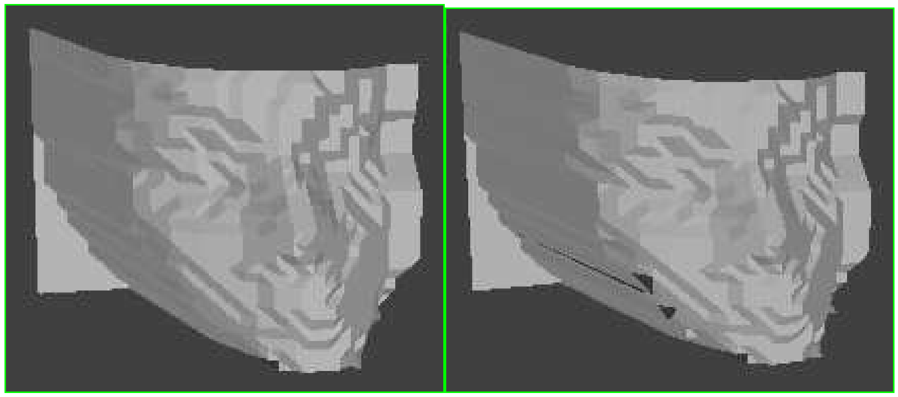


Figure 4-21 BMIB (left: BMIB, right: FMIF)

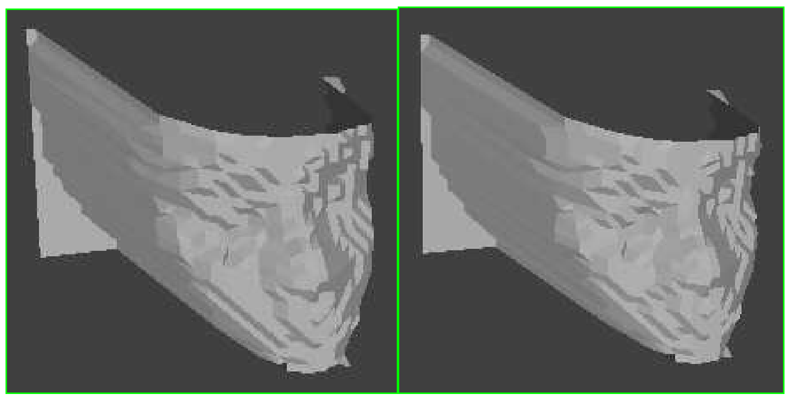


Figure 4-22 MIB (left: MIB, right: FMIF)

Basically there is not much difference between them. But in Figure 4-21 we can see that the BMIB result seems better than that of FMIF at the right cheek of the face object. The reason for why these peaks occur is still in question. And there are many possible explanations for it. But judging by only sparse examples is not the way it should be. More investigation should be done to model the focus, blur, and matching in order to analyze BMIB, MIB, and FMIF mathematically. But the idea of "match in blur" provides us with a new possible way of enhancing stereo matching.

The possible reasons for why MIB does not work better than FMIF as one expects in the above cases could be i) the pattern used in the experiments provides necessary contrast and in MIB there is not much to improve, ii) the depth of field of the camera in the vision system is deep due to small aperture and short focal length. The search range is covered by the depth of field so the blur effect is not evident. Experiments with new equipments that can adjust to examine the above two points should give us new information to develop new algorithms based on MIB.

4.6 Conclusion

In this chapter the integration of IDA, IFA, and various SIA techniques is presented. The experimental results on SVIS also indicate the feasibility and advantages of integrating multiple-baseline color stereo with IDA, IFA, and multiple-resolution matching technique. The advantages include improved accuracy of 3D shape and reduction in computation time. It is of no problem to extend the algorithms. For example, one can use more than two baselines or apply multiple-resolution matching technique with multiple-baseline stereo on color image data. It can also be extended to use other stereo matching techniques instead of the *SSD* based method.

The new concept of "match in blur (MIB)" is introduced. It is expected to improve the performance of stereo matching in SIA. Several preliminary experiments are performed to evaluate its effectiveness qualitatively. Although the results do not provide much information at present but the results compare well with others. Future research is needed to investigate it in depth.

Chapter 5 Application: Multiple View 3D Shape Reconstruction

5.1 Introduction

In this chapter I am going to present one practical application of recovering 360 degree 3D shape of an object using the algorithms and techniques developed in former chapters. The work in this chapter is done as collaboration with Mr. Huei-yung Lin and Mr. Xiangdong Qin in particular in Section 5.3. The recovered shape, obtained as in previous chapters, is defined as a "partial 3D model" of a 360 degree object. In other words, given a fixed object, one view of the 3D shape of that object could be obtained. By rotating the object (or changing the orientation of the object) different views of the 3D object could be recovered. Then each of the recovered "views" is defined as a partial 3D model. Those partial models could then be integrated into a whole 360 degree 3D model of the object. This chapter presents the details about recovering the 360 degree 3D shape of an object.

The 3D model of an object consists of two types of information (i) the 3D shape of the object, and (ii) the image texture on the outer visible surface of the object. Recovering the first type of information (3D shape) is a difficult problem in the computer vision area.[13] Some popular techniques are stereo, shading, focus analysis, structured light analysis, etc. As for the second type of information (image texture), it is recovered from the image(s) recorded by a camera. Stereo Image Analysis (SIA) is perhaps the most widely used technique for 3D shape recovery in computer vision (see chapter 1 and 2.) However, this technique has some inherent computational problems (e.g. correspondence and occlusion) related to matching stereo image pairs. In the previous chapters,[54,71] it is tried to mitigate these problems by integrating Image Focus Analysis (IFA)[22,43,44] and Image Defocus Analysis (IDA)[40,44] with SIA. In this chapter, IDA that can estimate the approximate range of an object is not literally used because the range is known (600 mm to 900 mm). Instead, one simplified version is used; in which the IDA is substituted with the known range. In the current work, IFA first provides an approximate 3D shape of the object, which simplifies the stereo matching problem in SIA. The approximate 3D shape is refined by SIA to obtain a more accurate 3D shape.

In this chapter the techniques used in previous chapters[54,71] on recovering the 3D shape and image texture of a single view of an object are extended in two main ways — computational algorithms and hardware architecture. The stereo image matching algorithms have been improved to obtain more accurate partial 3D models of objects. New computational algorithms are provided for representing and integrating partial 3D models obtained from different views into one complete 3D model of the object. Integrating partial models involves several steps. First, the measured coordinates of points on the surface of the object provided by partial models are transformed to an object centered cylindrical coordinate system. This requires calibrating the rotation axis of the object. A procedure for this purpose has been provided. The transformed coordinates of different partial models are merged by taking into account the rotation angles of the partial models. This results in a set of points corresponding to discrete sampling of the complete object's surface. These discrete sample points are used to interpolate the surface. Then the surface is resampled at regular intervals to model the surface with quadrilateral surface patches. The vertices of the quadrilaterals are projected back onto the focused images along different directions of view to obtain the image textures of the corresponding quadrilateral surface patches. The complete 3D model is then printed to a file in the 3D metafile format (3dmf) suitable for rendering by the Apple's Quick- Draw 3D Viewer software. The earlier computing and image digitization hardware of SVIS[54,71] used in previous chapters have been completely replaced with new and more powerful ones (Figure 5-1). A computer controlled motor system has been added to rotate the object by known amounts so that different views of the object are automatically obtained. At present, SVIS works well for simple objects defined as objects whose cross-sections perpendicular to their rotation axis can be defined in polar coordinates (with its origin at the rotation axis) by a function of the form $r(\theta)$. SVIS could be extended to work with more complicated objects in the future research. The results of 3D model acquisition for several objects are presented in this chapter.

5.2 Data acquisition

5.2.1 Experimental Setup

The integrated digital vision system is named SVIS (Stonybrook VIsion System.) As mentioned above it is an upgraded version of the previous SVIS system (as in chapter 3 and 4). The upgrading includes i) a new digitizer board — Matrox Meteor II® standard board and its entire associated updated MIL Lite® library, ii) a Pentium II® 450MHz PC with 256MB RAM and a 16MB video card, and iii) an object rotation stage with a MD2 stepper motor. The new hardware enhances a lot the speed of the data acquisition and the computation of the working algorithms. Objects are inserted vertically upon the object rotation stage and can be rotated with 360 degree of freedom. The new system's setup is depicted as Figure 5-1.

The objects that are used in this experiment are i) a Face object, ii) a Four-object box object, iii) a Four-object cylinder object, and iv) a plain Cylinder object. See Figure 5-2 - Figure 5-5 for their eight views of color focused images. The objects are vertically inserted on the rotation stage. Through the software user interface integrated within the vision system the number of views could be set as required. The user can automatically rotate the object to obtain four, six, or eight views according to the requirements. Actually any number of views could be set. Depending on how much information that the user prefers and the postprocessing program such as the stitching program (the partial model integrating program) needs.

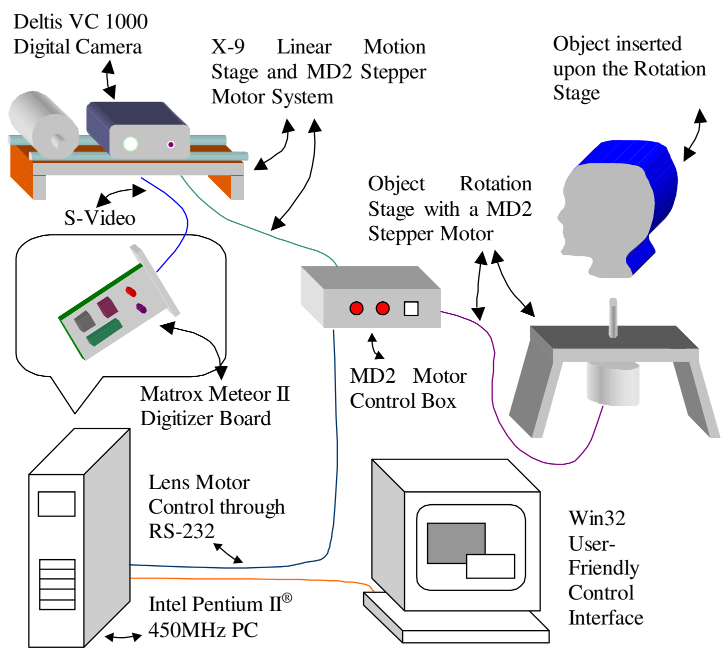


Figure 5-1 The upgraded SVIS system

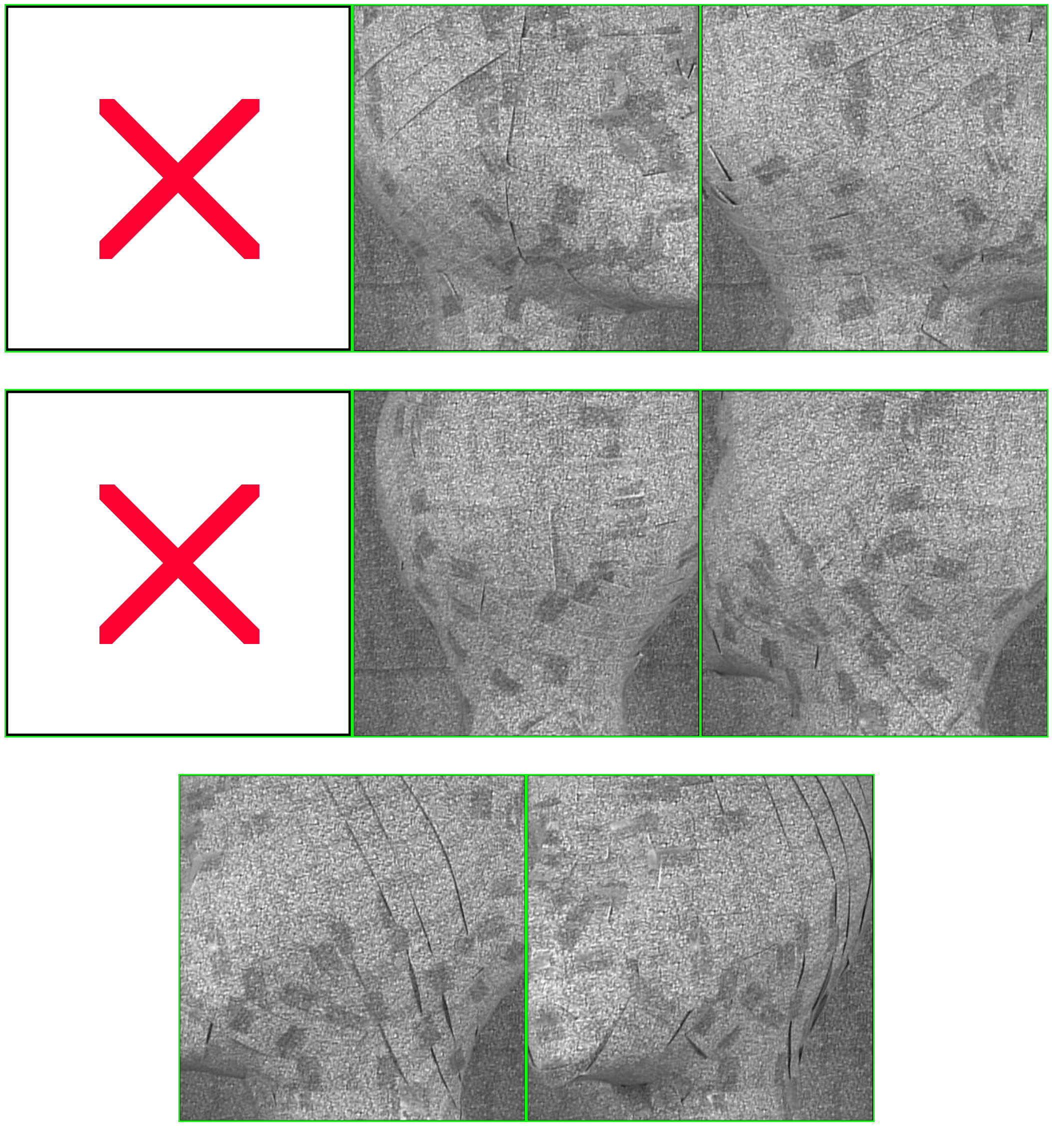
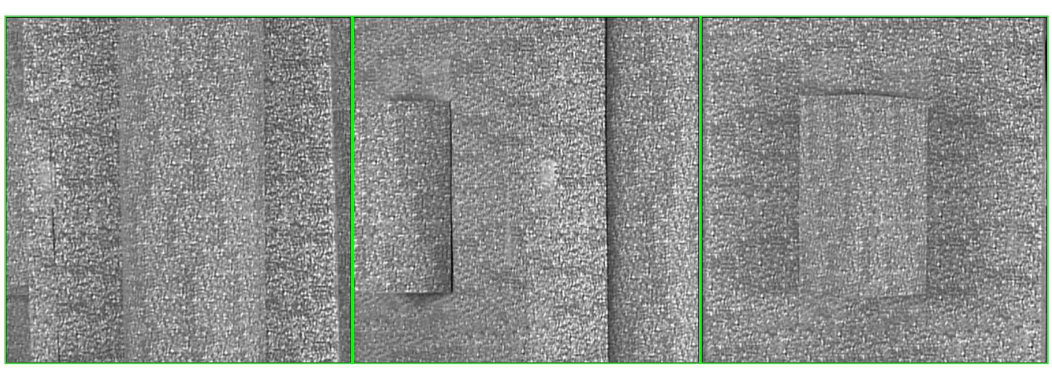


Figure 5-2 Eight views of the face object
(View0 to View7: left to right, top to bottom)



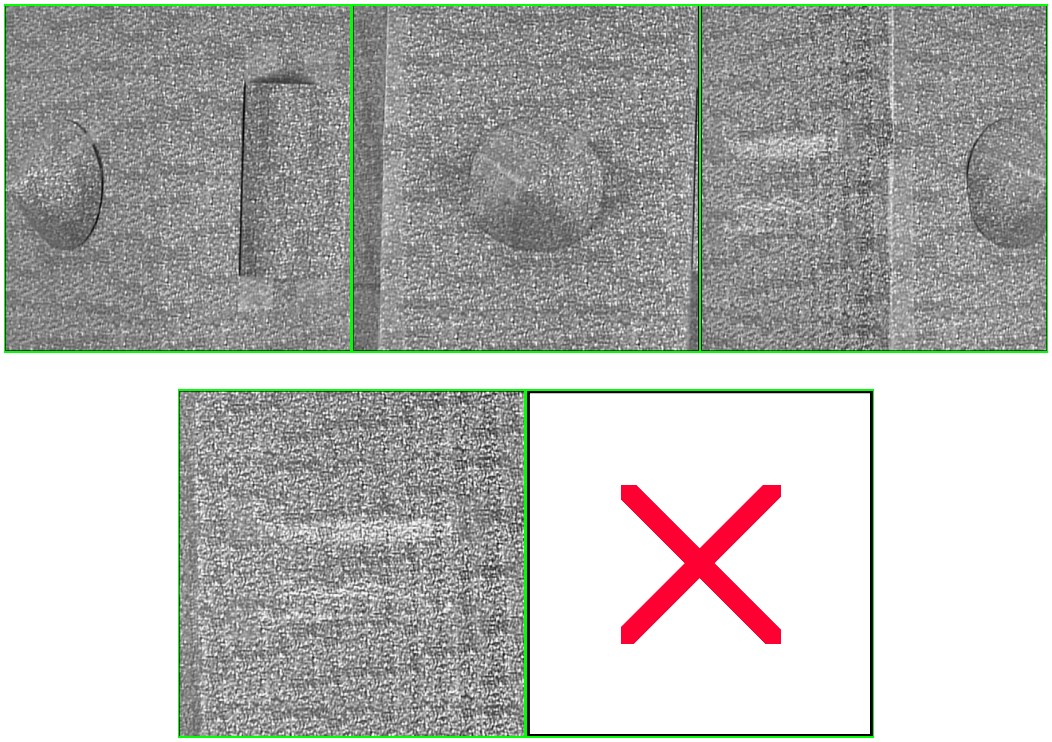
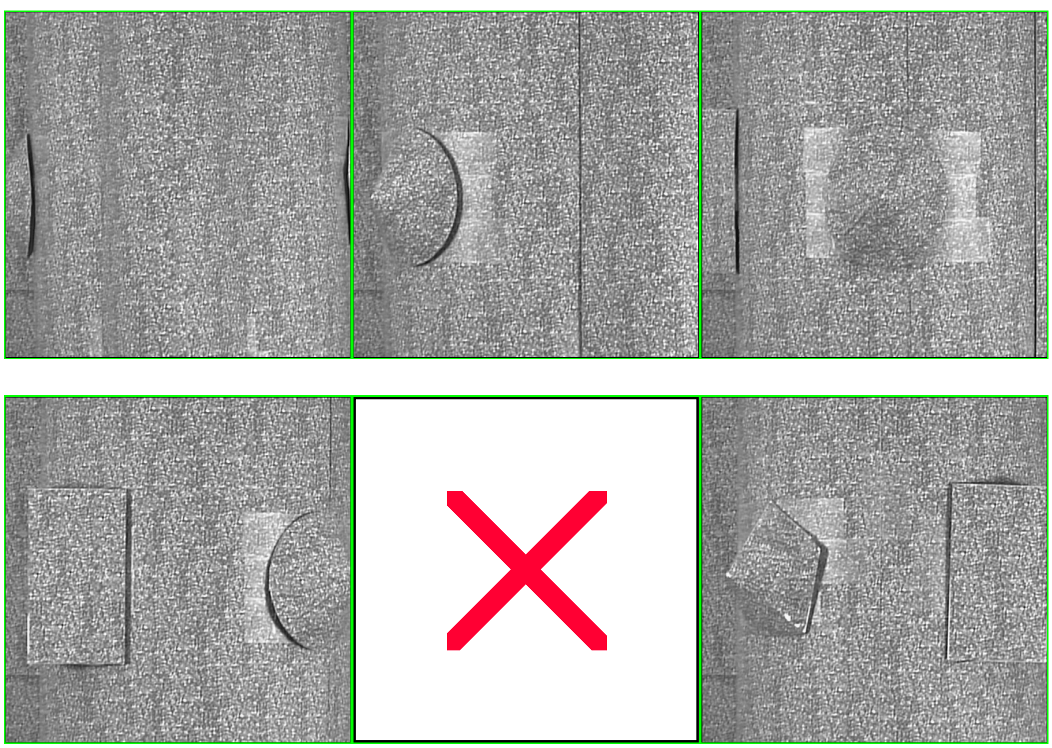


Figure 5-3 Eight views of the four-object box object
(View0 to View7: left to right, top to bottom)



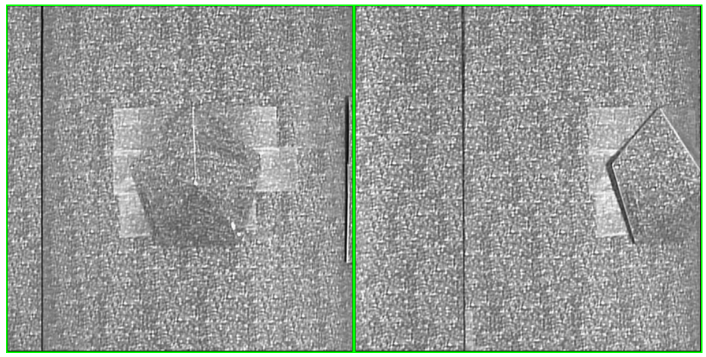


Figure 5-4 Eight views of the four-object cylinder object
(View0 to View7: left to right, top to bottom)

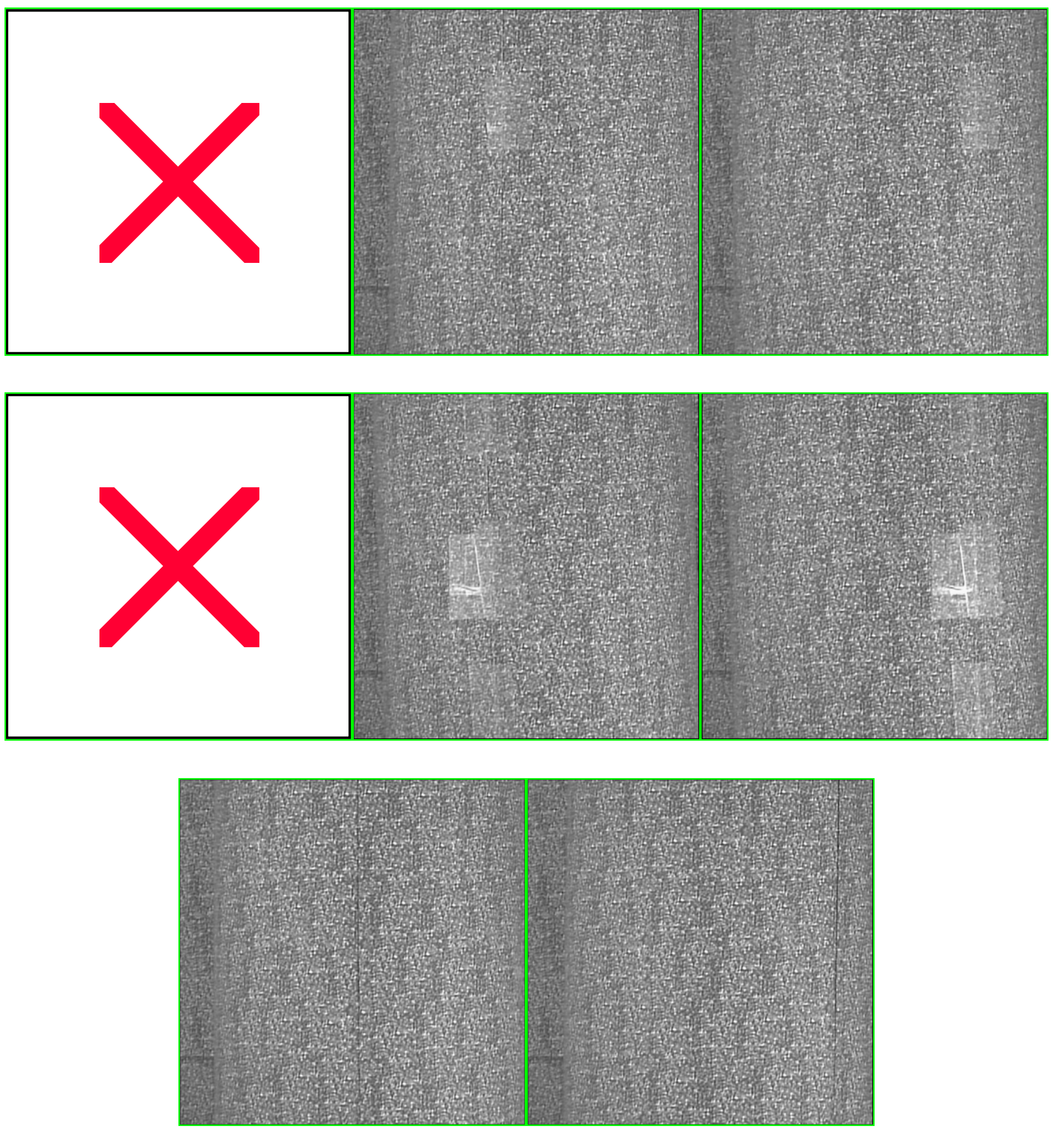


Figure 5-5 Eight views of the plain cylinder object
(View0 to View7: left to right, top to bottom)

The face object is a human head like object with some depth details in the front face and is smooth around the rest of the head. The four-object box object has four small objects; each of which lies on each of the four faces of the box object. The first object is a vertical half cylinder object. The second object is a vertical prism object. The third object is a cone object. The fourth one is a null object, i.e., no object is placed on the last face of the box object. The four-object cylinder object has also four small objects around a circumference of the cylinder object. They include a vertical half cylinder object, a cone object, a vertical prism object, and an object of cone-like polyhedron with a pentagonal base. The plain cylinder object is purely an upright cylinder without any attached small object. It is used as a contrast object.

All the objects are placed in a range of 600 mm to 900 mm before the camera (see Figure 5-6.) Objects are assumed to be able to fit inside a bounding cubic volume of size 300mm $\times$ 300mm $\times$ 300mm. For convenience high contrast color random dot pattern is pasted upon all the objects. Instead of pasting the pattern, the same pattern could be projected onto the objects to achieve the same effect of introducing enough contrast information. For a live object no pattern could be pasted onto it. Introducing the contrast by projection of patterns is necessary and has some advantages. One advantage is that the user could try different patterns to achieve the best result. And also after removing the projection of the pattern one could take pictures of the object to obtain the true and focused texture images with the help from the result of, say, IDA/IFA. So the texture mapping part of the recovered 360-degree shape will not be of the pattern but of the true texture of the object.

5.2.2 Algorithm used

The algorithm used in recovering the 360-degree shape of an object is called *Color FOST with background removal and occlusion detection*. *FOST* stands for *FO*cus and *ST*ereo. This is a modified IDA/IFA integration. Since the working range of the shape recovery is predetermined to be, say, 600 mm to 900 mm in front of the camera stage, IDA is simply replaced by the correspondent range of step numbers of the camera lens motor. In the case of a range from 600 mm to 900 mm, the correspondent range of step number is from step 113 to step 105. IFA is set to take pictures at a 2–step interval. So the total frames that are taken at each run of IFA method are *(113 - 105)/2 + 1 = 5* frames. In the algorithm of IFA, computation is done on gray level images obtained from color images through the formula:[71]

Gray level value = (Red + Green + Blue)/3 . Equation 5-1

For each view of the object, a 432×432 subimage is extracted from a 640×480 image recorded by the camera. The center of the 640×480 recorded is taken to be the point where the optical axis intersects the image plane. It was also the origin of the image coordinate system for perspective projection. IFA is applied in 16×16 image blocks to obtain a 27×27 depth-map and a 432×432 focused image. An estimated depth map (in step number) is thus obtained from the modified IDA/IFA method. This rough depth map is then fed into the SIA part of the FOST algorithm.

In FOST's Stereo matching part, an estimate of IFA error of 4 steps is used to obtain the possible range of disparity for searching best matches. Sometimes this parameter could be adjusted to achieve the degree of accuracy and the cost of computation that the user prefers. The initial depth-map estimate obtained by IFA is refined using SIA with color image data. The stereo baseline is 50 mm. The single digital camera is moved by a motor perpendicular to its optical axis to create the effect of a stereo camera. Focused images recovered by IFA corresponding to left and right stereo images are used in matching. Matching is done for 16×16 image blocks using the sum-of-squared-difference measure for color images[71].

In traditional Color FOST algorithm all the blocks in the image (including the background and the foreground of the scene) are matched. In cases that only the foreground objects are what we are interested in, it is a waste trying to match the irrelevant background objects (usually a vertical flat plane placed at the farthest position of the working range, say, 900 mm from the camera.) And most of the time there will be matching errors due to the occlusion of the background from the foreground. And these errors are usually severe and should be removed. In this research a new theory and method of removing the background from matching and thus reduces the computational time and also the probability of false matches due to occlusion is introduced (see also chapter 2.)

Not only the background should be removed from matching but also the occlusion caused by the object itself. If one part of the object is hidden from the viewer by another part of the object after the camera is moved to the other stereo position, say, the left position, then the matching will fail and errors occur due to the occlusion. This occlusion could be detected in the new algorithm that is used in this application. The occlusion detection could tell the user where there might be occlusion and let the user do some correction to it. The user could do interpolation, extrapolation, and other techniques to solve the occlusion problem once it is identified. In the algorithm used in this chapter a simple way of dealing with the occlusion is used. Wherever the occlusion occurs, its depth is substituted with the nearest reliable one, which is usually the one left to it (searching and matching are done from left to right along the epipolar line so the left one is usually a reliable one.) In this application successful detection is achieved and false matches due to occlusion are removed. Therefore a nice and clean result of each view of the 360-degree shape could be obtained. Comparing the result from Color FOST and the information of the occlusion detection one can even distinguish errors due to occlusion from those due to lack of contrast or some other reasons. This helps diagnose the matching errors. The details of these two techniques will be discussed in the following sections. The Color FOST algorithm is summarized as follows:

1. Apply IDA/IFA to get estimated depth maps (in steps) of the scene from color images taken at the two camera positions.
2. Separate foreground objects from the background by means of the result of the estimated depth maps from IDA/IFA at both camera positions.
3. Apply SIA: Match only foreground in both right and left images to save computational time and reduce the false matches due to foreground objects' bordering occlusion upon the background.
4. Detect occlusion: For each match found in the left image in SIA, back-match it to the right image to find its best match there. Compare the original matching block with the back-matched one to determine whether occlusion occurs (discussed in later sections.) If it does, substitute the depth found in SIA with its nearest reliable one (usually the left adjacent one) or solve by other techniques such as interpolation and extrapolation.

5. Repeat step 3 and step 4 till the whole depth map of one view of object is recovered.
6. Rotate the object and repeat step 1 to step 5 to obtain all the depth maps of the multiple views of the 360-degree object.

5.2.3 Background removal

As discussed in chapter 2, occlusion is usually due to abrupt depth changes among objects (usually happens at borders of objects.) Abrupt depth changes violate the *global and local smoothness constraints* adopted by most of the stereo matching algorithms. If abrupt depth change happens, some part of the objects cannot be seen in both stereo images (or will be hidden from each other.) This usually causes severe problem in stereo matching. In this application the object is rotated along a vertical axis. The background (usually an upright flat planar object at the farthest position among the working range, say, about 900 mm from the camera,) will have a depth discrepancy between itself and the foreground object, say, a dummy head object. This abrupt depth change will cause occlusion at the border of the head object and the background. It is necessary to separate the background from the foreground to avoid the false matches due to the occlusion at the border of the object.

One way of removing the background from matching is derived here (see Figure 5-6.) First, the objects that are used in this experiment are *simple objects*. Simple objects mean that every ray originating from the rotation axis and lying on a cross-section perpendicular to the axis will pierce the object surface at most once. This condition should be observed in most part of the object due to the limit of the current stitching algorithm but will be removed in later versions of the stitching algorithm. Next, the distance from the rotation axis to the background plane is named "*Depth from background to axis*" and is denoted as D_{BgAx}. Another parameter is "*Estimated Background Error*" and is denoted as Err_{Bg}. Err_{Bg} means the estimated error of the background plane in IDA/IFA processing. The depth of the background plane is denoted as D_{Bg}. All parameters are in steps. We can see from Figure 5-6 that their relationship is:

$$Err_{Bg} \le D_{BgAx} \quad . \qquad \text{Equation 5-2}$$

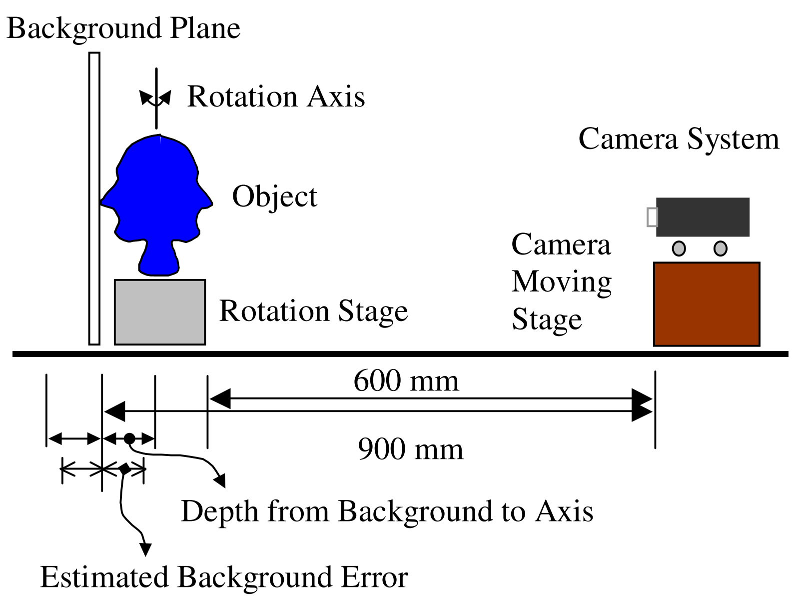


Figure 5-6 Background Removal

The background is determined by:

$$BG(m,n) = \begin{cases} 0 \ , & if \ \ D(m,n) \le D_{Bg} + Err_{Bg} \ \ and \ \ D(m,n) \ge D_{Bg} - Err_{Bg} \\ 1 \ , & otherwise \end{cases} \quad ,$$

Equation 5-3

where *BG(m,n)* is the background mask that indicated whether a block is a foreground or a background. *BG(m,n) = 0* indicates a background block and *BG(m,n) = 1* indicates a foreground. *D(m,n)* is the estimated depth map obtained from IDA/IFA algorithm. *(m,n)* is the index of blocks onto which IDA/IFA is applied. Here the block size is16×16 and the dimension of the depth map is 27×27 for images of size 432×432. The D_{Bg} is about step 105 in this experiment. The Err_{Bg} is set to be 2. It is adjustable and can limit the foreground to a degree that the user prefers.

Sometimes due to the uncontrollable conditions such as the sudden change in ambient light or camera electronics' problem there will be some errors that makes the background to be in the range of foreground and vice versa. These errors are usually few and sparse. Although they becomes errors in the background mask but in the stereo matching they will usually self-correct because one background error (viewed as foreground) in the right image is usually surrounded by other correct background blocks. While doing matching this erroneous foreground will match into the left image and usually cannot find correspondent foreground block there. So it just has no effect on the final result.

The background mask is produced at both camera positions. They are good guides for directing stereo matching not to match in background parts but only in foreground part in both images. When a block in the right image is at background, it is not going to be matched in the left image but is just skipped. When a block in the right image is at foreground, its match is search for in the left image. If any of those blocks being searched in the left image is at background, it is also skipped without matching. Only those blocks at foreground in the left image are matched. In this way the background is removed form matching and thus reduces the time for matching and also increases the accuracy while matching at borders of the object.

There is still one thing to say about finding match in the left image. If the center of the going –to-match block in the left image is at background then the center is shifted left and right by half of the block size. If the shifted center is in either case at foreground then the block is going to match no matter its center is at background. This is to allow blocks that are at the border and are half background and half foreground to be matched. This extension is due to the unsatisfied *local smoothness constraint* that states in one matching block the depth is constant (unique.) The result of background removal is shown in Figure 5-7 - Figure 5-9.

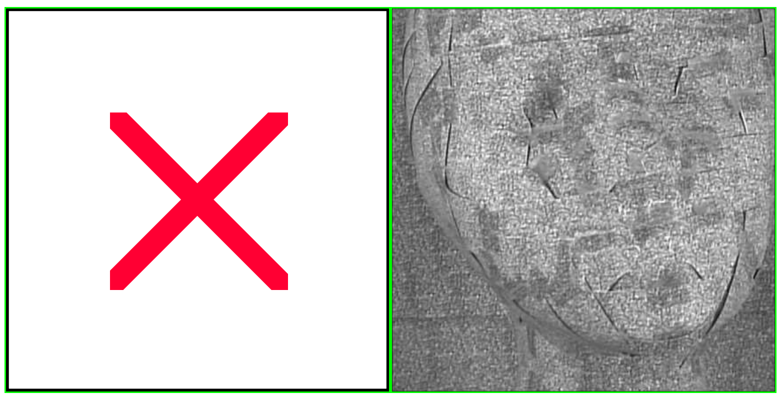


Figure 5-7 Right and Left Focused Images

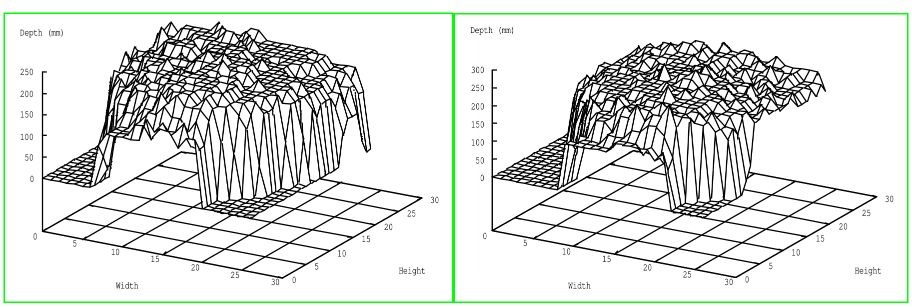


Figure 5-8 Right and Left IDA/IFA Estimated Depth Maps

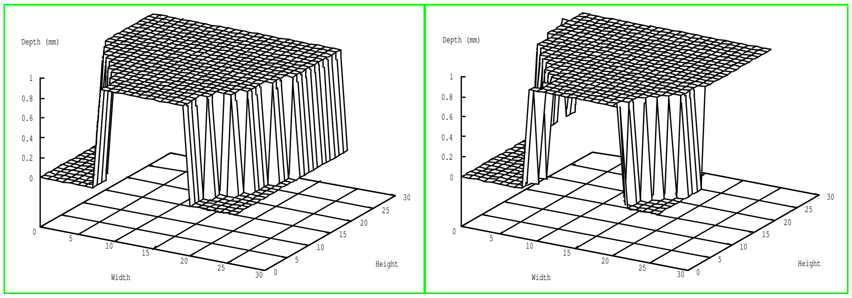


Figure 5-9 Right and Left Background Masks

5.2.4 Occlusion detection

Since the background is removed from the stereo matching. All the matching is now done on the true object. But the occlusion is still existent because object could be occluded by itself. And also around border the depth change is still high and can still cause occlusion-like problem. One new method is introduced here to detect these problematic places. The user can then choose how to deal with these occlusion errors once they are identified.

In the stage of SIA of the Color FOST algorithm, each matching block in the right stereo image searches for its best match in the left stereo image. The disparity-range-limited search will give it a match with minimum SSD value no matter it is correct or not. If occlusion occurs no match is possible but the matching will still give a false match. To avoid the errors caused by occlusion each "best match" of the matching block in the right image will be matched back to the right image with an estimated disparity range computed from the estimated depth map of the left image. The estimated depth map is obtained from IDA/IFA and background removal is also applied while back-matching from the left image to the right image. The "best match" of the back-matching is then compared to the original matching block. If their centers of block are shifted from each other over a predetermined *occlusion threshold*, then the original "best match" is considered as occlusion. If the centers of blocks are shifted under the occlusion threshold, it is considered as a minor "false match." (See Figure 5-10.)

The shift between the original matching block and the back-matched block is denoted as S_{bk}. How to choose the occlusion threshold O_{th} is a question in occlusion detection. One reasonable value is half of the matching block size. Suppose the original matching block is denoted as $B_{org}(i_0,j_0)$. The best match of the original matching block is $B_{bm}(i_1,j_1)$. The back-matched block is $B_{bk}(i_2,j_2)$. And the original matching process is denoted as $M(\bullet)$ and the back-matching process, $M_{bk}(\bullet)$. The occlusion detection mask is $OC(m,n)$. The occlusion value is $OV(m,n)$. Then the occlusion detection could be summarized as:

$$B_{bm}(i_1,j_1) = M(B_{org}(i_0,j_0)) \ ,$$

$$B_{bk}(i_2,j_2) = M_{bk}(B_{bm}(i_1,j_1)) \ ,$$

$$If \ \ OV(i_0,j_0) = \ S_{bk} = |\, j_2 - j_0 \,| > O_{th} \ , \ then \ OC(i_0,j_0) = 1 \ ,$$

$$Else \ OC(i_0,j_0) = 0 \ .$$

Note that the vertical shift is computed but not used in the current algorithm. It could be combined with horizontal shift to yield more complicated value.

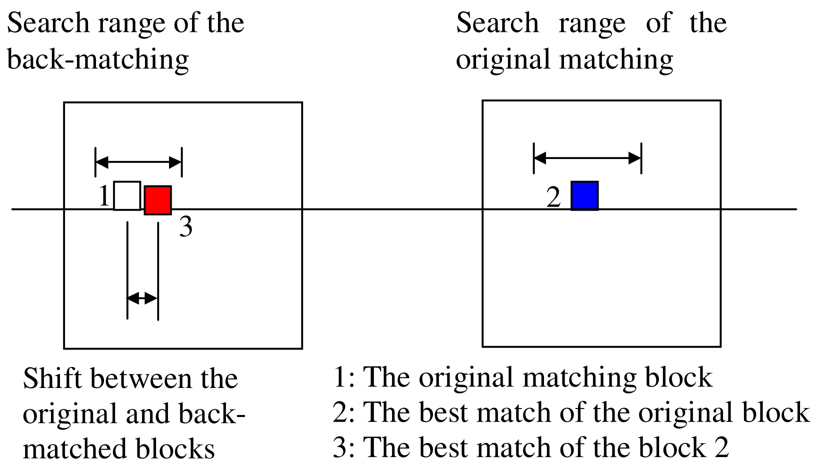


Figure 5-10 Occlusion Detection

An example of occlusion detection is shown in Figure 5-11 - Figure 5-13 and also Table 5-1. We can see that the occlusion occurs in two kinds of areas. One is the *unmatchable area* due to the limit of the size of the viewing window. This type of occlusion is not really occlusion. It is one example of the *correspondence problem*. This is shown in Figure 5-13 near the right border. Their shift values range from –8 to –22 in this case. The other area where occlusion happens is the area where abrupt depth change occurs. Form Figure 5-13 we can see that occlusion of this type occurs around the border of the face object, especially around the chin where depth change varies dramatically.

By the occlusion detection we can see that one visible peak at the chin (occlusion value is -11) is removed (replaced by the nearest reliable depth, i.e., the one left to the peak,) see Figure 5-11 and Figure 5-12. Other changes include the depth adjustments in the unmatchable area to the right of the face, and another depth replacement around the chin of occlusion value of 14 (to the right of the previous peak.) The occlusion threshold in this case is 8 (half the size of a matching block.) All shifts below this threshold are considered as minor false matches.

Figure 5-11 Before Detection   Figure 5-12 After Detection

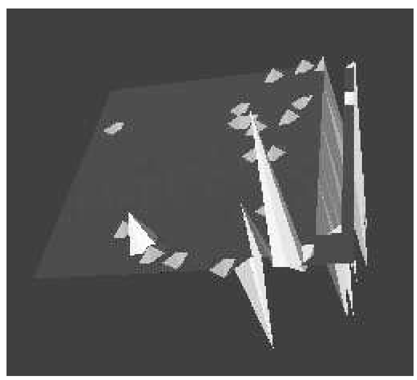


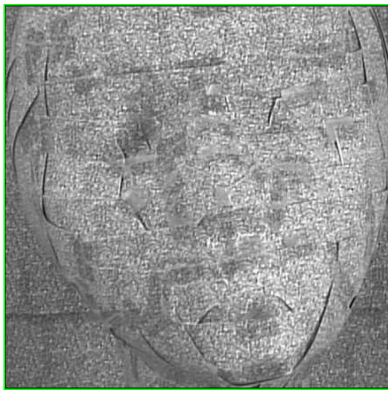
Figure 5-13 Result of Occlusion Detection

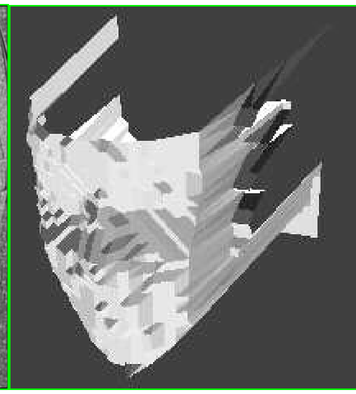
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	-8	-25
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	-8	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-10	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-18	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-17	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-13	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	-1	0	0	-13	-25
7	0	0	-1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-14	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	-1	0	0	0	1	0	0	0	-13	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	-20	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-22	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-17	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-18	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	1	0	0	0	0	-12	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-15	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-14	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-12	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-19	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-13	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-12	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	-13	0
21	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-13	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-11	0
23	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	0	0	0	0	0	14	-2	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	-11	7	0	0	0	0	0	0

Table 5-1 Result of Occlusion Detection in text

Another advantage of occlusion detection is that it can help verify the cause of the false matches. In an experiment one false match is generated after stereo matching (see Figure 5-14.) At first it is treated as an occlusion error due to its location in an area of high probability of occlusion (at the border of the chin, see Figure 5-15.) But after checking the result of occlusion detection it is excluded from the occlusion list (see Figure 5-16.; none of the peaks appearing in this figure of occlusion detection result corresponds to the peak at chin.) The occlusion value of that peak is 0 (indicating no occlusion at all.) With other efforts the pattern is found out to be the cause of error.

After providing with enough contrast around the chin, this error is removed (see Figure 5-17 and Figure 5-18.)





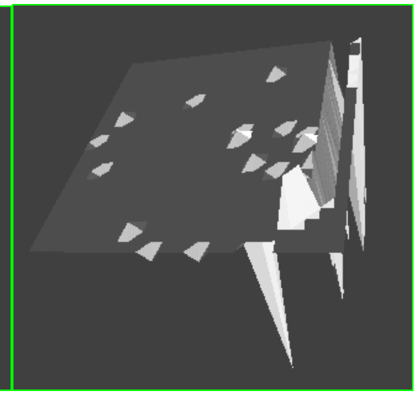
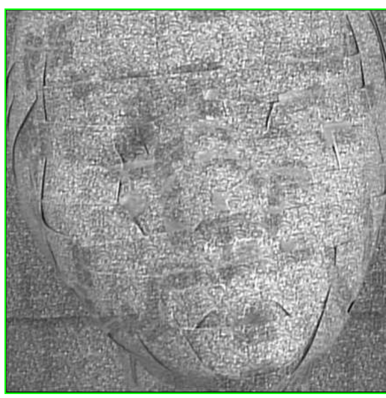


Figure 5-14 Figure 5-15 Figure 5-16

Figure 5-14: The focused image of the face object with a peak on chin.
Figure 5-15: The 3D view of the face object showing the peak on chin.
Figure 5-16: The result of occlusion detection showing no occlusion at the peak.



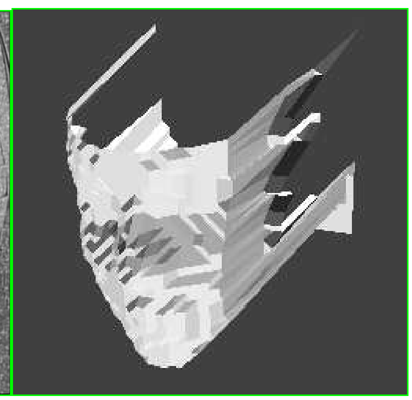


Figure 5-17 Figure 5-18

Figure 5-17: Contrast amended focused image
Figure 5-18: 3D view of face object with removed peak

5.2.5 Data collected

The visible surface from the camera's viewpoint is modeled in two parts (i) the 3D shape of the surface specified by the 27 × 27 depth-map array, and (ii) the image texture of the surface specified by the focused image recovered by IFA. This constitutes a partial model of the object as only a part of the object is visible from a given direction of view. The 3D shape of the surface is modeled as follows. The camera coordinates of the points in the depth-map are obtained by an inverse perspective projection for the camera and printed as (X_i, Y_i, Z_i) triples for $i = 1, 2, \ldots, n$, where n is the total number of points ($n = 27 \times 27$). These triples, constitute a list of vertices in the 3D space of camera coordinate system, where the vertices are uniquely identified by their indices i. The rectangular grid specified by the 27×27 array is used to create a list of quadrilaterals in the 3D space where each quadrilateral corresponds to one rectangle in the grid. The clockwise ordering of corner points of the rectangle in the grid are used to specify the quadrilateral as a list of four ordered vertex indices. For each quadrilateral, the image texture is obtained from the focused image in the corresponding rectangle in the rectangular grid. Thus, for a given direction of view, the partial 3D model of the object is obtained. This model can be displayed on a computer monitor using rendering software such as Apple's QuickDraw or GeomView of University of Minnesota. The above method for obtaining partial 3D model of an object is repeated for 4 to 8 different views of the object. Different views of the object are obtained by rotating the object using a computer controlled motor by known angles (45 to 90 degrees).

The results of the Color FOST algorithm and the partial 3D modeling are displayed below. Each of the objects has eight views taken in a 360 degree rotation. The angle in degree between two consecutive views is *360/number of views*. For example, the angle between two adjacent views for a eight-view rotation is $360/8 = 45$ *(degree)*. The rotation is counter-clockwise. The first object displayed here is the face object. See Figure 5-19.

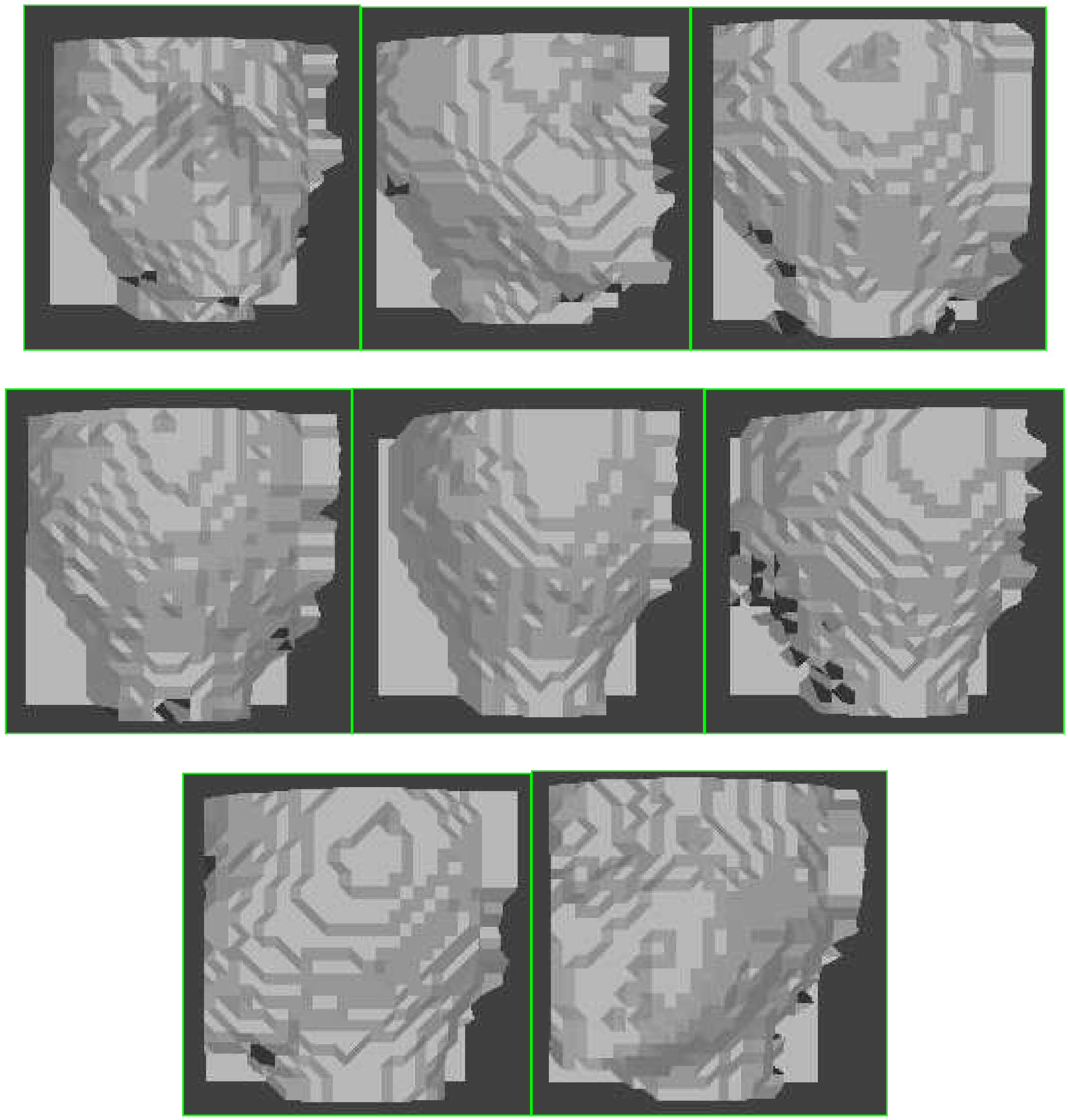
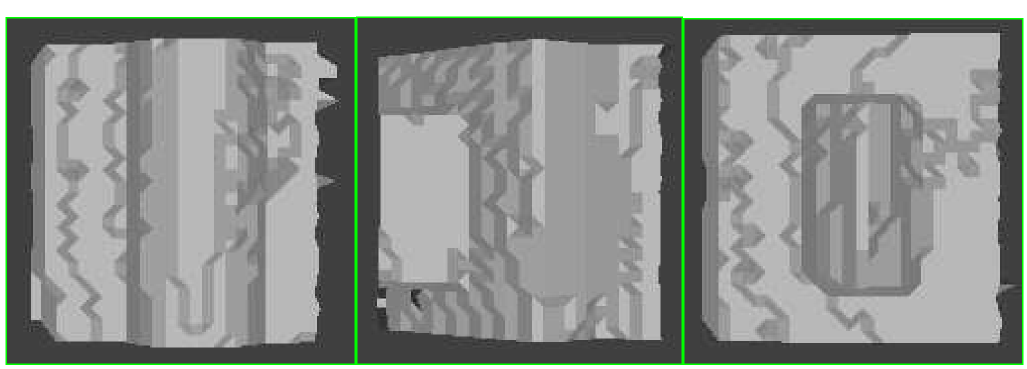


Figure 5-19 Face Object. From left to right, top to bottom: Front view, front right view, right view, rear right view, rear view, rear left view, left view, front left view.

The second object is a four-object box object. See Figure 5-20.



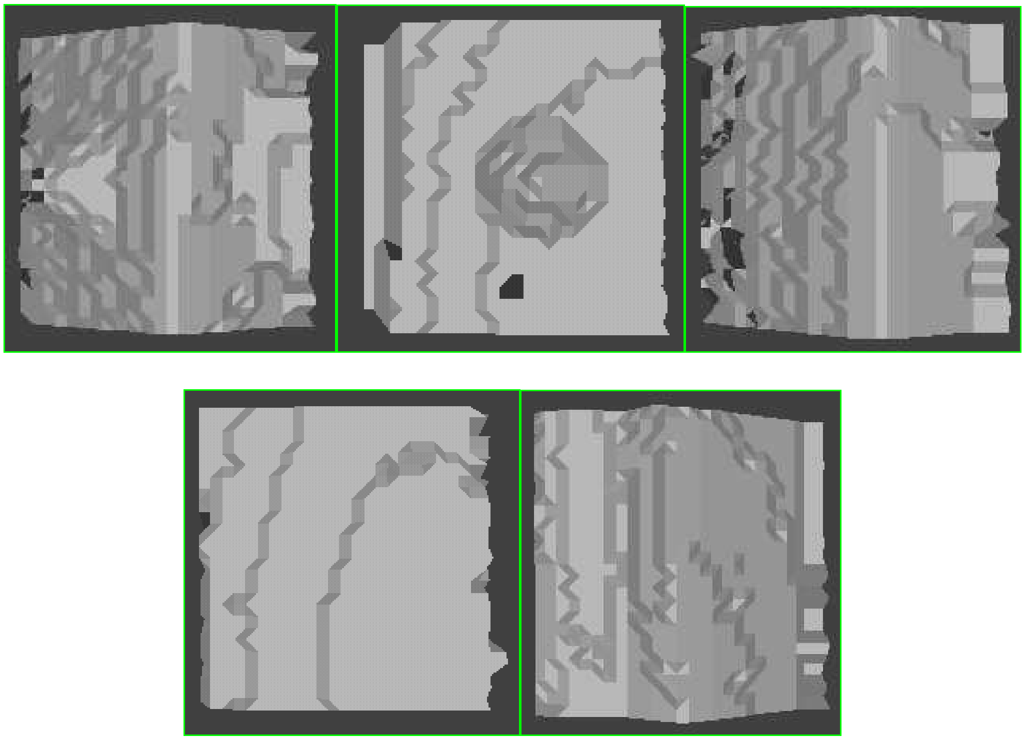
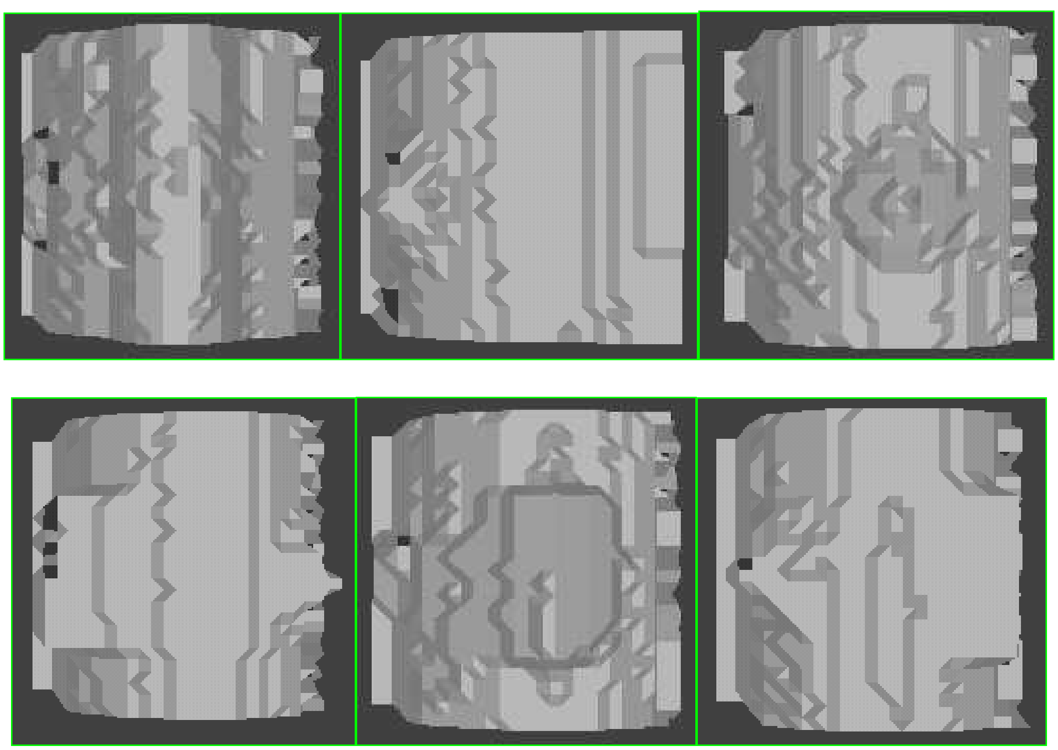


Figure 5-20 Four-object Box Object. From left to right, top to bottom: Front view, front right view, right view, rear right view, rear view, rear left view, left view, front left view.

The third object is a four-object cylinder object. See Figure 5-21.



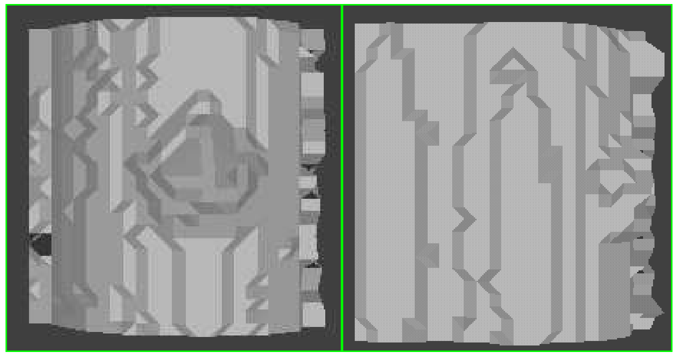
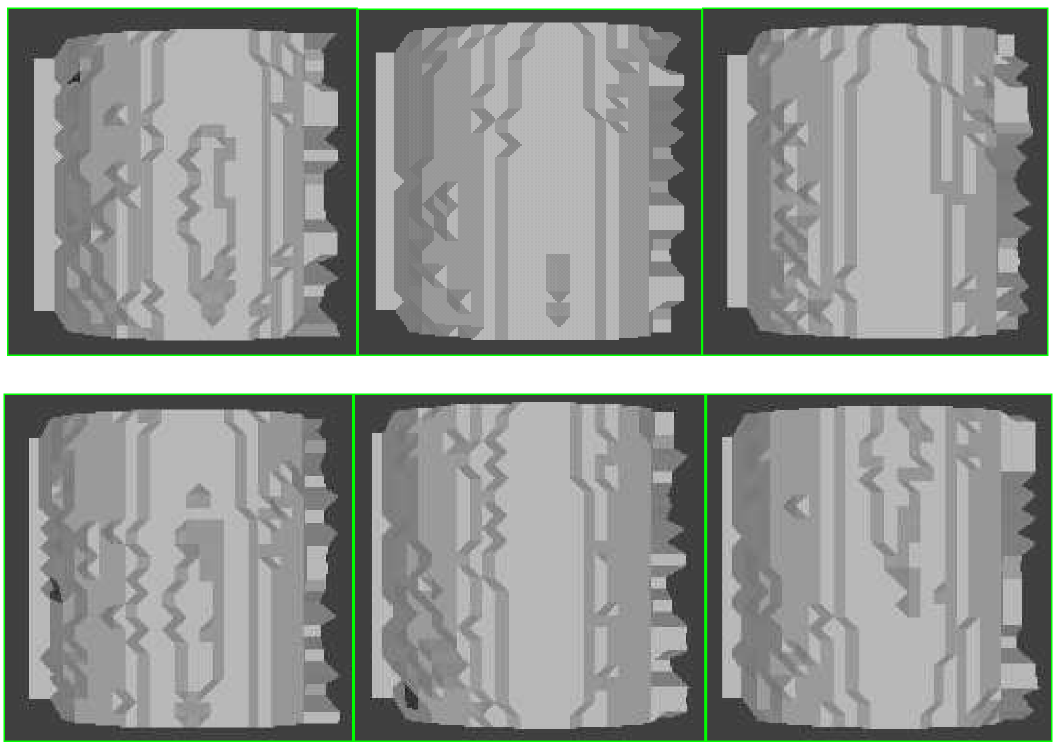


Figure 5-21 Four-object Cylinder Object. From left to right, top to bottom: Front view, front right view, right view, rear right view, rear view, rear left view, left view, front left view.

The forth object is a plain cylinder object. See Figure 5-22.



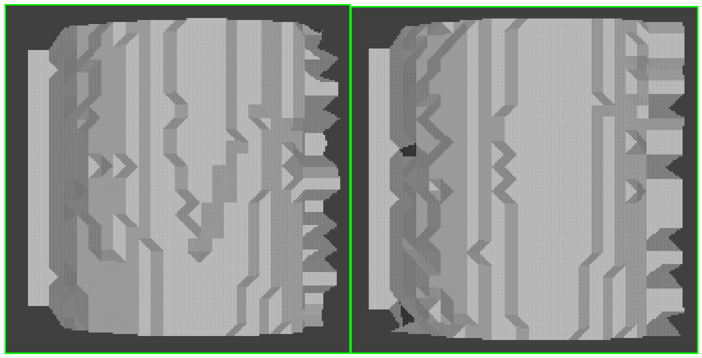


Figure 5-22 Four-object Cylinder Object. From left to right, top to bottom: Front view, front right view, right view, rear right view, rear view, rear left view, left view, front left view.

As for the focused images of the above partial 3D models please refer to Figure 5-2,Figure 5-3,Figure 5-4, and Figure 5-5.

5.2.6 Determination of Axis of Rotation

Before the above collected data could be sent to the postprocessing programs such as a stitching program. We need to provide those programs with the information about the rotation axis. The information of the rotation axis could be specified as a pair of parameters, i.e., (X_0, Z_0). The X_0 is the horizontal coordinate of the rotation axis with respect to the center of the viewing window (the center of the camera imaging component.) The Z_0 is the distance (in mm) of the rotation axis from the camera lens (see Figure 5-23.) (X_0, Z_0) is defined with respect to the camera coordinate system — (X,Y,Z), where Y is not relevant.

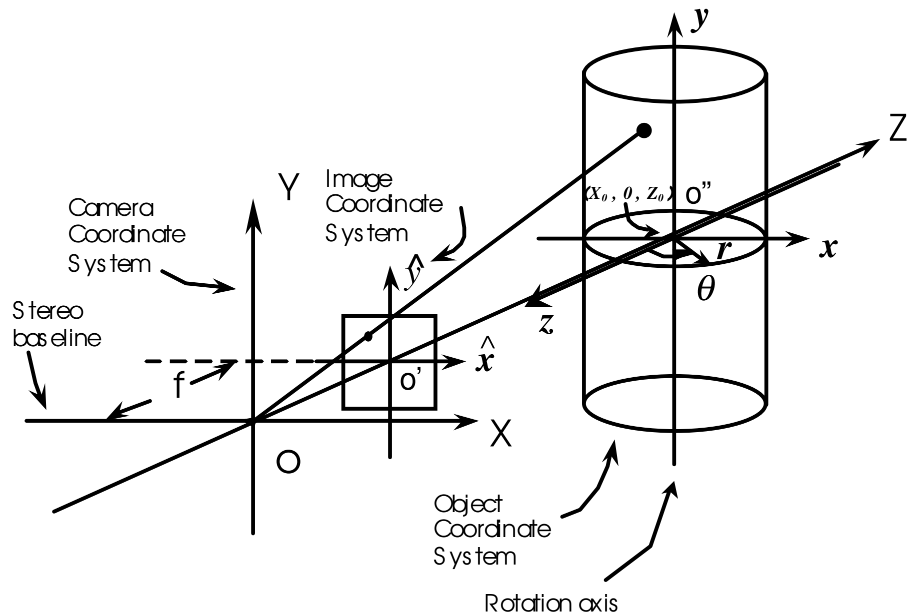


Figure 5-23 Coordinate systems

To obtain the Z_0, a straight vertical iron wire is attached to the base of the rotation stage, which is coincident with the rotation axis. Three methods are used to estimate the distance from the camera lens to the wire. From those results a reliable Z_0 could be obtained. First, manual measurements with a physical tape are carried out. The distance turns out to be about 750 mm from the camera. Next, IFA method — autofocusing method — is used. A vertical flat pattern of high contrast is attached to the wire before running the autofocusing program. After running the autofocusing program several times, an average distance from the wire to the lens is obtained. With the camera lens remains focused, the pattern is removed from the wire. By moving the camera on the linear moving stage the disparity of the tip of the wire at the two camera positions are noted down. Then the distance from the lens to the wire could be found through looking up the camera calibration table for disparity to distance. This third method is the SIA method that is taken to be the most accurate one. The final Z_0 is determined in this three-step procedure.

To obtain the X_0, at the right camera position with the straight vertical iron wire remains attached to the base of the rotation stage and in focus, the wire is rotated 8 times with equal angular displacement, say, 45 degrees. The x coordinates of the tip and the bottom of the wire in these eight rotations are recorded and averaged. The origin of the coordinates of the imaging device is assumed to be at the top left corner of the imaging window. The width of the window is 640 pixels and the height of the window is 480 pixels. Then the x coordinate of the rotation axis with respect to the center of the imaging window could be calculated by the following formula:

$$X_0 = \frac{(\overline{X} - 320) \times p \times Z_0}{f} \quad . \qquad \text{Equation 5-4}$$

Where $\overline{X}$ is the average x value obtained previously. 320 is the half of the width of the viewing window of the camera. p is the physical size of one pixel of the CCD array of the camera. Z_0 is the distance of the rotation axis from the camera as obtained before. f is the focal length of the camera lens.

After the (X_0, Z_0) is determined, it is sent with the collected data to the next stage for processing and recovering their 360-degree shapes. The results of the 360-degree shapes, which will be shown in later sections, verify the success of the Color FOST with Background Removal and Occlusion Detection algorithm. Further improvement could be done with more complicated algorithms. One way of doing that is try to integrate other techniques such as the multiple baseline SIA into it.

5.3 360 Degree Shape Reconstruction

The partial 3D models and the position of the rotation axis are gathered together and integrated into one 360-degree 3D model. The 360-degree 3D model is then written into a 3DMF (3D metafile) file and rendered by Apple's QuickDraw software on a PC. The integration consists of two tasks. One is the stitching of the eight-view shape data. The other is the texture mapping onto the stitched 3D shape. The algorithms and results are presented in the following sections

5.3.1 Stitching of Multiple Views of the Object

In order to simplify integrating the partial 3D models obtained in the previous section, the following assumptions are made.

1. The optical axes of the left and right cameras are perpendicular to the baseline, and parallel to each other. The camera coordinate system is a left handed system at the optical center of the right camera with the X-axis aligned with the baseline and the Z-axis aligned with the optical axis (Figure 5-23).
2. The rotation axis is perpendicular to the plane of the baseline and the optical axis.
3. For every cross-section of the object perpendicular to the rotation axis, the rotation axis is inside the cross-section.
4. The object is simple in the sense described earlier, i.e. for each cross section of the object perpendicular to the rotation axis, there is a 1-1 correspondence between any point on the contour and the angle θ of the

direction vector from the rotation center to that point (Figure 5-24). In other words, in a cylindrical coordinate system (r, θ, y) with it's y-axis coinciding with the axis of rotation, the contour of the object's cross-section at every y can be specified by a function of the form $r(\theta)$.

5. The rotation angles of the object provided by the computer controlled motor are accurate.

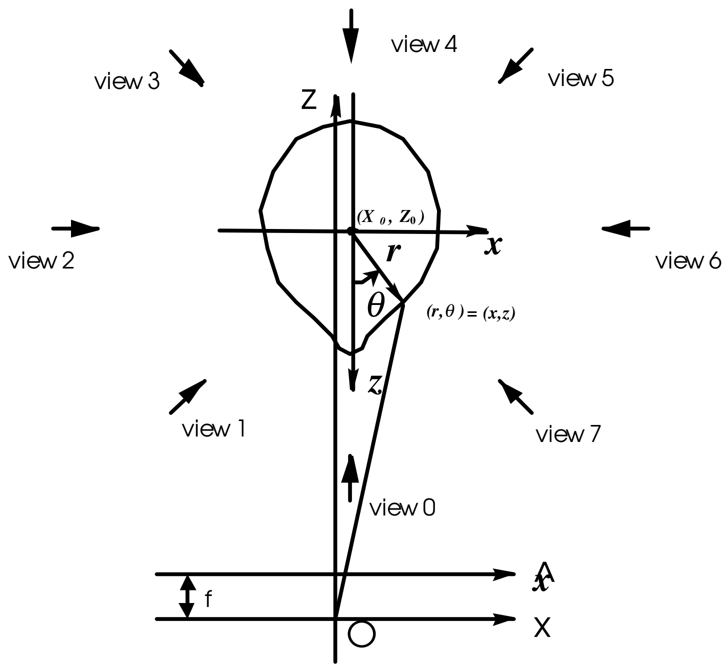


Figure 5-24 A Cross-section of an object and the eight views

Let the rotation axis intersect the XZ-plane at $(X_0,0,Z_0)$ in the camera coordinate system (Figure 5-23). This point is taken as the origin of the object coordinate system for representing the object. It is a cylindrical coordinate system (r, θ, y) with the axis of the cylinder aligned with the rotation axis (parallel to the Y axis) and the angle θ measured with reference to the z-axis. In order to integrate partial 3D models of the object to obtain a complete 3D model of the object, the following algorithm is developed. The algorithm for the case when eight partial 3D models of the objects are used corresponding to eight views of the object at 45 degree intervals is described. The views are referred to with their indices 0,1,2,…,7.

First, each of the eight partial 3D models represented in the camera coordinates (X,Y,Z) is converted to a representation in the cylindrical object coordinate system (r, θ, y). At this point, only the vertices that are part of the object are retained, and other vertices belonging to the background are eliminated. The background points are eliminated by using a threshold on the Z coordinate of the points. In the experiments, points near the left and the right borders of the image are also eliminated (columns 0 to 6 and 21 to 26 in the 27×27 depth-map as these columns contained points very close to the occlusion boundary of the object.) The coordinate (X,Y,Z) of each point is converted to the cylindrical coordinates using the following relations:

$$\theta = \tan^{-1} \frac{X - X_0}{Z_0 - Z}$$
$$r = \sqrt{(X - X_0)^2 + (Z - Z_0)^2} \qquad \text{Equation 5-5}$$
$$y = Y$$

The value of θ computed above is adjusted for rotation of the object for different views. If the views are denoted by $n = 0,1,2,\ldots,7$, and the rotation interval is 45 degrees, then the adjustment for θ in degrees is given by

$$\theta = \theta - n * 45. \qquad \text{Equation 5-6}$$

Having retained only points on the object, then compute cylindrical coordinates of the points, and adjust their θ for rotation, all the points from different views can be merged into one set. In the object coordinate system, this merged set of points represents a discrete sampling of the visible surface of the object. Since the visible surface is assumed to be simple so that it can be represented by a function of the form $r(\theta, y)$ in the cylindrical object coordinate system, the merged set of points can be thought of as a discrete sampling of this function. Given these discrete samples, one can obtain a complete 3D model of the object as follows.

The discrete sample points are used to interpolate and uniformly resample the object's surface in the (θ, y) space. In the experiments in this chapter, the object's surface is uniformly resampled by 27 points in the y space, and 120 points (3 degree intervals) in the θ space. The resulting rectangular sampling grid in the (θ, y) space is used to define a set of vertices (corresponding to sample points) and quadrilaterals (corresponding to rectangles in the grid) that give a piecewise approximation of the object's 3D shape. In order to render this 3D shape on a computer monitor, the coordinates of the vertices are computed in the Cartesian object coordinate system from their cylindrical coordinates as follows:

$$\begin{aligned} x &= r * \sin\theta \\ y &= y \\ z &= r * \cos\theta \end{aligned} \qquad \text{Equation 5-7}$$

In the experiments, a simple separable linear interpolation scheme for resampling is used. First the 27×27 depth-map obtained for each of the eight views of the object is resampled vertically at 27 points along the Y axis. Then the points on the object are represented in the cylindrical object coordinate system. After this, for each value of y, the surface is resampled at 3 degree intervals (120 points) using a simple linear interpolation scheme.

Having modeled the 3D shape of the object by a set of vertices and quadrilaterals, the image texture of the object is modeled by specifying the image texture of each of the quadrilateral. For each quadrilateral, the corresponding image texture can be computed as follows. The quadrilateral in the object space is projected onto one of the focused images obtained from different directions of view in Figure 5-24. A vertex at *(*X,Y,Z*)* (camera coordinates) of a quadrilateral projects to image coordinates $(\hat{x}, \hat{y})$, in the corresponding focused image given by

$$\hat{x} = \frac{X \cdot f}{Z} \qquad \hat{y} = \frac{Y \cdot f}{Z} \qquad \text{Equation 5-8}$$

where f is the focal length of the camera. The viewing direction for projecting a quadrilateral is taken to be that which is closest to the mean θ value of the four vertices of the quadrilateral. The image texture of the quadrilateral is the image within the projection area of the quadrilateral on the focused image. The above method of choosing a focused image for projecting the quadrilateral will be bad if the surface normal to the quadrilateral is almost perpendicular to the direction of view. In this case, the quadrilateral will project onto a very small region. Therefore the image texture will be distorted due to coarse sampling. This will show up when the quadrilateral is viewed directly along its surface normal. A better way is to find all direction of views in Figure 5-24 from which the quadrilateral is visible, and choose that direction for which the dot product of the surface normal of the quadrilateral and the direction of view is a maximum. In the experiments, each quadrilateral is first projected onto a focused image by projecting its vertices. The full focused image is a 432 × 432 image. A rectangular subimage just enclosing the projection of the quadrilateral is extracted (see Figure 5-25).

The boundaries of the extracted subimage are parallel to the boundaries of the full focused image. The subimage enclosing the projected quadrilateral, and the coordinates of the vertices of the projected quadrilateral on the subimage are used by the 3D metafile format for mapping image texture onto the quadrilateral.

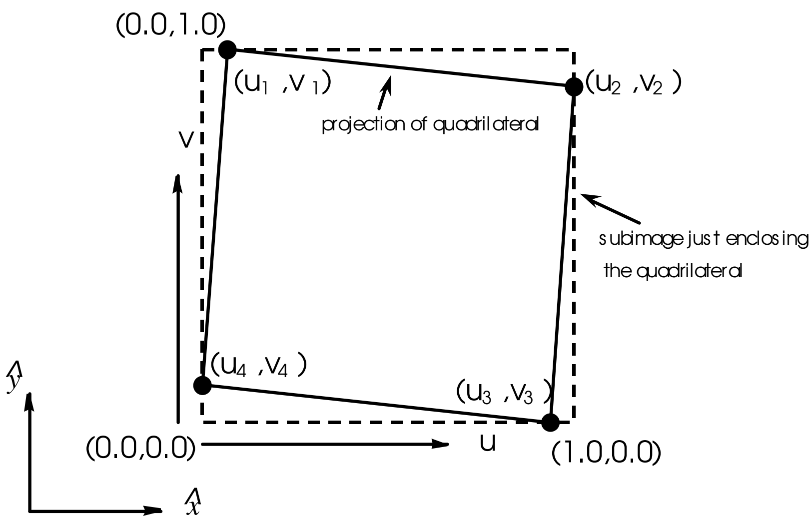
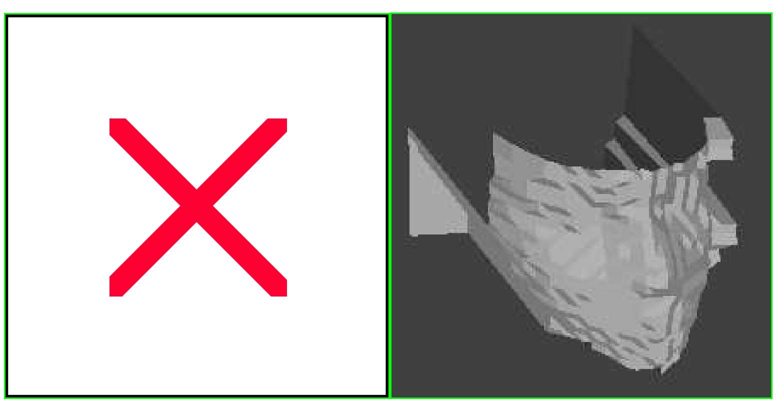


Figure 5-25 Texture Mapping

5.3.2 Recovered 360 degree 3D models

The recovered 360 degree 3D models are shown in FIG.



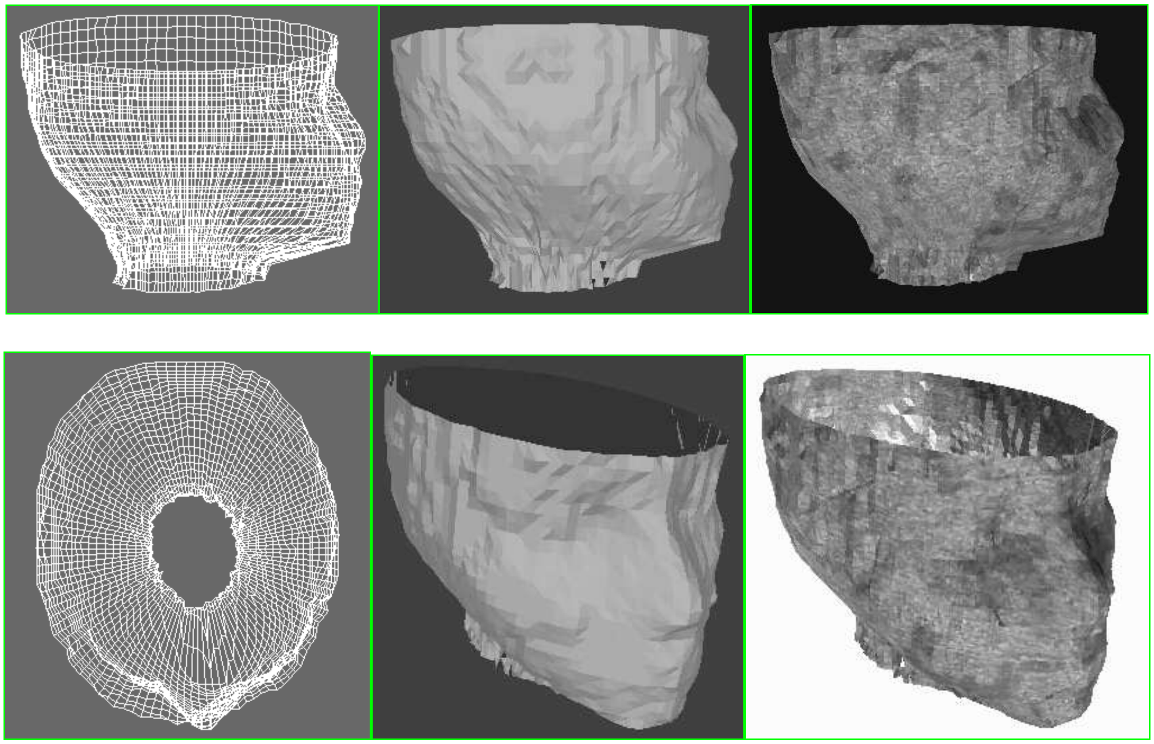
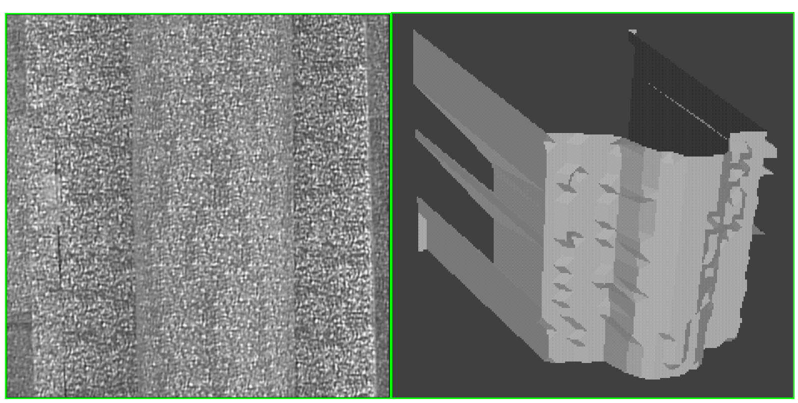


Figure 5-26 Face Object (From left to right, top to bottom: 1. Focused image of the front view, 2. Partial 3D model of the front view, 3. Wire frame view of one side of the 3D model, 4. Non-textured side view, 5. Side view with texture, 6. Wire frame view from the top of the 3D model, 7. Non-textured 3D model, and 8. 3D model with texture.)



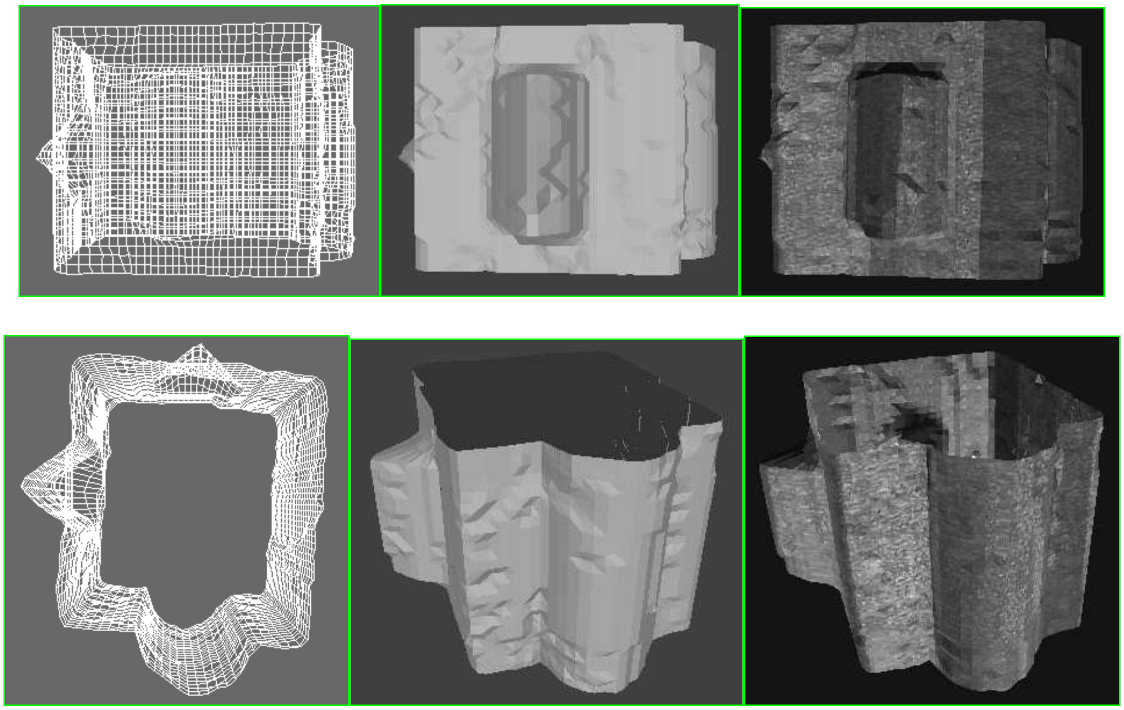
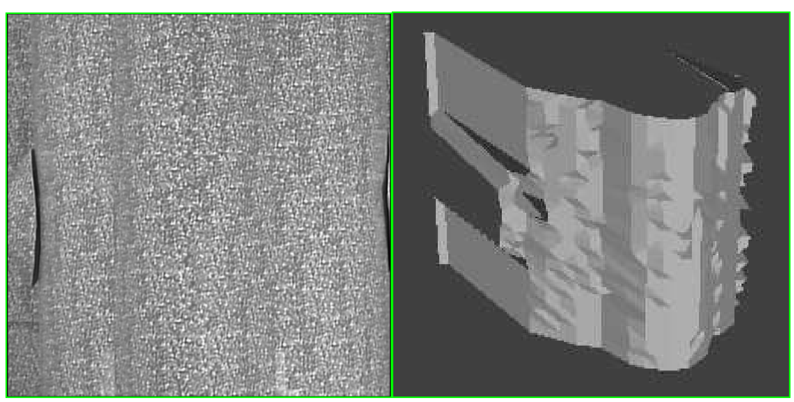


Figure 5-27 Box Object (From left to right, top to bottom: 1. Focused image of the front view, 2. Partial 3D model of the front view, 3. Wire frame view of one side of the 3D model, 4. Non-textured side view, 5. Side view with texture, 6. Wire frame view from the top of the 3D model, 7. Non-textured 3D model, and 8. 3D model with texture.)



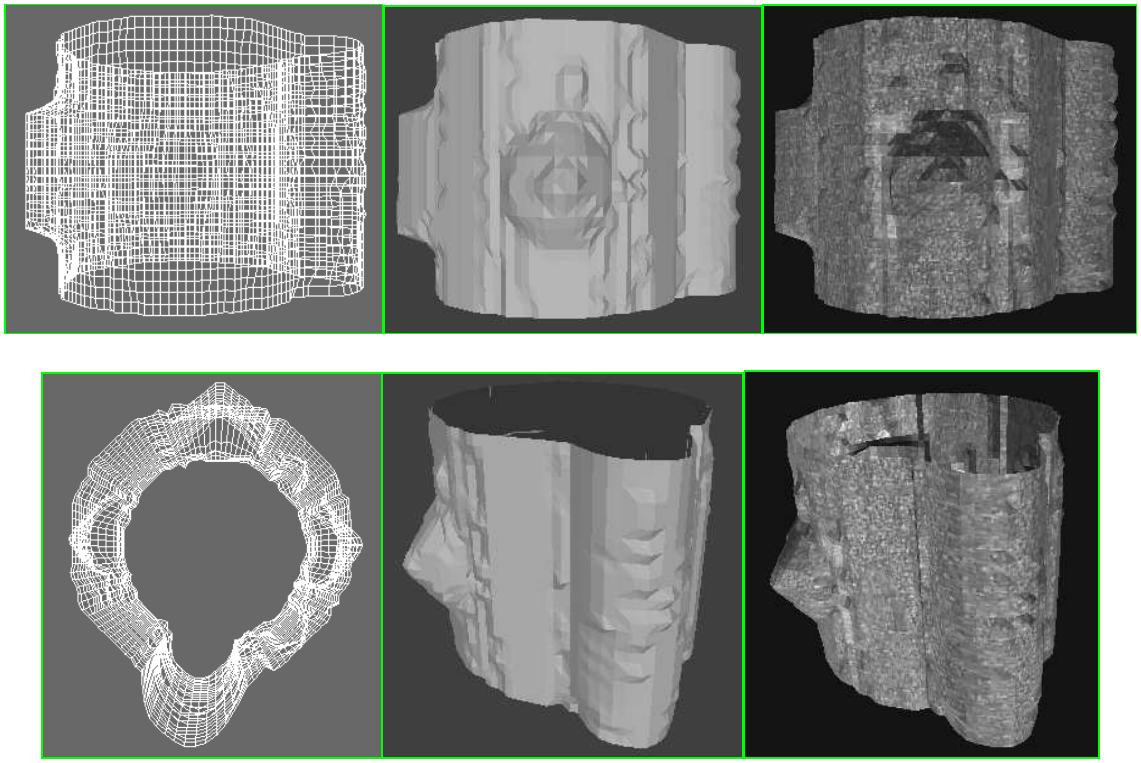
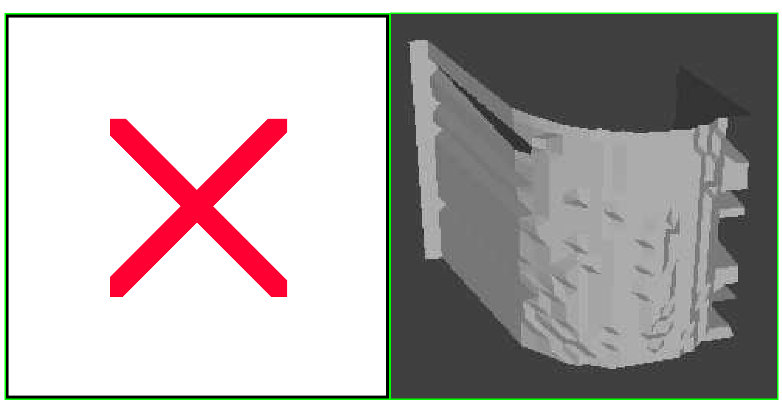


Figure 5-28 Cylinder Object (From left to right, top to bottom: 1. Focused image of the front view, 2. Partial 3D model of the front view, 3. Wire frame view of one side of the 3D model, 4. Non-textured side view, 5. Side view with texture, 6. Wire frame view from the top of the 3D model, 7. Non-textured 3D model, and 8. 3D model with texture.)



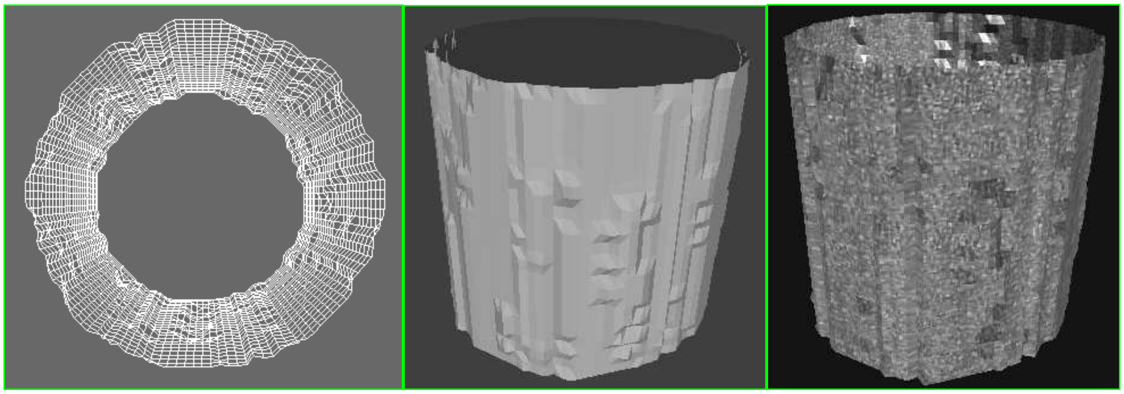


Figure 5-29 Plain Cylinder Object (From left to right, top to bottom: 1. Focused image of the front view, 2. Partial 3D model of the front view, 3. Wire frame view from the top of the 3D model, 4. Non-textured 3D model, and 5. 3D model with texture.)

The recovered 3D models demonstrate the feasibility and capability of the algorithms developed in the current research. More advanced algorithms and more complicated objects will be derived and used in the future research.

5.4 Conclusion

In this chapter a digital vision system for 3D model recovery using focus analysis and stereo matching is presented. Partial 3D models are acquired from eight views of objects, and the partial models are integrated into a complete 3D model. Computational algorithms for stereo matching and partial model integration are presented. Experimental results are presented for three representative objects and one contrast object. In the future, this work will be extended to get more accurate results with fewer views.

Chapter 6 Conclusion

In this dissertation a technique for 3D shape recovery and its associated theories and algorithms are developed and presented. A digital stereo vision system has been built for experimenting on various test objects for the validity and effectiveness of the technique. Solid results are appealing and have prospective usage in various 3D applications. Future work is applicable for enhancing the performance and improving the results in more complicated experiments.

6.1 Summary

Image Focus Analysis (IFA), Image Defocus Analysis (IDA), and Stereo Image Analysis (SIA) provide different ways of recovering three-dimensional shape of an object. Image focus analysis acquires a sequence of images, finds out the focused information among them by focus measure (FM), and then recovers the shape of the object. Image defocus analysis acquires two or three images, computes the degree of blur with defocused information contained in them, and recovers the 3D shape of the scene. Stereo image analysis looks for the conjugate image points in one stereo image pair and recovers the depth of that spatial point through triangulation method. Of the three methods, SIA is the most accurate one but suffers the inherent "correspondence problem." IFA is more accurate than IDA but acquires many more images for processing than IDA. IDA is faster in processing than IFA and has the potential of real-time shape recovery even for moving objects but it needs additional accurate and extensive calibrations on camera blurring characteristics such as to find out the PSF, and so on.

The new method integrates the IDA, IFA, and SIA. It uses IDA to find out a rough shape fast and limits the size of the image sequence acquired by IFA. Next it uses IFA to improve the rough depth map from IDA and generates a more accurate one. Finally it uses the improved depth map as a guide for SIA to avoid correspondence and occlusion problems and also to limit the range of search for matches. The result is a better shape than those of IDA and IFA. Experimental results obtained by this technique have proven its feasibility and effectiveness in recovering 3D shapes.

Experiments are done on a digital stereo camera system named SVIS (Stonybrook VIsion System.) One digital camera is attached to a linear moving stage for moving across to simulate a two-camera stereo vision system. The motorized camera lens could be controlled through a personal computer for focusing objects at different distance from the lens. Calibrations on SVIS are done in several aspects. The SVIS system successfully provides a user-controlled environment for doing experiments. Results obtained by SVIS are reliable and robust in the sense of easy setting up and repeating the experiments without encountering any out-of-control condition, and the results are convincing to have the same quality and are suitable for analysis.

Color image data processing is added in SIA. For more information it provides and more computation it costs, the accuracy of the recovered shape is improved. Multiple baseline SIA and Multiple resolution SIA are also integrated to enhance the accuracy of the recovered shape and also the matching speed. Results are shown to be consistent with the previous ones. Finally, a new concept of “match in blur” is brought forth for improving the stereo matching. Although due to the limit of current equipments the results are not displaying any improvement but they compare well with others. More investigation could be made in the future research.

New algorithms and methods are also carried out for obtaining partial 3D models by the integrated IDA/IFA/SIA technique and for integrating those partial models into one 360-degree model. Background is avoided from stereo matching and occlusion in stereo matching is also dealt with and solved. Different views of objects could be stitched together and mapped with texture. A complete 360-degree 3D model of an object rotated in front of the camera is thus obtained and rendered by rendering software. Several 3D models of different objects are recovered and displayed.

The digital stereo system SVIS stands for a prototype stereo camera system. The algorithms and techniques could be implemented on any stereo camera system, for either special-designed ones or general-purpose ones. The 3D models generated are useful in various fields of 3D applications.

6.2 Future work

This research could be extended or improved in the following aspects:

1. Error analysis: Although qualitatively the results are fairly good judged by human eyes, and at some special points such as the tip of a cone object and the nose of a face object, the real distance from those points to some fixed position (e.g. the background plane) could be measured manually and compared with the ones obtained from the recovery, overall quantitative error analysis is very difficult in the current work. There are two physical

difficulties: First, the true depth map is hard to get with these hand-made objects. Other scientists use computer-simulated objects to test their stereo matching programs. But the second difficulty is that those virtual objects cannot be simulated with IDA and IFA at the present stage. The overall analysis is still in need.

An overall automatic error analysis process needs to be derived and implemented.

2. Advanced stitching algorithm: The stitching algorithm could be improved to get more accurate results with fewer views of more complicated objects.
3. Advanced stereo matching algorithm: The stereo matching algorithm could be improved in many ways such as more complicated matching algorithms with different techniques, different matching criteria, and so on.
4. Advanced hardware: The hardware of the digital camera system could be updated with newer and more powerful ones. New methods such as MIB (match in blur) could be adapted to work on the new hardware to yield better results.

Bibliography

1. A. Gerhard, H. Platzer, J. Steurer, R. Lenz, "Depth Extraction by Stereo Triples and a Fast Correspondence Estimation Algorithm," *ICPR'86*, pp. 512-515, 1986.

2. A. Koschan and V. Rodehorst, "Dense depth maps by active color illumination and image pyramids," Advances in Computer Vision, pp. 137-148, 1997.

3. A. Koschan and V. Rodehorst, "Towards Real-Time Stereo Employing Parallel Algorithms For Edge-Based And Dense Stereo Matching," *Proc. Of the IEEE Workshop on Computer Architectures for Machine Perception*, CAMP'95, pp. 18-20, Sept. 1995.

4. A. Koschan, "Improving Robot Vision by Color Information," *Proc. 7th Int. Conf. on Artificial Intelligence and Information-Control Systems of Robots*, pp. 247-258, Sept. 1997.

5. A. Koschan, V. Rodehorst, K. Spiller, "Color Stereo Vision Using Hierarchical Block Matching and Active Color Illumination," *Proc. 13th Int. Conf. on Pattern Recognition ICPR'96*, Vol. I, pp. 835-839, Vienna, Austria, Aug. 1996.

6. A. Luo and H. Burkhardt, "An Intensity-Based Cooperative Bidirectional Stereo Matching with Simultaneous Detection of Discontinuities and Occlusions," *Int. J. of Computer Vision*, Vol. 15, pp. 171-188, 1995.

7. A. Rosenfeld and A.C. Kak, *Digital Picture Processing*, Vol. 1, Academic Press, New York, 1982.

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

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

10. B. Julesz, *Foundations of Cyclopean Perception*, Chicago, IL: Univ. of Chicago Press, 1971.

11. B. Ross, "A Practical Stereo Vision System", *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 93)*, pp. 148-153, New York, June 1993.

12. B.K.P. Horn, "Focusing," *Artificial Intelligence Memo*, No. 160, MIT, 1968.

13. B.K.P. Horn, *Robot Vision*, McGraw-Hill Book Company, 1986.

14. C.L. Zitnick, T. Kanade, "A Volumetric Iterative Approach to Stereo Matching and Occlusion Detection," *CMU-RI-TR-98-30*, Carnegie Mellon University, Dec. 1998.

15. D. Geiger, B. Ladendorf, A. Yuille, "Occlusions and Binocular Stereo," *Int. J. of Computer Vision*, Vol. 14, pp. 211-226, 1995.

16. D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proc. Royal Soc. London*, Vol. B204, pp. 301-328, 1979.

17. D. Marr, *Vision*, San Francisco, CA: Freeman, 1982.

18. D. Scharstein, "Stereo Vision for View Synthesis," *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 852-858, 1996.

19. D.C. Brockelbank and Y.H. Yang, "An Experimental Investigation in the Use of Color in Computational Stereopsis," IEEE Trans. Systems, Man, and Cybernetics, Vol. 19, No. 6, Nov./Dec. 1989.

20. E. Gurewitz, I. Dinstein, B. Sarusi, "More on the Benefit of a Third Eye for Machine Stereo Perception," *ICPR'86*, pp. 966-968, 1986.

21. E. Hecht, *Optics*, Addison-Wesley Publishing Co., 1987.

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

23. E.P. Krotkov and R. Kories, "Cooperative Focus and Stereo Ranging", *Proceedings The Fourth Conference on Artificial Intelligence Applications*, pp. 76-81, March 1988.

24. E.P. Krotkov, "Exploratory Visual Sensing for Determining Spatial Layout with an Agile Stereo Camera System", *Ph.D. Dissertation MSCIS-87-29*, University of Pennsylvania, May 1987.

25. G. Ligthart and F. Groen, "A Comparison of Different Autofocus Algorithms," *International Conference on Pattern Recognition*, pp. 597-600, 1982.

26. G. Surya, "Three-dimensional scene recovery from image defocus", Ph.D. Thesis, Dept. of Electrical Eng., SUNY at Stony Brook, 1994.

27. H.H Baker and T.O. Binford, "Depth from edge and intensity based stereo," *Proc. 7th Int. Joint Conf. Artificial Intell.*, pp. 631-636, Vancouver, Canada, Aug. 1981.

28. 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.

29. 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.

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

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

32. J.E.W. Mayhew and J.P. Frisby, "Psychophysical and computational studies towards a theory of human stereopsis," *Artificial Intell.*, Vol. 17, pp. 349-385, 1981.

33. 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 Engg., SUNY at Stony Brook, 1997.

34. J.W. Goodman, *Introduction to Fourier Optics*, McGraw-Hill, Inc., 1968.

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

36. M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo", *IEEE Comp. Soc. Conf. Computer Vision and Pattern Recognition*, 1991.

37. M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo", *IEEE Trans. Pattern Anal. Machine Intell.,* vol. 15, no. 4, April 1993.

38. M. Okutomi, O. Yoshizaki, G. Tomita, "Color Stereo Matching and It's Application to 3-D Measurement of Optic Nerve Head," *ICPR'92*, Vol. I, pp. 509-513, 1992.

39. 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.

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

41. M. Subbarao and J.K. Tyan, "The Optimal Focus Measure for Passive Autofocusing and Depth-from-Focus", *Proceedings of SPIE conference on Videometrics IV*, Philadelphia, Oct 1995.

42. M. Subbarao and T. Wei, "Depth from Defocus and Rapid Autofocusing: A Practical Approach," *Proc. of the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Champaign, Illinois, pp. 773-776, June 1992.

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

44. 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.

45. 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.

46. M. Subbarao, "Computational methods and electronic camera apparatus for determining distance of objects, rapid autofocusing, and obtaining improved focus images," U.S. patent application serial number 07/373,996, June 1989 (pending.)

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

48. 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 Engg., SUNY at Sony Brook, New York.

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

50. M. Subbarao, "Spatial-Domain Convolution/Deconvolution Transform," *Tech. Report No. 91.07.03*, Computer Vision Laboratory, Dept. of Electrical Engg., SUNY at Stony Brook, New York.

51. M. Subbarao, N. Agarwal, G. Surya, "Application of Spatial-Domain Convolution/Deconvolution Transform for Determining Distance from Image Defocus," *Tech. Report No. 92.01.18*, Computer Vision Laboratory, Dept. of Electrical Engg., SUNY at Stony Brook, New York, 1992.

52. M. Subbarao, T. Choi, and A. Nikzad, "Focusing Techniques", *Journal of Optical Engineering*, Vol. 32 No. 11, pp. 2824-2836, November 1993.

53. M. Subbarao, T. Wei, G. Surya, "Focused Image Recovery from Two Defocused Images Recorded with Different Camera Settings," *IEEE Trans. on Image Processing*, Vol. 4, No. 12, Dec. 1995.

54. M. Subbarao, T. Yuan, J.K. Tyan, "Integration of Defocus and Focus Analysis with Stereo for 3D Shape Recovery", *Proceedings of SPIE Conference on Three-Dimensional Imaging and Laser-Based Systems for Metrology and Inspection III*, Vol. 3204, Pittsburgh PA, October 1997

55. M. Yachida, Y. Kitamura, M. Kimachi, "Trinoculr Vision: New Approach for Correspondence Problem," *ICPR'86*, pp. 1041-1044, 1986.

56. N. Ayache and F. Lustman, "Trinocular Stereo Vision for Robotics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 13, No. 1, Jan. 1991.

57. Olivier Faugeras *et al.*, "Real Time Correlation-Based Stereo: Algorithm, Implementations and Applications," *Rapport Technique 2013*, INRIA, August 1993.

58. P. Grossman, "Depth from Focus", *Pattern Recognition Letters 5*, pp. 63-69, Jan. 1987.

59. P.P. Banerjee and T.-C. Poon, *Principles of Applied Optics*, Richard D. Irwin, Inc., 1991.

60. R.A. Jarvis, "A perspective on Range Finding Techniques for Computer Vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5, No. 2, pp. 122-139, March 1983.

61. R.C. Gonzalez and R.E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Co., 1993.

62. 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).

63. S.B. Kang and R. Szeliski, "3-D Scene Data Recovery Using Omnidirectional Multibaseline Stereo," *International J. of Computer Vision*, Vol. 25(2), pp. 167-183, 1997.

64. S.H. Lai, C.W. Fu, S. Chang, "A Generalized Depth Estimation Algorithm with a Single Image," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-14, No. 4, pp. 405-411, April 1992.

65. S.K. nayar, M. Watanabe, M. Noguchi, "Real-Time Focus Range Sensor," *CUCS-028-94*, Dept. of Computer Science, Columbia University, New York.

66. S.T. Barnard and M.A. Fischler, "Computational stereo," *Comput. Surveys*, Vol. 14, No. 4, pp. 553-572, Dec. 1982.

67. T. Choi, "Shape and Image Reconstruction from Focus," Ph.D. Thesis, Dept. of Electrical Engg., SUNY at Stony Brook, 1993.

68. T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE Trans. Pattern and Machine Intelligence*, Vol. 16, No. 9, Sept. 1994.

69. T. Kanade, A. Yoshida, K. Oda, H. Kano, M. Tanaka, "A Video-Rate Stereo Machine and Its New Applications", Proceedings of 15th Computer Vision and Pattern Recognition Conference, San Francisco, CA, June, 1996

70. T. Wei, "Three-dimensional machine vision using image defocus", Ph.D. Thesis, Dept. of Electrical Eng., SUNY at Stony Brook, 1994.

71. T. Yuan and M. Subbarao, "Integration of multiple-baseline color stereo vision with focus and defocus analysis for 3D shape measurement", *Proc. SPIE, Three-Dimensional Imaging, Optical Metrology, and Inspection IV*, Vol. 3520, p. 44-51, Dec. 1998.

72. U.R. Dhond and J.K. Aggarwal, "A Cost-benefit Analysis of a Third Camera for Stereo Correspondence," *International J. of Computer Vision*, Vol. 6, No. 1, pp. 39-58, 1991.

73. U.R. Dhond and J.K. Aggarwal, "Stereo Matching in the Presence of Narrow Occluding Objects Using Dynamic Disparity Search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, Jul 1995.

74. U.R. Dhond, J.K. Aggarwal, "Structure from Stereo - A Review", *IEEE Trans. on System, Man, and Cybernetics*, vol. 19, no. 6, November/December, 1989.

75. W. Fellenz, K. Schl ▪ ns, A. Koschan, M. Teschner, "An Active Vision System for Obtaining High Resolution Depth Information," *Proc. 7th Int. Conf. on Computer Analysis of Images and Patterns CAIP'97*, pp. 726-733, Sept. 1997.

76. W.E.L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System*, Cambridge, MA: M.I.T. Press, 1981.

77. Y. Nakamura, T. Matsuura, K. Satoh, Y. Ohta, "Occlusion Detectable Stereo — Occlusion Patterns in Camera Matrix —," *Proceedings of the 1996 Conference on Computer Vision and Pattern Recognition (CVPR '96)*, pp. 371-378, 1996.

78. Y. Xiong and S.A. Shafer, "Depth from Focusing and Defocusing," *CMU-RI-TR-93-07*, Carnegie Melon University, Pittsburgh, Pennsylvania, March 1993.

79. Y.F. Liu, "A Unified Approach to Image Focus and Defocus Analysis," Ph.D. Thesis, Dept. of Electrical Engg., SUNY at Stony Brook, 1998.