

**Three-Dimensional Modeling and Autofocusing
Technology for
New Generation Digital Cameras**

A Dissertation Presented

by

Tao Xian

to

The Graduate School

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

May 2006

Stony Brook University

The Graduate School

Tao Xian

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

Dr. Murali Subbarao, Professor, Advisor
Department of Electrical and Computer Engineering

Dr. Petar M. Djuric, Professor, Chair of Defense
Department of Electrical and Computer Engineering

Dr. Gene R. Gindi, Associate Professor
Department of Electrical and Computer Engineering and
Department of Radiology

Dr. Peisen S. Huang, Associate Professor, Outside Member
Department of Mechanical Engineering

This dissertation is accepted by the Graduate School.

Dean of the Graduate School

Abstract of the Dissertation

Three-Dimensional Modeling and Autofocusing Technology for New Generation Digital Cameras

by

Tao Xian

Doctor of Philosophy

in

Electrical Engineering

Stony Brook University

2006

This dissertation considers three new image-based technologies for modern digital cameras – 3D modeling with zoom calibration, autofocusing, and continuous tracking autofocusing of moving objects

In 3D Modeling, a new technique is introduced to extend current 3D modeling technology for variable zoom settings so that objects of very different sizes and at many different distances can be modeled by the system. A new calibration technique that includes translation and rotation parameters is developed for perspective projection matrices. For each new zoom setting, a new Perspective Projection Matrix (PPM) is dynamically

estimated from the original zoom parameters. Stereo images are rectified based on the generated PPM to simplify stereo matching. A new Vertical Sum of Squared Difference (V-SSD) refinement method is presented to improve the rectification result. To register and integrate partial shapes from eight different views, the rotation axis of multiple views under different zoom settings is needed; a new turntable matrix is dynamically estimated from the initial multi-view calibration without further multi-view calibration. Experimental results are presented to demonstrate the effectiveness of the new dynamic zoom calibration technique.

In autofocusing, several binary mask based Depth-from-Defocus (DFD) algorithms are proposed to improve autofocusing performance and robustness. A binary mask is defined by thresholding image Laplacian to remove unreliable points with low Signal-to-Noise Ratio (SNR). Three different DFD schemes— with/without spatial integration and with/without squaring— are investigated and evaluated, both through simulation and actual experiments. The actual experiments use a large variety of objects including very low contrast Ogata test charts.

A new spatial-domain DFD technique named Blur Equalization Technique (BET) is presented. The theoretical basis of BET relies on equalizing the *blur* or *defocus* of two different images recorded with different camera parameters. In contrast, comparable spatial-domain techniques rely on equalizing the *deblur* or *focus* of the two images. Also, BET facilitates modeling of images locally by higher order polynomials with lower series truncation errors. The accuracy of BET is further enhanced by discarding pixels with low Signal-to-Noise ratio by thresholding image Laplacians, and relying more on sharper of the two blurred images in estimating the blur parameters. These steps make BET a very accu-

rate and robust technique useful in practical applications such as depth recovery and autofocusing. Its performance is compared with three highly effective and significantly improved versions of STM1. BET is found to be superior to some of the best comparable DFD techniques in a large number of both simulation and actual experiments. Also as a counterpart, BM_HIGHORD is implemented for comparison. The only difference between BET and BM_HIGHORD is the blur equalization and the image equalization. Experimental results show the effectiveness of the new blur equalization technique.

Different factors that affect the performance of DFD algorithms are studied. In particular, nonlinear sensor response, lens vignetting, and magnification variation affect the accuracy of DFD. In order to implement DFD on off-the-shelf commercial digital cameras, these factors need to be calibrated and corrected. We present new calibration methods for these three factors. Their correctness and effects on performance of DFD have been evaluated with experiments.

DFD performance on draft mode readout is investigated. Seven different sets of draft readout modes are considered for camera autofocusing. DFD algorithms are implemented for these draft modes without the complex demosaic operation. Experimental results show that an error of about 12 steps (BET), and 14 steps (BM-OSOI), out of 1500 steps is achieved.

Continuous tracking focusing of moving object is realized by combination of 2D tracking and Z tracking. A multiple resolution pyramid SSD matching is proposed for 2D tracking in the first stage. A new multiple base point beta calibration is introduced for Z direction tracking. A direct calibration from lens design data is also studied. The initial results demonstrate that the camera calibration can be completed using the lens

design data only. The trial and error calibration can be replaced, this makes it possible for large scale manufacturing. Also experiments also indicated a novel autofocusing concept - Dynamic Autofocusing.

Dedicated to my family

Contents

List of Figures	xi
List of Tables	xv
Acknowledgements	xvi
1 Introduction	1
1.1 Review of 3D Sensing Techniques	1
1.2 Review of Autofocusing Techniques	4
1.3 Dissertation Organization	9
2 3D Sensing with Dynamic Zoom	12
2.1 Camera Calibration Through Perspective Projection	14
2.2 Dynamic Zoom Calibration	16
2.3 Rectification	21
2.4 Rectification Refinement	22
2.5 Multi-View Rotation Axis Estimation	25
2.6 Experimental Results	28
3 Depth-from-Defocus Techniques	32
3.1 S Transform	33
3.2 STM Autofocusing	34

3.3	Binary Mask Based STM1 Algorithms	37
3.3.1	SNR Mask	37
3.3.2	Spatial Integration	38
3.3.3	Squaring Scheme	39
3.3.4	Simulation Result	39
3.3.5	Experimental Result	41
3.4	Measuring Range Extension	52
3.4.1	Optimal Step Interval	52
3.4.2	Three-Image STM Scheme	53
3.5	Differentiation Filters	58
3.5.1	Chebyshev Polynomial Filter	59
3.5.2	2D Savitzky-Golay Filter	59
3.5.3	2D Laplacian of Gaussian Filter	62
3.5.4	Discussion of Results	63
3.6	Summary	65
4	Blur Equalization Technique (BET)	66
4.1	Blur Equalization Technique	67
4.1.1	Blur Equalization	67
4.1.2	SNR Based Thresholding	70
4.1.3	Image Switching Based on Sharpness Measure	70
4.1.4	Experimental Results	72
4.2	Counterpart: Image Equalization	78
4.3	Summary	81
5	DFD Compensation and Calibration	82
5.1	Nonlinear Sensor Response Compensation	83

5.1.1	Error Analysis of Non-linear Sensor Response	84
5.1.2	Direct Calibration	85
5.2	Lens Vignetting Compensation	87
5.3	Magnification Calibration	92
6	Performance of STM DFD Algorithm for Different Sensor Sub-	
	sampling Schemes	97
6.1	Draft Modes for Depth-from-Defocus	97
6.2	Experimental Results	104
7	Tracking Focusing using DFD	112
7.1	2D Tracking	113
7.1.1	2D Tracking Algorithm	113
7.1.2	2D Tracking Results	115
7.2	Z Tracking	119
7.2.1	Beta Calibration	120
7.2.2	New Beta Calibration Technique	121
7.3	Calibration From Lens Design Data	126
7.4	Dynamic DFD	136
8	Conclusions	137
	Bibliography	140

List of Figures

1.1	Optical configuration of an SLR camera	6
1.2	Principle of phase detection autofocusing	7
2.1	Camera calibration	14
2.2	Intrinsic parameters change with different zoom positions	18
2.3	Orientation parameters change with different zoom positions	19
2.4	Translation parameters change with different zoom positions	20
2.5	Rectification refinement	24
2.6	Partial shapes before/after rectification refinement	25
2.7	Multi-view integration and rotation axis estimation	27
2.8	SVIS-3 3D modeling system	28
2.9	Calibration target	29
2.10	3D models by estimated rotation matrix	30
3.1	Schematic diagram of camera system	34
3.2	Binary mask formation	39
3.3	Image database for IDS1	42
3.4	Sample output of IDS1 simulation system	43
3.5	Sigma table and RMS step error of different STM1 algorithms (synthetic data)	44
3.6	Step number vs. object distance	45

3.7	Sigma vs. step number	46
3.8	Test objects for DFD evaluation	48
3.9	Test object at different positions	49
3.10	Measurement results for different STM1 algorithms (real data)	50
3.11	RMS step error for different STM1 algorithms (real data)	51
3.12	Comparisons at different intervals (synthetic data) (1)	54
3.13	Comparisons at different intervals (synthetic data) (2)	55
3.14	Comparisons at different intervals (real data)	56
3.15	RMS step error for two-image and three-image based schemes (synthetic data)	57
3.16	RMS step error for two-image and three-image based DFD (real data)	58
3.17	Laplacian filter evaluations	64
4.1	Binary masks	71
4.2	Sum of Laplacian in Focusing Window (SLFW)	73
4.3	Switching mechanism	73
4.4	Flow chart of BET algorithm	75
4.5	Sigma vs. step number	76
4.6	Measurement results for test objects (real data)	76
4.7	RMS step error (real data)	77
4.8	Measurement results for BM_HIGHORD (real data)	80
5.1	Setup for nonlinear sensor response calibration	87
5.2	Nonlinear sensor response	88
5.3	Images obtained at different photo-quantity	88
5.4	DFD results before/after nonlinear sensor response calibration	89
5.5	Mean and max offsets before/after nonlinear sensor compensation	89
5.6	Vignetting calibration	92

5.7	Magnification calibration using pattern captured at different steps . . .	94
5.8	Project error from estimated transformation matrix M	95
6.1	Pixel map of draft mode S1	101
6.2	Pixel map of draft mode S2	102
6.3	Pixel map of draft mode S3	103
6.4	Pixel map of draft mode S4	104
6.5	Test objects	105
6.6	DFD performance on draft mode S1 (1/7 line skipping)	106
6.7	DFD performance on draft mode S2 (1/7 line skipping with pixel in- tegration)	107
6.8	DFD performance on draft mode S3 (2/7 line skipping)	107
6.9	DFD performance on draft mode S4 (2/7 line skipping)	108
6.10	DFD performance on draft mode S5 (2/7 line skipping)	108
6.11	DFD performance on draft mode M4 (4×4 pixel mixture)	109
6.12	DFD performance on draft mode M7 (7×7 pixel mixture)	109
7.1	Flowchart of XY 2D tracking	116
7.2	Tracking example	118
7.3	STM continuous focusing	119
7.4	Beta continuous focusing	121
7.5	G calibration table	122
7.6	Beta calibration table: Beta vs. Step	123
7.7	Beta calibration table: Beta vs. $1/u$	124
7.8	Beta tracking results	125
7.9	Beta tracking error	125
7.10	Calibration table generation	127
7.11	Defocus aberration	128

7.12	Calibration table from simulation	129
7.13	PSF changes with object distace (1)	130
7.14	PSF changes with object distace (2)	131
7.15	Sigma vs. object distance	132
7.16	Sigma vs. inverted object distance	132
7.17	Test objects for direct calibration from lens design data	133
7.18	Sigma table calculation from PSF data and actual images (draft mode: S)	133
7.19	Sigma table calculation from PSF data and actual images (draft mode: M4)	134

List of Tables

2.1	Non-linear measure K for different parameters, Zoom setting ranges from step number 70 to 190 at intervals of 10 steps, $N = 13$	17
3.1	Object positions in DFD experiments	48
3.2	DFD performance summary	51
3.3	Savitzky-Golay filter	60
5.1	DFD lens step error by mean and RMS before nonlinear sensor compensation	90
5.2	DFD lens step error by mean and RMS after nonlinear sensor compensation	90
6.1	Sampling scheme summary (line skipping)	99
6.2	Sampling scheme summary (pixel mixture)	100
6.3	Test positions for draftmode readout	106
6.4	DFD performance on different sampling modes	111
7.1	2D window registration time comparison	117

Acknowledgements

I would like to express my deep and sincere gratitude to my academic advisor, Professor Murali Subbarao. Without his constant understanding, encouragement, and personal guidance, this work would not be possible. He has been helping me in a variety of ways: giving me the opportunity to pursue the Ph.D. program, and guiding me through research work.

I am also grateful to my committee members: Professor Peisen S. Huang, Professor Petar M. Djuric, and Professor Gene R. Gindi for their precious time and effort in reviewing this work, and attending my preliminary defense and dissertation defense. Their advice is greatly appreciated.

I would like to acknowledge Dr. Nobuyuki Watanabe, Mr. Arai, Mr. Yachi and Ms. Furuki in Olympus Corporation for their helpful discussion about new generation digital cameras. Special thanks to my colleague Dr. Huei-Yung Lin, Dr. Soon-Yong Park for their suggestions during the early stage of my research work.

I would also like to thank Hui Zhang, Song Zhang, Shizhou Zhang, Jian Zhou, Jiahui Pan, Qingying Hu, Chengping Zhang, Karlheinz Strobl, Mei Yan, Xue Tu, Younsik Kang and all other colleagues and friends, for their help, support and friendship.

I wish to express my deepest gratitude for the constant support, understanding and invaluable love that I received from my wife Tao Duan, and my parents during the past years. I am also very grateful to my parents in law for their understanding and support.

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

Chapter 1

Introduction

With the development of high-quality and inexpensive digital cameras in the last few decades, image-based sensing techniques have become increasingly important in many applications. Some of the important application areas are 3D Modeling, autofocus, and machine vision. Scene 3D information can be extracted by many image-based techniques. In this dissertation, we first describe our research results so far on image-based techniques for 3D model sensing and rapid autofocus. Then we outline some related open research problems as future research topics for this dissertation. In this chapter, 3D sensing and modeling techniques are briefly reviewed in Section 1.1; autofocus techniques are summarized in Section 1.2; and Section 1.3 provides an overview of this dissertation.

1.1 Review of 3D Sensing Techniques

3D sensing techniques can be classified into two groups: contact methods and non-contact methods. A typical contact method uses a Coordinate Measuring Machine (CMM), which is widely used for dimension control in manufacturing. Because of the point-by-point contact measurement, the CMM takes a long time to complete a

measurement.

Optical methods play an important role among the non-contact methods [1, 2, 3, 4, 5, 6, 7, 8]. Optical methods are divided into two groups: active sensing and passive sensing. Active sensing can be understood as a modulation and demodulation procedure. A carrier signal is projected onto the object surface, and the modulated feedback signal is detected by the sensor. The measurement result is obtained from demodulation calculation. Some active sensing methods are Laser Triangulation, Moiré Interferometry, Fringe Projection, and Laser Radar. Laser Triangulation uses a laser point or structured light; Moiré Interferometry maps moiré pattern; Fringe Projection projects a fringe pattern onto the object to be measured; and Laser Radar is based on the measurement of the travel time or phase change of the modulated laser beam. Active methods usually need a complicated hardware system, and normally work in a controlled environment.

Passive sensing methods include Stereopsis, Motion Parallax, Shape from Shading, Shape From Focus and Shape From Defocus. Motion Parallax utilizes local correspondence and/or relative movement between two or more images to find the shape of objects in the scene; Shape from Shading uses different lighting setups to recover the orientation and the depth map of objects; Shape From Focus and Shape From Defocus both acquire images with different parameters of the imaging system. Shape From Focus methods acquire many images with different parameter settings. Then it searches for the best focusing parameter using a focus measure. Shape From Defocus methods compute the degree of blur of acquired two or three images, and recover the depth information in a scene. Stereopsis is a widely used technique for 3D shape reconstruction without active illumination. It recovers a partial shape by finding the corresponding points between two stereo images. The stereo correspondence problem is critical to the partial shape reconstruction. A simple and accurate technique is a correlation-based algorithm [9]. We used SSD (Sum of Squared Differ-

ence) based matching technique on multi-resolution stereo images generated from a Gaussian Pyramid.

Calibrating parameters of a stereo camera system gives position information about corresponding points in 3D space. Based on the relative geometry of the two cameras, there are two types of configuration for a stereovision system - parallel stereo and converged stereo. Parallel stereo configuration has only translation between two camera coordinates, while converged stereo configuration has both the translation and rotation. A perspective projection based calibration is used for the converged stereovision system in our research [10].

To obtain the full model of an object, either the object or the camera has to be moved. A simple way is to rotate the object using a turntable and take pictures at uniform intervals [11]. Since the partial shapes which are constructed from different rotated image pairs are referenced to different camera coordinate systems, it is necessary to register the partial shapes. This requires determining the relative position and orientation between the stereo camera coordinate system and the rotation stage coordinate system using a calibration technique.

After the registration step, multiple partial shapes have to be integrated into a complete 3D model. By assuming a volumetric cube that is visible from all view directions, integration of shapes is done based on voxel grids. Volumetric integration approach uses image space information to remove erroneous points. Implicit surface representation with Shape from Silhouettes technique is used to remove the errors effectively. By labeling 3D voxels based on the visibility from each view direction, partial shapes are integrated in 3D volume and converted to a surface model using the Marching Cubes (MC) algorithm [12].

Conventional 3D modeling uses a fixed zoom setting in reconstruction. For a fixed zoom setting, the relative position of lens components are static. Therefore the camera parameters are constants. When the zoom setting changes, the camera

parameters also vary. There is a great need for extending this technology to variable zoom camera settings so that objects of very different sizes and at many different distances can be modeled by the system.

There are two main problems in employing variable/dynamic zoom in 3D modeling. First, when the zoom changes, many internal camera parameters change. Their change is too complex to be expressed analytically, even for a simple lens system. Second, the relation between the camera coordinate system and the turntable rotation axis changes. Another difficulty arises from the inaccurate and non-linear mechanical control mechanism of a consumer camera. The residual error in positioning the zoom lens by the driving motor cannot be ignored.

Wilson and Shafer [13, 14, 15] introduced a "hold" calibration algorithm in which four camera parameters are selected by an iterative trial procedure. Then up to a 5th degree polynomial is used to estimate the parameters from fixed sampling points. Atienza [16] extended this calibration technique for gaze detection under the assumption that the orientation of the camera coordinate system is unchanged during the zoom change. When there is a noticeable change in zoom setting, the nodal point moves not only changed along the optical axis, but also perpendicular to the optical axis. Therefore the orientation of the camera coordinate is not constant anymore.

1.2 Review of Autofocusing Techniques

There are, again, two types of autofocus techniques: active and passive. Active autofocus systems emit beams of energy, such as ultrasonic, infrared or even structured light, to detect the distance of an object. The reflected beam is received by the detection sensor, and processed by the vision system. Then the system adjusts the lens position accordingly for autofocus. The typical principles used are triangulation, signal time difference, and signal intensity difference. The disadvan-

tage of an active system is the additional requirement of special hardware. For an active autofocus system based on ultrasonic or infrared, there also exist some distance-to-object limitations such as transparent materials (ultrasonic) and an open heat resource (infrared). A laser based system needs a controlled environment and is suitable for industrial purposes [17, 18]. The advantage of an active autofocus system is that it works in dark illumination.

Passive autofocus determines the distance to the object by computer analysis of the image itself. The camera actually “looks” at the scene and drives the lens back and forth searching for the best focus. Some examples are Phase Detection (PD), Depth from Focus (DFF), and Depth from Defocus (DFD). There is no distance-to-object limitation with passive autofocus like there is with active autofocus methods.

Phase Detection (PD) is widely used in high-end Single-Lens Reflex (SLR) cameras. This technology was first patented by Honeywell in 1970s [19, 20]. The autofocus sampling area of an image is split into two areas by a small lens and each is projected onto its own half of the camera’s AF light sensor. The two halves then compare results. In a way passive autofocus is simply an electronic and computerized version of a rangefinder’s focusing system. Fig. 1.1 shows autofocus modules in a SLR camera. Light from the scene passes through the lens onto the reflex mirror. Since the reflex mirror is semi-transparent, the reflected light from the reflex mirror goes up to the view finder through a pentaprism, while the transmitted light is directed onto the autofocus (AF) module by a submirror. The autofocus modules are typically made of several pairs of linear CCDs and corresponding separate lenses. The principle of phase detection is shown in Fig. 1.2. The autofocus sampling area of an image is split into two areas by a small lens and each is projected onto its own half of the camera’s AF light sensor. The two halves then compare results. When these points of light are a set distance apart, the subject is in focus. When the separation

is too small (middle panel), the focus is in front of the object and when it is too large the focus is behind the object. Separator lenses create two identical images, each of which falls on half a pair of CCD sensor arrays. Phase Detection can be implemented through the logic circuit comparator and can achieve high speed focusing. However, one AF module can only respond to a certain area in the scene. Many AF modules are needed in the camera to provide a flexible focusing ability in the whole field of view (FOV). Multiple AF modules add to the complexity of the camera optical design and manufacturing cost of alignment.

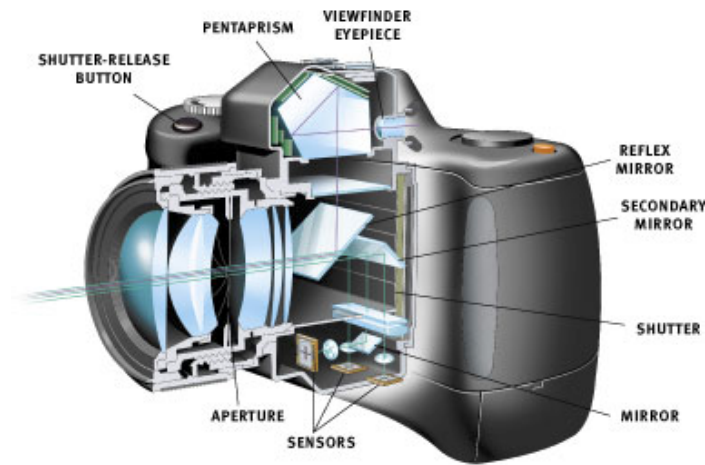


Figure 1.1: Optical configuration of an SLR camera

Depth From Focus (DFF) and Depth From Defocus (DFD) are image based auto-focusing technologies. Depth From Focus (DFF) acquires many images with different parameters. Then it searches for the best focusing parameters using a focus measure. Many research efforts are concentrated on the development and evaluation of different focus measures [21, 22, 23, 24]. Subbarao *et al.* [22] presented a theory for evaluating various focus measures based on the optical transfer function instead of experimental evaluation that may be limited by the selection of specific type of scenes. Two metrics such as Autofocusing Uncertainty Measure (AUM) and Autofocusing Root-

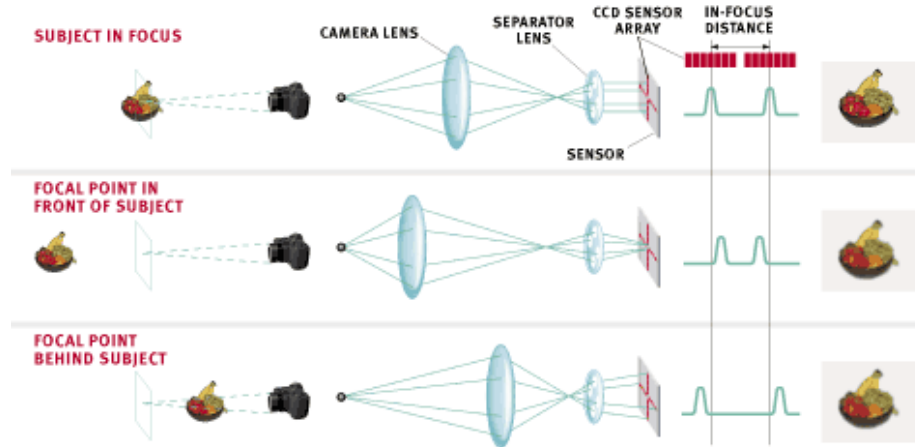


Figure 1.2: Principle of phase detection autofocus

Mean-Square Error (ARMS) were defined to estimate the noise-sensitivity of focus measures in [24]. The ideal focus measure should be unimodal, monotonic, and should reach a maximum when the image is focused. However, due to lens configuration, the residual error of focus motor and device noise, there exist many local maxima. The simple Fibonacci search method would be trapped in a local maxima. A more complicated searching strategy combined with quadratic curve fitting was demonstrated to find the global maximum from theory and experiment [24, 25]. Normally, DFF algorithm is slow because of the multiple image acquisition. It limits its application in real-time autofocusing of a camera system.

Depth From Defocus (DFD) only needs acquiring two or three images, and recovers the depth information in a scene by computing the degree of blur. This makes it suitable for real-time autofocusing and distance measurement. Early DFD algorithms extracted depth information from the blur measurement of an edge [26, 27]. DFD methods for arbitrary objects have been proposed by some researchers. They can be classified as frequency domain approaches [26, 28, 29, 25, 30, 31], spatial domain approaches [32, 33, 34, 35, 36, 37] and statistical approaches [38, 39, 40].

Pentland [26] compared two images, one formed with a pinhole aperture and the other image formed by normal aperture. The depth was recovered from inverse filtering in Fourier domain. Subbarao [28] removed the constraint of the pinhole aperture limitation and allowed several camera parameters to be varied simultaneously. In order to avoid the sensitivity to the frequency spectra of local scene textures, Xiong and Shafer [25] used moment filters to compensate for the frequency spectrum of the texture within the passband of each of the narrowband filters. A large filter bank that densely samples the frequency space is used to achieve better depth accuracy at the cost of computational efficiency. Watanabe and Nayar [31] proposed a class of broadband filters instead of a series of narrow-band filters. They claimed that it is sufficient for depth estimation of scenes with a small number of broadband filters even with complex textural properties. The inverting filter method processes the information in the Fourier domain. It also involves several filtering operations to avoid the filtering sensitivity. It needs more computation compared to the spatial domain method. Statistical approaches normally involve optimization operation which requires more images and computing. Statistical approaches generally need more computation and yield lower depth-map density than spatial domain methods.

The spatial domain approach has inherent advantages since it preserves the spatial correspondence information, which is helpful in various applications such as continuous focusing and object tracking. Moreover it requires less computation and resources. A Spatial-domain Convolution/Deconvolution Transform (S Transform) for n-dimensional signals for the case of arbitrary order polynomials was proposed in [41]. Subbarao and Surya have proposed a spatial domain approach that uses S transform [32, 33, 42, 34]. The S transform is a spatial domain convolution/deconvolution transform developed for images and n-dimensional signals for the case of arbitrary order polynomials. Using an appropriate smoothing filter, the images taken at different parameters could be fitted to a cubic polynomial, and the blur estimation can

be solved in the spatial domain. There are two basic variations: STM1 changes the focus step or lens position and STM2 varies the aperture diameter. This technique is explained in detail in Chapter 3. Ziou [43, 35, 44] fitted the images to a Hermite polynomial basis. They show that any coefficient of the Hermite polynomial computed using the more blurred image is a function of the partial derivatives of the other image and the blur difference. Thus the blur difference can be computed by resolving a system of equations. This work is seen as a continuation of the S transform method. Favaro *et al.* [36] use a singular value decomposition technique to estimate the operator that spans a left null space from a number of deblurred images.

1.3 Dissertation Organization

The goal of this dissertation is to develop novel image-based techniques for 3D modeling and autofocusing.

In Chapter 2, a new calibration technique is introduced to extend current 3D modeling technology for variable zoom settings so that objects of very different sizes and at many different distances can be modeled by the system. This novel calibration technique that includes translation and rotation parameters is developed for projective matrices. For each new zoom setting, a new projection matrix is dynamically estimated from the original zoom parameters. Stereo images are rectified based on the generated projection matrix to simplify stereo matching. A new Vertical SSD (V-SSD) refinement method is presented to improve the rectification result. To register and integrate partial shapes from eight different views, the rotation axis of multiple views under different zoom settings is needed; a new turntable matrix is dynamically estimated from the initial multi-view calibration without further multi-view calibration. Experimental results are presented to demonstrate the effectiveness of the new dynamic zoom calibration technique.

In Chapter 3, several binary mask based Depth From Defocus (DFD) algorithms are proposed to improve autofocusing performance and robustness. A binary mask is defined by thresholding image Laplacian to remove unreliable points with low Signal-to-Noise Ratio (SNR). Three different DFD schemes– with/without spatial integration and with/without squaring– are investigated and evaluated, both through simulation and actual experiments. The actual experiments use a large variety of objects including very low contrast Ogata test charts.

In Chapter 4, a new spatial-domain Depth-from-Defocus (DFD) technique named Blur Equalization Technique (BET) is presented. The theoretical basis of BET relies on equalizing the *blur* or *defocus* of two different images recorded with different camera parameters. In contrast, comparable spatial-domain techniques rely on equalizing the *deblur* or *focus* of the two images. Also, BET facilitates modeling of images locally by higher order polynomials with lower series truncation errors. The accuracy of BET is further enhanced by discarding pixels with low Signal-to-Noise ratio by thresholding image Laplacians, and relying more on sharper of the two blurred images in estimating the blur parameters. These steps makes BET a very accurate and robust technique useful in practical applications such as depth recovery and autofocusing. Its performance is compared with three highly effective and significantly improved versions of STM1. BET is found to be superior to some of the best comparable DFD techniques in a large number of both simulation and actual experiments. Also as a counterpart, BM_HIGHORD is implemented for comparison. The only difference between BET and BM_HIGHORD is the blur equalization and the image equalization. Experimental results show the effectiveness of the new blur equalization technique.

In Chapter 5, different factors that affect the performance of DFD algorithms are studied. In particular, nonlinear sensor response, lens vignetting, and magnification variation affect the accuracy of DFD. In order to implement DFD on off-the-shelf commercial digital cameras, these factors need to be calibrated and corrected. In this

chapter, we present new calibration methods for these three factors. Their correctness and effects on performance of DFD have been evaluated with experiments.

In Chapter 6, DFD performance on draft mode readout is investigated. Seven different sets of draft readout modes are investigated for camera autofocus. DFD algorithm is directly tested on these draft modes without using demosaic operation. Demosaic operation normally requires large computation. Experimental results show that an error of about 12 steps (BET), and 14 steps (BM-OSOI), out of 1500 steps is achieved in actual camera systems.

In Chapter 7, a multiple resolution pyramid SSD matching is proposed for 2D tracking in the first stage. A new multiple base point beta calibration is introduced for Z direction tracking. 3D DFD tracking focusing is realized by combining of 2D tracking and Z tracking. A direct calibration from lens design data is also studied. Initial results demonstrate that the camera calibration can be completed using the lens design data only. The trial and error calibration can be replaced.

Future research topics are outlined in Chapter 8.

Chapter 2

3D Sensing with Dynamic Zoom

Recent advances in consumer digital cameras have made low-cost 3D modeling systems feasible. Conventional 3D modeling techniques use a fixed zoom setting in 3D reconstruction. For objects at different distances and/or of different sizes, a vision system with variable zoom is critical for 3D modeling.

In the case of fixed zoom setting, the relative positions of lens components are static. When the zoom setting changes, the camera parameters also vary. To extend the fixed zoom setting camera model to adjustable zoom settings, several algorithms have been presented. Wilson and Shafer [13, 14, 15] introduced an iterative trial and error procedure in which four camera parameters are selected. These camera parameters are – the effective focal length f , the image center (u_0, v_0) , and the translation along the optical axis T_3 . Up to a 5th degree polynomial is used to estimate the camera parameters from fixed sampled points. Atienza and Zelinsky [16] extended this calibration technique to gaze detection under the assumption that the orientation of the camera coordinate remains unchanged during zoom change. However when the optical configuration of a vision system changes, this assumption is not valid, and a trial and error procedure will be needed to determine the critical parameters.

There are several main problems in employing variable/dynamic zoom in 3D modeling. First, many internal camera parameters vary nonlinearly with different zoom settings. Their variations are too complex to be expressed analytically, even for a simple lens system. Second, the relation between the camera coordinate system and the turntable rotation axis changes as it wobbles. Another difficulty arises from the inaccurate and non-linear mechanical control mechanism of a consumer camera. The residual error in positioning the zoom lens by the driving motor cannot be ignored.

The partial shape of a single view can be reconstructed from stereo images using a stereo matching technique. Stereo image rectification utilizes the epipolar geometry to reduce the search dimension of stereo matching from 2D to 1D, and also decreases the possibility of mismatches. In our research, we adopted a compact rectification algorithm for stereo pairs proposed by Fusiello *et al.* [45]. The rectification is based on the camera's intrinsic parameters, mutual position, and orientation. However, rectification imposes a higher accuracy requirement on camera parameters. Due to the non-linearity of lenses and the inaccuracy of mechanical parts, parameters from dynamic estimation are not accurate enough for a perfect rectification. A refinement based on vertical profile SSD is presented to reconstruct the partial shape from estimated projection matrix. For registering and integrating partial shapes, the rotation axis is estimated without further multi-view calibration.

In this chapter, a full zoom calibration is presented to avoid the empiric trial procedure. The error of estimated camera parameters from dynamic zoom is analyzed. A new rectification refinement technique is proposed to obtain a better partial shape. And the rotation axis of multi-view at different zoom setting is also estimated without further calibration. Full 3D models using the estimated rotation axis are demonstrated.

2.1 Camera Calibration Through Perspective Projection

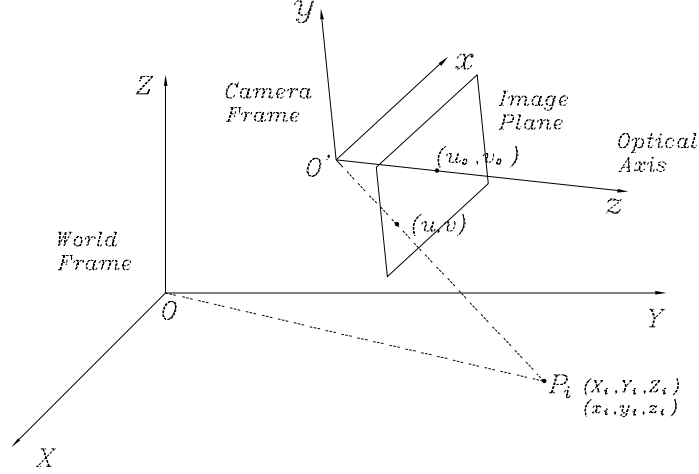


Figure 2.1: Camera calibration

Camera calibration is to find a mapping from 3D world frame to 2D image plane. It can be divided into two parts: first a rotation and translation between world frame and camera frame, then a perspective projection from camera frame to image plane, as shown in Fig. 2.1. Some of the calibration algorithms are direct calibration, Tsai's calibration [46], and Zhang's calibration [47]. In this research, the calibration based on perspective projection matrix is adopted [10]. However the dynamic zoom calibration method should be of general value, and can be extended to other algorithms.

A 3D point (X_i, Y_i, Z_i) in world coordinate is projected to a point (u_i, v_i) in the image plane; the corresponding perspective projection matrix (PPM) P can be expressed as:

$$\begin{bmatrix} u_i & v_i & 1 \end{bmatrix} = P \begin{bmatrix} X_i & Y_i & Z_i & 1 \end{bmatrix}^t \quad (2.1)$$

with

$$u_i = f \frac{x_i}{z_i} = f \frac{p_{11}X_i + p_{12}Y_i + p_{13}Z_i + p_{14}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \quad (2.2)$$

$$v_i = f \frac{y_i}{z_i} = f \frac{p_{21}X_i + p_{22}Y_i + p_{23}Z_i + p_{24}}{p_{31}X_i + p_{32}Y_i + p_{33}Z_i + p_{34}} \quad (2.3)$$

where (x_i, y_i, z_i) is the corresponding 3D point in the camera frame.

In the case that the calibration pattern contains more than 6 corresponding point pairs, P_0 can be estimated through least squares techniques. If we assume we are given N matches for the homogeneous linear system, we have:

$$Ap = 0 \quad (2.4)$$

with

$$A = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 & -x_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 & -y_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2Z_2 & -x_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2Z_2 & -y_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_NX_N & -x_NY_N & -x_NZ_N & -x_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_NX_N & -y_NY_N & -y_NZ_N & -y_N \end{bmatrix} \quad (2.5)$$

and

$$p = \left[p_{11}, p_{12}, \cdot \cdot \cdot, p_{33}, p_{34} \right]^t \quad (2.6)$$

Since A has rank 11, the vector p can be recovered from SVD related techniques as the column of V corresponding to the smallest singular value of A. Since P_0 is defined up to a scale factor, we can write:

$$P = \frac{P_0}{\gamma} \quad (2.7)$$

The absolute value of the scale factor $|\gamma|$ can be calculated from rotation components.

Since there are 8*6 world-point matches in our zoom calibration pattern, the least-square minimization technique is used to reduce the estimation error.

2.2 Dynamic Zoom Calibration

The PPM can be decoupled into an intrinsic matrix that describes the projection from camera coordinate to image plane, and an extrinsic matrix which describes the transform from the world coordinate system to the camera coordinate system. The factorization is expressed as:

$$P = I[R|t] \tag{2.8}$$

The intrinsic matrix I depends on the intrinsic parameters, and has the following format:

$$I = \begin{bmatrix} f_u & \alpha & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.9}$$

where f_u, f_v are focal length in effective pixel size along u and v direction of the image plane, (u_0, v_0) is the coordinate of the image center, and α is the aspect ratio. The extrinsic matrix describes the rotation and translation of the camera coordinate system, and can be expressed by a 3*3 rotation matrix R and a translation vector t . In the rotation matrix R , all the 9 elements are not independent. They will be further reduced to 3 independent rotation angles (roll, yaw, pitch) using 6 orthonormal constraints.

In dynamic zoom calibration, a series of perspective projection calibrations are conducted at a set of base points. Then perspective projection matrices are decomposed as intrinsic parameters, orientation angles, and translation vectors. These parameters change with different zoom positions. They are plotted respectively in

Fig. 2.2, Fig. 2.3 and Fig. 2.4. Camera parameters at an arbitrary intermediate zoom setting are estimated from the nearest calibrated zoom positions by interpolation. In the figures we observe that the changes of f_u , f_v , t_3 are similar to the result of Wilson and Atienza. However, for a convergent stereo vision configuration, the orientation of the camera coordinate (roll, yaw, pitch) is not a constant anymore, as can be seen in Fig. 2.3. Moreover, the optical center of the camera moves not only along the optical axis, but also shifts in a plane that is perpendicular to the optical axis.

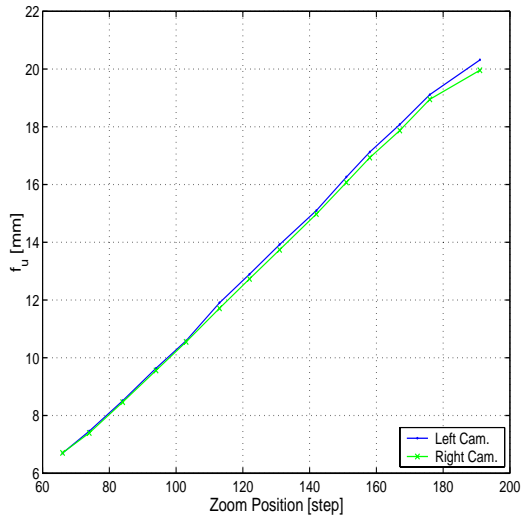
A nonlinear measure K is used as an index that indicates the relative error between the estimated parameters and the real ones:

$$K = \frac{1}{N} \sum_{i=1}^T \frac{|p_i - \pi(z_i, \alpha_0, \alpha_1, \dots, \alpha_n)|}{p_i} \quad (2.10)$$

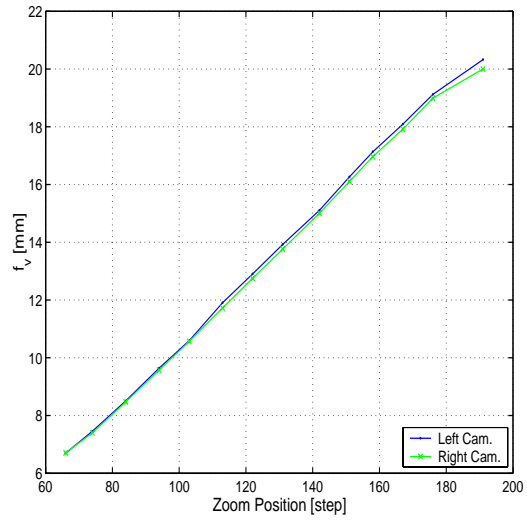
where p_i is the calibrated parameter and $\pi(z_i, \alpha_0, \alpha_1, \dots, \alpha_n)$ is the parameter estimated from dynamic zoom. In Table 2.1, the nonlinearity measure K is calculated by comparing the estimated camera parameters and the parameters from real calibration of left and right cameras at 13 different zoom positions. It demonstrates that t_1 , t_2 , u_0 , v_0 have relatively large estimation errors. This is caused by the non-linearity of the lens design.

K [%]	f_u	f_v	u_0	v_0	<i>roll</i>	<i>yaw</i>	<i>pitch</i>	t_1	t_2	t_3
Left	0.58	0.58	1.34	0.81	0.04	0.70	0.23	3.22	1.45	0.25
Right	0.50	0.50	1.13	0.78	0.02	0.84	0.16	1.53	0.90	0.18

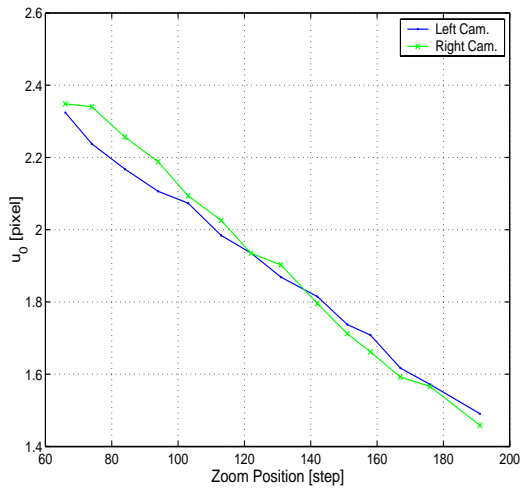
Table 2.1: Non-linear measure K for different parameters, Zoom setting ranges from step number 70 to 190 at intervals of 10 steps, $N = 13$



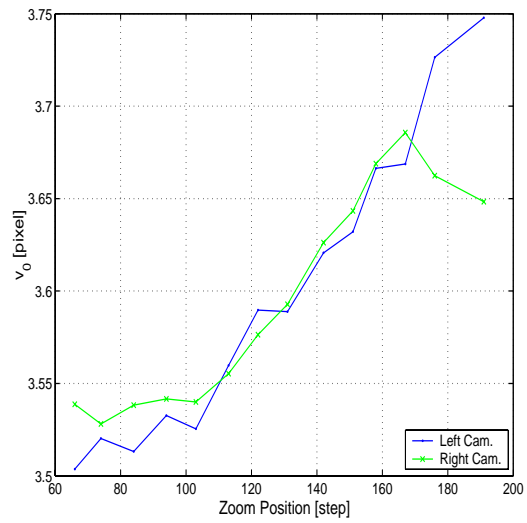
(a) f_u



(b) f_v

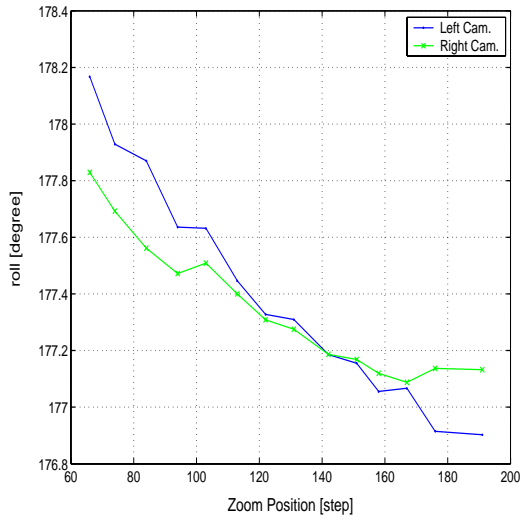


(c) u_0

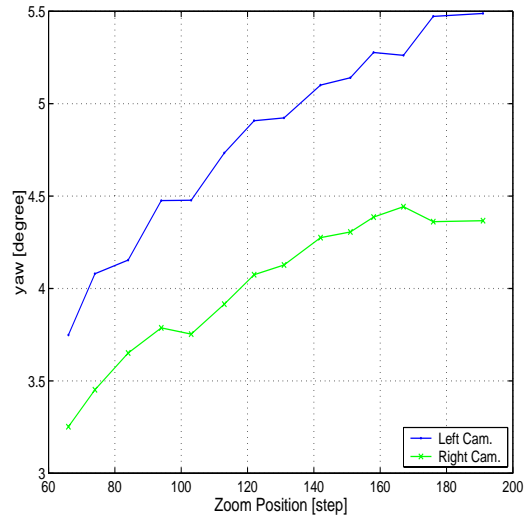


(d) v_0

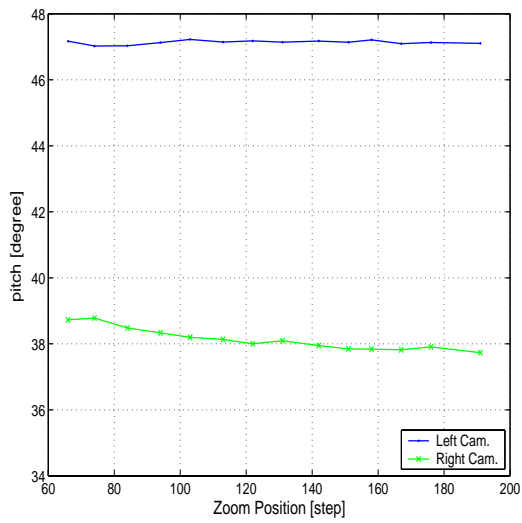
Figure 2.2: Intrinsic parameters change with different zoom positions



(a) Roll

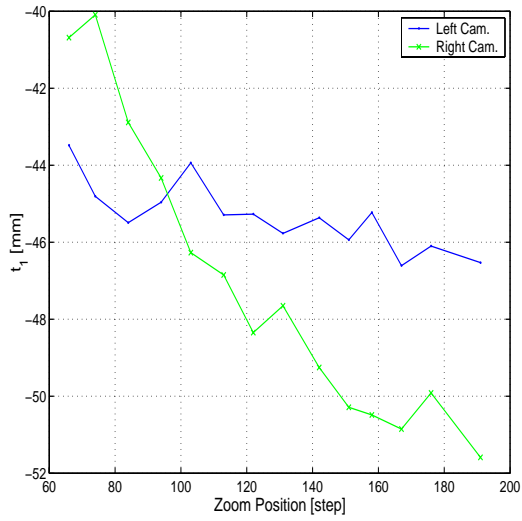


(b) Yaw

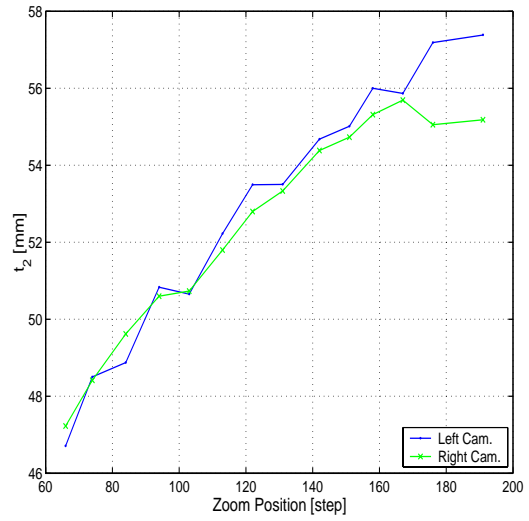


(c) Pitch

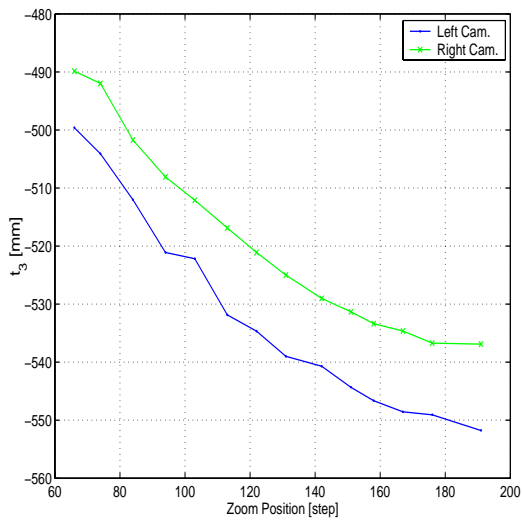
Figure 2.3: Orientation parameters change with different zoom positions



(a) t_1



(b) t_2



(c) t_3

Figure 2.4: Translation parameters change with different zoom positions

2.3 Rectification

While matching stereo images, rectification is used to reduce computation and the possibility of mismatching. Rectification based on PPM is briefly introduced here. In order to have horizontal epipolar lines, the baseline must be parallel to the new X axis of both cameras. In addition, corresponding points must have the same vertical position (Y coordinate). Consequently, the position of new optical centers is the same as that in the old ones after suitable rotations, and intrinsic parameters are the same for both cameras. Therefore, the new projection matrices will differ only in their optical centers.

Let us write the new PPMs in terms of their QR factorization:

$$P_{n1} = I[R|-Rc_1] \quad (2.11)$$

$$P_{n2} = I[R|-Rc_2] \quad (2.12)$$

The intrinsic parameter matrix I is same for both new projective matrices. The rotation matrix R is the same for both PPMs.

To rectify the left image and the right image, we need to compute a transformation mapping of the image plane $P_{o1} = [Q_{o1}|q_{o1}]$ onto the image plane $P_{n1} = [Q_{n1}|q_{n1}]$. We will see that the sought transformation is the collinearity given by 3*3 matrix $T_1 = Q_{n1}Q_{o1}^{-1}$. The same result applies to the right image.

For any 3D point w , we can connect it to a corresponding point m on the image plane by a PPM P . Thus for the same 3D point w , there are two points on the image plane that correspond to before and after rectification respectively.

$$m_{o1} = P_{o1}w \quad (2.13)$$

$$m_{n1} = P_{n1}w \quad (2.14)$$

Then the optical rays that connect image points m_{o1} , m_{n1} , and the optical center are described in parametric form as:

$$w = c_1 + \lambda_o Q_{o1}^{-1} m_{o1} \quad \lambda_o \in R \quad (2.15)$$

$$w = c_1 + \lambda_n Q_{n1}^{-1} m_{n1} \quad \lambda_n \in R \quad (2.16)$$

From Eqn. (2.15) and (2.16), we have:

$$m_{n1} = \lambda Q_{n1} Q_{o1}^{-1} m_{o1} \quad \lambda \in R \quad (2.17)$$

where λ is an arbitrary scale factor.

Reconstruction of 3D points by triangulation can be performed from the rectified image directly, using P_{n1} , P_{n2} .

2.4 Rectification Refinement

Rectification uses PPM from calibration or estimation as a starting point. In a successful rectification, the vertical disparity between the left and the right image pair should be zero. For this, we need an accurate PPM. However, in the case of dynamic zoom, due to the non-linearity of the lens and mechanical mechanisms, there will be errors in the estimated camera parameters. When the projection matrix for the dynamic zoom case is not accurate enough for rectification, the left and right image pair may have vertical shift of up to several pixels. This problem may be solved by increasing the search range of stereo matching at the cost of dramatically increasing the computation and the possibility of mismatches. Then the advantage of rectification is diminished. A rectification refinement is needed to speed up computation and reduce the stereo match error.

An analytical rectification refinement is very difficult due to the lack of constraints, if not impossible. In Table 2.2, we see that the main error source of zoom calibration

is t_1, t_2, u_0, v_0 due to the rotational nature of optical zoom lens design. Based on this observation, a refinement technique is introduced based on vertical profile Sum of Squared Difference (SSD). The vertical direction profiles of left and right cameras are obtained by projecting images onto the Y-axis:

$$Y_l(i) = \sum_{j=1}^N I_l(i, j) \quad (2.18)$$

$$Y_r(i) = \sum_{j=1}^N I_r(i, j) \quad (2.19)$$

where $I(i, j)$ is the gray level of images. Then an SSD is computed in a window of width W . The vertical image shift is calculated by Eqn. (2.20) and (2.21). The refined rectification is obtained by moving one image relative to the other one in the vertical direction by d .

$$c(i) = \sum_{k=-W}^W \psi \left[Y_l(i), Y_r(i + k) \right] \quad (2.20)$$

where ψ is the vertical SSD operator.

$$d = \min \left[c(i) \right] \quad (2.21)$$

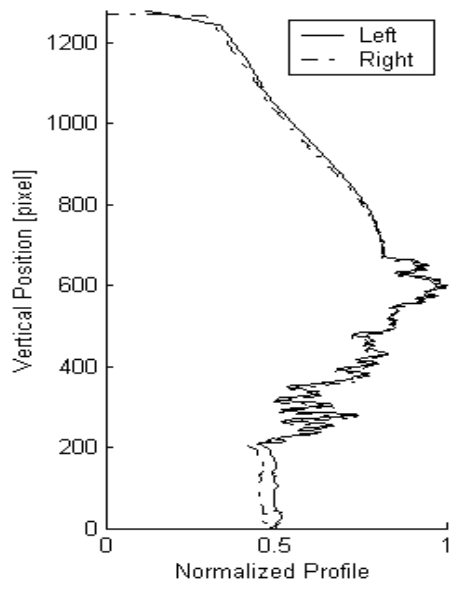
Fig. 2.5 (a) and (b) show images of a toy dog recorded by the left and the right cameras. The size of images is 960*1280 pixels. Their normalized vertical profile is plotted in (c). The vertical line difference from Y-SSD is shown in (d). The width of the SSD window is 180 pixels. There are three areas. Area 1 and Area 3 are noise-dominated due to non-uniform illumination. Area 2 is object-dominated, and the vertical line difference is a constant (5 lines). The partial shapes before and after rectification refinement are shown in Fig. 2.6 (a) and (b) respectively.



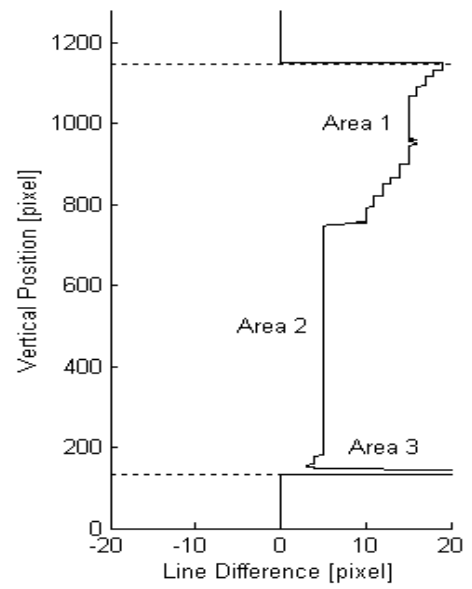
(a) Left image



(b) Right image



(c) Vertical profile of left and right image



(d) SSD result

Figure 2.5: Rectification refinement

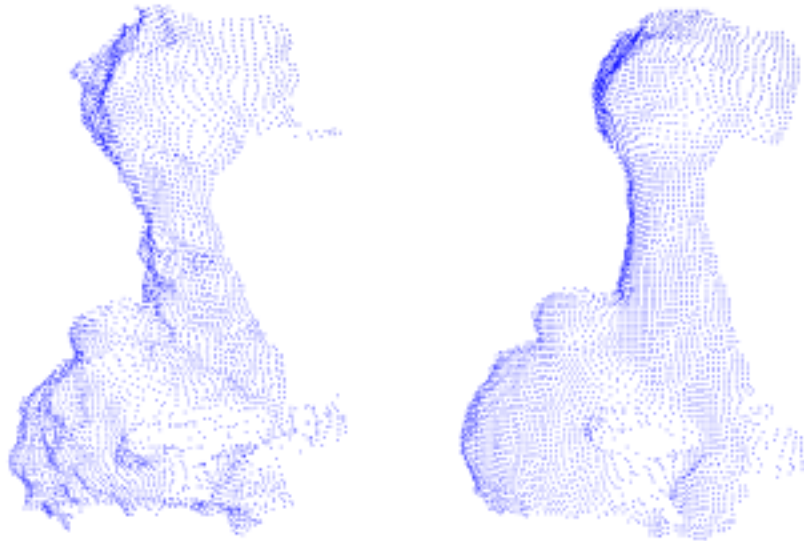


Figure 2.6: Partial shapes before/after rectification refinement

2.5 Multi-View Rotation Axis Estimation

For a complete 3D model, single partial shape from one view is not enough. A rotation stage is used in our stereo vision system to rotate the object. It is equivalent to fixing the object and rotating the stereo camera. In Fig. 2.7, the full 3D model are integrated from 8 partial shapes, and each partial shape is obtained from different views which range from View 1 to View 8. Since the partial shapes are referenced to different camera coordinate systems, it is necessary to register the partial shapes. Multi-view calibration describes the position and orientation of the rotation axis around which the different partial shapes are measured. The rotation axis is expressed by a turntable matrix (4×4). A multi-view calibration method that is similar to the camera perspective projection calibration has been developed. The result from multi-

view calibrations can be expressed as a turntable matrix:

$$T = \begin{bmatrix} R_{cs} & t_{cs} \\ 0 & 1 \end{bmatrix} \quad (2.22)$$

where R_{cs} is a 3×3 rotation transform, and t_{cs} is the translation vector with respect to the fixed optical center O_1 or O_2 .

As described in Section 2.2, the optical center of cameras wobbles around the optical axis with different zoom settings. As shown in Fig. 2.7, the optical center of the left and right camera shift from O_1, O_2 to O_{1n}, O_{2n} with the change of zoom position. That means the turntable matrix, which describes the rotation axis with respect to the optical center, also changes. However the origin of the world coordinate is still fixed and can be used as the connection between the epipolar geometry before and after zoom change.

The turntable matrix of the dynamic zoom can be calculated from a calibrated rotation matrix T , a calibrated PPM P_s , and the estimated PPM P_z from dynamic zoom. The new turntable matrix is obtained by:

$$T_n = \begin{bmatrix} R_n & t_n \\ 0 & 1 \end{bmatrix} \quad (2.23)$$

where:

$$R_n = (R_s^{-1}R_z)R_{cs} \quad (2.24)$$

$$t_n = (t_s - t_z) + t_{cs} \quad (2.25)$$

and R_s, R_z, t_s, t_z are factorized from the calibrated PPM P_s and estimated PPM P_z [45, 48]:

$$P_s = I[R_s|t_s] \quad (2.26)$$

$$P_z = I[R_z|t_z] \quad (2.27)$$

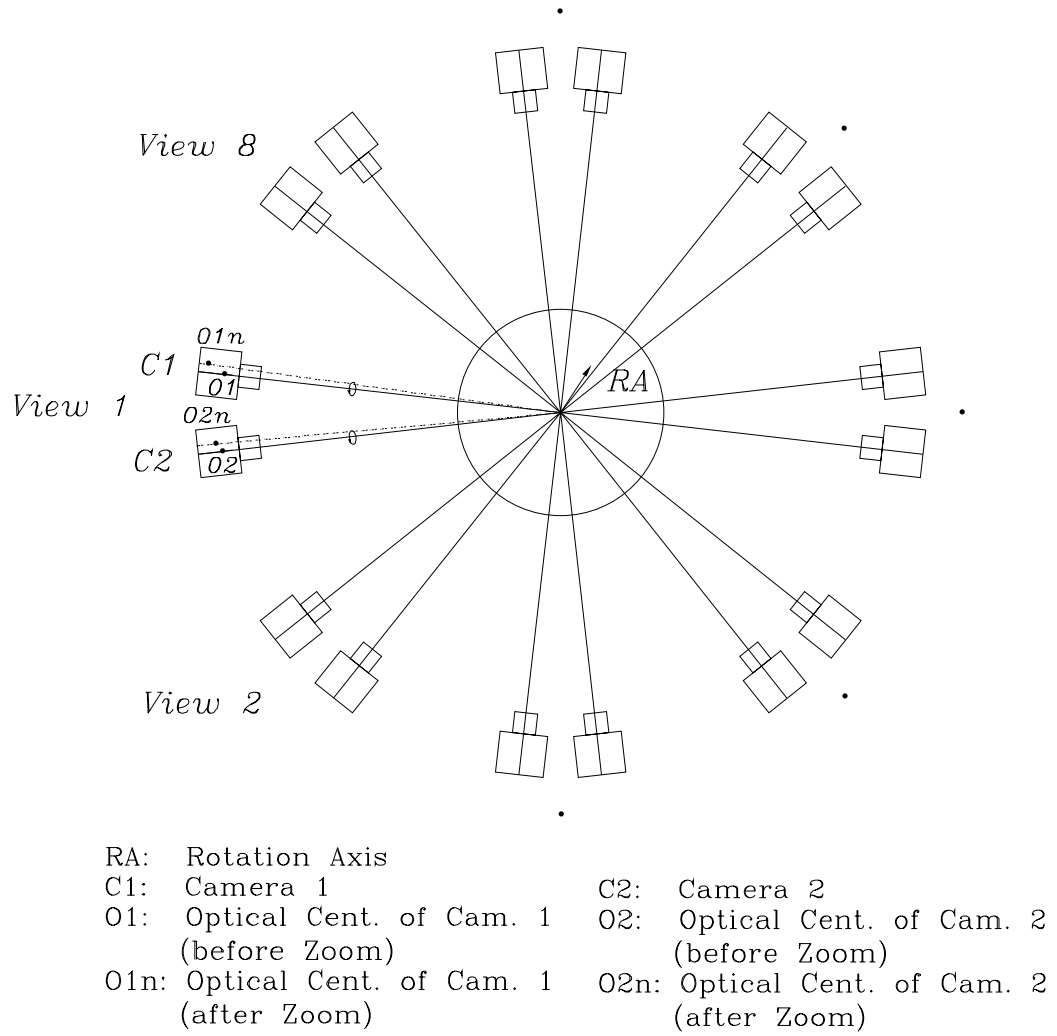


Figure 2.7: Multi-view integration and rotation axis estimation

2.6 Experimental Results

Experiments were conducted on the Stonybrook VISION System (SVIS-3). As shown in Fig. 2.8, SVIS-3 system is composed of a digital stereo camera, a rotation stage, and light sources. The digital stereo camera is made up of two vertically-mounted Olympus digital compact cameras. Two checkerboard planes are mounted perpendicularly as a calibration pattern, as displayed in Fig. 2.9. Olympus compact digital camera has 130 zoom levels that range from 65 to 195 steps. The full 3D model after multi-view calibration estimation is shown in Fig. 2.10. We see that the final full 3D models are quite good.

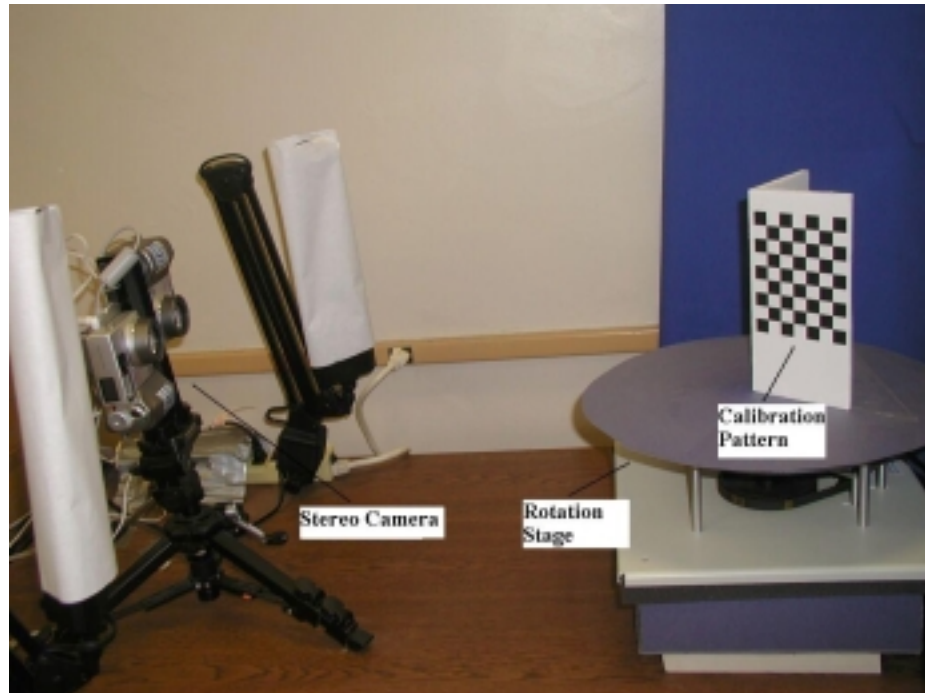
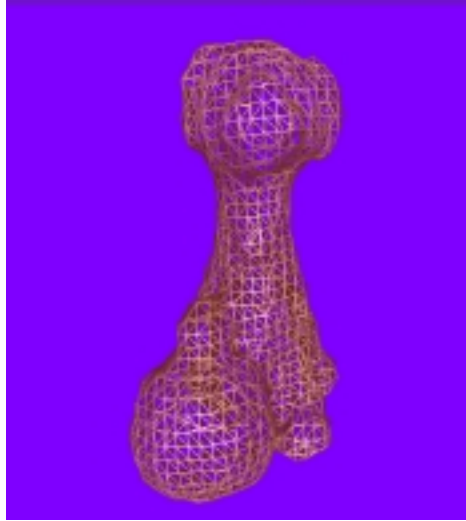


Figure 2.8: SVIS-3 3D modeling system

In this chapter, a new calibration method for estimating the internal camera parameters under dynamic zoom setting is proposed. We have also investigated a method for estimating the rotation axis of a turntable. It is used for obtaining multi-



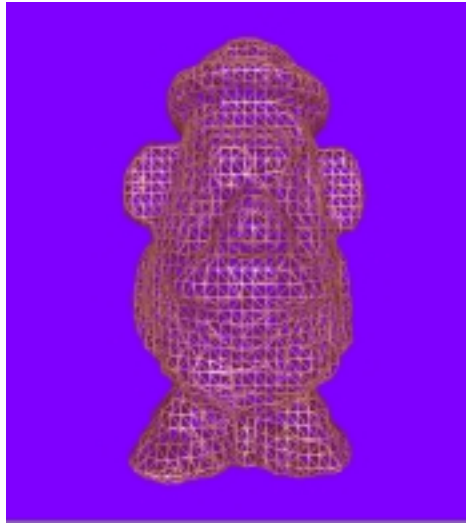
Figure 2.9: Calibration target



(a) Mesh model of dog



(b) Corresponding texture model



(c) Mesh model of potato head



(d) Corresponding texture model

Figure 2.10: 3D models by estimated rotation matrix

view stereo images for 3D modeling. A rectification refinement technique is also presented to improve the results of stereo rectification. Our methods are implemented and evaluated on an actual camera system used in 3D modeling. Experimental results show that our method is very useful for enabling stereo based 3D modeling systems to incorporate the variable zoom feature.

Chapter 3

Depth-from-Defocus Techniques

Passive techniques of ranging or determining the distance of objects from a camera is an important problem in computer vision. Depth From Focus (DFF) [21, 22] is essentially a parameter searching procedure which requires acquiring and processing many images. The search involves many mechanical motions of camera parts, thus limiting the speed of autofocusing. Depth From Defocus (DFD) is an elegant passive autofocusing method. It needs only two or three images, and recovers the depth information by computing the degree of blur.

In this chapter, new binary mask based Depth From Defocus (DFD) algorithms are proposed to improve autofocusing performance and robustness for arbitrary scenes. A binary mask is defined by thresholding image Laplacian to remove unreliable points with low Signal-to-Noise Ratios (SNR). Three different schemes – with/without spatial integration and with/without squaring – are investigated and evaluated, through both simulations and actual experiments. The actual experiments use a large variety of objects including very low contrast Ogata test charts. Experimental results show that the RMS step error for autofocusing is less than 2.6 lens steps, which corresponds to 1.73%. Although our discussion in this chapter is mainly focused on a

spatial domain method STM1, this technique should be of general value for different approaches such as STM2 and other spatial domain based algorithms.

The basic theory of STM is briefly reviewed here to introduce relevant formulas and to define the terms for future discussion.

3.1 S Transform

A Spatial-domain Convolution/Deconvolution Transform (S Transform) has been developed for images and n-dimensional signals for the case of arbitrary order polynomials [41].

If $f(x, y)$ is an image that is a two-dimensional cubic polynomial defined by:

$$f(x, y) = \sum_{m=0}^3 \sum_{n=0}^{3-m} a_{mn} x^m y^n \quad (3.1)$$

where a_{mn} are the polynomial coefficients. The restriction on the order of f is made to be valid by applying a polynomial fitting least square smoothing filter to the image.

Let $h(x, y)$ be a rotationally symmetric point spread function (PSF). In a small region of the image detector plane, the camera system acts as a linear shift invariant system. The observed image $g(x, y)$ is the convolution of the corresponding focused image $f(x, y)$ and the PSF of the optical system $h(x, y)$:

$$g(x, y) = f(x, y) \otimes h(x, y) \quad (3.2)$$

where \otimes denotes the convolution operation.

A spread parameter σ_h is used to characterize the different forms of PSF. It can be defined as the square root of the second central moment of the function h . For a rotationally symmetric function, it is given by:

$$\sigma_h^2 = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} (x^2 + y^2) h(x, y) dx dy \quad (3.3)$$

The above deconvolution formula can be written as:

$$f(x, y) = g(x, y) - \frac{\sigma_h^2}{4} \nabla^2 g(x, y) \quad (3.4)$$

For simplicity, the focused image $f(x, y)$ and defocused images $g_i(x, y), i = 1, 2$ are denoted as f and g_i from now on.

3.2 STM Autofocusing

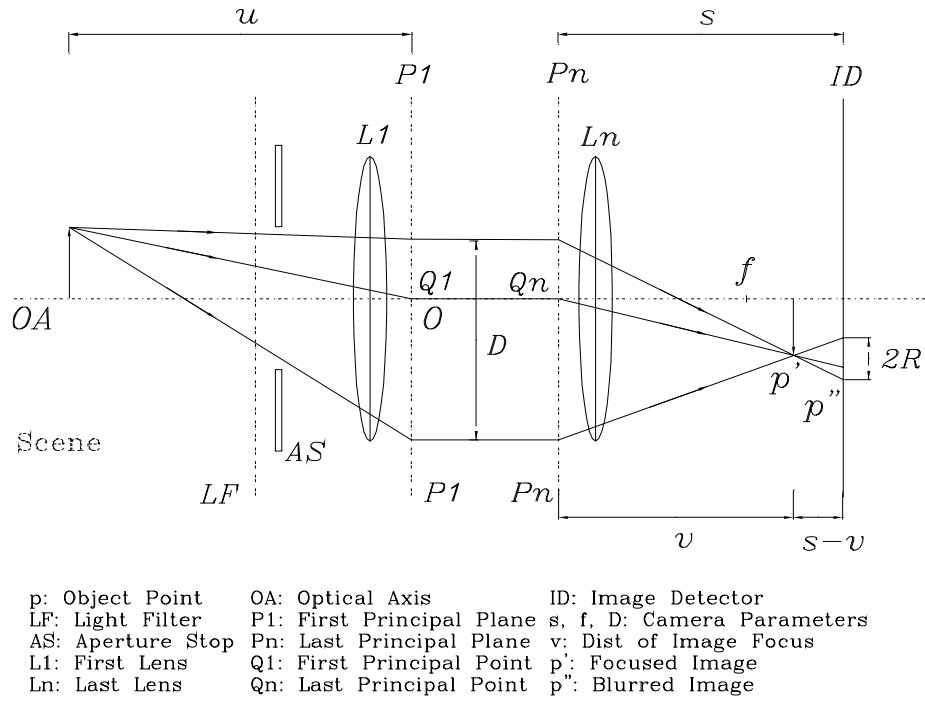


Figure 3.1: Schematic diagram of camera system

A schematic diagram of a camera system is shown in Fig. 3.1. The Aperture Stop (AS) is the element of the imaging system which physically limits the angular size of the cone of light accepted by the system. The field stop is the element that physically restricts the size of the image. The entrance pupil is the image of the aperture stop

(AS) as viewed from the object space, formed by all the optical elements preceding it. It is the effective limiting element for the angular size of the cone of light reaching the system. Similarly, the exit pupil is the image of aperture stop, formed by the optical elements following it. For a system of multiple groups of lenses, the focal length will be the effective focal length f_{eff} ; the object distance u will be measured from the first principal point (Q_1), the image distance v and the detector distance s will be calculated from the last principal point (Q_n). Imaginary planes erected perpendicular to the optical axis at these points are known as the first principal plane (P_1) and the last principal plane (P_n) respectively.

If an object point p is not focused, then a blur circle p'' is detected on the image detector plane. The radius of the blur circle can be calculated:

$$R = \frac{Ds}{2} \left[\frac{1}{f} - \frac{1}{u} - \frac{1}{s} \right] \quad (3.5)$$

where f is the effective focal length, D is the diameter of the system aperture, R is the radius of the blur circle, u , v and s are the object distance, image distance, and detector distance respectively. The sign of R here can be either positive or negative depending on whether $s \geq v$ or $s < v$.

After magnification normalization, the normalized radius of blur circle can be expressed as a function of camera parameter setting \vec{e} and object distance u as

$$R'(\vec{e}, u) = \frac{Rs_0}{s} = \frac{Ds_0}{2} \left(\frac{1}{f} - \frac{1}{u} - \frac{1}{s} \right) \quad (3.6)$$

σ is proportional to R' , and can be expressed by:

$$\sigma = k'R(\vec{e}, u) \quad (3.7)$$

Therefore from Eqn. (3.6) we have

$$\sigma = mu^{-1} + c \quad (3.8)$$

where

$$m = -\frac{kDs_0}{2} \quad \text{and} \quad c = -\frac{kDs_0}{2} \left(\frac{1}{f} - \frac{1}{s} \right) \quad (3.9)$$

Let g_1 and g_2 be the two images recorded with different parameter settings $\vec{e}_1 = (s_1, f_1, D_1)$ and $\vec{e}_2 = (s_2, f_2, D_2)$.

$$\sigma_i = m_i u^{-1} + c_i \quad i = 1, 2 \quad (3.10)$$

Rewriting Eqn.(3.10) by eliminating u^{-1} :

$$\sigma_1 = \alpha \sigma_2 + \beta \quad (3.11)$$

where

$$\alpha = \frac{m_1}{m_2} \quad \text{and} \quad \beta = c_1 - c_2 \frac{m_1}{m_2} \quad (3.12)$$

From Eqn. (3.4), for each defocused image we can obtain:

$$f = g_i - \frac{\sigma_i^2}{4} \nabla^2 g_i \quad i = 1, 2 \quad (3.13)$$

Then equating the right side of Eqn. (3.13):

$$g_1 - \frac{\sigma_1^2}{4} \nabla^2 g_1 = g_2 - \frac{\sigma_2^2}{4} \nabla^2 g_2 \quad (3.14)$$

Under the third order polynomial assumption in Eqn. (3.1), for the same object, we have:

$$g_1 - g_2 = \frac{1}{4} G \nabla^2 g \quad (3.15)$$

where $\nabla^2 g = \nabla^2 g_1 = \nabla^2 g_2$, and

$$G = \sigma_1^2 - \sigma_2^2 = \frac{4(g_1 - g_2)}{\nabla^2 g} \quad (3.16)$$

Now substituting for σ_1 in terms of σ_2 from Eqn. (3.11), and using the definition of G in Eqn. (3.16), we have:

$$\sigma_2^2 (\alpha^2 - 1) + 2\alpha\beta\sigma_2 + \beta^2 = G \quad (3.17)$$

where the definition of α and β are same as in Eqn. (3.12).

In STM1, lens position is changed during the acquisition of the two images g_1 and g_2 (STM1), but the aperture diameter remains unchanged. Therefore $\alpha = \frac{m_1}{m_2} = \frac{D_1}{D_2} = 1$, and we obtain:

$$\sigma_2 = \frac{G}{2\beta} - \frac{\beta}{2} \quad (3.18)$$

In STM2, only the diameter of camera aperture is changed in the image acquisition of two images g_1 and g_2 . In this case we have $\beta = 0$ and $\alpha = \frac{D_1}{D_2}$. Therefore Eqn. (3.17) reduces to:

$$\sigma_2 = \pm \sqrt{\frac{G}{\alpha^2 - 1}} \quad (3.19)$$

3.3 Binary Mask Based STM1 Algorithms

Because of camera noise, the original STM1 algorithm of Subbarao and Surya [32, 33] uses the steps of squaring, spatial integration, and mode selection of histogram. This was done as the cameras used in the past were of poor quality compared to the modern digital still cameras. With the development of improved digital cameras, the traditional scheme should be revisited, and simpler schemes can be implemented for DFD autofocus. These schemes improve robustness and performance of autofocus. In this chapter, we concentrate on STM1, but the same techniques can be applied to STM2.

3.3.1 SNR Mask

In previous mode selection, the histogram is built by computing σ_2 at each pixel in a 48*48 neighborhood, and then the mode of the histogram is regarded as the best estimation of σ_2 . Low contrast image regions yield low image Laplacian values. Due to camera noise and quantization, Laplacian estimates have very low SNR leading

to large error in the estimation of σ_2 . Therefore, a new binary mask is introduced to improve the robustness of the STM algorithm. The binary mask is defined by thresholding the Laplacian values. This operation removes unreliable points with low Signal-to-Noise Ratio (SNR). The binary mask is defined as:

$$M_0(x, y) = \begin{cases} 1, & |\nabla^2 g| \geq T \\ 0, & o.w. \end{cases}, \quad (x, y) \in W \quad (3.20)$$

where T is a threshold on the Laplacian which can be determined experimentally. An averaging of σ_2 based on the binary mask is used instead of the mode of its histogram.

The Binary Mask based STM1 With Squaring and With Integration (BM_WSWI) can be expressed as:

$$G = S(g_1, g_2) \frac{4}{U} \sum_{(x,y) \in W} M_0(x, y) \sqrt{\frac{\iint [g_1(x, y) - g_2(x, y)]^2 dx dy}{\iint [\nabla^2 g(x, y)]^2 dx dy}} \quad (3.21)$$

where $U = \sum \sum_{(x,y) \in W} M_0(x, y)$ is the weight of the binary mask. $S(g_1, g_2)$ is the sign function which is decided by the variance of two images $Var(g_1)$ and $Var(g_2)$.

$$S(g_1, g_2) = \begin{cases} -1, & Var(g_1) \geq Var(g_2) \\ +1, & Var(g_1) < Var(g_2) \end{cases} \quad (3.22)$$

3.3.2 Spatial Integration

Spatial integration reduces random noise at the cost of sacrificing spatial resolution; moreover, without thresholding, it may take some unreliable points into account. However, integration over an image region reduces the spatial resolution of depth-map.

To understand the effect of spatial integration, a variation of Binary Mask based STM1 With Squaring and Without Integration (BM_WSOI), which does not integrate over a small region is calculated by:

$$G = S(g_1, g_2) \frac{4}{U} \sum_{(x,y) \in W} M_0(x, y) \sqrt{\frac{[g_1(x, y) - g_2(x, y)]^2}{[\nabla^2 g(x, y)]^2}} \quad (3.23)$$

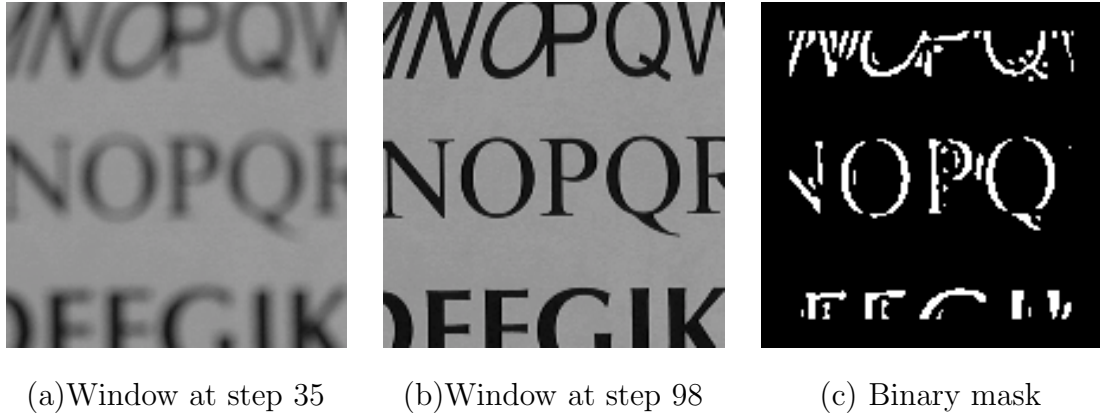


Figure 3.2: Binary mask formation

3.3.3 Squaring Scheme

The squaring scheme was introduced in the original STM1 algorithm to reduce the effects of camera noise. Squaring permitted integration over an image region without canceling positive and negative values of image Laplacians during summation. However, squaring also loses the sign information. Therefore the sign function $S(g_1, g_2)$ in Eqn. (3.22) is used.

Another variation directly uses Eqn. (3.14). It is carried out without squaring and without integration. In the Binary Mask based STM1 of Without Squaring Without Integration (BM_LSOI), the average of G is calculated based on the binary mask:

$$G = \frac{4}{U} \sum_{(x,y) \in W} M_0(x,y) \frac{[g_1(x,y) - g_2(x,y)]}{\nabla^2 g(x,y)} \quad (3.24)$$

The above algorithms are evaluated on both synthetic and real data.

3.3.4 Simulation Result

In practical experiments, there are many factors that are coupled together even in a single recorded image, such as lens aberration, vignetting, nonlinear sensor response, automatic gain control, and automatic white balance. In order to verify the theory

itself and evaluate the different variations under the same conditions, a computer simulation system Image Defocus Simulator 1 (IDS1) is implemented for generating a series of test images. Due to advances in VLSI technology, digital still cameras have improved in imaging capabilities compared to the video camera-image grabber architecture of the past. The original IDS of Lu [49, 50] has been simplified and updated to model a modern digital still camera.

A database of simulated images has been built for experiments. Images for synthetic database are displayed in Fig. 3.3. Fig. 3.4 shows 9 images of sample image "Boat" arranged in 3 rows and 3 columns. The distance of the object increased from 250 mm to 950 mm row-wise. The distance s between the lens and the image detector increases column-wise from left to right. The effective focal length and the F-Number are fixed at 19.5 mm and 2.8 respectively. In Fig. 3.4, the images are focused somewhere along the top-left to bottom-right diagonal direction. The image focus decreases on either side of the diagonal direction. This is consistent with the fact that image blur should increase when either the object is moved farther or closer from its focused position, or when the image detector is moved farther or closer from its focused position.

To compare the performance of BM_WSWI, BM_WSOI and BM_OSOI, The focal length of the camera is 19.5 mm, and the F number is set to 2.8. For each algorithm, a focusing window whose size is 144×144 is placed at the center of the test image. The size of the Gaussian filter and LoG filter are both 15×15 pixels. The sigma table corresponding BM_WSWI, BM_WSOI and BM_OSOI are shown in Fig. 3.5 (a), (b) and (c) respectively.

The rms step errors of the three variations are compared in Fig. 3.5 (d). Comparing (a) and (b), the scheme using spatial integration (BM_WSWI) has higher RMS step error than the scheme without spatial integration (BM_WSOI) in the range from 270 mm to 2200 mm. The RMS step error of BM_WSOI increases dramatically at

the far field and near field.

From Fig. 3.5 (b) and (c), variations BM_WSOI and BM_OSOI behave similar in RMS step error although some large step errors happen in some situations for BM_WSOI.

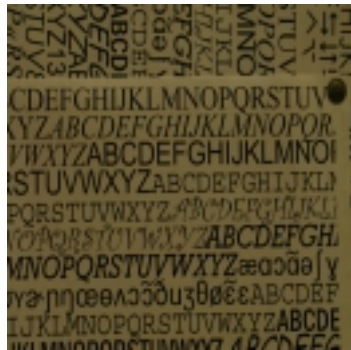
Another observation from Fig. 3.5 is that the RMS step error can be limited to be low at long range by suitably selecting the step interval for image capture or by using a third image. These will be discussed in a later section.

3.3.5 Experimental Result

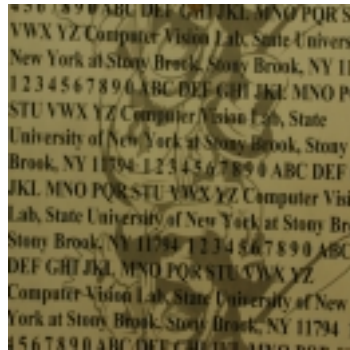
Binary Mask STM algorithms described above are implemented on a compact digital camera. The camera is controlled by a host computer (Pentium 4 2.4GHz) from a USB port. The lens' focus motor of the digital camera ranges in position from step 0 to step 150; Step 0 corresponds to focusing a nearby object at a distance of about 250 mm from the lens and Step 150 corresponds to focusing an object at a distance of infinity.

The lens designs in current digital cameras have several focusing modes such as Macro mode and Standard mode to improve the autofocus performance for different distance range. The relative position of lens elements changes when the focusing mode switches from one to another. The relationship between the lens step number and the reciprocal of the best-focused distance is no longer linear, and the practical object distance vs. focus step curve needs to be measured using Depth From Focus (DFF) technique. A double three step DFF algorithm is used to avoid local maxima in the best focus measure searching procedure [22]. The results are shown in Fig. 3.6. There exist two roughly linear segments, the first for Macro mode and the second for Standard mode. The transition area is around 787 mm (31").

To generate the sigma-step lookup tables for different variations, the defocused



(a) Letter



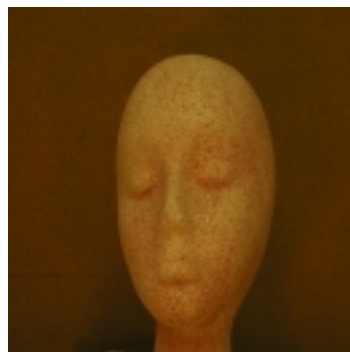
(b) DrawLetter



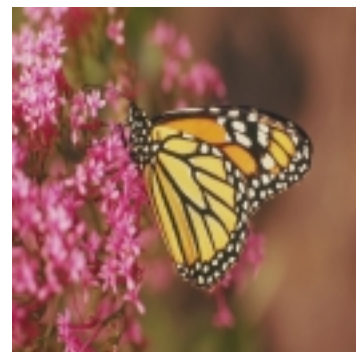
(c) CD rom



(d) Vacuum



(e) Head



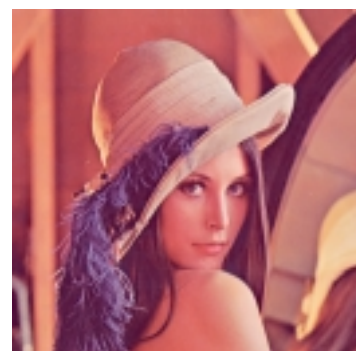
(f) Monarch



(g) Peppers



(h) Boat

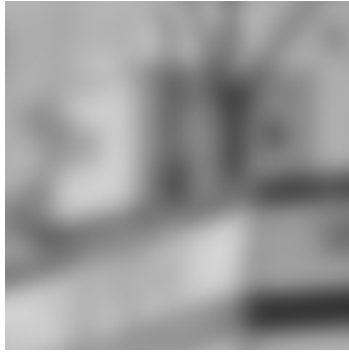


(i) Lena

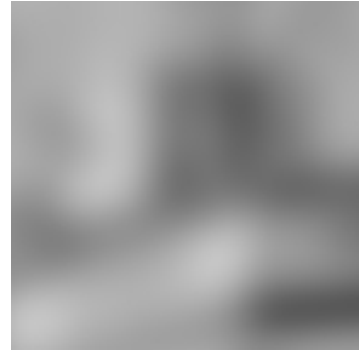
Figure 3.3: Image database for IDS1



(a) $u=250$ mm, step= 40



(b) $u=250$ mm, step= 80



(c) $u=250$ mm, step= 120



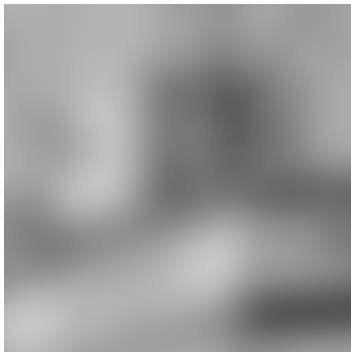
(d) $u=450$ mm, step= 40



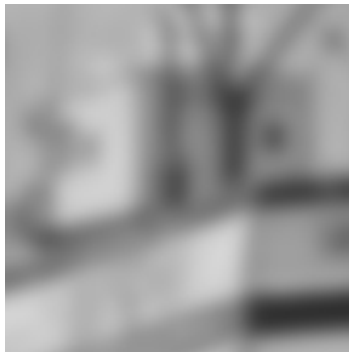
(e) $u=450$ mm, step= 80



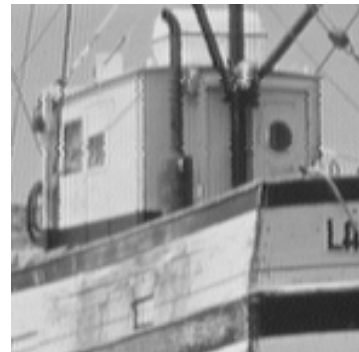
(f) $u=450$ mm, step= 120



(g) $u=950$ mm, step= 40

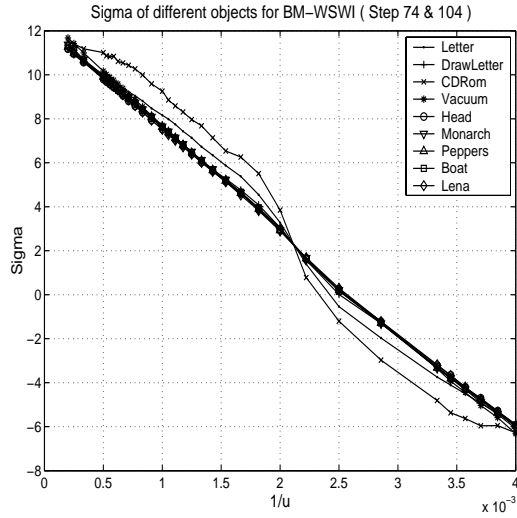


(h) $u=950$ mm, step= 80

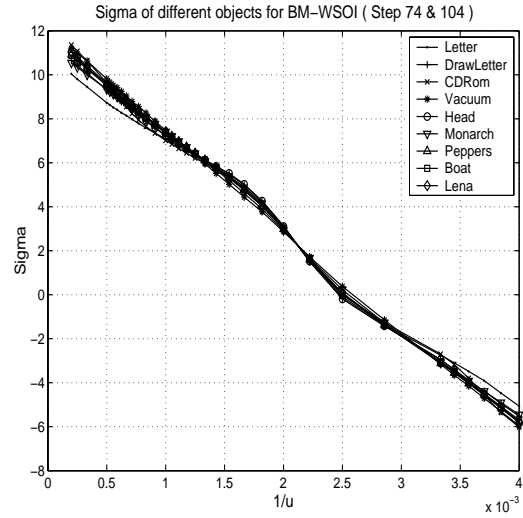


(i) $u=950$ mm, step= 120

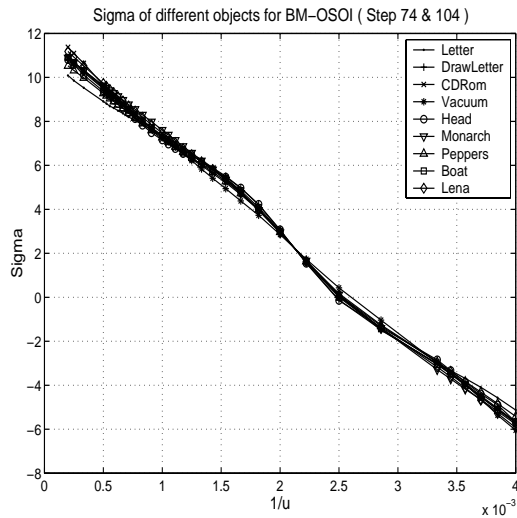
Figure 3.4: Sample output of IDS1 simulation system



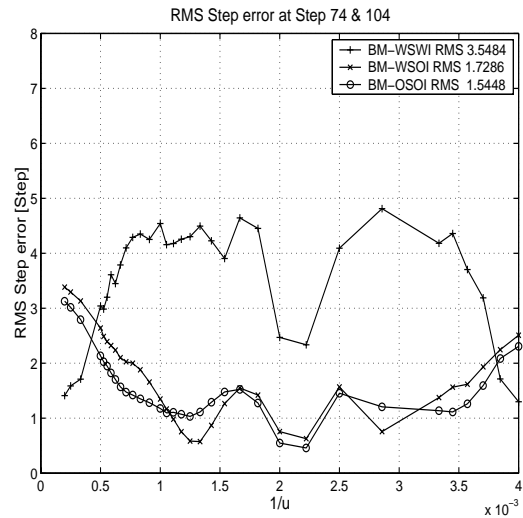
(a) BM_WSWI



(b) BM_WSOI



(c) BM_OSOI



(d) RMS step error for (a), (b), (c)

Figure 3.5: Sigma table and RMS step error of different STM1 algorithms (synthetic data)

images of a calibration object are acquired at 20 different distances. At each distance, two defocused images are obtained with the focus step numbers 35, and 98. Then σ_2 is estimated by the algorithms - BM_WSWI, BM_WSOI and BM_OSOI. The results are plotted in Fig. 3.7. There are common “flat” areas in the sigma-step curves that approximately lie in the range from step 60 to step 110 for the compact digital camera. In this area, we find that a small variation in sigma may cause a large fluctuation in the focusing step. However the corresponding change in focusing measure or image sharpness is not significant. This means that if we use focus step number as error metric, large errors are expected in the step 60 to 110 interval. This is misleading since the error in image sharpness will be small. Therefore, focus step is not the best error metric as it overstates the error, but focus measure difference is a better measure of performance as it corresponds to image sharpness in autofocus. However, to be conservative, we use the metric of lens step number with a hopeful note that the actual defocus error will be lower.

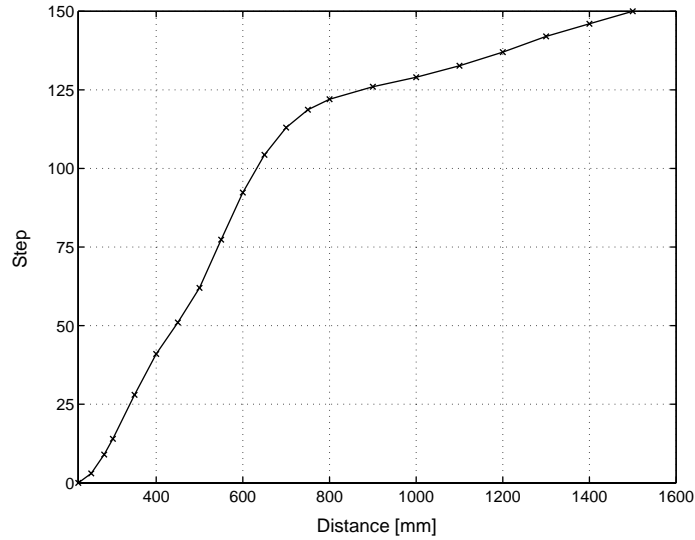


Figure 3.6: Step number vs. object distance

To bring out the capability of DFD algorithms, experiments are performed on

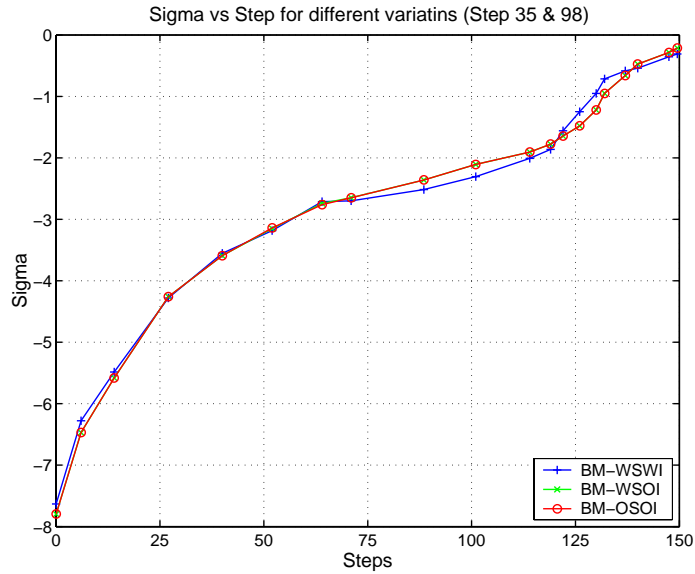


Figure 3.7: Sigma vs. step number

eight objects that are relatively difficult to measure, as shown in Fig. 3.8. Eight positions are randomly selected. The distance and the corresponding steps are listed in Table 3.1. One of the test objects at those positions are shown in Fig. 3.9. The F-number is set to 2.8, and the focal length is set to 19.5 mm. A focusing window is located at the center of the scene. Its window size is 96×96 . The Gaussian smoothing filter and LoG filter are 9×9 pixels. The sensor nonlinear response compensation [51] is utilized.

Measurement results and rms error for (a) BM_WSWI (b) BM_WSOI, and (c) BM_OSOI are plotted in Fig. 3.10 and Fig. 3.11 respectively. Comparing the schemes, With Spatial Integration (BM_WSWI) perform better than the schemes WithOut Spatial Integration (BM_WSOI and BM_OSOI) at far field positions (position 8), but sometimes they may give large errors due to unreliable points. The schemes without squaring perform better than schemes with squaring at some positions (position 2). This can also be observed from simulation results in the previous sections. The mean

of RMS step error of simulation results and experimental results are summarized in Table 3.2.

Errors at position 8 could be large as the first image processed in DFD will be highly blurred (first image is captured at a lens position of 35 whereas the focused step position is around 145). In this case, a third image closer to 145 should be recorded and processed for better accuracy. Taking all factors into account such as accuracy, computational requirements, simplicity of algorithm, resolution of depth-map, etc., we suggest BM_OSIOI for use in practical applications. Even when very low contrast objects such as those in Fig. 3.8 (g) and (h) are present, an RMS error of about 3 steps can be expected which gives very sharp focused images in autofocusing applications.

In this section, a new binary mask is defined based on thresholding of image Laplacian to remove unreliable points with low Signal-to-Noise Ratio (SNR) in DFD applications. This mask is exploited in different DFD schemes such as with/without spatial integration and with/without squaring, and their performances are investigated and evaluated both with simulation and actual experiments. Experimental results show that the autofocusing RMS step error is roughly similar for the different schemes. However, taking several factors such as accuracy and computational resources into account suggests that the DFD scheme of without squaring and without spatial integration (BM_OSIOI) is best suited for practical applications. While this paper deals with STM1, the conclusions here should be applicable to other spatial domain DFD methods such as STM2.

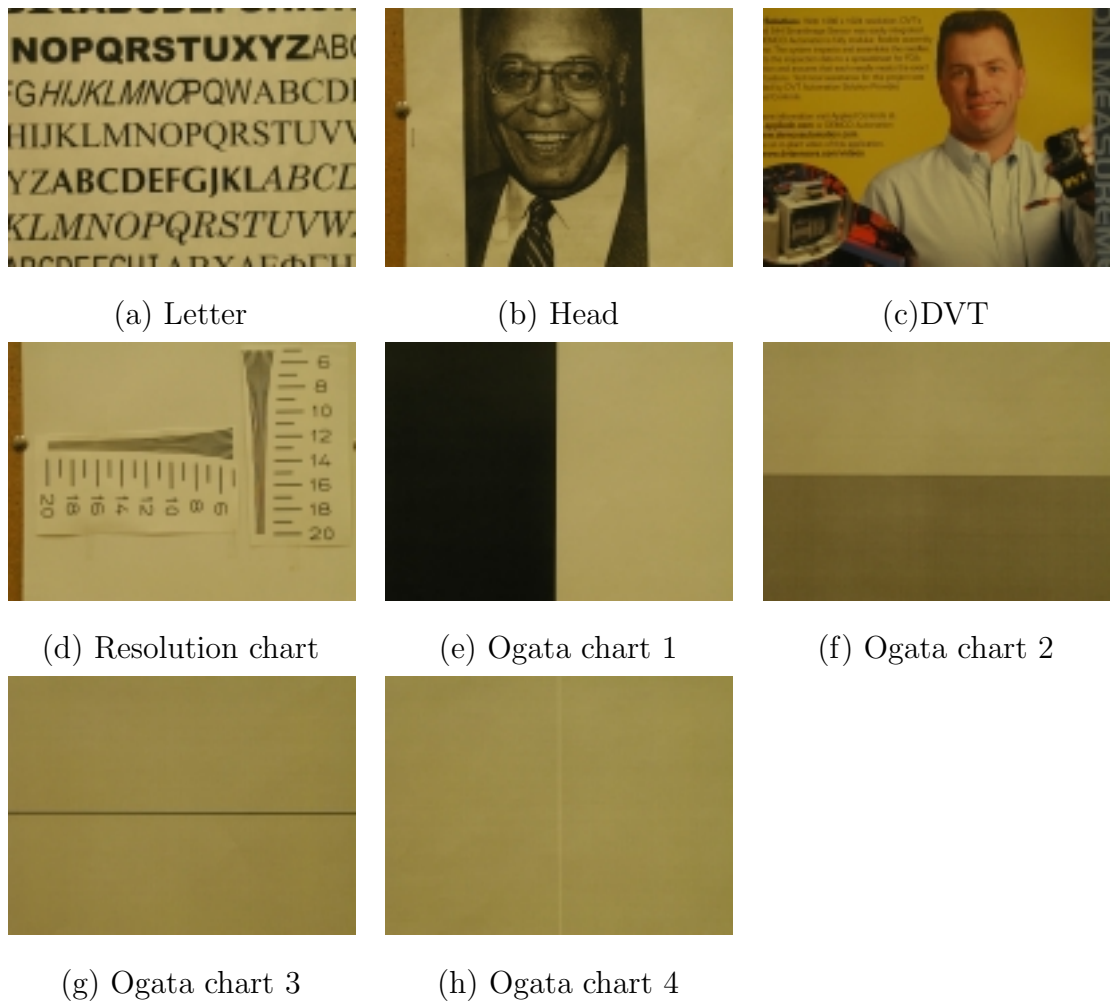
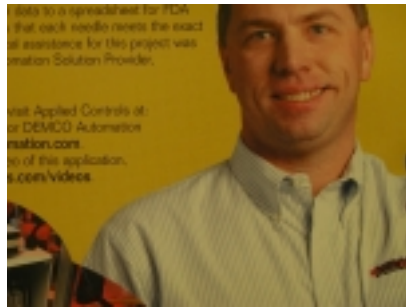


Figure 3.8: Test objects for DFD evaluation

Position	1	2	3	4
Dist [mm]	325.2	473.5	626.1	782.8
Step	19.00	55.00	96.50	120.50
Position	5	6	7	8
Dist [mm]	913.1	1055.3	1232.6	1350.7
Step	126.00	131.25	139.00	144.75

Table 3.1: Object positions in DFD experiments



(a) Position 1



(b) Position 2



(c) Position 3



(d) Position 4



(e) Position 5



(f) Position 6

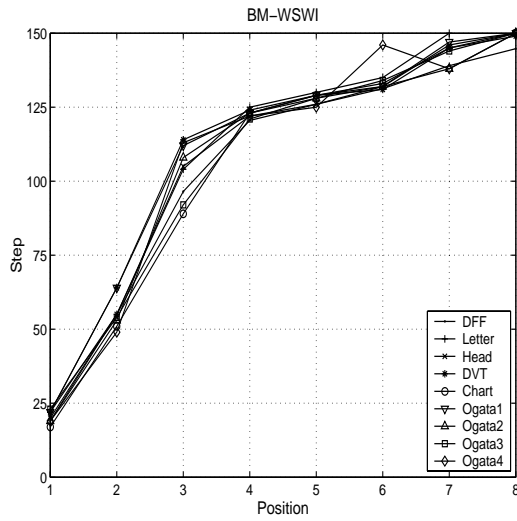


(g) Position 7

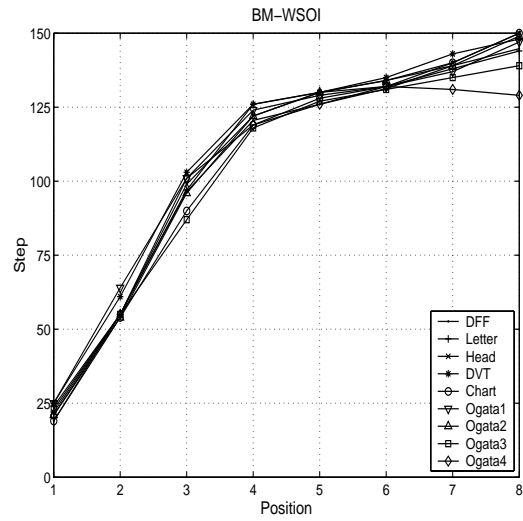


(h) Position 8

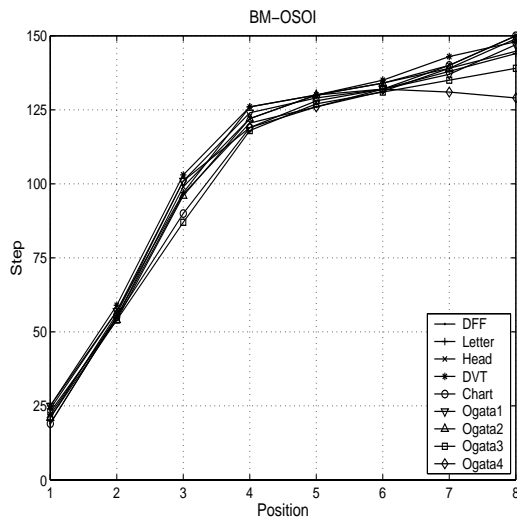
Figure 3.9: Test object at different positions



(a) BM_WSWI



(b) BM_WSOI



(c) BM_OSOI

Figure 3.10: Measurement results for different STM1 algorithms (real data)

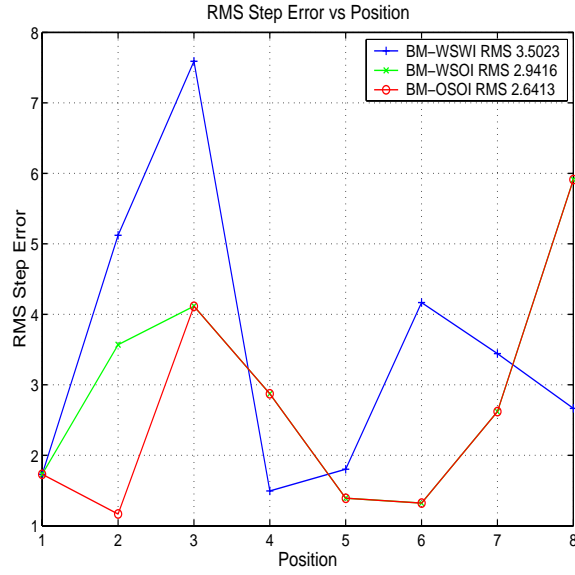


Figure 3.11: RMS step error for different STM1 algorithms (real data)

Algorithm	Mean RMS Step Error			
	Simulation Results		Experimental Results	
	Step	%	Step	%
BM_WSWI	3.55	2.37	3.50	2.33
BM_WSOI	1.73	1.15	2.94	1.96
BM_OSOI	1.54	1.03	2.64	1.76

Table 3.2: DFD performance summary

3.4 Measuring Range Extension

3.4.1 Optimal Step Interval

Direct approach to extend the measuring range, as mentioned in the previous section, is to find the optimal defocus positions for image acquisition to minimize the full range step error without adding a third image. From Eqn. (3.16), the essentially STM1 based DFD is a differentiation algorithm. If the two positions of image acquisition are too close, then the images will be too similar, which means the difference signal $g_1 - g_2$ is close to noise level. As a result, the error in lens step measurement could be large. If the two positions are very far apart, then one of the image may be blurred too much. This reduces the contrast too much and again the lens step error will be large. The essence of optimal interval problem is to find the best tradeoff under these constraints.

In this section, Binary Mask based STM1 Without Squaring Without Integration (BM_OSIOI) is selected as an example for different step interval, however the other algorithms are also suitable for the step optimization.

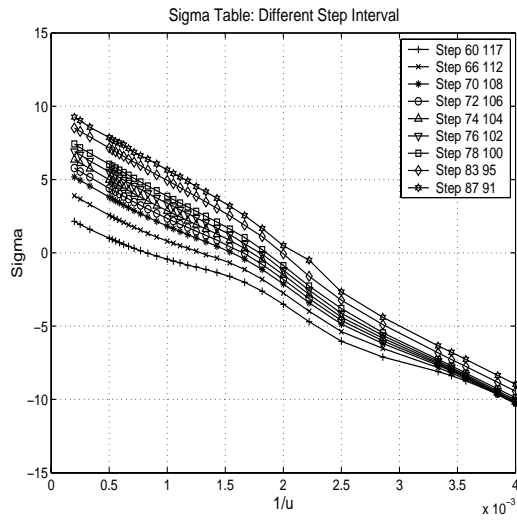
Due to the complexity of theoretical analysis, a simulation for ideal imaging system is conducted on the synthesized database obtained using the IDS1 simulation system. In the $1/u$, two positions that trisect the $1/u$ range can be used as an initial estimate of optimal positions for image acquisition. These are steps 76 and 102. A series of trials by increasing or decreasing the step interval are carried out. The results are shown in Fig. 3.12. Sigma table for different step interval are plotted in (a). The slope of the sigma curve decreases with increase in the step interval. The change of RMS step error with multiple step intervals are shown in (b). An overall measure – mean of RMS step error – is shown in (c). We see that the RMS step error tends to decrease first, after reaching the minimum RMS step error of 1.3 steps at interval 76

and 102, and later increases, as the step interval reduces from 37 steps to 26 steps and 26 steps to 4 steps. We can also fix one lens step at 76, then change another lens step from 83 to 117, a similar optimization curve can be observed, as shown in Fig. 3.13.

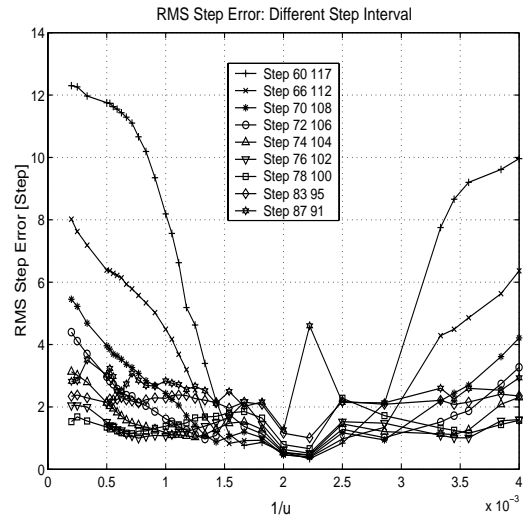
In actual experiments with real data, the same test objects are used. Step 35 is selected as the base defocused step, and step 45, 55, 65, 88, 98, 108 and 118 are selected as other defocused steps respectively. The RMS step error for different objects at six positions are plotted in Fig. 3.14(a). An overall measure – mean of RMS step error – is shown in Fig. 3.14(b). When the defocused image sensor planes are too close, the RMS step error is relatively high. The mean RMS step error is 18.1 steps for defocused steps 35 and 45. If the step interval increases, the mean RMS step error drops to 2.6 step for the defocused step pair 35 and 98. When the step interval is increased further, the mean RMS step error increases to 3.3 steps.

3.4.2 Three-Image STM Scheme

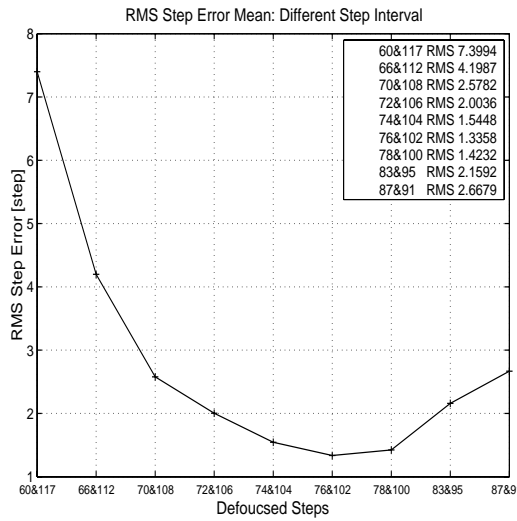
In previous sections, various STM algorithms were discussed. The RMS step error distribution is not uniform in the measurement range in both simulation and actual experiments. Some algorithms, as shown in Fig. 3.5(d) and Fig. 3.11, suffer from the increase of RMS step error at the near field and the far field. The distribution and magnitude of RMS step error vary. The scheme with spatial integration BM_WSWI results in a relatively flat RMS step error distribution at the cost of poor performance for high-contrast objects. Thus a relatively high step error may occur when a high-contrast object is involved. For 3D shape measurement application, the error could be controlled by placing the object to be measured in the area with low RMS step error. In autofocusing application, a flat step error distribution is desirable for the whole distance range.



(a) Sigma table

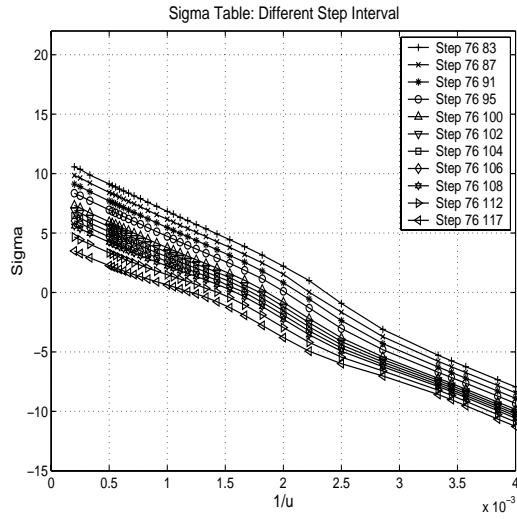


(b) RMS step error

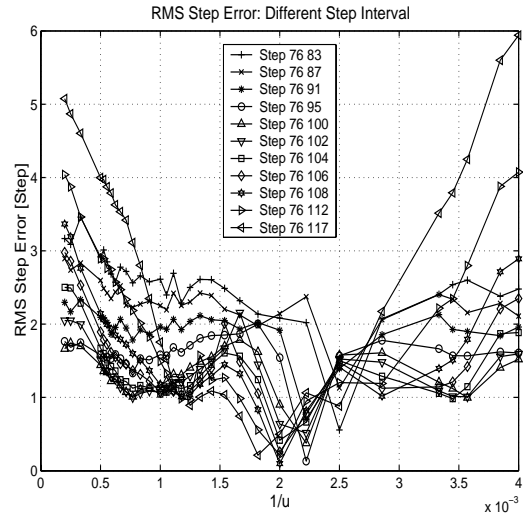


(c) Mean RMS step error

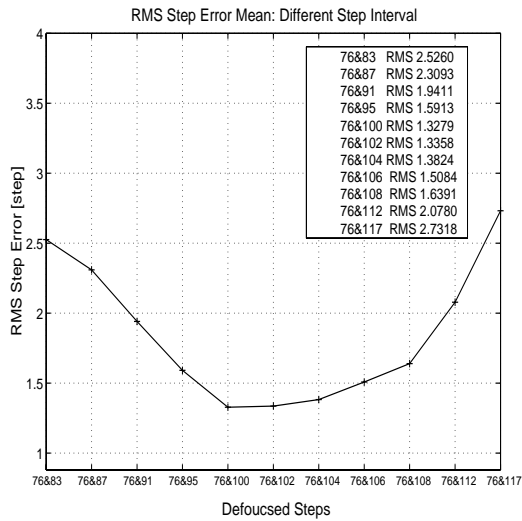
Figure 3.12: Comparisons at different intervals (synthetic data) (1)



(a) Sigma table

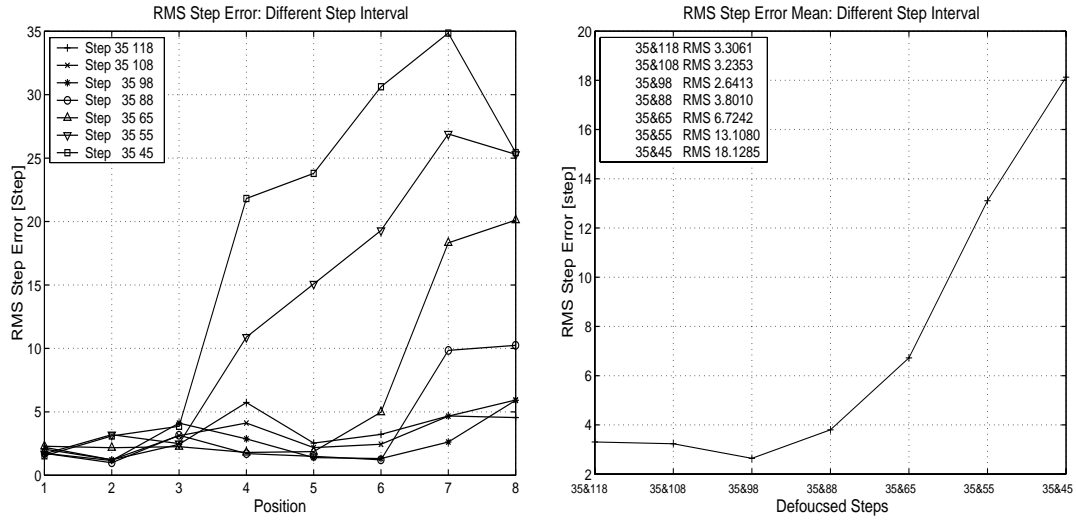


(b) RMS step error



(c) Mean RMS step error

Figure 3.13: Comparisons at different intervals (synthetic data) (2)



(a) RMS step error

(b) Mean RMS step error

Figure 3.14: Comparisons at different intervals (real data)

In this section, a variation with three-image based STM1 is developed to extend the measurement range without sacrificing accuracy. To evaluate the performance, Binary Mask based STM1 without Squaring and without Integration (BM_OS0I) is extended from two image version to three image version. Other algorithms can also be adapted. In this three-image based STM1, two images $g_i(x, y)$, $i = 1, 2$ are taken by moving the sensor plane to position i . When the measured distance is far from the camera, a third image $g_3(x, y)$ is taken after moving the lens to the step number estimated using g_1 and g_2 . Then the image pair g_2 and g_3 are used in the estimation process.

In computer simulation, the step interval for three-image scheme is taken to be the same as the one in two-image scheme (30 steps). The steps of image sensor plane are 59, 89 and 119 for the three images. The RMS step error is calculated based on the same synthesized image database shown in Fig. 3.3. Fig. 3.15 shows the step error of three-image based STM1, the counterpart of two-image based scheme in Fig.3.5(d)

is also plotted with it for comparison. The mean RMS step error dropped from 1.32 steps to 1.09 in the simulation.

In the real experiments, step 30, 65, 118 are selected as the steps for image acquisition. The test objects are same as in Fig. 3.8. The RMS step error for real experiment is plotted in Fig. 3.16. Compared with the previous two image based schemes that use step numbers 35 and 98, the RMS errors are reduced at positions 3, 4 and 8, the mean RMS dropped from 2.64 to 2.19 lens step.

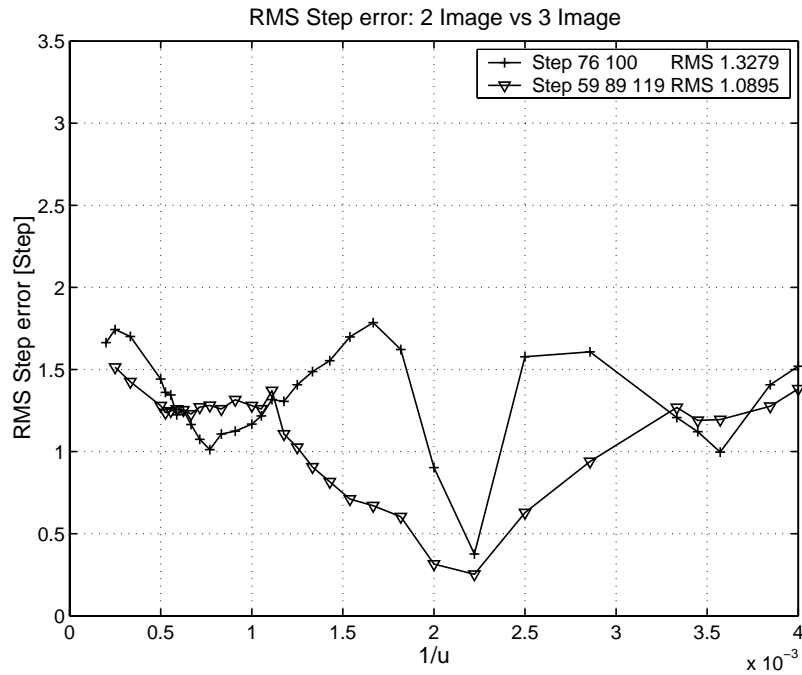


Figure 3.15: RMS step error for two-image and three-image based schemes (synthetic data)

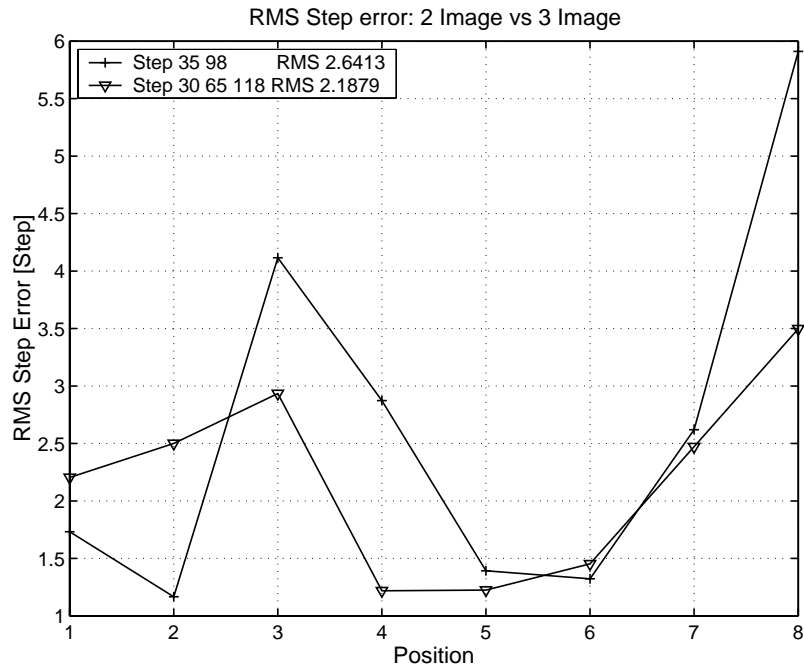


Figure 3.16: RMS step error for two-image and three-image based DFD (real data)

3.5 Differentiation Filters

In Eqn. (3.1), an assumption of a local cubic polynomial model is used for the focused image. This assumption can be removed by using a smoothing and differentiation filter. In this section we compare three filters that can be used for smoothing and differentiation. They are tested on a circularly symmetric cylindrical function. In the results, the Laplacian of the Gaussian gives the most circularly symmetric results and therefore it is selected for use in STM. The other two filters give non-symmetric results. Therefore, they are not selected for use. The reason for non-symmetry may be due to the separable nature of the filters.

3.5.1 Chebyshev Polynomial Filter

Meer and Weiss [52] have proposed a set of discrete image smoothing filters for estimating images and their derivatives. The polynomial fitting is implicit and it is done by least-square error minimization. The filters are based on Chebyshev polynomials.

$$L_0(n) = -\frac{3[5n^2 - (3N^2 + 3N - 1)]}{(2N - 1)(2N + 1)(2N + 3)} \quad (3.25)$$

where the support of the filter is $n = -N, -(N - 1), \dots, -1, 0, 1, \dots, N - 1, N$

The filter for estimating the second order image derivatives is:

$$L_2(n) = -\frac{30[3n^2 - N(N + 1)]}{N(N + 1)(2N - 1)(2N + 1)(2N + 3)} \quad (3.26)$$

The 2D Laplacian is calculated by:

$$\nabla^2 f(x, y) = f(x, y) \otimes L_2(x) + f(x, y) \otimes L_2(y) \quad (3.27)$$

where \otimes is one dimensional convolution and $L_2(y) = [L_2(x)]^t$

3.5.2 2D Savitzky-Golay Filter

This time-domain method of smoothing is also based on least-square polynomial fitting across a moving window within the data. The method was originally designed to preserve the higher moments within time-domain spectral peak data.

Consider a 7×7 patch in an image, the data can be expressed as Table 3.3. $f(i)$ is the pixel value, and the column vector m represents all the image data:

$$f = [f(0) \quad f(1) \quad f(2) \quad \dots \quad f(48)]^t \quad (3.28)$$

Let a third-order two-dimensional polynomial be fitted to this array:

$$\begin{aligned} f(i) &\approx f(x_i, y_j) \\ &= a_{00} + a_{10}x_i + a_{01}y_j + a_{20}x_i^2 + a_{11}x_i y_j + a_{02}y_j^2 \\ &\quad + a_{30}x_i^3 + a_{21}x_i^2 y_j + a_{12}x_i y_j^2 + a_{03}y_j^3 \end{aligned} \quad (3.29)$$

		x_i						
		-3	-2	-1	0	-1	-2	-3
y_i	-3	$f(0)$	$f(1)$	$f(2)$	$f(3)$	$f(4)$	$f(5)$	$f(6)$
	-2	$f(7)$	$f(8)$	$f(9)$	$f(10)$	$f(11)$	$f(12)$	$f(13)$
	-1	$f(14)$	$f(15)$	$f(16)$	$f(17)$	$f(18)$	$f(19)$	$f(20)$
	0	$f(21)$	$f(22)$	$f(23)$	$f(24)$	$f(25)$	$f(26)$	$f(27)$
	1	$f(28)$	$f(29)$	$f(30)$	$f(31)$	$f(32)$	$f(33)$	$f(34)$
	2	$f(35)$	$f(36)$	$f(37)$	$f(38)$	$f(39)$	$f(40)$	$f(41)$
	3	$f(42)$	$f(43)$	$f(44)$	$f(45)$	$f(46)$	$f(47)$	$f(48)$

Table 3.3: Savitzky-Golay filter

where (x_i, y_j) is the pixel coordinate of $f(i)$, and the coefficient of $x_i^m y_j^n$ is a_{mn} .

To compute the coefficients from the data we set up a matrix equation:

$$Xa = f \quad (3.30)$$

where

$$X = \begin{bmatrix} 1 & x_0 & y_0 & x_0^2 & x_0 y_0 & y_0^2 & x_0^3 & x_0^2 y_0 & x_0 y_0^2 & y_0^3 \\ 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 & x_1^3 & x_1^2 y_1 & x_1 y_1^2 & y_1^3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{48} & y_{48} & x_{48}^2 & x_{48} y_{48} & y_{48}^2 & x_{48}^3 & x_{48}^2 y_{48} & x_{48} y_{48}^2 & y_{48}^3 \end{bmatrix} \quad (3.31)$$

and a is the vector of polynomial coefficients:

$$a = \left[a_{00} \ a_{10} \ a_{01} \ a_{20} \ a_{11} \ a_{02} \ a_{30} \ a_{21} \ a_{12} \ a_{03} \right]^t \quad (3.32)$$

Eqn. (3.30) simply reproduces the polynomial for each pixel in the image patch.

We solve for the polynomial coefficients using least-squares:

$$a = Cf \quad (3.33)$$

where

$$C = (X^t X)^{-1} X^t \quad (3.34)$$

C is independent of the image. Each polynomial coefficient can be calculated as the inner product of one row of C and the column of pixel values f , then the polynomial coefficients are computed using a linear filter on the data. Just as one can reassemble m back into a rectangular patch of pixels, one can also assemble each row of C into the same size rectangle to get a traditional-looking image filter.

The three filters for smoothing and differentiation are listed in Eqn. (3.34), (3.35) and (3.36).

$$C_{00} = \begin{bmatrix} -0.0476 & -0.0136 & 0.0068 & 0.0136 & 0.0068 & -0.0136 & -0.0476 \\ -0.0136 & 0.0204 & 0.0408 & 0.0476 & 0.0408 & 0.0204 & -0.0136 \\ 0.0068 & 0.0408 & 0.0612 & 0.0680 & 0.0612 & 0.0408 & 0.0068 \\ 0.0136 & 0.0476 & 0.0680 & 0.0748 & 0.0680 & 0.0476 & 0.0136 \\ 0.0068 & 0.0408 & 0.0612 & 0.0680 & 0.0612 & 0.0408 & 0.0068 \\ -0.0136 & 0.0204 & 0.0408 & 0.0476 & 0.0408 & 0.0204 & -0.0136 \\ -0.0476 & -0.0136 & 0.0068 & 0.0136 & 0.0068 & -0.0136 & -0.0476 \end{bmatrix} \quad (3.35)$$

$$C_{20} = \begin{bmatrix} 0.0085 & 0.0085 & 0.0085 & 0.0085 & 0.0085 & 0.0085 & 0.0085 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 \\ -0.0051 & -0.0051 & -0.0051 & -0.0051 & -0.0051 & -0.0051 & -0.0051 \\ -0.0068 & -0.0068 & -0.0068 & -0.0068 & -0.0068 & -0.0068 & -0.0068 \\ -0.0051 & -0.0051 & -0.0051 & -0.0051 & -0.0051 & -0.0051 & -0.0051 \\ -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 \\ 0.0085 & 0.0085 & 0.0085 & 0.0085 & 0.0085 & 0.0085 & 0.0085 \end{bmatrix} \quad (3.36)$$

$$C_{02} = \begin{bmatrix} 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \\ 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \\ 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \\ 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \\ 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \\ 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \\ 0.0085 & 0.0000 & -0.0051 & -0.0068 & -0.0051 & 0.0000 & 0.0085 \end{bmatrix} \quad (3.37)$$

Smoothing an image is carried out as follows. A two-dimensional polynomial is conceptually fitted to the image patch surrounding each pixel, and then this polynomial is evaluated at the pixel. The local coordinate system that we use for the image patch has $(x, y) = (0, 0)$ in the middle of the patch. Thus the smoothed value of the pixel turns out to be merely a_{00} , which we can compute by applying the filter C_{00} to the image patch.

To compute Laplacian derivatives of the patch, the two partial derivatives of the fitted polynomial are

$$\frac{\partial^2 f(x_i, y_j)}{\partial x^2} = 2a_{20} + 6a_{30}x_i + 2a_{21}y_j \quad (3.38)$$

$$\frac{\partial^2 f(x_i, y_j)}{\partial y^2} = 2a_{02} + 2a_{12}x_i + 6a_{03}y_j \quad (3.39)$$

Evaluating at $(x, y) = (0, 0)$ the results are simply $\frac{\partial^2 f(0,0)}{\partial x^2} = 2a_{20}$ and $\frac{\partial^2 f(0,0)}{\partial y^2} = 2a_{02}$, which are computed with filters C_{20} and C_{02} above.

$$\nabla^2 f(x, y) = 2f(x, y) \otimes C_{20} + 2f(x, y) \otimes C_{02} \quad (3.40)$$

3.5.3 2D Laplacian of Gaussian Filter

Since convolution operation is associative, we can convolve the Gaussian smoothing filter with the Laplacian filter first, and then convolve this hybrid filter with the

image to achieve the required result. Doing things this way has two advantages. Since both the Gaussian and the Laplacian kernels are usually much smaller than the image, this method usually requires far fewer arithmetic operations. The Laplacian of Gaussian (LoG) kernel can be pre-calculated in advance so only one convolution needs to be performed at run-time on the image.

The 2D LoG function centered on zero and with Gaussian standard deviation σ has the form:

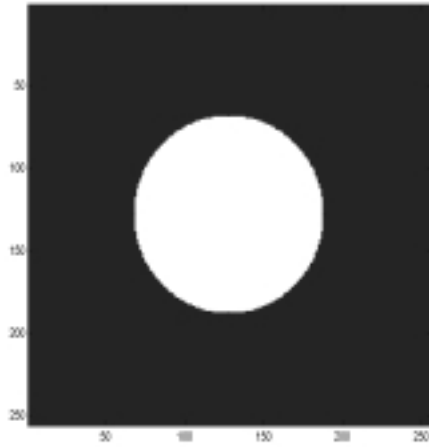
$$L_2(x, y) = \frac{\partial^2 L_0}{\partial x^2} + \frac{\partial^2 L_0}{\partial y^2} = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (3.41)$$

where $L_0(x, y)$ is a Gaussian smoothing filter:

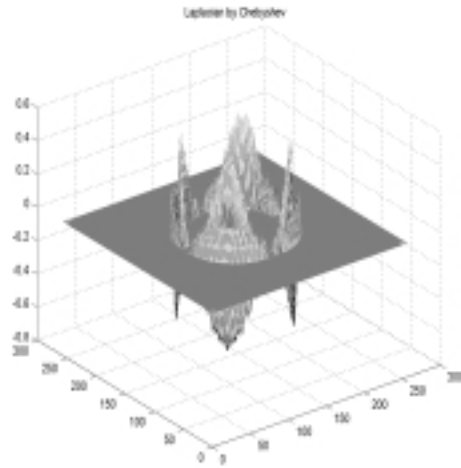
$$L_0(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (3.42)$$

3.5.4 Discussion of Results

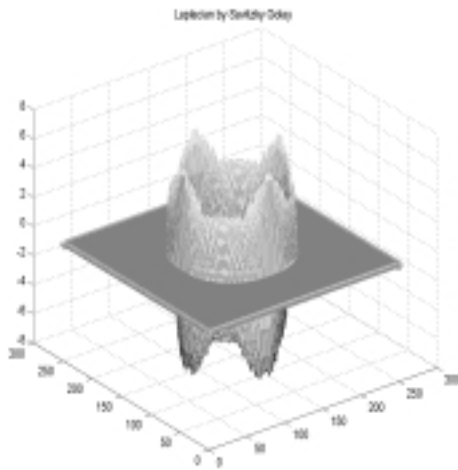
In the depth estimation for arbitrary scenes, the ideal filter should be isotropic, that is, the filter's effect should be the same in all directions in an image context, with no particular sensitivity or bias towards one particular set of directions. The problem of approximating the Laplacian operator in two dimensions not only inherits the inaccuracies of the one-dimensional finite-difference approximations, but also raises the issue of azimuthal asymmetry. To check the performance of Chebyshev, Savitzky-Golay, and LoG filter, a circle of radius 60 pixels is used as a test object. The Laplacian results are computed by convolving with the above 13×13 filters. The filter size is 13×13 pixels, as shown in Fig. 3.17. In the results, the Laplacian of Chebyshev filter has peaks at 45° , 135° , 225° and 315° . The symmetry of Savitzky-Golay filter is better. However peaks are still noticeable at 0° , 90° , 180° and 270° . In Fig. 3.17(d), the result of LoG filter is isotropic in every direction if the quantization noise is ignored. Therefore the LoG filter is selected as the Laplacian filter in our STM implementation.



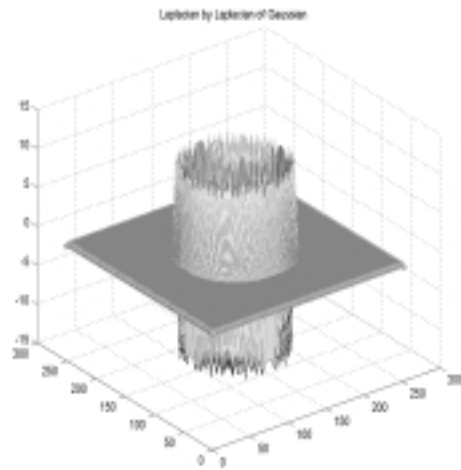
(a) Original image



(b) Laplacian by Chebyshev filter



(c) Laplacian by Savitzky-Golay filter



(d) Laplacian by LoG filter

Figure 3.17: Laplacian filter evaluations

3.6 Summary

In this chapter, several binary mask based Depth From Defocus (DFD) algorithms are proposed to improve autofocusing performance and robustness. A binary mask is defined by thresholding image Laplacian to remove unreliable points with low Signal-to-Noise Ratio (SNR). Three different DFD schemes – with/without spatial integration and with/without squaring – are investigated and evaluated, both through simulation and actual experiments. The actual experiments use a large variety of objects including very low contrast Ogata test charts. Experimental results show that autofocusing RMS step error is less than 2.6 lens steps, which corresponds to 1.73%. Although our discussion in this paper is mainly focused on a spatial domain method STM1, this technique should be of general value for different approaches such as STM2 and other spatial domain based algorithms.

Chapter 4

Blur Equalization Technique (BET)

In this chapter, a new spatial-domain Depth-from-Defocus (DFD) technique named Blur Equalization Technique (BET) is presented. It uses the S transform in the convolution mode [41] where as it is used in the deconvolution mode in STM [32], and [33]. The theoretical basis of BET relies on equalizing the *blur* or *defocus* of two different images recorded with different camera parameters. In contrast, comparable spatial-domain techniques rely on equalizing the *deblur* or *focus* of the two images. Also, BET facilitates modeling of images locally by higher order polynomials with lower series truncation errors. These differences seem to make the new technique very stable and accurate, even for low contrast objects with high levels of blur over a wide range of object distances.

The accuracy of BET is further enhanced by discarding pixels with low Signal-to-Noise ratio by thresholding image Laplacians, and relying more on the sharper one of the two blurred images in estimating the blur parameters. These steps, combined with careful calibration for sensor response, vignetting, and magnification correction, makes BET a very accurate and robust technique useful in practical applications such as depth recovery and autofocusing.

Its performance is compared with three highly effective and significantly improved versions of STM1 [34, 51]. BET is found to be superior to some of the best comparable DFD techniques in a large number of both simulation and actual experiments. Actual experiments used a large variety of objects including very low contrast digital camera test charts located at many different distances. In autofocusing experiments, BET gave an RMS error of just 1.2% in lens position compared to 1.8 %, 2.0 %, and to 2.3 % respectively for three other highly effective methods. This error of 1.2 % is very close to the best possible accuracy of about 1.0 % due to the inherent quantization errors in lens positioning (about 0.5 %), spatial resolution (1 pixel), grey-level resolution (8 bits/pixel), and noise.

For comparison, Binary Masked High Order STM (BM_HIGHORD) is presented as a counterpart. The only difference between BET and BM_HIGHORD is that BET uses the blur equalization, and BM_HIGHORD utilizes the image equalization. The experiment result shows the effectiveness of new blur equalization technique.

4.1 Blur Equalization Technique

4.1.1 Blur Equalization

In previous research, image equalization, i.e., equalizing the *focus* of the two images has been utilized in both spatial and frequency domain DFD. Ens and Lawrence [29] find a convolution filter h_3 that makes the filtered first image equal to the second image, this is image equalization in the frequency domain. In the spatial domain, equalizing the *focus* images is used in STM [32], and [33]. A commutative image equalization can be found in [53] and [54, 55].

In STM, the spatial-domain deconvolution equation $f = g_i - \frac{\sigma_i^2}{4} \nabla^2 g_i$ is used to eliminate the focused image term to obtain the blur parameter. This is equivalent

to deblurring the two defocused images g_1 and g_2 and equating the resulting focused image f . This operation may be termed *deblur* or *focus* equalization. Since this deconvolution amplifies high-frequency content, it may reduce the Signal-to-Noise Ratio (SNR). This operation, when combined with the cubic-model restriction for f as in $f(x, y) = \sum_{m=0}^3 \sum_{n=0}^{3-m} a_{mn} x^m y^n$ requires that $\nabla^2 g_1 = \nabla^2 g_2$. However, in practical applications, when the Laplacians are computed for the two images, they are almost never equal, and sometimes significantly different from each other. Using higher order terms with higher image derivatives may reduce SNR. Therefore, unnecessary errors are introduced in depth recovery, particularly when the object contrast is too low or too high.

BET is an alternative to STM. In BET, in contrast with STM, we use *blur* or *defocus* equalization by convolving the two defocused images with suitable PSFs that correspond to *attenuating* high-frequency content which improves SNR. Remarkably, this removes the requirement of $\nabla^2 g_1 = \nabla^2 g_2$. Following the previous notation, two defocused images $g_i(x, y), i = 1, 2$ can be expressed as:

$$g_i(x, y) = f(x, y) \otimes h_i(x, y) \quad i = 1, 2 \quad (4.1)$$

where $h_i(x, y)$ is the PSF of corresponding defocused image at position i . We have

$$g_1(x, y) \otimes h_2(x, y) = [f(x, y) \otimes h_1(x, y)] \otimes h_2(x, y) \quad (4.2)$$

$$g_2(x, y) \otimes h_1(x, y) = [f(x, y) \otimes h_2(x, y)] \otimes h_1(x, y) \quad (4.3)$$

From the commutative and associative property of convolution, the right side of Eqn. (4.2) equals the right side Eqn. (4.3), that is:

$$g_1(x, y) \otimes h_2(x, y) = g_2(x, y) \otimes h_1(x, y) \quad (4.4)$$

Using Forward S Transform for convolution:

$$g_1(x, y) \otimes h_2(x, y) = g_1(x, y) + \frac{\sigma_2^2}{4} \nabla^2 g_1(x, y) + \frac{\sigma_2^4}{24} (\nabla^2)^2 g_1(x, y) + R(O^6) \quad (4.5)$$

$$g_2(x, y) \otimes h_1(x, y) = g_2(x, y) + \frac{\sigma_1^2}{4} \nabla^2 g_2(x, y) + \frac{\sigma_1^4}{24} (\nabla^2)^2 g_2(x, y) + R(O^6) \quad (4.6)$$

Combining Eqn. (4.4), (4.5) and (4.6), and ignoring the higher order terms $R(O^4, O^6)$, we have:

$$g_1(x, y) + \frac{\sigma_2^2}{4} \nabla^2 g_1(x, y) = g_2(x, y) + \frac{\sigma_1^2}{4} \nabla^2 g_2(x, y) \quad (4.7)$$

In deriving the above equation, the approximation may be much less than it appears to be, because we are only neglecting the *difference* of fourth and higher order terms between the left and the right sides of the equation. We are not neglecting the terms on only one side, or separately on the two sides. Experimental results support this observation.

Using $\sigma_i = m_i u^{-1} + c_i, i = 1, 2$ we get:

$$a_1 \sigma_1^2 + b_1 \sigma_1 + c_1 = 0 \quad (4.8)$$

where the coefficients are defined as:

$$a_1 = \frac{\nabla^2 g_2}{\nabla^2 g_1} - 1 \quad (4.9)$$

$$b_1 = 2\beta \quad (4.10)$$

$$c_1 = - \left[\frac{4(g_1 - g_2)}{\nabla^2 g_1} + \beta^2 \right] \quad (4.11)$$

4.1.2 SNR Based Thresholding

Signal-to-Noise Ratio (SNR) of an image is affected by several factors such as object contrast, sensor noise, sampling rate, and quantization error. We found that image Laplacians have very low SNR leading to large errors in the estimation of σ_2 . Therefore, thresholding on the Laplacian values is introduced to improve the robustness. This operation removes unreliable points with low SNR. Laplacian Mask $M_0(x, y)$ is formed by:

$$M_0(x, y) = \begin{cases} 1 & \nabla^2 g_2 \geq T \\ 0 & \text{otherwise} \end{cases} \quad (4.12)$$

A Delta Mask $M_1(x, y)$ is also used here to guarantees the real property of the solution.

$$M_1(x, y) = \begin{cases} 1 & \Delta_1 \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (4.13)$$

where $\Delta_1 = b_1^2 - 4a_1c_1$.

The final binary mask $M_{f1}(x, y)$ is obtained from the BIT-AND operation:

$$M_{f1}(x, y) = M_0(x, y) \& M_1(x, y) \quad (4.14)$$

The computation of σ_1 is guided by $M_{f1}(x, y)$, as shown in Fig. 4.1(d). The best estimation of σ_1 is considered as the average based on $M_{f1}(x, y)$.

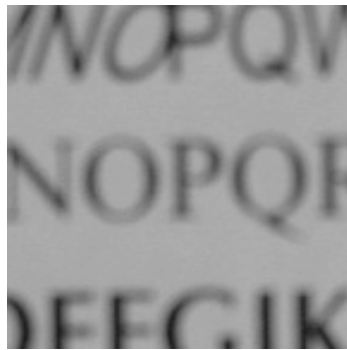
4.1.3 Image Switching Based on Sharpness Measure

Another quadratic equation in σ_2 can be derived as:

$$a_2\sigma_2^2 + b_2\sigma_2 + c_2 = 0 \quad (4.15)$$

and the coefficients are:

$$a_2 = 1 - \frac{\nabla^2 g_1}{\nabla^2 g_2} \quad (4.16)$$



(a) Blurred window at step 35



(b) Laplacian mask



(c) Delta mask



(d) Final mask

Figure 4.1: Binary masks

where

$$b_2 = 2\beta \quad (4.17)$$

$$c_2 = -\left[\frac{4(g_1 - g_2)}{\nabla^2 g_2} + \beta^2\right] \quad (4.18)$$

and similarly the binary mask $M_{f_2}(x, y)$ can be formed.

Mathematically, Eqn. (4.8)-(4.11) and Eqn. (4.15)-(4.18) are identical. However, computationally we find that the two equations have different noise sensitivity, perhaps due to noisy image Laplacians appearing in the denominator in the equations. It seems that solving for the blur parameter of the sharper or more focused image gives more accurate results. In this case, the absolute value of the image Laplacians appearing in the denominators will be larger than otherwise. A simple focus or sharpness measure is computed for the two images. We define the Sum of Laplacian in the Focusing Window (SLFW) $L_i = \sum_x \sum_y |\nabla^2 g_i(x, y)|, i = 1, 2$ as the sharpness measure. The sharper images will have a smaller blur parameter σ . Now the equation to be solved is determined as follows:

$$\begin{cases} a_1\sigma_1^2 + b_1\sigma_1 + c_1 = 0, & L_1 \geq L_2 \\ \sigma_2 = \sigma_1 + \beta \\ a_2\sigma_2^2 + b_2\sigma_2 + c_2 = 0 & L_1 < L_2 \end{cases} \quad (4.19)$$

A computer simulation is conducted using a high contrast object “Letter”. the reliability measure SLFW is plotted in Fig. 4.2, and the corresponding switching scheme is displayed in Fig. 4.3. In Fig. 4.3, “+” curve is calculated from Eqn. (4.8)-(4.11), and “*” curve is obtained by Eqn. (4.15)-(4.18).

4.1.4 Experimental Results

BET algorithm described above was implemented on a compact digital camera. The flowchart of the algorithm is shown in Fig. 4.4.

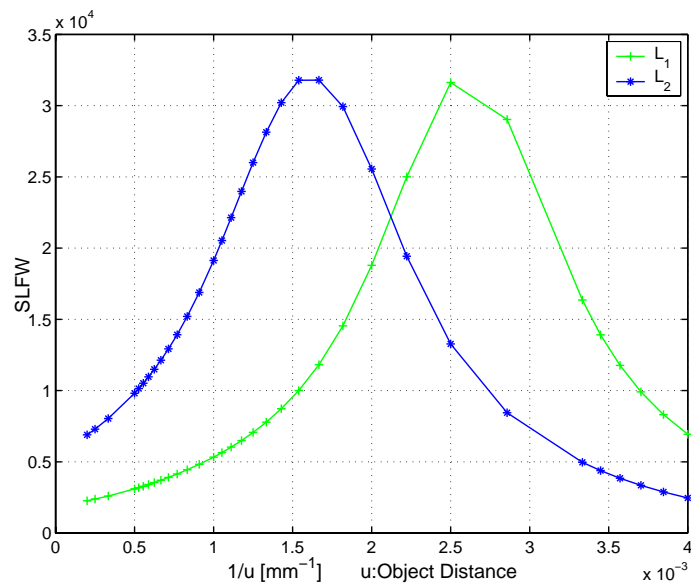


Figure 4.2: Sum of Laplacian in Focusing Window (SLFW)

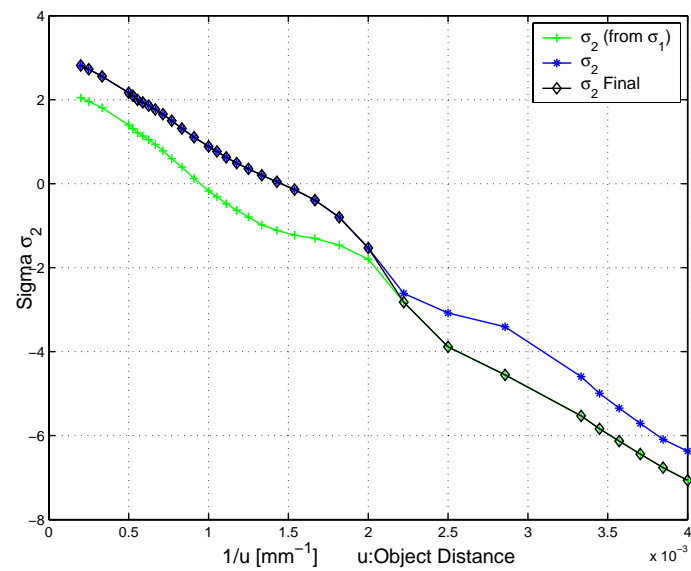


Figure 4.3: Switching mechanism

The performance of DFD algorithms are evaluated with real experiments using eight different objects shown in Chapter 3. Some of them are standard camera test charts used in evaluating the autofocus performance of new digital camera models by industry. The last two objects are of very low contrast that are very difficult to autofocus. Each object was placed at eight different distances in the range of 325 mm to 1350 mm at roughly 150 mm intervals. The distance and the corresponding steps are listed as before. The steps were obtained using the DFF algorithm. The F-number was set to 2.8, and the focal length was set to 19.5 mm. The focusing window size was 96×96 located at the center of the scene. At the beginning, a Gaussian smoothing filter with a sigma parameter of 1.8 pixels and size 9×9 was used to smooth images. The image Laplacians were also computed using the corresponding LOG filter of size 9×9 . The image Laplacians were thresholded to weed out low contrast pixels with low SNR. The non-linear sensor response was calibrated and corrected as described in [51]. One value of depth or blur parameter sigma was estimated in each pixel. The estimates were averaged in the 96×96 window.

Measurement results and RMS error for BET are plotted in Fig. 4.6 and Fig. 4.7 respectively. In order to compare BET's performance with some of the best existing DFD techniques, three substantially improved variations of STM (see [34] for details) were tested in the same set of experiments. These variations use "deblur" equalization as opposed to "blur" equalization as explained earlier. The results for BET and the three variations of STM are plotted together in Fig. 4.7. The RMS step error for BET is 1.8 lens steps, which corresponding 1.2% of step range. This error of 1.8 steps may be close to the best achievable as the step quantization error itself is 0.5 steps, and additional error is expected due to sampling, grey level quantization, and image noise. BET is better than the other STM methods which have rms errors of 2.6, 2.9, and 3.5 step errors. BET has a particularly better performance at far field such as positions 6, 7 and 8 when one of the image will be highly blurred. Therefore, since

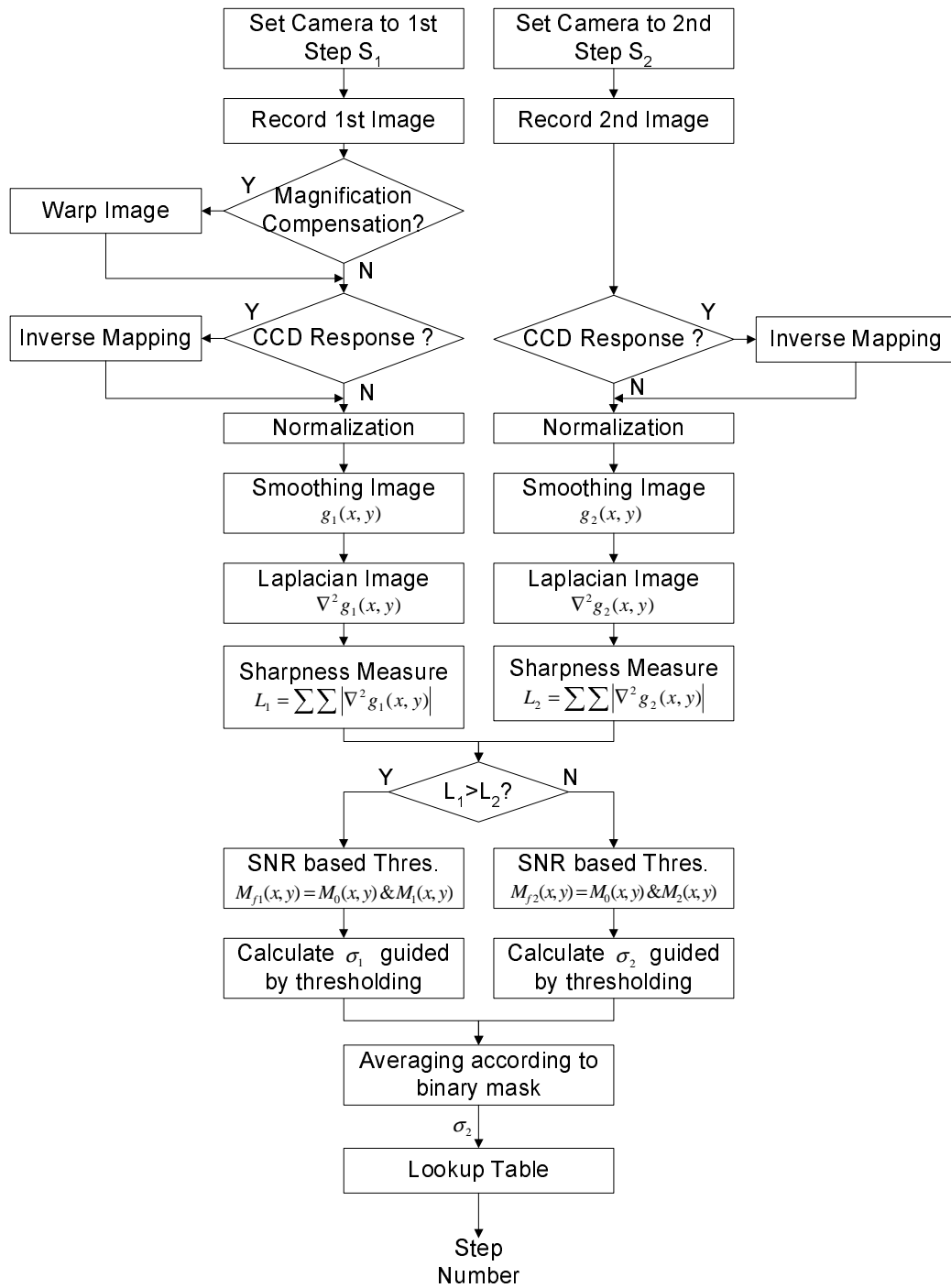


Figure 4.4: Flow chart of BET algorithm

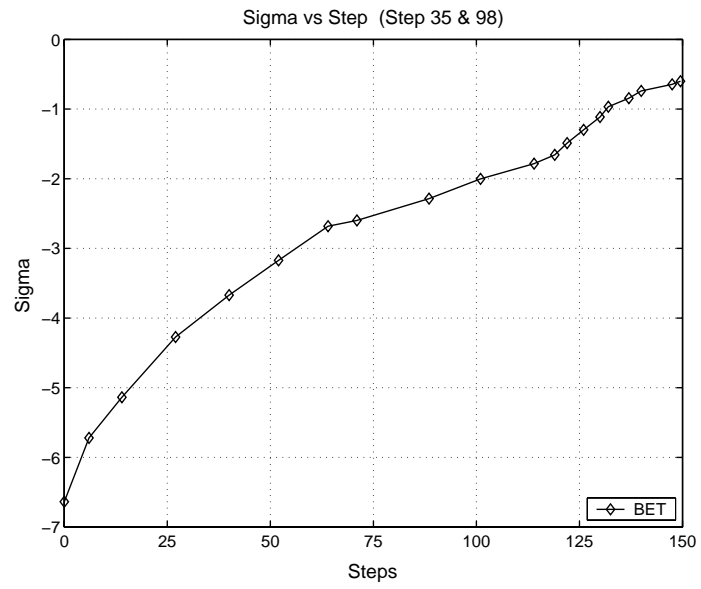


Figure 4.5: Sigma vs. step number

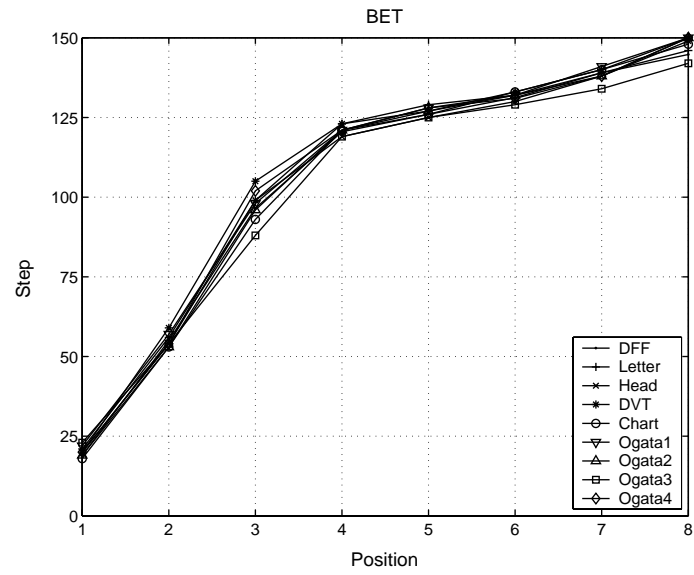


Figure 4.6: Measurement results for test objects (real data)

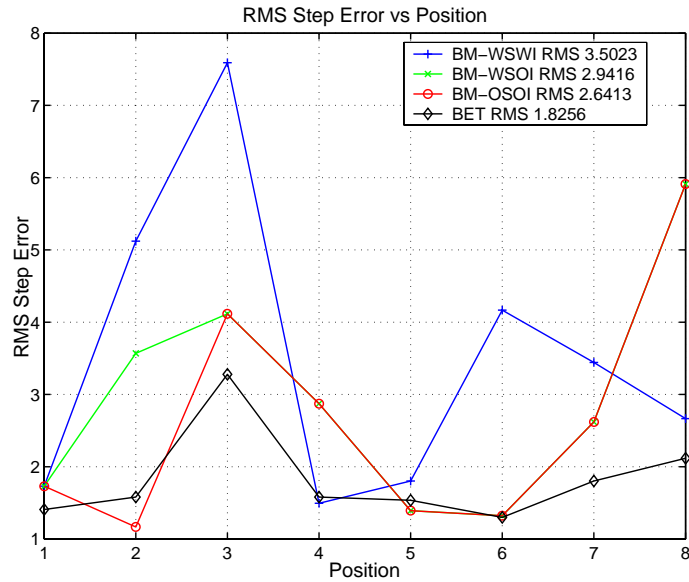


Figure 4.7: RMS step error (real data)

the computational requirement of BET and other STM methods are the same, BET is recommended for practical applications.

The maximum RMS step error of BET occurs at position 3. As Fig. 4.5 shows, position 3 is located in "flat" areas in the sigma-step curves that approximately lie in the range from step 60 to step 110 for the compact digital camera. In this area, we find that a small variation in sigma may cause a large fluctuation in the focusing step. However the corresponding change in focus or image sharpness measure is not significant. In this sense, BET's performance is somewhat better than what is indicated by the rms error in lens step position. BET was also tested and compared in extensive camera simulation experiments. The results and conclusions were similar to actual experiments reported here.

4.2 Counterpart: Image Equalization

$\nabla^2 g_1 = \nabla^2 g_2$ is valid only under the third order polynomial assumption. However, for arbitrary scenes, the output from low pass filter may be higher than the third order polynomial. Thus $\nabla^2 g_1 \neq \nabla^2 g_2$ is common in real applications.

In order to relax the assumption, a new algorithm named Binary Mask based STM1 by High Order (unequal Laplacian approximation) (BM_HIGHORD) is implemented.

Substituting Eqn.(3.11) into Eqn. (3.14), a quadratic equation is obtained:

$$a_3 \sigma_1^2 + b_3 \sigma_1 + c_3 = 0 \quad (4.20)$$

where

$$a_3 = \frac{\nabla^2 g_1}{\nabla^2 g_2} - 1 \quad (4.21)$$

$$b_3 = 2\beta \quad (4.22)$$

$$c_3 = - \left[\frac{4(g_1 - g_2)}{\nabla^2 g_2} + \beta^2 \right] \quad (4.23)$$

and σ_1 can be calculated by solving Eqn. (4.20).

Another quadratic equation about σ_2 can also be derived from Eqn. (3.11) and (3.14), and the binary mask $M_{f_2}(x, y)$ is formed similarly as Eqn. (4.12)-(4.14).

$$a_4 \sigma_2^2 + b_4 \sigma_2 + c_4 = 0 \quad (4.24)$$

where

$$a_4 = 1 - \frac{\nabla^2 g_2}{\nabla^2 g_1} \quad (4.25)$$

$$b_4 = 2\beta \quad (4.26)$$

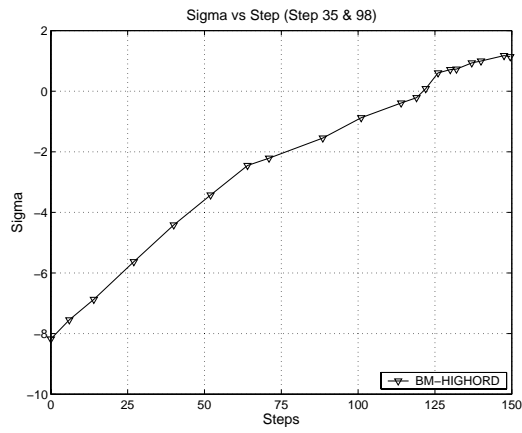
$$c_4 = - \left[\frac{4(g_1 - g_2)}{\nabla^2 g_1} - \beta^2 \right] \quad (4.27)$$

The switching mechanism is formulated as:

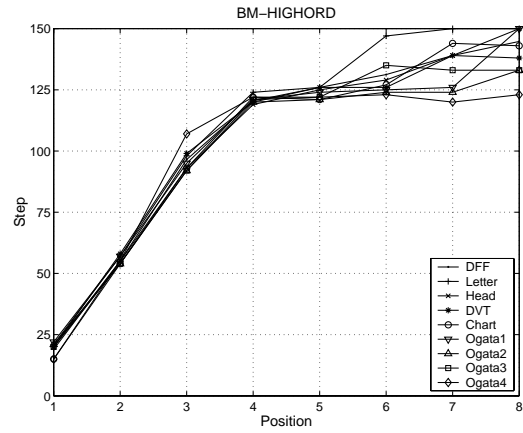
$$\left\{ \begin{array}{l} a_3\sigma_1^2 + b_3\sigma_1 + c_3 = 0, \quad L_1 \geq L_2 \\ \sigma_2 = \sigma_1 + \beta \\ a_4\sigma_2^2 + b_4\sigma_2 + c_4 = 0, \quad L_1 < L_2 \end{array} \right. \quad (4.28)$$

Binary Mask based STM1 by High Order Equation (BM_HIGHORD) has been tested on the same image database mentioned in Section 4.1. The same configuration as described in previous section is adopted for experiments. The size of acquired defocused images is 640×480 pixels. The size of focusing window is 96×96 pixels, and the size of Laplacian filter and LoG filter are 9×9 pixels. The same series of test objects as shown in Fig.3.8 are used. The object distance is same as listed in Table 4.1. Fig. 4.8(a) is the sigma-step table, (b) shows the measurement result for test objects, and the RMS step error is plotted in (c). Compared with BET algorithm, the accuracy express of RMS step error is similar at Position 2, 3, 4 and 5; when the object to be measured is far away from camera (Position 6, 7 and 8), the RMS step error increases dramatically; when the object is located in the near field, a higher RMS step error is observed. The mean RMS step error of BM_HIGHORD is 4.58 lens steps which corresponding to 3.05% of whole range. While the mean RMS step error for BET is 1.83 lens steps (1.22%). The only difference between BET and BM_HIGHORD is the blur equalization and the image equalization. This comparison experiment shows the effectiveness of the new blur equalization technique.

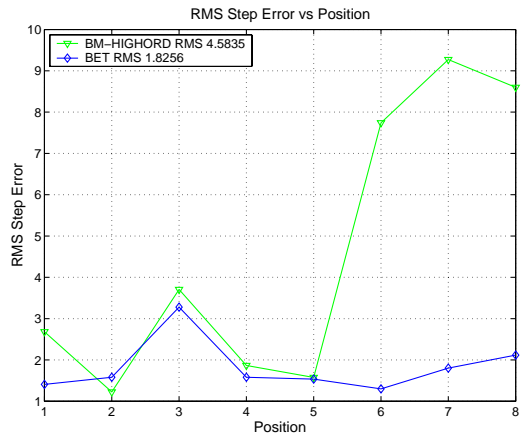
It should be mentioned that, a solution from fourth-order equation in Eqn. (3.13) and (3.14) have been attempted. But the measurement result is too sensitive to noise, and the computation is intensive.



(a) Sigma vs. step table



(b) Measurement results



(c) RMS step error

Figure 4.8: Measurement results for BM-HIGHORD (real data)

4.3 Summary

A new depth-from-defocus technique named Blur Equalization Technique is presented. It is a spatial-domain technique that can provide dense depth estimates using modest local computations in small image windows (e.g. 9×9). The theoretical basis of BET is different from other Depth-from-Defocus techniques. The computational algorithm is relatively simple. In both simulation and actual experiments, BET has been demonstrated to perform better than some of the best existing techniques, on many objects at different distances, including some standard test objects and very low contrast objects. The accuracy and robustness of BET seems to be close to the best possible. As a counterpart, BM_HIGHORD is also implemented for comparison. The only difference between BET and BM_HIGHORD is the blur equalization and the image equalization. The experiment result shows the effectiveness of new blur equalization technique. Therefore, it is recommended for use in practical applications. While the BET presented here uses the Spatial domain convolution/deconvolution transform, the basic idea of “blur equalization” can be extended to Fourier domain and other DFD techniques found in the current literature.

Chapter 5

DFD Compensation and Calibration

Many factors affect the performance of DFD algorithms. In particular, *nonlinear sensor response*, *lens vignetting*, and *magnification variation* affect the accuracy of DFD. In order to implement DFD on off-the-shelf commercial digital cameras, these factors need to be calibrated and corrected. In this chapter, we present new calibration methods for these three factors. Their correctness and effects on performance of DFD have been evaluated with experiments.

Most digital cameras utilize the nonlinear sensor response to extend the dynamic gray-level range through a log-like or gamma transform. DFD theory requires inverse mapping of this non-linear response to linear response through calibration. The intensity measured by the image sensor depends on illumination, exposure period, and reflectance. A method is proposed and tested for correcting this non-linear sensor response.

Optical vignetting is the phenomenon where the effective light energy transmitted by the optical system decreases with increasing inclination of light rays with respect to the optical axis. A vignetting calibration method is implemented and tested for its effects on DFD performance.

In DFD based autofocus where the lens position is moved, the magnification of an object will change when two images are recorded with different camera parameters. A magnification calibration method is implemented and the estimation error has been evaluated.

The calibration methods for nonlinear response and vignetting correction are direct methods based on illumination measurement using a digital lux tester. They do not need expensive and strictly controlled laboratory environment and can be used for off-the-shelf cameras. Therefore, these calibration methods should be of general value to other image based algorithms.

5.1 Nonlinear Sensor Response Compensation

The formation of a digital image on the image sensor of a camera can be described by:

$$g(x, y) = \int_0^{\tau} \int_0^{+\infty} q_s(x, y, \lambda, t) s(\lambda) d\lambda dt \quad (5.1)$$

where $g(x, y)$ is the photo-quantity of the specific sensor element (x, y) ; $q_s(x, y, \lambda, t)$ is the actual light energy falling on the image sensor (x, y) ; $s(\lambda)$ is the spectral sensitivity of an element of the sensor. τ denotes the integration period, which is controlled by exposure time of the camera. From this equation, photo-quantity $g(x, y)$ is neither a radiometric nor a photometric unit, since it is also related to the sensor spectral sensitivity $s(\lambda)$. For a specific camera system, the photo-quality depends on the light energy falling on the sensor cell per unit time, and camera exposure time.

Once the parameters of DFD (s_1, s_2, f, D for STM1) are fixed, the measurement from DFD algorithms should only be related to object distance, and should not be affected by other changes such as illumination and camera exposure. However, most digital cameras utilize the nonlinear sensor response to extend the dynamic gray-level

range through transforms (e.g. $\log(z)$).

5.1.1 Error Analysis of Non-linear Sensor Response

The nonlinear sensor response is a point-wise mapping, which can be formulated by a function K :

$$g'(x, y) = K [g(x, y)] \quad (5.2)$$

where $g'(x, y)$ is the distorted intensity after point-wise sensor response mapping, and $g(x, y)$ is the original photo-quantity formed as in Eqn. (5.1).

If digital images are quantized to n bits, the point-wise sensor mapping can be expressed by the transform vector \bar{k} without sacrificing generality:

$$K(i) = k_i I + b \quad (5.3)$$

where b is the dark offset, I is the original photo-quantity vector, and k_i is the i th coefficient to map from level i in the original photo-quantity g to distorted intensity g' .

$$I = [0 \quad 1 \quad \dots \quad 2^n - 1]^t \quad (5.4)$$

$$\bar{k} = [k_0 \quad k_1 \quad \dots \quad k_{2^n-1}] \quad (5.5)$$

For a linear mapping, the components in the coefficient vector for each level should be the same, i.e. $k_i = k_j = k$; while for a nonlinear mapping, $k_i \neq k_j$ is valid for some level i, j . DFD theory is based on a linear brightness mapping when a focused image is blurred by a convolution process.

The linearization of brightness can be obtained by inverse mapping of sensor response K^{-1} .

$$g_0 = K^{-1} [g'] = K^{-1} [K(g)] = g \quad (5.6)$$

5.1.2 Direct Calibration

Sensor response calibration is needed to compensate for its non-linearity. There are several approaches to measure the sensor response, through statistical averaging on the whole image for arbitrary scenes, such as [56] and [57]. However averaging on the image plane includes the effect of vignetting, which will be discussed in Section 5.2. The resulting response is a weighted average of on-axis points and off-axis points, and the weight depends on scene content.

A direct measurement method is presented here to calibrate the nonlinear response for off-the-shelf cameras. The setup of the nonlinear sensor response calibration is briefly outlined in Fig. 5.1. A diffusive white screen WS is illuminated by multiple light sources from L_1 to L_4 . The light sources are controlled from the linear lamp controller module LC to create variable/adjustable illumination. The intensity at the central area of the white screen is measured by a Digital Lux Tester YF-1065. The image of the white screen is acquired by a digital camera to be calibrated. A lookup table is established by changing the illumination incrementally while recording the image at each illumination step. The relationship between camera gray-levels and normalized illumination gray-levels is shown in Fig. 5.2. Mean brightness in a 10×10 image region is used for reducing noise in the central area of the white screen.

To evaluate the effect of nonlinear sensor response, a series of DFD experiments were conducted under different photo-quantity conditions. The ambient illumination is 253 lux measured at the center of an object plane, and the distance from the front surface of lens to the object plane is 540 mm. According to Eqn. (5.1), the photo-quantity can be changed by either illumination level or camera shutter speed. We control the shutter speed to obtain a wider range on photo-quantity. The shutter speed changes from 15.625 ms to 500 ms, which correspond to a change factor of 32 in photo-quantity, as displayed in Fig. 5.3. In Fig. 5.3, Images range from under-

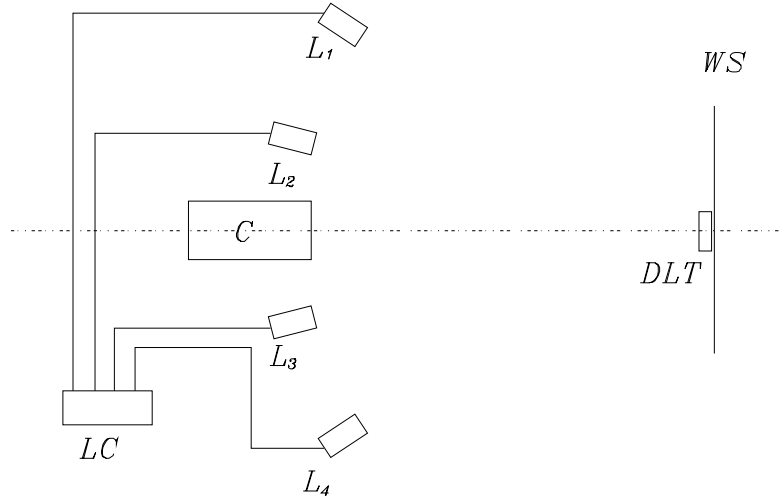
exposed as in (a), (b), to over-exposed as in (e), (f) . The photo-quantity is doubled at each stage from (a) to (f), however the gray level of observed image does not increase correspondingly due to camera range compression. From (a) to (f), images are captured with exposure time of 15.625, 31.25, 62.5, 125, 250 and 500 ms. Assume the photo-quantity in (a) as a unit measure, the photo-quantity doubles in each stage from (a) to (f).

About 384 DFD experiments were conducted at 8 different random positions and 6 different exposure levels. The 8 randomly selected positions are 325.1, 383.1, 474.1, 538.6, 630.0, 784.2, 1058.4 and 1353.6 mm respectively, which are measured from the front surface of the lens. The corresponding lens steps are obtained from Depth From Focus (DFF) experiments, and they are 25, 38, 58, 74, 101, 122, 131 and 145 lens step respectively.

As shown in Fig. 5.4, “woNL” means DFD STM1 without Non-Linear sensor response compensation; “wNL” stands for DFD STM1 with Non-Linear sensor response compensation; and “DFF” means result from Depth-from-Focus. STM1 without sensor response compensation has a mean error of up to 14 lens steps (Distance 630.0 mm, Shutter Speed 15.625 ms), while the corresponding RMS lens step error is 0.354 step. After nonlinear sensor response compensation, at the same photo-quantity, the mean lens step error is reduced to 1.667 step, and all 8 DFD measurements get exactly the same step number, since RMS lens step error is 0. The mean and max of the mean lens step errors are plotted in Fig. 5.5 (a) and (b) respectively. The detailed results of DFD without/with nonlinear sensor response compensation are listed in Table 5.1 and 5.2. Table 5.1 shows the mean and RMS lens step error before nonlinear sensor response compensation, while Table 5.1 shows errors after sensor response calibration.

When the photo-quantity continues to increase from (e) to (f), step shifts in far field for both DFF and DFD can be observed. In the image (f), the sensor is already saturated, and the observed image is no longer a correct measure of photo-quantity.

In this extreme condition, the error of DFD with sensor compensation (-4.431 ± 0.236 step) is still better than the one without compensation (-8.667 ± 0.535 step).



C: Camera L1, L2, L3, L4: Light Source
 WS: White Screen LC: Linear Lamp Controller
 DLT: Digital Lux Tester

Figure 5.1: Setup for nonlinear sensor response calibration

5.2 Lens Vignetting Compensation

Optical vignetting is the phenomenon wherein the effective light energy transmitted by the optical system decreases with increasing inclination of light rays with respect to the optical axis.

The consequence of optical vignetting for a focused scene is merely a reduced brightness towards the image corners. However, optical vignetting can also have a pronounced effect on out-of-focus parts of the image. Because the shape of an Out-Of-Focus Highlight (OOFH) mimics the shape of the clear aperture, this leads to the so-called cat's eye effect. With an increasing distance from the optical axis the shape

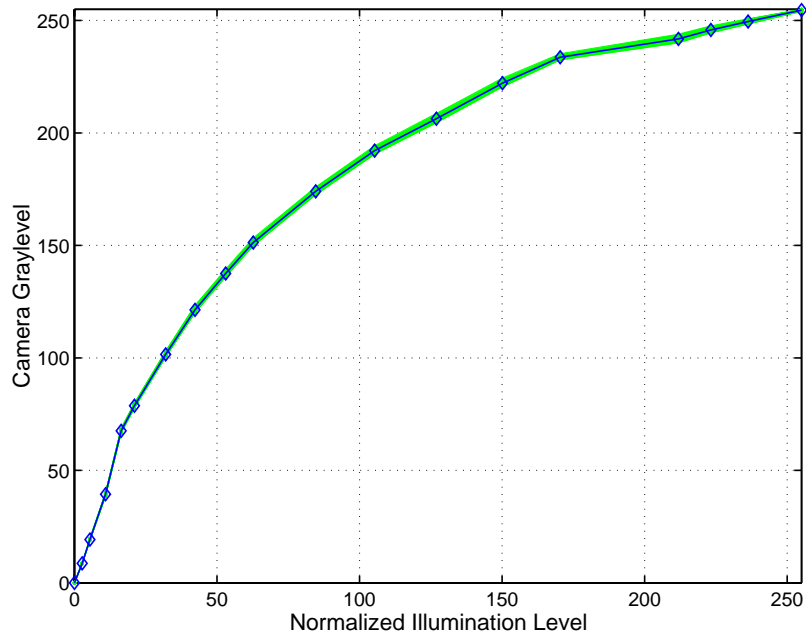


Figure 5.2: Nonlinear sensor response

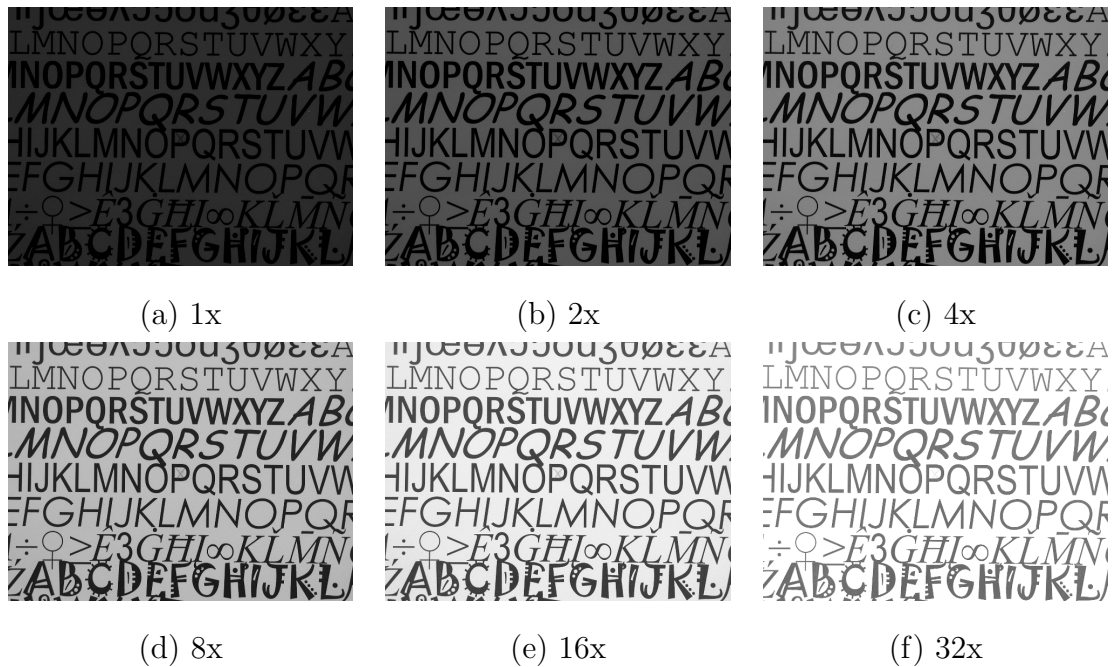


Figure 5.3: Images obtained at different photo-quantity

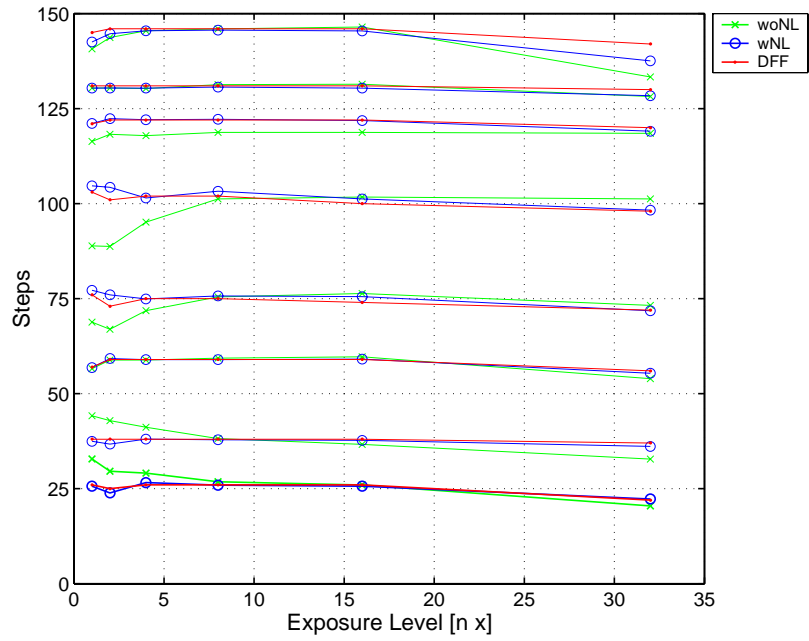
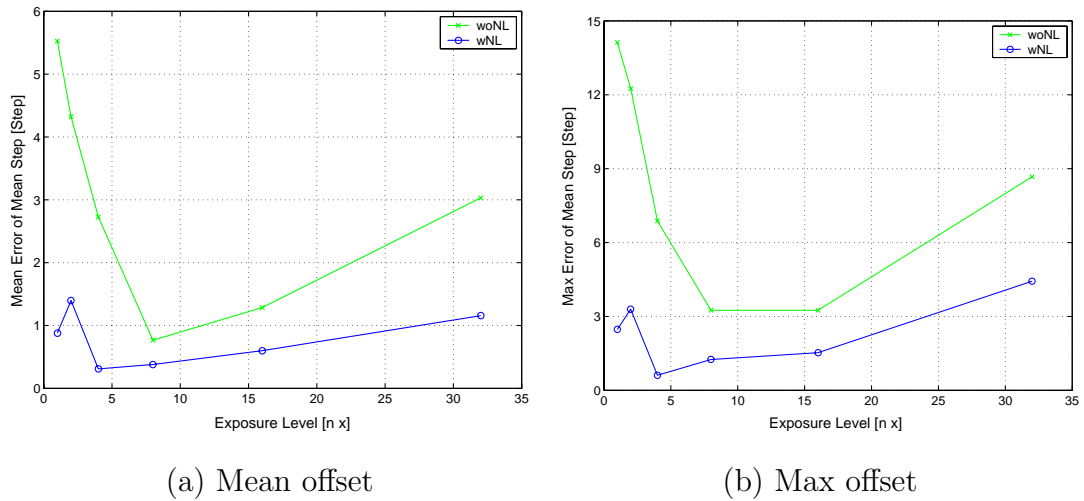


Figure 5.4: DFD results before/after nonlinear sensor response calibration



(a) Mean offset

(b) Max offset

Figure 5.5: Mean and max offsets before/after nonlinear sensor compensation

	Exposure 1		Exposure 2		Exposure 3		Exposure 4		Exposure 5		Exposure 6	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Pos 1	6.833	0.000	4.583	0.463	3.083	0.463	0.833	0.000	-0.042	0.354	-1.542	0.518
Pos 2	6.167	0.000	4.917	0.463	3.167	0.000	0.167	0.000	-1.333	0.535	-4.208	0.518
Pos 3	-0.417	0.463	-0.167	0.000	-0.167	0.000	0.333	0.535	0.708	0.354	-2.042	0.354
Pos 4	-7.167	0.000	-6.042	0.354	-3.167	0.000	0.458	0.518	2.333	0.535	1.208	0.518
Pos 5	-14.125	0.354	-12.250	0.707	-6.875	0.641	-0.750	0.463	1.750	0.463	3.250	0.707
Pos 6	-4.625	0.991	-3.750	0.463	-4.125	0.518	-3.250	0.463	-3.250	0.463	-1.500	0.535
Pos 7	-0.583	0.463	-0.583	0.463	-0.708	0.354	0.292	0.354	0.417	0.463	-1.833	0.000
Pos 8	-4.292	0.835	-2.292	0.991	-0.542	1.061	-0.042	0.991	0.458	0.744	-8.667	0.535

Table 5.1: DFD lens step error by mean and RMS before nonlinear sensor compensation

	Exposure 1		Exposure 2		Exposure 3		Exposure 4		Exposure 5		Exposure 6	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Pos 1	-0.347	0.236	-1.139	0.309	0.611	0.000	-0.056	0.000	-0.347	0.236	0.278	0.000
Pos 2	-0.528	0.309	-1.278	0.000	0.056	0.000	-0.153	0.345	-0.278	0.000	-0.861	0.309
Pos 3	-0.139	0.309	0.236	0.236	-0.056	0.000	-0.056	0.000	0.111	0.356	-0.639	0.309
Pos 4	1.194	0.309	2.986	0.236	-0.056	0.000	0.694	0.309	1.528	0.309	-0.181	0.345
Pos 5	1.667	0.000	3.292	0.236	-0.500	0.356	1.250	0.309	1.250	0.309	0.292	0.427
Pos 6	0.083	0.309	0.375	0.345	0.083	0.309	0.167	0.000	-0.125	0.236	-0.958	0.345
Pos 7	-0.611	0.000	-0.569	0.236	-0.611	0.000	-0.319	0.236	-0.611	0.000	-1.611	0.000
Pos 8	-2.472	0.992	-1.306	0.690	-0.514	0.496	-0.347	0.661	-0.556	0.356	-4.431	0.236

Table 5.2: DFD lens step error by mean and RMS after nonlinear sensor compensation

of the OOFH progressively narrows and starts to resemble a cat’s eye. The larger the distance from the image center, the narrower the cat’s eye becomes.

A vignetting calibration method is used to evaluate the effect of vignetting on DFD measurement. If a uniform illumination is available, the vignetting coefficient could be simply calculated from a single image of a diffusive white screen. However it is difficult to obtain a uniform illumination that is accurate, although not impossible. An alternative way is used in our calibration. The setup for vignetting calibration was similar to that in Fig. 5.1. A 5×5 grid pattern is used as a calibration pattern (see Fig. 5.6(a)). In each grid, illumination is measured by the Digital Lux Tester YF-1065 at the center of grids, and the image of the grid pattern is captured by the camera. The gray level obtained is a transformed value of real photo-quantity due to the nonlinearity of sensor response. A lookup table for the reverse mapping discussed in Section 3 is used. The vignetting coefficient is calculated by the ratio of illumination intensity at pixel (x, y) to the intensity at the center of the image. Due to the rotational symmetry property, the relation between vignetting coefficient and pixel distance in polar coordinate is obtained from a third-order polynomial fitting:

$$V(\rho) = -3.1064 \times 10^{-9} \rho^3 + 3.2488 \times 10^{-7} \rho^2 - 6.7845 \times 10^{-5} \rho + 1 \quad (5.7)$$

The result of vignetting factor vs. pixel distance is shown in Fig. 5.6(b). From Fig. 5.6(b), if the DFD AF window is in the center area, the distortion of vignetting can be ignored. (for a 96×96 focusing window, the intensity attenuation is 0.24%). When the focusing window is near a corner of the view, there could be a 12.1% difference in the diagonal direction. In this case, vignetting should be compensated by multiplying the reciprocal of the corresponding vignetting coefficient.

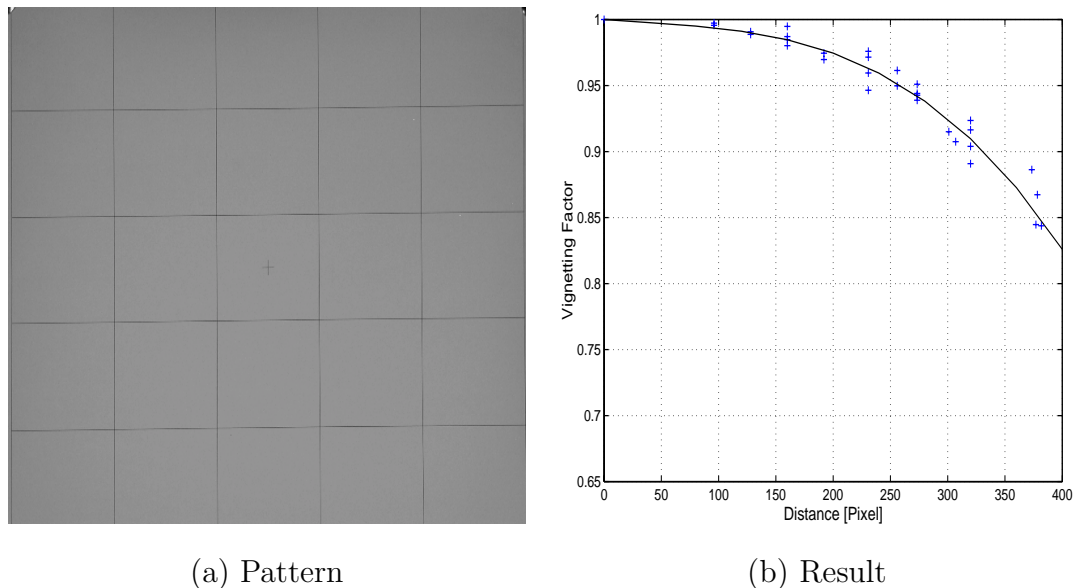


Figure 5.6: Vignetting calibration

5.3 Magnification Calibration

In STM1, when the object to be focused is fixed, there is a magnification change between defocused images acquired at two different lens steps. A magnification calibration method is developed to determine this variation.

A chessboard pattern whose grid size is 15 mm by 15 mm, is captured by the camera at focus steps 35 and 98, and the corner points are detected as shown in Fig. 5.7. The distance between the camera and the chessboard pattern is 500 mm, which corresponds to the focused image approximately at step 62. For convenience, we define the defocused image at step 35 as image 1, and the one at step 98 as image 2. The transformation between image 1 and image 2 can be calculated through a projection matrix.

The corners on image 1 and image 2 are detected and sorted row by row into two corner arrays respectively. Corners on the same position of array make a corresponding corner pair. There are 11×8 corner pairs as shown in Fig. 5.7(a) and

(b).

For each corresponding corner pair, the coordinates of the corners can be expressed as:

$$V_i = MU_i \quad i \in N \quad (5.8)$$

where U_i and V_i are coordinates of corner pairs in image 1 and image 2 respectively. These coordinates are expressed in projective space, i.e. $U_i = [x_{1,i} \ y_{1,i} \ 1]^t$, and $V_i = [x_{2,i} \ y_{2,i} \ 1]^t$. N is the total number of corresponding pairs. M is a 3×3 transformation matrix. Since we have no prior knowledge about the transformation, there are 9 unknown elements in M .

For N pairs of corner pairs, a least-square matrix can be obtained:

$$AM' = b \quad (5.9)$$

where A is a $3N \times 9$ matrix, and b is a $3N$ column vector that is made from lapping over V_i , $i = 1, \dots, N$.

$$A = \begin{bmatrix} x_{1,1} & y_{1,1} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{1,1} & y_{1,1} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{1,1} & y_{1,1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{1,N} & y_{1,N} & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & x_{1,N} & y_{1,N} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & x_{1,N} & y_{1,N} & 1 \end{bmatrix} \quad (5.10)$$

$$b = \begin{bmatrix} x_{2,1} & y_{2,1} & 1 & \cdots & x_{2,N} & y_{2,N} & 1 \end{bmatrix}^t \quad (5.11)$$

and M' is reorganized from M :

$$M' = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{21} & m_{22} & m_{23} & m_{31} & m_{32} & m_{33} \end{bmatrix}^t \quad (5.12)$$

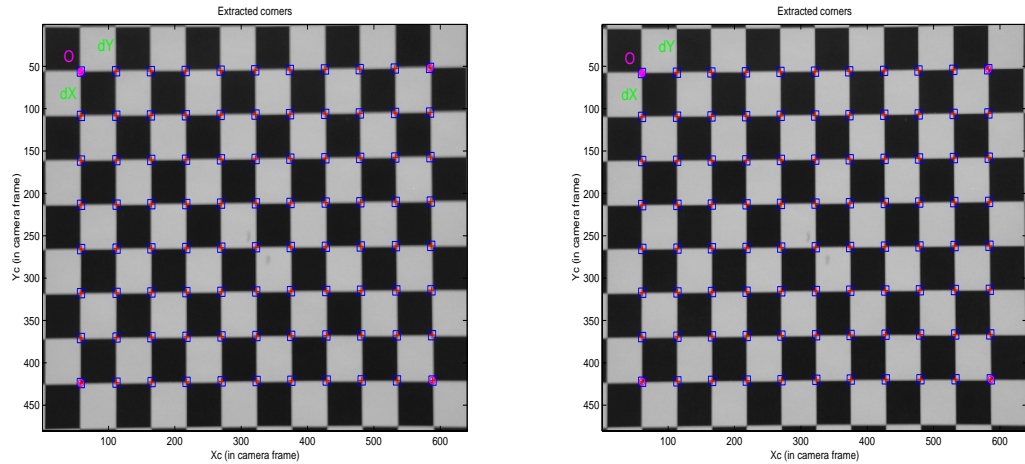
Then the transform matrix M can be calculated by:

$$M' = (A^t A)^{-1} A^t b \quad (5.13)$$

The transformation matrix M between image 1 and image 2 is calculated and reformed from Eqn. (5.13):

$$M = \begin{bmatrix} 1.0083 & 0 & -2.6526 \\ 0 & 1.0082 & -2.0510 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.14)$$

Using this matrix M to project image 1 to image 2, the projection error for each corner is plotted in Fig. 5.8. The projection error can be expressed by mean and RMS pixel error ($-2.283e-013 \pm 0.061$, $-1.389e-014 \pm 0.050$), and the maxim error is less than 0.2 pixel both in x and y direction. This is an acceptable error as the sampling interval is quantized to 1 pixel. The new image can be generated by a bicubic interpolation of image 1.



(a) Chessboard pattern captured at step 35

(b) at step 98

Figure 5.7: Magnification calibration using pattern captured at different steps

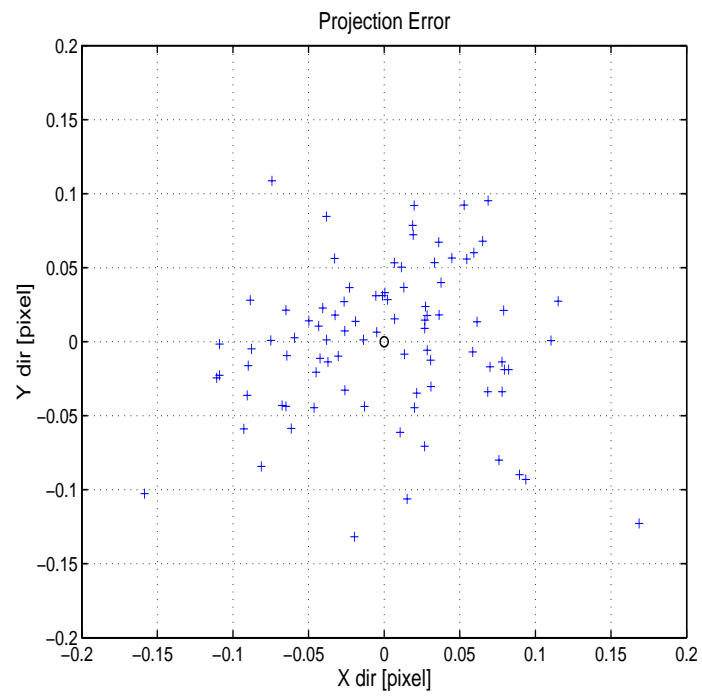


Figure 5.8: Project error from estimated transformation matrix M

In this chapter, calibration methods and procedures for nonlinear sensor response, optical vignetting, and magnification variation are presented. The correctness and effects on the performance of DFD have been evaluated with experiments. These calibrations do not need expensive and strictly controlled laboratory environment. They can be used for off-the-shelf cameras. Therefore, these calibration methods should be of general value to other image based algorithms.

Chapter 6

Performance of STM DFD Algorithm for Different Sensor Sub-sampling Schemes

6.1 Draft Modes for Depth-from-Defocus

Digital cameras use CCD or CMOS image sensors. CCD image sensors are used in digital compact and SLR cameras. In the case of CCD sensors, the sensed images are often subsampled for LCD display, autofocusing, gain control, white balancing, etc. Sub-sampling becomes necessary due to the limited bandwidth of the sensor readout circuitry. The full resolution capture rate is about 15-20 frames/sec. Sub-sampled draft mode readout is higher at 30-60 frames/sec. Since a fast autofocusing response is needed, particularly in tracking autofocusing of moving objects, it becomes necessary to use sub-sampled images for DFD. Evaluating the performance of DFD technique for various sub-sampling schemes found in digital cameras is of much interest to camera manufacturers.

The typical implementation for draft mode readout are line skipping and pixel mixture [58, 59, 60]. The line skipping mode sends one line over the neighbor n lines to the readout register. The number n usually adopts odd numbers such as 3, 5

or 7 to guarantee the continuation of Bayer pattern in the sub-sampling set. The pixel mixture mode mixes n neighboring pixels in the same color plane. The scene information is compressed to $1/n$, and the corresponding readout time can be save to $1/n$. The line direction could be either row or column direction, which is decided by the structure of image sensors.

Often subsampling is done at non-uniform pixel spacing. In such cases, an interpolation scheme is used to resample at uniform intervals. Experimental results show that typical subsampling schemes do not result in any significant degradation of performance. The BET and BM-OSOI techniques both give acceptable autofocusing RMS errors of 12 steps and 14 steps respectively out of a maximum of 1500 lens position steps.

In this section, seven different sets of draft readout modes were considered and compared for DFD performance. The different draft readout modes are named S1 and S2 for $1/7$ sampling rate, S3 to S5 for $2/7$ sampling rate, M4 (4×4) and M7 (7×7) for pixel mixture sampling . The details of implementations are summarized in Table 6.1 and Table 6.2.

- **Sampling Scheme S1**

Draft Mode Vertical $1/7$ Skip, Horizontal $1/7$ Skip in Pixel Map(PM).

In vertical lineskipping $1/7$ readout mode, one line/column is read out for each 7 consecutive lines. The sampling rate in vertical direction is not equal to the one in horizontal direction. Non-isotropic sampling will cause directional sensitivity in DFD autofocusing.

To obtain a uniform sampling, the simplest way is to also skip in the horizontal direction. This draft mode sampling is suitable for CMOS imager with random access ability. For the CCD imager, this mode does not use all the available data in the horizontal direction. As Fig. 6.1(b), the readout forms a new Bayer pattern. Instead

Draft Mode	Sampling Rate	Sampling Scheme Description
S1	1/7	(1) Vertical 1/7 Skip, Horizontal 1/7 Skip in Pixel Map (2) Convolution to get intensity map
S2	1/7	Vertical 1/7 Skip, Horizontal 1/7 Integration in Pixel Map
S3	2/7	(1) Vertical 1/7 Skip in 1 st Frame, (2) Set Offset=4, Vertical shift offset, Vertical 1/7 Skip in 2 nd Frame, (3) Then change to gray level image Img1 and Img2 (4) Intersection Img1 and Img2 in horizontal (5) Vertical uniform skip by offset 1:4:7
S4	2/7	(1) Vertical 1/7 Skip in 1 st Frame, (2) Set Offset=4, Vertical shift offset, Vertical 1/7 Skip in 2 nd Frame, (3) Horizontal 2/7 Integration using averaging and nearest neighbor borrowing to get corresponding B or R components
S5	2/7	(1) Vertical 1/7 Skip in 1 st Frame, (2) Set Offset=4, Vertical shift offset, Vertical 1/7 Skip in 2 nd Frame, (3) Horizontal 2/7 Integration using averaging and interpolation to get corresponding B or R components

Table 6.1: Sampling scheme summary (line skipping)

Draft Mode	Sampling Rate	Sampling Scheme Description
M4	1/4	(1) 4×4 integration in R, G_1 , G_2 , B channel (2) Convolution to get intensity map
M7	1/7	(1) 7×7 integration in R, G_1 , G_2 , B channel (2) Convolution to get intensity map

Table 6.2: Sampling scheme summary (pixel mixture)

of demosaicing or interpolating, a gray intensity is obtained from 2×2 local averaging. The resulting gray image is used in DFD algorithms.

- **Sampling Scheme S2**

Draft Mode Vertical 1/7 Skip, Horizontal 1/7 Integration in Pixel Map.

In S2 sampling scheme, a horizontal pixel is integrated to utilize all the available data. After vertical skipping and integration, a compound cell whose size is 7×7 can be formed. The repeating pattern is a 14×14 size tile, as noted by the yellow area in Fig. 6.2(b). Cells contain R/G information at odd rows and G/B information at even rows. To obtain the intensity, the missing B or R channel are interpolated by nearest neighbor strategy.

In the first and alternate rows, B is copied from the nearest neighbor. In the second and alternate rows, R is copied from the nearest neighbor. Then averaging between R, G and B color plane is used to create gray level images.

- **Sampling Scheme S3**

Interlace CCD sensor only captures half of the vertical information. The scan function breaks the integration period into two sequential field scans. Then the two

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
1	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
2	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
3	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
4	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
5	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
6	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
7	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
8	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
9	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
10	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
11	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
12	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
13	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
14	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
15	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
16	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
17	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
18	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
19	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
20	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G
21	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R	G	R
22	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G	B	G

(a)

	1(1)	2(8)	3(15)	4(22)
1(1)	R	G	R	G
2(8)	G	B	G	B
3(15)	R	G	R	G
4(22)	G	B	G	B

(b)

Figure 6.1: Pixel map of draft mode S1

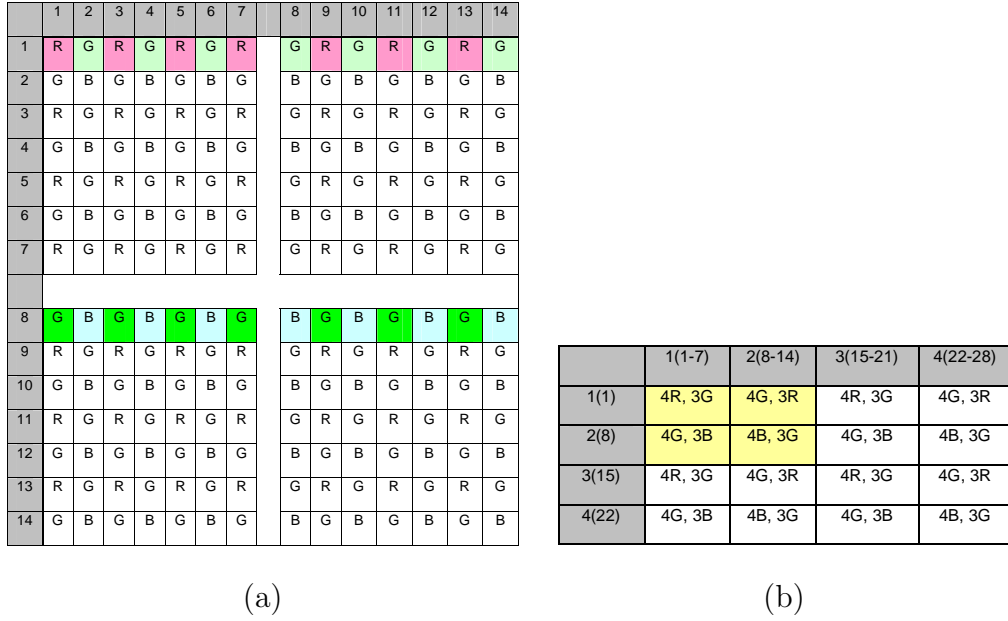


Figure 6.2: Pixel map of draft mode S2

scan periods are combined to create the whole, interlaced picture. This technique is widely adopted in TV display. It can generate higher spatial resolution with a limited data transform bandwidth.

The sampling scheme S3 is used for interlace CCD imager. In the first frame, a vertical 1/7 line skipping is conducted. In the next frame, a line offset is set as 4, then an additional line skipping is performed. As displayed in Fig. 6.8(b). The intensity images are calculated from the corresponding pixel map 1 and pixel map 2 by local averaging to avoid demosaic operation. The whole image is generated from interlacing two intensity images.

• **Sampling Scheme S4**

In the previous draft mode S3, a coupling happens between odd lines and even lines without demosaic. In sampling scheme S4, compound cells are integrated from 4 or 3 neighboring pixels alternatively in a horizontal scan. As displayed in Fig. 6.4,

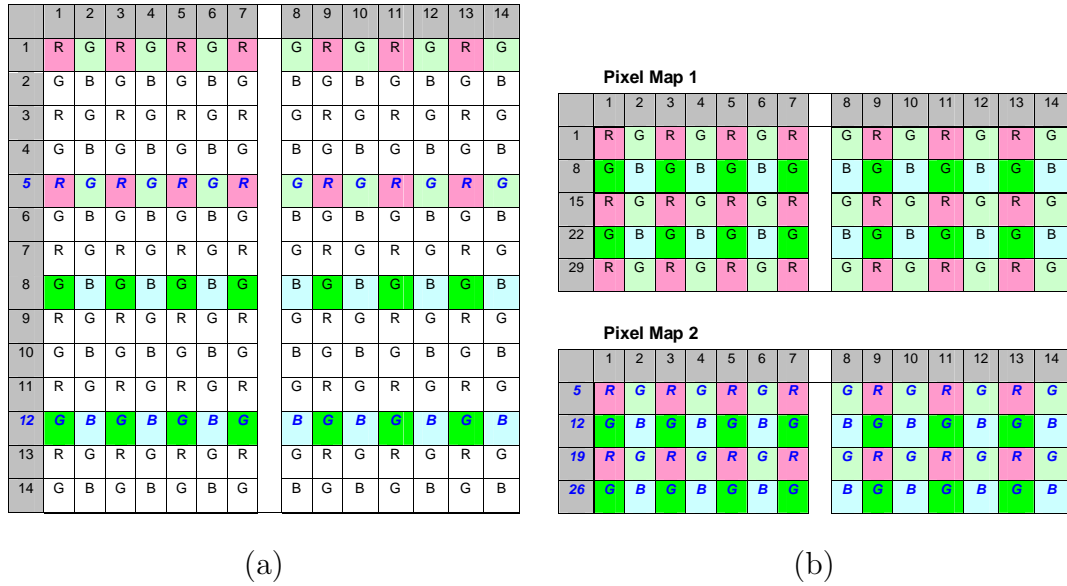


Figure 6.3: Pixel map of draft mode S3

a nearest neighbor borrowing can be used to get corresponding B or R components. The repeatable cell pattern size is 14×14 , as shown in the yellow area.

- **Sampling Scheme S5**

Nearest neighbor is a computationally efficient algorithm, but it may cause false edge. The compound cells are formed follow sampling scheme S4. As a counterpart, the missing B or R value can be linearly interpolated using vertical 3 components.

- **Sampling Scheme M4**

The pixel mixture draft mode is widely used for video recording on digital cameras and video recorders. In 4×4 pixel mixture M4 sampling scheme, each pixel value is mixed with neighboring 4 by 4 pixel arrays in the same color plane. Then the 2×2 local averaging is implemented to get a gray intensity.

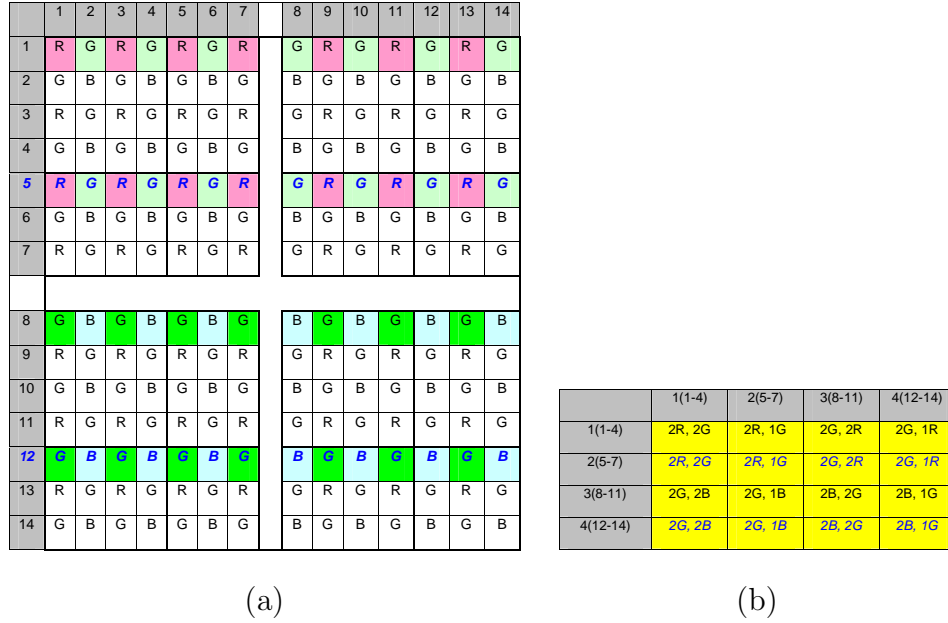


Figure 6.4: Pixel map of draft mode S4

- **Sampling Scheme M7**

In 7×7 M7 pixel mixture sampling, each pixel value is integrated with the neighboring 7 by 7 pixels, which reduce the cutoff frequency of low pass filter in the Fourier domain. And a gray intensity is obtained from 2×2 local averaging.

6.2 Experimental Results

In our experiment, a digital SLR-like Olympus camera is used to evaluate different draft mode readout. 9 objects of different levels of focusing difficulty (from (a) to (i) as shown in Fig. 6.5) were used in the experiments. 6 of the 9 objects are standard AF accuracy test charts (Ogata Charts) for digital cameras. Each object was placed at 6 different distances from 250 mm to 1750 mm, as listed in Table 6.3. At each distance, 2 images were captured at lens positions of step number 400 and 600.

Two STM1 methods were tested — BM-OSOI (Binary mask based, without squar-

ing, without integrating), and BET (Blur equalization technique). For each case, two types of results are reported as compared with the result of DFF — (i) mean lens step error and (ii) RMS (root-mean square) lens step error.

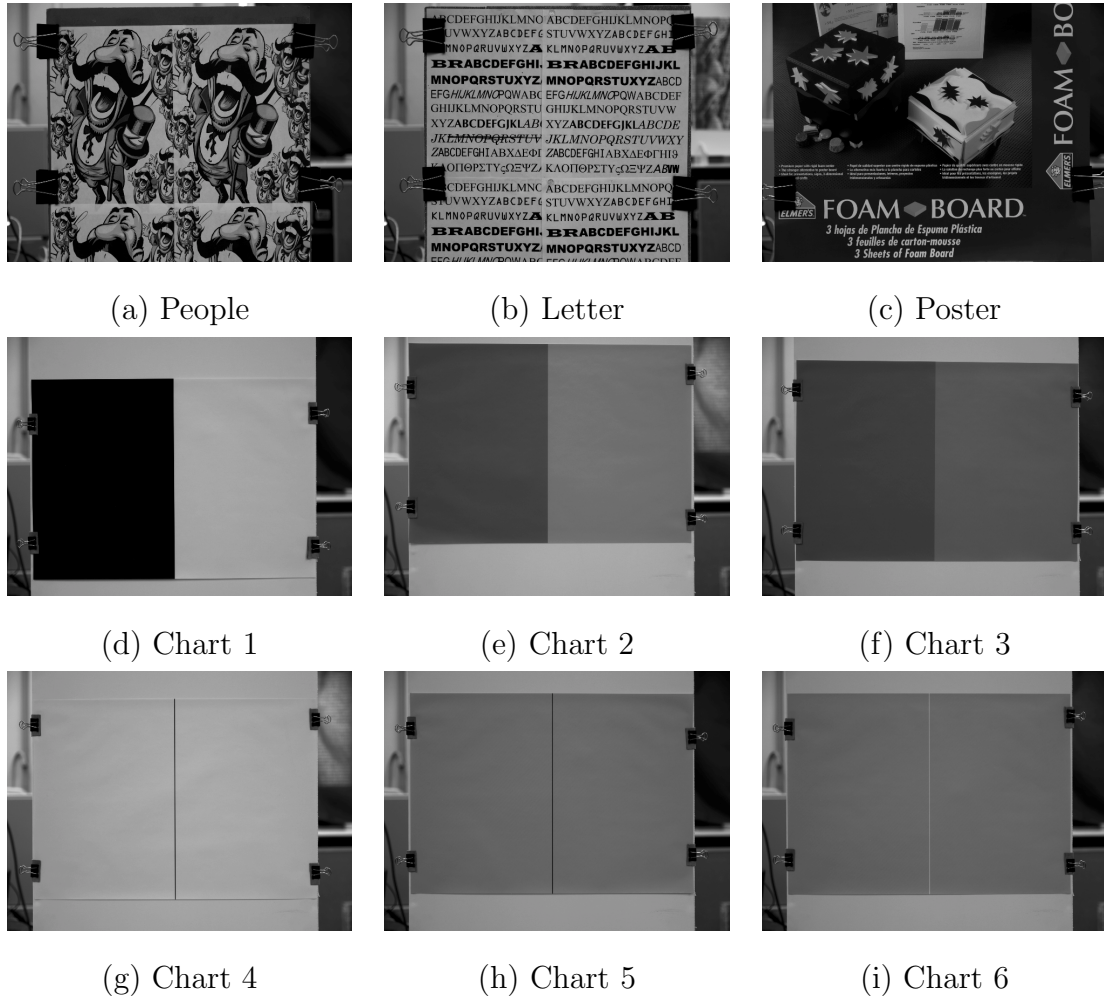


Figure 6.5: Test objects

Experimental results shows that DFD STM1 works reasonably well in both 1/7 line skipping (S1, S2), 2/7 draft mode (S3, S4, S5) and pixel mixture mode (M4, M7).

In 1/7 draft mode, as demonstrated in Fig. 6.6, the Mean/RMS errors are 13.2/17.3 lens steps for BET (Blur Equalization Technique), while the Mean/RMS errors are

Position	1	2	3	4	5	6
Distance [mm]	246.5	616.0	915.5	1195.2	1458.8	1745.1
Focus Steps [step]	754.67	440.33	347.67	290.00	258.67	231.00

Table 6.3: Test positions for draftmode readout

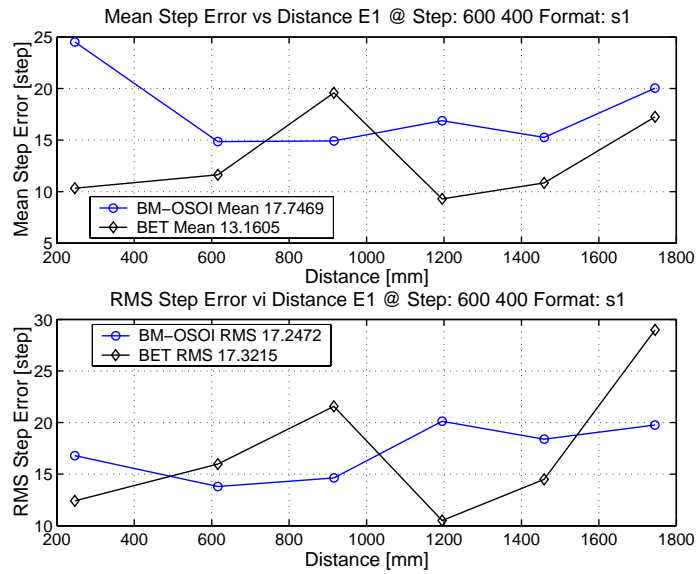


Figure 6.6: DFD performance on draft mode S1 (1/7 line skipping)

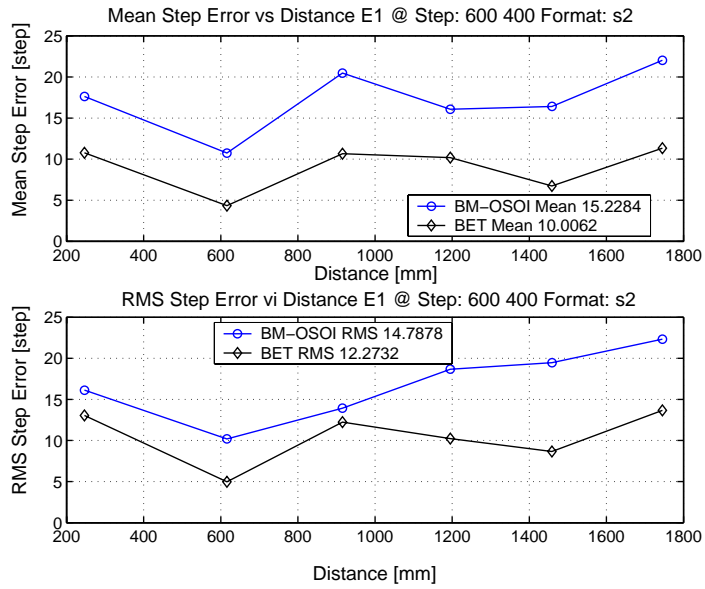


Figure 6.7: DFD performance on draft mode S2 (1/7 line skipping with pixel integration)

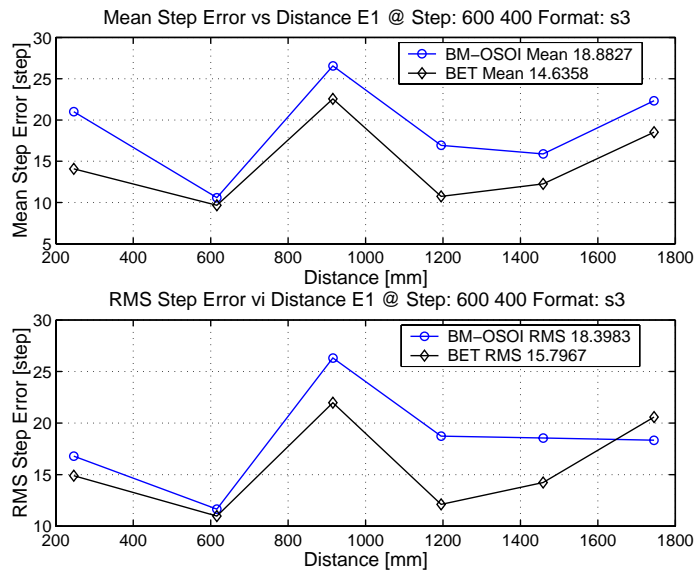


Figure 6.8: DFD performance on draft mode S3 (2/7 line skipping)

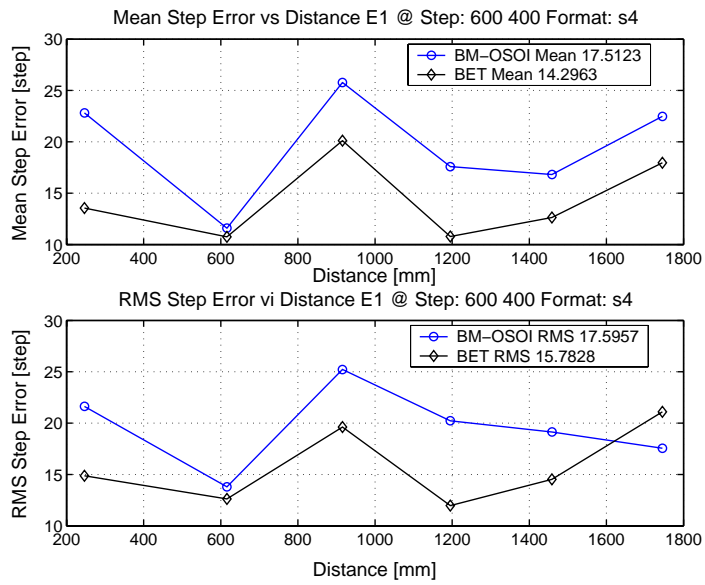


Figure 6.9: DFD performance on draft mode S4 (2/7 line skipping)

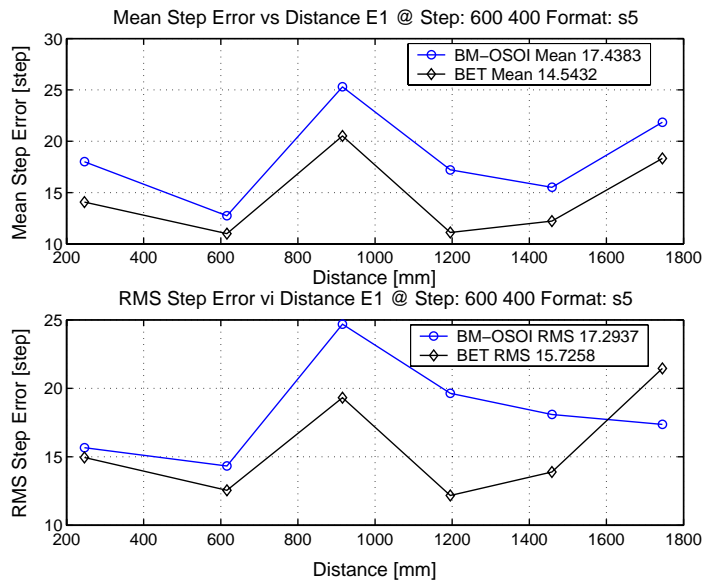


Figure 6.10: DFD performance on draft mode S5 (2/7 line skipping)

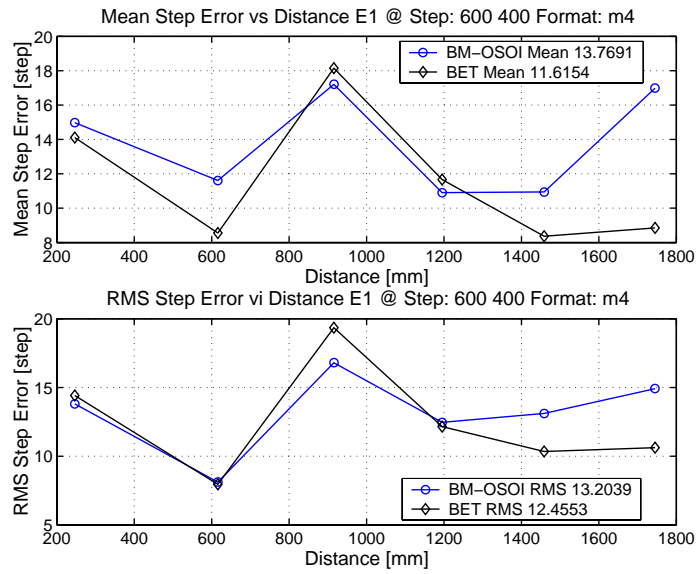


Figure 6.11: DFD performance on draft mode M4 (4×4 pixel mixture)

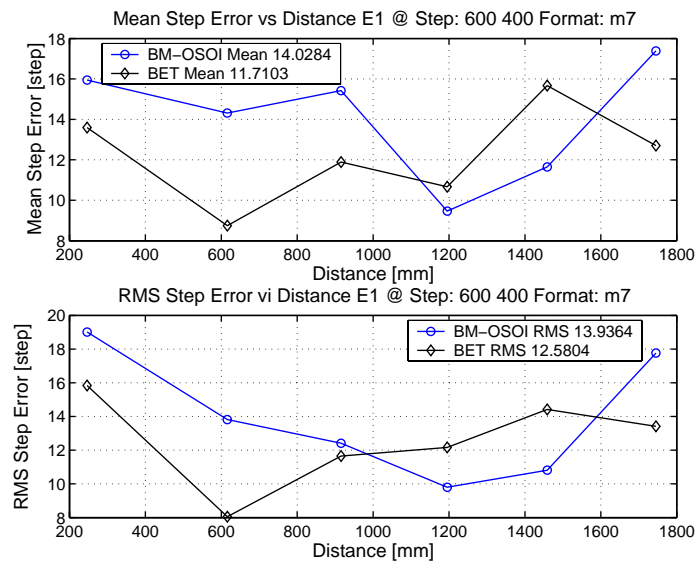


Figure 6.12: DFD performance on draft mode M7 (7×7 pixel mixture)

17.7/17.2 steps for BM-OSOI. BM-OSOI has a similar RMS error compared to BET.

For 1/7 draft mode S2, due to the integration between horizontal pixels, the performance of BET (Mean/RMS: 10.0/12.3 steps) and BM-OSOI (Mean/RMS: 15.2/14.8 steps) are improved compared to draft mode S1. The performance of BET is better than BM-OSOI. It is worthwhile to mention that the RMS error reduces to 0.82% (BET) and 0.99% (BM-OSOI) in the full step range.

In the three different 2/7 draft modes S3 to S5, the performance of DFD STM1 is similar. For BET, the Mean/RMS step errors for S3, S4, S5 are about 14/15 steps, and Mean/RMS step errors for BM-OSOI are around 17-18/17-18 steps. The measurement results are not benefit from a higher sampling rate. The unequal sub-sampling schedule and demosaic like interpolation may cause the frequency overlapping which causes a higher measurement error.

For the pixel mixture mode M4, the Mean/RMS errors for BET is 11.6/12.5 lens steps, and the Mean/RMS errors for BM-OSOI is 13.8/13.2 steps respectively. When the sampling rate decreases to 7×7 pixel mixture, the performance of BET is 11.7/12.6 lens steps. While the performance of BM-OSOI is 14.0/13.9 lens steps. From the experimental results, we observe that there is no significant change when the sub-sampling rate increases from M4 to M7. The reason is that both M4 and M7 inherently provide low pass filtering to effectively suppress the frequency overlapping due to sub-sampling effect.

Under the same 1/7 sub-sampling rate, S2 line skipping mode and M7 pixel mixture mode have similar performance as BET. However, the RMS measurement result of BM-OSOI improves from 14.8 lens steps (0.99% of full measurement range) to 13.9 steps (0.93% of full range). The performance of DFD algorithms on seven draft modes are summarized in Table 6.4.

By carefully tuning the DFD STM1 parameters, the performance of BM-OSOI and BET may be improved a little more. For example, in the current experiments,

the lens positions for image capture are 400 and 600. There may be better positions.

Experimental results show that an error of about 12 steps (BET) to 14 steps (BM-OSOI) out of 1500 steps is acceptable for use in actual camera systems.

Algorithm	Mean RMS Step Error in Draft Mode							
	S1		S2		S3		S4	
	Step	%	Step	%	Step	%	Step	%
BM_OSOI	17.24	1.15	14.79	0.99	18.39	1.23	17.60	1.17
BET	17.32	1.15	12.27	0.82	15.80	1.05	15.78	1.05
	S5		M4		M7			
	Step	%	Step	%	Step	%		
	BM_OSOI	17.29	1.15	13.20	0.88	13.94	0.93	
BET	15.72	1.05	12.46	0.83	12.58	0.84		

Table 6.4: DFD performance on different sampling modes

Chapter 7

Tracking Focusing using DFD

Continuous focusing of a moving object or Region Of Interest (ROI) is of much interest in digital cameras. The movement of objects could be an arbitrary combination of 3D translations and/or 3D rotations. For a digital camera, tracking focusing includes two stages. The first stage is to locate the correspondence between adjacent frames in 2D image space for a given ROI. In the second stage, an fast autofocusing algorithm is required for z-tracking or finding the lens position for best focusing in the ROI.

In this chapter, an adaptive feature-block based pyramid SSD matching is proposed for 2D tracking in the first stage. A new multiple base point beta calibration is introduced for Z or depth tracking. 3D DFD tracking focusing is realized by a combination of 2D tracking and Z tracking. A direct calibration technique based on lens design data is also studied. Initial results demonstrate that camera calibration can be done effectively using only the lens design data.

7.1 2D Tracking

Object tracking algorithms can be divided into two categories: feature based tracking [61, 62, 63] and correlation based tracking [64, 65]. Correlation based technique finds correspondence in a sequence of images which maximize a cross-correlation function. It is easy to implement and robust, but computationally expensive. With growing availability of computing power, some approaches have been investigated for fast matching by employing multi-resolution techniques [66]. Gaussian pyramids are used for generating multi-resolution images, and coarse-to-fine matching techniques reduce computation time [67].

The challenge of tracking focusing lies in that the movement of object involves not only X, Y and Z translations, but also 3D rotations such as roll, yaw, pitch. Object size and magnification change with object position and orientation. Further, processing speed should be fast enough to match the frame rate of digital cameras.

7.1.1 2D Tracking Algorithm

- **Feature Block Selection**

Given an image window that needs to be focused and tracked, a feature block within the window is selected. The entire focusing window is divided into many image blocks and one block with the highest gray-level variance is selected as a feature block with the highest contrast. The relative position of this block within the full focusing window is recorded. After selecting the feature block in the first image frame of a time/video sequence, in the subsequent frames, the best matching block for the selected feature block is found. The location of the best matching block is then used to determine the location of the full focusing window.

- **Multiple Resolution Registration**

The 2D Tracking algorithm is based on a multiple resolution pyramid matching technique. Matching is done by minimizing the Sum of Squared Difference (SSD) between image blocks f_1 and f_2 , i.e.

$$SSD[i][j] = \sum_k \sum_l (f_1[i][j] - f_2[i+k][j+l])^2 \quad (7.1)$$

A flow chart of the algorithm is shown in Fig. 7.1. At the beginning of 2D tracking, an image block with the highest variance or contrast is selected from current AF window (frame 0). The searching space in the next frame $i + 1$ is defined by the block location (x_i, y_i) and a searching range parameter. Then, after down-sampling, the coarse registration step calculates the approximate block location (x_{i1}, y_{i1}) by SSD. The down sampling ratios are typically powers of 2 or 4. The coarse block position in frame $i + 1$ is used as a starting point for registration refinement in the original resolution image. Using SSD based matching, the refined block position in frame $i + 1$ is calculated, and the AF window in frame $i + 1$ is located.

One of the algorithms involves matching at three resolutions, first at 1/4-th of the original resolution, then 1/2 of full resolution, and finally the original full resolution. The matching at 1/4 resolution has less computation but also less accuracy. The accuracy is improved by going to the 1/2 resolution and re-matching in a small region around the estimated resolution. This accuracy is further improved by re-matching in the original resolution images. Some minor variations of this basic algorithm are possible. For example, we may consider image resolutions of 1/16, 1/4 and full original resolution instead of 1/4, 1/2 and full resolution. Also, the computation can be further reduced by predicting the motion of an image block by extrapolation of the locus of motion of the image block. In general XY tracking works quite well giving fast and robust results under limited X, Y and Z translations and limited rotations

about X,Y and Z axes.

- **Adaptive Target Block**

One of the challenges of tracking focusing is the possibility of simultaneous 3D translations and rotations. The image of an object can translate, rotate, change scale, and undergo other shape and contrast deformations, including partial occlusion. In the absence of a priori information on 3D motion parameters of the object, it becomes necessary to use an adaptive image block for tracking instead of the fixed original image block.

In the continuous shooting of a digital camera, the relative motion between one frame to the next frame is relatively small. The 3D translation and rotation is an accumulative result over multiple frames. Adaptive feature block tracking is based on this assumption. In order to continuously track an image block from frame i to $i + 1$, and then from $i + 1$ to frame $i + 2$, the best matching block in frame $i + 1$ is used as the new image block which needs to be matched in frame $i + 2$. The original image block in frame i is not used for matching blocks in frame $i + 2$. The search range in frame $i + 2$ can be estimated from previous block positions in frames $(i - 1, i, i + 1, \dots)$. Thus 2D tracking can be made quite robust for XY translation, Z translation, axis rotation (X, Y and Z) and combinations of translation/rotation.

7.1.2 2D Tracking Results

To evaluate the performance of 2D tracking, a compact digital camera is used to record movies. The image size of the movie is 320×240 provided in the draft mode readout. A 400-frame movie is recorded for test purpose. In this image sequence, a piece of paper with printed letter is randomly moved around. The motion effects includes rotation, translation, and non-rigid movements simultaneously. The tracking

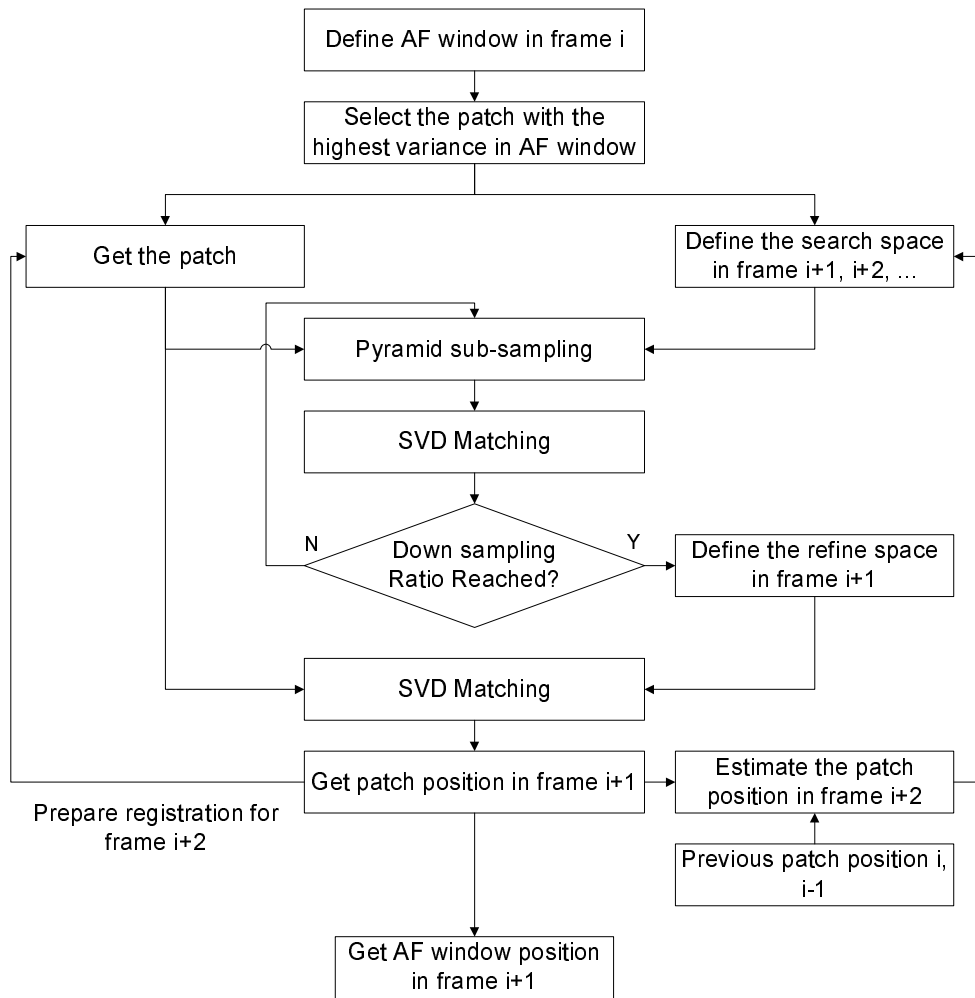


Figure 7.1: Flowchart of XY 2D tracking

result is shown in Fig. 7.2. The 2D tracking is quite robust for XY translation, Z translation, axis rotation (X, Y and Z) and also combinations of translation/rotation. Due to the change of viewpoints, sometimes there exist significant distortions. Even in these situations, the 2D tracking algorithm could still locate the focusing window around the correct position.

The parameters in 2D tracking can be adjusted for robustness and computation speed. In our experiments, a significant portion of the time was consumed by data transfer from the camera to the computer. The image data was transferred through a USB 1.1 interface. In Table 7.1, time (which includes data transfer) for three different 2D tracking variations are compared. The images are processed using a Pentium 4 3.0 GHz Personal computer. The table demonstrates the advantage of multi-resolution image matching.

Scheme		Processing Time [ms/frame]			
		1/2+Full	1/4+1/2+	Full	1/3+Full
Patch Num.	4	31.15	25.43		24.64
Patch Num.	6	13.91	11.84		11.89

Table 7.1: 2D window registration time comparison

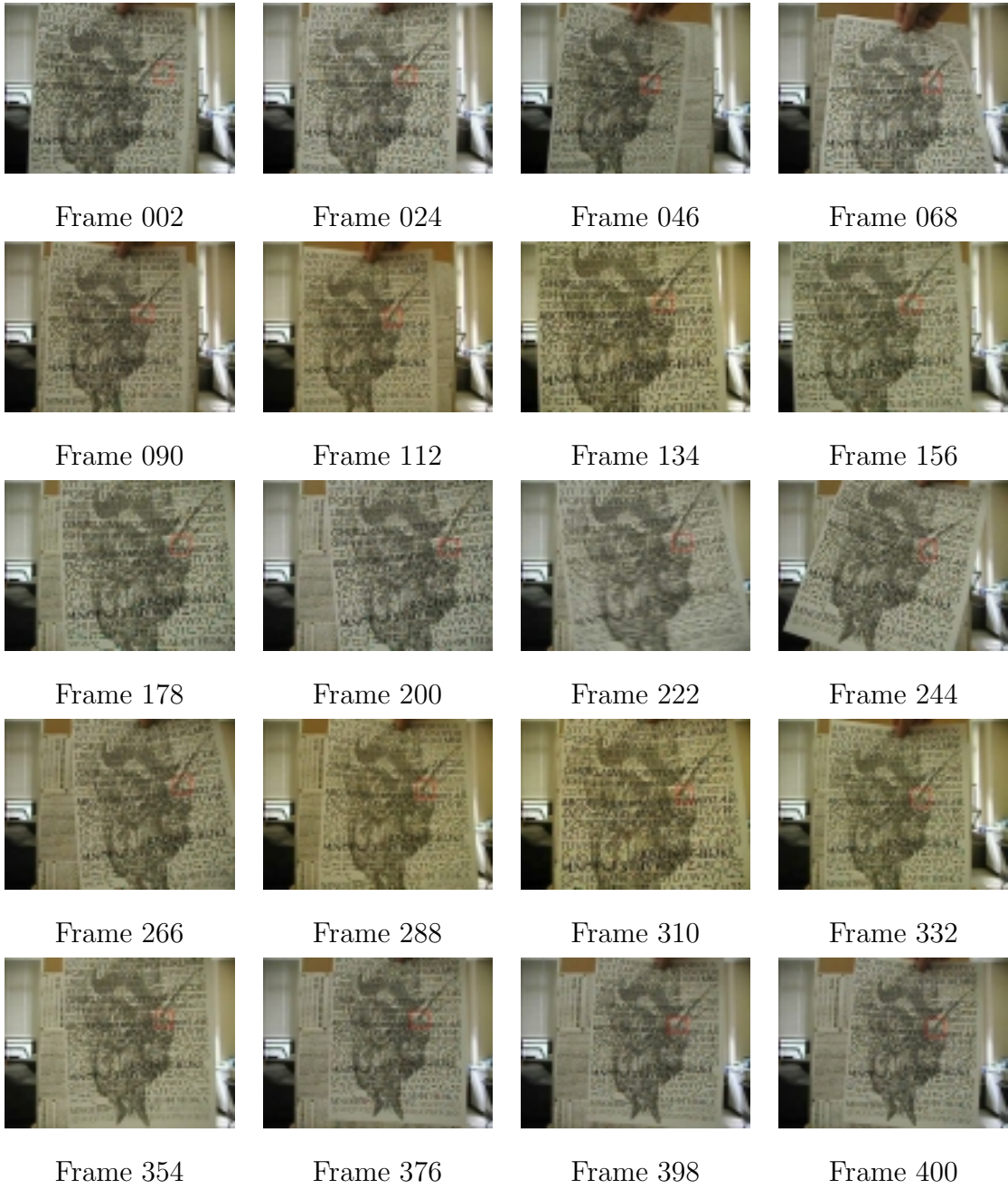


Figure 7.2: Tracking example

7.2 Z Tracking

Object movement along the optical axis (Z direction) changes the lens position of best focus. To continuously capture the focused image, the lens position needs to track the Z-movement of the focused image of the object. A straightforward way of Z-tracking is to directly extend DFD to continuous focusing. As shown in the flow chart of Fig. 7.3, two defocused images $g_1(t_i)$ and $g_2(t_i)$ are captured at two fixed steps $Step_1, Step_2$. Then the focused image $g_f(t_i)$ is obtained. However the performance of this direct approach is not satisfactory as it needs to capture 3 images– two defocused and one focused. A scheme for capturing only two images will be considered next.

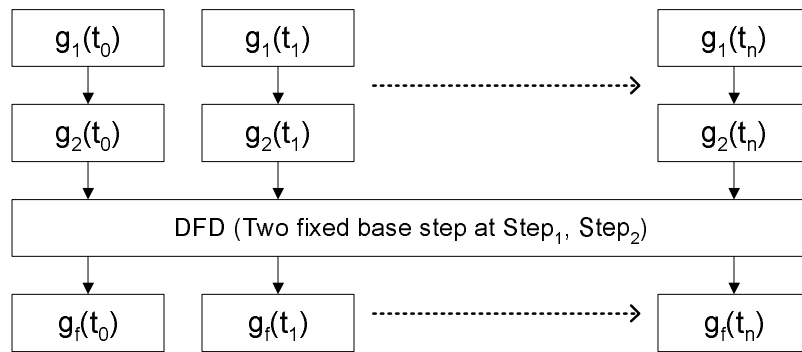


Figure 7.3: STM continuous focusing

Wei and Subbarao [68] proposed a Fourier domain approach for focusing of moving objects. They reported a focusing accuracy about 4.3%. In the spatial domain, Continuous Spatial-domain Transform Method (CSTM) was proposed by Surya and Subbarao [69]. There are two variations - CSTM1 and CSTM2. CSTM1 involves multiple calibrations for different lens position pairs of the camera. In CSTM2 the camera is calibrated only for one lens position pair, and the data for other lens positions are approximated through an interpolation scheme. In the next section we propose an improved scheme for obtaining accurate calibration data through the so

called Beta calibration technique.

7.2.1 Beta Calibration

In Z tracking based on DFD, the main problem is to keep the number of images captured and processed to a minimum of only two images. Park and Subbarao [70] proposed a beta calibration technique to reduce the required number of images to two. In this technique, the first image is focused, as shown in the flow chart of Fig 7.4.

From Eqn. (3.16), G can be expressed by:

$$G = \sigma_1^2 - \sigma_2^2 = \frac{4(g_1 - g_2)}{\nabla^2 g} \quad (7.2)$$

When either image is fully focused, one of σ_i , $i = 1, 2$ equals zero. Eqn. (7.2) can be rewritten as:

$$G = \begin{cases} -\sigma_2^2, & \sigma_1 = 0 \\ \sigma_1^2, & \sigma_2 = 0 \end{cases} \quad (7.3)$$

When a calibration object is placed at a fixed distance, the corresponding lens position for best focus is $Step_f$ in camera. If the lens position changes continuously from near field to far field of the camera, the difference of σ with respect to the clear/focused image in $Step_f$ corresponds to the parameter beta in Eqn. (3.11). It can be calculated from Eqn. (7.2).

In continuous focusing, two defocused images $g_1(t_0)$ and $g_2(t_0)$ are captured at time t_0 . If $g_2(t_0)$ captured at step number n is focused, a lookup table links the step number n and $G(n(t))$. Suppose the object moves and two images $g_1(t_1)$ and $g_2(t_1)$ are obtained at the same steps, then $\sigma_2(t_1)$ will change corresponding to focus step $n(t+1)$. By assuming that σ_2 is locally linear, we can estimate the next focus position as $n(t+1) = G^{-1}(t_1)$.

In the experiment, a G vs $Step$ lookup table is created and used for interpolation. The base step 65 is selected approximately around the middle of the step range. However, as shown in Fig. 7.5, the beta calibration has a relatively flat slope in the middle field of the camera. Moreover, a large step RMS error will occur for distances whose focus step is around the flat slope region. The comparative experimental results are shown in Fig. 7.9. It will be discussed in the next subsection where a new beta calibration technique is introduced.

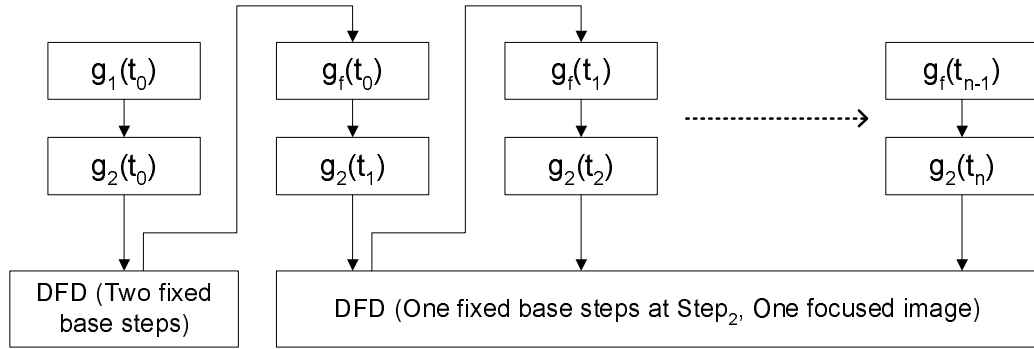


Figure 7.4: Beta continuous focusing

7.2.2 New Beta Calibration Technique

A new multiple base point beta calibration technique is presented here. From Chapter 3, if two images are recorded with different parameter settings $\vec{e}_1 = (s_1, f_1, D_1)$ and $\vec{e}_2 = (s_2, f_2, D_2)$, we have

$$\sigma_i = \frac{D_i s_i}{2k} \left(\frac{1}{f_i} - \frac{1}{u_i} - \frac{1}{s_i} \right) \quad (7.4)$$

where D_i is the diameter of aperture, f_i is the focal length and s_i is the image distance. k is a camera PSF constant (approximate value is $\sqrt{2}$).

When lens positions s_1 and s_2 change, image magnification also varies. A normalization magnification operation is needed. This normalization is with respect to the

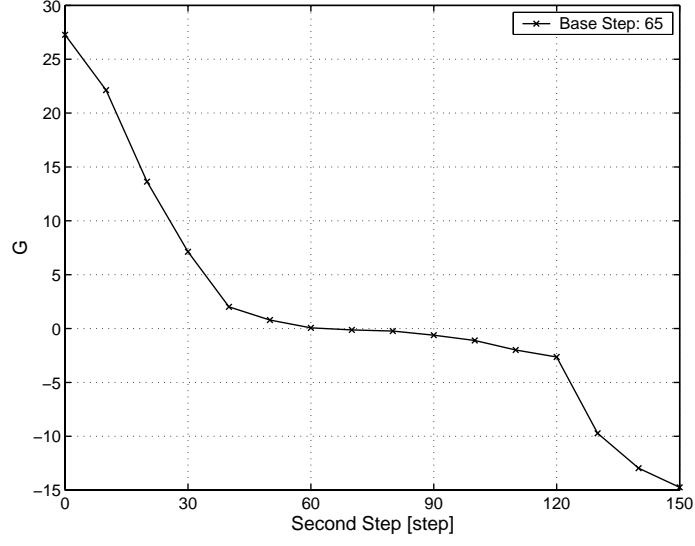


Figure 7.5: G calibration table

base step, as suggested in [70].

$$\frac{D_1 s_1}{D_2 s_2} \sigma_2 = \frac{D_1 s_1}{2k} \left(\frac{1}{f_2} - \frac{1}{u_2} - \frac{1}{s_2} \right) \quad (7.5)$$

For the same object distance under different parameter settings, $u_1 = u_2$. Therefore

$$\sigma_1 = \frac{s_1}{s_2} \sigma_2 + \frac{D_1 s_1}{2k} \left(\frac{1}{f_1} - \frac{1}{f_2} - \frac{1}{s_1} + \frac{1}{s_2} \right) \quad (7.6)$$

$$\sigma_1 = \alpha \sigma_2 + \beta \quad (7.7)$$

where

$$\alpha = \frac{D_1 s_1}{D_2 s_2} \quad (7.8)$$

$$\beta = \frac{D_1 s_1}{2k} \left(\frac{1}{s_2} - \frac{1}{s_1} \right) + \frac{D_1 s_1}{2k} \left(\frac{1}{f_1} - \frac{1}{f_2} \right) \quad (7.9)$$

When $Step_2$ is selected as the base step, and the object distance $u = u_1 = u_2$ is chosen as the distance that corresponds to the base step $Step_2$, the image taken at

$Step_2$ will be a focused image. Then Eqn. (7.7) can be rewritten as:

$$\sigma_1 = \beta \quad (7.10)$$

By changing $Step_1$, a β vs. Step calibration Table can be obtained. For improving the linearity of calibration curve, a new β to as a function of the reciprocal of object distance $1/u$ is computed.

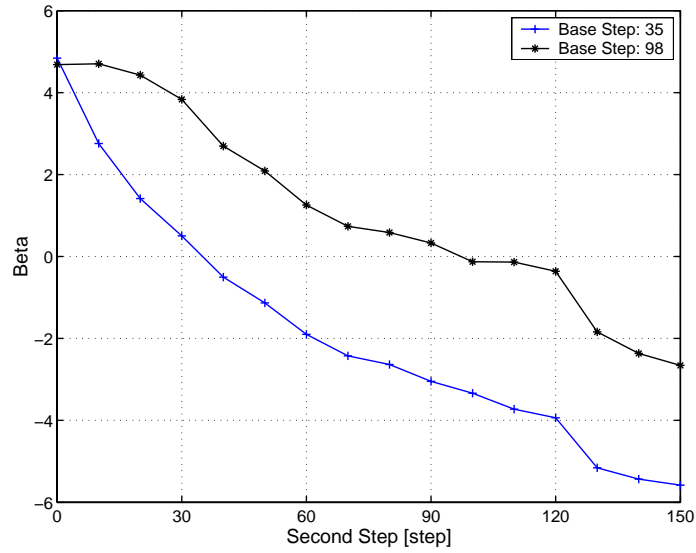


Figure 7.6: Beta calibration table: Beta vs. Step

The two fixed steps (Step 35 and step 98) are used as base steps. A switching mechanism is used for selecting one of these base steps depending on the current focused step position. The scheme for selection is as below.

Require: load Beta Calibration 1 for $Step_1$

Require: load Beta Calibration 2 for $Step_2$

if $Current_Step \geq Threshold_1$ **then**

$Base_Step \leftarrow Step_2$

else if $|Current_Step - Step_1| > Threshold_2$ **then**

$Base_Step \leftarrow Step_1$

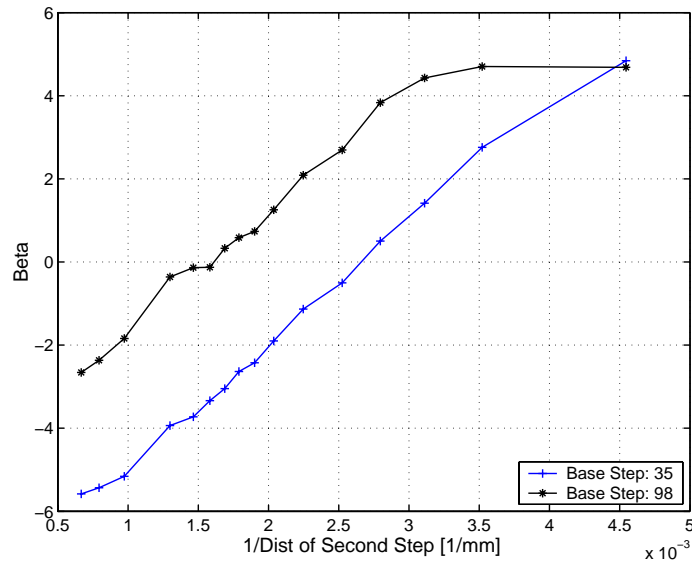


Figure 7.7: Beta calibration table: Beta vs. $1/u$

else

$Base_Step \leftarrow Step_2$

end if

do Z Tracking using $Current_Step$ & $Base_Step$

update $Current_Step$

Experimental results of tracking are shown in Fig. 7.8. The corresponding RMS step error is shown in Fig. 7.9. In this Figure we see that the RMS Lens step error is greatly improved at positions 5, 6 and 7. For example, the RMS lens error reduced from approximately 20 steps to 3 steps at position 5. Compared to the old beta calibration, the new beta calibration significantly improves the overall RMS step error from 9.43 to 3.36 lens steps.

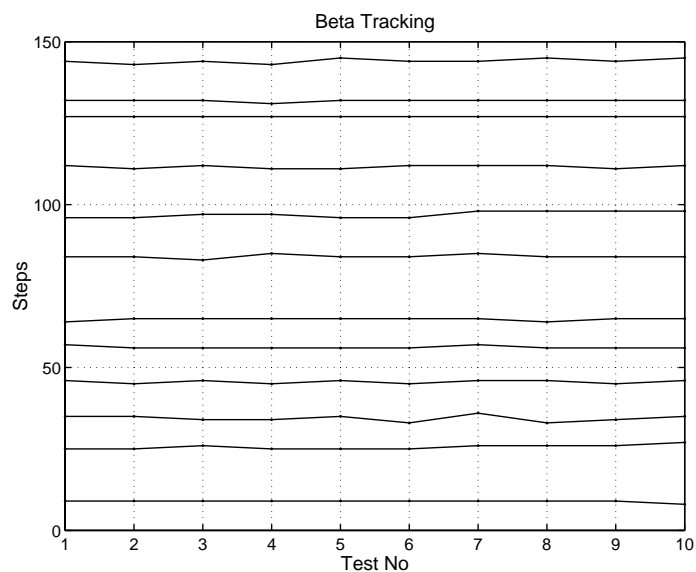


Figure 7.8: Beta tracking results

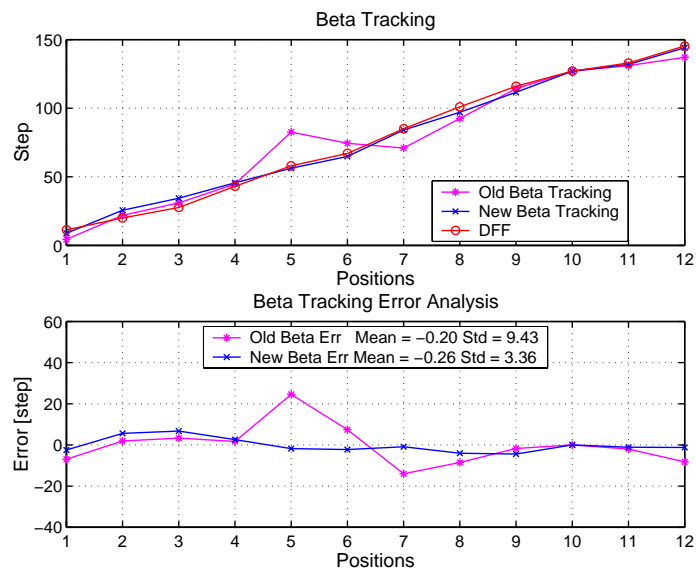


Figure 7.9: Beta tracking error

7.3 Calibration From Lens Design Data

DFD calibration links the blur parameter sigma to the object distance u or lens position in steps. The procedure requires changing the object distance to cover the full range of measurement. Moreover, at each object distance, the calibration for continuous tracking focusing also needs to be measured for many different pairs of lens step positions. DFD calibration is a time-consuming procedure.

An alternative to extensive calibration work is to generate DFD calibration table from lens information. There are two approaches as shown in Fig. 7.10. The first approach generates the calibration table from geometric optics. It involves less computation, however there exist some error in the near focus positions; The second approach use wave optics to compute the sigma look-up table. This approach needs more computation, and are able to obtain more accurate results.

A computer simulation is conducted to compare the sigma calculated from geometry optics and wave optics. From wave optics [71, 72], the wavefront on the exit pupil of the optical system with defocus aberration is a spherical wave, and its phase is plotted in Fig. 7.11 (a). The corresponding Point Spread Function (PSF) in spatial domain and Modulation Transfer Function (MTF) in spatial frequency domain are displayed in Fig. 7.11 (b) and (c) respectively. The change of sigma with respect to object distance u , inverse of object distance $1/u$ and lens step are calculated and shown in Fig. 7.12 (a), (b) and (c). Fig. 7.12 (d) is a partially enlarged plot from (c), it shows the error of geometry optics approach at the near focus position.

We used the PSF data obtained from the optical design software to compute the sigma vs. $1/u$ table. In order to make sure that the sigma vs. $1/u$ table computed by this method will be the same as that obtained using the calibration method using real images, experimental verification has been investigated. In the experiments, a digital SLR-like camera's PSF data provided by Olympus is used. These PSF data are

generated by the lens design program. The design PSF data are plotted in Fig. 7.13 and Fig. 7.14.

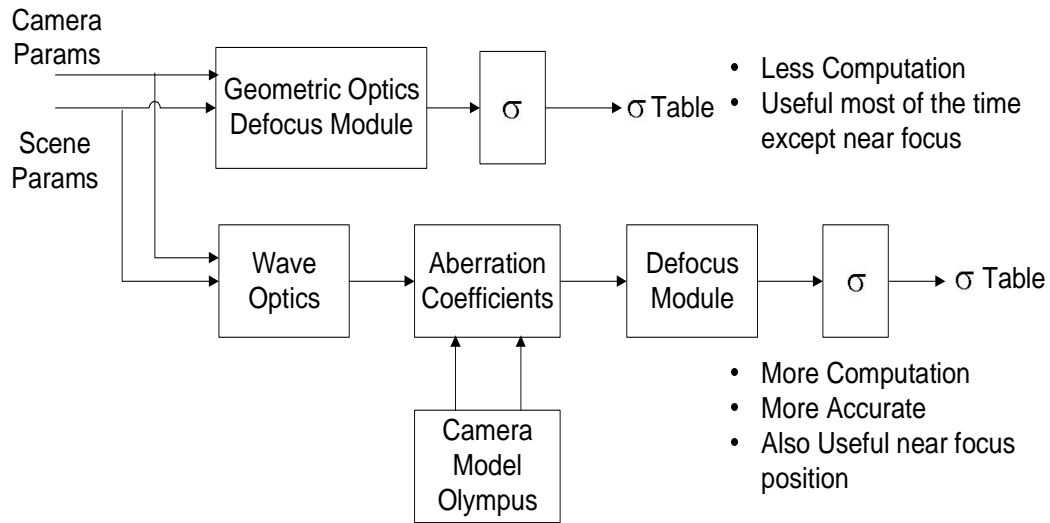
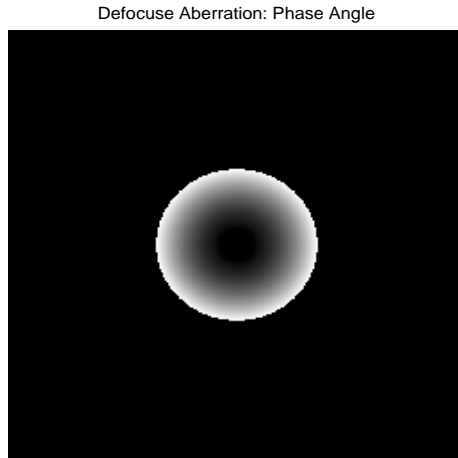


Figure 7.10: Calibration table generation

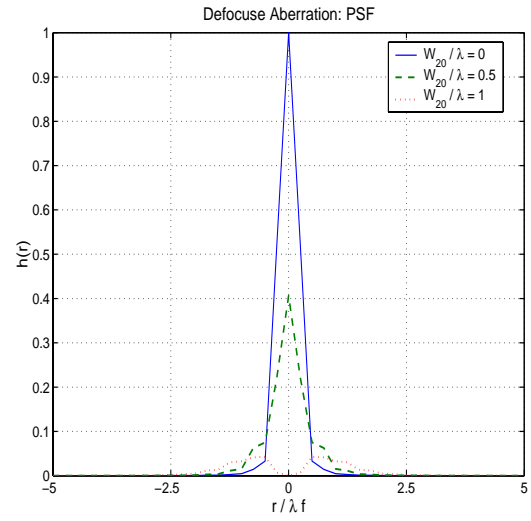
The PSF data is for lens step positions 500 and 600 (note that 400 and 600 would have been better according to previously described experiments). The data was given for many different object distances. This PSF data was used to compute the sigma vs. $1/u$ table. Then we took the actual camera images provided by Olympus (TeleA, TeleB, TeleC, TeleE, TeleF) and computed the sigma vs. $1/u$ table. The results are shown together in the Fig. 7.18 and Fig. 7.19 for comparison.

In the figures, the sigma vs. $1/u$ curves of the two different methods match well in the middle range but not so well in the region near the end points. This is as expected because, near the end points, the real images are blurred too much and therefore the estimated sigma values are noisy. Note that, when the images are blurred too much, the Laplacian of computed images are noisy and therefore the estimated sigmas are noisy.

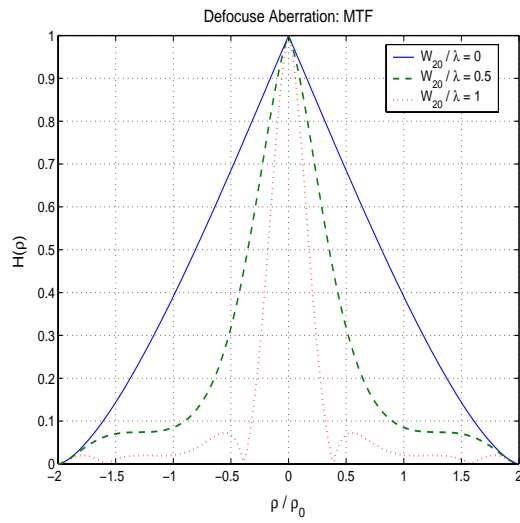
Also the sigma plot computed from the PSF data is remarkably linear (i.e. a



(a) Phase angle

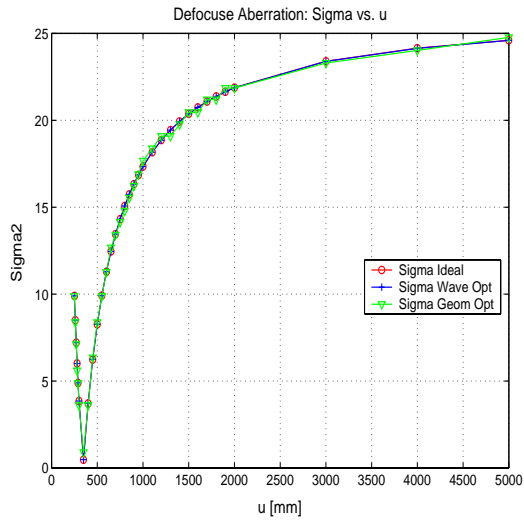


(b) PSF

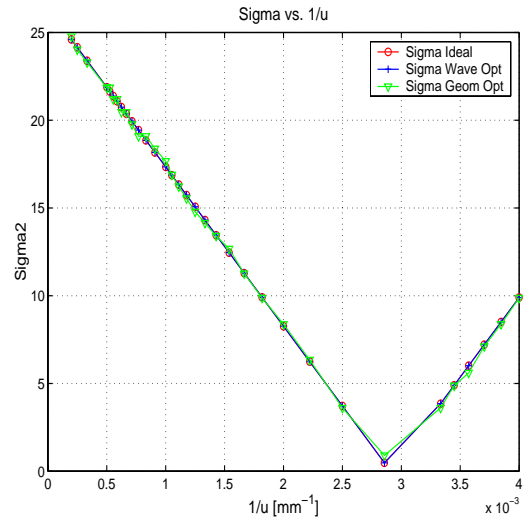


(c) MTF

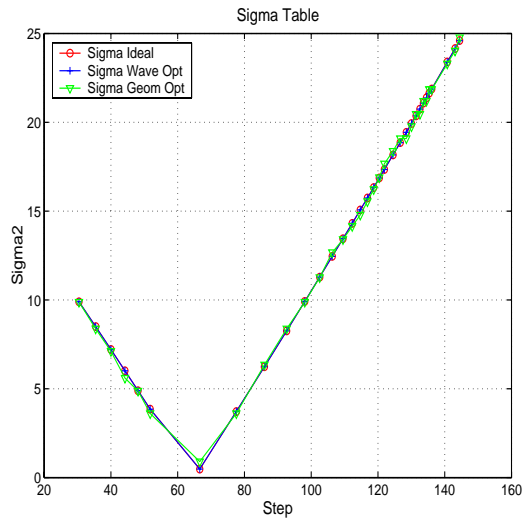
Figure 7.11: Defocus aberration



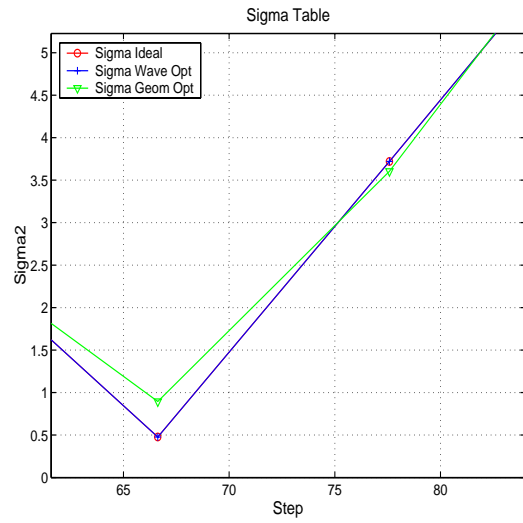
(a) Sigma vs. u



(b) Sigma vs. 1/u

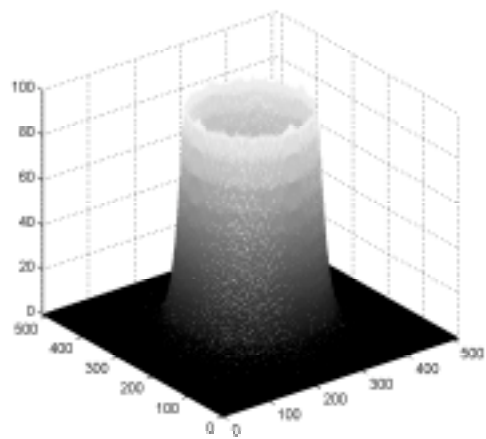


(c) Sigma vs. step

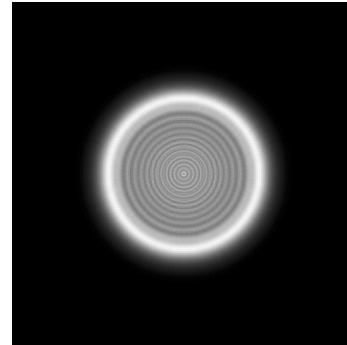


(d) Partially enlarged sigma vs. step

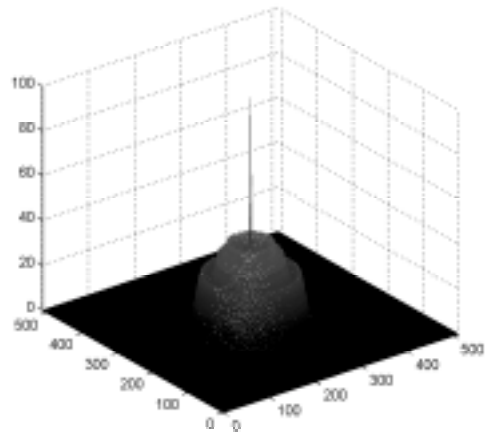
Figure 7.12: Calibration table from simulation



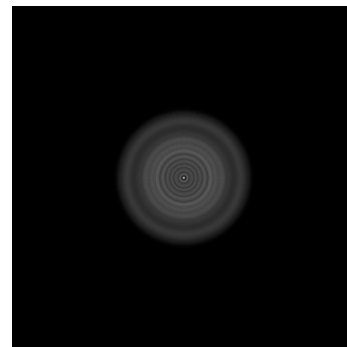
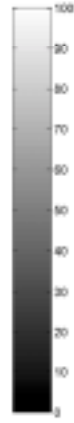
Focus distance = 500 mm



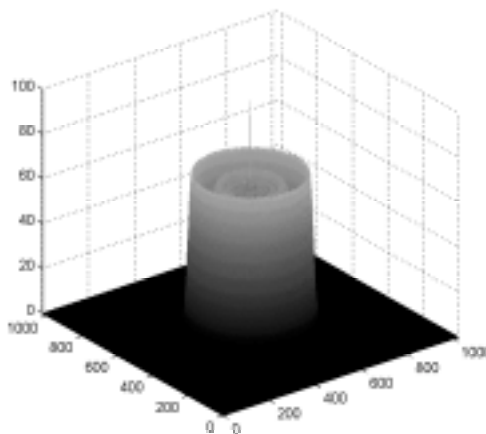
Object distance = 400 mm



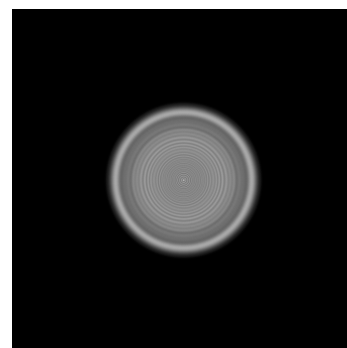
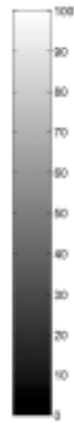
Focus distance = 500 mm



Object distance = 600 mm

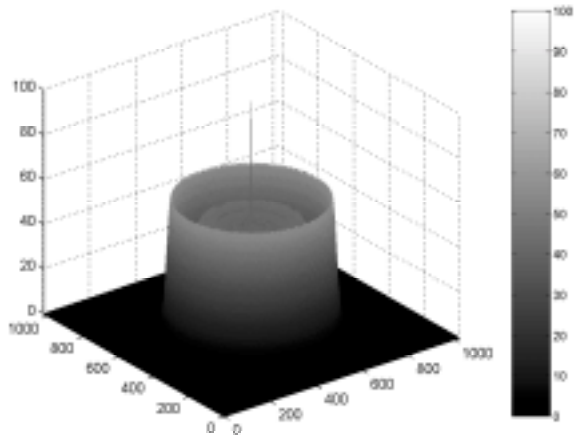


Focus distance = 500 mm

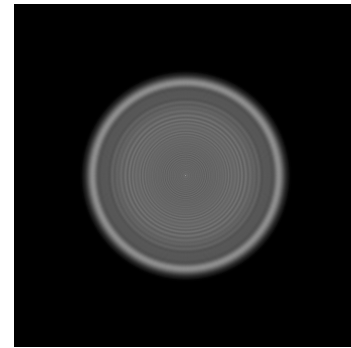


Object distance = 800 mm

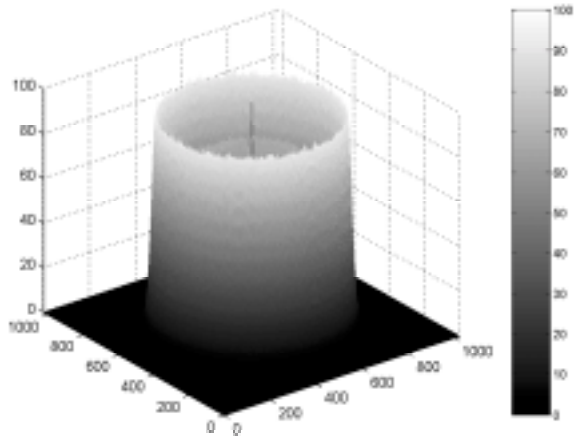
Figure 7.13: PSF changes with object distance (1)



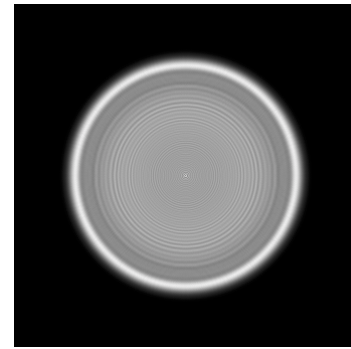
Focus distance = 500 mm



Object distance = 1000 mm



Focus distance = 500 mm



Object distance = 1200 mm

Figure 7.14: PSF changes with object distance (2)

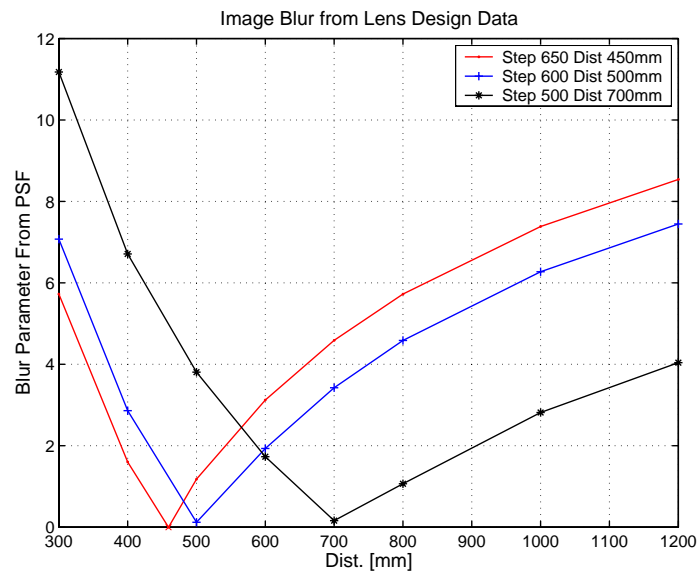


Figure 7.15: Sigma vs. object distance

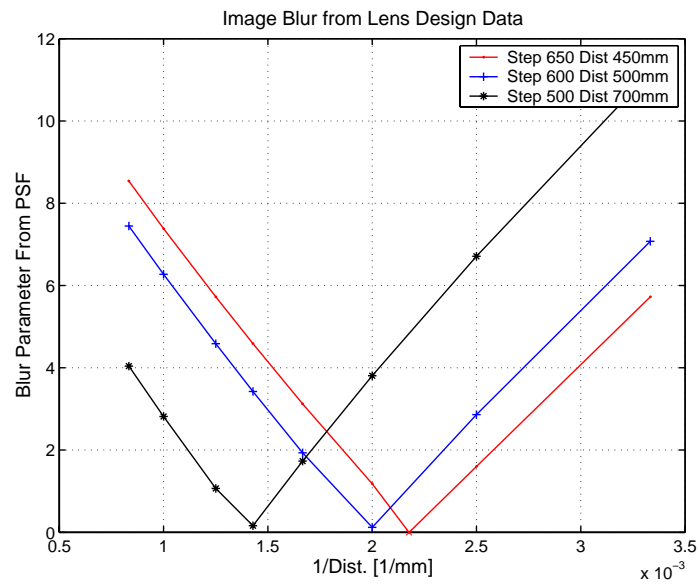


Figure 7.16: Sigma vs. inverted object distance

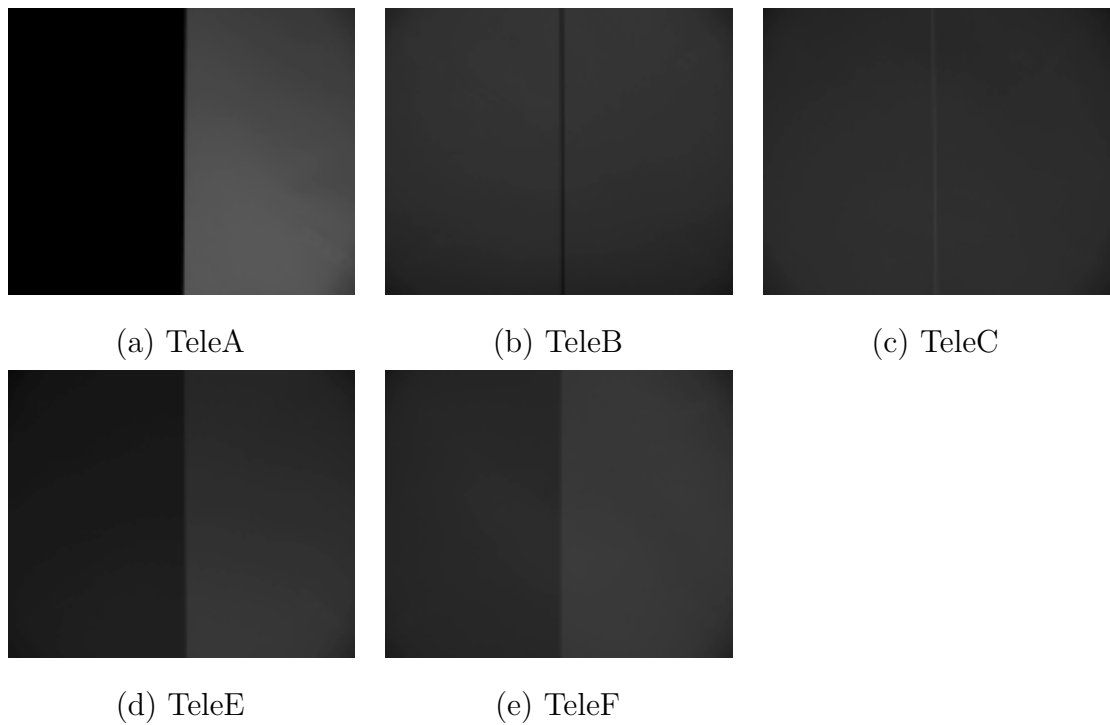


Figure 7.17: Test objects for direct calibration from lens design data

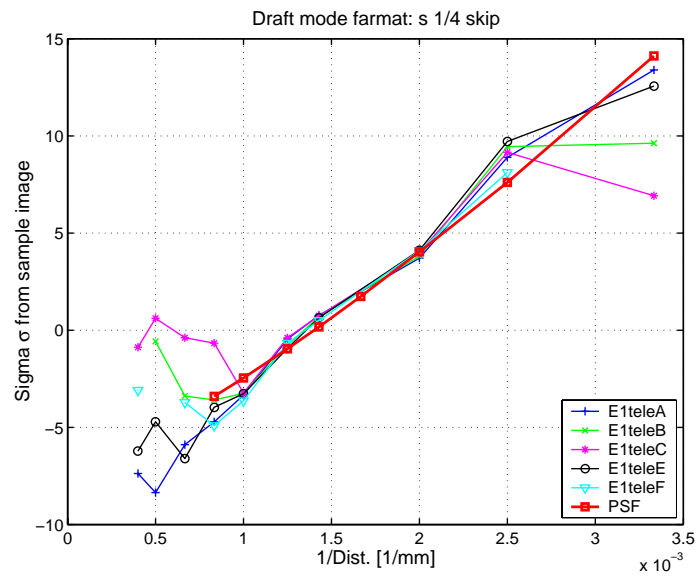


Figure 7.18: Sigma table calculation from PSF data and actual images (draft mode: S)

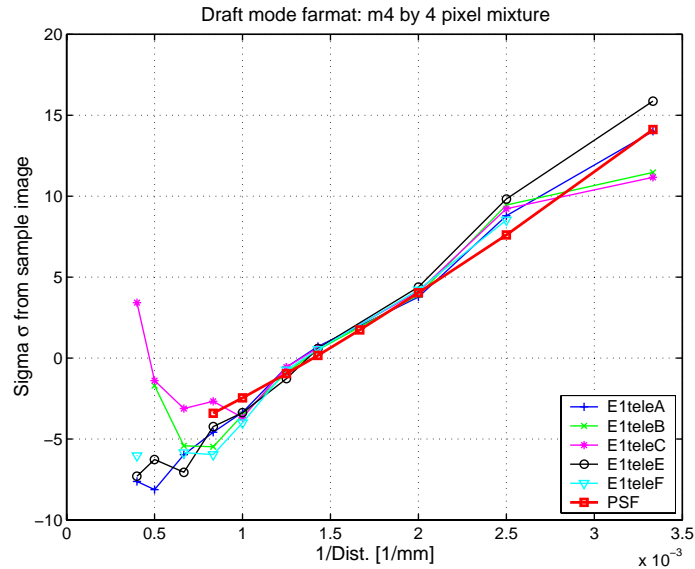


Figure 7.19: Sigma table calculation from PSF data and actual images (draft mode: M4)

straight line) with respect to $1/u$. This shows that interpolation and extrapolation of this data will give very accurate results for sigma when the design data is not available for a particular distance u . This reduces the design PSF data that we need in actual implementation of DFD-AF.

The most important result of the above experiments is that we have verified that, whenever we process images which are blurred moderately or less (i.e. the blur is not too much), then the sigma computed from the real images matches well with the sigma computed from the PSF data provided by the lens design program. In actual DFD-AF, the sigma computed from actual images will be used to look up the sigma-table computed from the PSF data to find the object distance (or lens focus position). Therefore, the calibration data obtained from the PSF data is accurate and sufficient for practical applications.

In Z tracking, the two images captured are blurred only moderately or less because

the Z motion from one frame to the next is not too much (less than 200 lens steps). For example, let the focusing position of the previous frame be step 700 and the focusing position of the current frame be step 800. The image for step position 700 will already be known, and a second image which is always a fixed 200 steps more than the current position ($700+200=900$ in this example) can be captured. Then, we can use the dynamically computed calibration table for steps 700 and 900 in DFD-AF to find that the focusing position is 800. In this case, the images will be blurred only moderately or less (corresponding to only 100 step lens displacement).

In conclusion, we have experimentally verified that the sigma vs. $1/\text{distance}$ calibration table computed from real images and from the PSF data provided by the lens program are almost the same.

The sigma vs. $1/u$ plot computed from the PSF design data is linear. Therefore interpolation and extrapolation can be used to compute the sigma vs. $1/u$ table to a degree of high accuracy for those distances u at which the design PSF data is not available. This greatly reduces the amount of design data needed for practical implementation of DFD based autofocusing. Also, it facilitates dynamic computation of sigma vs. $1/u$ table for any u during Z-tracking.

DFD will work as well for Z tracking as it does for still (non-moving) objects because the required calibration table can be computed dynamically using the PSF data. For non-moving objects, earlier experiments on DFD-AF have given an error of about 15 lens steps. Therefore, for moving objects with Z-tracking and using the PSF design data, the error should be again around 15 lens steps (out of a maximum of 1500 possible steps).

7.4 Dynamic DFD

In this dissertation, research efforts on DFD based autofocusing are concentrated on improving the accuracy and robustness of DFD algorithms. Various STM and BET algorithms are presented to improve the accuracy of DFD. Different factors that affect the performance and robustness of DFD algorithms are calibrated and compensated. From above results, the critical factor for DFD algorithm is SNR.

In order to further improving DFD autofocusing, a novel concept – Dynamic DFD is proposed for further research. In the dynamic DFD, the sigma table can be dynamically generated for different lens positions, and the selection of lens position always keeps DFD AF in high SNR region, as shown in Fig. 7.18 and Fig. 7.19.

The initial results on calibration from the optical design software make this concept possible. Dynamic DFD will further improve the accuracy of DFD even in a weak-contrast condition, and it also can reduce 1 more image in continuous tracking focusing.

Chapter 8

Conclusions

In this dissertation, novel image-based techniques are proposed and developed for 3D modeling with zoom calibration, autofocusing, and continuous tracking autofocusing of moving objects.

A new technique is introduced to extend current 3D modeling technology for variable/dynamic zoom settings so that objects of very different sizes and at many different distances can be modeled by the system. A new calibration technique that includes translation and rotation parameters is developed for projective matrices. Experimental results are presented to demonstrate the effectiveness of the new technique.

Several binary mask based Depth-from-Defocus (DFD) algorithms are proposed to improve autofocusing performance and robustness. A binary mask is defined by thresholding image Laplacian to remove unreliable points with low Signal-to-Noise Ratio (SNR). Three different DFD schemes– with/without spatial integration and with/without squaring– are investigated and evaluated, both through simulation and actual experiments. The actual experiments use a large variety of objects including very low contrast Ogata test charts.

A new spatial-domain DFD technique named Blur Equalization Technique (BET) is presented. The theoretical basis of BET relies on equalizing the *blur* or *defocus* of two different images recorded with different camera parameters. In contrast, comparable spatial-domain techniques rely on equalizing the *deblur* or *focus* of the two images. Also, BET facilitates modeling of images locally by higher order polynomials with lower series truncation errors. The accuracy of BET is further enhanced by discarding pixels with low Signal-to-Noise ratio by thresholding image Laplacians, and relying more on sharper of the two blurred images in estimating the blur parameters. These steps makes BET a very accurate and robust technique useful in practical applications such as depth recovery and autofocusing. Its performance is compared with three highly effective and significantly improved versions of STM1. BET is found to be superior to some of the best comparable DFD techniques in a large number of both simulation and actual experiments.

Different factors that affect the performance of DFD algorithms are studied. In particular, nonlinear sensor response, lens vignetting, and magnification variation affect the accuracy of DFD. In order to implement DFD on off-the-shelf commercial digital cameras, these factors need to be calibrated and corrected. We have presented new calibration methods for these three factors. Their correctness and effects on performance of DFD have been evaluated with experiments.

DFD performance on different sub-sampling schemes are investigated. Experimental results show that most sub-sampling schemes that are of interest in digital imaging technology introduce only a small additional error.

A multiple resolution pyramid SSD matching is proposed for 2D tracking in the first stage. A new multiple base point beta calibration is introduced for Z direction tracking. 3D DFD tracking focusing is realized by combination of 2D tracking and Z tracking. A direct calibration from lens design data is also studied. The initial results demonstrate that the camera calibration can be completed using the lens design data

only.

Further research on improving accuracy and speed of DFD, object tracking, and restoration of focused images from blurred images, will enhance the capabilities of modern digital cameras.

Bibliography

- [1] F. Chen, G.M. Brown, and M. Song. Overview of three-dimensional shape measurement using optical methods. *Optical Engineering*, 39(1):10–22, 2000.
- [2] M. Takeda and K. Motoh. Fourier transform profilometry for the automatic measurement of 3-d object shapes. *Applied Optics*, 22(24):3977–3982, 1983.
- [3] Q.C. Zhang and X.Y. Su. High-speed optical measurement for the drumhead vibration. *Optics Express*, 13(8):3110–3116, 2005.
- [4] V. Srinivasan, H.C. Liu, and M. Halioua. Automated phase measuring profilometry of 3-d diffuse object. *Applied Optics*, 23(18):3105–3108, 1984.
- [5] K.G. Harding and J.S. Harris. Projection moire interferometry for vibration analysis. *Applied Optics*, 22(6):856–861, 1983.
- [6] T. Xian and X.Y. Su. Area modulation grating for sinusoidal structure illumination on phase-measuring profilometry. *Applied Optics*, 40(8):1201–1206, 2001.
- [7] P.S. Huang, Q.Y. Hu, and F.P. Chiang. Double three-step phase-shifting algorithm. *Applied Optics*, 41(22):4503–4509, 2002.
- [8] C.P. Zhang, P.S. Huang, and F.P. Chiang. Microscopic phase-shifting profilometry based on digital micromirror device technology. *Applied Optics*, 41(28):5896–5904, 2002.

- [9] U.R. Dhond and J.K. Aggarwal. Structure from stereo—a review. *IEEE Transaction on System, Man and Cybernetics*, 19(6):1489–1510, 1989.
- [10] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall PTR, New Jersey, USA, 1998.
- [11] S.Y. Park and M. Subbarao. A new technique for registration and integration of partial 3d models. In *Proc. SPIE: Machine Vision and Three-Dimensional Imaging Systems for Inspection and Metrology II*, pages 65–74, October 2001.
- [12] S.Y. Park and M. Subbarao. Automatic 3d model reconstruction using voxel coding and pose integration. In *IEEE Intl. Conf. on Image Processing (ICIP 02)*, pages 533–536, October 2002.
- [13] R. Willson and S. Shafer. A perspective projection camera model for zoom lenses. In *Proceedings of Second Conference on Optical 3-D Measurement Techniques*, October 1993.
- [14] R. Willson. Modeling and calibration of automated zoom lenses. In *Proc. of SPIE: Videometrics III*, pages 170 – 186, October 1994.
- [15] R. Willson. *Modeling and Calibration of Automated Zoom Lenses*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 1994.
- [16] R. Atienza and A. Zelinsky. A practical zoom camera calibration technique: An application of active vision for human-robot interaction. In *Proc. of Australian Conference on Robotics and Automation (ACRA 2001)*, pages 85 – 90, 2001.
- [17] C.P. Zhang, P.S. Huang, D. Donskoy, and Y. Yin. Autofocusing control for a divx burst-cutting-area recording system. *Optical Engineering*, 40(7):1337–1342, 2001.

- [18] M. Taylor, P. Davies, D. Brown, W. Woods, I. Bell, and C. Kennedy. Pulsed co2 tea laser rangefinder. *Applied Optics*, 17(21):885–3844, 1978.
- [19] J. L. Roberts. Quadrature correlation pulse detector. US Patent No. 4038540, July 1977. Assignee: Honeywell Inc.
- [20] W.P. Harthill and J.L. Roberts. Quadrature correlation phase determining apparatus. US Patent No. 4071821, January 1978. Assignee: Honeywell Inc.
- [21] E. Krotkov. Focusing. *Intl. Journal of Computer Vision*, 1(3):223–237, 1987.
- [22] M. Subbarao, T. S. Choi, and A. Nikzad. Focusing techniques. *Optical Engineering*, 32(11):2824–2836, 1993.
- [23] S.K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(8):824–831, 1994.
- [24] M. Subbarao and J.K. Tyan. Selecting the optimal focus measure for autofocus-ing and depth-from-focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8):864–870, 1998.
- [25] Y. Xiong and S. Shafer. Depth from focusing and defocusing. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR '93)*, pages 68–73, June 1993.
- [26] A.P. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9(4):523–531, 1987.
- [27] M. Subbarao and N. Gurumoorthy. Depth recovery from blurred edges. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR '88)*, pages 498–503, June 1988.

- [28] M. Subbarao and T. Wei. Depth from defocus and rapid autofocusing: A practical approach. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR '92)*, pages 773–776, June 1992.
- [29] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(2):97–108, 1993.
- [30] S.K. Nayar, M. Watanabe, and M. Noguchi. Real-time focus range sensor. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(12):1186–1198, 1996.
- [31] M. Watanabe and S.K. Nayar. Rational filters for passive depth from defocus. *Int. J. of Comp. Vision*, 27(3):203–225, 1998.
- [32] M. Subbarao and G. Surya. Depth from defocus by changing camera aperture: A spatial domain approach. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR '93)*, pages 61–67, June 1993.
- [33] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *Int. J. of Comp. Vision*, 13(3):271–294, 1994.
- [34] T. Xian and M. Subbarao. Performance evaluation of different depth from defocus (dfd) techniques. In *Proc. of SPIE: Two- and Three-Dimensional Methods for Inspection and Metrology III (Optics East 2005)*, October 2005.
- [35] D. Ziou and F. Deschenes. Depth from defocus estimation in spatial domain. *Computer Vision and Image Understanding*, 81(2):143–165, 2001.
- [36] P. Favaro, A. Mennucci, and S. Soatto. Observing shape from defocused images. *Int. J. of Comp. Vision*, 52(1):25–43, 2003.
- [37] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(3):406–417, 2005.

- [38] A.N. Rajagopalan and S. Chaudhuri. A recursive algorithm for maximum likelihood based identification of blur from multiple observations. *IEEE Trans. on Image Processing*, 7(7):1075–1079, 1998.
- [39] A.N. Rajagopalan and S. Chaudhuri. An mrf model-based approach to simultaneous recovery of depth and restoration from defocused images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(7):577–589, 1999.
- [40] A.N. Rajagopalan and S. Chaudhuri. *Depth From Defocus: A Real Aperture Imaging Approach*. Springer, New York, USA, 1999.
- [41] M. Subbarao. Spatial-domain convolution/deconvolution transform. Technical Report 91.07.03, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY at Stony Brook, 1991.
- [42] M. Subbarao, T. Wei, and G. Surya. Focused image recovery from two defocused images recorded with different camera settings. *IEEE Trans. on Image Processing*, 4(12):1613–1628, 1995.
- [43] D. Ziou. Passive depth from defocus using a spatial domain approach. In *Proc. of the Intl. Conf. of Computer Vision*, pages 799–804, January 1998.
- [44] F. Deschenes and D. Ziou. Homotopy-based computation of defocus blur and affine transform. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR'03)*, pages I: 398–404, June 2003.
- [45] A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Mach. Vision Appl.*, 12(1):16–22, 2000.
- [46] R. Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, RA-3(4):323–344, 1987.

- [47] Z.Y. Zhang. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334, 2000.
- [48] V. Frmont and R. Chellali. Turntable-based 3d object reconstruction. In *2004 IEEE International Conf. on Cybernetics and Intelligent Systems*, pages 1–3, December 2004.
- [49] M. Subbarao and M.C. Lu. Computer modeling and simulation of camera defocus. *Machine Vision and Applications*, 7:277–289, 1994.
- [50] M.C. Lu. *Computer Modeling and Simulation for Computer Vision Problems*. PhD thesis, Dept. of Electrical Engineering, SUNY at Stony Brook, Stony Brook, NY, May 1993.
- [51] T. Xian and M. Subbarao. Camera calibration and performance evaluation of depth from defocus (dfd). In *Proc. of SPIE: Two- and Three-Dimensional Methods for Inspection and Metrology III (Optics East 2005)*, October 2005.
- [52] P. Meer and I. Weiss. Smoothed differentiation filters for images. *Journal of Visual Communication and Image Representation*, 3:58–72, 1992.
- [53] Z. Myles and N. da V. Lobo. Recovering affine motion and defocus blur simultaneously. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(6):652–658, 1998.
- [54] P. Favaro, S. Osher, S. Soatto, and L. Vese. 3d shape from anisotropic diffusion. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR 2003)*, pages 179–186, June 2003.
- [55] H. Jin, P. Favaro, and R. Cipolla. Visual tracking in the presence of motion blur. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR 2005)*, pages 18–25, June 2005.

- [56] S. Mann. Comparametric equations with practical applications in quantigraphic image processing. *IEEE Trans. Image Proc.*, 9(8):1389–1406, 2000.
- [57] S.Y. Park. Automatic focusing of a digital still camera using a depth-from defocus technique: An approach for compensating non-linear camera response function. Technical report, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY at Stony Brook, 2004.
- [58] T. Yamada, K. Hatano, M. Morimoto, M. Furumiya, Y. Nakashiba, S. Uchiya, A. Tanabe, Y. Kawakami, T. Nakano, S. Kawai, S. Suwazono, H. Utsumi, S. Kato, D. Syohji, Y. Taniji, N. Mutoh, K. Orihara, N. Teranishi, and Y. Hokari. A 1/2-in 1.3 m-pixel progressive-scan it-ccd for digital stillcamera applications. *IEEE Trans. on Electron Devices*, 48(2):222–230, 2001.
- [59] L.L. Cam, J.T. Bosiers, A.C. Kleimann, H.C. van Kuijk, J.P. Maas, M.J. Beenhakkers, H.L. Peek, P.C. van de Rijt, and A.J.P Theuwissen. A 1/1.8" 3m pixel ft-ccd with on-chip horizontal sub-sampling for dsc applications. In *Digest of Tech. Papers on IEEE Intl. Solid-State Circuits Conf. (ISSCC 2002)*, pages 34–35, February 2002.
- [60] I. Tsubaki and K. Aizawa. Restoration and demosaicing for pixel mixture images in dsc video clips. In *Proc. of the 2004 Intl. Conf. on Image Processing (ICIP 2004)*, pages 3291–3294, October 2004.
- [61] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley, Massachusetts, USA, 2001.
- [62] J. Shi and C. Tomasi. Good features to track. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR '94)*, pages 593–600, Seattle, June 1994.

- [63] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *Proc. of IEEE Comp. Soc. Conf. on Comp. Vision and Pattern Recogn. (CVPR '98)*, pages 178–183, June 1998.
- [64] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(10):1042–1052, 1993.
- [65] P. Brasnett, D.R. Bull, and C.N. Canagarajah. Multi-resolution parametric region tracking for 2d object replacement in video. In *Proc. of the 2004 Intl. Conf. on Image Processing (ICIP 2004)*, pages 1013–1016, October 2004.
- [66] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Trans. on Communications*, COM-31(4):532–540, 1983.
- [67] C. Lee and L. Chen. A fast motion estimation algorithm based on the block sum pyramid. *IEEE Trans. on Image Processing*, 6(11):1587–1591, 1997.
- [68] T. Wei and M. Subbarao. Continuous focusing of moving objects using dfdlf. In *Proc. of SPIE: Electronic Imaging Science and Technology*, February 1994.
- [69] G. Surya and M. Subbarao. Continuous focusing of moving objects using image defocus. In *Proc. of SPIE: Machine Vision Applications, Architectures, and Systems Integration III*, October 1994.
- [70] S.Y. Park and M. Subbarao. Image based calibration and its application in continuous focusing. Technical report, Computer Vision Laboratory, Dept. of Electrical Engineering, SUNY at Stony Brook, 2004.
- [71] J.W. Goodman. *Introduction to Fourier Optics*. Roberts Company Publishers, 2004.

- [72] M. Born and E. Wolf. *Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. Cambridge University Press, 1999.