Accurate Depth from Defocus Estimation with Video-Rate Implementation

by

Alex Noel Joseph Raj

A thesis submitted in partial fulfilment of the
requirements for the degree of
Doctor of Philosophy

School of Engineering
University of Warwick

September 2009

Table of Contents

List of Figures

List of Tables

Acknowledgements

For the past four years, I have been working on my Ph.D. research project concerning optical depth measurement. The thesis is the result of my study on this topic, which would not have happened without the help of many people. At this moment I would like to thank all those who have encouraged me towards the study.

First and foremost, I would like to thank my supervisor: Dr. Richard Staunton for his continuous guidance and support over the last four years.

I would like to thank my colleague, Mr. Sheng Cheng for providing helpful advice on the use of FPGA's.

I am grateful to Mr. Charles Joyce for constructing the miniature models or real objects used in the experiments.

I would personally like to thank my brother, Mr. Anil Kumar and my brother-in-law, Mr. Saju Rebello for providing great emotional and financial support.

The University of Warwick has funded this research in the form of a Warwick Post Graduate Research Fellowship, I am really grateful to them.

Finally, I would like to thank God for keeping me as a precious gift in his eyes.

Dedication

To my parents, Dr. L. Jean Margaret and Mr. R. Joseph Raj

To my grandparents, Mrs.V. Raju and Mr. D.L. Raju

To my entire family back in Tambaram, Chennai, India

Declaration

This thesis is submitted in partial fulfilment for the degree of Doctor of Philosophy under the regulations set out by the Graduate School at the University of Warwick.
This thesis is solely composed of research completed by Alex Noel Joseph Raj under the supervision of Dr. Richard Staunton.
None of the work presented here has been published or submitted for another degree.

Abstract

The science of measuring depth from images at video rate using ‘defocus’ has been investigated. The method required two differently focussed images acquired from a single view point using a single camera. The relative blur between the images was used to determine the in-focus axial points of each pixel and hence depth.

The depth estimation algorithm researched by Watanabe and Nayar was employed to recover the depth estimates, but the broadband filters, referred as the Rational filters were designed using a new procedure: the Two Step Polynomial Approach. The filters designed by the new model were largely insensitive to object texture and were shown to model the blur more precisely than the previous method. Experiments with real planar images demonstrated a maximum RMS depth error of 1.18% for the proposed filters, compared to 1.54% for the previous design.

The researched software program required five 2D convolutions to be processed in parallel and these convolutions were effectively implemented on a FPGA using a two channel, five stage pipelined architecture, however the precision of the filter coefficients and the variables had to be limited within the processor. The number of multipliers required for each convolution was reduced from 49 to 10 (79.5% reduction) using a Triangular design procedure. Experimental results suggested that the pipelined processor provided depth estimates comparable in accuracy to the full precision Matlab’s output, and generated depth maps of size 400 x 400 pixels in 13.06msec, that is faster than the video rate.

The defocused images (near and far-focused) were optically registered for magnification using Telecentric optics. A frequency domain approach based on phase correlation was employed to measure the radial shifts due to magnification and also to optimally position the external aperture. The telecentric optics ensured pixel to pixel registration between the defocused images was correct and provided more accurate depth estimates.

Abbreviations

ASIC	Application Specific Integrated Circuits
BRDF	Bidirectional Reflectance Distribution Function
BSB	Base System Builder
CC	Correlation Coefficient
CCD	Charge Couple Device
CLB	Configurable Logic Blocks
CS	Complex Spectrogram
DAQ	Data Acquisition System
DCM	Digital Clock Manager
DDR2 SDRAM	Double-Data-Rate Synchronous dynamic random access memory
DFD	Depth from Defocus
DFF	Depth from Focus
EDK	Embedded Development Kit
ESF	Edge Spread Function
FFT	Fast Fourier Transform
FIFO	First In and First Output
FPGA	Field Programmable Gate Array
IOB	Input Output Block
LOG	Laplacian of Gaussian
LSF	Line Spread Function
LUT	Look Up Tables
$\frac{M}{P}$	Amplitude ratio between the differences of the amplitude of the defocused images to the sum
MAP	Maximum a Posteriori
MHz	Mega Hertz
MRF	Markov Random Field
MSE	Mean Square Error
PE	Processing Elements
PLB	Processor Local Bus
PPC	Power PC

CHAPTER 1

Introduction

In the real world, objects are perceived in three dimensions (3D); length, breadth and depth. Humans observe 3D by utilising one or a combination of the available depth clues:- texture blur; edge blur, size perspective; binocular disparity; motion parallax; occlusion effects; and variations in shading [20]. The problem arises when the 3D objects are imaged by a photographic system. Here a 3D plane is mapped on to a 2D plane with reduced height and width information. The task of retrieving the depth information from one or more 2D images is an active research topic within the broad area of Computer Vision. The recovered depth information plays a vital role in Industrial and Medical applications such as component inspection, robotic manipulations, autonomous vehicle guidance, and 3D endoscopy.

The image formation process provides a geometric correspondence between the points in the 3D scene and the 2D image. In Perspective Projection, the light rays from the object that pass through a pinhole aperture define the image. Here each point in the image corresponds to a particular point of the object. In Orthographic Projection, light rays parallel to the optical axis form the image. By hypothesis, it corresponds to perspective projection when the camera is at an infinite distance from the object, and the lens has an infinite focal length. Figures (1.1a) and (1.1b) explain perspective and orthographic projections, where the x, y plane lies perpendicular to the optics axis and the z direction along it. It is the x, y image plane that provides the data for the range calculation.

Range acquisition methods can be broadly classified as optical and non-optical methods. Non-optical methods are based on (1) Mechanical; (2) Inertial; (3) Magnetic; and (4) Ultrasound. Together with LASERs they provide accurate single point depth measurements but require expensive computations and scanners to provide a dense depth map. On the other hand, 2D optical methods provide acceptable depth accuracy with a high possibility of recovering the shape (dense depth map) from images. Methods can also be broadly classified as Active or Passive. Active methods operate in a controlled environment aided by the use of either controlled energy beams (as in ultrasonic and in optical time-of-flight approaches) or using strip and grid lighting, and Moire fringe patterns (as in Contrived lighting based approaches) [20] [93]. These methods find usage in indoor laboratory and factory environments.

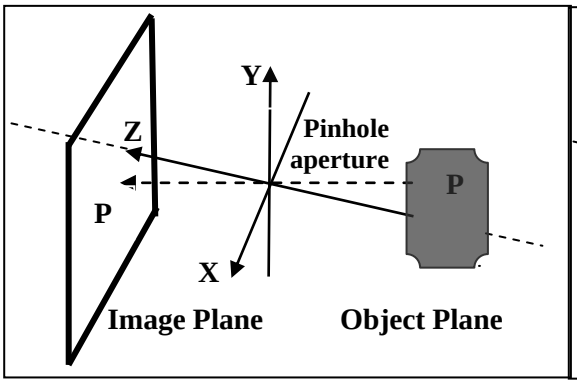


Figure 1.1a: Perspective Projection [53]

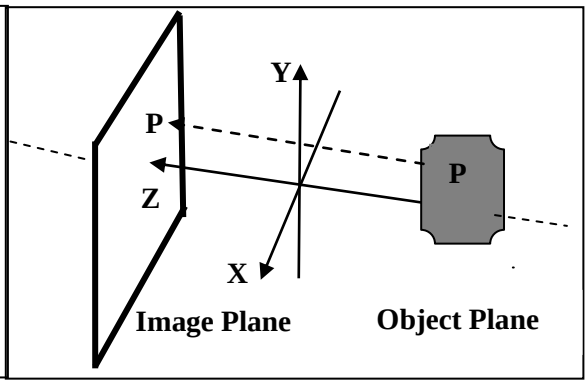


Figure 1.1b: Orthographic Projection [53]

Passive methods imitate the human biological vision system and therefore constantly search for 'depth clues' within the acquired images. They are not limited to any environmental constraint and find usage in military, medical and industrial applications [20]. The research here has concentrated on passive optical depth recovery. A brief description of passive depth recovery is presented in the next Section.

1.1. Passive Depth Recovery Methods

Optical depth estimation techniques can be categorized as Monocular or Binocular. Monocular techniques allow depth estimation using a single camera by considering depth clues such as the relative size of the objects, the distribution of light and shade, movement parallax of subject and background, and by measuring the amount of focus or defocus [20] [40] [93]. Binocular vision techniques require at least two images acquired from different viewpoints. These images are compared and the disparity between the images is related to the actual depth. Depths from Stereo and Structure from Motion are examples of Binocular vision techniques. Shape from Shading, Shape from Silhouettes, Depth from Focus and Depth from Defocus are instances of Monocular vision techniques.

1.1.1. Depth from Stereo

A simple stereoscopic system requires two images captured from different viewpoints. The viewpoints are separated by a suitable distance so as to provide two disparate images. The depth information is recovered by calculating the disparity information between these images. The typical stereo system is shown in Figure (1.2).

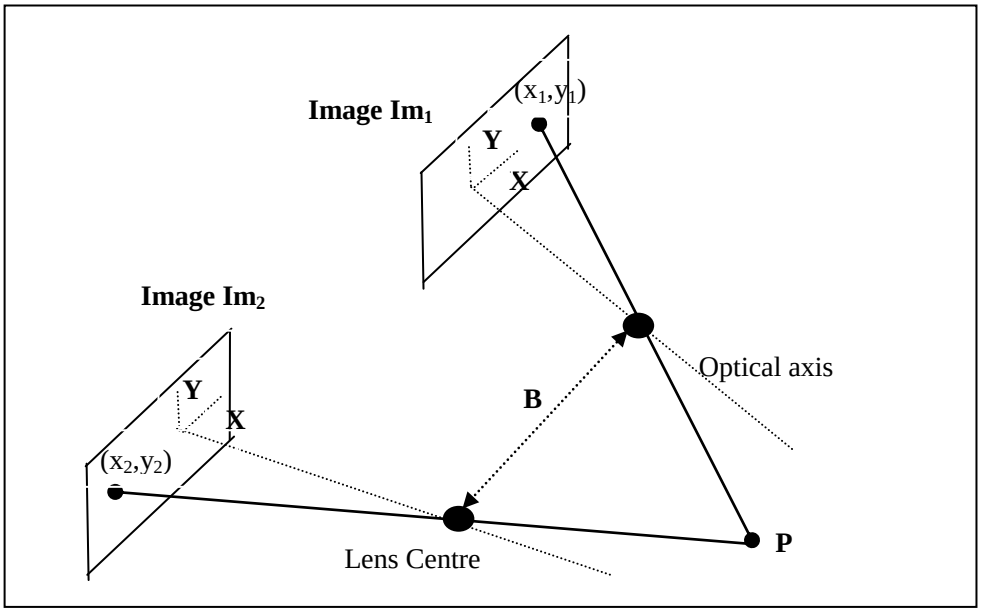


Figure 1.2: Depth from Stereo [20]

The two images, *Im_1 and Im_2* of the object *P* (see Figure (1.2)) are captured at two different viewpoints separated by a baseline distance *B*. If *f* is the focal length of the lens and *d* is the stereo disparity between the objects in the images, then the depth of the object *z* is inversely proportional to the disparity and is given by [20]

$$z = \frac{f(B-d)}{d} \quad \text{----- (1.1)}$$

The major difficulty in stereo imaging arises when establishing the correspondence between the objects in the two images. This process requires unique matching points to establish a pairing relationship and proves uncertain when the scene under investigation has:- (1) uniform intensity; or (2) is prone to occlusion effects (missing part problem) [20] [93]. Stereo pair analysis based on edge data has been presented by Baker [94], where the correspondence problem was solved by using an edge

correlation procedure. Marr and Poggio [95] tackled the correspondence problem by considering a cooperative computational procedure. Their work was motivated by results of Julesz [96] on random-dot stereograms, which suggested that monocular vision does not provide any high level clue for disparity analysis [20], but in 1987 Pentland re-examined monocular technique and suggested that blurred edges can provide valid depth clues [1]. Marr *et al*. [97] [98] presented an algorithm that was analogous to the low-level human biological system. The disparity information was obtained from the zero crossing of the edges extracted from the right and left images, and the correspondence problem was solved by using disparity matches of gross line structures [93]. Stereopsis is analogous to the human visual system. It can provide dense depth maps, since an entire frame can be processed and depth estimates can be recovered for each pixel. Further, the depth maps generated are reliable, since there is no mechanical movement involved in the whole process.

1.1.2. Structure from Motion

The method of recovering surface information using the relative motion between the object and the camera is referred to as Structure from Motion (SFM). It differs from stereo where the camera motion is restricted to a limited lateral displacement [20]. The common approach is to compute the observables such as points, lines, occluding contours and optical flow, and relate them to the structures and events in the space [20]. In feature based approaches, the point correspondences between the images are first computed. Next, the motion parameters are determined from the image coordinates by solving a set of equations. The estimated motion parameters specify the object distance. Williams [99] proposed a method where the planar objects were assumed to be oriented only in the vertical and horizontal directions. A segmentation procedure was first employed to define the extent of the planar regions, and later an optimization strategy was adopted to determine the distance. Prazdny [100] employed an optical flow method and recovered the instantaneous egomotion (observer motion) and the surface normal map. The surface normal map provided the required range information [20]. An illustration of shape recovery using SFM has been presented in [20]. Here the depth information was determined from five matching points that relate the image-space and the object-space. An optimization procedure was employed to solve the non-linear equations which provided the

required depth. Like stereo, SFM techniques also suffer from correspondence matching problems, and reliable depth maps are achieved only when accurate matching points are available. Further, these techniques require high resolution images to accurately determine the motion parameters. Additionally, SFM techniques are sensitive to the presence of noise in the observation and prove expensive in terms of storage and computation [20].

1.1.3. Shape from Shading

Shape from Shading (SFS) refers to the problem of extracting surface orientation from the gradual variation of brightness (shading) in the image [65]. Horn [53] discovered that the 3D shape of a surface can be recovered from a single image by considering the surface reflectance properties and the spatial distribution of the light sources. The brightness of the surface is described by the Bidirectional Reflectance Distribution Function (BRDF), which is the ratio of the radiance of the surface patch as viewed from the direction (θ_e,φ_e) to the irradiance resulting from illumination from the direction (θ_i,φ_i) (see Figure (1.3)).

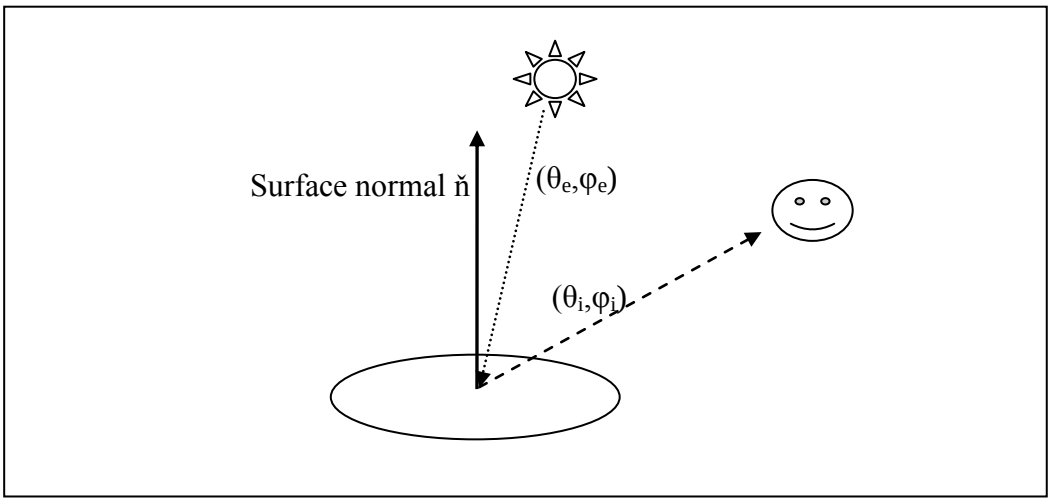


Figure 1.3: Structure from Shading [53]

The reflectance map, $R(p,q)$, describes how the target radiance varies with the surface orientation for a given source distribution. It also presents a relationship for 3D shape recovery in terms of brightness (shading). This relation is expressed by Horn [53] in terms of image irradiance and is given by the equation,

$$R(p,q) = E(x,y) \text{ --- } (1.2).$$

Here p,q denote the slopes of the surface along the x and y directions, and $E(x,y)$ denotes the intensity at points x and y. The idea is to first compute the reflectance map, $R(p,q)$ of the scene and then determine the gradients p,q for each point along x and y for a set of $R(p,q) = E(x,y)$ equations with different light source positions. The camera and the scene are kept stationary [93]. SFS techniques are divided into four main approaches:- minimization, propagation, local and linear. A detailed report on the performance of these techniques is presented in [65]. SFS techniques require a prior knowledge of the scene reflectance and hence are not suitable for arbitrary scenes whose reflectance is unknown. Moreover, these methods cannot recover absolute depth and thus depend on a hybrid algorithm (mostly combined with stereo) to generate a reliable depth map [20] [65]. Further, the shape information from shadow areas is not recovered and so additional information has to be provided via techniques such as Shape from Shadow [65].

1.1.4. Shape from Silhouettes

Shape from Silhouettes (SFSh) is a method of reconstructing 3D models from the silhouettes of scenes acquired from different view points. A 2D silhouette is a set of close contours that outline the projection of an object onto the image plane. A general SFSh technique would segment the silhouette of the object under investigation from the rest of the image and use a combination of the silhouettes acquired from different views to provide a strong clue for image reconstruction.

The object segmentation is achieved either by simple differencing or by a blue screen segmentation technique. Once the silhouettes are obtained, they are back-projected on to the 3D space to define the volume (shape). The intersection volume has been referred to as the visual hull by Laurentini et al. [101]. Most SFSh methods are based on the voxel-based data-structure approach described by Szeliski [102]. A method described in [106] reconstructed the 3D shape by: - (1) Obtaining a set of voxels (Octree) from SFSh [102]; (2) Generating the surface triangles using the marching cubes (MC) technique [103]; and (3) Estimating the surface normal. Though SFSh methods can recover the 3D surface from arbitrary objects without the any assumptions about the images, they fail to recover the shapes of regions such as cavities (eg. coffee cup handle) or holes, and hence required additional information

to reconstruct these regions. An improved surface reconstruction algorithm that aggregated the local surfaces constructed by the 3D convex hull method has been presented by Shin and Tjahjadi [104]. They used the connectivity information of an octree and provided an improved surface reconstruction for the imperfect MC result. SFSh methods are particularly good when crude 3D models of the real world objects are required, and hence find usage in commercial 3D modelling packages [105]. However they suffer the drawback that the shape reconstruction could be affected by the type of the object and also by the camera position.

1.1.5. Depth from Focus

In Depth from Focus (DFF), the knowledge of the camera parameters is used to estimate the depth of an object. The sharpness of focus is measured on a sequence of images captured over a range of lens positions and related to the actual depth using the lens law [43]. A typical setup for estimating the range using focus is shown in Figure (1.4).

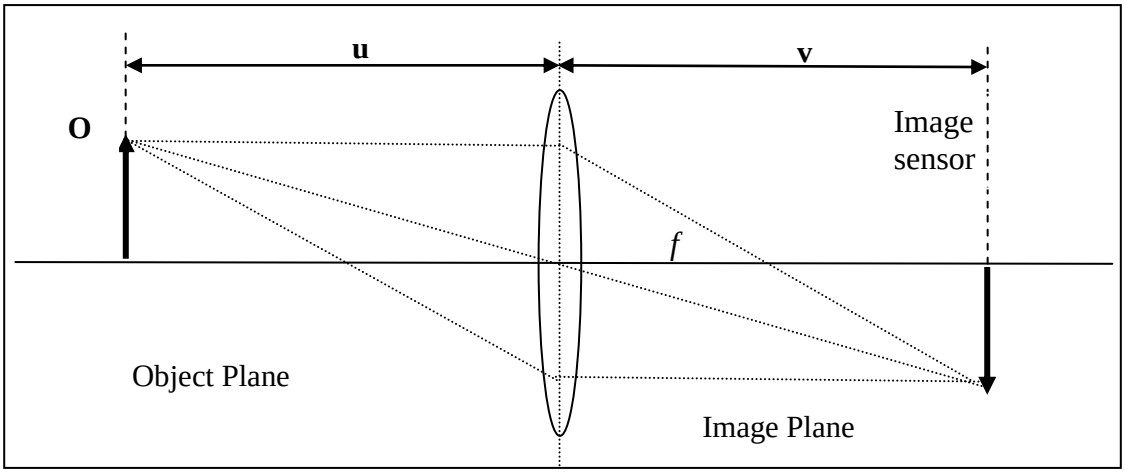


Figure 1.4: Depth from Focus

For an aberration free convex lens, when the object *O* at distance *u* from the lens is in focus, the image *I* is formed at a distance *v* on the image sensor. The relation between the focal length of the lens *f*, the object distance *u* and the image distance *v* is given by the lens law:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v} \quad \text{---- (1.3).}$$

In DFF, the idea is to obtain a sharply focused image by adjusting either the focal length f or the image distance v or both. The measured f and v are substituted in equation (1.3) to determine the object distance u. In practice, a series of images are captured by continuously varying the image distance and the sharpest image for each object is found by employing a focus measure operator. Jarvis [73] suggested three focus measures based on computational simplicity, effectiveness, consistency, and implementation feasibility. They are: - (1) entropy; (2) variance; and (3) sum modulus difference. Darrell and Wohn [39] employed Laplacian and Gaussian pyramids to measure the sharpness criterion. A pipelined processor capable of generating a depth map in 10sec was also presented. The other researchers who actively contributed to DFF are Subbarao [92], Grossman [4] and Nayar [75].

Depth estimation using the focus criterion is a simple procedure that provides a direct relation to the actual depth using the lens law. It is monocular (only one camera position is involved) and hence does not suffer from the correspondence problem. Further, no additional hardware is required except for a computer controlled motor to adjust the lens position. DFF is essentially a search technique that requires the acquiring and processing of at least 10 to 12 images. This forms the fundamental weakness of the method since additional time is required to adjust the camera parameters before capturing each image. Further, during the entire period of adjusting the camera parameters the scene must remain stationary.

1.1.6. Depth from Defocus

The Depth from Defocus (DFD) technique is based on the inherent inability of a practical optical system to focus at all distances in a scene. When a point light source is in focus, all the rays radiated from the object that are intercepted by the lens converge at a point on the image plane. But when the point light source is not in focus, its image is not a point but a blurred circular disc of finite radius r_b. The disc radius is a function of the lens parameters and the disc is referred to in the literature as the circle of confusion. The basic idea behind DFD is to measure the blur radius and relate it to the actual depth using the simple lens law. However in practice every point in the scene provides an overlapping blur circle and this complicates the calculation. Figure (1.5) shows a conventional DFD system.

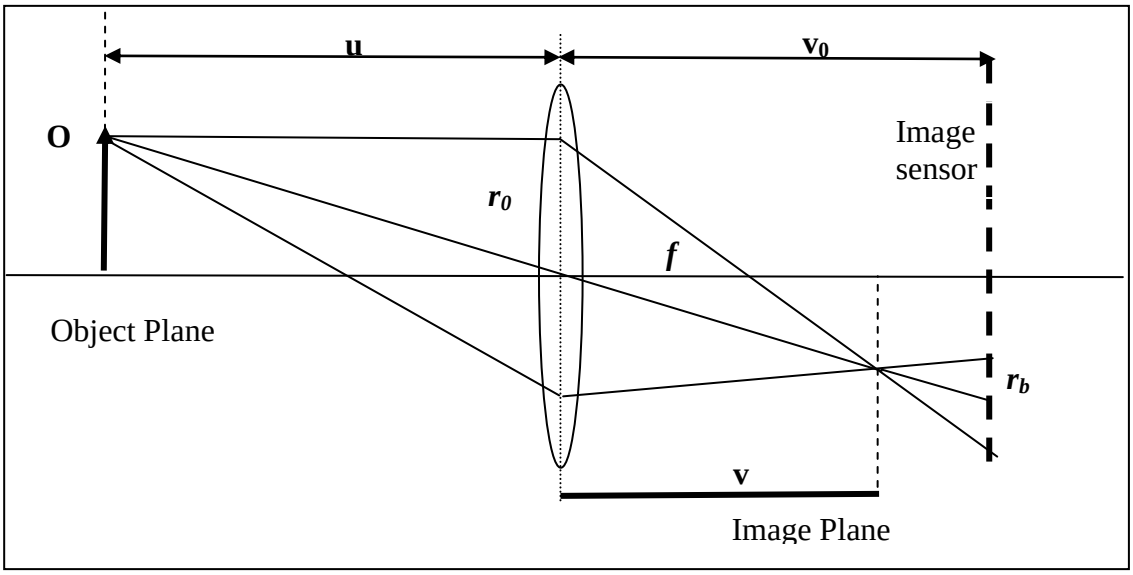


Figure 1.5: Depth from Defocus

In Figure (1.5), *u* refers to the object distance, *v* denotes the distance from the lens to the focused image, v_0 refers to the distance between the lens and the image sensor, r_0 refers to the radius of the lens aperture and r_b denotes the blur circle radius. From the lens geometry and the similarity of triangles, the blur circle radius is given as

$$r_b = r_0(\frac{v_0}{v} - 1) \text{ ---- (1.4)}$$

From the lens law, $\frac{1}{v} = \frac{1}{f} - \frac{1}{u}$. Substituting $\frac{1}{v}$ in equation (1.4) gives the relationship between blur circle radius and the object distance as

$$r_b = r_0 v_0(\frac{1}{f} - \frac{1}{v_0} - \frac{1}{u}) \text{ ---- (1.5)}$$

The blur radius r_b can be either positive or negative depending on whether the object is in front or behind the focused plane. The ambiguity can be overcome by constraining the sensor distance v_0 to be always greater than the image distance *v*. In this case the depth is recovered only if prior knowledge of the scene's characteristics is known. To overcome this constraint, researchers have suggested the use of two images acquired on either side of the focused image. These images (near and far-focused) are identical except for the degree of blurring. The change in blur information is used to recover the depth information. Depth estimation is based on modelling the defocused image as the convolution between the focused image and the 2D Point Spread Function (psf) of the lens. Three different psf models have been

considered by researchers. They are: - (1) Gaussian; (2) Pillbox; and (3) Generalised Gaussian. A detailed review of the DFD techniques with their merits and demerits is presented in Chapter 2. Unlike DFF, the DFD methods do not search for the best focused image and hence require only a few images (usually 2) to provide a reliable depth map. Further, there is no correspondence matching problem as is attributed to the stereo and motion algorithms. DFD finds usage in applications where the imaging geometry prevents the use of multiple viewpoints. The limitations of DFD include [20]:- (1) Need for accurate modeling of the optical system; (2) Ensuring a sufficient amount of spectral information to measure the blurring between the images; (3) Edge bleeding due to windowing and (4) Need for accurate calibration of the camera parameters.

Passive depth recovery methods have their own limitations and can suffer from one or more of the following drawbacks:-

(1) Missing parts and the correspondence matching problem. Stereo and SFM techniques suffer from the above problem. Depth estimation is possible only at places where features are matchable, and thus require interpolation techniques to provide a dense depth map. Further, SFM techniques involve solving nonlinear equations by optimisation and thus require good initial guesses to arrive at a favourable solution.

(2) Controlled illumination requirement. SFS techniques do require environments which can offer control over the incident illumination. Since these techniques rely on accurate modelling of the surface reflectance, they are not suitable for complex natural scenes with arbitrary depths. Further, depth recovery is not possible for regions in shadow.

(3) Computational complexity. All the methods explained above are computationally complex to an extent. For example in stereo, significant time is required to solve the correspondence problem. DFF techniques provide reliable depth estimation, but are inherently slow since they need to acquire and process at least 10 to 12 images. Further, additional hardware is also required to adjust the lens position.

DFD techniques do have their own limitation (refer to Section 1.1.6) but due to their simplicity in operation, they can compare favourably to other depth estimation methods. With passive illumination, they require a minimum of two images acquired from the same viewpoint to produce a dense depth map, and thus can be useful for real-time depth recovery systems and for auto-focussing applications. Moreover, Schechner and Kiryati [88] claimed that for the same physical dimension, the DFF and DFD systems do not completely avoid the occlusion problem, but they are more stable in the presence of such disruptions than stereo. In addition, DFD methods are robust, since they involve modelling a single 2D psf rather than two distinct responses as in stereo. Considering the advantages of the DFD technique over other optical range methods, this research work investigates the use of the passive DFD method to develop a real-time depth / shape recovery system. The novelty lies in designing the texture invariant broadband filters using the Two Step Polynomial Approach and implementing the DFD algorithm on a Field Programmable Gate Array. A detailed report is presented in later chapters.

1.2. Organisation of the thesis

Chapter 2: Provides a detailed review of existing DFD techniques. The techniques were categorized based on: - (1) The method, active or passive; (2) The number of images required; and (3) The mode of operation. Further, the merits and the limitations of each technique along with the achieved depth accuracy are reported. A comparison Table based on the above categories is presented in Appendix 5.

Chapter 3: Presents an algorithm to measure the magnification changes between two defocused images using the Fourier technique: Phase Correlation. The measured magnification variation was helpful in setting up the Telecentric optics. A brief review of the existing image registration techniques and the experimental results for shift detection in simulated and real images is presented in this chapter.

Chapter 4: Presents a novel design procedure that determines the rational filter coefficients by accurately modelling Watanabe and Nayar's $\frac{M}{P}$ ratio curves [14]. The method referred as the Two Step Polynomial Approach determines the filter coefficients by considering the discrete $\frac{M}{P}$ ratio space. A comparison study is presented to determine how well the filters designed by both the Watanabe and the Two Step Polynomial Approach fit the theoretical $\frac{M}{P}$ ratio curves. Further, the depth estimation results for a single frequency test image and a real checkerboard image are presented. In addition, the chapter also investigates the effects of focal length, the aperture diameter, and the pixel size of the sensor on the rational filter's design, and on the working distance of a given experimental setup.

Chapter 5: Presents a hardware implementation of the DFD algorithm on the Virtex 2P FPGA. A pipelined architecture with two separate channels was employed to implement the five filtering stages in parallel. Further, a procedure referred as the Triangular method was used to reduce the number of multipliers required for the convolution process. Finally, a comparison study is performed where the depth results from the pipelined processor are compared against the full precision Matlab output.

Chapter 6: Presents depth estimation results for 3D objects with natural textures. Again, a detailed statistical comparison is presented for the depth estimates obtained from Matlab and from the pipelined processor.

Chapter 7: Summarises the contribution of the research work and presents some avenues for future research.

CHAPTER 2

Review of the Depth from Defocus Techniques

Introduction

Depth from Defocus (DFD) methods like any ranging techniques can be broadly classified as Active or Passive. Active methods required an external illumination pattern to be projected under controlled environmental conditions on to the object that requires measurement, whereas Passive methods recover depth under ambient lightning conditions. However, if the scene under investigation has weak texture or is textures-less, then the only possibility is to employ an active illumination.

Early methods [1] [2] [5] were based on single images, where the defocus information was obtained from the blur measurement. These techniques required prior knowledge of the scene and were sufficient enough to recover depth only at certain image contours [15]. DFD methods for arbitrary objects using multiple images (two or more) have been proposed by many researchers. These methods can be further classified based on the mode of operation as Spatial, Frequency, Wavelet or Statistical techniques.

In this chapter an attempt has been made to categorize the available DFD techniques based on (1) The method, active or passive; (2) The number of images required, single or multiple images; and (3) The mode of operation, spatial domain, frequency domain, statistical, wavelets and other un-conventional techniques. Figure (2.1) illustrates pictorially the available DFD methods based on the above categories.

The chapter first explains the Passive methods where an in-depth analysis of the DFD techniques is reported based on the proposed categories. Section 2.1.1 describes the single image passive DFD techniques and Section 2.1.2 describes the multiple image passive DFD techniques. Later, Section 2.2 describes the active methods which are classified as per the categories. A comparison chart is presented in Appendix 5, where the detailed information about the authors, their techniques, the merits and demerits of their method, and the achieved depth accuracy is reported.

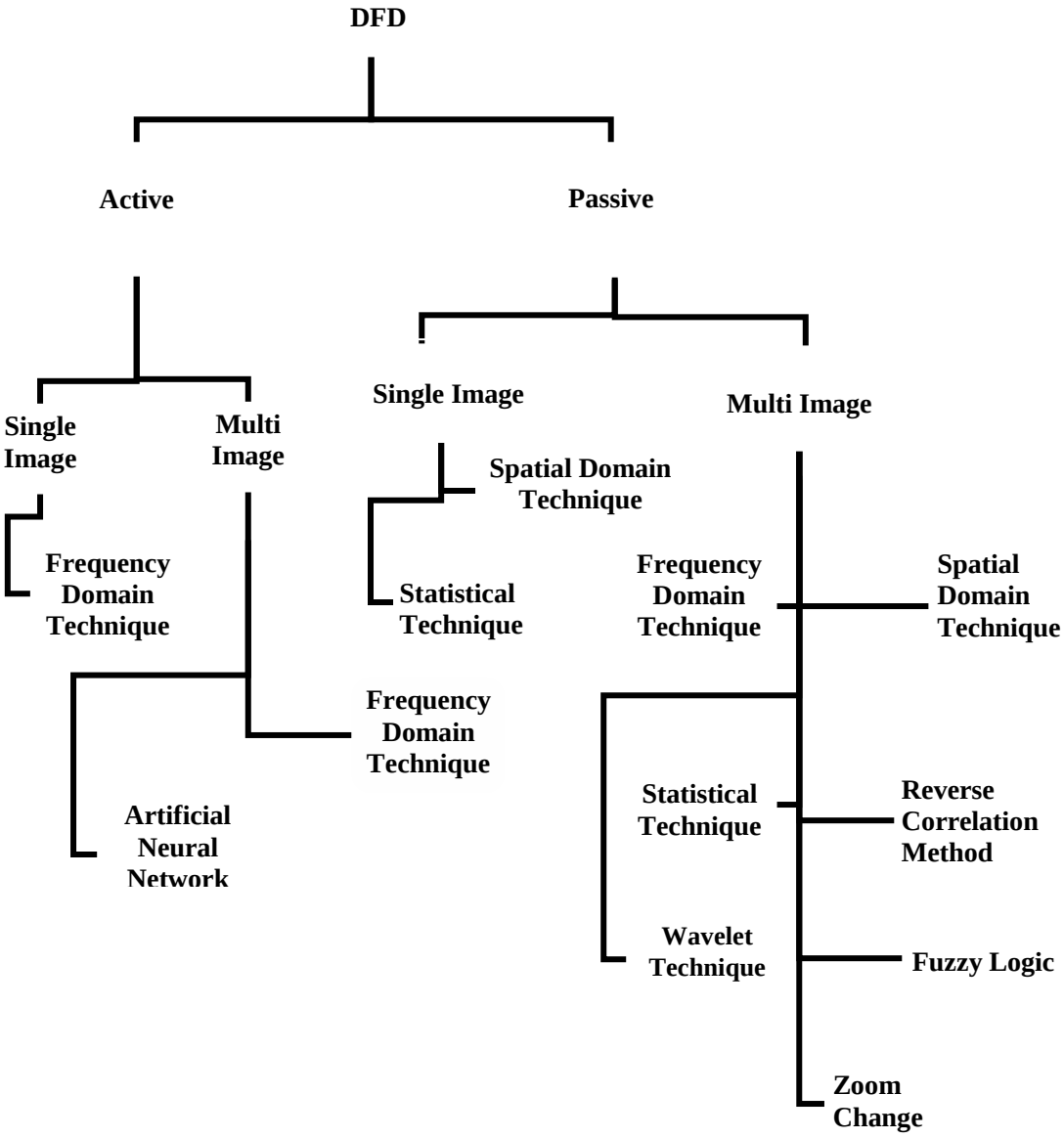


Figure 2.1: Pictorial representation of the available DFD methods based on the categories.

2.1. Passive Methods

Passive DFD methods are attractive since they estimate the depth of the scene under the ambient conditions. They avoid the usage of any illumination patterns and hence are suitable for depth as well as shape recovery. The main disadvantage which is common to any passive technique is the requirement of a textured pattern since a texture-less object will 'look the same' whether focused or unfocussed. This Section describes the available passive DFD techniques.

2.1.1. Single Image Techniques

Alex Pentland was the first investigator who employed 'defocus' as a clue to estimate the depth of an object. He observed that Depth from Focus (DFF) techniques and auto-focussing algorithms employed an exhaustive search mechanism to find the 'best' focussed image from a collection of 30 or more images. These techniques were time consuming and required sophisticated parallel hardware for effective operation. He realised that search for the best focused image was unnecessary and presented a novel method where the depth was estimated from a single image by measuring the error in focus; the focal gradient [1]. The amount of defocus (blurring) was related to the distance of the object from the focused image and the characteristics of the lens. The object depth D was measured using the relation,

$$D = \frac{f v_0}{v_0 - f - \sigma F} \text{ ---- (2.1)}$$

where, f was the focal length of the lens, v_0 the distance between the lens and the image plane, F the f-number of the lens[1] and σ the spatial constant of the 2D Gaussian psf of the defocused lens. The only unknown in equation (2.1) is σ of psf which is a measure of the rate of change of image intensity at sharp discontinuities in the images (e.g. edges). A Laplacian operator was employed and the zero crossing of the Laplacian provided the maximum rate of change of image intensity i.e. the edges. By using a linear regression model, σ was estimated and substituted in equation (2.1) to obtain the object distance. Although the method looked simple it had two main disadvantages :- (1) The method required the prior knowledge of the scene

1. **Here F represents the f-number of a conventional lens. $F=f/d$ where d is the aperture diameter.**

characteristics and hence could be used to measure depth only at step discontinuities; and (2) the ambiguity whether the image was formed in front or behind the plane of exact focus had to be overcome by a suitable scene setup. These demerits were later addressed by Pentland [1] [2] by considering two images of the same scene taken with different aperture settings. The algorithm had the potential to produce depth plane segmentation but was not accurate enough to produce a dense depth map. Grossman [4] has achieved an accuracy of $\pm$ 1.25cm using the above method.

After Pentland, Subbarao and his research associates were the most active advocates of the DFD method. In 1988, Subbarao and Gurumoorthy [5] proposed a method similar to Pentland's [1] where the Line Spread function (LSF) corresponding to the psf of the lens was computed from a blurred edge. The spread of the LSF, measured from the second central moment (standard deviation distribution of the LSF) was linearly related to the inverse distance using the equation

$$\sigma_l = mu^{-1} + c \quad \text{--- (2.2)}$$

where, σ_l was the spread of the LSF, m and c were the camera parameters and u^{-1} the object distance. The approach differed from Pentland's in the computational simplicity of measuring the magnitude of the blurred edge and relaxed the assumption that the psf to be modelled was Gaussian. Here the psf was considered to be rotationally symmetric. The algorithm works well on isolated edges but causes depth estimation errors in the presence of other edges.

Lin and Gu [69] proposed a model that estimated the blur by employing a histogram technique that measured the pixel intensity distribution of a single image. The estimated histogram was then related to the actual depth using a pre-calibrated mathematical model. Experimental results with real images suggested a RMS error less than 3% when the furthest point was at 1200mm.

Namboodiri and Chaudhuri [67] proposed a statistical method based on the inhomogeneous reverse heat equation that estimated the blur information and depth perception using a single image. The model formulated the Gaussian psf in terms of the heat equation and related the blurring parameter σ to the diffusion coefficient as $\sigma^2 = \frac{tc}{\gamma}$, where t is the time variable, c is the diffusion coefficient and γ is the size

of the blur in terms of pixel units. The heat equation was inhomogeneous since the coefficient c and the time t are related to the depth location. The depth information was measured as a disparity between the observed image and the reconstructed image, and the estimated depth map was further refined using a Markov Random Field (MRF). Although the depth map was retrieved from a single image, it was actually similar to Favaro's multi-image diffusion method [68]. Theoretical results were not provided due to the ambiguity of whether the object was in front or behind the focussed image.

2.1.2. Multiple Image Techniques

With multiple image techniques, two or more images acquired with different camera settings are compared to provide the required depth estimate. The methods offer two main advantages over the single image technique: - (1) They avoid the extensive pre-calibrated depth model required for single image techniques [69], since the ambiguity whether the object is in front or behind the focus plane has been overcome; (2) They do not require any prior knowledge of the scene and hence can be applied for arbitrary objects with any random shape. Further, the Section provides a sub-classification based on the core technique used.

2.1.2.1. Frequency Domain Techniques

Pentland's second approach [1] [2] was based on a multiple image frequency domain technique, where two images of the same scene captured using different aperture settings (smaller aperture to capture a focused image and a larger aperture to capture a defocused image) were used to estimate the amount of defocus. Since the aperture sizes were different, the same point on the scene was focused differently in the each image, leading to a difference in focal error between them. This focal error estimated over the entire image provided a dense depth estimate of the scene. The psf was modelled as a 2D Gaussian and the spatial constant σ was measured by fitting a regression model to the Fourier Transform ratio of the focused image to that of the defocused image. The measured σ was then related to the object distance using equation (2.1).

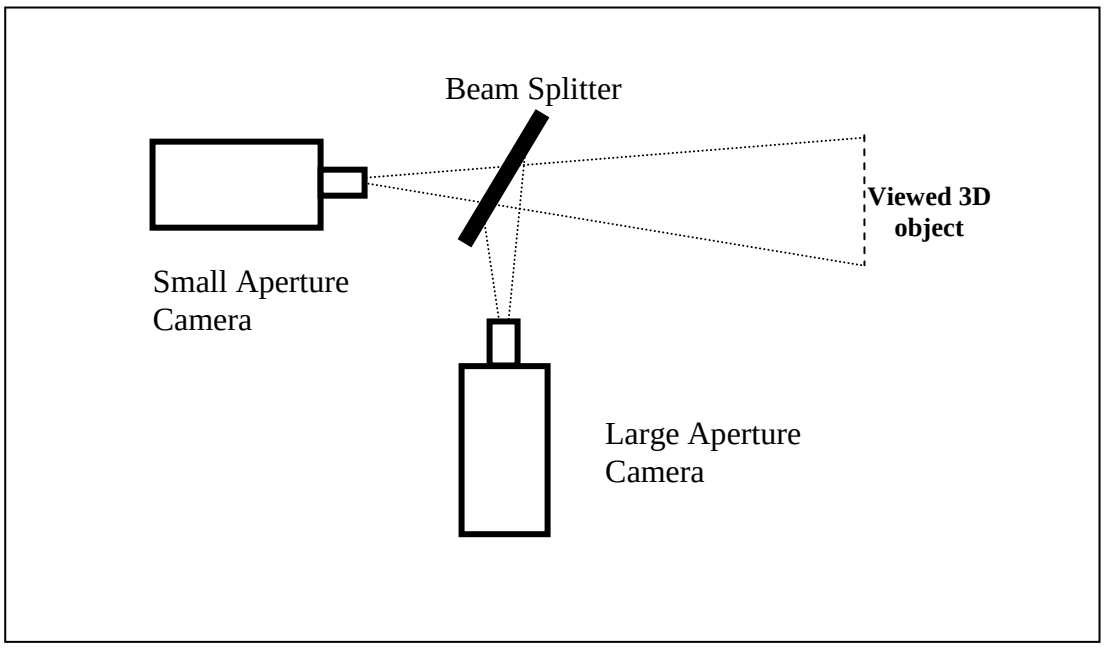


Figure 2.2: Passive DFD optical setup based on Pentland's approach [1]

In practice the images were first convolved with a 8x8 Laplacian filter and averaged using a 8x8 Gaussian filter to produce a 'power image'. This provided an estimate of the power of the central spatial frequency of the Laplacian filter at each image location. The two transformed images were then compared and a look up-table was used to estimate the depth. The algorithm was implemented on a Datacube image processing system and included a beam splitter to capture two images simultaneously as shown in Figure (2.2). The system processed 8 frames per second with an accuracy of 6% standard error over a 1 cubic meter measurement volume. The accuracy was improved to 2.5% standard error by considering a Laplacian pyramidal architecture where the Fourier powers were estimated at several frequencies instead of single frequency. The disadvantage of the algorithm was its assumption that one of the images was taken using a pin-hole camera which was unacceptable as such a tiny aperture required a long exposure time and produced diffraction effects that were more pronounced as the width of the aperture was decreased [53].

Subbarao [6] relaxed Pentland's requirement of a pinhole aperture and recovered depth by considering two images (which may or may not be in focus) acquired with different camera settings. The depth was recovered by changing: - (1) The distance between the lens and the image detector; (2) The focal length of the lens; and (3) The aperture diameter. The ratio of the Power Spectral Density (PSD) over a small local

area was employed to estimate the spread parameter (σ_1, σ_2) of the two defocused images. These were then related to the inverse of the actual depth by equation (2.2). Experiments proved that the computed σ was strictly a monotonic function between two set intervals and provided accurate depth results for nearby objects. For far away objects the method provided qualitative information.

In [7] [10], Subarao and Wei employed the DFD technique for autofocus applications. The method referred to as DFD1F, was based on computing the one dimensional Fourier coefficients as opposed to two dimensional and hence provided computational advantage and robustness for practical applications. The approach was based on the accurate calibration of the psf which was computed from the LSF of blurred step edges as explained in [5]. In actual practice, the estimated blur parameter σ was used as an index for a look-up table that provided a calibrated psf, modelled either as a Gaussian or a Pillbox. It was reported, that for low levels of blur the Gaussian psf model provided better results than the Pillbox, and for higher blur levels the Pillbox proved more accurate. The algorithm was implemented on their SPARCS camera system, and provided an accuracy of 3.7% RMS error for auto-focusing applications over a distance of 0.6m to infinity. For ranging application, the RMS error was 4% at 0.6m and linearly increased to 30% at 5m distance.

In 1995, Xing and Shafer [50] [54] used a large bank of Moment filters to estimate the depth information of the scene. Moment and Hyper-geometric filters were narrow band and hence estimated the spectral power at a large number of individual frequencies. The recursive properties of the filters allowed the effects of finite width windows and fore-shortening (caused by non-stationary transformation between two images) to be explicitly analysed and eliminated [54]. Two variants of their algorithm were proposed: - (1) Moment filters without slope estimation (MFF1); and (2) Moment filters with slope estimation (MFF2). Both the techniques were compared with Subbarao's frequency domain method [6]. It was reported that the RMS error of the estimated depth map using Subbarao's method was 4 times higher than that of MFF1 and 27 times higher than MFF2. Though the method provided good accuracy, from the computational perspective, since the filters required more logic support, the method was not suitable for real-time depth estimation [14].

In 1998, Watanabe and Nayar [14] provided an improvement to the existing methods [1] [2] [6] [50] by considering the normalised $\frac{M}{P}$ ratio of the defocused images (amplitude ratio between the difference of the amplitude of the defocused images to the sum) instead of the conventional amplitude ratio. A set of broadband filters were designed in the frequency domain that accurately modelled the $\frac{M}{P}$ ratio curves. Since the filters were broadband in the frequency domain they were narrow-band in the spatial domain and hence suitable for real-time implementation. A Pillbox psf model was considered for the implementation and four 7x7 2D texture invariant filters (including a pre-filter) were designed to effectively retrieve the depth information. It was reported that the depth detection error was less than 1% irrespective of the texture frequency. The depth accuracy was between 0.5% and 1.2% with respect to the distance from the lens. Though real-time implementation was not presented, the authors have claimed that by using their customised Datacube MV200 pipeline processor, the algorithm can deliver six depth maps of size 512 x 480 pixels in one second.

The magnification variation between the defocused images was addressed by Watanabe and Nayar [41] by employing telecentric optics. An aperture stop was introduced at the front focal plane of the lens and a FFT phase based local shift detection method was employed to detect the magnification changes. The magnification was reduced to 0.03% from 3% (reported by Subbarao in [8]) by employing the telecentric aperture.

Raj and Staunton [87] proposed a technique based on Phase Correlation [28] [29] [30] to determine the magnification change between the defocused images. The method considered the magnification change within the sub-block as a local translation problem and estimated the shift by inverse transforming the normalised Cross Power Spectrum. The approach was more practical and robust to noise than Watanabe's method which determined the shift by fitting a plane to the noisy phase data [29].

Recently, Favaro and Duci [64] proposed two methods that exploited the results of Fourier analysis and Singular Value Decomposition (SVD) in the frequency domain to estimate the depth and the radiance of the scene. In their first method they considered the psf as a 3D Gaussian function and represented the imaging model as a convolution between the 3D psf and the transformed volume density (depth estimate). The method required a dense set of defocused images, usually more than 100 and employed deconvolution techniques [47] to estimate the depth. The maximum achievable accuracy for the given setup conditions can be determined directly from the model which was based on the camera settings, the number of input images and the resolution of the image. The second method considered the linearity of the imaging model and employed the SVD in the frequency domain to estimate the depth based on a least squares solution. The method required less than 5 defocused images and was stated to be efficient for practical purposes. For both the methods the radiance of the scene was reconstructed from the additional information provided from the geometry and photometry of the imaged scene. Though theoretical results were not provided, the authors have compared the results with their existing algorithm based on the least squares solution described in [66]. The depth maps and the radiance of the scene were recovered reasonably accurately.

2.1.2.2. Spatial Domain Techniques

In 1993, Ens and Lawrence [12] proposed a Spatial domain technique based on a matrix regularization approach to recover depth information from two defocused images. Their method was stated as an alternative approach to that of the inverse filtering methods advocated by Pentland in [1] [2], where windowing effects are more pronounced. They approached the problem by identifying the psf, $h_3(x,y)$ such that

$$h_1(x,y) \otimes h_3(x,y) = h_2(x,y) \quad \text{--- (2.3)}$$

where $h_1(x, y)$ and $h_2(x, y)$ are the psfs of the two defocused images and $h_3(x,y)$ is the convolution ratio of the defocused operators $h_1(x,y)$ and $h_2(x,y)$ or the extra defocus that is required to make $h_1(x,y)$ equal to $h_2(x,y)$. The estimated $h_3(x,y)$ provided a unique indicator of the required depth. The authors presented three methods to recover $h_3(x,y)$, where their most general solution, iteratively searches for

the best pattern of $h_3(x,y)$ from a pre-computed lookup table that minimized the objective function

$$\sum_{x=0}^{n-k}\sum_{y=0}^{n-k}[i_1(x,y)\otimes h_3(x,y)-i_2(x,y)]^2 = \min \text{ --- (2.4).}$$

Here $i_1(x,y)$ and $i_2(x,y)$ are the two defocused images. The lookup table was derived based on the theoretical or experiment models of the psf. Results with theoretical psf models resulted in an RMS error of 1.7% in terms of distance but reduced to 1.3% when an experimental psf was used. The disadvantages of the method are that it was based on a smoothness assumption and it was computationally intensive [8].

An improved psf measurement technique was proposed by Claxton and Staunton [49]. The method employed a knife edge technique, where a super resolution Edge Spread Function (ESF), obtained by imaging a knife edge on a light box was differentiated to provide a more accurate model of the psf. The method proved simple and effective for shift invariant DFD models since the psf was averaged over the entire length of the edge. Three different psf models (Pillbox, Gaussian and Generalised Gaussian) were considered, and it was observed that the Generalised Gaussian model performed better over a wide working range with different aperture settings. The mean square error (MSE) of the fit of the psf using the Generalised Gaussian model was 8 times better than the Gaussian model and 14 times better than the Pillbox model.

Subbarao and Surya [8] actively employed their Spatial Domain Convolution/ Deconvolution Transform (S Transform) to effectively recover the depth information of an object in the spatial domain. Their method referred as ‘S’ transform method (STM) required only two or three blurred images and provided results that were comparable to Depth from Focus techniques. The forward ‘S’ transform expressed the defocused image as a two variable cubic function using Taylor’s series (equation (2.5)), and the inverse ‘S’ transform (deconvolution operation) which provided the focussed image was obtained by subtracting a constant times the Laplacian of the blurred image from the blurred image, as given in equation (2.6)

$$g(x,y)=\sum_{0\le m+n\le 3}\frac{(-1)^{m+n}}{m!n!}f^{m,n}(x,y)h_{m,n} \text{ --- (2.5)}$$

$$f(x,y) = g(x,y) - \frac{1}{4}\sigma^2\Delta^2 g(x,y) \text{ --- (2.6).}$$

Here, *g(x,y)* is the defocused image, *f(x,y)* the focused image, $h_{m,n}$ is the rotationally symmetric psf, σ the second central moment of the point spread function and Δ^2 is the Laplacian operator. In practice the images captured using different camera settings (as explained in [6]) were approximated as focussed images within a small neighbourhood of 9x9 pixels and expressed as

$$f_1(x,y) = g_1(x,y) - \frac{1}{4}\sigma_1{}^2\Delta^2 g_1(x,y) \text{ --- (2.7), and}$$

$$f_2(x,y) = g_2(x,y) - \frac{1}{4}\sigma_2{}^2\Delta^2 g_2(x,y) \text{ --- (2.8).}$$

The depth was obtained by comparing the approximated focussed images as given by

$$f_2(x,y) - f_1(x,y) = \frac{1}{4}(\sigma_2^2 - \sigma_1^2)\Delta^2\left(\frac{f_2(x,y) + f_1(x,y)}{2}\right) \text{ --- (2.9).}$$

where σ_1^2 and σ_2^2 refer to the second central moment of the psf, and $f_1(x,y)$ and $f_2(x,y)$ refer to the approximated focused images. The initial assumption that the focused image should be modelled as a cubic polynomial was relaxed by using a generalized 'S' transform which incorporated the use of smoothing filters proposed by Meer and Wiles [11]. The estimated σ was then linearly related to the inverse distance as given by the equation (2.2). Two versions of STM were implemented. In STM1, the diameter of the aperture was fixed and two images were taken by changing the lens position. The percentage error in terms of distance was about 2.3% at 0.6m and it linearly increased to about 20% at a 5m distance. In the STM2 the lens position was fixed but the diameter was changed and the RMS error estimated was similar to that of STM1. In the case of 3D objects the error depended on the shape and appearance of the objects. For objects with small depth variations STM calculates the average distance of the objects in the scene. Results on auto-focusing experiments suggested that STM performed better and faster for medium levels of blur, and the DFDIF [7] method proved more effective for higher levels of blur [10].

A continuation of Subbarao work was carried out by Ziou and Deschenes [15] [21]. They approached the problem through a local image decomposition technique using higher order Hermite Polynomials, and demonstrated that any coefficients of the Hermite Polynomial that were computed from a more defocused image can be

expressed as a function of the partial derivatives of the other image. Their 2D model involved the calculation of blur difference β which was obtained by solving four mathematical equations and determining the 'best β' through error analysis. The estimated β was then related to the inverse object distance as given in equation (2.10).

$$\frac{1}{z} = \frac{1}{F} - \frac{1}{2v+\delta v}\left(1+\sqrt{1+\frac{f^2(2v+\delta v)}{F^2\delta v}k^2\beta^2}\right) \text{--- (2.10)}$$

where z is the object distance, β is the blur difference between the defocused images and v, F, f, k are the camera parameters. Tests were performed on step edges, line edges and on junction like L, V, T, Y and X, and compared with Subbarao 'S' Transform method [8]. It was observed that the latter method was capable of estimating the blur only at line edges. At step edges and on junctions the 'S' Transform failed since Laplacian of Gaussian was zero at these points [23]. The RMS error reported for a planar object whose furthest point was at 125cm and the nearest at 115cm was 2.21% against 4.22% for Subbarao's 'S' Transform method. The depth densities for the methods are 97.4% and 85.3% respectively. Considering the spatial errors involved in camera movements while image acquisition, Deschenes *et al.* [22] extended their Hermite Polynomial model to include the spatial shifts; horizontal, vertical, zooming and 2D motion. The RMS error reported was 1.68% with a depth density of 100%. An improvement of the above method was proposed in [61] where the spatial shifts and the zoom disparities were simultaneously computed along with the blur using a Homotopy based approach with several higher order derivatives calculated for the image.

In 2004, Simon *et al.* [58] proposed a method similar to Subbarao [5], where the spread parameter σ of the Gaussian psf was estimated by considering the ratio between the sharp and the blurred edges of the images (gradient ratio). A generalised model was proposed to estimate the gradient ratio for entire thick and thin edges. A Prewitt edge detector was employed to determine the gradient ratio and the direction of the edge. The spread parameter σ computed from the gradient ratio, was later related to the depth. The main drawback for the method was the acquisition of a sharp image which required additional lighting conditions. This problem was later addressed by them in [59] where three blurred images were used to recover the

spread parameter; however this introduced additional complexity in the image acquisition and increased the processing time of the algorithm.

Leroy *et al.* [60] extended the work of Simon *et al.* [58] [59] and proposed a simpler algorithm which required only two defocused images. Their work was based on Subbarao [8] and Deschenes [15] [21] [22], where the magnitude of the Laplacian gradient at the edges for step, ramp or roof was computed to determine the depth. Though real-time implementation was not presented, the authors have stated the algorithm could compute a depth map of 800 x 600 pixels in 23ms. The maximum mean depth error reported was 20.05mm between a range of 790mm and 990mm. The main drawback of the method was the influence of the edge density and the characteristic of the image textures on the accuracy of the estimated depth. It was stated in [60] that the edges with high density provided more accurate depth results.

In 2007, a neural network based technique was suggested by Jong [81] which estimated the spread parameter σ of the Gaussian psf in the spatial domain. The model was based on a supervised learning network that employed the Radial Basis Function (RBF). The RBF was preferred over a Back Propagation network (BPN), since it provided a better approximation to a continuous function [82]. Experiments were performed on edges with objects placed between 220mm and 355mm. A 5% error relative to the object distance was reported.

The above mentioned techniques (except Favaro's and Chaudhuari's) considered the imaging model as a linear shift invariant system and expressed the defocused image as a convolution between the focused image and the shift invariant psf [53]. However, Tu *et al.* [83] proposed a technique based on inverting the shift variant blur model in the spatial domain to recover the depth and the focussed image from two defocused images. The method was an extension of Subbarao's 'S' Transform approach [8] for shift invariant blur models, and incorporated Subbarao's Rao Transform [84] [85]. It was developed primarily for image restoration, and used a linear integral equation. The algorithm was based on an exhaustive search strategy to find the 'best shape' parameter that minimises an error function. Experiments with simulated images suggested a maximum error of 3% with respect to the distance.

Though experiments were performed on real images, the theoretical details about the depth accuracy were not presented.

2.1.2.3. Statistical Techniques

Rajagopalan and Chaudhuri [48] applied the Space Frequency Representation to the problem of DFD. Their approach was to determine the shift variant blur parameter by calculating the blur difference of the defocused image using the complex spectrogram (CS) and pseudo-Wigner distribution (PWD). Since the complex spectrogram and Wigner distribution estimated the blur independently of the neighbouring pixels, the recovered depth map was quite noisy with large depth discontinuities. Hence a variation approach with smoothness constraint was proposed where the degree of smoothness of the estimated blur at each pixel was governed by a regularisation parameter. Experiments showed that the algorithm provided a smooth depth map with less depth variations. The RMS error reported was 4.84% for the scene whose farthest point was at 115cm from the lens surface. In 1999, they proposed an algorithm [19] where the shift variant blur parameter was modelled as a Markov Random Field (MRF) and the depth information along with the focused image was simultaneously recovered from a pair of defocused images. The algorithm was based on minimization using the Simulated Annealing technique and the recovered depth map was compared with Subbarao's Fourier domain method [6]. It was observed that the Fourier method estimated a noisy depth map with a RMS error of 5.76cm compared to the proposed method were the RMS error was only 3.02cm. However, in terms of speed the proposed method was less suitable for practical purposes, since it incorporated minimization techniques.

In 2003, Rajan and Chaudhuri [18] extended their earlier results based on MRF to recover depth estimates at higher spatial resolution, thereby generating a super-resolved image of the scene. They modelled two separate MRFs: - (1) To represent the shift variant blur parameter; and (2) To represent the intensity field. Again the Simulated Annealing technique was used to simultaneously recover the Maximum a Posteriori (MAP) estimates of the high resolution spatially variant blur and the super-resolved image. Experimental results showed an RMS error of 1.76cm equivalent to a ranging error of 1.96% when the farthest block was at 96.6cm.

In 2000, Favaro and Soatto [44] developed an iterative algorithm based on the minimization of Information Divergence between the defocused images. It recovered both 3D shape and the radiance for the scene. They extended the results of Ceizar [46] and Snyder *et.al* [47], and analyzed equifocal imaging models where the psf was considered to be translation invariant. The minimization was done using a descent method, and it was observed that the Information Divergence decreased for any kernel model satisfying the positivity and smoothness constraint. Though theoretical results were not provided, the depth maps generated were quite dense as the iteration progressed. In [45], Favaro and Jin considered the 3D shape and radiance recovery as an infinite dimensional optimization problem, and recovered the global shape of an object instead of the depth. This proved an improvement in terms of computation since the radiance of the overlapping regions was not required to be recomputed. Their algorithm was not restricted to equifocal imaging models where the scene to be recovered was assumed to be parallel to the focal plane. This was claimed as an advantage over other existing models [1] [2] [8] [14] where the psf was assumed to be translation invariant.

Favaro and Soatto [17] [36] proposed a novel algorithm based on matrix multiplication which was relatively insensitive to the psf, and recovered the 3D geometry and radiance of the object. In one of their models, if the complete characteristic of the psf was known then the orthogonal operators required for the 3D geometry were computed using functional Singular Value Decomposition (SVD) of a small window of 7x7 or 9x9 pixels of the defocused images. If the characteristics of the psf were unknown, then the orthogonal operators were obtained by a learning process, from a collection of blurred images acquired over a finite dimensional range. The two main features of the algorithm were its robustness to noise and its feasibility for parallel implementation. It was also efficient in the way that orthogonal operators obtained from set of simulated images can be effectively used to recover depth of real objects. Experiments were carried out between distances of 520mm and 850mm with 51 equifocal planes simulated with randomly generated scenes. The known psf variant of the algorithm provided an average depth estimation error of 31mm and when an unknown psf variant of algorithm was used, the depth error was reduced to 27mm. For 3D objects, the authors have provided visual depth

maps comparable to the depth maps recovered using Watanabe's method [14] with no accuracy information.

In 2003, Favaro *et al.* [68] proposed a novel algorithm to estimate the depth by inferring the diffusion coefficient of an anisotropic heat equation. The method employed a forward heat equation that determined the diffusion (match) required between the two defocused images. A gradient regularisation technique was used to estimate the diffusion coefficient that provided a dense depth map (one-to-one) of the scene. Later, texture mapping was adopted to recover the radiance of the scene. The shape and the radiance estimated were quite favourable, and one variant of their algorithm was used for an application which involved the 3D shape segmentation of the scene. Namboodiri and Chaudhuri [78] analysed Favaro's diffusion model and stated two main drawbacks:- (1) The model cannot handle a departure from the Gaussian blur model assumption in the case of self-occlusions; and (2) The diffusion coefficient was assumed to be a convex function. Further, the gradient regularisation employed by Favaro resulted in overly smooth depth estimates [79]. Subsequently, the drawbacks were addressed by Namboodiri and Chaudhuri [78], where a stochastic blur model was incorporated into the heat diffusion equation to handle the variations (due to lens and aperture deformation) from the standard Gaussian blur model. Experiments with real images suggested better results than Favaro's [68] at self-occluded points.

In 2007, Namboodiri and Chaudhuri [80] used the linear diffusion heat model in the frequency domain and proposed an "Extended Defocus Space Model" that presented the equivalent means of estimating the depth from the known lens parameters either using DFD or DFF techniques. The model provided poor depth estimates at homogeneous regions and necessitated a suitable regularization function. These demerits were later addressed in [78], where the Markov Random Field was used to model the diffusion coefficient, ensured robustness and spatial regularisation of the estimated depth.

2.1.2.4. Wavelet based Techniques

Wavelet transforms were used to recover the depth estimates by Hor *et al.* in [55] and Choi *et al.* in [56] [57]. The method was considered as an alternate to the frequency and spatial domain approaches, since they suffer either from windowing problems (as in frequency domain approach [12]) or from frequency uncertainties (as in spatial domain approaches [55]). In wavelet analysis, the spatial and frequency content variations are localised in the phase domain and hence maximum resolution is obtained both in space and in frequency [55].

Hor *et al.* [55] considered the defocus as a spatially variant blur model and proposed spatial variant transform analogues to the Fourier transform model. The modulated Gaussian function was chosen as the mother wavelet and the standard deviation extracted after applying the wavelet transform was used to measure the depth of the scene. Though quantitative results have not been provided, the method recovered the visual depth map quite favourably.

In [56] [57], Choi *et al.* estimated the blur parameter σ of the Gaussian psf by considering the ratio of wavelet powers of the defocused images. Parseval's theorem was employed to measure the energy (power) using the wavelet coefficients which were outputs of the wavelet transform [56]. The estimated power ratio later provided the required depth information when substituted into their design model. Experimental results with a slanted planar object demonstrated that the recovered depth using wavelets had a lower RMS error of 0.8181cm when compared to methods such as Fourier, Spatial and Laplacian, where the RMS errors were 2.119 cm, 1.3251cm and 1.8517cm respectively. The working range of the experiments was between 150cm and 180cm.

2.1.2.5. Fuzzy Logic based approach

A fuzzy logic DFD method was suggested by Swain *et al.* [86] to improve the accuracy of the depth estimates. The model required two inputs: - (1) The focus quality, which determined the amount of defocus present in the images; and (2) The focus error, which measured the difference in focus between the corresponding

points in the image. A Sobel edge detector was employed to measure the focus quality (method was referred as Tenengrad), and a 3x3 Laplacian operator was used to calculate the focus error. The images after applying the Laplacian operator were normalised for brightness and compared to provide the required depth estimates. The measured depth estimates were fed to a fuzzy logic algorithm which provided the necessary depth corrections. The membership functions of the fuzzy logic algorithm are carefully determined through trial and error. Experiments reported a depth error of less than 1.5% over a working range of 2133mm to 3352mm. The following drawbacks were reported: - (1) Test images should contain high frequencies; (2) The window selected for depth estimation should have a single depth and (3) The membership function of the fuzzy logic set was required to be tuned for different camera settings, which was time consuming and based on trial and error.

2.1.2.6. Reverse Projection Correlation principle for Depth from Defocus

In 2006, McCloskey *et al.* [62] approached the DFD problem by considering the correlation between the adjacent pixels of the blurred images. Their motivation was based on the observation that the pixel transfer function increases as the scene gets further away from the plane of focus. This resulted in an increase in the correlation between the adjacent pixels. The change in correlation coefficient (CC) between the adjacent pixels was measured and a look-up table was used to relate the CC to the blur radius of the scene. Later, the blur radius was related to the actual depth using the equation (2.1). For experiments, the authors have constructed a look-up table by considering 25 pairs of images acquired with different combinations of depth and viewing angle, and the change in CC was quantized into bins of width 0.005. The RMS error in terms of absolute depth for the simulated images was between 0.4% and 0.8%. For real images, the change in CC was determined from an image window of 51x51 pixels. The authors have presented a 1D cross Sectional view of the estimated depth with no theoretical information about the accuracy.

2.1.2.7. Depth Estimation by change in Zoom

Depth estimation based on a change in the zoom settings of the lens was proposed by Baba *et al.* [63]. They observed that a change in zoom resulted in a change in blur circle radius similar to the variations in focal length and aperture of the lens. A thin lens zoom model was proposed that related the effective focal length and the effective aperture diameter of the lens. I.e. the relation between zoom and the focus was described in terms of effective focal length, and the relationship between zoom, focus, and aperture was described in terms of effective aperture diameter. From their model, they observed that the estimated blur width changed in proportion to the square of the effective focal length which was in-turn related to the zoom control value. In experiments with scenes having single depth, the object was at 1250mm and the blur width was measured from 192 images, with 4 aperture levels, 8 focus levels and 6 zoom levels. The mean distance estimated was 1236mm with a high standard deviation of 27.4mm. For multi-zoom images where the depth was estimated from the change of blur width from a continuous change in zoom, the mean depth estimated was 1233mm with a reduced standard deviation of 19.8mm. Experiments with multiple targets placed at several depths resulted in a maximum error of 1945.9mm when the target was at 3000mm. The experimental results provided by the authors were based on measuring the blur width at the edges of the objects and details about their dense depth recovery were not presented.

2.2. Active DFD Methods

Active DFD methods are effective when depth analysis is performed on weak or texture-less surfaces. The idea is either to project an illumination pattern on to the object under investigation and measure the defocus by comparing with the focused pattern, or by modeling a filter that responds to the single dominant frequency of the projected pattern. Though active methods provide accurate depth measurements they need controlled illumination and sophisticated pattern fabrication techniques. This Section describes existing DFD methods based on active illumination.

Pentland *et al.* [3] pioneered active DFD, where a standard slide projector was used to project a structured light pattern on to the object that required measurement. To

avoid distortions, the light source was projected along the optical axis and a video camera was used to capture the blurred images. Depth was estimated by comparing each point of the defocused pattern to the known focused pattern, and a lookup table was used to relate the energy within the blurred region (hump energy) [3] to the radius of the blur circle r_c , as shown in Figure (2.3).

With active illumination, a 0.5% RMS error was reported for 64 x 64 resolution depth maps. A stroboscopic extension to active DFD was constructed to measure depth of fast moving objects, where the strobes replaced the slide projector. These strobes were synchronized with the video camera such that alternate frames were illuminated with structured light and white light. An example of depth recovery of a rolling golf ball was presented in [3]. The RMS error of 5% was reported for this experiment.

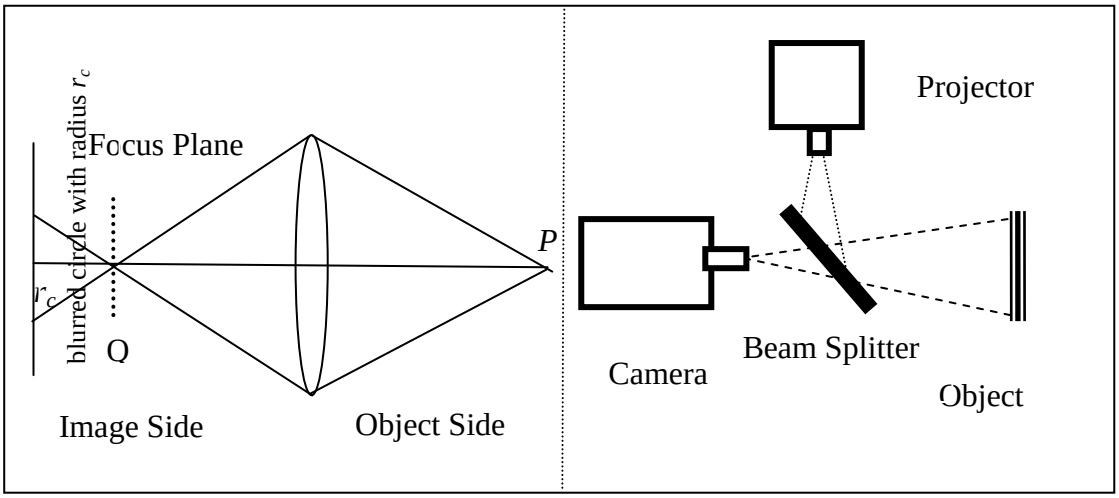


Figure 2.3: Active DFD method based on Pentland [3] (left) ray diagram, (right) optical setup

Nayar *et al.* [13] developed a prototype range sensor based on active illumination that generated 512 x 480 depth maps at 30 frames per second with an accuracy of 0.3% relative to the object distance. The illumination pattern was accurately determined using optimization techniques that maximised the accuracy and robustness of the depth estimation. The designed pattern was then fabricated on to the sensor using micro-lithographic technique. The pattern developed was a checkerboard with a horizontal and vertical period of t_x and t_y, such that $t_x = 4p_x$ and $t_y = 4p_y$. Here p_x and p_y are the CCD pixel pitch in the horizontal and vertical directions. A 5x5 Laplacian kernel was used as a tuned focus operator that responded

to the single dominant frequency ($1/t_x$,$1/t_y$) corresponding to the pattern frequency. The depth was estimated using the normalized ratio (ratio to the difference of amplitude to the sum of amplitude of the defocused images) and a loop-up table was employed to relate the normalized depth to the actual distance. A detailed description about the hardware implementation using a MV200 Datacube pipelined processor and the experimental results are presented in [13]. Further, the illumination pattern was incorporated into a Microscopic Shape from Focus system [74] [75], and was effectively used to recover the shape of a silicon substrate with 13µm features, and solder joints that were 150µm high and 100µm wide.

Ghita and Whelan [70] [76] developed a video rate sensor based on active illumination that processed 10 frames of size 256 x 256 pixels in a second. The imaging setup was considered to be linear shift invariant and the blurring effect was modeled as the convolution between the focused image and the Gaussian psf. A linear interpolation technique along with a strip grids pattern of density 10 lines/mm was employed to avoid the expensive fabrication technique required to determine the illumination pattern described in [13]. The performance of a Laplacian kernel (4 and 8 neighborhood), and Watanabe's rational filters (3x3 and 7x7) [14] as focus operators was investigated, and it was reported that the Laplacian (4) and rational filters of size 3x3 provided more linear depth estimate compared to the rational filters of size 7x7 and the Laplacian (8). The lowest accuracy achieved was 3.4% normalized with respect to the distance. In [77] a bin picking system based on this active DFD technique was presented.

A neural network based depth detection technique with added illumination was proposed by Li Ma and Staunton in [71]. In their algorithm, the object was first isolated from its background and the depth was estimated using a three layered neural network designed using the Back-Propagation algorithm. A multi-resolution segmentation algorithm, which included three sub-modules (image pyramid formation, linkage adaptation and unsupervised learning) were effectively used to segment the object from its background with an error of 0.637%. Though the model was trained with checkerboard images, it also effectively recovered the depth map of images with natural textures. High resolution data was used by the authors to maximize the depth accuracy.

2.3. Discussion

The chapter provides a classification of DFD techniques based on the method used and the mode of operation. Most approaches consider blurring as a linear shift invariant process (frequency and spatial domain) and represent the defocused image as the convolution of the focused image with the psf of lens. The blur information was then retrieved by the deconvolution process either in the frequency or spatial domain, and then related to the actual distance using the appropriate depth model. These methods offer an advantage in terms of computation and simplicity in implementation of the algorithm. The other methods (mostly statistical methods) consider the blurring as a shift variant process and retrieve a unique depth value not only along the optical axis but also along the x and y directions of the scene under investigation. These methods prove efficient since they simultaneously retrieve depth and the radiance of the scene, but are not suitable for practical purposes since they are based on error minimisation techniques which require extensive computations. Since the objective of this research was to develop a real-time depth estimation system that can be effectively implemented on a Field Programmable Gate Array (FPGA) with a usage in medical and industrial applications, the DFD methods require two images to recover the depth and hence be useful for real-time depth estimation. In terms of accuracy, DFD methods are comparable to DFF techniques [8] and require less processing time. Simon *et al.* [58] [59] suggested a three image technique where the blur parameter was recovered from three blurred images, but this in-turn introduced additional complexity in the image acquisition process and also failed to show good depth results [60]. After an in-depth analysis into different methods, the technique described by Watanabe and Nayar [14] based on the use of texture invariant broadband filters was chosen for implementation. Though the filters were designed in the frequency domain, the algorithm can be implemented in the spatial domain by employing five 2D convolutions and thus should be suitable for real-time implementation. In terms of accuracy, the maximum RMS error reported was 1.2% with respect to distance (which was better than comparable methods), with a depth detection error of less than 1% irrespective of the texture frequency. The main drawback of the method was the requirement for a less complicated procedure to model the rational filters for any given defocus condition. This problem was

subsequently addressed in this research work where a novel method referred as the 'Two Step Polynomial Approach' was employed to design the rational filters (refer to chapter 4). To provide a good accuracy comparison with Watanabe's filters the algorithm was based on the Pillbox psf model, rather than Gaussian or Generalised Gaussian suggested by Claxton and Staunton [49]. Further the Pillbox psf model is a good approximation of a more blurred image [49] and also provided better depth results for highly blurred images as stated by Subbarao [7] [10]. The 1D equation for each of the three psf's are presented in Chapter 4. New research presented in this thesis also addresses: - (1) An algorithm to estimate the magnification variations between the defocused images (Chapter 3); and (2) The implementation of the DFD algorithm on the Virtex 2P FPGA (Chapter 5). Experimental results and comparison with Watanabe's filters are provided in these chapters.

CHAPTER 3

Estimation of Image Magnification using Phase Correlation

Introduction

One of the fundamental tasks in Image Processing is to acquire a set of images which are registered with each other but in practice this is not always possible since changes in image acquisition parameters cause misalignment. In order to compare the acquired images, the shift, rotation and scaling between the images needs to be determined. Once these differences have been estimated they can be used to correct the position of one image relative to the other. In this chapter a new method is described which was devised to effect this, as problems arise with standard methods when images have been defocused. To increase the accuracy of the depth estimation, the defocused images (near and far-focused) must be registered to compensate for magnification, and in practice, translation. Since the depth measurement method was based on Watanabe and Nayar [41], an optical method using telecentric optics was used to correct the magnification changes. This method requires the precise placement of an external aperture at the front focal plane of the lens. The method is readily suitable for real-time depth estimation since it avoids the use of any interpolation technique for registering the image and is achieved using a setup prior to depth estimation. The chapter discusses an effective technique based on Fourier analysis to measure the magnification changes between the near and the far-focussed images. Section 3.1 provides an overview of the various image registration techniques, followed by telecentric optics (Section 3.2) in which a comparison is provided between the conventional lens and telecentric lens model. Sections 3.3 and 3.4 explain the algorithm for image magnification measurement and finally Section 3.6 provides experimental results for simulated and real images.

3.1. Overview of the Image Registration Techniques

3.1.1. Correlation Techniques

The Correlation technique provides a statistical measure between the image and its template. It is useful in template matching applications [24] [25]. For example let $I(x, y)$ be the principal image and $T(x, y)$ be the template that needs to be matched then the 2D normalized Cross Correlation function can be found using the equation:

$$C(u,v) = \frac{\sum_x \sum_y T(x,y) I(x-u, y-v)}{\sqrt{\left[\sum_x \sum_y I^2(x-u, y-v) \right]}} \text{ --- (3.1).}$$

If the original image and the template were identical and were translated by a spatial shift (*i,j*), then the normalised Cross Correlation function, *C(u,v)* would include a peak at the spatial location (*i,j*) indicating a match. Hence by computing *C(u,v)* over all possible coordinates, the similarity measure between an image and its template can be determined. A related statistical measurement, which is advantageous when absolute measurement is needed, is the correlation coefficient. It measures the similarity between the template and the original image on an absolute scale ranging from -1 to +1. A registration algorithm incorporating correlation would first determine the cross correlation at each transformation and then relate the largest measure as an indication to the similarity between the images [25]. Barnea and Silverman [26] proposed the Sequential Similarity Detection Algorithm [SSDA] which provided an improvement over the conventional method. In their algorithm the similarity measure was determined by computing the absolute differences between the pixels of the two images that needed to be compared. A threshold based sequential search method also was introduced to reduce the number of required computations. Although the correlation technique is widely used in image registration, it has limitations. In cases where images taken with different brightness levels are to be registered, the peak of the normalised Cross Correlation is not uniquely defined causing ambiguity in the matching process. Furthermore, the computation cost is directly related to the number of transformations (translation, rotation, angle), which makes this method computationally expensive. For these reasons, methods based on Fourier Transforms and Point Matching are generally preferred [25].

3.1.2. Fourier Domain Techniques

When Fourier domain techniques are used for image registration, the images are transformed to the frequency domain and the mathematical properties of the Fourier Transform are used as parameters for the image registration algorithms. By using the Fast Fourier Transform the computation time for an image of size n x n is reduced

from O (n^4) to O (n^2log (n)) thus making it ideal for real-time application. Usually the Fourier techniques are based on the shift property of the Fourier Transform i.e. if a function is shifted in the positive direction by an amount *a*, the amplitude of the Fourier spectrum remains the same but changes are present in the phase component. Each frequency component of the spectrum is delayed in phase by an amount proportional to the frequency i.e. the higher the frequency, the greater is the change in phase angle. The linear change of phase with the frequency is given by the constant $2\pi a$, where *a* represents the shift. Hence the greater the shift, the greater is the rate of change of the phase for a given frequency [27]. This property of the Fourier Transform can be applied to determine the shift between two similar images. Kuglin and Hines [28] proposed a method called Phase Correlation to align images. In principle if the images f_1 and f_2 differ by a shift (x_0, y_0) then f_2 can be expressed as

$$f_2(x, y) = f_1(x - x_0, y - y_0) \qquad \text{--- (3.2)}$$

the corresponding Fourier domain relationship is

$$F_2(\varepsilon,\eta) = e^{-j2\pi(\varepsilon x_0 + \eta y_0)} * F_1(\varepsilon,\eta) \quad \text{--- (3.3)}$$

where F_1 and F_2 are the Fourier Transforms of the images f_1 and f_2, and $e^{-j2\pi(\varepsilon x_0 + \eta y_0)}$ is the phase shift. They define the normalised Cross Power Spectrum of the two images as

$$\frac{F_2(\varepsilon,\eta) * F_1^*(\varepsilon,\eta)}{\left|F_2(\varepsilon,\eta) * F_1(\varepsilon,\eta)\right|} = e^{-j2\pi(\varepsilon x_0 + \eta y_0)} \text{--- (3.4)}$$

where F^* is the complex conjugate of *F*.

If the images acquired differ only by a translation, then their Fourier Transforms have the same magnitude and the phase component of the normalised Cross Power Spectrum, is equivalent to the phase difference between the images. The Inverse Fourier Transform of the normalised Cross Power Spectrum results in an impulse function that is approximately zero everywhere except at the displacement. The *x* (horizontal) and *y* (vertical) shift of the impulse is used to register the two images. Figure (3.1) shows the original image and its shifted variant. Here the shift induced was, *x* = 50 and *y* = 100 pixels. Figure (3.2) illustrates the results computed by the Phase Correlation function where a sharp distinct peak is seen at the location 50, 100 pixels (x, y axis).

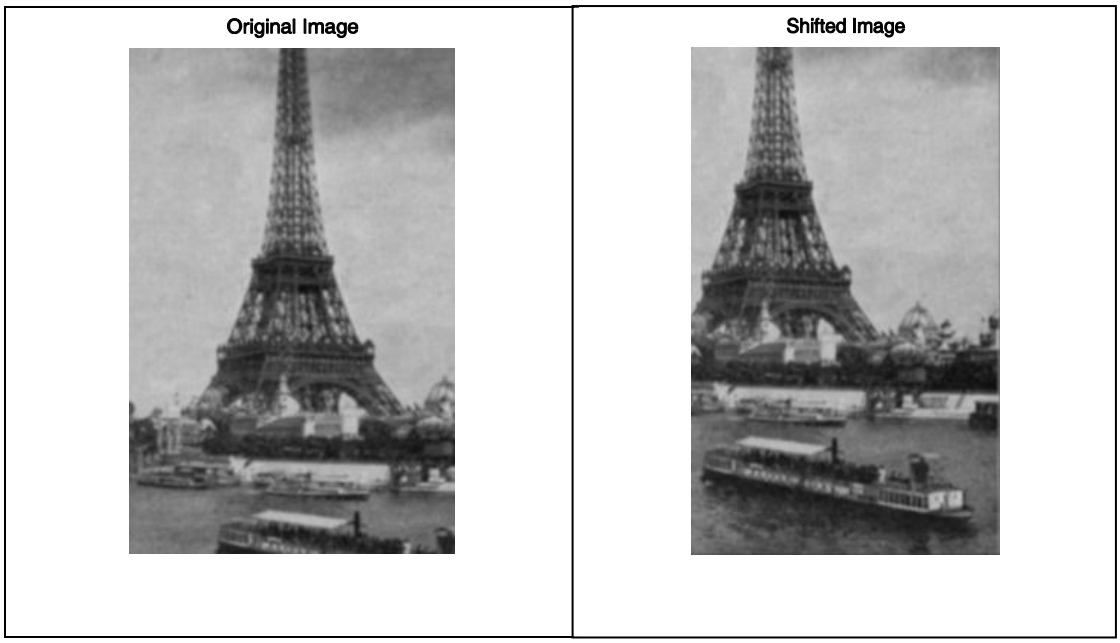


Figure 3.1: Original Image (Right) and the Shifted Image (Left)

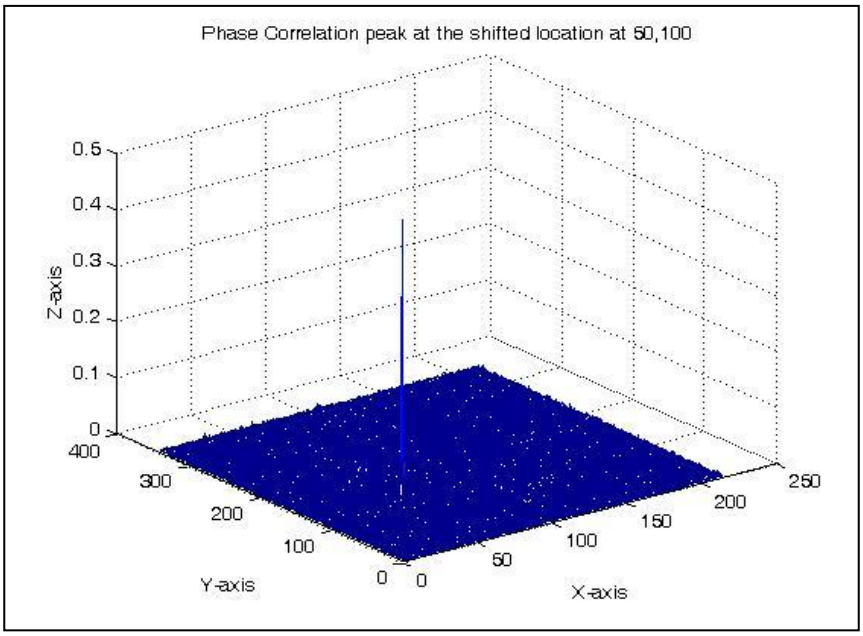


Figure 3.2: Resultant peak at 50,100 pixels computed using Phase Correlation Method

Foroosh and Zerubia [29] extended the Phase Correlation technique to sub-pixel levels. They stated that the signal power of the Phase Correlation function for a down-sampled image is not concentrated in a single discrete peak but rather in several coherent peaks mostly adjacent to each other [29]. They have also reported that the Phase Correlation method provides a distinct sharp peak at the point of registration, whereas standard Cross Correlation yields several broad peaks and the main peak is not always centred at the right point. Further, due to whitening of the signals by normalisation, Phase Correlation methods are more robust to noise that are correlated with the image functions such as uniform variations in illumination, offsets in average intensity and fixed gain errors due to calibration. Takita and Muquit [30] employed an analytical function fitting technique to establish the position of the peak. It was reported in their paper that the Phase Correlation method could estimate displacements between images with an accuracy of 1/100 pixel when the image size was 100 x 100 pixels.

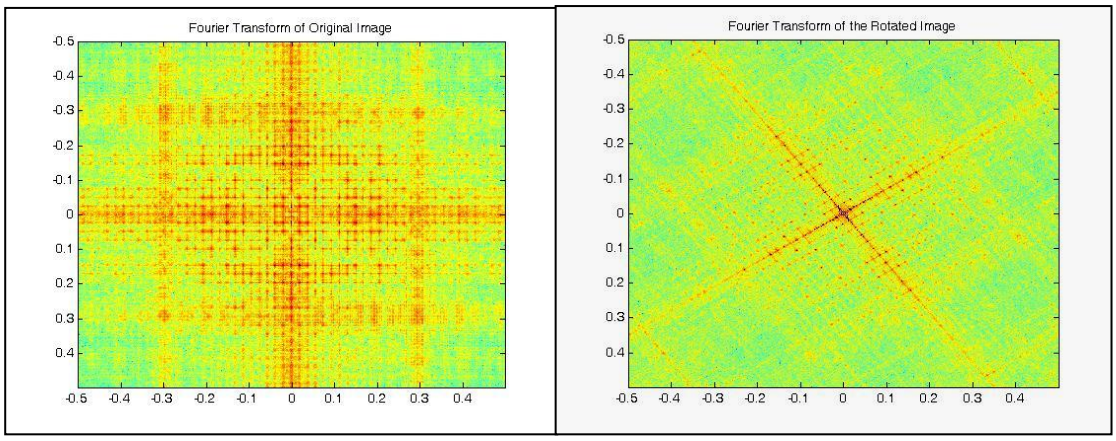


Figure 3.3: Fourier Transform of the original Image (Left) and the Fourier Transform of the Rotated Image (Right)

Rotation is invariant with the Fourier Transform hence rotating the image by an angle rotates the Fourier Transform of the image by the same angle [25] [37]. Suppose if an image *f(x,y)* is rotated by angle θ then the Fourier Transform *F(u,v)* also rotates by a angle θ and is given by

$$f(x\cos\theta - y\sin\theta, x\cos\theta + y\sin\theta) = F(u\cos\theta - v\sin\theta, u\cos\theta + v\sin\theta) \text{ -- (3.6).}$$

Figure (3.3) shows the Fourier Transforms of the original image (left) and its rotated variant (right). It can be inferred from the Figures that the angle at which the image has been rotated can be determined by comparing the Fourier Transforms of both the

images. De Castro and Morandi [31] developed a two step approach to register images that are translated and rotated. The method first uses the polar coordinate variation of the Phase Correlation function to determine the angle of the rotation and then proceeds to find the translation shifts. A similar approach was reported by B.S. Reddy and B.N. Chatterji [32] to register images that are scaled, rotated and shifted. The algorithm provides accurate results to the second decimal place and computes the matching parameters irrespective of the amounts of translation, scaling and rotation.

Though Fourier techniques exploit the mathematical properties of the Fourier Transform, they have their limitations. It is noted that Fourier domain techniques would render inaccurate results if the images have significant white noise spread across the entire frequency band, and since they rely on the invariant properties of the Fourier Transform they are applicable only to well defined transformations like rotation and translation.

3.1.3. Points, Features and Elastic Models

Point and feature based techniques can be used to register images with unknown misalignment [25]. The method works by identifying well defined features or points on the images, and uses interpolation techniques for registration. The control points can be corners, line intersections, identifiable landmarks, or anatomical structures that are recognisable within the image. Since these techniques require sophisticated search strategies, they are computationally expensive and can be demanding if many matching points are used. In one of the methods described in [33] the algorithm first determines all possible matching pairs for the given control point. Then accordingly the matching pairs are rated (weighted) as to how close they are to the actual displacement. Likewise the procedure is performed in parallel for all the control points and the matching pair which has the highest rating provides the optimum displacement. However it was reported in [33] and in [34] that the computational cost of the algorithm was O (n^4), where ‘n’ represents the number of control points defined in the system. In Elastic models, the registration is based on image structure matching [25]. The algorithm was based on an iterative principle and either adopts a piecewise Spline Interpolation technique based on feature mapping, or uses a cost function model to minimise the energy between the deformed image and the similar

image. Point and Elastic methods require clearly identifiable feature or matching points to accurately measure the transformation (rotational, scaling and translation) between the images, but would not be suitable for low accuracy images (here, defocused images) [35]. Moreover point based methods work well when the images have a smooth surface i.e. a scene with smooth depths, however if the scene has large depth variation, then occlusions are more likely and the matching accuracy between the images is low. Here, the objective is to measure the radial shifts due to magnification between two defocused (blurred) images, and thus a registration technique capable of measuring the shifts at low frequencies is an upmost requirement. Based on the survey [25] it was found that the Fourier techniques are specifically well suited for images with low frequency acquired under varying lightning or atmospheric conditions. Therefore a Fourier technique based on the Phase Correlation was employed to measure the magnification change between the two defocused images. The detailed algorithm is presented in Section 3.4.

3.2. Image Magnification Measurement and Correction

Change in the focus setting between the far-focused image i_1 and the near-focused image i_2 (see Figure 3.4) results in an undesirable change in magnification. It was reported in [41] that variations in magnification pose problems for vision techniques like Depth from Focus and Depth from Defocus, but was ignored since in practice the magnification change accounts for less than a 3% error [8]. Though registration can be improved either by camera calibration [38] or by image wrapping [39] techniques, these require computer controlled zoom lenses and very accurate image re-sampling methods. Furthermore, these techniques would introduce smoothing and aliasing. Due to the computational overhead these methods are not applicable for real-time depth estimation. Watanabe and Nayar [41] eliminated the magnification problem optically by converting a conventional lens into a telecentric optic. They introduced an aperture stop at the front focal plane of the lens and reported that this reduced the magnification between the images to 0.03%. They employed a FFT phase based local shift detection method to detect the magnification change between the images. A plane was fitted to the phases of the ratio of the spectra. The gradient of the plane provided an estimate of the shift between the images. Since this proposed research work aimed to determine depth in real-time, the optical method

using the telecentric optics was adopted to correct the magnification change between the images. A novel, practical and more robust technique using Phase Correlation [28] [29] [30] [31] is reported in Section 3.3 to determine the magnification change between the two defocused images and to optimally position the telecentric aperture.

3.2.1. Conventional Image Formation Model

The simple lens model consists of a single lens element with two refracting surfaces. The image formation model is based on the Gaussian lens law [40] with the assumption that the lens is 'thin' (the physical vertex to vertex distance is usually 1/10 of its diameter) and the aperture position coincides with the lens. Figure (3.4) shows the simple image formation model for the Depth from Defocus (DFD) application.

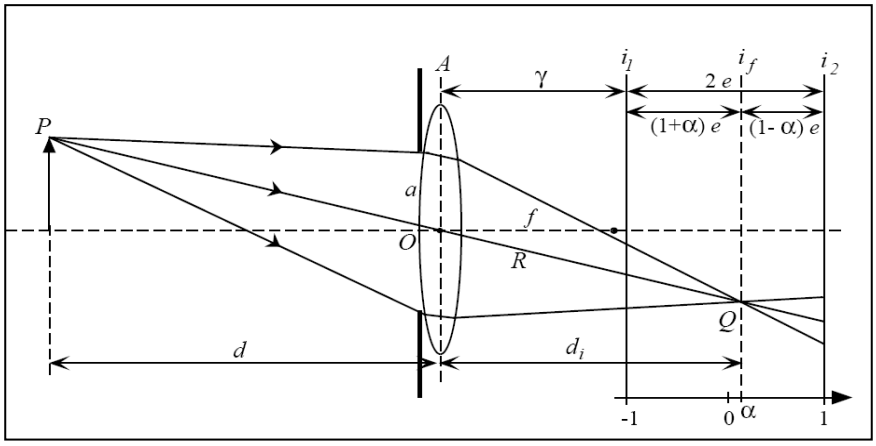


Figure 3.4: Conventional Imaging model for DFD based on Gaussian Optics

The energy flux (radiance) emitted from the point P at a distance d on the object side is mapped on to a point Q in the focussed plane i_f at a distance d_i. The relation between object distance d, image distance d_i and the focal length f is given by

$$\frac{1}{f} = \frac{1}{d_i} + \frac{1}{d} \quad \text{--- (3.7)}$$

In the generalised model of the DFD [2] [14], where the image structure in unknown, depth is calculated from the amplitude ratio of the two defocused images i_1 and i_2 on either side of the focused image i_f. Hence for accurate depth estimation the images i_1, i_2 are needed to be registered in terms of magnification. In a conventional lens model

shown in Figure (3.4), as the image sensor is moved from defocused image i_1 to the defocused image i_2, the location of the point P is displaced along the Principal Ray R. This variation induces image magnification which manifests as a correspondence like problem in Depth from Defocus [42]. Watanabe and Nayar [41] [42] solved the magnification problem by using a telecentric optic as explained in the next Section.

3.2.2. Telecentric Optics

Telecentric Optics differs from the conventional model with an addition of an aperture stop placed either at the front focal or at the back focal plane. If the aperture stop is situated at the back focal plane then the entrance pupil will be at infinity and all the principal rays in the object space will be parallel to the optical axis. This system is termed as telecentric on the object side [43] and most commercially available telecentric lenses are constructed like this [41]. In DFD, regardless of the change in focus, the magnification between the defocused images needs to be constant. This is achieved by placing the aperture stop in the front focal plane, i.e. the exit pupil will be at infinity and the principal rays in the image space will be parallel to the optical axis, thus ensuring constant magnification between the two defocused images. This optical setup is termed as telecentric on the image side and is shown in Figure (3.5) [43].

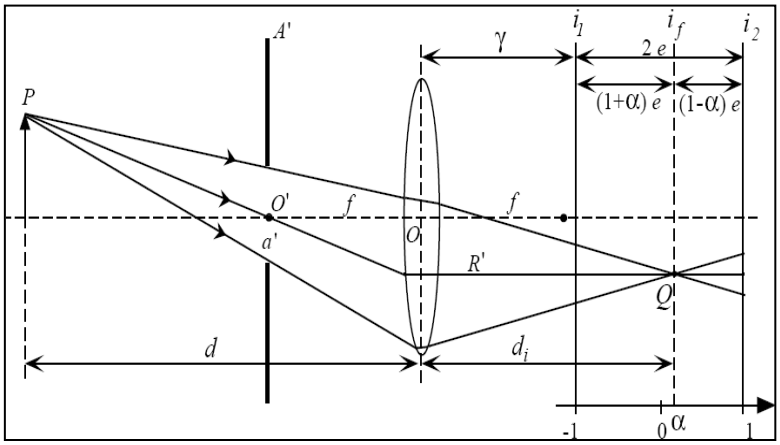


Figure 3.5: Imaging model for DFD based on Telecentric Optics

To determine the front focal plane of the lens the procedure described in [41] was used. The Focal Plane is the surface (plane) on which the image transmitted by the

lens is brought to sharp focus. To find the approximate position of the front focal plane the lens is held between the screen and a distant source, and the light energy from the distant source is made to enter through the rear end of the lens. A focused image is obtained by adjusting the screen. The plane passing through this location perpendicular to the optical axis is termed as the front focal plane. It is at this point the external aperture is placed so that the Principal Ray *R'* becomes parallel to the optical axis. For the experiments a 50mm Nikon lens and 35mm Hannimar lens were converted to telecentricity. In both the cases the front focal plane resides outside the lens casing so fixing an external aperture was not the problem, but it can be for short focal length lenses. For the 50mm lens an external adjustable aperture was fixed to one end of a custom designed screw thread which is mounted on to the lens outer casing. The position of the aperture was adjusted by moving the screw thread manually. The front focal plane was determined based on the conventional procedure described above and was found to be at a distance of 25mm outside the lens surface. In the case of the 35mm lens the front focal plane was found to reside on the lens outer surface, so the external aperture was fixed closely to the lens outer surface. Figure (3.6) shows the 50mm and 35mm lenses which have been converted to telecentric by the introduction of an external multi-leaf adjustable aperture at the front focal plane.

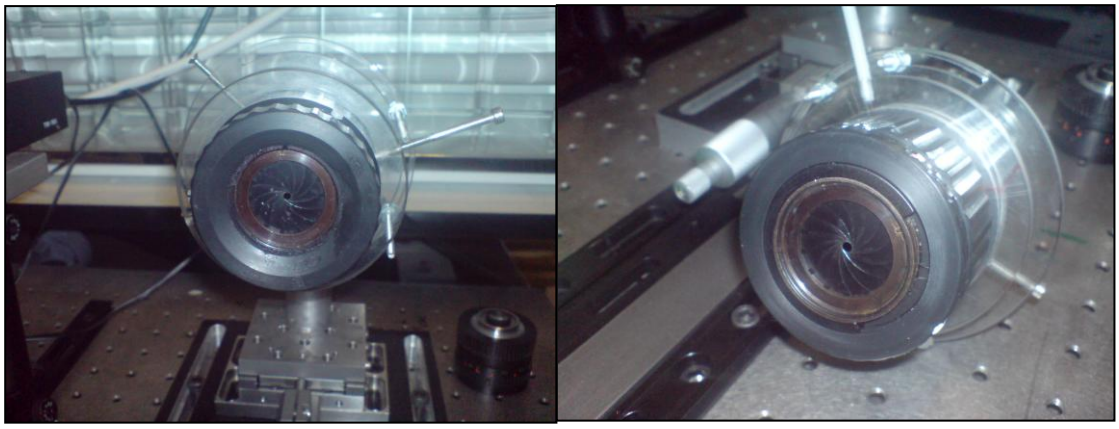


Figure 3.6: 35mm lens converted to telecentric (Left) and 50mm lens converted to telecentric (Right)

3.3. Extension of Phase Correlation Technique to Measure Image Magnification

In this Section the results of Foroosh and Zerubia [29] were extended to estimate the magnification changes between the near and far-focussed images. Here the results of the Phase Correlation method explained in Section 3.1.2 are recalled. If the images f_1 and f_2 differ by a shift x_0, y_0 then the normalised Cross Power Spectrum of the two images f_1 and f_2 with their Fourier Transforms, F_1 and F_2 is defined as

$$\frac{F_2(\varepsilon,\eta) * F_1^*(\varepsilon,\eta)}{\left|F_2(\varepsilon,\eta) * F_1(\varepsilon,\eta)\right|} = e^{-j2\pi(\varepsilon x_0 + \eta y_0)} \quad \text{--- (3.7)}$$

where F^* is the complex conjugate of F.

The above expression can be solved for translation detection either:- (1) By directly working in the Fourier domain where a plane is fitted to the phase difference data and the gradient along the x and y directions provide the required shift; or (2) By inverse transforming the expression which results in an impulse function at some particular spatial coordinates. These coordinates give the shift required to register the two images. It was reported in [14] that Watanabe and Nayar applied the first method to determine the shifts between the defocused images. The defocused images were divided into sub-blocks of 64 x 64 pixels and the spectral ratio was computed for the individual sub-blocks. A plane was fitted to the phase data of the spectral ratio and the gradient of the plane provided the required shift. However, Foroosh and Zerubia in their general paper [29] reported that the above approach would render inaccurate shift results since it requires a plane to be fitted to the noisy phase data. In the proposed approach, the image was divided into sub-blocks and the magnification shift was estimated as a local translation problem within the individual sub-block. Here the radial shifts due to magnification were assumed to be constant within the individual sub-block and the centre pixel of the image coincides with the centre of the lens. The Fast Fourier Transform (FFT) was applied to the individual sub-block and then the normalised Cross Power Spectrum was calculated as per equation (3.7). To estimate the shifts, the second method was adopted where the inverse transform of the normalised Cross Power Spectrum provided the required spatial shift. It was found that this approach was more practical and robust to noise than the method adopted by Watanabe [29]. Based on the assumptions, it can be stated that theoretically when an image is magnified, the radial shift due to magnification at the

centre sub-block of the image is negligible or zero but when an image is translated and magnified the centre sub-block has a shift introduced equivalent to the translation shift induced by the lens system. Therefore the translation shift estimated at the centre sub-block can be used as a correction factor to estimate the radial shifts of the non-central sub-blocks. In practice, as the focus setting of the lens was changed to capture the near and the far-focused images, it was found that the lens introduced both translation and magnification shifts, so the algorithm described in Section 3.4 first estimated the collective shifts due to the magnification and translation for the individual sub-blocks. Later, the shift estimated at the centre sub-block, termed the global translation shift was used to correct the translation shifts from the non-central sub-blocks. Once the translation was corrected, the radial shifts due to magnification are measurable.

Finally to increase the accuracy of the system, the results were extended to sub-pixel estimations. Foroosh and Zerubia [29] have reported that the signal power of the Phase Correlation function is not concentrated at a single discrete sample point, but is distributed over several low resolution samples surrounding the nominal sub-pixel position of the actual peak. In this model, once the position of the maximum value low resolution sample was found, the sub-pixel displacements were computed by considering the signal power of the Phase Correlation function at each of the four samples surrounding the main peak. Suppose the maximum sample value was located at (0, 0) (see Figure (3.7)) then to determine the sub-pixel shifts along the horizontal axis (Δx), the signal power at the locations (1,0) and (-1,0) were compared. If the signal power at (1, 0) was greater than at (-1,0) then the sub-pixel displacement lies between the spatial coordinates (0,0) and (1,0). In this case to determine the sub-pixel shift these two signal powers (C(0,0) and C(1,0)) at these locations are substituted into equation (3.8) to compute the displacement(Δx). This is a linear interpolation (proportionality). Detailed analysis of the equation (3.8) is presented in Appendix 7. Peak finding using higher order polynomial was not employed, since the signal power of the Phase Correlation function is not evenly distributed around the main peak as stated in [29] and also found by experiment as a part of this project. Basically there was not enough data's to enable a good fit. In equation (3.8), *C(-1,0)* is replaced by *C(1,0)* if its power is greater. Similarly, calculations are performed along the vertical axis (Δy) to locate the peak.

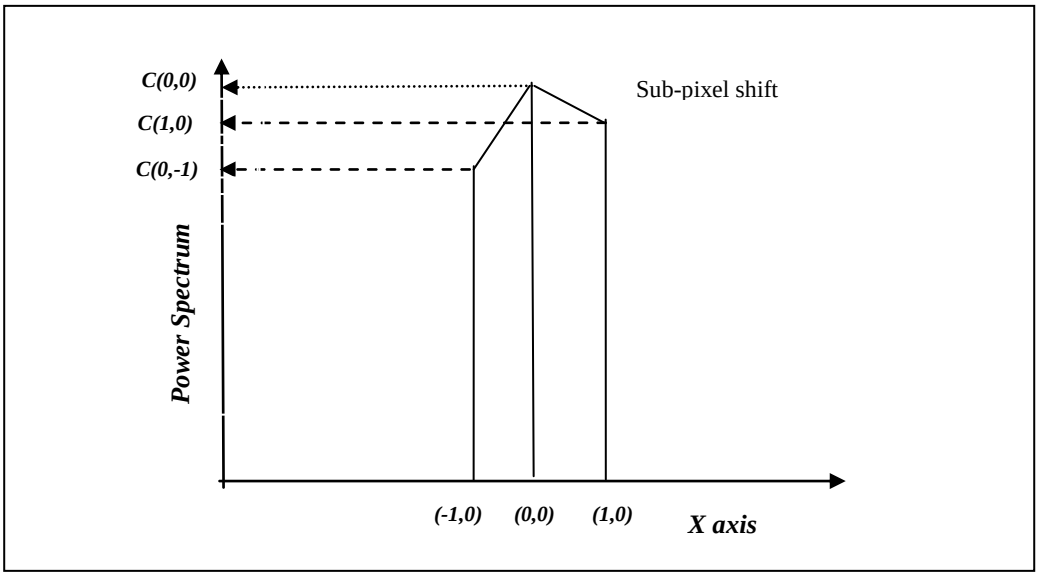


Figure 3.7: Sub-pixel shift measurement – a pictorial explanation

So, the sub-pixel displacements Δx and Δy are given by [29]

$$\Delta x = \frac{C(1,0)}{C(1,0) \pm C(0,0)} \qquad \text{and} \qquad \Delta y = \frac{C(0,1)}{C(0,1) \pm C(0,0)} \text{ --- (3.8)}$$

The magnitude and the orientation of the shifts between the two defocused images are shown as a needle diagram (refer to Section 3.6), where the pixel positions of the individual sub-blocks of the original image are shown as dark dots and those of the magnified image are shown as light dots. For illustration purposes the magnitude of the shifts are multiplied by a factor of 5. The next Section describes the algorithm to estimate the magnification change based on the proposed method.

3.4. Algorithm for Magnification Estimation using the Phase Correlation technique [87]

Step 1: The images (near and far-focussed) are divided into sub-blocks of size n x n. In the experiments the sub-blocks were of 65x65 pixels for simulated images and 150x124 for real images.

Step 2: To avoid the effects of spectral leakage, the sub-blocks were multiplied with a 2D Hanning window prior to applying the FFT.

Step 3: The FFT was applied to the individual sub-blocks and then the normalised Cross Power Spectrum was calculated using equation (3.7).

Step 4: By applying inverse FFT to the normalised Cross Power Spectrum, the integer shifts between two sub-blocks are determined. Later, the sub-pixel shifts ($\Delta x, \Delta y$) are determined by considering the peaks adjacent to the main peak as described in Section 3.3.

Step 5: Steps 2 to 4 are repeated for all individual sub-blocks in the image.

Step 6: The shift at the centre sub-block termed as the global translation between the images is used as a correction factor to determine the shifts due to magnification in the non-central sub-blocks.

Step 7: Once the translations have been corrected from the non-central sub-blocks the radial shifts due to magnification are easily visible and are measured in isolation.

3.5. Design of Experiment

In the experiment the pattern described in [41] and shown in Figure (3.8) was used. In general the texture pattern should have high energy at high spatial frequencies, and the image sub-block area should be larger than the periodicity of the pattern to avoid phase ambiguities [41]. The effect of the discontinuity at the sub-block border was reduced by applying a 2D Hanning window defined by $w(n_1, n_2) = \frac{1+\cos(\frac{\pi n_1}{M_1})}{2} \frac{1+\cos(\frac{\pi n_2}{M_2})}{2}$.

Experiments were performed on simulated images with sub-pixel translation and radial integer shifts. Sub-pixel images were obtained for simulation by shifting and down-sampling a high resolution image. Radial integer shifts were obtained by indexing so that the shift at the centre sub-block was zero and the sub-blocks on either side of the centre block are progressively shifted by indexing so as to simulate the zoom effect. The shifts within a sub-block were assumed to be constant.

By experiment it was found that using interpolation methods such as nearest neighbour, bilinear and cubic to generate scaled images resulted in an output image that was non-symmetric about the centre or smoothed by the transformation. These effects lead to errors while finding the peak in the Cross Power Spectrum and hence radial shifts were limited to integer pixel shifts.

3.6. Experiments with Real and Simulated Images

In this Section the shift measurement results are presented for simulated and real images. The images considered were translated as well as magnified. Sections 3.6.1, 3.6.2 and 3.6.3 discuss the results based on simulated images, and Section 3.6.4 those from a real image, both with and without the telecentric aperture. The shifts measured at each individual sub-block are shown in the form of a needle diagram. For each experiment the maximum error recorded along the row and column has been presented.

3.6.1. Experiment on a Simulated Image with sub-pixel Translation

The proposed algorithm was tested on images with sub-pixel shifts. The sub-pixel images were obtained from a high resolution image that was down sampled by factor *ds*. Then by shifting one low resolution image by *s* pixels with respect to the other image, shifts of the ratio *s/ds* pixels were introduced. By choosing appropriate values for *s* and *ds*, fractional shift can be introduced into the images. The two test patterns used in the experiment are shown in Figure (3.8), left shows the original image and the pattern shown in the right was shifted by -4 pixels along the row and -1.5 pixels along the column.

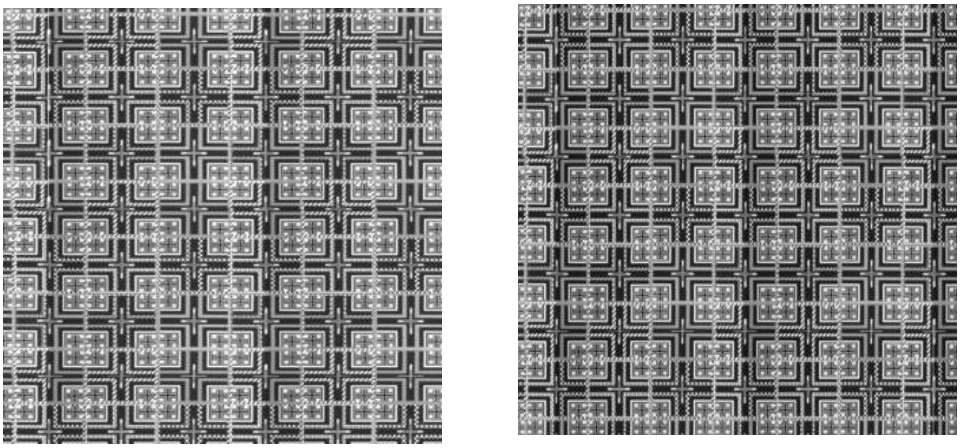


Figure 3.8: Original image (Left) and the Shifted image (Right)

Figure (3.9) shows the resultant shifts determined from the patterns. The dark dot represents the original image and the light square, the shifted image. The maximum

error recorded using the proposed algorithm was 0.1167 pixels along the row and 0.1395 pixels along the column. The needle diagram illustrates the shifts after scaling by a factor of times 5 so that the direction can be more easily seen.

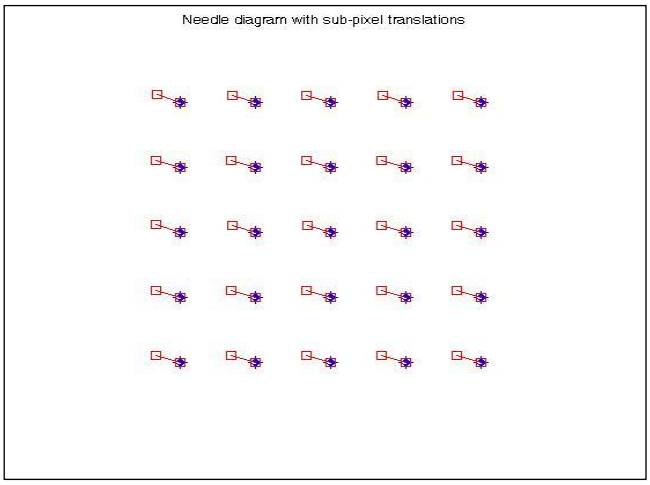


Figure 3.9: Shift Detected between the Patterns

3.6.2. Experiment on Simulated Image with Radial Shift

Image magnification causes the pixels to move radially outwards, so to determine the accuracy of the algorithm for magnified images simulations were carried on images that were radially shifted by indexing.

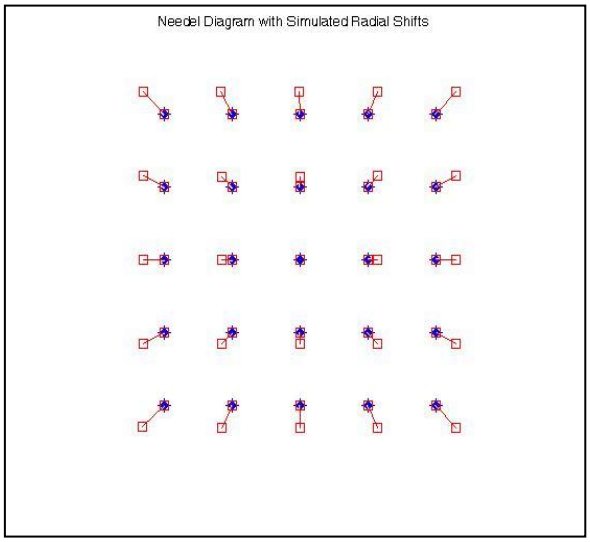


Figure 3.10: Estimated Radial shift

Figure (3.10) shows the measured shift estimated from the images shifted radially by indexing. The images were shifted by ± 4 pixels (row and column) in the sub-blocks near the border and ± 2 pixels (row and column) in the sub-blocks adjacent to the centre sub-block. The shift at the centre block of the image was zero. The maximum error recorded was 0.1815 pixels and 0.0787 pixels along the row and column respectively. Again the needle diagram illustrates the shifts after scaling by a factor of times 5.

3.6.3. Experiment with sub-pixel Translation together with Integer Radial Shift

Simulations were carried out on images with both sub-pixel translation and integer radial shift. Figure (3.11) illustrates the shift between the images that had a translation of -4,-1.5 pixels along the row and column respectively, and radial shifts of ± 4 pixels (row and column) in the sub-blocks near the border of the image and ± 2 pixels (row and column) in the sub-blocks adjacent to the centre sub-block. As discussed earlier, when an image is scaled and translated, the shift estimated at the centre block would measure the translation. In our experiment the shift estimated at the centre sub-block was -3.9475 and -1.4823 pixels indicating that the shift did indeed include translation.

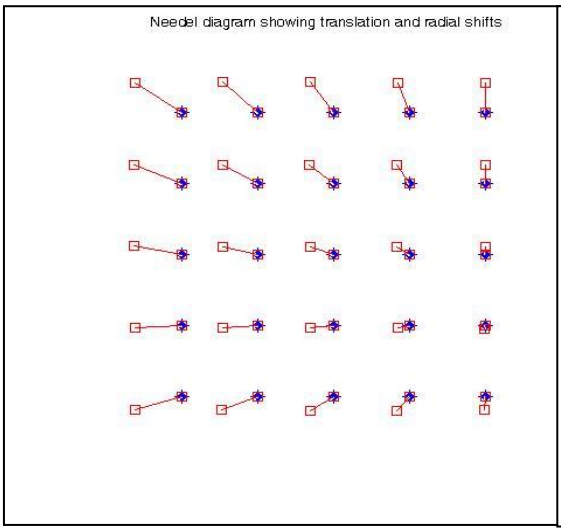


Figure 3.11: Translation and Radial Shifts

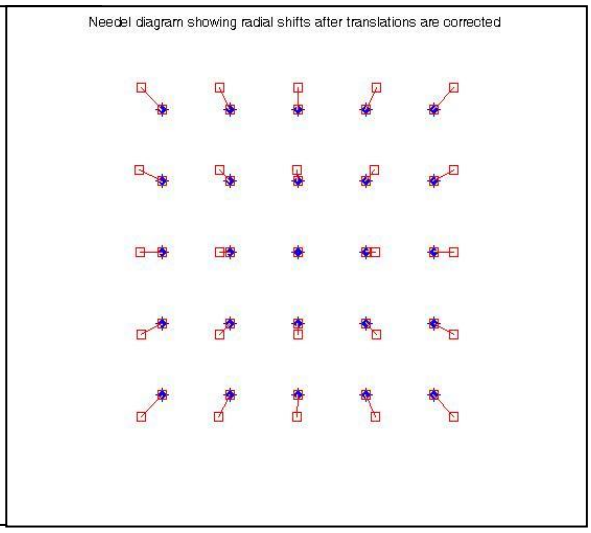


Figure 3.12: Estimated Radial Shifts after Translation correction

The shifts due to magnification were determined after translations were removed from the individual sub-blocks and Figure (3.12) shows the radial shifts obtained after the translations had been removed. The maximum error recorded was 0.1815 pixels along the row and 0.0318 pixels along the column. The shifts illustrated are multiplied by a factor of 5.

3.6.4. Experiments with Real Images

Images were captured using a PULNIX TM-765 monochrome camera and 50mm manual lens. The pattern was placed at a distance of 824mm from the lens and two defocused images, near-focussed at 767mm and far-focussed at 874mm were captured. Two sets of experiments were performed. In the first experiment the telecentric aperture was removed and the magnification changes between the near and the far-focused images were determined. Figure (3.13) shows the near and far-focused images and Figure (3.14) illustrates the resultant magnification before and after translation correction.

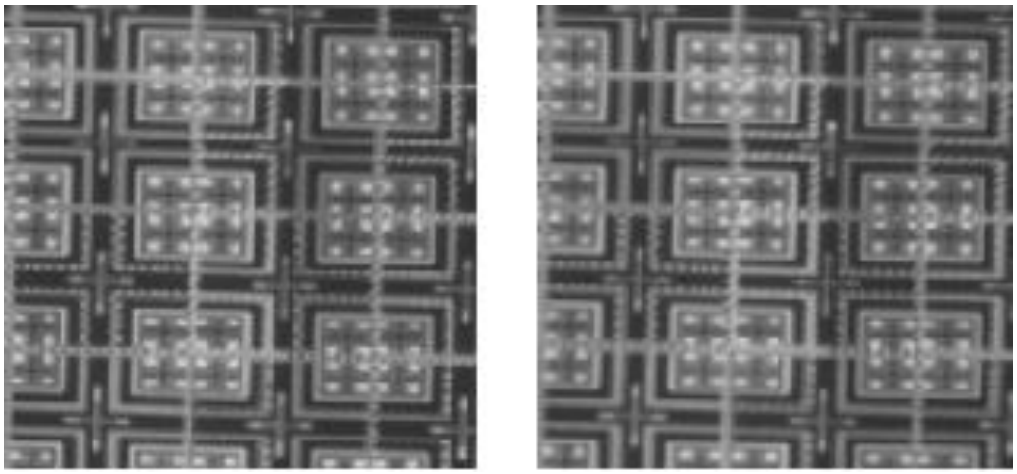


Figure 3.13: Near and far-focussed Images

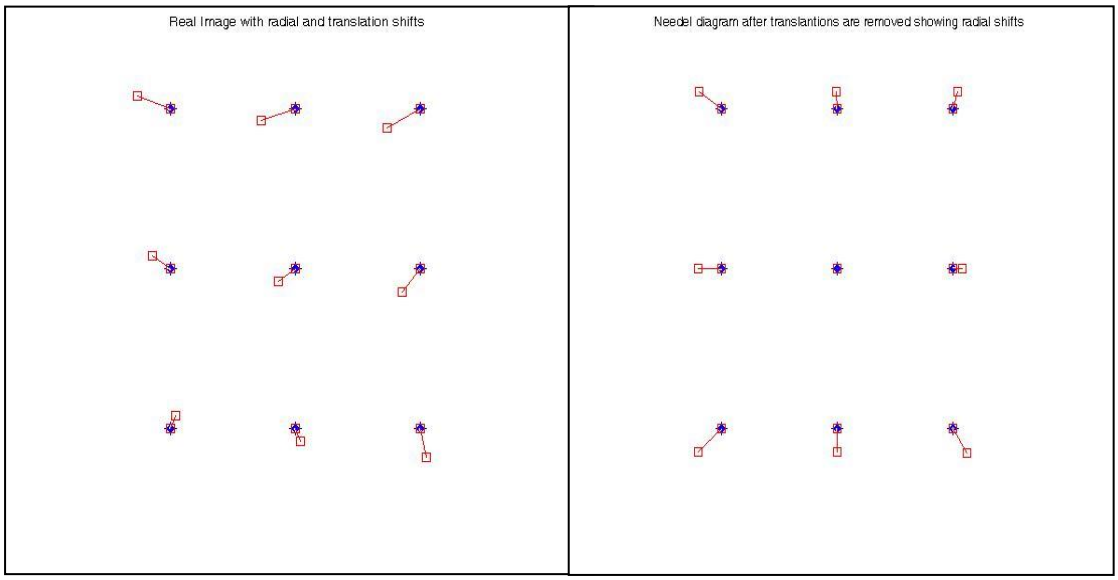


Figure 3.14: Resultant shifts before translation correction (Left) and the Resultant shift after translation correction (Right)

The needle diagram on the left of Figure (3.14) indicates the collective shifts due to magnification and translation present in the optical setup as the focus setting is changed from the near-focused to the far-focused position. It should be noted that when the global translation is removed from non-central sub-block the radial shifts due to magnification are visible (Figure (3.14 right)) and can be measured in isolation. The maximum absolute shift recorded was 4.4865 pixels along the column and 4.8387 pixels along the row. The shifts recorded are shown in Table 3.1. In the needle diagram (Figure (3.14)) the pixel positions in the original image are shown as dark dots and those in a magnified image are shown as light squares. The shifts shown are multiplied by a factor of 5.

Shift measured along	Global Translation Recorded	Min shift recorded	Max shift Recorded	Mean shift Recorded
Row in pixels	2.3881	0.0458	4.8387	2.5991
Col in pixels	-3.3921	0.0917	4.4865	2.9027

Table 3.1: Shifts recorded on a conventional DFD lens systems

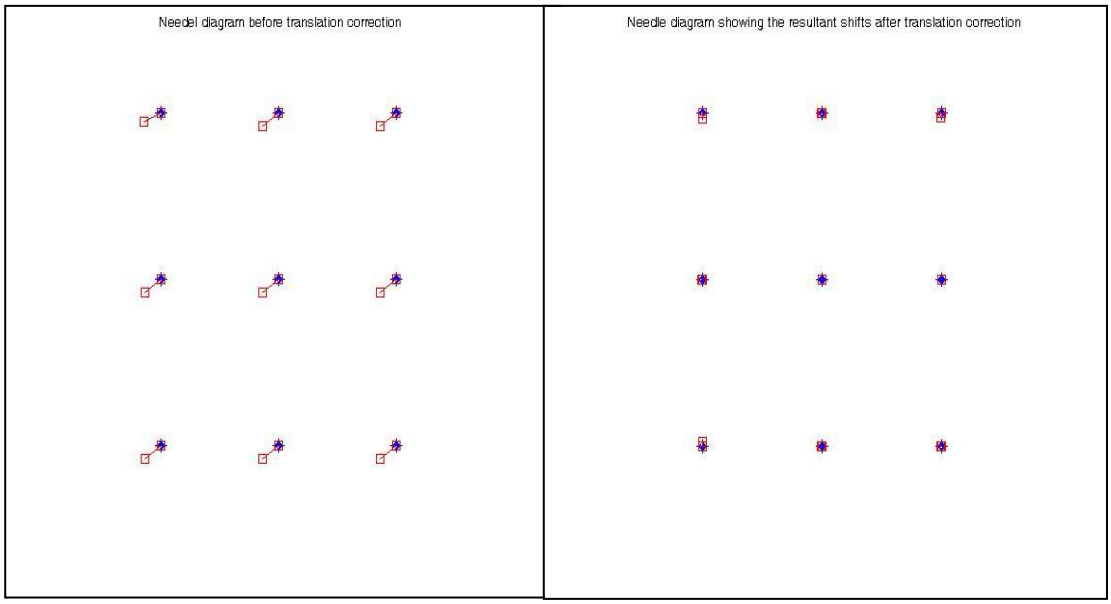


Figure 3.15: Resultant shift before translation correction (Left) and Resultant shift after Translation correction (Right)

In the second experiment the conventional lens was converted into a telecentric by placing an external aperture of diameter 6.5mm at the front focal plane of the lens. The positions of the near and far-focused images remain the same. Again the needle diagram on the left of Figure (3.15) represents the collective shift due to magnification and translation. Here since the lens system was converted into telecentric, the shifts present are mainly due to translation and when the global translation has been corrected the radial shifts due to magnification are easily noticeable (Figure (3.15 right)). The shifts recorded after the inclusion of the aperture stop are recorded in Table 3.2. It can be seen that the shift estimated at the centre sub-block in the both the experiments are almost equivalent proving that the centre sub-block estimates the global translation. However the shifts due to magnification have been reduced from pixel to sub-pixel levels by introduction of the telecentric aperture.

Shift measured along	Global Translation recorded	Min shift recorded	Max shift Recorded	Mean shift Recorded
Row in pixels	2.4143	0.0077	0.8194	0.1380
Col in pixels	-3.3645	0.0156	0.1524	0.0571

Table 3.2: Shifts recorded on a Telecentric DFD lens system

Conclusion

In this chapter an optical method using telecentric optics has been proposed to correct the magnification changes between the near and far-focused images (Section 3.4). A simple and robust technique using Phase Correlation was reported to estimate the radial shifts due to magnification between the two defocused images. From the experiments, for a conventional defocus model, the measured maximum absolute radial shift was 4.48 pixels along the column and 4.83 pixels along the row, but on inclusion of telecentric optics the maximum shift was reduced to 0.1524 pixels and 0.8194 pixels respectively. It can be clearly seen that the inclusion of an aperture stop at the front focal plane ensures that the Principal Rays on the image side are parallel to the optical axis, and the magnification change between the near and the far-focused images have been considerably reduced. Reducing the shifts due to magnification to less than a pixel ensured pixel to pixel registration between the near and far-focused images was correct, and increased the accuracy of the recovered depth. In practice the registration algorithm was used to position the aperture correctly at the front focal plane, and to provide a translation correction factor to the depth recovery program. In the next chapter a novel procedure to determine the coefficients for the Rational filters has been explained and then the implementation of the DFD algorithm.

CHAPTER 4

Design of Rational Filters by Two Step Polynomial Approach

Introduction

The chapter describes a simple and efficient procedure to determine the filter coefficients to accurately model the $\frac{M}{P}$ curves described in Section 4.2. The filters provide a fast practical processing of the depth information. The method referred to as the Two Step Polynomial Approach determines the filter coefficients by simplifying the $\frac{M}{P}$ ratio into a Linear Model and Cubic Error Correction Model. Since the method avoids the use of iterative minimisation techniques, it requires minimal computations when compared to Watanabe's [14] method. Furthermore, for a given defocus condition, the validity of the model has been verified by comparing the modelled $\frac{M}{P}$ ratio (obtained from the designed filters) with the theoretical $\frac{M}{P}$ ratio obtained from the 2D discrete $\frac{M}{P}$ ratio space. It must be noted that Watanabe and Nayar [14] have not verified their model which they have designed by another method. Experimental results with simulated and real images have been used to illustrate that the filter coefficients determined by the new method estimated the depth to a higher accuracy than Watanabe filters.

The chapter first describes the principle of DFD based on the optical setup described in [14] and then proceeds with an explanation of the $\frac{M}{P}$ curves. Here the $\frac{M}{P}$ curves described by Watanabe and Nayar have been used, but the filter coefficients have been determined using the new Two Step Polynomial Approach described in Section 4.3. A detailed comparison of filters designed by the Two Step Polynomial Approach and Watanabe and Nayar's approach is provided in Section 4.5. A new single frequency test image method has been used. Section 4.7 and 4.8 provide the experimental results for both real and simulated images and Section 4.9 reports the options available to increase the working distance for a given experimental setup.

4.1. Principle of Depth from Defocus

Depth from Defocus (DFD) is a technique where the defocus parameter is used as a clue to estimate the distance of an object. In the generalised model of the DFD [2] [14], where the image structure in unknown, depth is calculated from the amplitude ratio of the two defocused images i_1 and i_2 on either side of the focused image i_f. Here i_1 refers to the far-focused image and i_2 to the near-focused. In a conventional lens system the energy flux (radiance) emitted from the point P at a distance d in the object side is mapped on to a point Q in the focussed plane i_f at a distance d_i. The relation between object distance d, image distance d_i and the focal length f is given by the lens law as

$$\frac{1}{f} = \frac{1}{d_i} + \frac{1}{d} \quad \text{--- (4.1)}$$

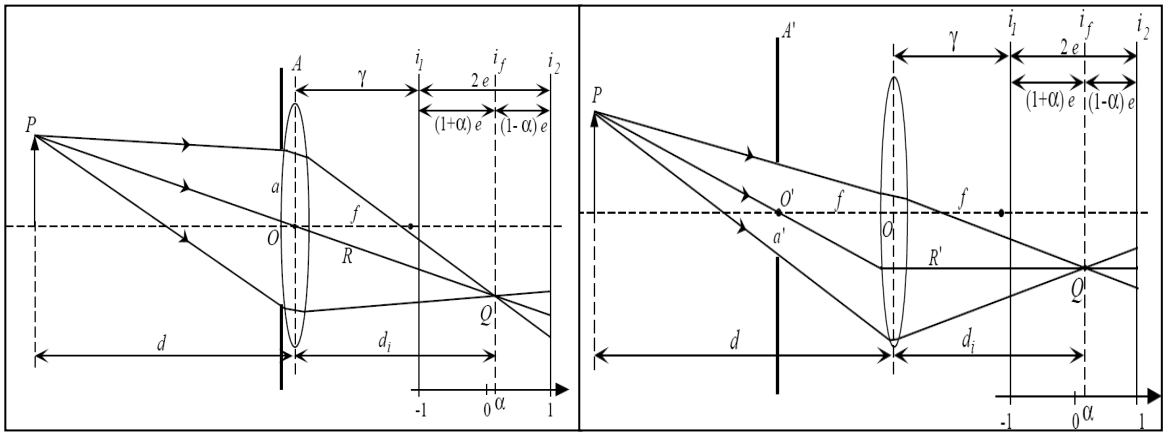


Figure 4.1a: Conventional DFD Optical Setup

Figure 4.1b: DFD system based on Telecentric optics [14]

To recover depth from an object as described in [14], the algorithm requires the defocused images i_1 and i_2 to be separated by a known physical distance of $2e$. In Figure (4.1a) the sensor planes on which the two defocused images are formed are separated from the focused image by a distance $(1 \pm \alpha)e$, where α, the normalised depth ranges between -1 for the far-focused image and +1 for the near-focused image. For accurate depth estimation the defocused images i_1 and i_2 need to be registered in terms of magnification (or zoom), but in a conventional lens model shown in Figure (4.1a) as the image sensor is moved from defocused image i_1 to the defocused image i_2, the location of the point P is displaced along the Principal Ray R,

which induces image magnification. To obtain constant magnification between the defocused images, a DFD system with a telecentric optics shown in Figure (1.4b) was used. The principle of telecentric optics has been presented in chapter 3.

The light intensity at any point in the images i_1 and i_2 can be modelled as the convolution of the focussed image i_f with the corresponding point spread function (psf) *h(x,y).* Theoretically for a defocused lens under geometric optics the psf can be modelled either as a Gaussian, Pillbox or Generalised Gaussian. For notation simplicity the mathematical equations of the psf are illustrated in 1D.

(1) Gaussian [49]

$$h_g(x) = \frac{1}{\sqrt{2\pi\sigma}} \exp\left\{ -\frac{1}{2} \frac{(x-\bar{x})^2}{\sigma^2} \right\} \text{---- (4.2)}$$

where σ the standard deviation and the psf is centred at $x - \bar{x}$

(2) Pillbox [49]

$$h_p(x) = \frac{1}{2\sigma}\left[u(x+\sigma) - u(x-\sigma)\right] \text{---- (4.3)}$$

where σ is the radius of the Pillbox

(2) Generalised Gaussian [49]

$$h_G(x) = \frac{p^{1-\frac{1}{p}}}{2\sigma\Gamma(\frac{1}{p})} \exp\left\{ -\frac{1}{p} \frac{\left|x-\bar{x}\right|^p}{\sigma^p} \right\} \text{----- (4.4)}$$

where $\bar{x}$ refers to the mean, σ to the standard deviation, $\Gamma()$ to the gamma function and p to the power of the function required. The function takes the Gaussian psf when $p = 2$ and a Pillbox when $p = \infty$.

Here since the depth estimation algorithm was based on Watanabe's method [14], it was assumed that the images were defocused by the Pillbox psf. In the Spatial domain the 2D Pillbox psf can be modelled as

$$h(x,y) = h(x,y;z,F_e) = \frac{4F_e^{\,2}}{\pi z^2 e} \Pi\left(\frac{F_e}{ze}\sqrt{x^2+y^2}\right) \text{----- (4.5)}$$

where z ($z = (1 \pm \alpha)e$) denotes the distance between the sensor plane and the focussed image i_f, Fe ($F_e = \frac{f}{a'}$ for a telecentric lens) represents the *f* number of the

lens, and $\Pi(r)$ is the rectangular function that takes the value 1 if $|r| < \frac{1}{2}$ and 0 otherwise. Here *x, y* are the coordinates in the horizontal and vertical directions.

The frequency domain equivalent of equation (4.5) can be expressed using the Bessel Function of order one [14] given by

$$H(u,v) = H(u,v;z,F_e) = \frac{2F_e}{\pi z\sqrt{u^2+v^2}} J_1\left(\frac{\pi z}{F_e}\sqrt{u^2+v^2}\right) \text{----- (4.6)}$$

where $J_1(r)$ denotes the first order Bessel Function and *u, v* are the frequencies in the horizontal and vertical directions respectively. The spatial and frequency plots of the defocus function are shown in Figures (4.2) and (4.3) for in-focus and out-of-focus cases respectively.

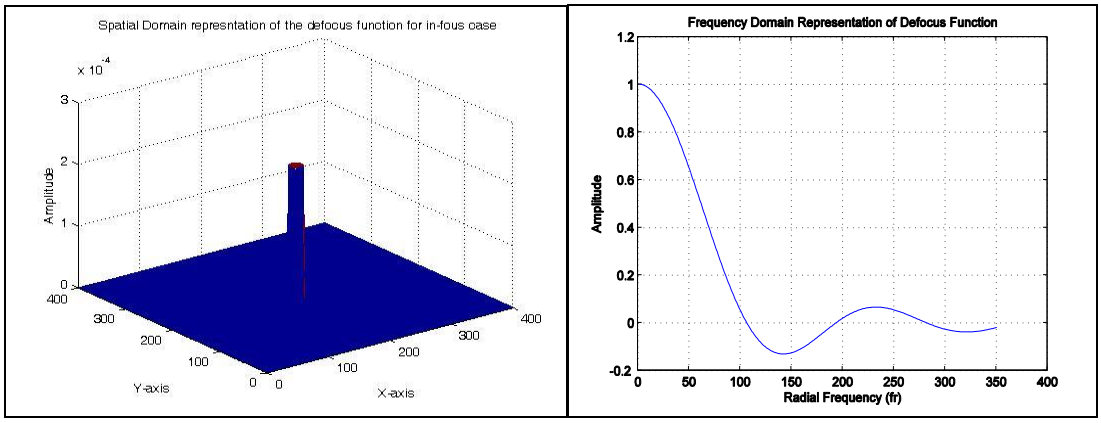


Figure 4.2: Defocus function (in-focus) - Spatial (Left) and 1D frequency domain model (Right)

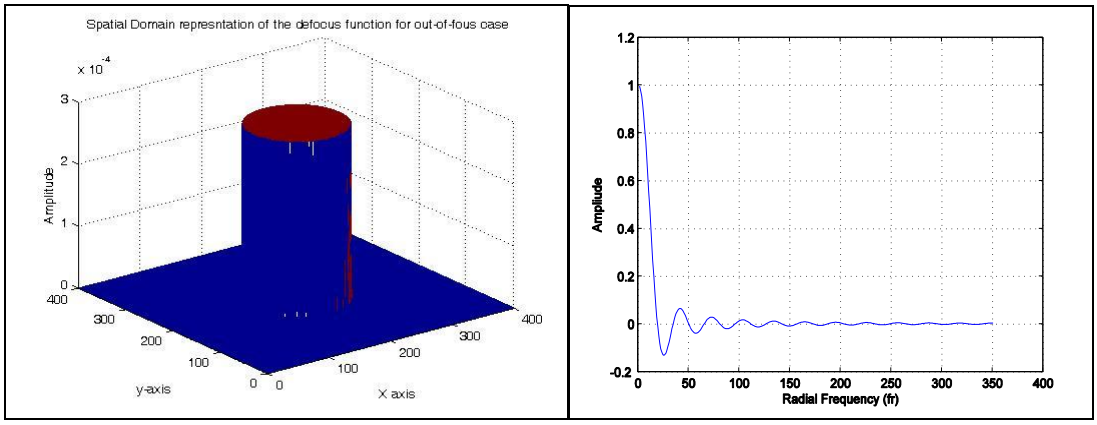


Figure 4.3: Defocus function (out-of-focus) - Spatial (Left) and 1D frequency domain model (Right)

So, the defocused image i_1 can be modelled as the convolution of the focused image $i_f(x,y)$ with the 2D Pillbox function $h_1(x,y)$ and equations (4.7) and (4.8) represent the spatial and frequency domain equivalents of the defocused image i_1.

$$i_1(x,y) = i_f(x,y) * h_1(x,y) \text{ ---------- } (4.7)$$

$$I_1(u,v) = I_f(u,v) H_1(u,v) \text{ ---------- } (4.8)$$

Similarly the defocused image i_2 can be modelled as

$$i_2(x,y) = i_f(x,y) * h_2(x,y) \text{ ---------- } (4.9)$$

$$I_2(u,v) = I_f(u,v) H_2(u,v) \text{ ---------- } (4.10)$$

From the above equations the spectra of the images are different. These are analysed to produce the $\frac{M}{P}$ ratio that can be used to determine the in-focus axial position of each pixel and hence the depth.

4.2. Normalised $\frac{M}{P}$ Ratio

This Section describes the importance of the $\frac{M}{P}$ ratio and based on the results of Watanabe and Nayar [14], the $\frac{M}{P}$ curves were reproduced for a wide range of radial frequencies over the normalised depth range of -1 to +1. Earlier methods based on a frequency domain approach [1] [6] [50], estimated the depth by considering the amplitude ratio of the near and far-focussed images I_1 and I_2 at the particular radial frequency $f_r = \sqrt{u^2 + v^2}$. Watanabe and Nayar [14] provided an improvement to the existing ratio by considering the ratio of the difference in amplitude of the defocused images to the sum of the amplitudes of the defocused images. The ratio is termed the normalised $\frac{M}{P}$ ratio, where M represents the difference in amplitudes of the defocused images and P represents the sum. This estimated the depth with a higher accuracy. Furthermore, since the ratio is independent of image intensity, the estimated depth is more stable than with the earlier techniques [1] [50].

So, spatial domain: $\frac{m}{p}(x,y) = \frac{i_2(x,y) - i_1(x,y)}{i_2(x,y) + i_1(x,y)}$ and in frequency domain: $\frac{M}{P}(u,v;\alpha) = \frac{I_2(u,v) - I_1(u,v)}{I_2(u,v) + I_1(u,v)}$.

Now substituting the value of I_1 and I_2 from equation (4.8) and (4.10) in the frequency domain equation, and dividing through to remove I_f we have

$$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{H(u,v;(1-\alpha)e,F_e) - H(u,v;(1+\alpha)e,F_e)}{H(u,v;(1-\alpha)e,F_e) + H(u,v;(1+\alpha)e,F_e)} \text{ ------ (4.11)}$$

This $\frac{M}{P}$ ratio is independent of image intensity.

From equation (4.11) it can be inferred that the $\frac{M}{P}$ ratio is directly related to the normalised depth α, and the depth at a particular radial frequency can be determined by considering the $\frac{M}{P}$ ratio of the two defocused images.

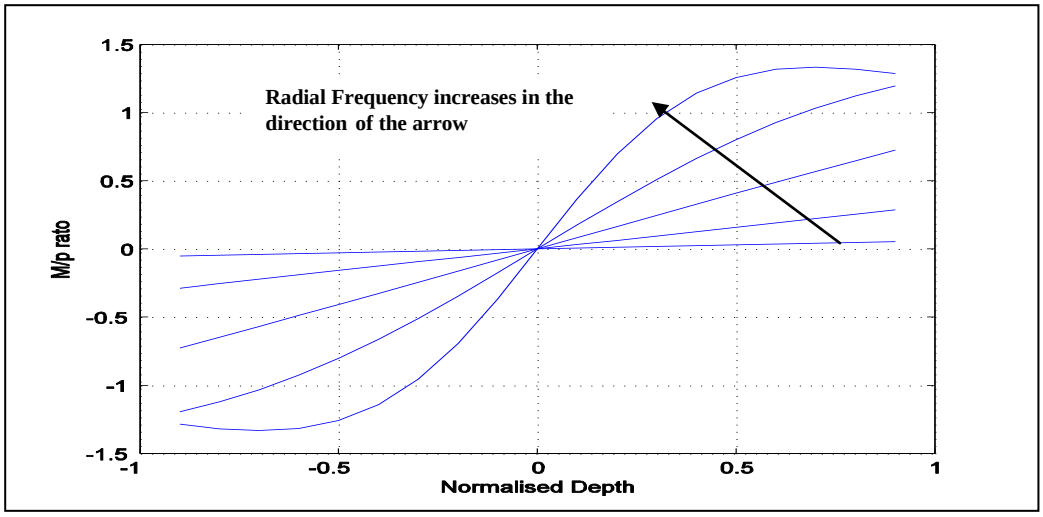


Figure 4.4: $\frac{M}{P}$ ratio vs. Normalised Depth α

Figure (4.4) shows the theoretical relationship between the $\frac{M}{P}$ ratio and the normalised depth α at different radial frequencies. Here the Pillbox psf function was modelled as the first order Bessel function, $J_1(z)$, where $z = (1 \pm \alpha)e$. From the plot it can be inferred that as the radial frequency increases, the $\frac{M}{P}$ ratio loses its

monotonic property. The maximum frequency below which the $\frac{M}{P}$ ratio was found to be monotonic was $f_r \leq 0.61\frac{F_e}{e}$. Monotonicity ensures that each $\frac{M}{P}$ ratio maps onto a unique depth. From the numerical analysis of the first order Bessel function, it was noted that the first zero crossing occurs on the horizontal axis at 3.83171 as found in [53], which is equivalent to $1.22\frac{F_e}{z}$, where $z = (1 \pm \alpha)e$, but in practice, by using the rotationally symmetric Pillbox psf model, the maximum radial frequency below which the $\frac{M}{P}$ ratio is monotonic has increased by a factor of 1.2 of the maximum frequency i.e. $f_r = 0.73\frac{F_e}{e}$. The simulated results demonstrate the relationship between the $\frac{M}{P}$ ratio and the normalised depth α at different radial frequencies. It is evident that a unique depth estimate is available for each individual frequency and the depth information can be recovered from the $\frac{M}{P}$ ratio, provided the response of the designed filters accurately models the $\frac{M}{P}$ curves. Hence in the next Section a procedure based on a Two Step Polynomial Approach is described to effectively determine the filter coefficients capable of accurately modelling the $\frac{M}{P}$ curves.

4.3. Design of Rational Filters by a Two Step Polynomial Approach

From the $\frac{M}{P}$ ratio plot (Figure (4.4)), it can be inferred that for every radial frequency there is a unique $\frac{M}{P}$ curve which was found to be monotonic below the radial frequency $f_r = 0.73\frac{F_e}{e}$ and the normalised depth can be determined directly from the plot provided the amplitudes of the defocused images I_1 and I_2 are known. The objective was to design a 7x7 spatial filter kernel capable of accurately

modelling the $\frac{M}{P}$ ratio for all possible radial frequencies, i.e. determining a kernel which was insensitive to object texture frequency. In practice the designed filter would accurately model the $\frac{M}{P}$ ratio for each frequency thereby providing a depth estimate for every individual pixel. Earlier methods suggested the use of narrow band filters [50] to estimate the power at a large number of individual frequencies, but since the interest lies in real-time depth estimation, this approach seems to be impractical as these filters require more logic support. The second approach is to design broadband filters [14] which are insensitive to the image texture frequency. Since these filters are broadband only a few coefficients are required, so extensive computation can be avoided, and this method can be used for real-time depth computation. To clarify the contribution provided by this work, the $\frac{M}{P}$ curves described by Watanabe and Nayar [14] were implemented directly, but the filter coefficients were determined using the novel Two Step Polynomial Approach. Step 1 involves modelling the linear filters by fitting a linear model to the theoretical $\frac{M}{P}$ ratio and Step 2 determines the correction filter by computing the error between the theoretical and the linear model, and fitting a cubic function to it. This results shown in the later Sections prove that the above model estimates the depth map with a higher accuracy than Watanabe's filters.

4.3.1. Design procedure using the Two Step Polynomial Approach

This Section describes the procedure based on polynomials to determine the filter coefficients. Since the model was based on the $\frac{M}{P}$ ratio, it required the knowledge of the psf of the defocused lens, so for a range of α (normalized depth) values, the psf was pre-computed using the Pillbox psf model. In the design model the range of the normalized depth α was from 0 to 0.99. So based on the earlier assumption, the Pillbox psf was modelled in the frequency domain using the equation

$$H(u,v) = H(u,v;z,F_e) = \frac{2F_e}{\pi z\sqrt{u^2+v^2}} J_1\left(\frac{\pi z}{F_e}\sqrt{u^2+v^2}\right) \text{ --- (4.12).}$$

Here z ($z = (1 \pm \alpha)e$) denotes the distance between the sensor plane and the focused image i_f, Fe ($F_e = \frac{f}{a'}$ for a telecentric setup) represents the f number of the lens, $J_1(r)$ denotes the first order Bessel Function, and u, v are the frequencies in the horizontal and vertical directions respectively.

Once the psf's had been computed, the two dimensional $\frac{M}{P}$ space was discretized into 2^n equally spaced frequency samples below the folding frequency of 0.5 $pixel^{-1}$, and the $\frac{M}{P}$ ratio was computed for all possible radial frequencies f_r over the normalised depth range α using the equation

$$\frac{M(f_r;\alpha)}{P(f_r;\alpha)} = \frac{H(f_r;(1-\alpha)e,F_e) - H(f_r;(1+\alpha)e,F_e)}{H(f_r;(1-\alpha)e,F_e) + H(f_r;(1+\alpha)e,F_e)} \text{ --- (4.13).}$$

Hence the two dimensional $\frac{M}{P}$ space has been transformed into a two dimensional $\frac{M}{P}$ ratio space, where a unique $\frac{M}{P}$ ratio exits for each radial frequency f_r and the normalized depth α (see Figure (4.5)). Hitherto the two dimensional $\frac{M}{P}$ ratio space will be referred as the discrete $\frac{M}{P}$ ratio space.

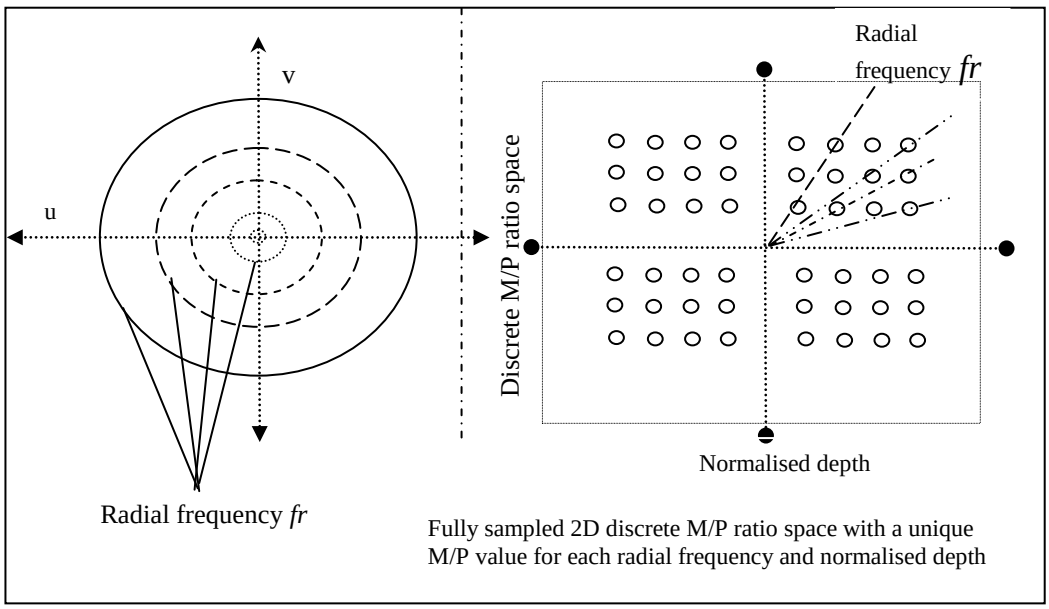


Figure 4.5: 2D discrete $\frac{M}{P}$ ratio space.

To proceed with the filter design, the $\frac{M}{P}$ ratio was modelled as a linear combination of the three filters, Gm_1. Gp_1 and Gp_2 (equation (4.14)) as described in [14].

$$\text{So, } \frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{Gp_1(u,v)}{Gm_1(u,v)}\beta + \frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3 \text{ ---- (4.14)}$$

The above model can be simplified and rewritten as a Linear Model and an Error Correction model as

$$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{M^{'}(u,v;\beta)}{P^{'}(u,v;\beta)} + \frac{M^{'''}(u,v;\beta)}{P^{'''}(u,v;\beta)} \text{ ----- (4.15)}$$

where $\frac{M(u,v;\alpha)}{P(u,v;\alpha)}$ represents the theoretical $\frac{M}{P}$ ratio, $\frac{M^{'}(u,v;\alpha)}{P^{'}(u,v;\alpha)}$ represents the linear model $\frac{Gp_1(u,v)}{Gm_1(u,v)}\beta$ and $\frac{M^{'''}(u,v;\alpha)}{P^{'''}(u,v;\alpha)}$ represents the Error correction model $\frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3$. Here α , β denote the actual and the estimated depth, and *u, v* are the frequencies in the horizontal and vertical directions respectively.

Now, consider the Linear Model $= \frac{Gp_1(u,v)}{Gm_1(u,v)}\beta$. The model can be considered as a straight line passing through the origin and can be represented using the linear function $y = Ax$ where the gradient at a radial frequency $A(f_r)$ is equal to ratio $\frac{Gp_1(f_r)}{Gm_1(f_r)}$ and $f_r = \sqrt{u^2 + v^2}$. To proceed, the Linear Model requires knowledge of:- (1) The gradient functions, $A(f_r)$ at each radial frequency; and (2) The response of either Gm_1 or Gp_1. To compute the gradient, $A(f_r)$ the discrete $\frac{M}{P}$ ratio space can be utilised since it provides a unique $\frac{M}{P}$ ratio for each normalized depth corresponding to a particular radial frequency. Hence to determine $A(f_r)$ at a radial frequency, f_{r1}, where $f_{r1} = \sqrt{u^2 + v^2}$, a linear function, $y = Ax$ was fitted to the $\frac{M}{P}$ ratios over the normalised depth range, α = 0 to 0.99 for the corresponding radial frequency f_{r1}. Thus by considering all possible radial frequencies, $A(f_r)$ was computed for each discrete radial frequency in the $\frac{M}{P}$ space. Now, the frequency

response of either Gp_1 or Gm_1 must be predefined. Since the required filter needs to possess a band-pass filter characteristic together with rotational symmetry [14], the response of Gp_1 was modelled as a Laplacian of Gaussian (LOG) based on the equation

$$Gp_1(f_r) = (\frac{f_r}{f_{spread}})^2 \exp(1-(\frac{f_r}{f_{spread}})^2), \text{ --- (4.16)}$$

where f_{spread} = 0.4*fnyquist*. Here the constant 0.4 was used as it ensures an acceptable width of the LOG filter. Note: to avoid the divide by zero problem, the second filter, Gm_1 was not modelled as band-pass. Once the frequency response of Gp_1 and $A(f_r)$ are determined, the response of Gm_1 can be determined with ease as $A(f_r) = \frac{Gp_1(f_r)}{Gm_1(f_r)}$. Here the only unknown $Gm_1(f_r)$ can be calculated from the gradient as $Gm_1(f_r) = \frac{Gp_1(f_r)}{A(f_r)}$. The filter Gm_1 designed based on the Linear Model has the characteristics of a low pass filter together with rotational symmetry. Thus by employing the Linear Model, the frequency response of the filters Gp_1 and Gm_1 have been modelled. The next Section discuses the Error Correction model where the frequency response of the filter Gp_2 has been modelled.

4.3.2. Error Correction Model

In this Section the response of the filter Gp_2 was modelled by considering the error between the theoretical $\frac{M}{P}$ ratio, $\frac{M(u,v;\alpha)}{P(u,v;\alpha)}$ and the Linear Model, $\frac{M'(u,v;\alpha)}{P'(u,v;\alpha)}$.

So, $Error(u,v;\alpha) = \frac{M'''(u,v;\alpha)}{P'''(u,v;\alpha)} = abs(\frac{M(u,v;\alpha)}{P(u,v;\alpha)} - \frac{M'(u,v;\alpha)}{P'(u,v;\alpha)}) = \frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3$ --(4.17)

It can be inferred that the Error Correction Model $= \frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3$ can be modelled as a cubic function, $y = Cx^3$, where the gradient C at a particular radial frequency f_r corresponds to the ratio $\frac{Gp_2(f_r)}{Gm_1(f_r)}$, hence by computing the gradient, $C(f_r) = \frac{Gp_2(f_r)}{Gm_1(f_r)}$, the frequency response of the filter Gp_2 can be

determined. To compute the gradient a cubic function, $y = Cx^3$ was fitted to the absolute error between the theoretical $\frac{M}{P}$ ratio and the Linear Model. Once the gradient has been computed for all possible radial frequencies in the $\frac{M}{P}$ space, the response of filter Gp_2 can be determined directly from the gradient itself since the response of Gm_1 has been determined earlier. Thus Gp_2 can be modelled as $Gp_2(f_r) = C(f_r)Gm_1(f_r)$. Once the filters had been modelled, a higher order 2D polynomial was fitted to the respective models to smooth their frequency response. After experimentation the optimum polynomial order used in the design was found to be 12. Figure (4.6) shows the frequency responses of 1D version of the designed filters.

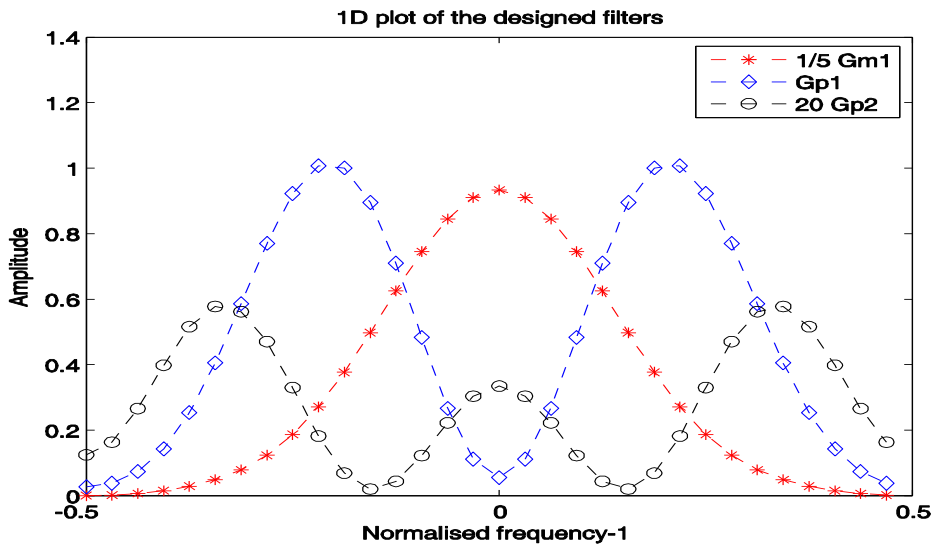


Figure 4.6: 1D plot of the designed rational filters

4.3.3. Model Verification

The designed model was verified by working backwards to determine how well the designed filters fit the theoretical $\frac{M}{P}$ ratio. At this point it needs to be mentioned that Watanabe and Nayar [14] have not done any verification of their designs and since the numerical results for their 32 x 32 frequency samples were not available, comparison of the results was not feasible at this stage. Comparisons can be done only on the estimated depth maps as described in a later Section. However a rough comparison with their filters was done by transforming their 7x7 spatial kernels into 32 x 32 frequency responses. The results are presented in Section 4.5.

To provide a useful comparison with Watanabe's filters, the kernel size (k_s) and the number of frequency samples were chosen as per [14]. The designed filters were 7x7 with eight fold symmetry (as Watanabe's) and hence the number of independent filter coefficients required to provide the desired frequency response was 10. This was further reduced to 6 for a 5x5 kernel (refer to Section 5.2 for details). Having a 5x5 eight fold symmetric filter would be advantageous in-terms of computation, but 6 independent coefficients were too small to provide the desired frequency response [14], and hence lead to poor mapping of the $\frac{M}{P}$ curves. Increasing the kernel size would provide filters with smoother pass-band response, but this in-turn may increase the overall processing time of the application. Therefore the optimum kernel size was chosen as 7.

In the verification process, the frequency band up to which the $\frac{M}{P}$ ratio was monotonic was determined. Here, to enable a useful accuracy comparison with Watanabe [14], the results were based on the defocus condition $\frac{e}{Fe} = 2.307 pixels$. By using the table provided in Appendix 3, the following results were calculated: - (1) The distance between the near and the far-focussed images was $2e$; (2) The effective focal length was Fe; (3) The minimum frequency below which the response is suppressed by the pre-filter was $\min f_r$; and (4) The maximum frequency below

which the $\frac{M}{P}$ ratio is guaranteed monotonic was $\max f_r$. The results are summarised in Table 4.1.

Defocus condition	***e in pixels***	***Fe***	$\min fr \geq \frac{2}{k_s}$ $pixel^{-1}$	$\max fr = 0.73\frac{Fe}{e}$ $pixel^{-1}$	***Max blur diameter*** $\frac{2e}{Fe} \leq 0.73k_s$ $pixel$
$\frac{e}{Fe} = 2.307 pixels$ focal length f = 50mm Kernel size ks=7 Aperture diameter=6.5mm	17.746	7.6923	0.2857	0.3164	4.1614

Table 4.1: Calculated values for the defocus function of 2.307 pixels

From the Table it can be inferred that for the defocus condition $\frac{e}{Fe} = 2.307 pixels$, the frequency range lies between $0.2857 \leq f_r \leq 0.3160 pixel^{-1}$. A Matlab program was written to plot the theoretical $\frac{M}{P}$ ratio, the Linear Model and the Error Corrected Model for a range of frequencies and normalised depth values. Figure (4.7) shows the plots of the theoretical $\frac{M}{P}$ ratio, Linear Model and the Error Corrected Model. The mean square error estimates between the theoretical $\frac{M}{P}$ ratio to the Linear Model, and the theoretical $\frac{M}{P}$ ratio and the Error Corrected Model, for different radial frequencies are provided in Table 4.2. It can be inferred that the filters devised by the new method fit well with the theoretical ones. More results with simulated and real images are presented in later Sections.

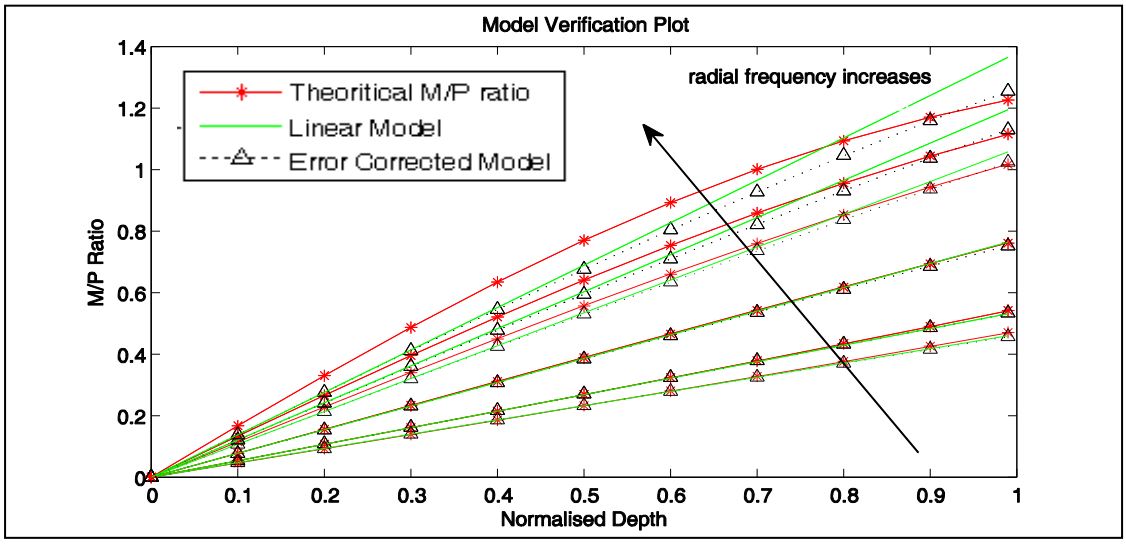


Figure 4.7: Model Verification Plot

Radial frequency in $pixel^{-1}$	MSE between Th.M/P ratio and Linear Model	MSE between Th.M/P ratio and Error Corrected Model
0.3141	0.0703	0.0636
0.3125	0.0630	0.0533
0.3078	0.0499	0.0397
0.2965	0.0296	0.0266

Table 4.2: Comparison of MSE between Linear Model and the Error Corrected Model

4.3.4. Summary of the algorithm for Rational Filter Design based on the Two Step Polynomial Approach

The procedure to determine the 7x7 filter coefficients capable of modelling the response of the Gm_1, Gp_1 and Gp_2 is as follows:-

- Pre-compute the psf for a range of normalised depth using the Pillbox psf equation given in equation (4.6).
- Discretize the $\frac{M}{P}$ space into 2^n equally spaced samples up to the nyquist critical frequency (*fnyquist*) of 0.5 $pixels^{-1}$, and determine the discrete $\frac{M}{P}$ ratio space by computing the $\frac{M}{P}$ ratio at each radial frequency and normalised depth, α . In the model, n = 5 and α was varied between 0 and 0.99.
- Consider the Linear Model, $\frac{M'(u,v;\alpha)}{P'(u,v;\alpha)} = \frac{Gp_1(u,v)}{Gm_1(u,v)}\beta$ and determine the gradient $A(f_r) = \frac{Gp_1(f_r)}{Gm_1(f_r)}$ by fitting a linear function of the form, $y = Ax$ to each radial frequency in the discrete $\frac{M}{P}$ ratio space.
- Model the response of $Gp_1(u,v)$ as a rotationally symmetric LOG filter [14] with a spread factor of f_{spread} = 0.4*fnyquist*. So that $Gp_1(f_r) = (\frac{f_r}{f_{spread}})^2 \exp(1-(\frac{f_r}{f_{spread}})^2)$ [the constant 0.4 ensures an acceptable width for the LOG filter]
- Once the gradient and the frequency response of filter Gp_1 are determined, the response of the filter Gm_1 can be modelled from the gradient as, $Gm_1(f_r) = \frac{Gp_1(f_r)}{A(f_r)}$.
- To ensure a smooth transition along with minimum depth error, a higher order 2D polynomial with a weighting function described in [14] was fitted to the response of the filter Gm_1. The polynomial takes care of the DC value at

zero frequency which would otherwise need to be added as in the previous method. The weighting function used is described in the Appendix1.

- To determine the response of the filter Gp_2, the error between the Theoretical $\frac{M}{P}$ ratio and Linear $\frac{M'}{P'}$ was modelled as cubic function, $y = Cx^3$ where the gradient, $C(f_r) = \frac{Gp_2(f_r)}{Gm_1(f_r)}$ and $f_r = \sqrt{u^2 + v^2}$.
- Once the gradient, $C(f_r)$ has been determined the response of the filter Gp_2 can be modelled directly from the gradient as Gm_1 has been determined earlier. So from the gradient, $Gp_2(u,v) = C(f_r)Gm_1(u,v)$. Again a higher order polynomial was fitted to smooth the response.
- Finally the 7x7 filter coefficients are obtained by inverse Fourier transforming the frequency response of the designed filters as described in Section 4.4.2.

4.4. Pre-processing and Spatial Transformation of the filters

This Section discusses the need for a pre-filter and the transformation of the 2^n x 2^n frequency samples into the corresponding 7x7 spatial coefficients. Once the spatial equivalents of the designed filters have been determined, depth can be resolved by convolving the defocused images with their respective kernels as described in Section 4.6.

4.4.1. Pre-filter

The purpose of the pre-filter is to remove the DC component, and the high frequency components which violate the monotonic requirement of the $\frac{M}{P}$ ratio. The images are pre-filtered before the rational filters are applied. Since Gm_1 is a low pass filter, any DC component would propagate into the depth algorithm causing uncertainties in the estimation. The pre-filter needs to be a band-pass filter with rotational symmetry. From [14] it was found that equation (4.16), used to design the filter Gp_1 can also be used to design the pre-filter if the spread factor f_{peak} = 0.74*fmax*, where

$fmax = 0.264\ pixels^{-1}$. More details are provided in Appendix2. The magnitude response of the pre-filter is shown in Figure (4.9). It should be noted that the response of the pre-filter is not rotationally symmetric, but can be further refined if its kernel size is larger. This non-symmetry does not affect the depth estimation, since the $\frac{M}{P}$ ratio mainly depends on the filters Gm_1, Gp_1 and Gp_2. These filters are 8 fold rotationally symmetric [4 fold symmetry with 2 fold refection symmetry] and thus provide uniform sensitivity to textures in all directions. Further the sharp transition of the pre-filter at the lower stop-band and the increase in the width of the pass-band has provided a smooth roll-off when compared with Watanabe's filters. Additionally these constraints have relaxed the specifications of the correction filter Gp_2, allowing it to have a sharper transition (refer to Section 4.5) and lower attenuation at DC. This enabled a good fit to the theoretical $\frac{M}{P}$ ratio.

4.4.2. Design of 7x7 Spatial Kernels

The frequency response of the filters have a total of 2^n x 2^n samples, where n = 5 in the proposed model. Transforming them directly into the spatial domain and convolving them with the image would result in a computationally expensive process. It was found from [52] that when the filter is 8 way symmetric and has 10 degrees of freedom, it would require a minimum of 10 distinct spatial coefficients to provide the desired magnitude response. Hence the frequency samples were down sampled by a factor 'm' (here m = 4) and inverse Fourier transformed. It was observed that filter coefficients along the border were redundant and the central 7x7 coefficients are the required spatial coefficients that generate the desired frequency response. An example set of the 7x7 filter kernels and their respective magnitude responses are shown in Figures (4.8) and (4.9).

$$gm_1 = \begin{bmatrix} -0.0001 & 0.0038 & 0.0146 & 0.0201 & 0.0146 & 0.0038 & -0.0001 \\ 0.0038 & 0.0277 & 0.0834 & 0.1197 & 0.0834 & 0.0277 & 0.0038 \\ 0.0146 & 0.0834 & 0.2629 & 0.3884 & 0.2629 & 0.0834 & 0.0146 \\ 0.0201 & 0.1197 & 0.3884 & 0.5770 & 0.3884 & 0.1197 & 0.0201 \\ 0.0146 & 0.0834 & 0.2629 & 0.3884 & 0.2629 & 0.0834 & 0.0146 \\ 0.0038 & 0.0277 & 0.0834 & 0.1197 & 0.0834 & 0.0277 & 0.0038 \\ -0.0001 & 0.0038 & 0.0146 & 0.0201 & 0.0146 & 0.0038 & -0.0001 \end{bmatrix}$$

$$gp_2 = \begin{bmatrix} 0.0008 & -0.0008 & -0.0036 & -0.0038 & -0.0036 & -0.0008 & 0.0008 \\ -0.0008 & -0.0029 & 0.0019 & 0.0062 & 0.0019 & -0.0029 & -0.0008 \\ -0.0036 & 0.0019 & 0.0059 & -0.0025 & 0.0059 & 0.0019 & -0.0036 \\ -0.0038 & 0.0062 & -0.0025 & -0.0277 & -0.0025 & 0.0062 & -0.0038 \\ -0.0036 & 0.0019 & 0.0059 & -0.0025 & 0.0059 & 0.0019 & -0.0036 \\ -0.0008 & -0.0029 & 0.0019 & 0.0062 & 0.0019 & -0.0029 & -0.0008 \\ 0.0008 & -0.0008 & -0.0036 & -0.0038 & -0.0036 & -0.0008 & 0.0008 \end{bmatrix}$$

$$gp_1 = \begin{bmatrix} -0.0022 & -0.0079 & -0.0170 & -0.0229 & -0.0170 & -0.0079 & -0.0022 \\ -0.0079 & -0.0323 & -0.0477 & -0.0444 & -0.0477 & -0.0323 & -0.0079 \\ -0.0170 & -0.0477 & 0.0362 & 0.1412 & 0.0362 & -0.0477 & -0.0170 \\ -0.0229 & -0.0444 & 0.1412 & 0.3340 & 0.1412 & -0.0444 & -0.0229 \\ -0.0170 & -0.0477 & 0.0362 & 0.1412 & 0.0362 & -0.0477 & -0.0170 \\ -0.0079 & -0.0323 & -0.0477 & -0.0444 & -0.0477 & -0.0323 & -0.0079 \\ -0.0022 & -0.0079 & -0.0170 & -0.0229 & -0.0170 & -0.0079 & -0.0022 \end{bmatrix}$$

$$\text{pre-filter} = \begin{bmatrix} -0.0020 & -0.0434 & -0.0307 & -0.0096 & -0.0307 & -0.0434 & -0.0020 \\ -0.0434 & -0.0541 & -0.0107 & 0.0231 & -0.0107 & -0.0541 & -0.0434 \\ -0.0307 & -0.0107 & 0.0633 & 0.1098 & 0.0633 & -0.0107 & -0.0307 \\ -0.0096 & 0.0231 & 0.1098 & 0.1616 & 0.1098 & 0.0231 & -0.0096 \\ -0.0307 & -0.0107 & 0.0633 & 0.1098 & 0.0633 & -0.0107 & -0.0307 \\ -0.0434 & -0.0541 & -0.0107 & 0.0231 & -0.0107 & -0.0541 & -0.0434 \\ -0.0020 & -0.0434 & -0.0307 & -0.0096 & -0.0307 & -0.0434 & -0.0020 \end{bmatrix}$$

Figure 4.8: Derived filter kernels for the defocus condition of 2.307 pixels

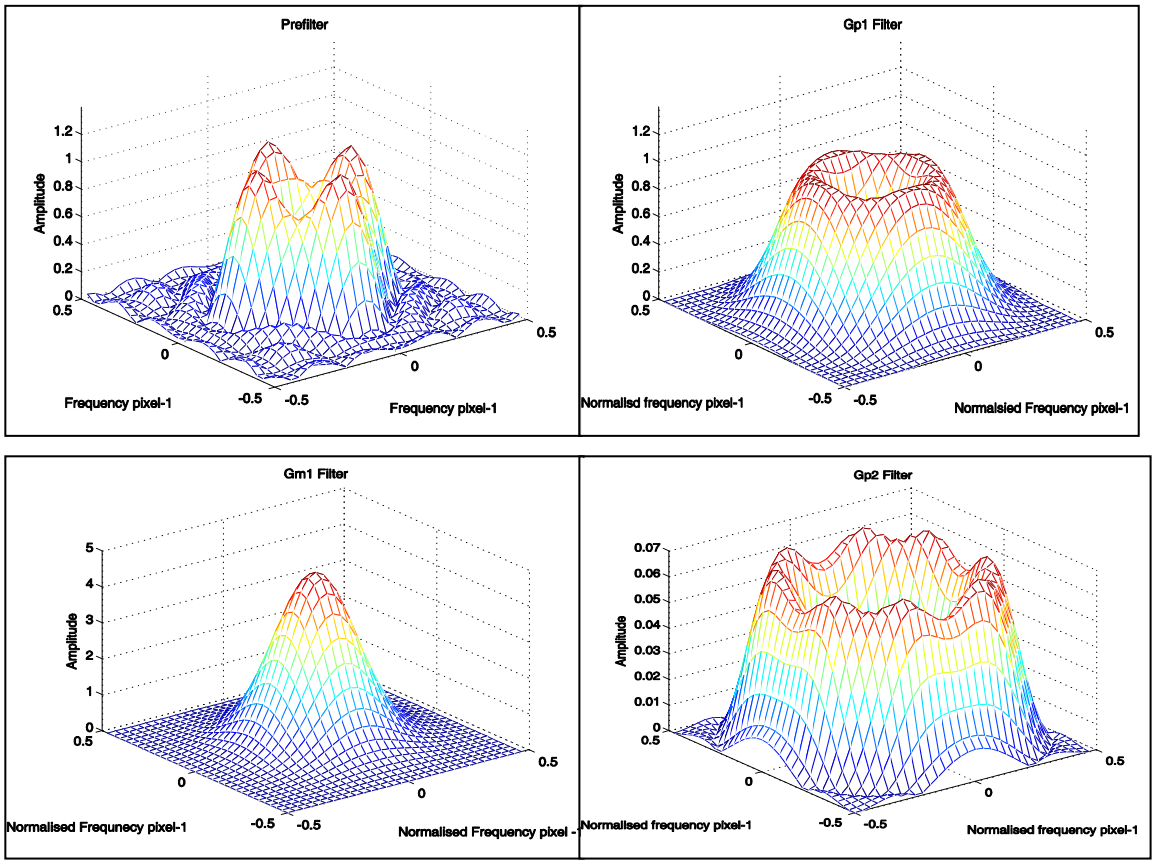


Figure 4.9: Magnitude responses of the designed filters for defocus condition of 2.307 pixels

4.5. Comparison with Watanabe and Nayar Filters

This Section provides a detailed comparison between the filters designed by the Two Step Polynomial Approach and those designed by Watanabe and Nayar [14]. The 7x7 filter coefficients are transformed into their 32x32 equivalent frequency samples (frequency response) and verified as to how well they fit the theoretical $\frac{M}{P}$ ratio. Figure (4.10) shows the plot of the theoretical $\frac{M}{P}$ ratio, Watanabe's Model, and the Two Step Polynomial model. Here the maximum frequency applicable for the defocus condition $\frac{e}{Fe} = 2.307 pixels$ was used (refer to Table 4.1). Figure (4.11) shows the RMS error plots for Watanabe and Nayar's model and for the Two Step Polynomial model for all the frequencies within the applicable range. The RMS error was lower for the Two Step Polynomial method particularly as the normalised depths approaching, and hence the design was better than Watanabe and Nayar's filter.

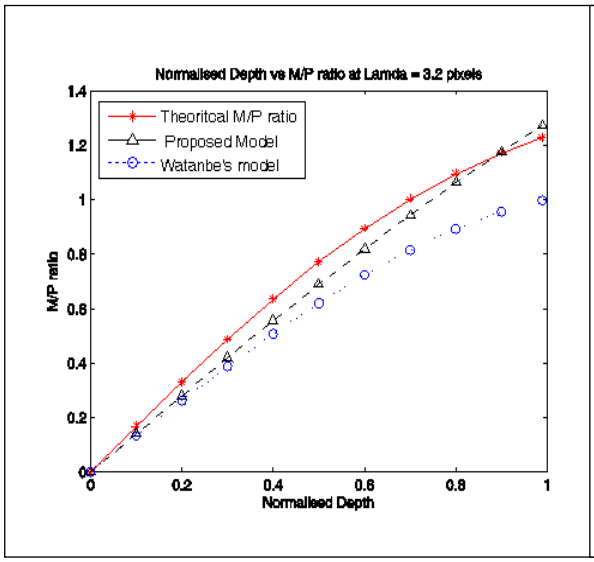


Figure 4.10: Normalised depth vs. Theoretical M/P ratio for both the models

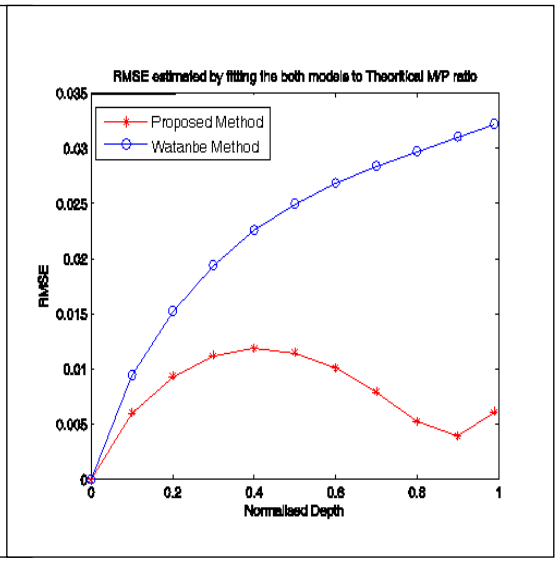


Figure 4.11: RMSE between Theoretical M/P ratio for both the models

In the next step, the normalised magnitude and phase responses of the designed filters are compared with Watanabe and Nayar's filters. The 1D magnitude response (see Figure (4.12)) of the linear filters Gm_1 and Gp_1 for both the models are quite similar but there is a considerable dissimilarity in the response of the correction filter Gp_2. It is noted that the Gp_2 designed by the new model has a sharper transition (from pass band to stop band) and a higher DC magnitude compared to Watanabe's model. The DC does not propagate in the depth estimation since the pre-filter suppresses any frequency response below the minimum cut-off frequency. Moreover the pre-filter designed by the new method has a smooth roll-off in the transition band compared to sharp transition of Watanabe's pre-filter which can propagate a ringing effect [52].

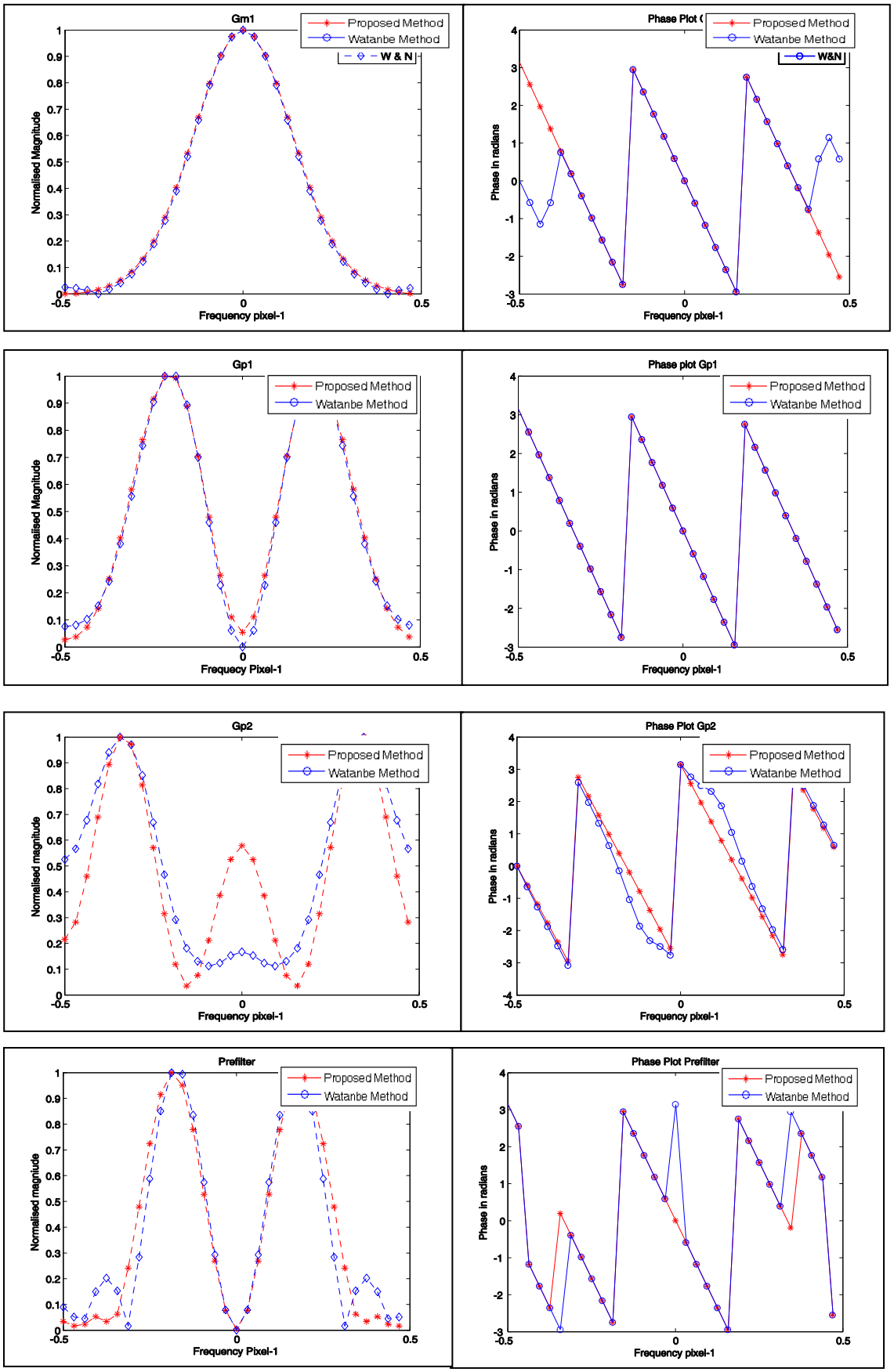


Figure 4.12: Magnitude and Phase response of Gm_1, Gp_1, Gp_2 and Pre-filter.

Finally the depth maps generated by both the filters have been compared using a circular sinusoidal test pattern (see Figure (4.13a)) with a wavelength of $\lambda = 3.2$ pixels (the maximum frequency applicable for the defocus condition $\frac{e}{Fe} = 2.307 pixels$) and the normalised depth of $\alpha = 0.99$. The depth maps generated using both the models are shown in Figure (4.13b). The mean depth error and standard deviation for the Two Step Polynomial model was 0.0454 and 0.0128, and for Watanabe's model 0.3615 and 0.2008 respectively. From the standard deviation results it can be inferred that the depth map generated by the new filters is smooth compared to Watanabe's and the relative non-circularity of the Watanabe's filters have resulted in a less planar pattern in the depth map.

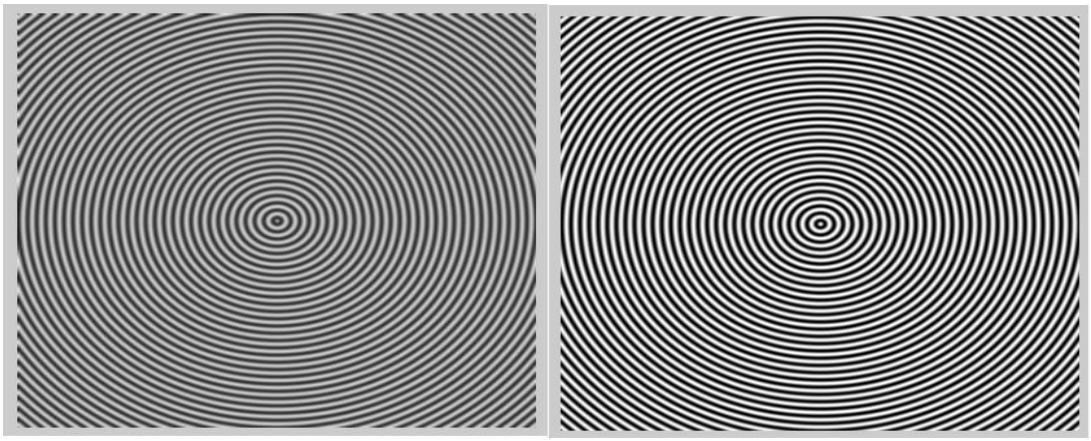


Figure 4.13a: Single frequency sinusoidal test pattern near-focused (Left) far-focused (Right)

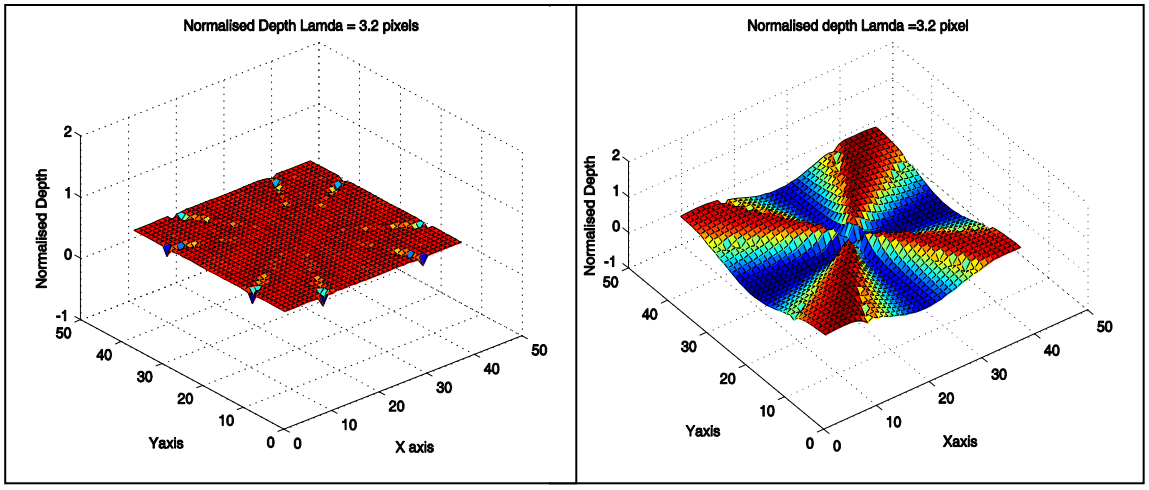


Figure 4.13b: Depth map estimated using the filters designed by the proposed method (Left) and from Watanabe's filters (Right)

4.6. Algorithm for Depth Estimation

The algorithm described in [14] was used to estimate the depth using the filters designed by the Two Step Polynomial Approach. Here the far and the near-focused images were added, subtracted, and then convolved with the pre-filter to remove DC as well as high frequency components. The low pass filter gm_1 was convolved with the subtracted image and LOG filter, gp_1 and the correction filter, gp_2 were convolved with the added image. To avoid uncertainties due to division by zero, the images underwent a smoothing process by local averaging, and the depth map was recovered using the Newton-Raphson method. Finally to ensure smoothness, the recovered depth map was post-filtered by a 9x9 median filter. The experimental results with simulated and real images are presented in the next Section.

4.7. Experimental Results with Simulated Images

In order to verify the design model, sinusoidal patterns with a single spatial frequency and different normalised depth values (α) were developed (see Figure (4.14)). Since the defocus condition used for the simulation was $\frac{e}{Fe} = 2.307 pixels$, the usable frequency range lay between $0.2857 \leq fr \leq 0.3160 pixel^{-1}$ (see Table 4.1) and therefore the wavelength λ lies between $3.2 \leq \lambda \leq 3.5 pixels$. The normalised depth range used was between 0.1 and 0.99. To produce a depth staircase, the single frequency test images were defocused using the Pillbox psf model in a way that for every 40 pixel along the horizontal axis there was a step change in depth. This simulation enabled the estimated depth map to be viewed as a 3D staircase structure. Experiments were performed on two test images with wavelength of λ=3.5 pixels and 3.2 pixels. The estimated depth maps are shown in Figure (4.15). The resolution of the images used was 400 x 400 pixels, but for illustration purposes, depth maps from a local area of 38 x 38 pixels are shown. The linearity and the smoothness of the depth estimated by the filter coefficients designed by the proposed method and Watanabe's model are compared in Figure (4.16a) and Figure (4.16b). It can be inferred that for wavelength λ = 3.5 pixels, which corresponded to a lower radial frequency, f_r = 0.2857, the depth map estimated by both the filter models are reasonably linear but for a lower wavelength of 3.2 pixels (higher radial frequency f_r

= 0.3125), the filter coefficients designed by the Two Step Polynomial Approach provided a smoother and more accurate fit to the actual depth than Watanabe's filters. This increase in accuracy can be attributed to the new design model which fits more closely to the theoretical $\frac{M}{P}$ ratio (refer to Figures (4.10) and (4.11)). The statistics were calculated from a local area of 17x371 pixels which fitted well along each individual step.

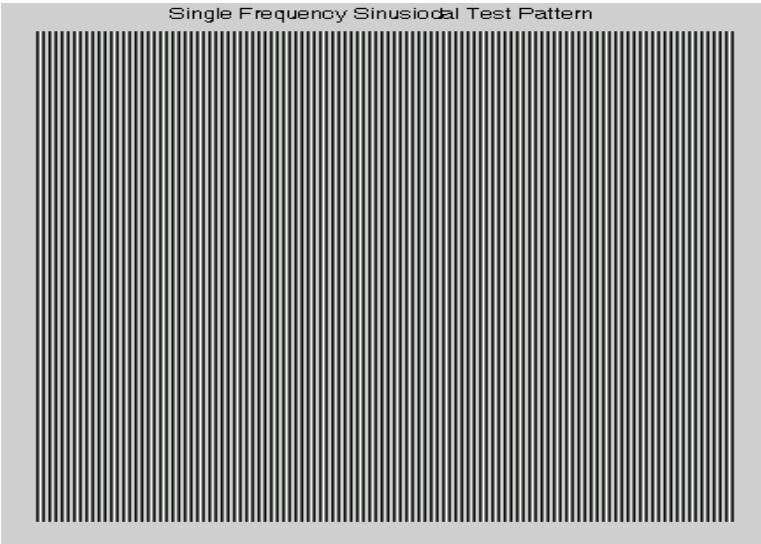


Figure 4.14: Single frequency sinusoidal test pattern with wavelength λ = 3.5 pixels

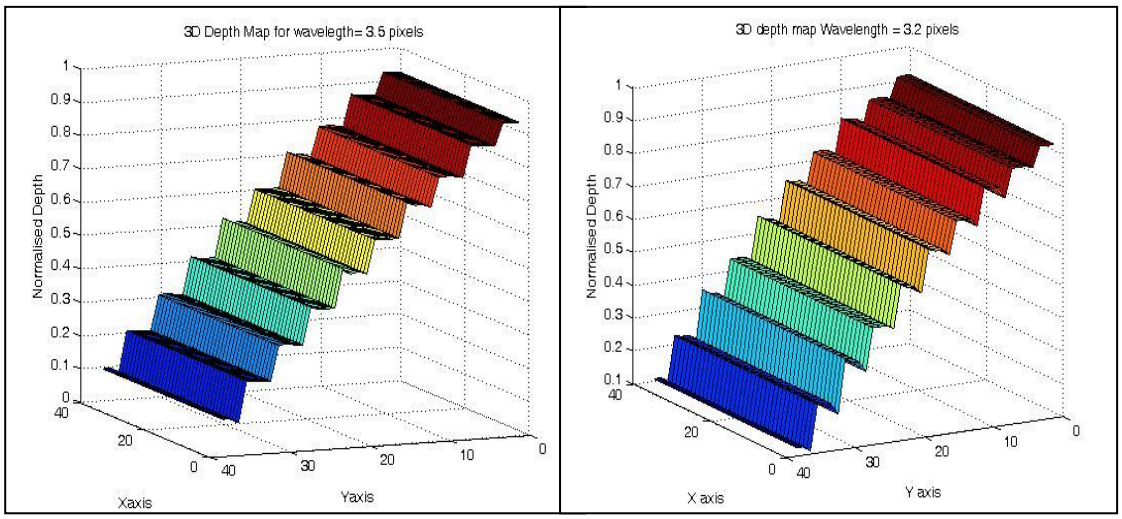


Figure 4.15: Depth Map for λ = 3.5 pixels (Left) and the depth map for λ = 3.2 pixels (Right)

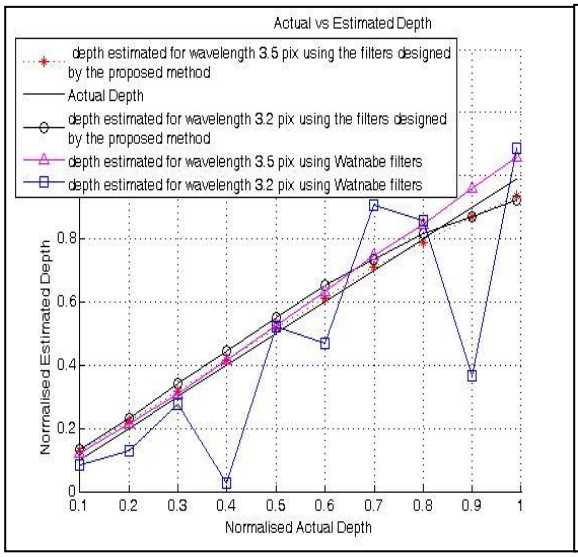


Figure 4.16a: Actual vs. Estimated depth at different normalised depths using filters designed by Two Step Polynomial approach and filters designed by Watanabe

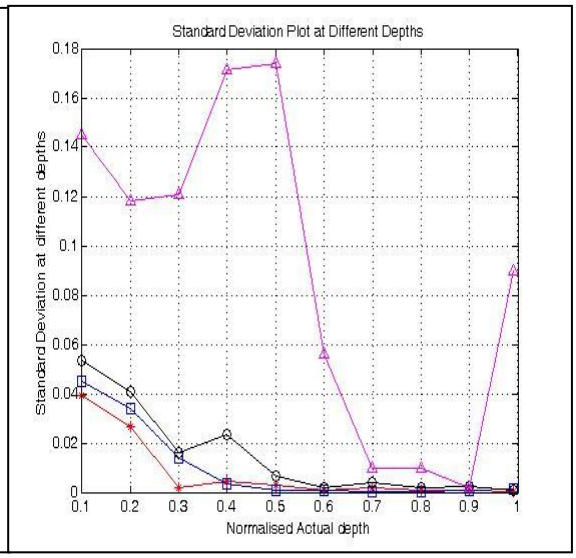


Figure 4.16b: Standard Deviation plot at different depths for both the design models

To verify the invariance of the filter coefficients to the image texture, a textured pattern devised by Watanabe [14] was used and the original pattern was defocused using the Pillbox psf to simulate a 3D staircase structure as explained earlier. The near and far-focussed images along with the gray depth map are shown in Figures (4.17) and (4.18).

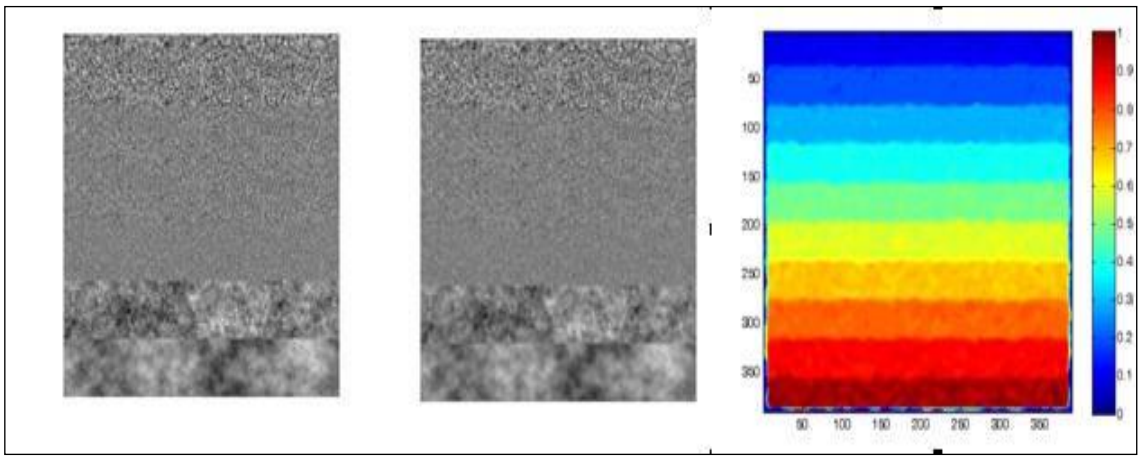
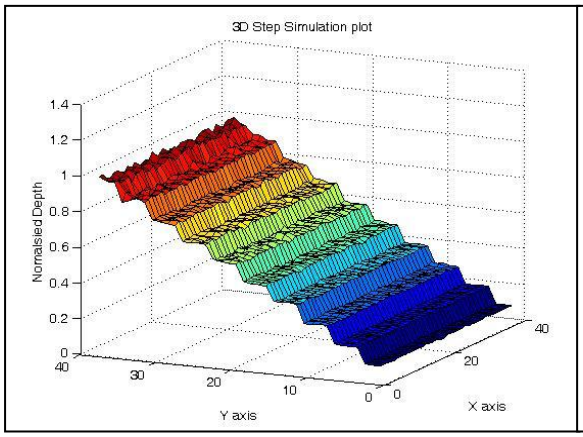


Figure 4.17: Near and far focussed images

Figure 4.18: Gray scale depth map



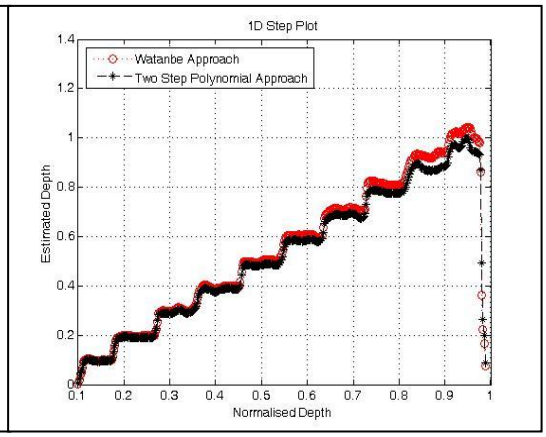
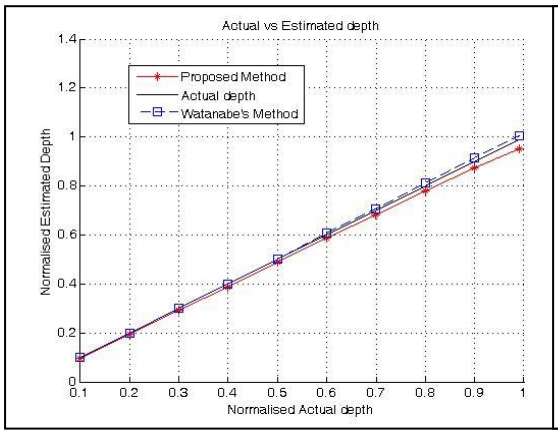


Figure 4.19a: 3D view of the estimated depth

Figure 4.19b: 1D plot of the estimated depth using filters designed by both the models.



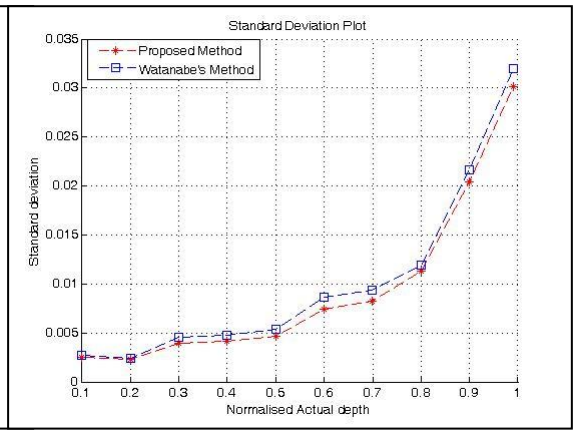


Figure 4.20a: Actual vs. estimated depth for filters designed by both the models

Figure 4.20b: Standard deviation plot at different depths for both the models.

Figure (4.19a) shows the estimated depth in 3D, and Figure (4.19b) shows the depth estimated at different normalised depths for the filters designed by the new method and those designed by Watanabe and Nayar. The linearity and the smoothness of the depth estimates for both the models are compared in Figures (4.20a) and (4.20b). It can be inferred that the filter coefficients designed by the new method are invariant to texture and provide a better fit to the actual depth than Watanabe's filters. In the next Section, experiments are performed on real images and the accuracy of the depth estimated using the designed filters are reported.

4.8. Experiments to determine the accuracy of the designed model

In this Section the accuracy of the designed filters was determined by using them to measure a known distance. In experiments 1 and 3 the results are compared with Watanabe and Nayar's filters and in experiments 2 and 4, since the defocus setting was changed from that of Watanabe's, a new set of filter coefficients was determined. In all the experiments the depth estimation results along with RMS error plots are provided. For these experiments a checkerboard image was used as the test pattern. Experiments with natural textures are presented in chapter 6.

4.8.1. Experiment 1- with defocus condition 2.307 pixels

The apparatus included a 50mm photographic quality lens with an external aperture diameter set to 6.5mm, and a monochrome camera with a CCD sensor of pixel size 7.4 x $7.4\mu m$. To enable a useful accuracy comparison with Watanabe [14], the defocus condition was set to $\frac{e}{Fe} = 2.307 pixels$. Based on Appendix3, the working range was calculated to be 56mm, this is quite short but it is limited by the pixel size of the camera and the aperture set. A larger pixel size or a narrower aperture would have provided an increase in the working distance. The far-focussed image was set at 800mm and the near-focussed at 744mm. A checkerboard pattern was moved along the optical path between these points and a pair of defocused images was recorded at every 10mm interval. The normalised mean depth was calculated and mapped to the real world coordinates using the Gaussian lens law.

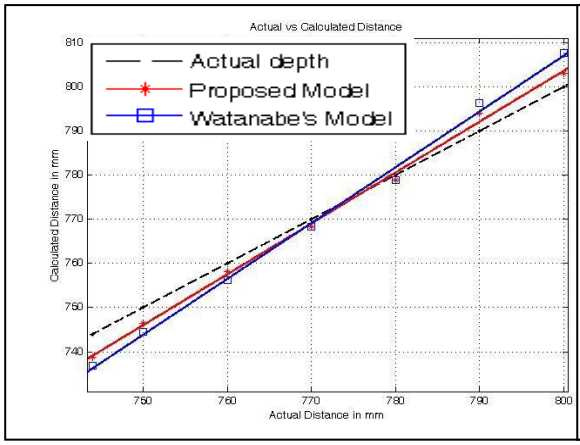


Figure 4.21a: Actual Distance vs. Estimated Dist. (mm)

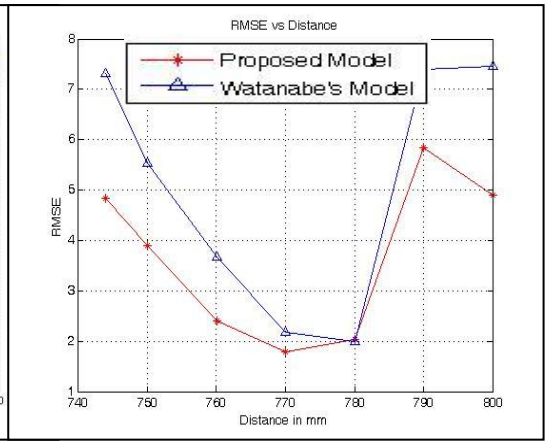


Figure 4.21b: Act. Dist. vs. RMSE (mm)

Figure (4.21a) shows the plots of the actual and the estimated depth. For these results, a centre offset correction was performed to compensate for the experimental error while determining the centre of the compound lens. A detailed description is given in Appendix 4. The RMS error plot for the depth range is shown in Figure (4.21b). The RMS error for the new method was 0.6122% at the far-focussed and 0.6516% at the near-focussed planes, and for Watanabe the errors were 0.9321% and 0.98425% respectively. From the plots it is seen that the depth estimates are reasonably linear but the filters designed using the Two Step Polynomial Approach provided a better fit to the actual depth compared to the Watanabe filters.

4.8.2. Experiment 2 - with defocus condition 2.3587 pixels

In the second experiment the working distance was extended to 140mm by setting a smaller aperture of 2.27mm. The far-focussed and the near-focussed images were at 800mm and 660mm respectively, and the defocus condition based on Appendix3 was $\frac{e}{Fe} = 2.3587\,pixels$. The parameters are summarised in Table 4.3 and a new set of filter coefficients were designed and used for depth estimation. For this defocus condition there were no Watanabe's results available for comparison.

Defocus condition	e *pixels*	Fe	$\min fr \geq \frac{2}{k_s}$ $pixel^{-1}$	$\max fr \leq 0.73\frac{Fe}{e}$ $pixel^{-1}$	Max blur diameter $\frac{2e}{Fe} \leq 0.73k_s$ $pixel$
$\frac{e}{Fe} = 2.3587\,pixels$ Focal length, f =50mm Kernel size k_s=7 Aperture diameter=2.27mm	51.891	22	0.2857	0.3107	4.7174

Table 4.3: Calculated values for the defocus condition $\frac{e}{Fe} = 2.3587\,pixels$

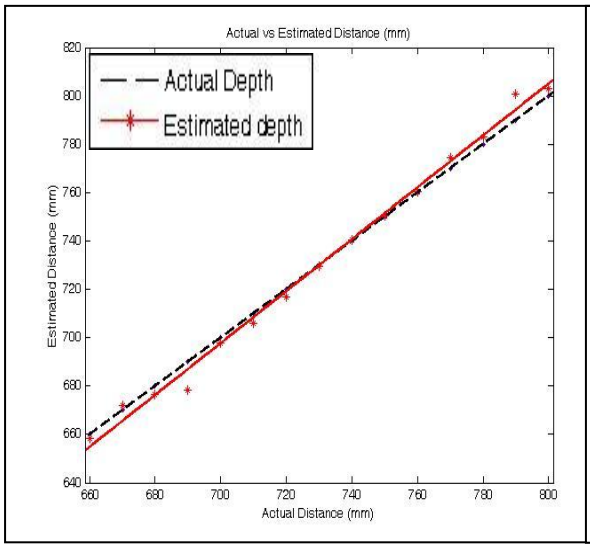


Figure 4.22a: Actual Distance vs. Estimated Distance (mm)

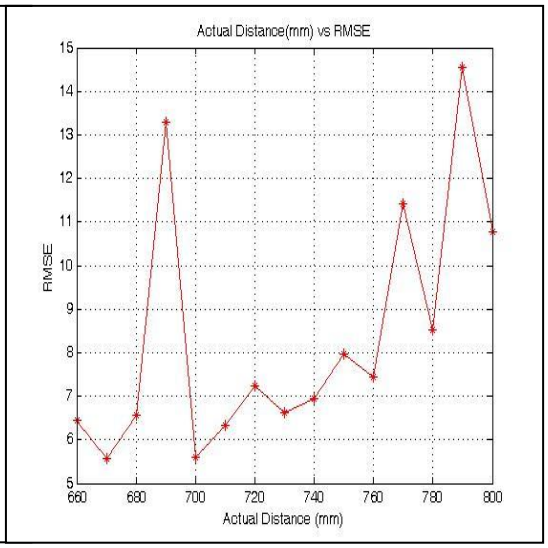


Figure 4.22b: Actual Distance vs. RMSE (mm)

Again the square pattern was moved in steps of 10mm along the optical axis and the normalised depth was calculated from the two defocused images. Figure (4.22a) shows the plot between the actual and the estimated depth, and the plot in Figure (4.22b) shows the RMS error estimated at the individual distances. The RMS error with respect to the distance from the lens was between 0.8310% and 1.8427%. The increase in RMS error can be attributed to the increase in working distance and to the decrease in aperture size which results in darker images. The sharp increase in RMS error at distances of 790mm and 690mm coincides with the theoretical model as shown in Figure (4.11). The distance of 730mm corresponds to the centre of range where normalised depth was zero.

4.8.3. Experiment 3 - with defocus condition 2.307 pixels

In the third experiment a 35mm photographic lens was used and the external aperture set to 4.55mm giving the defocus condition $\frac{e}{Fe} = 2.307 pixels$. Here the results were compared with Watanabe's filters. The working distance calculated based on the procedure in Appendix3 was 107mm; the far-focussed image was set at 800mm and the near-focussed image at 693mm. The normalised depth was calculated at 10mm intervals. Figure (4.23a) shows the plots of the actual and estimated depths for Watanabe's filters, and for the filters designed by the new method. The RMS error

(refer Figure (4.23b)) was plotted with respect to the distance from the lens with values between 0.8291% and 1.3496%, and for Watanabe's filters the error was between 0.8691% and 1.5301%. From the plots it can be inferred that the filters designed by the Two Step Polynomial method provide a closer fit to the actual depth. The results are corrected for centre offset.

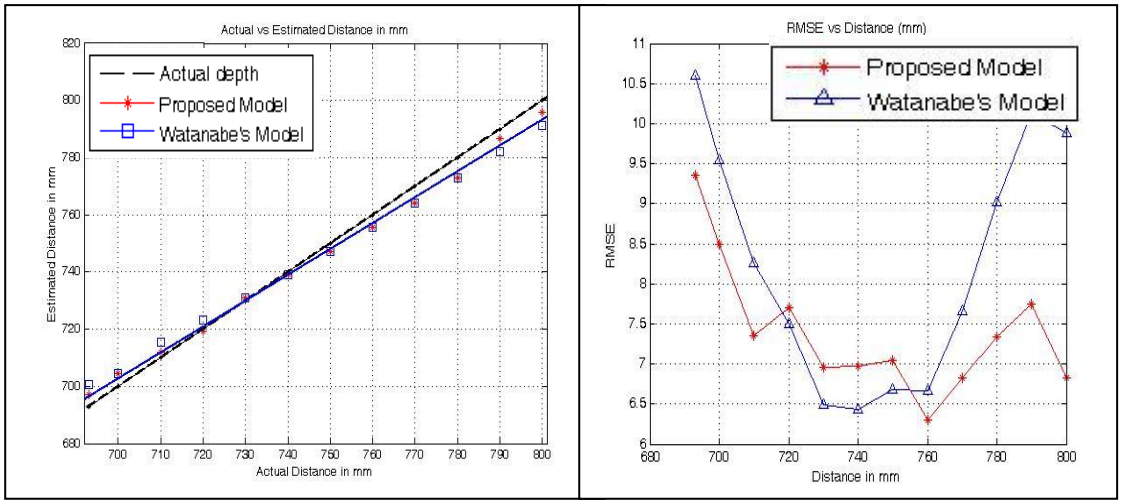


Figure 4.23a: Actual vs. Estimated Distance (mm)

Figure 4.23b: Actual Distance vs. RMSE (mm)

When compared with the results in experiment (1) (Section 4.8.1), the defocus condition was the same, but a different lens was used with different working ranges. In each case the estimated depth was linear with the step number, and the RMS error followed a similar shape. However the RMS error was higher for the 35mm lens with a working range of 107mm when compared to a 50mm lens with a working range of 56mm. Here it should be observed that the increase in working range by decreasing the aperture size has had a considerable effect on the accuracy of depth estimation. This led to an investigation of the available options to increase the working range for a given experimental setup. The detailed description is presented in Section 4.9.

4.8.4. Experiment 4- with defocus condition 2.3944 pixels

In the final experiment a 35mm lens was used but the working distance was extended to 200mm by using a smaller aperture of 2.2mm. The far and near-focussed images were at 800mm and 600mm, and the defocus condition calculated based on Appendix3 was $\frac{e}{Fe} = 2.3944\,pixels$. The rational filters were redesigned for the parameters summarised in Table 4.4.

Defocus condition	e *pixels*	Fe	$\min fr \geq \frac{2}{k_s}$ $pixel^{-1}$	$\max fr = 0.73\frac{Fe}{e}$ $pixel^{-1}$	Max blur diameter $\frac{2e}{Fe} \leq 0.73k_s$ $pixel$
$\frac{e}{Fe} = 2.3944\,pixels$ Focal length, f =35mm Kernel size k_s=7 Aperture diameter=2.2mm	38.310	16	0.2857	0.305	4.78

Table 4.4: Calculated values for the defocus condition $\frac{e}{Fe} = 2.3937\,pixels$

The normalised depth was calculated at 20mm intervals. Figure (4.24a) shows the plots of the actual and the estimated depths, and plot (4.24b) shows the RMS error estimated for individual distances. The maximum RMS error recorded was 7.3%. This large RMS error could be due to the smaller aperture used to increase the working range and also due to the focal error that might be present in the lens used. In order to clearly show linearity of the depth estimation with respect to the actual depth, the results have been corrected for both focus offset and centre offset.

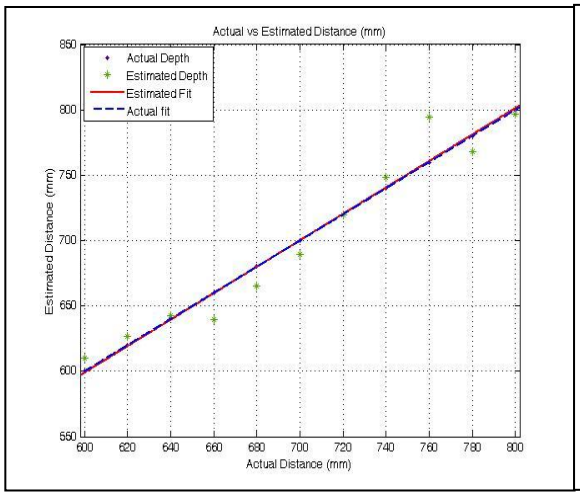


Figure 4.24a: Actual Distance vs. Estimated Distance (mm)

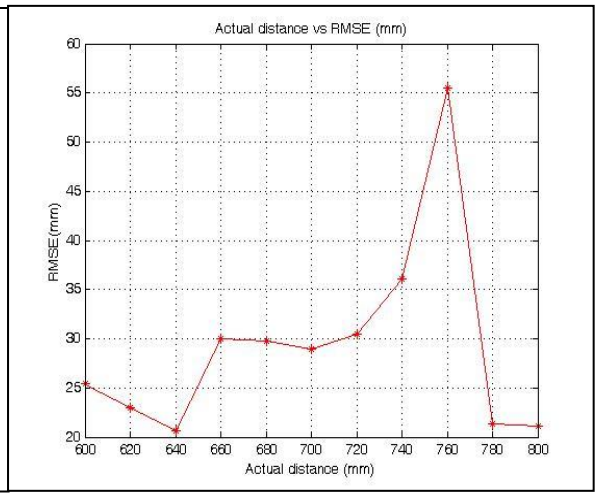


Figure 4.24b: Actual Distance vs. RMSE (mm)

4.9. Effect of focal length, *f*-number of the lens and the pixel size of the sensor on the Rational filter design and Working distance

The depth estimation error varied significantly between experiments 1 to 4 of Section 4.8, so the effect of focal length, *f*-number of the lens and the pixel size of the sensor, on the defocus condition and working distance were investigated. Numerically, the appropriate working distance for two different lenses (35mm and 50mm), two different CCD sensors with pixel size 7.4µm and 13µm, and several different aperture settings were calculated. Figures (4.25a) and (4.25b) show the appropriate working distance for a camera with a pixel size 13µm against different *f*-numbers when the focal length was 50mm and 35mm respectively. Similarly Figures (4.26a) and (4.26b) show the appropriate working distance for a camera with a pixel size of 7.4µm against different *f*-numbers when the focal length was 50mm and 35mm respectively. In each case the results were simulated with the far-focussed image at 800mm and the arrowed dotted line parallel to the horizontal axis represents the minimum frequency ($2/k_s$ where k_s=7) below which the frequency response would be suppressed by the pre-filter.

→ Minimum frequency

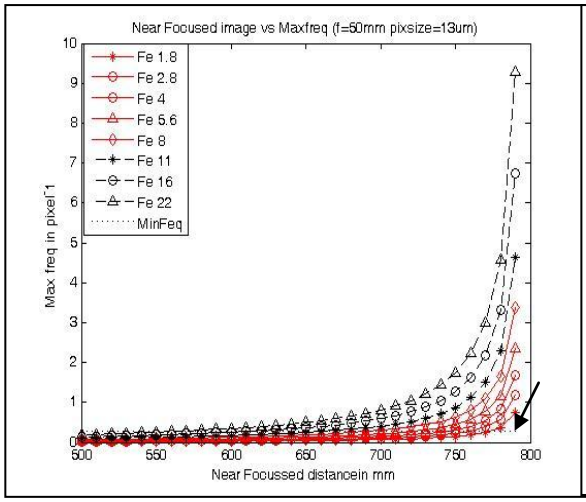


Figure 4.25a: Working Distance for a 50mm lens with pixel size of 13µm against different aperture settings

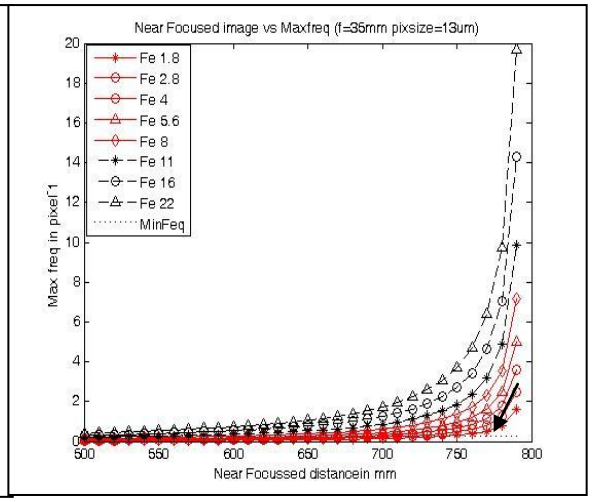


Figure 4.25b: Working Distance for a 35mm lens with pixel size of 13µm against different aperture settings

→ Minimum frequency

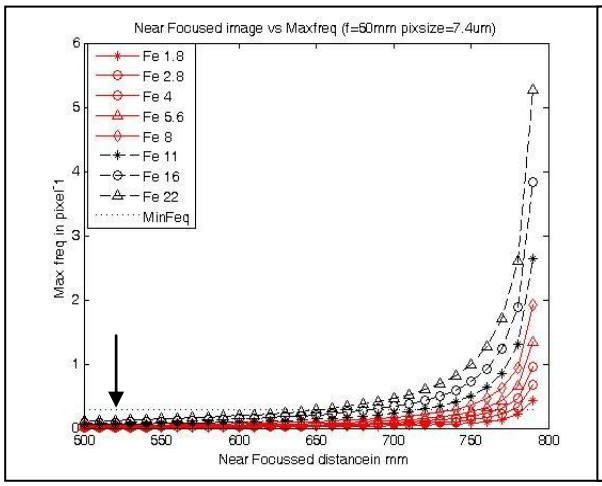


Figure 4.26a: Working Distance for a 50mm lens with pixel size of 7.4µm against different aperture settings

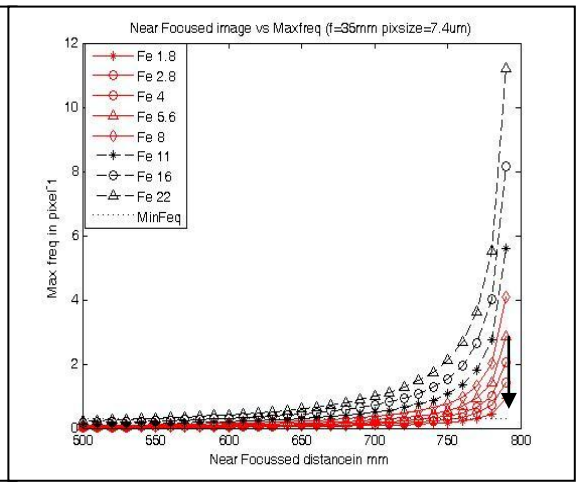


Figure 4.26b: Working Distance for a 35mm lens with pixel size of 7.4µm against different aperture settings

From the plots it can be inferred that the working distance can be increased by:- (1) Reducing the aperture size; (2) Reducing the focal length of the lens; and (3) By using a CCD sensor with a larger pixel size. Since the depth experiments were based on a pixel size of 7.4µm, the options 1 and 2 were practically verified and the parameters based on Appendix 3 are summarised in the previous Section. A 1D comparison of the normalised frequency responses of the filters designed for:- (1) Defocus condition of 2.307 pixels (Watanabe's defocus condition) used in experiments 1 and 3; (2) Defocus condition of 2.3944 pixels used in experiment 4; and (3) Defocus condition of 2.358 pixels used in experiment 2 are shown in Figure (4.27). It can be observed that though the filters have been designed for different experimental setups (based on different *f*-number or focal length), their frequency responses have a similar shape. From the calculations based on the method in Appendix 3, it can be inferred that whenever the working distance was at the maximum range, the defocus condition $\frac{e}{Fe}$ always lies close to 2.3 pixels, and hence the rational filters designed for defocus condition say $\frac{e}{Fe} = 2.3 pixels$ can be effectively used to recover depth of an object in any setup (different *f*-number or focal length) provided the maximum working distance is used.

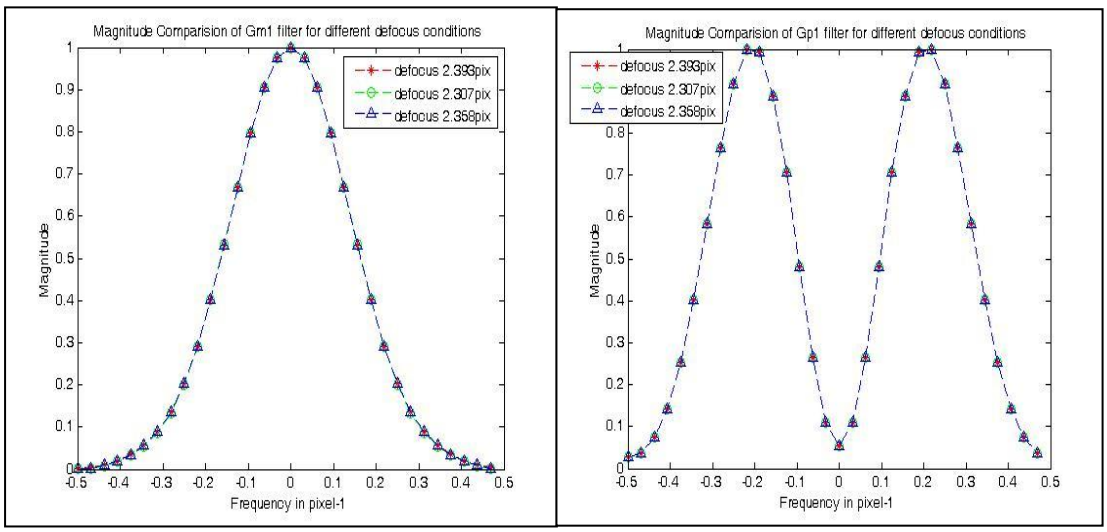


Figure 4.27: Magnitude plots of Gm_1, Gp_1, Gp_2, and Pre-filter (left to right) designed for different experimental setups

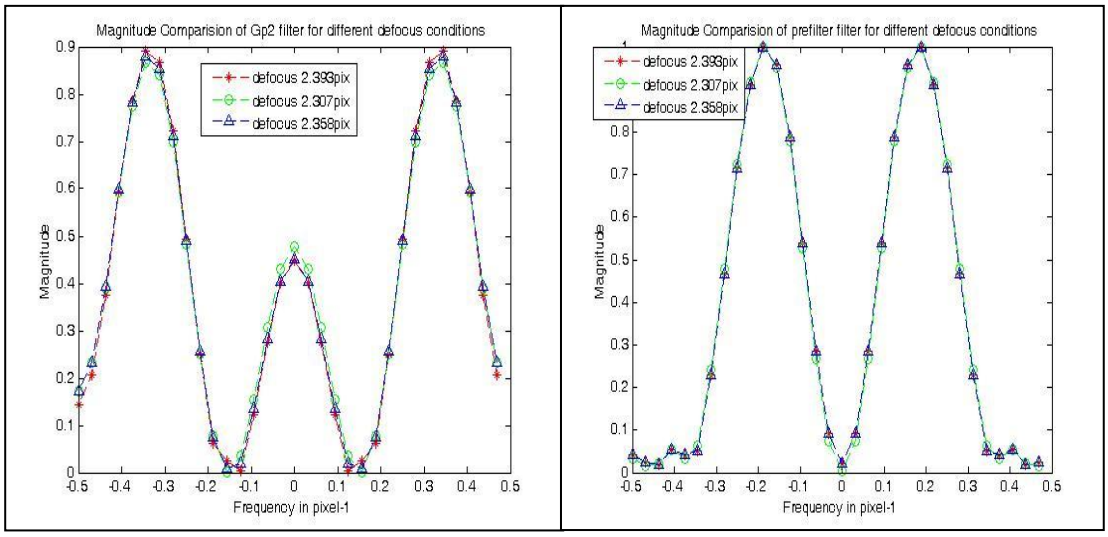


Figure 4.27: (continued) Magnitude plots of Gm_1, Gp_1, Gp_2, and Pre-filter (left to right) designed for different experimental setups

4.9.1. Discussion

From the experiments, it was found that when the *f*-number of the lens was close to 8 the maximum RMS error over a range of 56mm for the 50mm lens was 0.6516% and for the 30mm lens over a range of 107mm was 1.3496 %. It was evident from experiments (2) and (4) that an increase in working distance achieved by decreasing the aperture size (increasing the *f*-number) leads to erroneous depth estimation. Thus it can be concluded that there is a lower optimum limit for the aperture diameter. Based on the experiments, an *f*-number close to 8 provided acceptable depth accuracy over a wide working distance.

Increasing the working distance by reducing the focal length does have some practical problems with locating the front focal plane of the lens and converting it for telecentricity. In the case of the 50mm lens the front focal plane was 25mm outside the lens, and for the 35mm the front focal plane was on the lens outer surface. So converting the lens to telecentric by fixing an external aperture was not complicated. But for the case of a standard 16mm or 25mm lens the front focal plane can reside within the lens. For a complex photographic lens with many elements, converting the lens to be telecentric would require fixing an additional convex lens within the lens casing as mentioned in [41]. If a custom designed wide angle lens were manufactured in a way that their front focal plane resides outside the lens casing then using a lower focal length lens to increase the working distance would be a good

option. Finally, by having a camera CCD array with a larger pixel size allows an increased working distance. From the plots in Figure (4.25) and (4.26) it can inferred that for a 35mm lens with an aperture setting of *f*-number 8 the working distance for a camera with a pixel size of 7.4µm was 120mm, but this was increased to 210mm when a camera with pixel size of 13µm is used. Hence by using a camera with a larger pixel size (approximately twice) the working distance for the given setup can be increased by almost 100mm.

Conclusion

In this chapter a new method was proposed for determining the 7x7 filter coefficients described by Watanabe and Nayar [14]. The procedure was based on two steps: Step1: fitting a linear model to the $\frac{M}{P}$ ratio and Step 2: determining the error between the actual and the linear model and fitting an error correction model. The designed model was verified by comparing it with the theoretical $\frac{M}{P}$ ratio, and it was observed that the filters determined using the new model fitted the $\frac{M}{P}$ ratio more closely than the filters designed by Watanabe's model. The designed filters were tested with real checkerboard images and compared with Watanabe's filters. The maximum RMS error for the defocus condition $\frac{e}{Fe} = 2.307 pixels$ over the working range of 107mm was 1.349% for the filters designed by the Two Step Polynomial Approach and 1.53% for the filters designed by Watanabe and Nayar. Later, filter coefficients were designed for different setup conditions and useful suggestions were provided about the choice of aperture, focal length, and CCD sensor size that would enable a good depth recovery over a wide working distance.

Thus it can be concluded that the filters designed by the Two Step Polynomial Approach:-

- Provide a better fit to the theoretical $\frac{M}{P}$ ratio since they are directly derived from the 2D discrete $\frac{M}{P}$ ratio space.
- The depth estimated using the filters provides a smooth and flat depth map thereby increasing the depth accuracy.
- And finally the method based on the Two Step Polynomial Approach is quite simple as filter coefficients for different defocus conditions can be derived by just modelling the psf.

CHAPTER 5

FPGA Implementation of the Depth from Defocus Algorithm

Introduction

In this chapter a novel design procedure has been described to implement the proposed DFD algorithm on a Field Programmable Gate Array (FPGA). The FPGA is a semiconductor device containing an array of logic blocks that can be configured as per the designer's requirement. They are broadly classified as Fine Grained and Coarse Grained. Fine Grained are made up of small gates, transistors, or small macro cells, while coarse grained are made up of bigger macro cells which contain flip-flops, and look up tables (LUT), which constitute the combinatorial logic function. Since their introduction in 1985, FPGAs have become increasing important to the electronics industry. They have the potential for higher performance and lower power consumption than microprocessors. When compared with Application Specific Integrated Circuits (ASICs), they offer lower non-recurrent engineering costs, reduced development time, easier debugging, and reduced risks [89].

Here the FPGA considered for implementation belongs to the Xilinx Virtex 2ProX family of devices. The architecture of the Virtex 2P device, the Xilinx University program board (XUP), and the programming techniques considered for implementation are discussed in Section 5.1. In Section 5.2, a design procedure, referred to as the Triangular method is employed to perform the 2D convolutions by exploiting the symmetry of the designed filters. Section 5.3 describes the implementation architecture of the DFD algorithm on the FPGA and Section 5.4 provides a detailed analysis of the test pattern and the bit-widths considered for the design model. Finally, Sections 5.5 and 5.6 provide the experimental results of the depth recovered by the designed model that has been implemented on the FPGA. Further, the Sections also provide a detailed comparison between the depth maps recovered using a desktop PC employing Matlab, and the depth maps recovered using a fixed width pipelined processor implemented on FPGA. The results prove that the processor can indeed generate depth maps comparable to Matlab's output.

5.1. Architecture overview of the Virtex 2ProX device

The Virtex-2ProX family of devices are user-programmable gate arrays with various configurable elements and embedded blocks, optimized for high-density and high-performance system designs [90]. The architecture of the device is shown in Figure (5.1) and it includes:- (1) Embedded IBM PowerPC 405 RISC processor blocks that can be clocked up to 400 MHz; (2) Configurable Logic Blocks (CLB) that provide functional elements for combinatorial or synchronous logic implementation; (3) Programmable Input Output Block (IOB) (Ultra-Select IO) that provide high speed interfaces between the FPGA pins and the internal configurable logic; (4) Block Select RAM (Block RAM) provides pre-defined memory blocks that can be as large as 18Kb (Kilo bits). They can be configured either as Single Ports or Dual Ports Memory modules; (5) Dedicated Embedded Multiplier blocks of width 18 x 18 bits; (6) Digital Clock Manager (DCM) provides support for clock distribution, delay compensation, clock multiplication and division; and (7) An Embedded High Speed Serial Trans-receiver (Rocket IO) provides Giga bit transfer rates.

Several development boards incorporating Virtex 2P FPGA devices and peripherals were available for use in the project. The XUP board (Xilinx University program) was considered the most suitable due to its ease of use and connectivity. The features available on the XUP board are explained in the next Section.

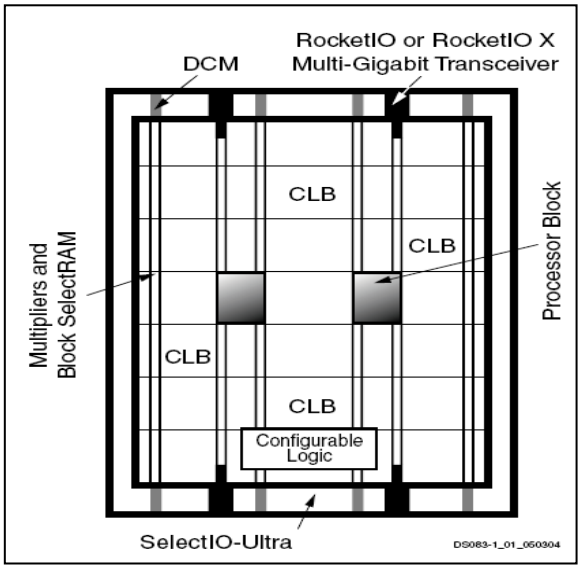


Figure 5.1: Architecture of Virtex 2PX device [90]

5.1.1. Xilinx University Program Virtex 2Pro (XUP 2VP) Development Board [91]

The XUP-2VP development board produced by Digilent Inc was used to implement the DFD algorithm on the FPGA. The board employs a Virtex-2P XC2VP30 FPGA with 30,816 Logic Cells, 136 18-bit multipliers, 2,448Kb of block RAM, and two 405 PowerPC Processors. It has slots for DDR2 SDRAM (double-data-rate synchronous dynamic random access memory) and a Compact Flash Card. It can be interfaced to an external device either through RS232, SATA, 10/100 Ethernet or using USB 2. The board also provides support for Two 2x20 right-angle female sockets, a 100-pin Hirose FX2 connector, Audio in/out, VGA and PS/2 connectors. There are six clock sources, a 100MHz system clock, 75MHz clock for Serial Advanced Technology Attachments (SATA), a 32MHz clock for System ACE interfaces, a dual footprint through-hole for user supplied alternate clock, an external clock for Giga-byte transceivers, and a high speed clock for an expansion module. Figure (5.2) shows a picture of the board as displayed on the Digilent website.

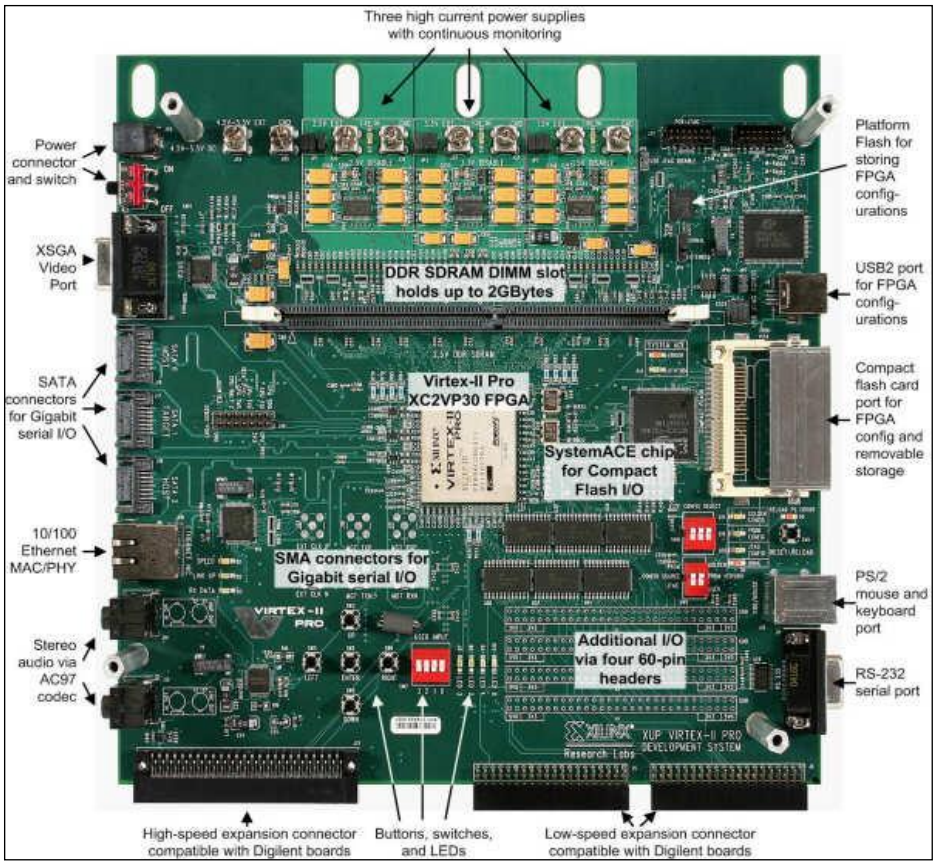


Figure 5.2: XUP 2VP Development Broad [91]

Though the board provides support for various peripherals, the manufacturer has not provided many of the driver files required to employ them in the design module, hence for this implementation, an external SRAM (Static RAM) was used to store the defocused images and the PowerPC was used as an interface between the User-IP (custom designed DFD application) and the desktop PC. It needs to be mentioned that the operation of the User-IP was independent of the PowerPC and also controlled by the common system clock.

5.1.2. Block diagram illustration of the internal architecture of XUP 2P board

Figure (5.3) shows the internal block diagram of XUP 2VP board as per the EDK 10.1 (Embedded Development Kit provided by Xilinx) architecture.

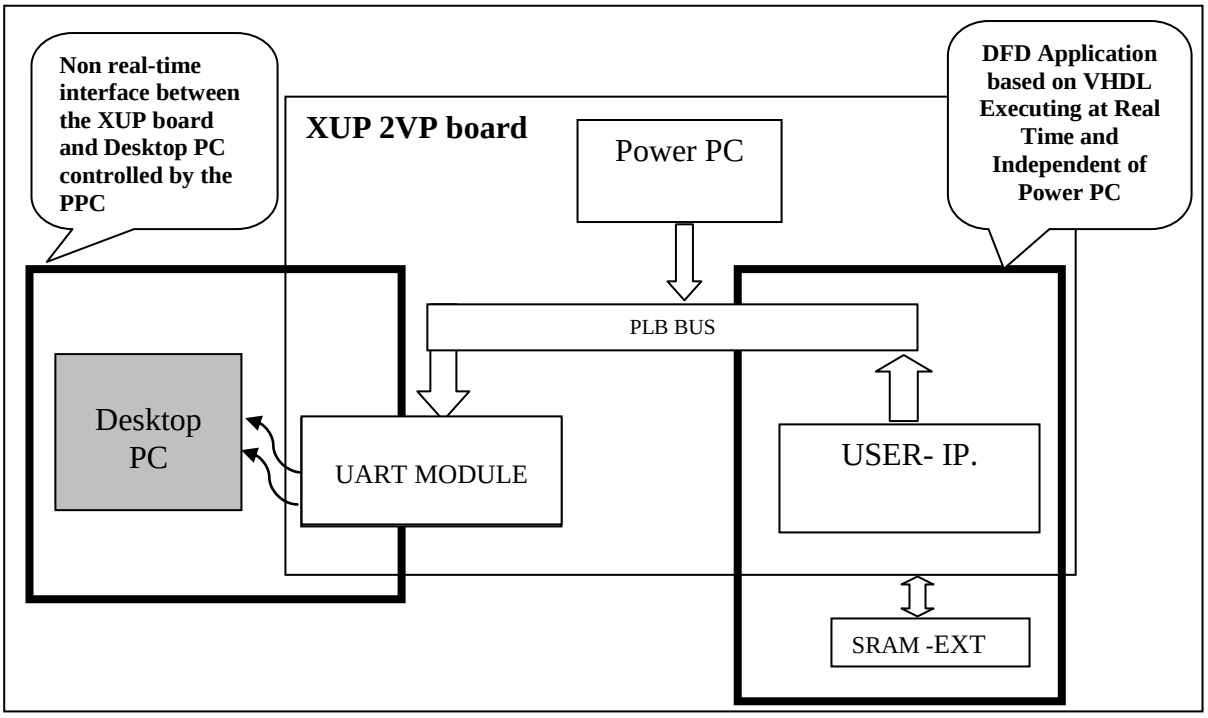


Figure 5.3: Block Diagram - Internal architecture of XUP 2VP board

From the diagram it can be seen that the PowerPC (PPC) controls the peripherals and as per the design requirement, the peripherals are connected to the PowerPC through the Processor Local Bus (PLB). The Base System Builder (BSB) tool provided by the EDK was used to add the required peripherals to the design module. Since the project employed the PPC module as an interface between the desktop PC and the

User-IP, initially only the inbuilt UART module was selected through the BSB tool. Later, the User-IP that defined the custom designed DFD application was added to the PLB bus. The high video rate of the DFD calculation meant that the User-IP had to execute independently of the PPC. I.e. the application was controlled by the system clock rather than by the PPC. The input data from the external SRAM connected synchronously into the Pipelined Architecture (see Figure (5.8)), where the filter convolution operations were performed based on the Triangular method explained in Section 5.2. The output (recovered depth per pixel) was then stored in the inbuilt RAM module provided on the chip, and then transferred to the desktop PC through the UART interface. It should be noted that the User-IP incorporating the DFD program executes at a video rate (at least 25fps), but the interface between the XUP board and desktop PC is non real-time. Further, investigation is underway to employ a Data Acquisition board (DAQ) to capture images directly from a camera system, process them and finally display the depth map on a TV monitor at the video rate. The next Section provides a brief discussion about the programming techniques considered to implement the DFD algorithm.

5.1.3. Programming Techniques

In this Section two different programming techniques are discussed that exploit the internal architecture of the Virtex 2P FPGA device, and that enable parallel execution of the DFD application. The first approach discussed was a ‘C’ based Multi-threaded architecture incorporating the Xilinx Xilkernel and the Power PC (PPC), and the second was based on the Hardware Descriptive Language, VHDL.

Xilinx Xilkernel is a software based embedded processor kernel provided by Xilinx EDK that can be customised for the design requirements. The kernel provides features like multi-tasking, priority-driven pre-emptive scheduling, inter-process communication, synchronization facilities, and interrupt handling. Additionally, a large collection of standard ‘C’ based libraries are available for programming purposes along with functions to accesses the peripherals. For the proposed DFD application, to estimate depth output at each pixel, the algorithm required five 2D convolutions to be performed in parallel. Hence a multithreaded architecture, based on Xilkernel can be employed to compute the convolutions. However, since the execution time for each thread primarily depended on the scheduling capability of

the Xilkernel, which in-turn depended on the overheads present on the PPC, elaborate programming techniques were required to achieve complete synchronisation between the threads. Hence a processor independent language was chosen to program the DFD application.

VHDL (Very High Speed Integrated Circuits Hardware Descriptive Language) is a commonly used language for FPGA implementation. It has constructs to handle parallelism inherent in fast hardware designs and can implement synthesizable logic functions without the intervention of a programmable microprocessor. VHDL can be used to describe an electronic device at different levels of abstraction. The Behavioural level represents the working model of the device without any details about the clock and the delays present within the logic gates. The RTL (Register Transfer level) has an explicit clock and the designed module operates based on the clock cycle but with no detailed delay analysis below the clock cycle provided. The Gate Level description provides a network of gates and registers that constitute the designed module and provides information relating to the actual delays associated with each logic element. To implement a hardware design, the programmer has to describe the designed module in the behavioural abstraction level, and then the synthesis tool generates the netlists (network of gates and registers) that implement the functionality of the described model. The project employed the Xilinx ISE 10.1 design suite to synthesise the netlist targeted for the XC2VP30 Virtex 2P device, and later, the generated netlist was added as a module (User-IP) to the PowerPC using the EDK provided by Xilinx. Finally, using the dedicated software, the generated bit-stream was downloaded to the FPGA.

5.2. 2D Convolution that exploits the symmetry of the designed filters

Depth recovery based on the proposed method [14] required five 2D filtering operations to be performed in parallel, and these filtering operations were performed in the spatial domain using 2D convolution. The filter coefficients (kernel) were rotated by 180 degrees (flipped) and placed over a small image sub-block and the convolution output was calculated using the 2D convolution equation,

$$y(m,n) = x(m,n) \otimes h(m,n) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x(i,j)h(m-i,n-j) \text{ --- (5.1)}$$

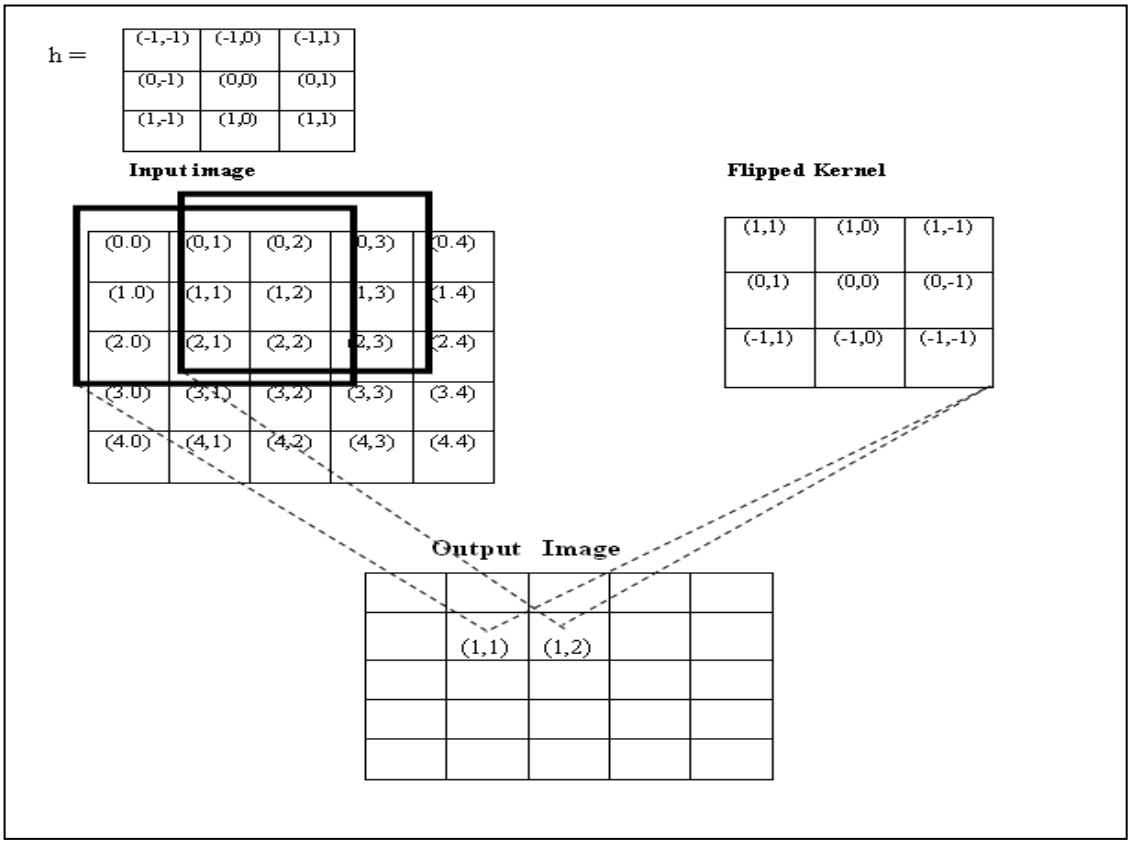


Figure 5.4: Example of 2D Convolution Operation

To understand the convolution operation, an example has been provided. Consider a 7x7 image sub-block that needed to be convolved with the 3x3 kernel as shown in Figure (5.4). To compute the convolution output at the image coordinates *Inputimage(1,1)*, the kernel, *h(i,j)* was flipped and placed over the 3x3 image sub-block keeping the centre pixel of the image sub-block aligned with the centre coefficient of the kernel, *h(0,0)* as shown in Figure (5.4). The convolution output, *y(1,1)* was calculated from

$$y(1,1) = \sum_{j=-\infty}^{\infty} \sum_{i=-\infty}^{\infty} x(i,j)h(1-i,1-j) \text{ --- (5.2)}$$

Expanding,

$$\begin{aligned} y(1,1) &= x(0,0)h(1,1) + x(1,0)h(0,1) + x(2,0)h(-1,1) \\ &+ x(0,1)h(1,0) + x(1,1)h(0,0) + x(2,1)h(-1,0) \\ &+ x(0,2)h(1,-1) + x(1,2)h(0,-1) + x(2,2)h(-1,-1) \end{aligned}$$

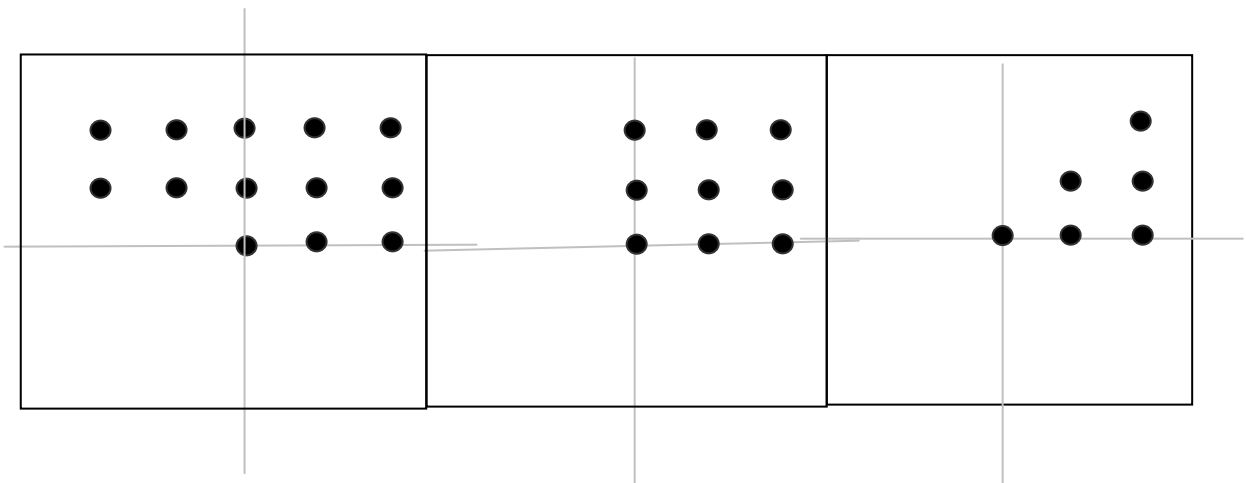
To compute the convolution result for the next pixel, the kernel window was slid over the next 3x3 image sub-block and the output was calculated by replacing the corresponding image coordinates in the above equation. Hence to determine the convolution output for the entire image, the kernel must be slid over each pixel of the image and the convolution result has to be computed based on equation (5.2). In terms of hardware realisation, each convolution output based on a 3x3 kernel required 9 multipliers and 8 adders, and the computation process can be demanding if the kernel size is large. For the 7x7 kernel used in the proposed method for depth estimation (refer to chapter 4), each convolution output required 49 multipliers and 48 adders, and to implement five 2D convolutions in parallel, the selected hardware must have enough logic support to accommodate 245 multipliers and 240 adders. However, the idea was to implement the DFD algorithm on a Virtex 2P FPGA, where only 136 inbuilt dedicated multipliers are available, different methods were investigated to reduce the required number of multipliers. If the designed filters are separable, the multipliers can be reduced by considering the 2D separable convolution equations as in [51], but here the designed filters were rotationally symmetric [52] rather than separable. Hence a design procedure was devised to exploit the symmetry of the filter coefficients and to implement the convolution operations with a reduced number of multipliers. The method referred to as the Triangular method is explained in the next Section.

5.2.1. Triangular Method

The objective was to reduce the number of multipliers required for the convolution operation by exploiting the symmetry of the designed filters. The definition of zero phase filters is given by J.S.Lim in [52]. A digital filter $h(n_1,n_2)$ is said to be zero phase if the frequency response $H(w_1,w_2)$ is a real function such that $H(w_1,w_2)=H^*(w_1,w_2)$ where * refers to the conjugate. A zero phase filter, $h(n_1,n_2)$, is symmetric with respect to a line through the origin and approximately half of the filter coefficients are independent. This symmetry is referred as the two fold symmetry. The filter coefficients: $h(n_1,n_2)= h(-n_1,-n_2)$ (see Figure (5.5a)) and the frequency response is given by $H(w_1,w_2)= H(-w_1,-w_2)$. The number of independent filter coefficients that provide the desired frequency response depend on the symmetry of the designed filter. If the filter possesses four-fold symmetry, then

rotational symmetry can be achieved by considering the coefficients in a single quadrant. Given the coefficients of the first quadrant (Figure (5.5b)) the rotationally symmetric four-fold filter has its coefficients arranged as $h(n_1,n_2) = h(-n_1,n_2) = h(n_1,-n_2) = h(-n_1,-n_2)$ and the frequency response: $H(w_1,w_2) = H(-w_1,w_2) = H(w_1,-w_2) = H(-w_1,-w_2)$. For example, a 5x5 rotationally symmetric filter, with four fold symmetry requires only 9 independent coefficients to provide the desired frequency response and thus provides a reduction in the number of arithmetic operation required for implementation. Similarly for an eight fold rotationally symmetric filter, the filter structure represents a four-fold symmetry about the origin and two-fold refection symmetry every 45 degrees. For the first quadrant the coefficients are arranged as $h(n_1,n_2) = h(n_2,n_1)$ and the response is given by $H(w_1,w_2) = H(w_2,w_1)$. Hence a 5x5 rotationally symmetric filter with eight-fold symmetry would only require 6 independent coefficients to provide the desired response. The filter coefficients are arranged as shown in Figure (5.5c). The 7x7 kernels designed by the Two Step Polynomial Approach are Zero Phase, rotationally symmetric with eight fold symmetry (see Figure (5.6a and 5.6b)), and hence require only 10 independent coefficients to provide the desired response. Therefore a design procedure was employed to compute the 2D convolutions by considering the independent coefficients present on the triangle (see Figure (5.7b)) of the eight fold symmetric filter. The next Section provides a detailed explanation of the Triangular method

.



(a) 5x 5 rotationally symmetric filters with 2 fold symmetry with req. independent coefficients

(b) 5x 5 rotationally symmetric filters with 4 fold symmetry with req. independent coefficients

(c) 5x5 rotationally symmetric filter with 8 fold symmetry and req. independent coefficients

Figure 5.5: Diagram showing the independent coefficients of a 5x5 rotationally symmetric filter

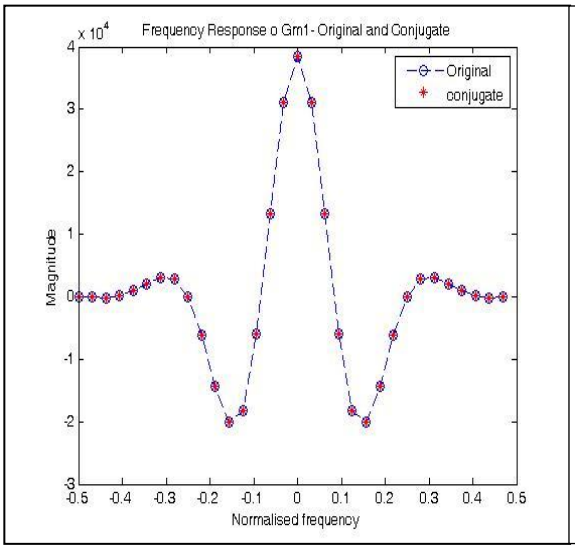


Figure 5.6a: Frequency response Original and Conjugate

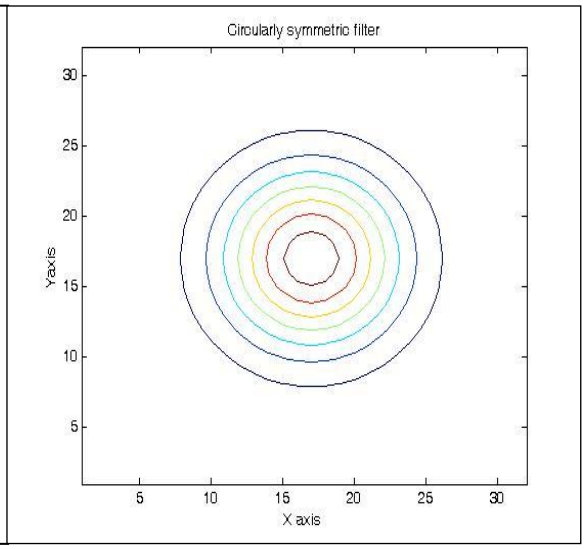


Figure 5.6b: Rotationally Symmetric Low Pass filter

5.2.2. Procedure - 2D Convolution based on the Triangular Method

The convolution process based on the Triangular method is similar to the conventional method except the redundant filter coefficients are arranged to reduce the number of multipliers. As explained earlier, to compute a 2D convolution, the n x n kernel was placed over the n x n image sub-block and the convolution output was calculated based on equation (5.1). In the Triangular method the same procedure was adopted but the convolution operation was rearranged. For illustration purposes consider a 7x7 image sub-block shown in Figure (5.7a) that needs to be convolved with a rotationally symmetric, eight fold symmetric filter as shown in Figure (5.7b).

In the Triangular method, pixels of the sub-block were added wherever possible before being multiplied with the corresponding filter coefficient. In the example considered, the pixel coordinates with the same colour are first added together and then multiplied with the corresponding filter coefficients having the same colour as shown in Figures (5.7a) and (5.7b).

Im1	Im2	Im3	Im4	Im5	Im6	Im7
Im8	Im9	Im10	Im11	Im12	Im13	Im14
Im15	Im16	Im17	Im18	Im19	Im20	Im21
Im22	Im23	Im24	Im25	Im26	Im27	Im28
Im29	Im30	Im31	Im32	Im33	Im34	Im35
Im36	Im37	Im38	Im39	Im40	Im41	Im42
Im43	Im44	Im45	Im46	Im47	Im48	Im49

Figure 5.7a: 7x7 Image sub-block

a	b	c	d	c	b	a
b	e	f	g	f	e	b
c	f	h	i	h	f	c
d	g	i	j	i	g	d
c	f	h	i	h	f	c
b	e	f	g	f	e	b
a	b	c	d	c	b	a

Figure 5.7b: 7x7 rotationally symmetric filter with 8 fold symmetry

Based on the conventional method, the convolution output for the centre pixel Im_{25} can be determined as $C_{25} = Im_1*a+Im_2*b+Im_3*c+Im_4*d+Im_5*c+Im6*b+Im_7*a+\ldots\ldots Im_{49}*a$, but in this Triangular method since the redundant filter coefficients are taken in common and the convolution output was simplified and represented as

$C_{25} = a*(Im_1+Im_7+Im_{43}+Im_{49})+b*(Im_2+Im_6+Im_8+Im_{14}+Im_6+Im_{44}+Im_{48}+Im_{42})+ \ldots + \ldots j*Im_{25}$. Here C_{25} represents the convolution output for the pixel Im_{25} and $a, b, c \ldots j$ represent the filter coefficients as shown in Figure (5.7b). As explained earlier, since the 7x7 kernel designed by the Two Step Polynomial Approach required 10 independent filter coefficients to provide the required response, the convolution procedure based on Triangular method required only 10 multipliers against 49 multipliers for the conventional method. Hence exploiting the symmetry of the designed filter would provide considerable saving in the required hardware. The general equations for implementing 2D convolution based on the Triangular method for 7x7 rotationally symmetric filter with eight fold symmetry are given below

$A= a*(Im_1+Im_7+Im_{43}+Im_{49})$

$B= b*(Im_2+Im_6+Im_8+Im_{14}+Im_{36}+Im_{44}+Im_{48}+Im_{42})$

$C=c*(Im_3+Im_5+Im_{15}+Im_{29}+Im_{45}+Im_{47}+Im_{35}+Im_{21})$

$D=d*(Im_4+Im_{22}+Im_{28}+Im_{46})$

$E=e*(Im_9+Im_{37}+Im_{41}+Im_{13})$

$F=f*(Im_{10}+Im_{16}+Im_{30}+Im_{38}+Im_{40}+Im_{20}+Im_{12}+Im_{34})$

$G=g*(Im_{11}+Im_{23}+Im_{39}+Im_{27})$

$H=h*(Im_{17}+Im_{31}+Im_{33}+Im_{19})$

$I=i*(Im_{18}+Im_{24}+Im_{26}+Im_{32})$

$J=j*Im_{25}$

$Convoutput=A+B+C+D+E+F+G+H+I+J$ --- (5.3)

Here *Convoutput* provides the required convolution output for the sub-block shown in Figure(5.7a) and *A, B. C, D, E, F, G, H, I, J* are the intermediate results that were implemented in parallel on the hardware. The generalised procedure of exploiting the symmetry of the filter coefficients to reduce the multipliers can be extended to two-fold and four-fold rotationally symmetric filters as well. A four-fold symmetric filter required 16 independent coefficients and hence 16 multipliers were needed to provide the required convolution output. Similarly, a two-fold symmetric filter required 24 multipliers and so 24 equations must be implemented in parallel. The chapter proceeds with the next Section where a detailed description about the implementation architecture of the DFD algorithm is discussed.

5.3. Implementation Architecture for the Depth from Defocus Application

This Section describes the implementation procedure of the DFD calculation using a Virtex 2P FPGA. It is based on the proposed algorithm given in [14]. The far and the near-focused images are added, subtracted and then convolved with the pre-filter to remove DC as well as high frequency components. The low pass filter gm_1 was convolved with the subtracted image and at the same time, the LOG filter gp_1 and the correction filter gp_2 are convolved with the added image. Later the convolved outputs were smoothed by a local averaging technique and the divider stage provided the required depth. The implementation represented a pipelined architecture with two parallel channels and five different stages. The two parallel channels process the added and the subtracted images, and the five stages are: - addition and subtraction; pre-filtering; rational filtering; smoothing; and divider. Here two depth outputs (Linear and Error corrected models) are shown for experimental reasons but in practice a look-up table would be employed to provide the depth estimates. The pictorial representation of the DFD algorithm is shown in Figure (5.8).

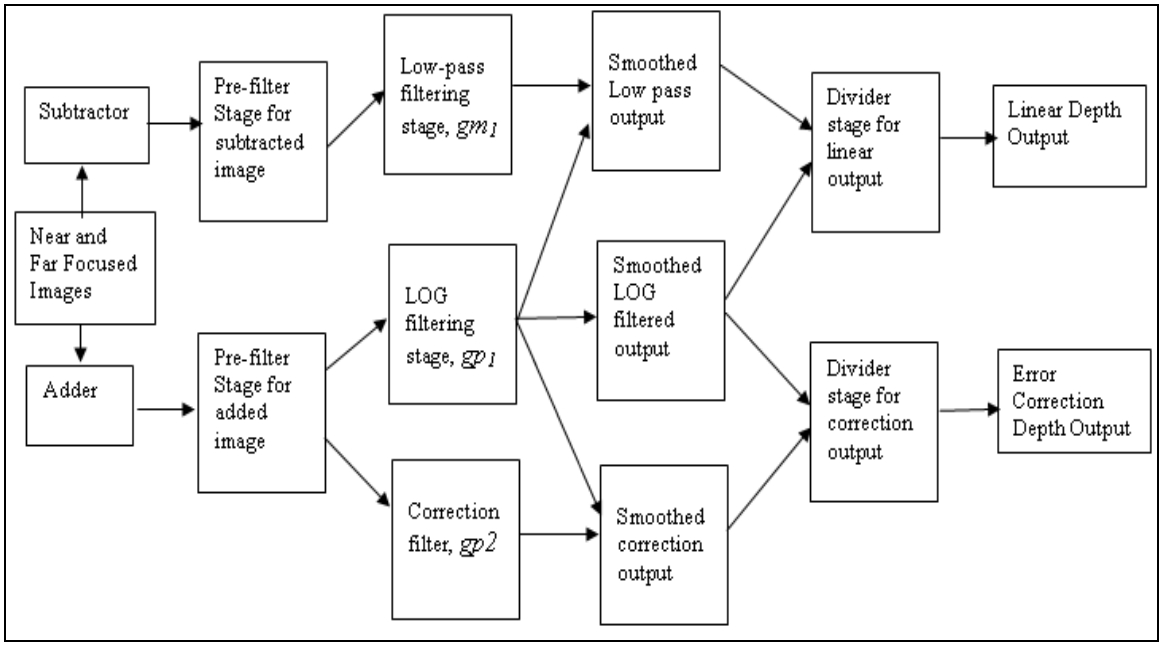


Figure 5.8: Two Channel five stage pipelined architecture

The processing elements (PE) of the pipelined architecture can execute in parallel and the combinatorial logic blocks (adder, subtractor and multiplexers) within the PE are considered as separate components that can execute in parallel, and are synchronous with the system clock. The architecture can be termed as systolic since the input data (D0 to D4) advances into the designed module sequentially, and is controlled by the system clock as illustrated in Figure (5.9). As the input data progresses into each module, the corresponding operations are executed by the processing elements and the final output is obtained in a sequential manner based on the system clock. For every data input, there is a calculated depth output.

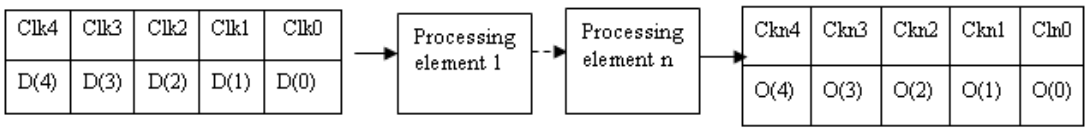


Figure 5.9: Illustration of the Systolic movement of the data

The adder and subtractor stages were implemented using simple logic gates. Subsequently, the added and subtracted data proceeds to the pre-filter stage where the filter module was implemented using multiplier, adders, and shift registers to suit the design requirements. Since the architecture of the pre-filter and rational filter stages have similar structure but with different filter coefficients, a generalised architecture is presented to illustrate the filtering process. For simplicity, only a single processing element (PE) representing the filtering module is explained. It should be noted that the actual design incorporates 5 PEs to compute the five 2D convolution operations corresponding to each stage of the pipelined architecture. The filter module shown in Figure (5.10) consisted of 49 shift registers (SR), 6 RAM based FIFO blocks (first in and first output), 10 multipliers and 48 adders. The bit-width of each module depended on the required accuracy and the available logic. More details about bit-width selection are provided in Section 5.4.

The shift registers were implemented using flip-flops and were arranged to form a 2D array structure with 7 rows and 7 shift register blocks per row. The output of the 7th shift register (SR17) in the first row was connected to the input of FIFO 1, where it was delayed for the completion of the image row. The output from the FIFO 1 was then looped to the input of the shift register (SR21) in the next row. Likewise, the outputs of the 7th shift register in each row were connected to the FIFO in the same row and the FIFO outputs are connected to the shift registers in the next row. This arrangement incorporating the shift registers and FIFO was a systolic array architecture, and the movement of the input data through the design module (shift registers and the FIFO) was synchronised to the common clock. The array when implemented on hardware stored a 7x7 sub-image that when multiplied by the pre-stored coefficients and summed, provided the filtered output. The latency at each filtering stage depended on: - (1) The kernel size; (2) The horizontal resolution of the image; and (3) Any internal buffering present within the PE. Here, filtering operations were performed on test images of resolution of 400 x 400 pixels using a 7x7 kernel and each PE required an internal signal buffering that corresponded to 3 clock cycles. Hence the latency for a filtering process was 1207 clock cycles. A Table illustrating the latency present at each stage of the pipelined processor is provided in Section 5.5.

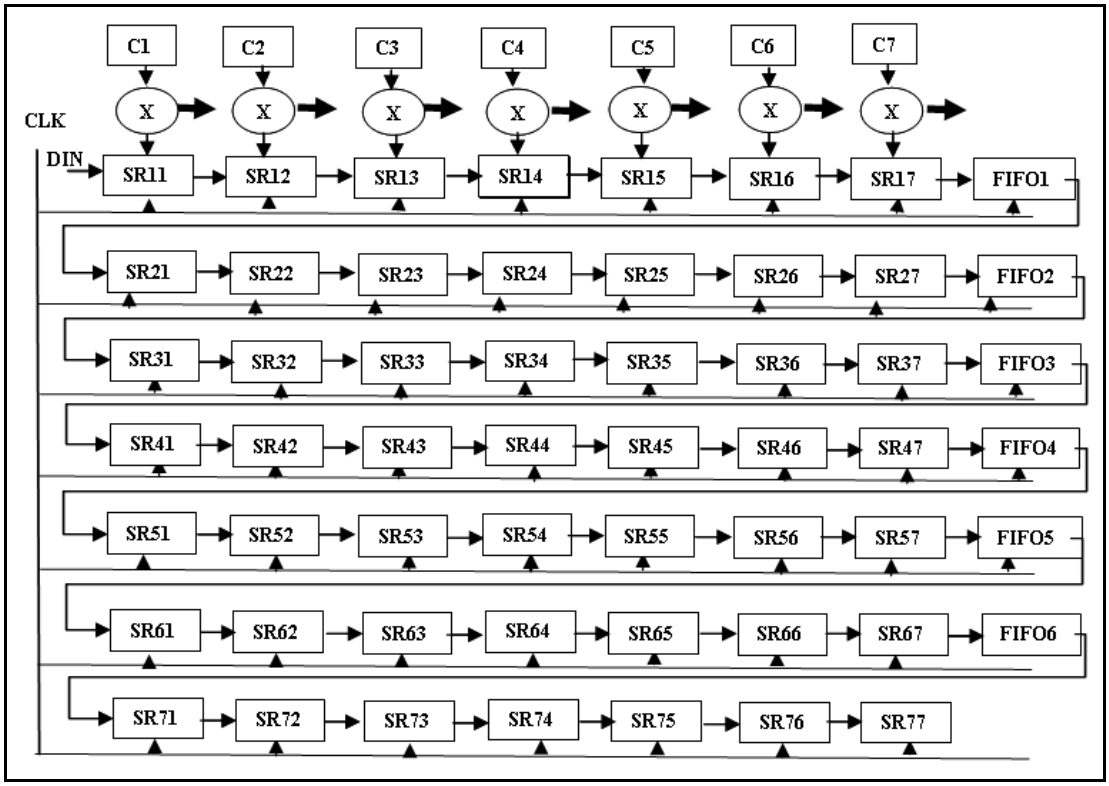


Figure 5.10: Filter block module with Shift registers and FIFOs

The pre-filter output then progressed into the rational filtering stage were the design architecture remained the same, but different filter coefficients were used. After the rational filtering stage, the filtered pixels advanced into the smoothing stage. The smoothing stage provided the required smoothing operation based on local averaging, and was implemented using a 5x5 systolic array incorporating shift registers and FIFOs. Finally the smoothed data advanced into the divider stage, the output of which provided the required depth estimate. The depth output from the divider was stored on an inbuilt dual port RAM and then transferred to the desktop PC through the UART interface. The next Section provides a detailed analysis of the test pattern and the required bit-widths at each stage of the pipelined DFD calculation.

5.4. Analysis – Test pattern and Computation of bit-widths at each stage of the Processor Module

The chip area of the logic for the circuit required for the design module depended on the bit-width requirement at each stage of the pipelined architecture. Larger bit-widths provided an increase in accuracy of the depth estimation, but resulted in an overall increase in the number of logic circuits required and involved complex signal routing schemes, and longer delays through the critical data path. Efforts were taken to reduce the amount of logic required at each stage of the pipelined process thereby providing simple routing schemes with reduced delay. To calculate the optimum number of bits required at each stage, the design model required a test pattern that contained all possible frequencies within the applicable range of the defocus conditions. Here the defocus condition used was $\frac{e}{Fe} = 2.307 pixels$ and the acceptable frequency range lay between $0.2857 \leq f_r \leq 0.3160 pixel^{-1}$ where f_r represented the radial frequency. The test patterns that were considered for simulation were the checkerboard pattern and the pattern devised by Watanabe and Nayar (see Figure (5.13)). The checkerboard patterns used for simulation had wavelength of 8 pixels (4 black and 4 white) and 10 pixels (5 black and 5 white). The power spectral density (PSD) plots for the patterns are shown in Figures (5.11a) and (5.12a). The patterns were treated as wide-sense stationary random processes and the PSDs of the patterns were computed by considering the Fourier Transform of the autocorrelation function. The patterns were defocused with a normalised depth α = 0.5.

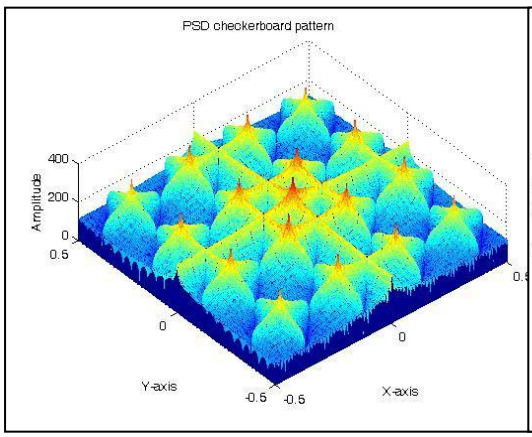


Figure 5.11a: PSD of the checkerboard pattern for wavelength 8

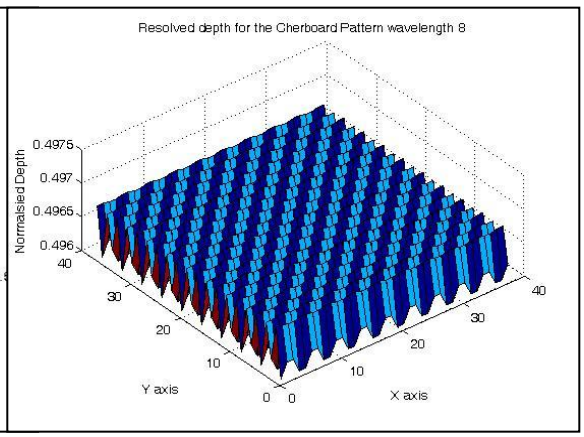
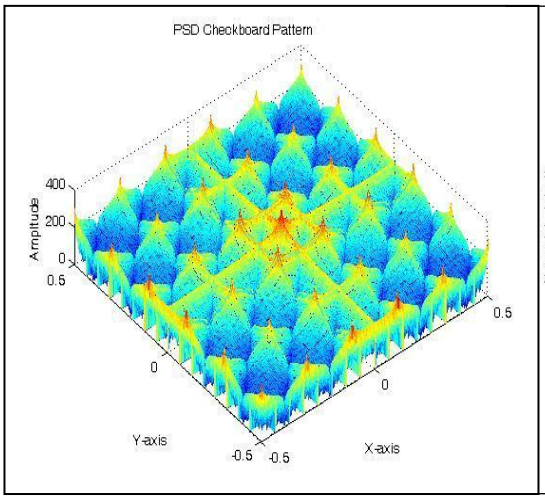


Figure 5.11b: Estimated depth map showing the artefacts



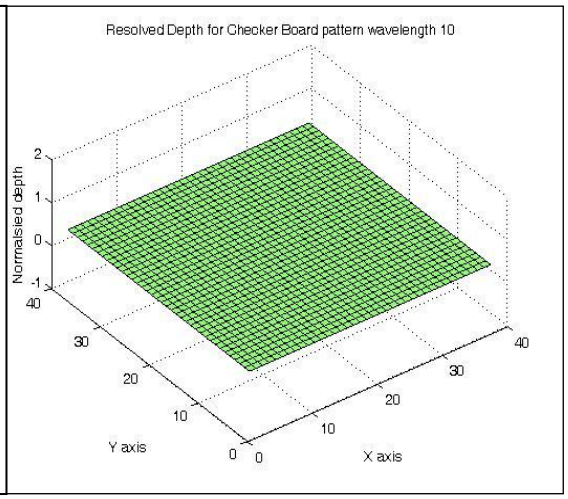


Figure 5.12a: PSD of checkerboard pattern for wavelength 10

Figure 5.12b: Estimated depth map without the artefacts

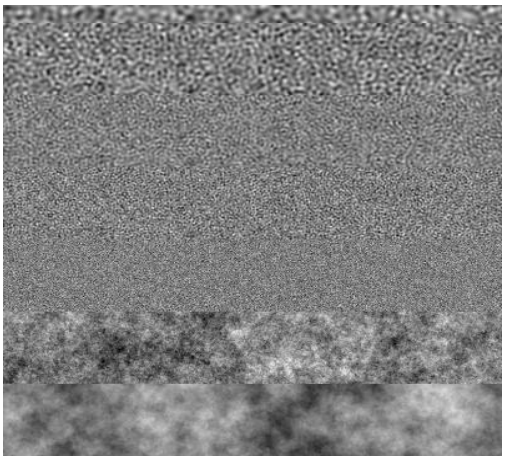
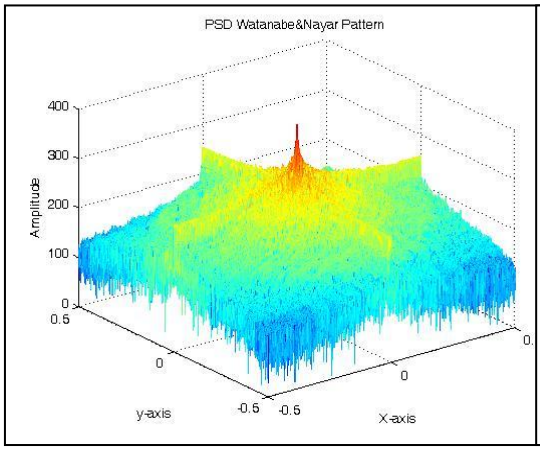


Figure 5.13: Watanabe's pattern



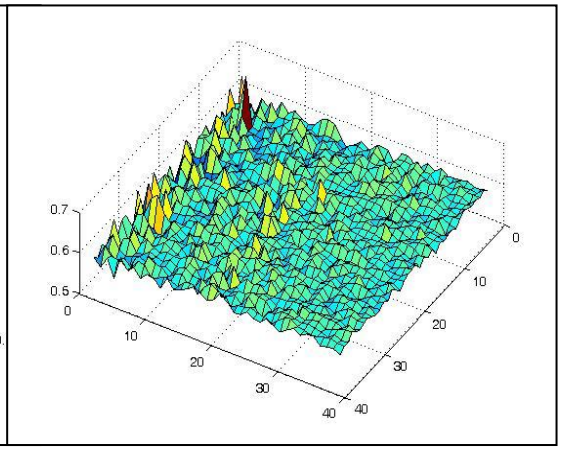


Figure 5.14a: PSD of Watanabe's pattern

Figure 5.14b: Estimated depth map without post-processing

From the PSD plot of the checker-board pattern with wavelength 8 pixels (Figure (5.11a)), it can be inferred that the spectral energy of the pattern was concentrated mostly at first at harmonic ($0.125\,pixel^{-1}$), and gradually reduced at the third ($0.375\,pixel^{-1}$) and the fifth, ($0.625\,pixel^{-1}$). Since the acceptable frequency range for the given defocus condition lies between $0.2857 \le f_r \le 0.3160\,pixel^{-1}$, it can be verified that the given pattern has a low spectral energy within the acceptable range. However, when experiments were carried out with the test pattern, the designed filters were able to recover the depth map quite accurately but with a prominent artefact, that resembled the texture of the pattern (Figure (5.11b)). Further investigation revealed that even though the spectral energy was low within the acceptable range, the edges between the black and the white pixels provided considerable information, and the designed filters were able to recover the depth information using the defocused step edges in the two images. To verify this, a checker-board pattern with a period of 10 pixels was considered. Again the spectral power (see Figure (5.12a)) was concentrated on the odd harmonics but since the 3rd harmonic ($0.3\,pixel^{-1}$) lies within the acceptable range for the defocus condition, the depth recovered by the filters showed no artefacts. The recovered depth was then smooth as shown in Figure (5.12b). Hence the designed filters were able to recover the depth by considering only the third harmonic that was present within the acceptable range. From the above investigation, it should be noted that for a checker-board pattern, the spectral information is only non-zero at the edges, or as the single frequency sinusoids corresponding to the odd frequency harmonics. Alternatively, for the pattern devised by Watanabe and Nayar [14], spectral energy was spread over a broad range of frequencies within the acceptable frequency range of the defocus condition (see Figure (5.14a)). Hence this pattern was used as a generalised pattern to determine the bit-widths at each stage and also to provide a useful accuracy comparison between the Matlab output and the FPGA output.

The filter coefficients designed by the Two Step Polynomial Approach were real valued numbers and hence to use them on the available hardware, they needed to be transformed into integers. Since multiplication and division operations can be effectively implemented on the hardware using shift and add operators, the filter coefficients were scaled by a factor of 2^n where n was chosen by comparing the RMS error estimates between the scaled variant of the frequency response of the filter

(computed after convolving with the impulse function) and the 64 bit Matlab output. Figure (5.15) shows the response of the pre-filter for four possible values of *n* and Table 5.1 illustrates the RMS error values. From the RMS error values the scaling factor of the filter coefficients was chosen to be 13 for both pre-filter and the rational filter coefficients.

The generalised block diagram of the pipelined DFD architecture implemented on the FPGA is shown in Figure (5.16). Table 5.2 provides a comparison between four design models in terms of the bit-width requirement at each stage, the scaling factor of the filter coefficients, the percentage of logic required for the implementation of the model, and the RMS error between the 64 bit Matlab depth output and the 10 bit FPGA output. Here, the final output from the pipelined processor was chosen to be 10 bits (9 bit data with 1 sign bit), since the depth maps recovered from the patterns considered for the experiments fall within this range. Moreover, in-terms of accuracy and logic usage, the choice of having a 10 bit output seemed reasonable. Experiments were performed with the pattern shown in Figure (5.13) which was defocused for a normalised depth of 0.99. The variables A, B, C, D, E, F, G and H in Figure (5.16) represented the bit-widths at each stage of the pipelined architecture. From Table 5.2, it can be inferred that Model 3 required less logic support and also provided acceptable depth accuracy when compared with the 64 bit Matlab output; hence bit-widths related to Model 3 were chosen for the implementation.

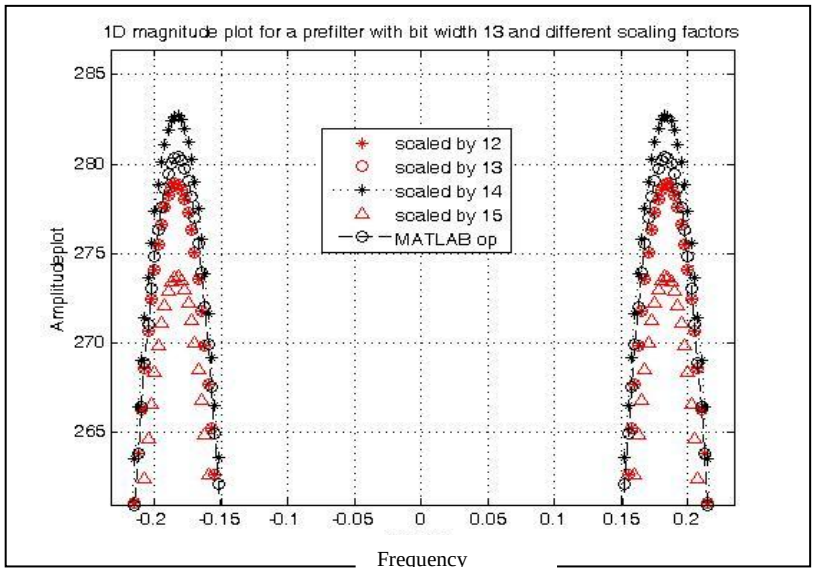


Figure 5.15: Comparison between Matlab frequency response and the scaled frequency response of the pre-filter

Coefficient Scaling, *n*	RMSE for a pre-filter of width 13
2^12	0.69
2^13	0.69
2^14	3.31
2^15	5.69

Table 5.1: RMS error for different scaling factors

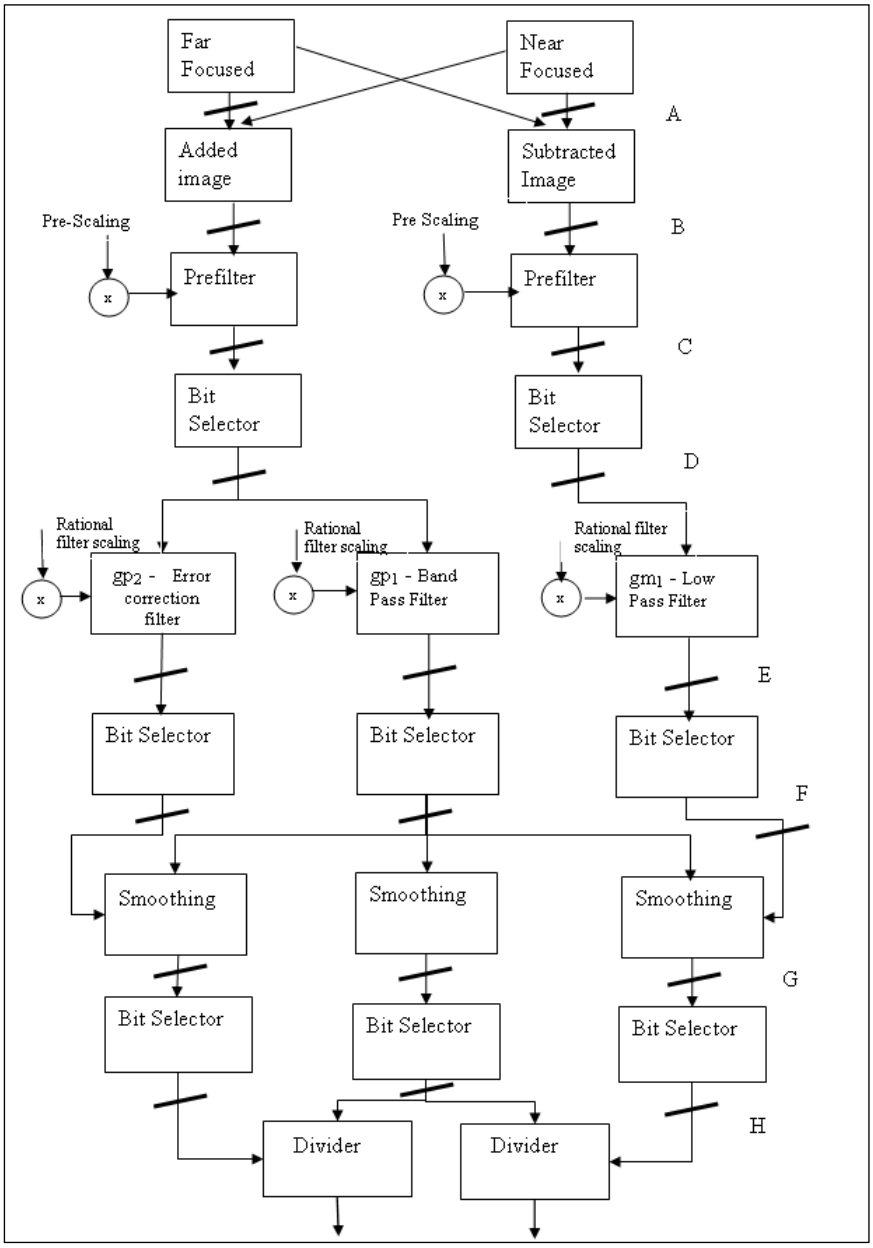


Figure 5.16: Generalised block diagram showing bit-widths at each stages of the pipelined processor

Model	**Coefficient. scaling**	**A bits**	**B bits**	**C bits**	**D bits**	**E bits**	**F bits**	**G bits**	**H bits**	**Chip area required by the logic as a % of chip area**	**RMSE between Matlab and FPGA outputs**
Model 1	Pre-filter - 2^13 Rational Filter – 2^13	8	16	32	16	32	16	32	32	78%	0.9505 For 10 bit output
Model 2	Pre-filter - 2^13 Rational Filter – 2^15	8	10	23	16	32	16	32	24	59%	0.9512 For 10 bit output
Model 3	Pre-filter - 2^13 Rational Filter – 2^13	8	13	26	16	32	16	32	20	50%	0.9505 For 10 bit output
Model 4	Pre-filter - 2^13 Rational Filter – 2^13	8	13	26	16	32	16	32	20	50%	0.9526 For 8 bit output

Table 5.2: Bit-width requirement for the four models considered along with the chip area used, and the RMSE between Matlab and FPGA depth outputs

5.5. Design of Experiment

The proposed DFD algorithm was coded for the Virtex 2P FPGA using the Hardware Descriptive Language, VHDL. The behavioural and the structural models were synthesized using Xilinx ISE 10.1 and the generated net-list was targeted for the XC2VP30 Virtex device. The data clock frequency was chosen as 12.5 MHz since, at that rate, a 400 x 400 pixel resolution image can be processed within 25ms. The clock source also controls the systolic movement of the input data through the pipelined architecture enabling an output every 80ns. To provide parallel and synchronous movement of the input data through the two parallel channels of the designed module, a multiplexer module operating at twice the data clock rate was used to accesses the input data (defocused images) from the external SRAM. In a practical implementation, a two CCD sensor system would be employed to acquire the defocused images and hence would not require any image storage.

Based on the results of the experiments with test patterns (Section 5.4), Model 3 was implemented on the FPGA and the bit-widths provided in Table 5.2 were used. The shift registers were implemented using D flip-flops and FIFOs, and the pipelined divider modules were implemented using the modules provided by the Xilinx ISE. The output from the divider module was stored in the on-chip RAM and then transferred to the desktop PC through the UART interface. The delays present at each stage of the pipelined architecture were estimated using the simulation, and these are listed in Table 5.3. The delays listed contribute to the latency and not to the data throughput. The time taken to process a frame of 400 x 400 pixels was 13.06ms and hence a total of 76.56 frames of size 400 x 400 can be processed in one second and, thus the processor operates at video rate. Though the entire block diagram shown in Figure (5.8) was implemented on the Virtex 2P device, the depth measurements related to the linear depth model (β) results have been displayed for the simulated and real images in the next Section. A combined depth result with linear (β) and error corrected depth outputs (β^3) would require the usage of a pre-computed lookup table, the implementation of which is currently under study.

	Input Stage File /Ram	Pre-filter Stage	Rational Filter Stage	Smoothing Stage	Divider Stage
Delay in clock cycles	4	1207	1207	802	26

Table 5.3: Delays at each stage of the pipelined architecture based on the simulation report

5.6. Experiments with Simulated and Real Images

The Section describes the detailed experimental analyses that were performed to estimate the accuracy of the depth measurements obtained from the pipelined processor described in Section 5.3. Here, for simulated images, the pattern devised by Watanabe [14] (shown in Figure (5.13)) was used as the test pattern. The reason for choosing the test pattern is explained in Section 5.4. For simulated images, the Pill- box psf was used as the defocus operator. Section 5.6.1 illustrates the depth map resolved from the test pattern when defocused for the maximum normalised depth α = 0.99. Section 5.6.2 shows the depth map recovered from the pattern when simulated as a 3D depth staircase structure and Section 5.6.3 illustrates the depth map recovered from a real checkerboard image acquired using the defocused condition $\frac{e}{Fe} = 2.307 pixels$. For the simulated images, four different depth comparison results are provided:- (1) Matlab 64 bit post-filtered depth map which included both linear and error corrected depth estimates; (2) Matlab fixed point (truncated) post-filtered linear depth output where the bit-widths at each filter stage are set according to the FPGA bit-widths (refer to model 3 in Table 5.2); (3) FPGA 10 bit linear depth output without post-filtering; and (4) FPGA 10 bit with post-filtered output. For the real images a comparison between Matlab and FPGA output has been provided. Here a 9x9 median filter was employed as the post-filter. It should be noted that the post-filtering operation has been performed using Matlab.

5.6.1. Result for the Simulated Images defocused for the maximum normalised depth

In this Section, the test pattern was contained in a plane perpendicular to the optical axis and defocused for the maximum normalised depth, α = 0.99. The near and the far-focused images are shown in Figure (5.17). The recovered 64 bit Matlab post-filtered depth map and the Matlab truncated depth output are shown in Figures (5.18a) and (5.18b) respectively. The 10 bit FPGA depth maps with and without the post-filtering operation are shown in Figure (5.19a) and (5.19b). The statistical results for the four different depth estimates are provided in Table 5.4. These results were calculated from a local area of 38x38 pixels obtained across the depth map.

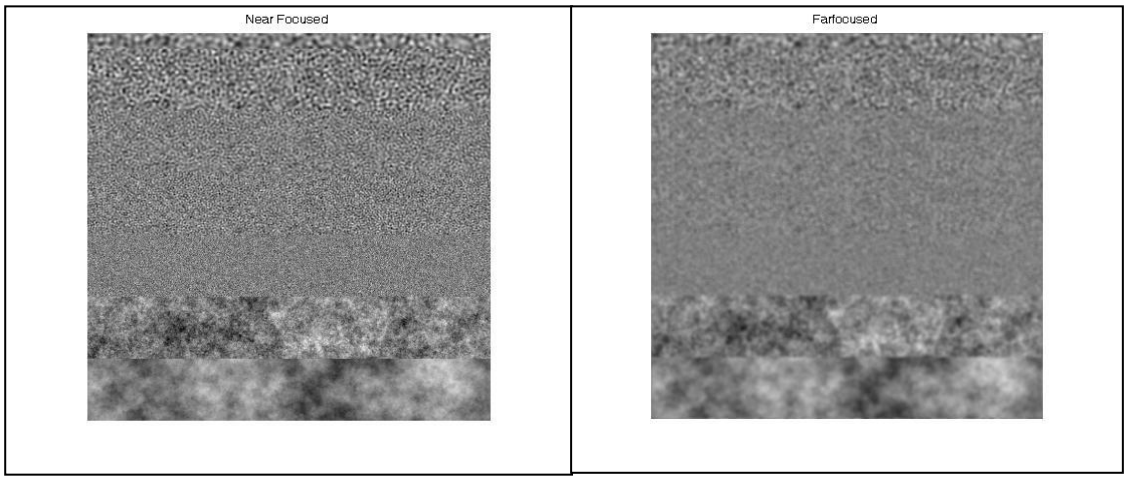
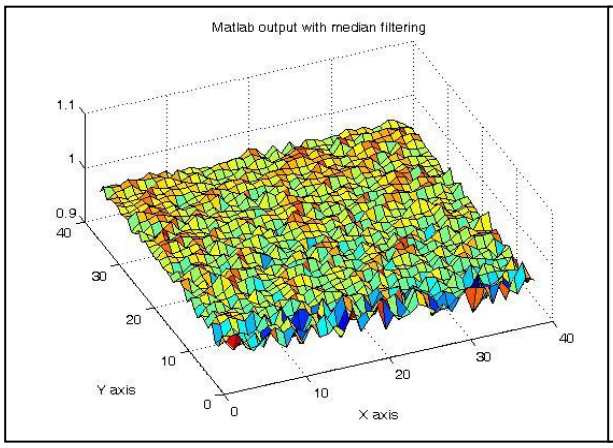


Figure 5.17: Near and far-focused images of the pattern



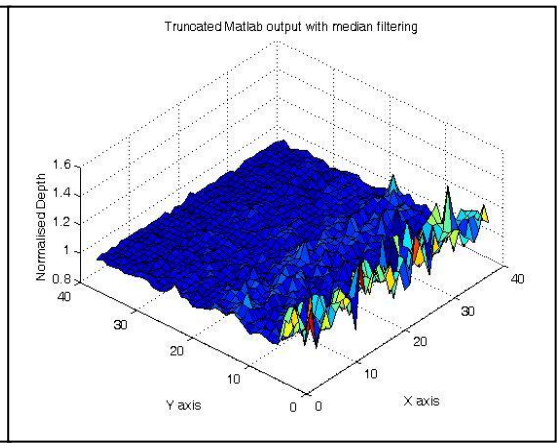


Figure 5.18a: Matlab 64 bit depth output with post-filtering

Figure 5.18b: Matlab truncated output with post-filtering

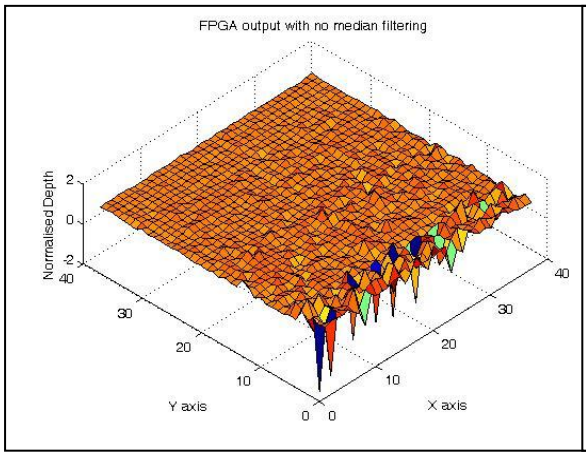


Figure 5.19a: FPGA depth map without post-filtering

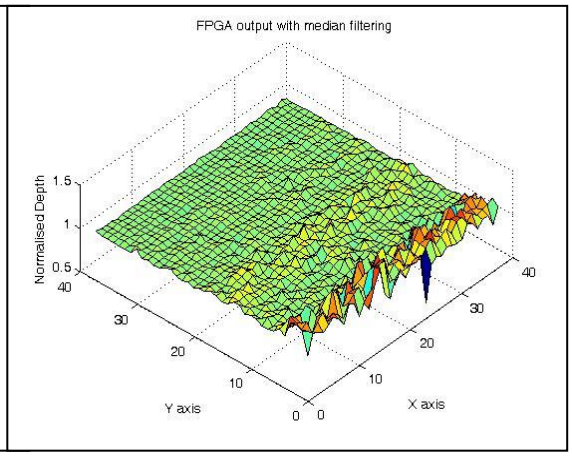


Figure 5.19b: FPGA depth map with post-filtering

Statistical results from a local area of 38x38 pixels.	Matlab 64 bit post-filtered linear and error corrected output	Matlab truncated post-filtered linear depth output	FPGA linear depth output without post-filtering	FPGA linear depth output with post-filtering
Mean	0.9743	1.0244	1.011	1.019
Std. Deviation	0.0087	0.0706	0.238	0.0606
Variance	7.5e-3	0.005	0.0566	0.0037

Table 5.4: Comparison between Matlab and FPGA depth outputs

Table 5.4 provides a comparison between the Matlab and the FPGA outputs. It can be inferred that the FPGA provided acceptable depth accuracy but the resolved depth map was not as smooth and flat as the Matlab 64 bit output. The reason can be attributed to:- (1) The FPGA output provides only the linear depth output whereas Matlab 64 bit output provides both linear and error corrected depth result; (2) Rounding errors present at each stage of the pipelined architecture and the scaling factor used to scale the filter coefficients; and (3) The lack of texture at the bottom part test pattern can be related to the unstable depth estimates that are clearly visible in the form of spikes on the recovered depth map. It can be also verified that the post-filtered output based on the median filter provided a smooth depth result. Currently, investigation is underway to implement a median filter on the FPGA.

5.6.2. Experiment with a simulated 3D scene

In this experiment the test pattern was simulated as a 3D staircase structure with four different normalised depths. Again, the pattern was contained in a plane perpendicular to the optical axis and was defocused using the Pillbox psf. The four normalised depths used were 0.2, 0.5, 0.8 and 0.99. Here 0.2 represented the nearest depth and 0.99 the furthest. The near and far-focused images are shown in Figure (5.20). It should be noted that the test pattern was defocused such that the well textured top part represented the nearest depth. This was done to validate the accuracy of the depth estimates for the poor textures placed further away from the camera. The recovered 64 bit Matlab depth output and Matlab truncated output are shown in Figures (5.21a) and (5.21b) respectively. The FPGA outputs with and without the post-filtering process are shown in Figures (5.22a) and (5.22b). The grey scale depth outputs of Matlab and FPGA are shown in Figure (5.23). The statistical results obtained from a local area of 61x 361 pixels are provided in Table 5.5.

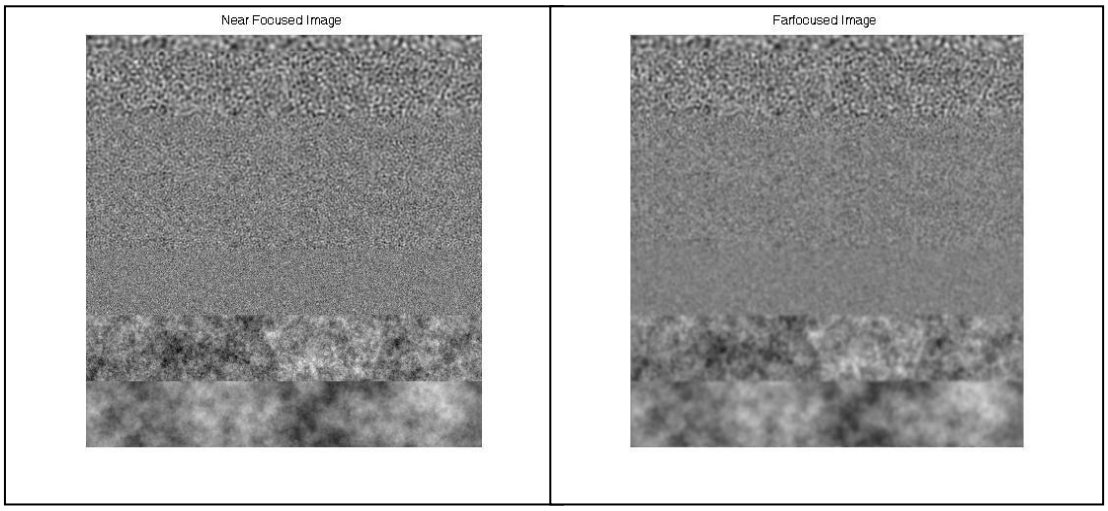


Figure 5.20: Near and far-focused images

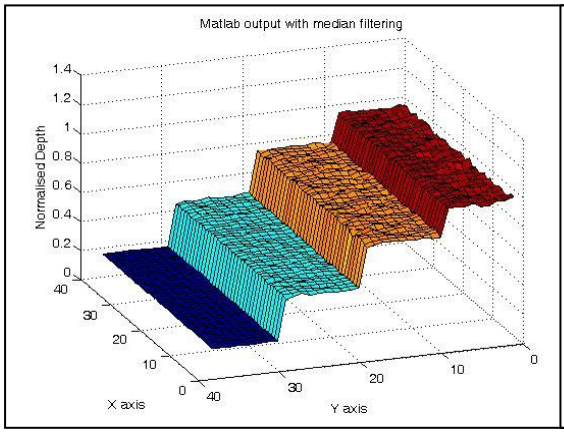


Figure 5.21a: Matlab 64 bit depth map with post-filtering

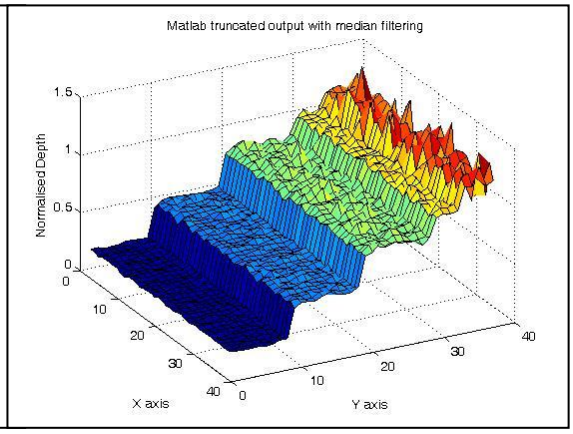


Figure 5.21b: Matlab truncated depth map with post-filtering

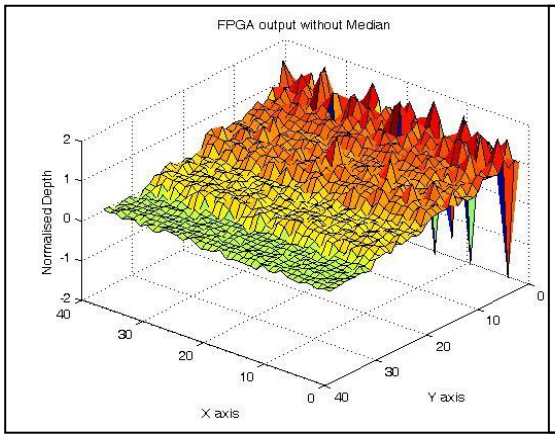


Figure 5.22a: FPGA depth map without post filtering

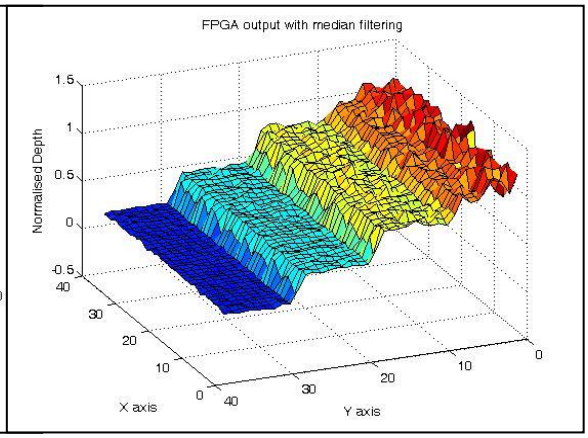


Figure 5.22b: Depth map with post filtering

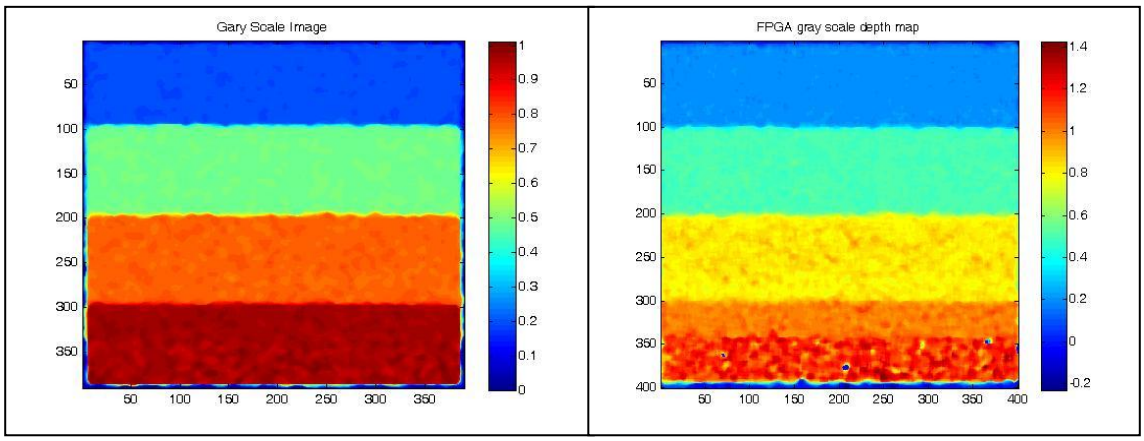


Figure 5.23: Gray scale post-filtered depth map estimated from Matlab (left) and FPGA (right)

Depth estimated from		Depth 1 = 0.2	Depth 2 = 0.5	Depth 3 = 0.8	Depth 4 = 0.99
Matlab 64 bit post-filtered linear and error corrected output	Mean	0.1931	0.4832	0.7830	0.9691
	Std.Dev.	0.0025	0.0048	0.0073	0.0118
	Var.	6.14e-5	2.32e-5	5.35e-5	1.4e-4
Matlab truncated post-filtered linear depth output	Mean	0.194	0.4979	0.83	1.09
	Std.Dev.	0.007	0.0115	0.027	0.102
	Var.	4.4e-5	1.3e-4	7.3e-4	0.0105
FPGA linear depth output without post-filtering	Mean	0.1936	0.4983	0.8402	1.0146
	Std.Dev.	0.0326	0.0542	0.1064	0.446
	Var.	0.011	0.0029	0.0113	0.1994
FPGA linear depth output with post-filtering	Mean	0.1941	0.4976	0.8305	1.06
	Std.Dev.	0.0073	0.0114	0.0265	0.09
	Var.	5.29e-5	1.29e-4	6.97e-4	0.085

Table 5.5: Comparison between Matlab and FPGA depth outputs

From the above Table, it can be inferred that for the normalised depths 0.2 and 0.5 the recovered depth map from the processor was comparable to Matlab's 64 bit output, but for the depths at 0.8 and 0.99, the depth maps were quite noisy. As explained earlier, since the FPGA output provides only the linear depth estimates that do not accurately map the $\frac{M}{P}$ curves at greater distances, this leads to the less accurate depth measurements. Further, the lack of enough texture at the bottom of the pattern and the errors due to rounding also add to the reduction in the depth accuracy. The post-filtered output clearly shows the 3D stair case structure which is comparable to the Matlab output.

5.6.3. Experiment with a Real Checkerboard Image

A checkerboard pattern was placed at a distance of 770mm from the lens and two defocused images were captured based on the defocused condition $\frac{e}{Fe} = 2.307 pixels$. The near-focused was at 800mm and far-focused at 744mm. The resolved depth

maps for the Matlab 64 bit output and for the FPGA output, with and without post-filtering operations are shown in Figures (5.24) and (5.25) respectively. Table 5.6 provides a comparison between the Matlab and FPGA outputs obtained from a local area of 38 x 38 pixels.

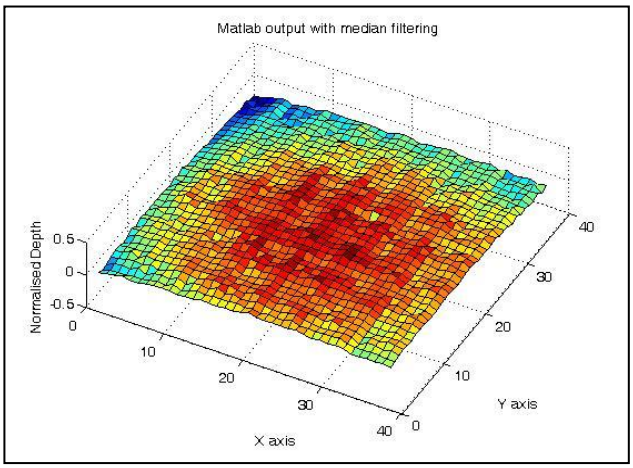


Figure 5.24: Matlab 64 bit post-filtered depth map

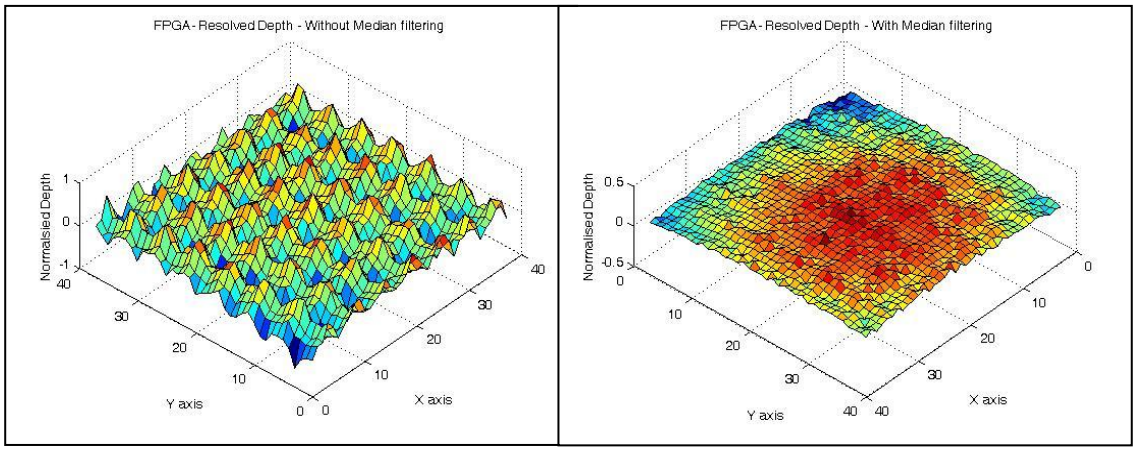


Figure 5.25: FPGA depth map without post-filtering (left) and with post-filtering (right)

Depth output calculated from 38x38 pixels	Matlab 64 bit post-filtered linear and error corrected output	FPGA linear depth output without post-filtering	FPGA linear depth output with post-filtering
Estimated Depth in mm	769.129	768.883	769.095
RMS error in mm	1.97	6.2929	1.969

Table 5.6: Comparison between Matlab and FPGA depth outputs

The depth outputs without median filtering show a prominent artefact similar to the texture of the checkerboard pattern. The reason for these artefacts as explained in Section 5.3 was due to the low spectral power within the acceptable range of the defocus condition, and hence the depth has been recovered by considering the disparity between the edges. When the depth map was post-filtered the artefacts become less visible and provide higher depth accuracy as shown in Table 5.6. It can also be inferred from the Table that there is no significant difference between the FPGA and the Matlab results. More depth estimation results for arbitrary objects with natural textures are presented in the next chapter.

Conclusion

The chapter described a procedure to implement the DFD algorithm on a Virtex 2P FPGA device. The researched software program required five 2D convolutions to be processed in parallel and these convolutions were effectively implemented on the hardware using the Triangular method described in Section 5.2. Four design models were considered for implementation, and the model with acceptable accuracy and minimum logic usage was implemented as described in Sections 5.3 and 5.4. The synthesis report that describes the logic usage in terms of multipliers, RAM blocks, LUT etc. is presented in Appendix 6. The depth estimation results along with the comparison with the 64 bit Matlab outputs are provided in Section 5.6. It can be inferred from the results that the 10 bit FPGA outputs are comparable to the Matlab's 64 bit outputs and that both required post-processing operations to restore the smoothness of the depth estimates. Currently, the hardware implementation of a median filter, optimised for speed is under study [72] [109]. The implemented model (Model 3) used 50% of the available chip logic, and processed a frame of size 400 x 400 pixels in 13.06ms with an acceptable accuracy as presented in Table 5.2. The processing time and the errors due to rounding could be reduced further if more advanced FPGA devices (Virtex 4) were used.

CHAPTER 6

Experimental results with 3D Objects and Natural Textures

Introduction

The chapter provides the depth estimation results and 3D shapes that were recovered using the filters designed by the Two Step Polynomial Approach (refer to chapter 4). It should be noted that a similar experiment was presented in chapter 4, but these results were based on a real checkerboard pattern. Later, in chapter 5, it was observed that for a checkerboard pattern, the spectral information was present only at edges or as single frequency sinusoids. Hence to determine the depth accuracy and to verify the invariance of the filters to different textures, experiments were performed on natural objects with arbitrary textures. All these experiments are based on the defocus condition $\frac{e}{Fe} = 2.307 pixels$. A 50mm photographic quality lens with an external aperture diameter set to 6.5mm was used. The near and the far-focused images were at 800mm and 744mm. Based on Appendix3, the working range was calculated to be 56mm. To increase the depth accuracy a calibration procedure described in Section (6.1) was adopted. The Section also presents depth calculation results of a randomised textured pattern: sand paper, and a comparison between the depth estimates obtained using the filters designed by the Two Step Polynomial Approach and Watanabe's filters. The depth results related to non-planar objects are presented in Section 6.2. For all the experiments, the actual depth measurements and the corresponding measurement errors are also presented. Finally, Section 6.3 illustrates the shape recovered from complex objects incorporating both arbitrary and homogeneous textures. These depth results include a 3x3 Gaussian smoothing operation. A comparison between Matlab depth outputs and FPGA depth outputs is also provided for Sections 6.2 and 6.3.

6.1. Experiment with a random textured natural pattern: Sand Paper

This Section provides depth estimation results of a natural sand paper pattern (Figure (6.1a)) contained in a plane perpendicular to the optical axis. The pattern also served as the reference pattern to calibrate the system since the PSD plot (Figure (6.1b)) showed enough spectral density within the acceptable frequency range of the defocus condition. A simple calibration procedure was adopted to ensure that the estimated normalised depth falls within -1 and +1, where -1 referred to the far-focused image

and the +1 to the near-focused. To achieve this, the pattern was first placed at the farthest point and two images were captured: - (1) A focused image, corresponding to the furthest distance, 800mm; and (2) A defocused image, corresponding to the nearest distance, 744mm. The two images were processed and their depth estimates were analysed. If the normalised depth was -1, for image (1), then the calibration was perfect, else small adjustments are required. A similar experiment was carried out with image (2). In practice this one time calibration adjustment provided an increase in the accuracy of the depth measurements. It should be noted that for experiments in Sections 6.2 and 6.3, the sand pattern was used as the background.

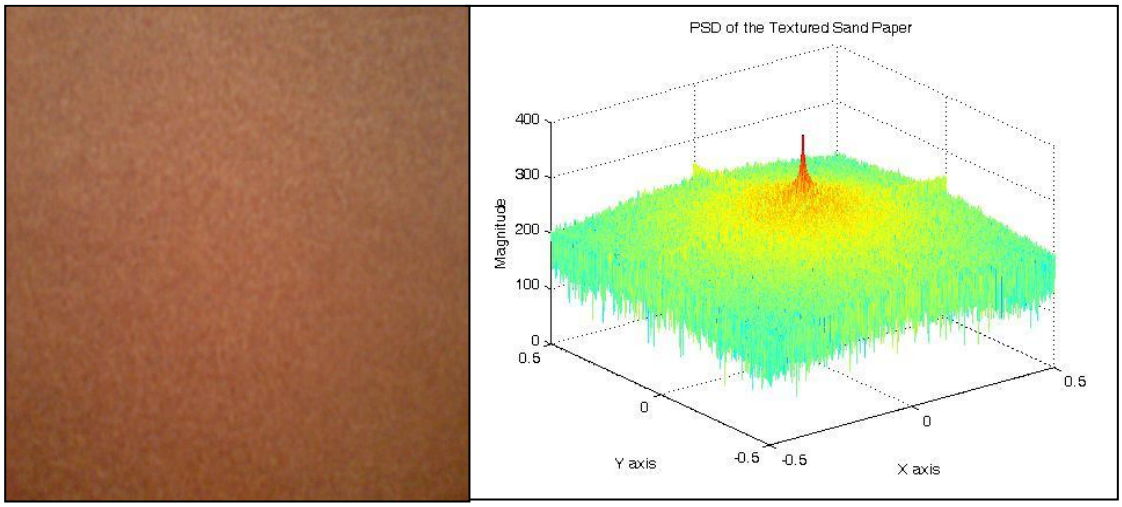


Figure 6.1a: Sand paper pattern Figure 6.1b: PSD plot of the sand paper pattern

To determine the accuracy of the setup, the sand paper pattern was moved along the optical path between 800mm and 744mm, and a pair of defocused images was recorded at every 10mm interval. The captured images were processed using the 64 bit Matlab program. The normalised mean depth was mapped to real world coordinates using the Gaussian lens law. The depth estimation results for the filters designed by the Two Step Polynomial Approach and for Watanabe's filters were compared, and shown in Figure (6.2a). The RMS error plot at each distance is presented in Figure (6.2b). From the plots it can be seen that the depth estimates for both the filters are reasonably linear. The RMS errors for the new filters were 9.489mm and 6.8717mm at the furthest and the nearest distance respectively. These correspond to 1.186% and 0.9236% with respect to the distance compared to 1.547% and 1.258% for Watanabe's filters. From these results it can be inferred that the

filters designed using the Two Step Polynomial Approach provided an improved accuracy compared to Watanabe's filters for these natural textures. The next Section provides depth results for 3D objects with natural textures.

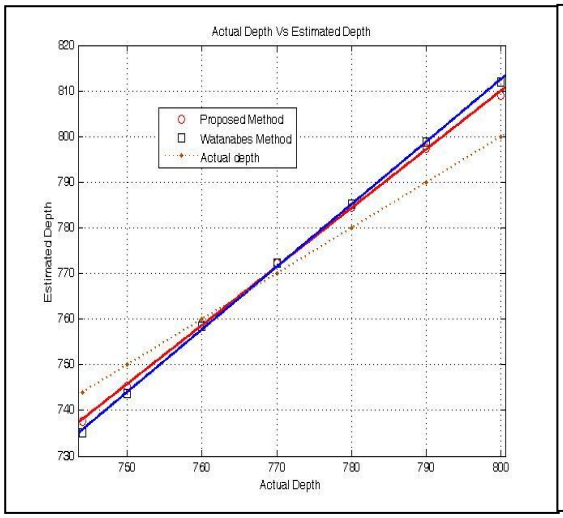


Figure 6.2a: Actual vs. Estimated distance (mm

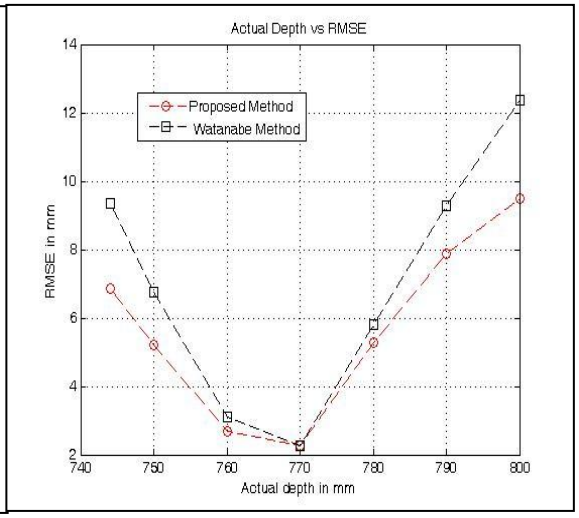


Figure 6.2b: RMSE vs. Actual distance (mm)

6.2. Experiments with 3D structures

This Section provides depth estimation results for real 3D objects with different natural textures. These objects were custom made to set dimensions and thus enable accuracy estimation. Sections 6.2.1 and 6.2.2 present depth measurements of single and multi-step staircase structures made from mild steel. Section 6.2.3 describes the 3D shape recovered from a Cross like structure made from natural wood. For these experiments, three different depth comparison results are presented: - (1) Matlab 64 bit post-filtered depth map which includes both linear and error corrected depth estimates; (2) Matlab fixed point (truncated) post-filtered linear depth output where the bit-widths at each filter stage are set to the corresponding FPGA bit-width (refer to model 3 in Table (5.2)); and (3) FPGA 10 bit post-filtered linear depth output.

6.2.1. Depth estimation results for the 3D, single step staircase structure

A steel gauge of thickness 10mm was contained in a plane perpendicular to the optical axis and placed in a way to provide a sharp change in depth. The Sand paper was used as the background. Figures (6.3) and (6.4) show the scene under investigation along with the near and far-focused images.

Figure 6.3: 3D view of the scene and its corresponding real image

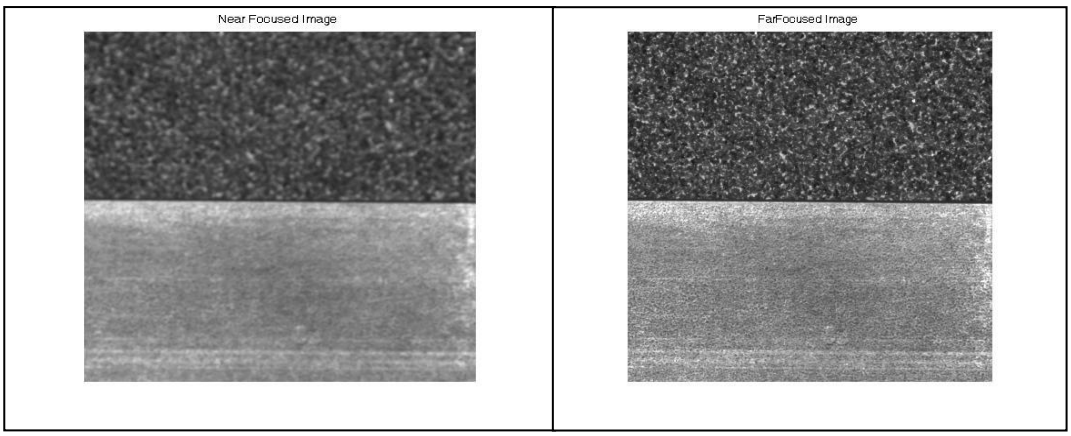


Figure 6.4: Near and far-focused images

The recovered 64 bit Matlab post-filtered depth map and the Matlab truncated depth map are shown in Figure (6.5a) and (6.5b) respectively. The 10 bit post-filtered FPGA depth map is shown in Figure (6.5c). The statistical analysis corresponding to the local depth is provided in Table 6.1.

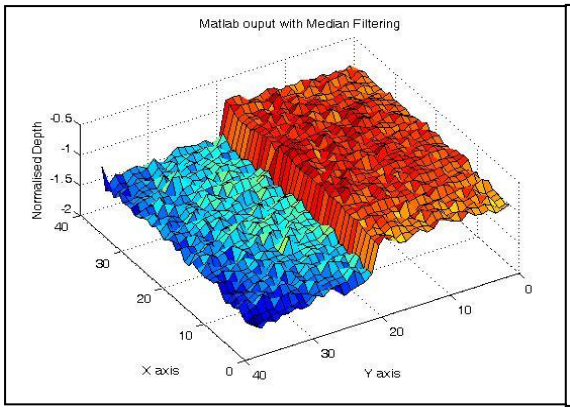


Figure 6.5a: 64 bit Matlab post-processed output

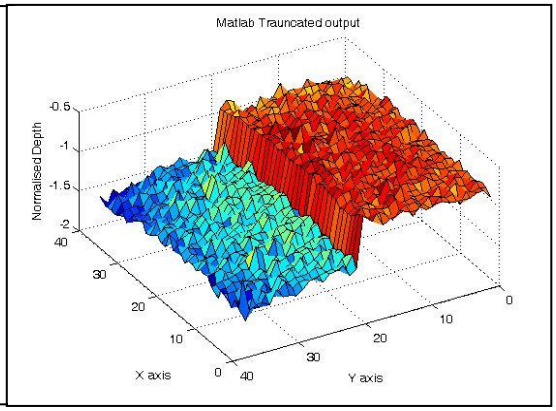


Figure 6.5b: Matlab truncated post-processed output

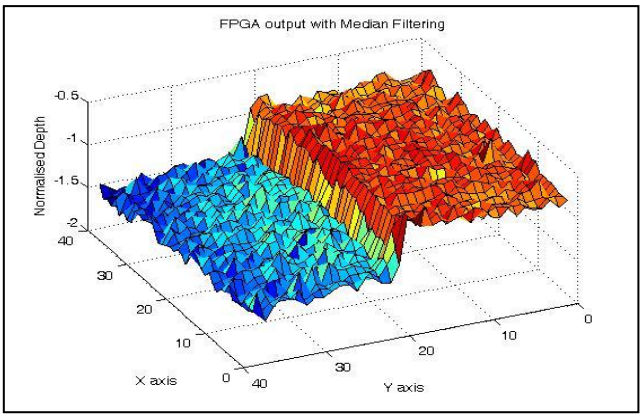


Figure 6.5c: FPGA post-processed output

Depth output from	Results obtained from Depth1= 201x 141 pixels Depth2= 101x 201 pixels.	Depth1: 800mm	Depth2: 790mm.
Matlab 64 bit post-filtered linear and error corrected output.	Estimated distance in mm	810.57	793.604
	Error in mm	+10.57	+3.604
Matlab truncated post-filtered linear depth output	Estimated distance in mm	810.803	794.269
	Error in mm	+10.803	+4.269
FPGA linear depth output with post- filtering	Estimated distance in mm	810.803	793.936
	Error in mm	+10.803	+3.936

Table 6.1: Comparison between Matlab and FPGA depth outputs

From the above Table, it can be inferred that the pipelined processor provides depth measurements comparable to Matlab's depth output. It can be also verified that the designed filters are indeed texture invariant and can be effectively used to recover the depth information from natural textures as shown in Figures (6.5a, b and c). The percentage depth error using the FPGA processor was +1.35% at 800mm and +0.49 % at 790mm. This error is comparable to Matlab 64 bit depth output.

6.2.2. Depth estimation results for the 3D, Multi-step staircase structure

In this experiment three mild steel gauges of thickness 10mm were placed to form a staircase structure as shown in Figure (6.6). The mild steel gauges have different reflectance patterns and hence different textures. The near and far-focused images are shown in Figure (6.7) and the resultant depth maps for the 64 bit Matlab post-filtered depth map and the Matlab truncated depth map are shown in Figures (6.8a) and (6.8b) respectively. The 10 bit post-filtered FPGA depth map is shown in Figure (6.8c). Detailed statistical analysis corresponding to the local depth are provided in Table 6.2.

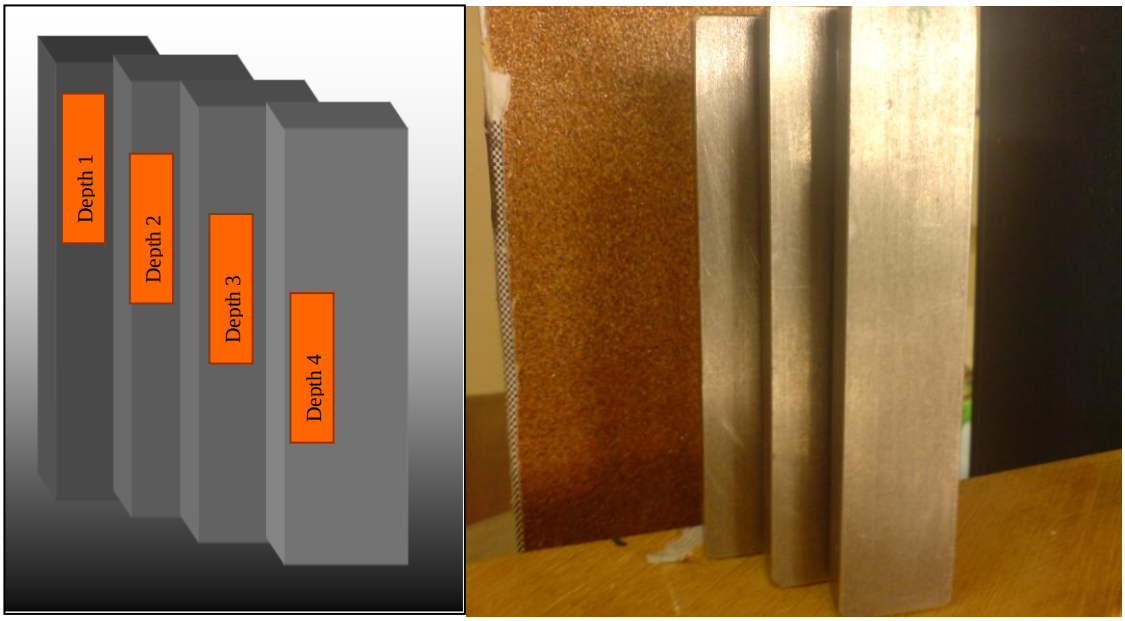


Figure 6.6: 3D view of the scene and its corresponding real image

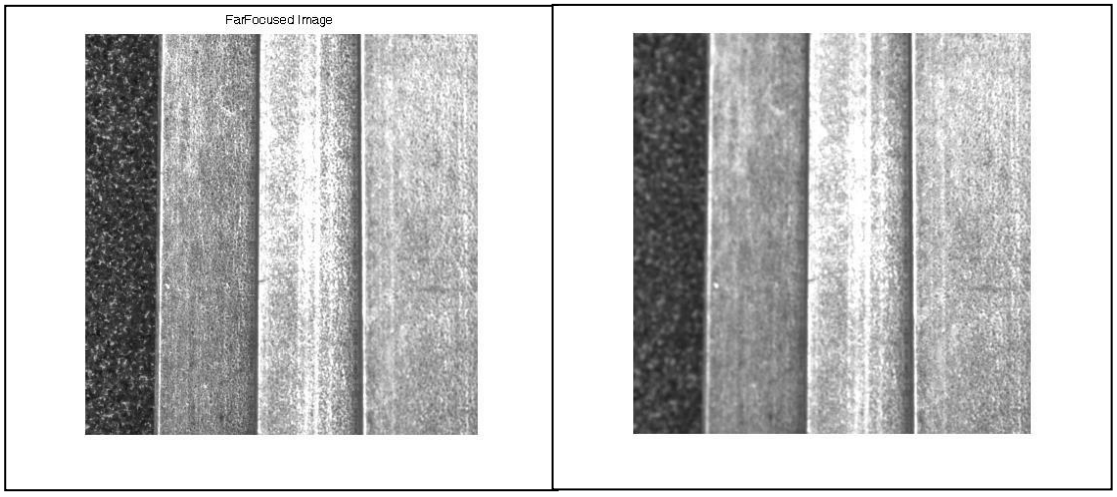


Figure 6.7: Near and far-focused images

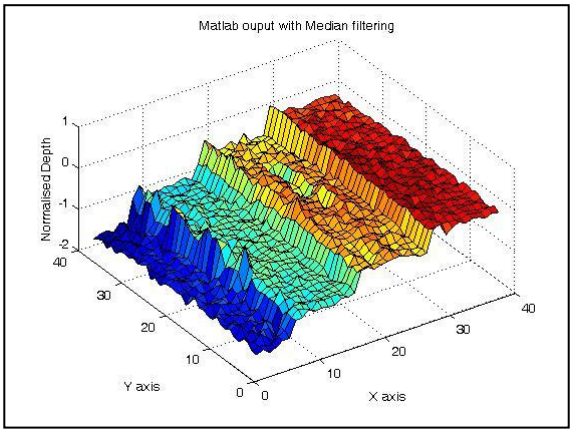


Figure 6.8a: 64 bit Matlab post-processed output

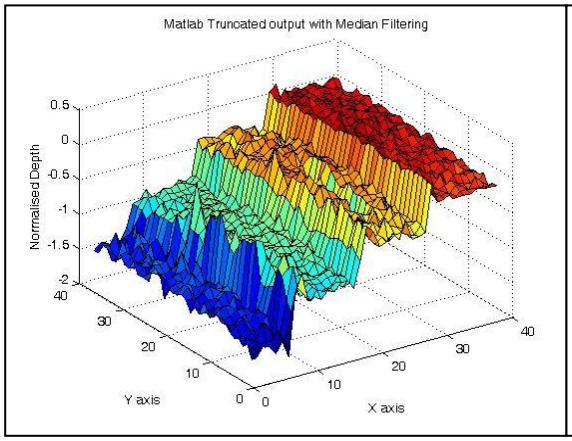


Figure 6.8b: Matlab truncated post-processed output

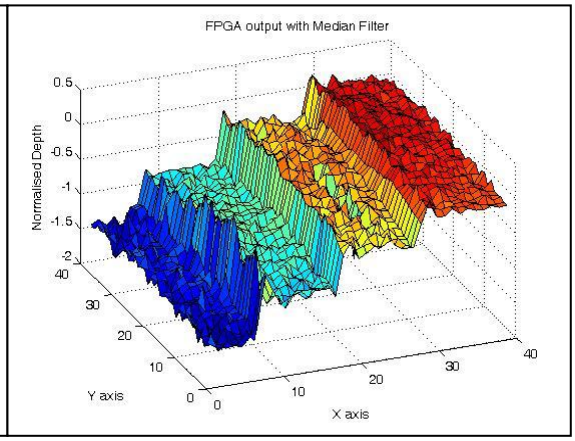


Figure 6.8c: FPGA post-processed output

Depth estimated by	Results calculated from 301x70 pixels	Depth 1 = 800mm	Depth 2 = 790mm	Depth 3 = 780mm	Depth 4 = 790mm
Matlab 64 bit post-filtered linear and error corrected output	Estimated distance in mm	811.96	795.6	780.353	770.46
	Error in mm	+11.96	+5.6	+0.353	+0.461
Matlab truncated post-filtered linear depth output	Estimated distance in mm	812.427	796.045	780.417	770.2535
	Error in mm	+12.427	+6.045	+0.4175	+0.2535
FPGA linear depth output with post-filtering	Estimated distance in mm	812.195	795.378	780.140	770.2535
	Error in mm	+12.195	+5.378	+0.140	+0.2535

Table 6.2: Comparison between Matlab and FPGA depth outputs

It can be seen from the Figures that the depth estimates are fairly stable except at the edges and at regions of high reflectance, the homogenous texture as seen on Depth 3. The sharp spikes seen on the edges are mainly due to the shadowing effects that can be overcome with suitable lighting. The poor depth estimation at homogenous areas is a known drawback of passive depth measurement techniques which can be improved by using external illumination (Active method). The statistical results presented in Table 6.2 illustrate that the pipelined processor provides depth measurements comparable to the Matlab's 64 bit output.

6.2.3. Depth estimation results for the 3D Cross Structure

In this Section, a wooden Cross like structure was contained in a plane perpendicular to the optical axis as shown in Figure (6.9). The thickness at each leg of the cross are presented in Table 6.3. The near and far-focussed images are shown in Figure (6.10). The sand paper texture was used as the background.

The depth results are shown in Figures (6.11 a, b and c). It can be verified, that the depth measurement using the pipelined processor was reasonably accurate and comparable to Matlab's output. The recovered shape was quite smooth and hence proves that the depth recovery using the designed filters has not been affected by the scene's texture. The depth measurement result for each leg of the wooden structure

are presented in Table 6.3. The maximum depth error was +8.955mm at the furthest distance and -4.66mm at the nearest point. These results are comparable to the RMS error estimates for the sand paper pattern presented in Section 6.1.

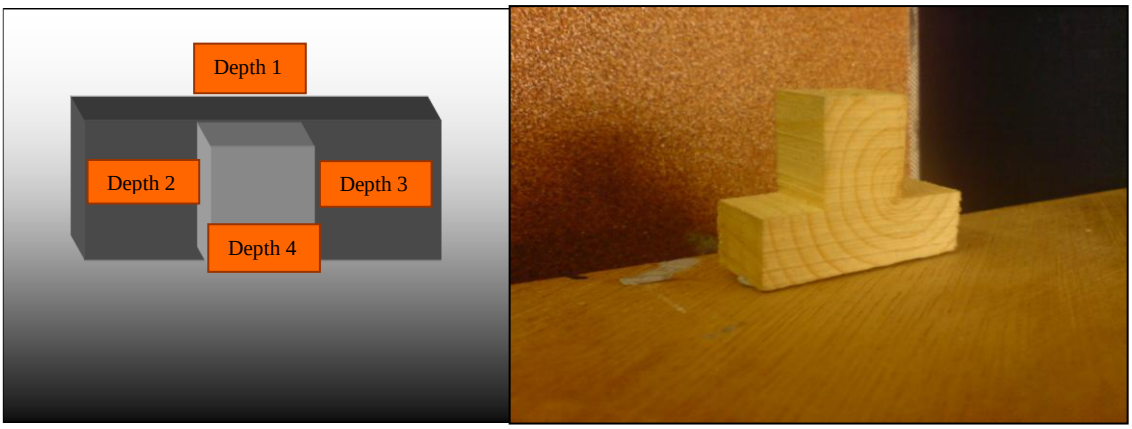


Figure 6.9: 3D view of the scene and its corresponding real image

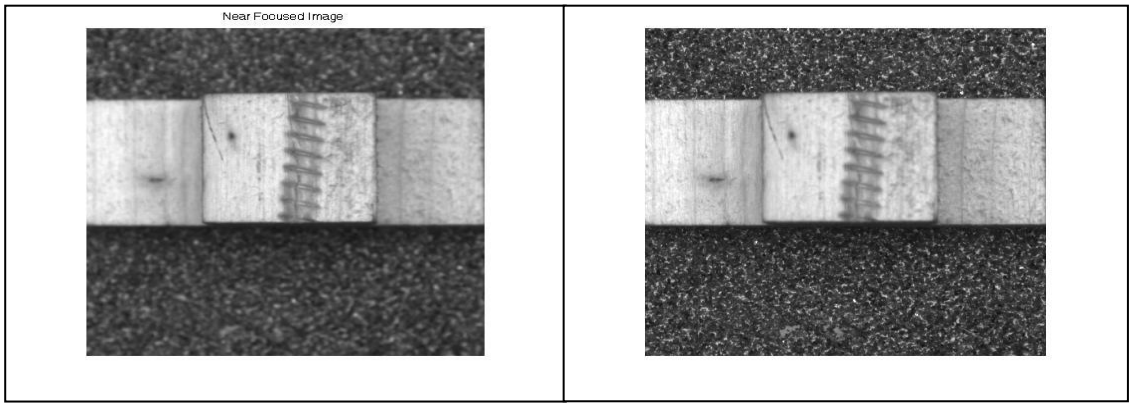


Figure 6.10: Near and far-focused images

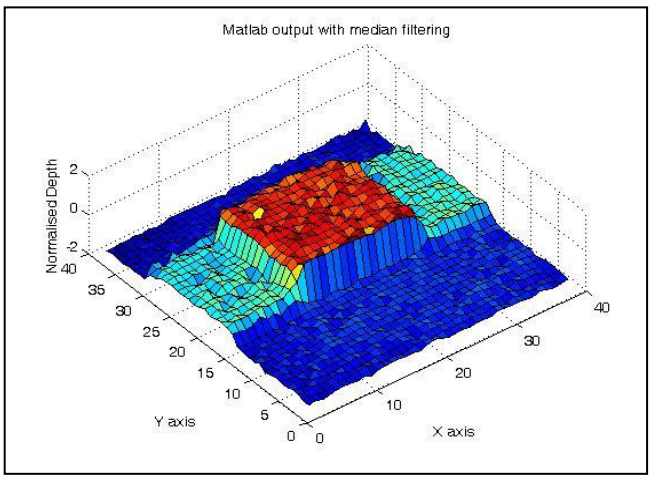


Figure 6.11a: 64 bit Matlab post-processed output

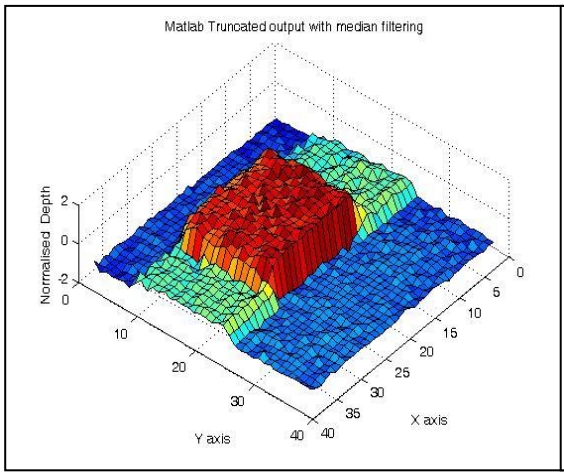


Figure 6.11b: Matlab truncated post-processed output

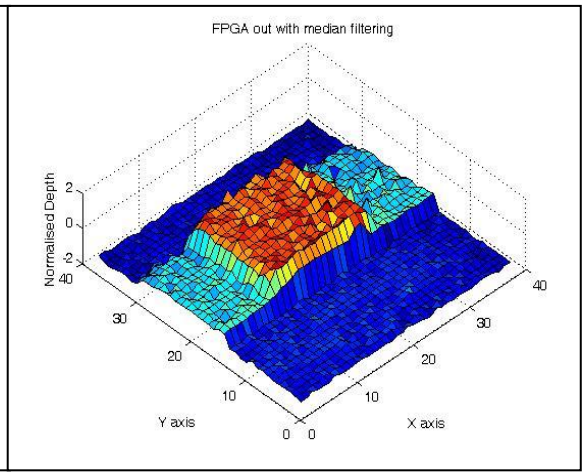


Figure 6.11c: FPGA post-processed output

Depth estimated by	Results calculated from Depth1=100x300 pixels Depth2&3=100x60 pixels Depth4=100x150 pixels	Depth 1 = 800mm	Depth 2 = 785.1mm	Depth 3 = 785.1mm	Depth 4 = 765.5mm
Matlab 64 bit post-filtered linear and error corrected output	Estimated distance in mm	808.95	792,94	791.61	761.642
	Error in mm	+8.956	+7.842	+6.51	-3.858
Matlab truncated post-filtered linear depth output	Estimated distance in mm	809.186	794.712	793.162	760.429
	Error in mm	+9.186	+9.6125	+8.0629	-5.0709
FPGA linear depth output with post-filtering	Estimated distance in mm	808.955	793.39	792.942	760.833
	Error in mm	+8.955	+8.283	+7.842	-4.666

Table 6.3: Comparison between Matlab and FPGA depth outputs

6.3. Shape recovery from complex scenes

This Section provides depth estimation results for objects with arbitrary shapes and textures. Here, the depth results obtained from the pipelined processor are post-filtered using a 9x9 Median filter and then smoothed using a 3x3 Gaussian filter. As before, the post-filtering and the smoothing operations are performed using Matlab. Section 6.3.1 presents depth results related to a wooden object that resembles a temple and Section 6.3.2 shows the shape recovered from arbitrary scene made from sponge. These experiments provide a comparison process of the recovered shapes from the Matlab and FPGA processes.

6.3.1. Shape recovery of the wooden temple

The wooden temple shown in Figure (6.12) was placed perpendicular to the optical axis and two defocused images were captured. As before the far-focused image was at 800mm and near-focused at 744mm. The defocused images are shown in Figure (6.13). The depths recovered using Matlab and FPGA are shown in Figures (6.14a) and (6.14b) respectively.

Figure 6.12: Wooden temple used in the experiment

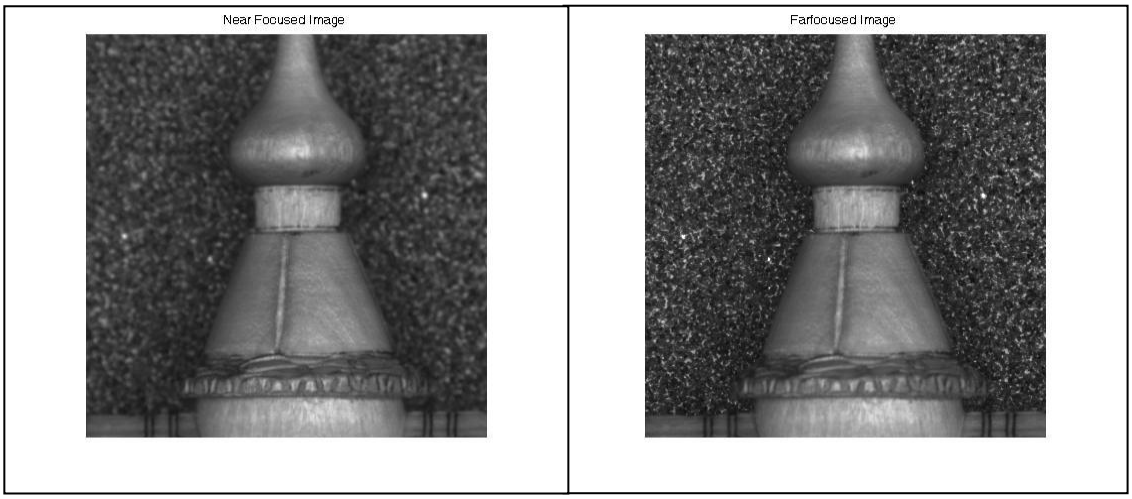


Figure 6.13: Near and far-focused images

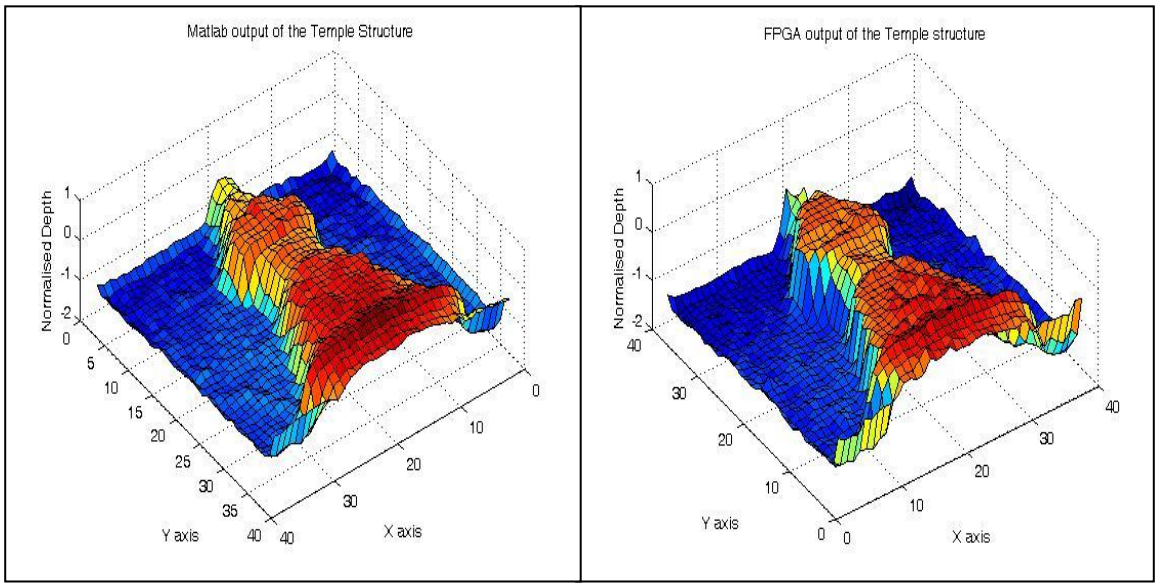


Figure 6.14a: Matlab depth map with 3x3 Gaussian smoothing

Figure 6.14b: FPGA depth map with 3x3 Gaussian smoothing

6.3.2. Shape recovery from a complex scene made from sponge

The test scene made from a sponge and the corresponding defocused images are shown in Figures (6.15) and (6.16) respectively. The test scene, due to its homogenous texture, posed a challenge for a reliable depth estimation using the designed filters. The recovered depth maps using Matlab and FPGA are presented in Figures (6.17a) and (6.17b) respectively.

Figure 6.15: Sponge structure used in the experiment

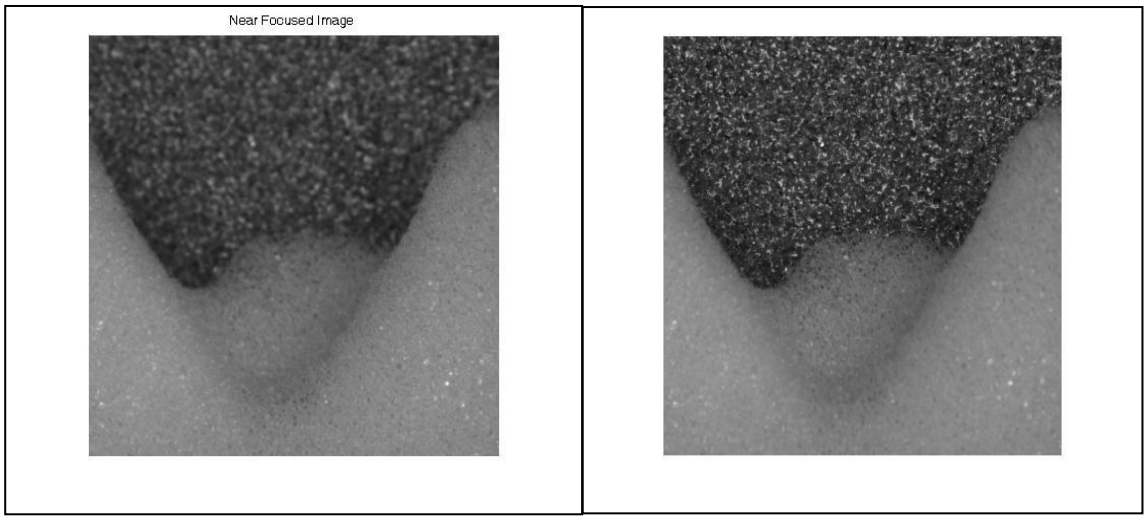


Figure 6.16: Near and far-focused Images

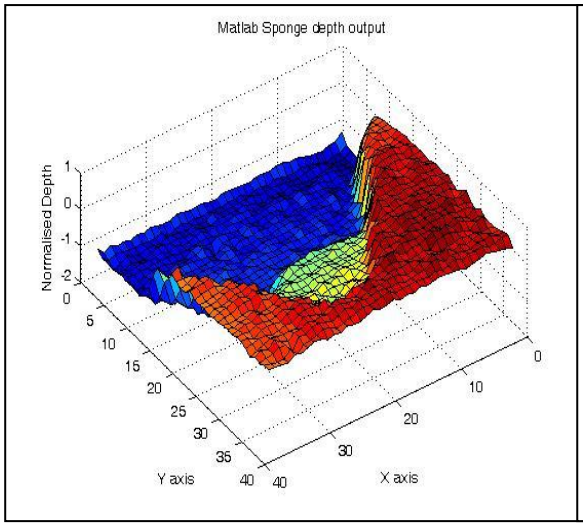


Figure 6.17a: Matlab depth map with 3x3 Gaussian smoothing

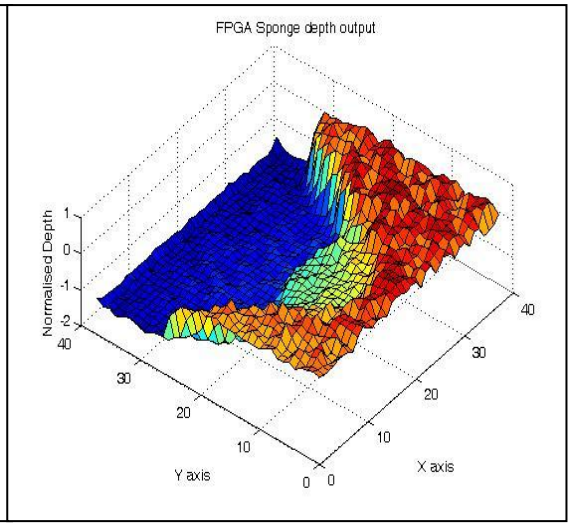


Figure 6.17b: FPGA depth map with 3x3 Gaussian smoothing

The recovered depth maps clearly distinguish the objects placed at different depths and hence the homogeneity of the scene's texture does not seem to interfere with the depth recovery process. Though the depth maps are smoothed using a 3x3 Gaussian, there are still a few random bumps present in the final depth map. These could be minimised by increasing the bit-widths at each stage of the filtering process. Further, the depth accuracy can be improved by applying both the Linear and Error corrected models.

Conclusion

The chapter provided experimental results for 3D planar and non planar objects with natural textures. These results proved that the designed filters were indeed texture invariant and hence can be effectively employed for depth estimation and shape recovery of natural objects. The maximum % RMS error with respect to distance for a flat Sand paper texture was 1.18% for the filters designed by the Two Step Polynomial Approach and 1.54% for the Watanabe's filters (refer to Section 6.1). For 3D objects the maximum absolute error measured was 12mm for the steel step gauge. Experimental results with 3D arbitrary objects (refer Section 6.3) suggested that further smoothing of the depth estimates using a 3x3 Gaussian filter provided a more reliable depth map. At present the post-processing (Median filtering and Gaussian smoothing) operations of the FPGA depth map were performed using Matlab. Steps are underway to implement these in hardware. Further, the depth maps presented in Sections 6.2 and 6.3 verified that the pipelined processor provided depth estimates comparable to Matlab's 64 bit depth output. The accuracy of the depth measurement can be improved by considering:- (1) An increase in the bit-widths at each filtering stage to reduce the rounding errors; and (2) Implementing a lookup table to compute the depth results based on both linear (β) and error corrected depth (β^3) outputs. These improvements require a larger chip area and hence more advanced FPGA device.

CHAPTER 7

Conclusions and Future work

Introduction

The report presents a monocular technique capable of estimating depth in real-time. The technique employed the passive variant of the DFD method, and recovered depth from two differently focused images. The research work can be categorized into three Sections: - (1) Estimation of Image Magnification using Phase Correlation; (2) The design of Rational filters using Two Step Polynomial Approach; and (3) FPGA Implementation of the DFD algorithm. The chapter summarises the research work and recommends avenues for further improvement.

7.1. Estimation of Image Magnification using Phase Correlation

The DFD method requires two images. The simplest way is to capture images with different focus settings, but this would result in an undesirable change in magnification between the defocused images. An optical method referred as telecentric optics was employed by Watanabe and Nayar [41] to reduce the magnification variations. In chapter 3, an algorithm using a Phase Correlation technique was employed to estimate the magnification change between the images and also to optimally position the telecentric aperture.

7.1.1. Analysis and contributions of the research work

Watanabe and Nayar [41] introduced an aperture stop at the front focal plane of the lens that provided an accurate registration between the two defocused images in-terms of magnification. To estimate the magnification, they employed a FFT phase based local shift detection technique, which involved fitting a plane to the phase ratio of the spectra. However, Foroosh and Zerubia [29] in their general paper claimed that the above procedure would render inaccurate results since it was based on fitting a plane to the noisy phase data. Therefore a simple and a more robust method was required to estimate the magnification change. This problem was subsequently addressed in this research work where a novel method based on the Phase Correlation [29] [87] principle was adopted to estimate the magnification.

The algorithm considered the radial shifts due to magnification as analogous to the translation of a local sub-block. First the collective shifts due to magnification and translation for each individual sub-block was calculated and then the global translation estimated from the centre sub-block was used to correct the translations of the non-central sub-blocks. Once the translations were corrected the radial shifts due to magnification were visible and able to be estimated in isolation. Further, the accuracy of the system was increased by considering sub-pixel displacements along the x and y directions. A detailed description of the algorithm along with experimental results for simulated and real images are presented in Sections 3.4 and 3.6 of chapter 3. From these experiments, for a conventional DFD system without an external aperture, the maximum absolute radial shifts measured were 4.48 pixels along the column and 4.83 pixels along the row, but reduced to less than a pixel (0.1524 and 0.8194 pixels) with the inclusion of the telecentric aperture. Hence, telecentric optics ensured pixel to pixel registration between the defocused images. In practice the algorithm was used to position the aperture correctly at the front focal plane and also to provide a translation correction factor for the given experimental setup.

7.1.2. Future Work

The procedure described to determine the focal plane in chapter 3 would only provide an approximate position of the front focal plane for a set distance (working range); since it involved manual adjustments of the screen that contained the screw thread and the multi-leaf adjustable aperture (refer to Figure 3.6). To increase the accuracy of determining the correct focal plane for different set distances, a motorised system is required that controls both the focus setting and the rotational moment of the screw thread. The idea here is to first determine the approximate position of the front focal plane using the conventional method, and then to measure the magnification shifts obtained from a sequence of images acquired on either side of the approximate position. In practice, this is achieved by slight adjustment of the screw thread from its approximate position, and capturing the two defocused images corresponding to the near and the far-focused positions. The images are processed for shift detection using the algorithm explained in chapter 3. The measured shifts would be plotted and the position of the screw thread corresponding to the minimum radial

shift would provide the optimal front focal plane. A pictorial description of the procedure is explained in Figure (7.1). The horizontal axis provides the information of the screw position and the vertical axis presents the estimated radial shifts. The optimum focal position can be accurately determined by considering the screw thread position corresponding to the minimum radial shift.

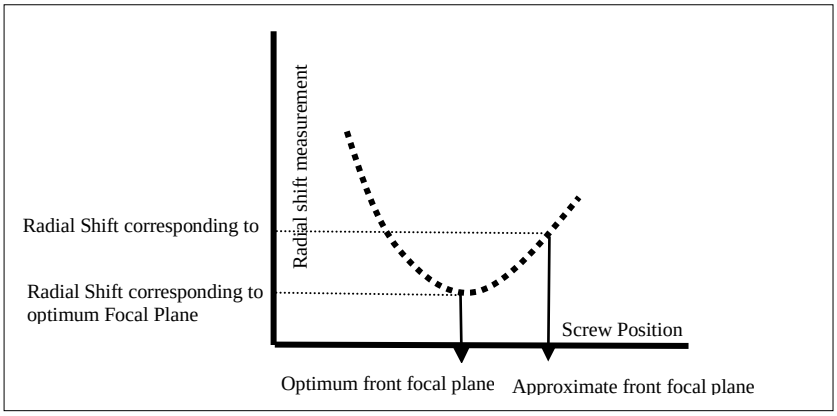


Figure 7.1: Pictorial representation for finding the optimum front focal plane

7.2. Design of Rational filters by the Two Step Polynomial Approach

DFD methods based on the frequency domain approach [1] [2] [6] estimate the depth by considering the amplitude ratio of the defocused images at a particular radial frequency. Watanabe and Nayar [14] provided an improvement. They considered the amplitude ratio between the differences of amplitude of the defocused images to the sum ($\frac{M}{P}$ ratio), and developed a set of broadband filters that accurately modelled the $\frac{M}{P}$ ratio curves. The main advantages of this method were: - (1) Higher accuracy in depth estimation, the RMS error reported was 1.2% with respect to distance; (2) Invariance to scene texture; and (3) A feasible hardware implementation. The main drawback of the method was the complicated and poorly described design procedure to model the rational filters for any given defocus condition. It should be noted that none apart from Watanabe and Nayar [14] have either reproduced the filters or supplemented their work. Further, Watanabe and Nayar have not verified how well their designed filters fit the theoretical $\frac{M}{P}$ ratio curves, and have provided a set of

filters modelled for a single defocus condition $\frac{e}{Fe} = 2.307 pixels$. In this report, the above problems have been addressed. A novel method referred as the Two Step Polynomial Approach was employed to model the rational filters for any given defocus condition. The algorithm and the experimental results are presented in chapter 4.

7.2.1. Analysis and contributions of the research work

The design procedure based on the Two Step Polynomial model was simple and elegant, and provided a better fit to the theoretical $\frac{M}{P}$ ratio than Watanabe's filters (refer to Sections 4.3 and 4.5). Tests with simulated single frequency sinusoidal patterns (λ = 3.2 and α = 0.99) provided a mean depth error of 0.0454 and standard deviation of 0.0128 for the proposed method, and 0.3615 and 0.2008 for Watanabe's model. From the depth results shown in Figure (4.13b), it can be inferred that the proposed model generated a much smoother depth map. With Watanabe's filters, a predominant artefact was seen, which is mainly due to the non-circularity of the filters. Experiments with simulated textures have proved that the designed filters are indeed texture invariant, and that the filters designed by the new model provide a better fit to the actual depth (refer to Section 4.7). Tests with real checkerboard patterns (for a defocus condition of 2.307 pixels) returned an RMS error of 0.6122% at the far-focused and 0.6516% at near-focused positions, for the proposed method. For Watanabe's filters these errors were comparatively higher; 0.9321% and 0.98425% respectively (refer to experiment 1 of Section 4.8). For arbitrary natural textures (Sand paper) the errors were slightly higher (Section 6.1). For the filters designed by the Two Step Polynomial Approach the RMS error with respect to the distance was 1.186% at the far-focused and 0.9236% at the near-focused points, compared to 1.547% and 1.258% for Watanabe's filters. From these results it can be inferred that the filters designed by the proposed method estimated the depth with a higher accuracy. Moreover, the design procedure explained in Section 4.3 can be effectively applied for any defocus conditions by simply modelling the psf.

7.2.2. Future Work

In chapter 4, the rational filter coefficients were obtained by modelling the $\frac{M}{P}$ ratio as a linear combination of the three filters, Gm_1. Gp_1 and Gp_2.

Hence, $\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{Gp_1(u,v)}{Gm_1(u,v)}\beta + \frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3$ ----- (7.1), where $\frac{M(u,v;\alpha)}{P(u,v;\alpha)}$ represents the theoretical $\frac{M}{P}$ ratio, $\frac{Gp_1(u,v)}{Gm_1(u,v)}\beta$ represents the linear model, $\frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3$ represents the error correction model, α and β denote the actual and the estimated depth, and u, v are the frequencies in the horizontal and vertical directions respectively. Here, the $\frac{M}{P}$ ratio was modelled as a cubic polynomial [14], and the filters coefficients were determined by considering the linear and the error correction models, described by the Two Step Polynomial Approach (Section 4.3). An improvement over the existing method would be to model the $\frac{M}{P}$ ratio using a higher order polynomial. For example a fifth order polynomial (equation 7.2) would provide a much closer fit to the theoretical $\frac{M}{P}$ ratio, but this would require an additional refinement filter Gp_3.

So, $$\frac{M(u,v;\alpha)}{P(u,v;\alpha)} = \frac{Gp_1(u,v)}{Gm_1(u,v)}\beta + \frac{Gp_2(u,v)}{Gm_1(u,v)}\beta^3 + \frac{Gp_3(u,v)}{Gm_1(u,v)}\beta^5 \text{ ---- (7.2).}$$

Now, the filter design procedure is based on: - (1) Determining the Linear filters Gm_1 and Gp_1; (2) Determining error correction filter Gp_2, by calculating the error between the actual and the linear model, and fitting a cubic error function to it; and (3) Determining the refinement filter Gp_3, by fitting a fifth order polynomial to the error difference between the actual and error corrected model. Although the above model would provide an improvement to the depth accuracy, it would in-turn increase the overall processing time of the application. Further, from the point of implementation, the DFD algorithm would require more chip area, since the refinement filter Gp_3 requires additional logic.

The next improvement in the filter design would be to accurately measure the psf of the lens. Here, for the given defocus condition the discrete $\frac{M}{P}$ ratio space was determined based on the Pillbox psf. However, Claxton and Staunton [49] observed that the Generalised Gaussian model provided a close fit to the actual psf of the lens. They reported that the psf model fitted using the Generalised Gaussian function performed 8 times better than the Gaussian psf and 14 times better than the Pillbox psf. Hence, the discrete $\frac{M}{P}$ ratio space determined from a more accurate psf model could provide a considerable improvement in the depth accuracy.

7.3. FPGA implementation of the DFD algorithm

The objective of the research work was to develop a real-time depth recovery system. Pentland [3], Nayar [13], and Whelan [77] have developed video rate range sensors, that were based on Active DFD. Though real-time implementation was not presented, Leroy *et al.* [60] have claimed to have developed a passive range system capable of estimating an 800 x 600 pixel depth map in 23ms. The drawbacks reported were the influence of the image texture and the edge density on the recovered depth map. Therefore, a prototype passive DFD system which is insensitive to image texture, and can recover depth in real-time would ideally supplement the existing research. In chapter 5, the DFD algorithm based on [14] was effectively implemented on a Virtex 2P FPGA. The depth recovery results and their comparison to a full precision Matlab output have been presented in Sections 5.6, 6.2 and 6.3.

7.3.1. Analysis and contributions of the research work

The DFD algorithm required five 2D convolutions to be processed in parallel. A two channel five stage pipelined architecture (shown in Figure 5.8) was effectively used to implement the DFD algorithm on the FPGA. The pipelined processor processed a depth map of 400 x 400 pixels in 13.06ms. The number of multipliers required at each stage of the filtering process was reduced from 49 to 10 (79.5% reduction) by

adopting the Triangular design procedure (Section 5.2.1). Four different design models were considered for implementation, and Model 3 (Section 5.4) was implemented since it provided acceptable depth accuracy with minimum chip usage (50%). Experiments with simulated and natural textures suggested that the pipelined processor provided depth maps comparable to Matlab's depth output. Further, since the designed filters were texture invariant, the recovered depth maps were less influenced by the scene's texture.

7.3.2. Future Work

The existing pipelined processor can be further improved to provide more accurate depth measurements. The suggested improvements are:- (1) To employ a pre-computed lookup table that provides an error corrected depth result for each pixel, determined by combining the linear (β) and error corrected depth outputs (β^3); (2) The rounding errors present at each stage of the pipelined processor can be reduced further by increasing the bit-widths at each filtering stage; and (3) The scaling factor and the bit-widths of the filter coefficients can be increased to provide a higher precision. Finally, the DFD program based on the pipelined architecture executes above video rate (13.06ms), but still the image acquisition and the display remain non-real-time. To enable a real-time depth recovery, a DAQ (Data Acquisition board) board capable of capturing and outputting image data at 12.5MHz is a requirement. The input video would comprise two channels for the near and far-focused images. A beam splitter could be used with two CCDs and different path lengths (focus) to provide these (see figure (2.2)). The DFD process would operate at the video rate on the FPGA, and the depth output could be scaled and converted to video for display on a TV monitor.

7.4. Overall Conclusion

A DFD system capable of presenting depth information in real-time would find its usage in industrial and medical applications. In machine vision systems, DFD techniques can be used for segmenting objects based on the depth levels and also in controlling the movements of a robotic arm say on automatic welding inspection

system or in a computerized car assembly centre. In medical applications, a video rate DFD system could be employed in endoscopy and in tele-surgery (remote surgery). In an endoscopic application the DFD system would provide a complete 3D visual inspection of the internal organs. For example a visual inspection of a tumour through a normal endoscope would provide information about the size of the tumour (area information) but not its height. Using a 3D endoscope, the volumetric data of the tumour is available for analysis. This additional information would be helpful in identifying the nature of the tumour (malignant or benign) and planning its treatment. The other areas where a DFD system would find its usage include: - (1) 3D Face recognition systems for Biometric security checks; (2) Virtual reality systems to create 3D characters and for segmenting the objects from the background; and (3) Metrological systems to provide high precision measurements.

To further increase the depth accuracy, a hybrid system incorporating the three main depth recovery techniques (Stereo, Focus and Defocus) can be employed. Subbarao *et al*. [107] has attempted a depth recovery system that integrated the above techniques. The algorithm first computed a rough estimate of the depth (intermediate depth map) using DFD and DFF. Later, stereopsis was used to provide a more accurate 3D shape of the object. The intermediate depth map efficiently reduced the overall computation time of the stereo algorithm, since the correspondence matching was reduced to a narrow image region determined by the approximate shape. The processing time required to compute a depth map of 640 x 480 pixels was about 3min on a desktop PC, but this could be reduced further if dedicated hardware was used. Abbot and Ahuja [108] have combined focus, camera vergence and stereo to develop a stereo-camera imaging system. Here DFF and camera vergence provided the coarse depth map (intermediate depth map), which was later refined by stereopsis. Although the integration of stereo, DFF and DFD can increase the accuracy of the depth estimates, more elaborate investigation is required to identify the relative strengths and weaknesses of the individual techniques. This, in turn, would pave the way in the future to develop a more robust depth measurement system.

APPENDICES

Appendix 1

Weighting Function used for the Rational Filters designed by the Two Step Polynomial Approach

To ensure smoothness and to minimise the depth estimation error a 2D polynomial was fitted to the frequency response of the low pass filter Gm_1, using the weighting function $\sigma g_{m_1}(f_r) = \frac{KGm_1(f_r)}{P(f_r;\alpha)Gp_1(f_r)}$ -- (A_{11}) described in [14], where K=1.

The filter Gp_1 was modelled as a LOG filter, $Gp_1(f_r) = (\frac{f_r}{f_{peak}})^2 \exp(1-(\frac{f_r}{f_{peak}})^2)$, and f_{peak} = 0.4*fnyquist*. The added image $P(f_r;\alpha)$ can be approximated as $P(f_r;\alpha) = \frac{1}{f_r^{\,n}}$.

Here the radial frequency $f_r = \sqrt{u^2+v^2}$ and n the fractal dimension was 2.5. From the Linear Model Gm_1 can be computed as, $Gm_1 = \frac{Gp_1(f_r)}{A(f_r)}$ where $A(f_r)$was the gradient function.

Substituting the values in A_{11}, we get $\sigma g_{m1} = \frac{\left(Gp_1(f_r)/A(f_r)\right).f_r^{\,2.5}}{Gp_1(f_r)}$ -- (A_{12})

Cancelling the effect of Gp_1 we get $\sigma g_{m1} = \frac{f_r^{\,2.5}}{A(f_r)}$ -- (A_{13}).

Thus the frequency samples are weighted using σg_{m1} to ensure smoothness, and with minimum depth error.

For Gp_2 all the coefficients were equally weighted at 1.

Appendix 2

Pre-filter Weighting Function

The pre-filter designed by Watanabe [14] was a LOG filter based on the equation $prefilter(f_r) = (\frac{f_r}{f_{peak}})^2 \exp(1-(\frac{f_r}{f_{peak}})^2)$,where $f\max = 0.246\,pxiel^{-1}$ and $f_{peak} = 0.4 f_{\max}$. By experimentation it was found that when the $f\max = 0.74\,pxiel^{-1}$, the prefilter had a smoother transition compared to Watanabe's [14], and therefore largely avoided the Gibbs effect.

Appendix 3

Necessary Conditions for Setting up the Working Distance for Depth Estimation using Rational Filter Designed by a Two Step Polynomial Approach

f - Focal length of the lens in mm

Fe - *f*-number of the lens

d - Diameter of the aperture in mm

2e - Distance between the near and far-focused images in mm

pixsize- Pixel size of the camera (7.4μm for our AVT Guppy Monochrome camera and 13μm for Watanabe camera)

k_s – Size of the kernel (7 pixels)

Diameter of the lens aperture $d = \frac{f}{Fe}$ mm.

Defocus condition (blur circle radius) = $\frac{e}{Fe * pixsize}$ pixels.

Minimum Frequency that could be resolved by the filter $= \frac{2}{ks}\ pixel^{-1}$.

Maximum Frequency that could be resolved $= 0.73 * \frac{Fe * pixsize}{e}\ pixel^{-1}$.

Maximum Blur Circle Diameter, $\frac{2e}{Fe} \leq 0.73ks\ \ pixels$.

An increase in *Fe* gives an increase in the working distance

An increase in pixel size gives an increase in the working distance

A decrease in the focal length of the lens gives an increase in the working distance

Example Setup for the AVT Guppy with a 50mm lens for defocus condition 2.307 pixels

Defocus Condition	*e in pixels*	*Fe*	$\min fr \geq \frac{2}{k_s}$ $pixel^{-1}$	$\max fr = 0.73\frac{Fe}{e}$ $pixel^{-1}$	Max blur diameter $\frac{2e}{Fe} \leq 0.73k_s$ $pixel$	Near Focussed Distance	Far Focussed distance
$\frac{e}{Fe} = 2.307pix$ Focal length, *f*= 50mm Kernel size k_s=7 Aperture diameter 6.5mm	17.746	7.6923	0.2857	0.3164	4.1614	744mm	800mm

Appendix 4

Offset Correction

An offset correction along the principal axis was required, since it was difficult to mark the exact centre of the compound lens, which is the origin to which the depth measurements refer. The procedure here was to set the Far-Focussed object distance relative to a known scale using a slip-gauge and to estimate the depth at different distances within the depth of field. Once the relative depth measurement had been found a correction offset factor was calculated by fitting a straight line such that the midpoint of the depth of field had ‘zero’ normalised depth. The correction factor can be positive or negative and in practice if the correction factor is positive the far-focussed distance is increased and if it is negative decreased. Once the offset correction was completed the depth measurement experiment was performed and the results recorded.

Appendix 5

Review of the DFD Techniques

Single Image Passive DFD Methods

Author	Technique	Accuracy, merits and demerits
Pentland [1]	Spatial Domain – Edge Based Used Laplacian to determine the maximum rate of image intensity at the edges	±1.25 cm was reported by Grossman. Can recover depth only at edges and requires prior knowledge of the scene
Subbarao and Gurumoorthy [5]	Spatial Domain – Edge Based The spread of the line spread function (LSF), measured from the second central moment (standard deviation distribution of the LSF) was linearly related to the inverse distance	Accuracy not reported. The method works well on isolated edges and causes depth estimation errors in presence of other edges.
Lin and Gu [69]	Spatial Domain – The amount of blurring was estimated from the distribution of pixel intensity estimated using a histogram	RMS error less than 3% when the furthest point was at 1200mm. Required a pre-calibrated mathematical model to relate the blur radius to the actual distance
Namboodiri and Chaudhuri [67]	Statistical Technique - based on inhomogeneous reverse heat equation that estimated the blur information and depth perception using a single image.	Accuracy not reported. The results were compared to Favaro's multiple image diffusion model.

Multi-Image Passive DFD Methods

Author	Technique	Accuracy, merits and demerits
Pentland [1][2]	Frequency Domain- based on comparing the focal error between images taken with different aperture settings	2.5% standard error over 1 cubic meter. Assumed one of captured image to be compete focused which required a pin-hole camera

Subbarao and his research associates [6] [7] [8] [10]	Frequency Domain Approaches 1. Based on estimating the spread parameter using Power Spectral Density. 2. Based on calculating the 1D Fourier Coefficients as in 1D DFD method, where the blur parameter estimated from the LSF was used an index for a look-up table that provided a calibrated psf modelled either as Gaussian or Pillbox. Spatial Domain Approaches 1. Based on STM (Spatial Domain Convolution Deconvolution Transform) 2,Based on inverting a Shift Variant Blur Model using Rao Transform	1. Relaxed Pentland's requirement of a pinhole aperture and recovered depth by considering two images (which may or may not be focused) acquired with different camera settings. 2. Accuracy of 3.7% RMS for auto-focusing applications over a distance of 0.6meters to infinity. For ranging application, the RMS error was 4% at 0.6 meter and linearly increased to 30% RMS error at 5 meter distance 1. Percentage error in terms of distance was about 2.3% at 0.6 meter and it linearly increased to about 20% at the 5 meter distance. 2. With simulated images suggested a maximum error of 3% with respect to the distance.
Ens and Lawrence [12]	Spatial Domain – Based on Matrix Regularization Approach	RMS error of 1.3%. The disadvantages of the method are that it was based on smoothness assumption and it was computationally intensive
Xing and Shafer [50] [54]	Frequency Domain approach based on Moment and Hyper geometric narrow band filters	Results were 27 times better than Subbarao's frequency domain approach. From the computational perspective, the filters required more logic support, hence not suitable for practical implementation.
Watanabe and Nayar [14]	Frequency Domain Approach based on Rational Filters	An improvement over the existing techniques was provided my considering the normalised M/P ratio. Four 7x7 texture invariant filters were designed to retrieve the depth. Magnification variations between the defocused images were corrected

		using Telecentric optics Accuracy reported was between 0.5 and 1.2% with respect to the distance from the lens. Can be implemented for real time depth estimation.
Chaudhuri and his research associates [18] [19] [48] [78][79]	Statistical Approach: 1. Applied the Space Frequency Representation to the problem of depth defocus and modelled the shift variant blurring image using the complex spectrogram (CS) and pseudo-Wigner (PWD) distribution . 2. Algorithm based on Markov Random Field (MRF) with Simulated Annealing Technique.	1.RMSE reported was 4.84% for the scene whose farthest point was at 115 cm from the lens surface 2. Simultaneously recovered the depth information and the radiance of the scene. RMS error was 1.96% when the furthest distance was at 96.6cm
	3.Algorithm based on the linear diffusion heat model where the blurring related to the diffusion coefficient was modelled as using Markov Random Field	Accuracy not reported but results with real images suggested better results than Favaro's model [68]
Favaro and his research associates [44] [45][64]	Statistical Approach: 1.Iterative method based on Information Divergence 2. Optimization method considering 3D shape and radiance recovery as a finite dimensional optimization problem 3. Algorithm based on Matrix Multiplication using Singular Value decomposition (SVD). The depth map along with the radiance of the object was recovered. 4. Algorithm based on Anisotropic diffusion heat equation. Fourier Domain Approach 1.Based on modelling a 3D psf and applying SVD in the frequency domain	1.No theoretical results were presented. 2. Considered the blurring model as a shift variant process but no theoretical results were presented 3. the depth error reported 27 mm for a scene placed between 520mm and 850mm 4. Depth maps recovered are favourable and the algorithm can be employed for 3D shape segmentation No theoretical results were presented The depth and the radiance of the scene were recovered quite accurately

Deschenes and his research associates [15] [21] [22] [61]	Spatial domain approach- based on Subbarao 'S' transform method 1.Model based on local image decomposition using higher order Hermite Polynomials 2.Algorithm based on considering the spatial errors involved during image acquisition process 3. Algorithm based on determining the magnitude of the blurred edges.	1.Obervesed 'S' transform was only capable of estimating depth at line edges and hence extended Subbarao's work using Hermite Polynomials to estimate depth junction like L, V, T, Y and X . Accuracy reported was 2.21% for a planar object whose furthest point was at 125 cm. 2. Accuracy reported was 1.68% with depth density of 100. 3. The maximum mean depth error reported was 20.05mm between a working range of 790mm and 990mm with real time depth computation at 23ms.The main drawback of the method was the influence of the edge density and the characteristic of the image textures on the accuracy of the estimated depth.
Simon and his research associates [58]	Spatial domain method similar to Subbarao, by comparing the gradient ratio between a thick and a thin edge	No theoretical results were provided but acquisition of a sharp image was required external lighting, this drawback was overcome by using three image, but this introduced additional complexity in image acquisition process
Choi *et al.* and Hor *et al.* [55] [56] [57]	Algorithms were based on Wavelet transform	Choi *et.al* reported for planar slanted object the depth recovered using wavelets had a lower RMS error of 0.8181cm when compared to other methods; Fourier, Spatial and Laplacian., where the RMS errors were 2.119 cm, 1.3251 cm and 1.8517 cm respectively. The working range of the experiments was between 150 cm and 180 cm.
Swain *et al.* [86]	A method based on Fuzzy Logic was suggested to improve the accuracy of the	Accuracy reported was depth error of less than 1.5% over a working

	depth estimates.	range of 7 feet (2133mm) to 11 feet (3352mm) Drawbacks (1) Test images should contain high frequencies (2) the window selected for depth estimation should have a single depth and (3) the membership function of the fuzzy logic are required to be tuned for different camera settings, which was time consuming and based on trial and error.
McCloskey *et al.* [62]	Algorithm based on reverse correlation principle	The RMS error in terms of absolute depth for the simulated images was between 0.4% and 0.8 %. Results with dense depth maps were not presented for real image.
Baba *et al.* [63]	Algorithm based on zoom changes	Experiments were performed only on edges, with multiple targets placed at several depths provided a maximum error 1945.9mm when the target was at 3000 mm.

Active DFD methods

Author	Technique	Accuracy, merits and demerits
Pentland and his research associates [3]	Single image technique based on Frequency Domain, where the 'hump energy 'of defocused pattern was compared with the known focused pattern.	RMS error of 0.5% was reported for planar stationary object and 5% for rolling golf ball example
Nayar and his research associates [13]	Frequency domain Technique Two defocused images were used and the tuned focus operator (Laplacian) was employed to respond to the single dominant frequency of the projected pattern	Accuracy of 0.3% RMS error with respect to distance. Involved extensive optimization and expensive fabrication techniques to determine the optimum pattern Developed a real-time range sensor

		capable of estimating depth at 30 frames per second
Ghita and Whelan [70] [76]	Frequency domain Technique Based on Watanabe's method but employed interpolation methods and avoided the pattern fabrication method suggested by Nayar et.al.	The lowest accuracy achieved was 3.4% normalized with respect to the distance. A bin picking system based on active depth from defocus technique was presented
Li Ma and Staunton [71]	Method based on Artificial Neural Networks. The object was first isolated from its background and the depth was estimated using a three layered neural network designed using the Back-Propagation algorithm.	The model was trained with checkerboard images but it effectively recovered the depth map of images with natural textures. High resolution data was used by the authors to maximize the depth accuracy.

Appendix 6

FPGA Simulation Report based on Xilinx ISE 10.1

```
=========================================================================
*                    Final Report                    *
=========================================================================
Final Results
RTL Top Level Output File Name     : topmod.ngr
Top Level Output File Name         : topmod
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : NO

Design Statistics
# IOs                              : 65

Cell Usage :
# BELS                             : 21738
#      GND                         : 32
#      INV                         : 86
#      LUT1                        : 686
#      LUT2                        : 1668
#      LUT2_L                      : 15
#      LUT3                        : 2534
#      LUT3_L                      : 28
#      LUT4                        : 4948
#      LUT4_D                      : 29
#      MULT_AND                    : 246
#      MUXCY                       : 6013
#      MUXF5                       : 43
#      VCC                         : 32
#      XORCY                       : 5378
#      Flip-Flops/Latches          : 7609
#      FD                          : 1695
#      FDC                         : 319
#      FDCE                        : 1210
#      FDE                         : 1551
#      FDP                         : 260
#      FDPE                        : 1437
#      FDR                         : 1092
#      FDS                         : 45
#      RAMS                        : 29
#      RAMB16_S36_S36              : 29
# Shift Registers                  : 34
#      SRL16                       : 26
#      SRL16E                      : 8
```

```
# Clock Buffers                    : 1
#       BUFGP                      : 1
#    IO Buffers                    : 32
#        OBUF                      : 32
#   MULTs                          : 120
#   MULT18X18                      : 75
#  MULT18X18S                      : 45
=========================================================================
=============
```

Device utilization summary:

Selected Device: 2vp30ff896-7

```
 Number of Slices:                    6899  out of  13696    50%
 Number of Slice Flip Flops:          7609  out of  27392    27%
 Number of 4 input LUTs:             10028  out of  27392    36%
 Number used as logic:                9994
 Number used as Shift registers:      34
 Number of IOs:                      65
 Number of bonded IOBs:               33  out of    556     5%
 Number of BRAMs:                     29  out of    136    21%
 Number of MULT18X18s:               120  out of    136    88%
 Number of GCLKs:                     1  out of     16     6%
```

Partition Resource Summary:

No Partitions were found in this design.

==========

Appendix 7: Sub-pixel Estimation - Derivation

Consider two images $f_1(x,y)$ and $f_2(x,y)$ with a sub-pixel shift $\frac{x_0}{M}, \frac{y_0}{N}$ obtained by down-sampling a high resolution image by factors M and N along x and y axes, then the normalised Cross Power Spectrum of the two images with their Fourier Transforms, $F(u,v)$ and $F(u,v)exp(-i(ux_0+vy_0))$ is defined as [29]

$$C(u,v) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} h_{mn}(u,v)\exp(-i(\frac{u+2\pi m}{M}x_0, \frac{v+2\pi m}{N}y_0)) \text{ --- (A7.1)}$$

where $h_{mn} = \dfrac{f(\frac{u+2\pi m}{M}, \frac{v+2\pi n}{N})}{\sum_{m'=0}^{M-1}\sum_{n'=0}^{N-1} f(\frac{u+2\pi m'}{M}, \frac{v+2\pi n'}{N})}$

The inverse transform of the of $C(u,v)$ yields a Dirchlet function which closely approximates a sinc function [29]. Therefore,

$$C(x,y) = \frac{\sin(\pi(Mx-x_0))}{\pi(Mx-x_0)}\frac{\sin(\pi(Ny-y_0))}{\pi(Ny-y_0)} \text{ ---------- (A7.2)}$$

Now if the signal power along the x-axis is concentrated between the coordinates 0,0 and 1,0, then

$$C(0,0) = \frac{\sin(\pi x_0)}{\pi x_0}\frac{\sin(\pi y_0)}{\pi y_0} ----------(A7.3)$$

$$C(1,0) = \frac{\sin(\pi(M-x_0))}{\pi(M-x_0)}\frac{\sin(\pi y_0)}{\pi y_0} ----------(A7.4)$$

Dividing A7.4 by A7.3,

$$\frac{C(1,0)}{C(0,0)} = \frac{\sin(\pi(M-x_0)\pi x_0}{\sin(\pi x_0)\pi(M-x_0)} \text{ ---------- (A7.5)}$$

Rearranging

$$\frac{C(1,0)\sin(\pi x_0)}{C(0,0)\pi x_0} = \frac{\sin(\pi(M-x_0)}{\pi(M-x_0)} \text{ ---- (A7.6)}$$

using $\sin(A \pm B) = \sin A \sin B \pm \cos A \sin B$ we get $\sin(\pi(M-x_0)) = \pm \sin \pi x_0$, where $M=1,2,3$.. Taking x_0 common we have

$x_0\left(C(0,0) \pm C(1,0)\right) = MC(1,0)$ --- (A7.7). Therefore the sub-pixel shift

$$\Delta x = \frac{x_0}{M} = \frac{C(1,0)}{C(0,0) \pm C(1,0)} \text{ ------- (A7.7)}$$

BIBLIOGRAPHY

[1] A.P. Pentland, *A new sense for depth of field*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 9, No. 4, pp. 523-531, July 1987.

[2] A. Pentland, T. Darrell, M. Turk, W. Huang, A simple, real-time range camera, IEEE Proceedings of Computer Vision and Pattern Recognition (CVPR), Vol.4, Issue 8, pp. 256 – 261, Jun 1989.

[3] A. Pentland, T. Darrell, S. Scherock and B.Girod, *Simple range cameras based on focal error*, Journal of Optical Society of America (JOSA) A, Vol. 11, No. 11, pp. 2925-2934. Nov 1994.

[4] P. Grossman, *Depth from Focus*, Pattern Recognition, Vol. 9, No.1, pp.63-69, 1987.

[5] M. Subbarao and N. Gurumoorthy, *Depth recovery from blurred edges*, Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp.498-503, June 1988.

[6] M. Subbarao, *Parallel depth recover by changing camera parameters*, 2nd International Conference on Computer Vision, pp. 149-155, Florida, Dec. 1988.

[7] M. Subbarao and T.C Wei, *Depth from defocus and rapid auto focussing: a practical approach*, Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 773-776, June 1992.

[8] M. Subbarao and G. Surya, *Depth from defocus: spatial domain approach*, International Journal of Computer Vision, Vol.13, No.3, pp.271-294, 1994.

[9] S.Y. Park and J.Y. Moon, *Image based calibration of spatial domain depth from defocus and application to automatic focus tracking*, 7th Asian Conference on Computer Vision, Vol. 3851, pp.754-763, Jan 13-15, 2006.

[10] M.Subbarao, T.C.Wei, G.Surya, *Focused image recovery from two defocused images recorded with different camera setting*, IEEE Transactions on Image Processing Vol.4.No.12.pp.1613-1628, December 1995.

[11] P.Meer and I.Weiss, *Smoothed differentiation filters for images*, Journal of Visual Communication and Image Representation, Vol. 3, pp.58-72, 1992

[12] J. Ens and P. Lawrence, *An investigation of methods for determining depth from defocus*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.15, Issue 12, pp. 97-108, 1993.

[13] S.K. Nayar, M.Watanabe and M. Noguchi, *Real-time focus range sensor,* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.18, Issue 12, pp.1186-1198, 1996.

[14] M. Watanabe and S.K. Nayar, *Rational filters for passive depth from defocus,* International Journal of Computer Vision, Vol.27, No.3, pp.203-225, 1998.

[15] D. Ziou and F. Deschenes, *Depth from defocus-estimation in spatial domain,* Computer Vision and Image Understanding, Vol.81, Issue 2, pp.43-165, 2001.

[16] P. Favaro, A. Mennucci and S. Soatto, *Observing shape from defocused images,* International Journal of Computer Vision, Vol.52, Issue 1, pp. 25-43, 2003.

[17] P. Favaro and S. Soatto, *A geometric approach to shape from defocus,* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.27, Issue 3, pp. 406-417, 2005

[18] D. Rajan and S. Chaudhuri, *Simultaneous estimation of super-resolved scene and depth map from low resolution defocused observations,* IEEE Transactions on Pattern and Machine Intelligence(PAMI), Vol.25, No.9, pp. 1102-1117, 2003.

[19] A.N. Rajagopalan and S. Chaudhuri, *An MRF model-based approach to simultaneous recovery of depth and restoration from defocused images,* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 21, No. 7, pp. 577-589, 1999.

[20] A.N. Rajagopalan and S. Chaudhuri, *Depth from defocus: a real aperture imaging approach,* Springer, New York, USA, 1999.

[21] D.Ziou, *Passive depth from defocus using spatial domain approach,* Proc. of 6th the International Conference of Computer Vision, pp. 799-804: January 1998.

[22] F. Deschênes, D. Ziou, P. Fuchs, *Homotopy-based estimation of depth cues in spatial domain,* Proc. 16th International Conference on Pattern Recognition (ICPR'02) - Vol. 3, pp.627- 630, 2002.

[23] S. Tabbone, D. Ziou, *On the behaviour of the Laplacian of Gaussian for junction models,* 2nd Annual Joint Conference on Information Sciences, pp. 304-307, 1995.

[24] R.C. Gonzalez and R.E. Woods, *Digital Image Processing,* 2nd Edition, Prentice Hall, 2002.

[25] L.G. Brown, *Survey of image registration techniques,* ACM Computing Surveys, Vol. 24, No.4, December 1992.

[26] D.L. Barnea and H.F. Silverman, *A class of algorithms for fast digital image registration,* IEEE Transactions on Computers, Vol. 21, pp. 179-186, 1972.

[27] R. Bracewell, *The Fourier transforms and its applications,* McGraw-Hill Inc, 1965.

[28] C.D. Kuglin and D.C. Hines, *The phase correlation image alignment method,* Proc. of IEEE International Conference on Cybernetics and Society, pp. 163-165, 1975

[29] H. Foroosh and J.B. Zerubia, *Extension of Phase Correlation to sub-pixel registration,* IEEE Transaction on Image Processing Vol.11, No.3, pp. 188-200, March 2002.

[30] K. Takita and M.A. Muquit, *A sub-pixel correspondence search technique for computer vision applications,* IEICE Transaction Fundamentals, Vol. E87-A, No.8, pp. 1913-1923, Aug 2004.

[31] E. DE Castro and C. Morandi, *Registration of translated and rotated images using Finite Fourier Transforms,* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 9, Issue 5, pp. 700-703, 1987.

[32] B.S. Reddy and B.N. Chatterji, *An FFT based technique for translation, rotation and scale - invariant image registration,* IEEE Transaction on Image Processing, Vol. 5 No 8, pp. 1266-1271, August 1996.

[33] S. Ranade and A. Rosenfeld, Point pattern matching by relaxation, Pattern Recognition, Vol.12, pp. 269-275, 1980.

[34] J. Ton and A.K. Jain, *Registering Landsat images by point matching*, IEEE Transactions on Geoscience and Remote Sensing, Vol. 27, Issue 5, pp.642-651, Sept 1989.

[35] C.G.Schiek , R.Aaiza, J.M. Hurtado, A.A. Valasco, V.Kreinovich and V.Sinyansky, *Images with Uncertainty: Efficients Algorithm for Shift, Rotation, Scaling and Registration, and their Application to Geosciences,* Soft Computing in Image Processing, Recent Advances, Springer Verlag, pp. 35-64, 2007.

[36] P.Favaro and S.Soatto, *Learning Shape from Defocus,* Lecture notes on Computer Science, Vol. 2351. also in Proc. of 7th European Conference of Computer Vision, Part 2, pp. 735-745, 2002.

[37] R. Bracewell, *Two Dimensional Imaging,* Prentice-Hall Inc., 1995.

[38] R.G. Willson and S.A. Shafer, *Modelling and Calibration of automated zoom lens,* Technical Report CMU-RI-TR-94-03, Robotics Institute, Carnegie Mellon Univ. Pittsburgh, 1994.

[39] T. Darrel and K.Wohn, *Pyramid based depth from focus*, IEEE Proceedings of Conference of Computer Vision and Pattern Recognition (CVPR), pp. 504-509, June 1988.

[40] SF. Ray, *Applied Photographic Optics*, Focal Press, London and Boston, 1988.

[41] M. Watanabe and S.K. Nayar, *Telecentric Optics for focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.19 No.12, pp. 1360-1365, Dec 1997.

[42] M. Watanabe and S.K. Nayar, *Minimal operator set for texture invariant depth from defocus*, Technical Report CUCS-031-95, Dept. of Computer Science, Columbia University, New York, U.S.A, 1995.

[43] M. Born and E.Wolf, *Principles of Optics*, Pergamon Press, 1975.

[44] P. Favaro and S. Soatto, *Shape and Radiance Estimation from Information-divergence of Blurred Images*, In Proceedings of European Conference of Computer Vision - Part 1, pp. 755 – 768, 2000.

[45] H. Jin and P. Favaro, *A Variational Approach to Shape From Defocus*, Proceedings of 7th European Conference on Computer Vision- Part 2, pp. 18–31, 2002.

[46] I. Csiszar, *Why least squares and maximum entropy? An axiomatic approach to linear inverse problems*, Annals of Statistics, Vol. 19, No. 4, pp. 2032-2066, 1991.

[47] D.L. Snyder, T.J. Schulz, J.A. O'Sullivan, *Deblurring subject to non-negativity constraints*, IEEE Transactions on Signal Processing, Vol. 40, Issue 5, pp. 1143- 1150, 1992.

[48] A.N. Rajagopalan and S. Chaudhuri, *A Variational Approach to Recovering Depth From Defocused Images*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, Issue 10, pp. 1158-1164, 1997.

[49] C.D. Claxton and R.C. Staunton, *Measurement of point spread function of a noisy imaging system*, Journal of Optical Society of America (JOSA) A, Vol. 25, Issue 1 , pp. 159-170, 2008.

[50] Y. Xiong and S.A. Shafer, *Moment filters for high precision computation of focus and stereo*, Proc. of the International Conference on Intelligent Robots and Systems, Vol. 3, pp. 3108, 1995.

[51] Dan E. Dudgeon and Russell M. Mercurial, *MultiDimensional Digital Signal Processing*, Prentice-Hall, Inc., New Jersey, 1984.

[52] Jae S. Lim, *Two Dimensional Imaging Signal and Image Processing*, Prentice Hall, Inc, 1990.

[53] B.K.P. Horn, *Robot Vision*, The MIT Press, 1986.

[54] Y. Xiong and S.A. Shafer, *Moment and Hypergeometric Filters for High Precision Computation of Focus, Stereo and Optical Flow*, International Journal of Computer Vision, Vol. 22, Issue 1, pp. 25-59, 1997.

[55] MawKae Hor, J.Y. Chen, and Kuo-Shen Chen, *Wavelet Transform in Depth Recovery* Proceedings of SPIE, Vol. 2055, pp. 463-474, Aug 1993. Intelligent Robots and Computer Vision XII: Algorithms and Techniques (Proceedings Volume).

[56] M. Asif, A.S. Malik , Tae-Sun Choi, *3D shape recovery from image defocus using Wavelet analysis*, International Conference on Image Processing (ICIP), Vol.1, pp. 1025- 1028, Sept. 2005.

[57] M. Asif, Tae-Sun Choi, *Depth from defocus using Wavelet Transforms* IEICE Transactions on Information and Systems, Vol. E87-D, No.1, pp.250-253, Jan 2004.

[58] C. Simon, F. Bicking, T. Simon, *Depth estimation based on thick oriented edges in images*, IEEE International Symposium on Industrial Electronics, Vol.1, pp. 135- 140, 2004.

[59] C. Simon, T. Simon, *Depth Perception from three blurred images*, 32nd Annual Conference on IEEE Industrial Electronics (IECON 2006), pp. 3222-3226, Nov. 2006.

[60] J.V. Leroy, T. Simon, F. Deschenes, *Real time monocular Depth from Defocus*, 3rd International Conference on Image and Signal Processing, pp.103-111, Jul 2008.

[61] F. Deschenes, D. Ziou and P. Fuchs, *Improved estimation of defocus blur and spatial shifts in spatial domain: homotopy-based approach*, Pattern Recognition, Vol.36, Issue 9, pp. 2105-2125, 2003.

[62] S. McCloskey, M. Langer, K. Siddiqi, *The Reverse Projection Correlation Principle for Depth from Defocus*, Proc. of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT), pp. 607-614, June, 2006.

[63] M. Baba, A. Oda, N. Asada and H. Yamashita, *Depth from defocus by zooming using thin lens-based zoom model*, Electronics and Communications in Japan (Part 2- Electronics), Vol. 89, Issue 9, pp. 53-62, 2006.

[64] P.Favarao and A.Duci, *A Theory of Defocus via Fourier Analysis*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.1-8, June 2008.

[65] R. Zhang, P.S. Tsai, J.E. Cryer, M. Shah, *Shape from Shading*: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 21 No. 8, pp.690-706, Aug.1999.

[66] P. Favaro and S. Soatto, *3-d shape estimation and image restoration: Exploiting defocus and motion-blur*, Springer-Verlag, 2006.

[67] V.P. Namboodiri and S. Chaudhuri, *Recovery of relative depth from single observation using uncalibrated (real- aperture) camera*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-6, June 2008.

[68] P. Favaro, S. Osher, S. Soatto and L.A. Vese, *3d shape from anisotropic diffusion*, Proc. of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Vol.1, pp. 179-186, 2003.

[69] H.Y. Lin and K.D. Gu, *Depth recovery using defocus blur at infinity*, 19th International Conference on Pattern Recognition, pp.1-4, 2008.

[70] O. Ghita and P.F. Whelan, *A video-rate sensor based on depth from defocus*, Optics and Laser Technology, Vol. 33, Issue 3, pp.167- 176, 2001.

[71] Li Ma and R.C. Staunton, *Intergration of multiresolution image segmentation and neural networks for object depth recovery*, Pattern Recognition, Vol. 38, Issue 7, pp. 985- 996, 2005.

[72] E.R. Davies, *Fast implementation of generalized median filter*, Electronics Letters, Vol.43, No. 9, pp.505–507, 2007.

[73] R.A. Jarvis, *Focus optimization criteria for computer image processing, Microscope*, Vol. 24, No.2, pp. 163-180, 2nd quarter, 1976.

[74] M. Noguchi and S.K. Nayar, *Microscopic Shape from Focus Using Active Illumination*, Proc. of International Conference on Pattern Recognition (ICPR 94), Vol.1, pp. 147-152, Oct 1994.

[75] S.K. Nayar and Y. Nakagawa, *Shape from Focus*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 16, Issue 8, pp. 824-831, August 1994.

[76] O. Ghita and P.F. Whelan, *Real time 3D estimation using Depth from Defocus*, Proc. of Irish Machine Vision and Image Processing Conference, pp.167-181, 1999.

[77] O. Ghita and P.F. Whelan, *A bin picking system based on depth from defocus* Machine Vision and Applications, Vol.13, No.4, pp. 234 - 244, 2003.

[78] V.P. Namboodiri and S. Chaudhuri, *Shape recovery using stochastic heat flow*, Proc. of British Machine Vision Conference (BMVC), Warwick, UK, Sep 2007,

[79] V.P Namboodiri, S. Chaudhuri and S. Hadap, *Regularized depth from defocus* 15th IEEE International Conference on Image Processing, pp. 1520-1523, 2008.

[80] V.P. Namboodiri and S. Chaudhuri, *On defocus, diffusion and depth estimation*, 4th Indian Conference on Computer Vision, Graphics and Image Processing, Vol. 28, Issue 3, pp. 311-319, 2007.

[81] S.M. Jong, *Depth from Defocus using Radial Basis Function Networks*, Proc. of the 6th International Conference on Machine Learning and Cybernetics, Vol.4, pp. 1888-1893, Aug.2007.

[82] H.C. Andrews and B.R. Hunt, *Digital Image Restoration*, Prentice-Hall, Inc., New Jersey, 1987.

[83] X. Tu, M. Subbarao and Y.S. Kang, *A New Approach to 3D Shape Recovery of Local Planar Surface Patches from Shift-Variant Blurred Images*, 19th International Conference on Pattern Recognition, pp.1-5, Dec. 2008.

[84] M. Subbarao, Y. Kang, S. Dutta and X. Tu , *Localized and Computationally Efficient Approach to Shift-variant Image Deblurring*, 15th IEEE Computer Society's International Conference on Image Processing, pp. 657- 660, San Diego, Oct. 2008.

[85] M. Subbarao, *Rao Transforms*, U.S. Copyright No. TX 6-195-821, June 1, 2005.

[86] C. Swain, A. Peters and K. Kawamura, *Depth estimation from image defocus using fuzzy logic*, Proceedings of the 3rd IEEE Conference on Fuzzy Systems, Vol.1, pp. 94-99, June 1994.

[87] A.N.J. Raj and R.C. Staunton, *Estimation of Image Magnification using Phase Correlation*, International Conference on Computational Intelligence and Multimedia Application, Vol. 3, pp. 490-494, 2007.

[88] Y.Y.Schechner and N.Kiryati, *Depth from Defocus vs. Stereo: How Different Really are They?*, International Journal of Computer Vision, Vol.39,Issue 2, pp. 141-162, 2000.

[89] P.H.W. Leong, *Recent Trends in FPGA Architectures and Applications*, 4th IEEE Symposium on Electronic Design, Test and Applications, Vol.23, Issue 25, pp. 137 - 141, 2008.

[90] Virtex-II Pro / Virtex-II Pro X, Complete Data Sheet (All four modules), http://www.xilinx.com/support/documentation/virtex-ii_pro.htm

[91] EDK-XUP-V2ProPack.zip, http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,400,453&Prod=XUPV2P

[92] M. Subbarao and T. Choi, *An accurate recovery of 3D shape from image focus*, IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 17, Issue 3, pp. 266-274, 1995.

[93] R.A. Jarvis, *A Perspective on Range Finding Techniques for Computer Vision*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 5, No. 2, pp.122-139,1983.

[94] H.H. Baker, *Edge-based Stereo Correlation*, Proc. of DARPA Image Understanding Workshop, pp.168-175, University of Maryland, April 1980.

[95] D. Marr and T. Poggio, *Cooperative Computation of Stereo Disparity*, M.I.T., A.I. Lab., Memo.364, June 1976.

[96] B. Julesz, *Binocular depth perception without familiarity cues*, Science, Vol.145, No. 3630, pp. 356-362, 1964.

[97] D. Marr and T. Poggio, *Computational approaches to image understanding*, M.I.T., A.I. Lab, see also Proc. of Royal Society of London B. Vol. 204, pp. 301-328, 1979.

[98] D. Marr and E.C. Hildreth, *Theory of Edge Detection*, Proc. of Royal Society of London B. Vol. 207, pp. 187-217, 1980.

[99] T.D. Williams, *Depth from camera motion in a real world scene*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 2, No.6, pp. 511-516, Nov 1980.

[100] K. Prazdny, *Motion and structure from optical flow*, Proc. of the 6th International Conference on Artificial Intelligence, pp. 702-704, Tokyo, Japan, 1979.

[101] A. Laurentini, *The visual hull concept for silhouettes-based image understanding*, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.16, No.2, pp.150-162, 1994.

[102] R. Szeliski, *Rapid octree construction from image sequence*, Comput.Vis. Graph. Image Process. (CVGIP), Vol. 58, No. 1, pp. 23–32, 1993.

[103] W. E. Lorensen and H. E. Cline, *Marching cubes: A high resolution 3D surface reconstruction algorithm*, Comput. Graph, Vol. 21, no. 4, pp. 163–169, 1987.

[104] D. Shin, T. Tjahjadi, *Local Hull-Based Surface Construction of Volumetric Data from Silhouettes*, IEEE Transactions on Image Processing, Vol. 17, No. 8, pp.1251-1260, August 2008.

[105] W.Niem and R.Buschmann, *Automatic modelling of 3d natural objects from multiple images*, In European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production, 1994.

[106] B. Mercier and D. Meneveaux, *Shape from silhouette: Image pixels for marching cubes*, 13th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, Vol. 13, pp. 112–118, 2005.

[107] M. Subbarao, T. Yuan, and J. K. Tyan, *Integration of defocus and focus analysis with stereo for 3d shape recovery*, Proc. of SPIE, Vol. 3204, pp. 11-23, Photonics East, Oct. 1997.

[108] A.L. Abbott and N. Ahuja, *Surface reconstruction by dynamic integration of focus, camera vergence, and stereo*, in Proc. of 2nd International Conference on Computer Vision, pp. 532-543, Dec. 1988.

[109] S. A. Fahmy, P. Y. K. Cheung, and W. Luk, *Novel fpga-based implementation of median and weighted median filters for image processing*, International Conference on Field Programmable Logic and Application, pp.142-147, 2005