

TECHNICAL UNIVERSITY OF CRETE  
SCHOOL OF ELECTRONIC AND COMPUTER ENGINEERING  
MICROPROCESSOR AND HARDWARE LABORATORY



MASTER THESIS

DEPTH PERCEPTION FROM A SINGLE CAMERA AND MULTIPLE LIGHT SOURCES

REMATSKA GEORGIA

Examining Committee:

Prof. Apostolos Dollas (Supervisor)

Assoc. Prof. Ioannis Papaefstathiou

Prof. Michael Zervakis

## ABSTRACT

3D Vision has always been a subject of research for many decades. The use of images results in the loss of the 3<sup>rd</sup> dimension. Many techniques have been developed over the years that aim in acquiring depth from images. Among the most widespread are stereo vision, fringe projection, laser scanning or combinations of them, all of which have shown high quality results but unsuitable for some classes of applications. For example, laser scanning rangefinders are not low-cost system. Hence, there is a need to develop a low cost, high speed, high resolution system for 3D vision. In this thesis a novel approach is presented to estimate depth using a single camera combined with two spectrally distinct light sources for road surface measurements. The light sources consist of two sets of LED arrays. Depth information can be extracted, by processing the different reflections in the image from the two light sources. The basic information of depth estimation lies in their blue to red ratio, which with appropriate calibration is correlated to depth. The methodology was first developed in MATLAB and then was implemented on FPGA. The system is real time and fully optimized to have the minimum hardware resources and high frequency of operation for high resolution images.

## ΠΕΡΙΛΗΨΗ

Η 3D όραση υπήρξε αντικείμενο έρευνας για πολλές δεκαετίες. Χρησιμοποιώντας εικόνες η 3<sup>η</sup> διάσταση χάνεται. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλές τεχνικές που ως στόχο έχουν την εύρεση του βάθους μέσα από εικόνες. Τα πιο διαδεδομένα είναι η Στερεοσκοπική όραση(Stereo Vision), Fringe Projection, Laser Scanning ή συνδυασμός τους. Όλες οι τεχνικές αποδίδουν εξαιρετικά για κάποιες εφαρμογές αλλά υστερούν σε άλλες. Για παράδειγμα, το Laser Scanning range finder είναι ένα υψηλού κόστους σύστημα, επομένως υπάρχει η ανάγκη για την ανάπτυξη ενός αντίστοιχου χαμηλού κόστους, υψηλής ταχύτητας και υψηλής ευκρίνειας. Σε αυτή την εργασία παρουσιάζεται μια νέα προσέγγιση για την εκτίμηση του βάθους χρησιμοποιώντας μια κάμερα σε συνδυασμό με δυο φασματικά διακριτές πηγές για την χαρτογράφηση της επιφάνειας του δρόμου. Μπορούμε να εξάγουμε το βάθος με το να επεξεργαστούμε τις διαφορετικές αντανάκλασεις από της δυο πηγές στην εικόνα. Η βασική πληροφορία βρίσκεται στο λόγο μπλε προς κόκκινο των δύο φωτεινών πηγών, ο οποίος με κατάλληλη βαθμονόμηση συσχετίζεται με το βάθος. Η μεθοδολογία αναπτύχθηκε πρώτα στη MATLAB και στη συνέχεια υλοποιήθηκε σε FPGA. Το σύστημα είναι βελτιστοποιημένο ώστε να καταλαμβάνει τους ελάχιστους πόρους και να επιτυγχάνει υψηλή συχνότητα ενώ μπορεί παράλληλα να θεωρηθεί και σύστημα πραγματικού χρόνου.

## ACKNOWLEDGMENTS

I owe my deepest gratitude to my supervisor Prof. Apostolos Dollas whose ideas and help made this thesis possible. I would like to place on record my sincere thank you to Assoc. Prof. Ioannis Papaefstathiou for supporting this idea. I also wish to thank Prof. Michalis Zervakis for accepting reviewing this work and participate in the examination committee. I also thank my parents and friends for their support and attention.

# TABLE OF CONTENTS

ABSTRACT .....	i
ACKNOWLEDGMENTS .....	ii
CHAPTER 1 INTRODUCTION.....	1
1.1 PROBLEMS ADDRESSED IN THE THESIS.....	1
1.2 BASIC IDEAS .....	1
1.3 CONTRIBUTION .....	1
1.4 STRUCTURE.....	2
CHAPTER 2 RELATED WORK .....	3
2.1 STEREO VISION.....	3
2.2 3D SCANNERS .....	5
2.3 FRINGE PROJECTION TECHNIQUE.....	7
2.3.1 PRINCIPLE.....	7
2.3.2 PHASE STEPPING ALGORITHMS .....	8
2.3.3 PHASE AND HEIGHT.....	9
2.3.4 FOURIER TRANSFORM METHOD.....	10
2.4 SYNTHETIC APPERTURE RADAR .....	11
CHAPTER 3 SYSTEM MODELING .....	13
3.1 PROPOSED METHODOLOGY.....	13
3.2 CALIBRATION.....	19
3.2.1 THRESHOLD SELECTION.....	28
3.2.2 VERIFICATION.....	28
CHAPTER 4 SYSTEM DESIGN AND IMPLEMENTATION .....	31
4.1 SYSTEM ARCHITECTURE .....	31
4.2 PERFORMANCE RESULTS.....	36
4.3 HARDWARE RECOURSES .....	37
CHAPTER 5 DESIGN VERIFICATION .....	39
CHAPTER 6 CONCLUSIONS AND FUTURE WORK .....	41

# LIST OF TABLES

Table 3.1 Camera intrinsic parameters .....	28
Table 4.1 The Maximum theoretical throughput of our system .....	36
Table 4.2 Maximal theoretical frequency for different resolutions .....	37
Table 4.3 Hardware recourses of the depth estimation core.....	37
Table 4.4 Hardware recourses of the system. ....	38

# LIST OF FIGURES

Figure 2.1. Stereo geometry of camera system adopted from [17].	4
Figure 2.2. General description of the depth calculation process	5
Figure 2.3 System Configuration adopted from [6]	6
Figure 2.4. Typical setup of a digital fringe projection and phase shifting system adopted from [4].	8
Figure 2.5. Optical geometry of the general configuration adopted from [4].	9
Figure 3.1 System Setup.	13
Figure 3.2. System principle of operation	14
Figure 3.3. Original image	15
Figure 3.4 Ratio of the Blue and Red Images.	15
Figure 3.5 3D mesh of the ratio.	16
Figure 3.6 Top view of the 3D mesh.	16
Figure 3.7 Bottom view of the 3D mesh	17
Figure 3.8 3D mesh where the range of the ratio is between 0-1.	17
Figure 3.9 Normalized view of the 3D mesh	18
Figure 3.10 Reconstruction of the road surface where slope occurs.	18
Figure 3.11 Histogram of intensities of red and blue image	20
Figure 3.12. Histogram of the intensities in the blue channel of the red image	21
Figure 3.13 Histogram of the intensities in the red channel of the blue image	21
Figure 3.14 The scene containing the stripes of white papers shaped with triangles	22
Figure 3.15 .Ratio of blue and red images.	23
Figure 3.16 Top view of the 3D mesh of the ratios	24
Figure 3.17 Bottom view of the 3D mesh of the ratios	24
Figure 3.18 Ratio after applying 9x9 gaussian filter to each color channel.	25
Figure 3.19 Unfiltered Ratio	25
Figure 3.20 Processed Ratio of blue and red images	27
3.21 Measured slope : (a) Blue LEDs (b) Red LEDs	27
Figure 3.22.	29
Figure 3.23 Reconstruction of the triangles	30
Figure 4.1 RGB format	31
Figure 4.2 General Block Diagram	32
Figure 4.3 Lines Buffer	33
Figure 4.4 Gaussian Filter	33
Figure 4.5 Ratio Calculation Component	34
Figure 4.6 Threshold estimation	34
Figure 4.7 Pre Threshold selection Component	34
Figure 4.8 Threshold selection component	35
Figure 4.9 Distance Calculation Module	35
Figure 4.10 System setup	36
Figure 5.1 System setup for verification	39
Figure 5.2 Difference between software hardware	40
Figure 5.3 Difference between software hardware of the triangles	40



# CHAPTER 1

## INTRODUCTION

### 1.1 PROBLEMS ADDRESSED IN THE THESIS

By studying existing methodologies on depth estimation system, we came across some difficulties on applying these methodologies to address our problem on depth estimation of road surfaces. Different approaches were examined as will be analyzed in the following chapter. Stereovision methods are expected to fail as in the case of flat one-color surfaces with little structure such as road the process of finding the corresponding pixels is difficult leading to non accurate disparity estimation. Laser scanners range finders although have been used in road 3D surface modeling, they usually are costly and do not provide depth information with millimeter accuracy. Fringe projection techniques on the other hand, are very promising, but using them on a moving platform error would occur due to vibrations induced by the road. Our approach, although not yet fully developed, provides a low cost system with millimeter accuracy on our depth estimation results.

### 1.2 BASIC IDEAS

In this thesis, we present a novel method for road surface measurements using a single camera combined with tow spectrally distinct light sources, as a lower cost alternative to scanning laser rangefinder apparatuses. The light sources consist of two sets of LED arrays, red in one array and blue in the other. The need for arrays of LED (a single color per array) is due to the large area needed to be covered. The method is based on the different reflectance of each color, depending on the slope of the obstacle (or depression).

Each of the Red and Blue channels of the color image acquired is processed separately. We consider their ratio to be the gradient of the surface providing the necessary information for depth estimation. The basic idea is that surface slopes facing the blue and red LED arrays have larger intensities on each side respectively. The ratio is analogous to the slope of the surface and with the appropriate calibration we have been able to correlate the values observed with the actual distance.

### 1.3 CONTRIBUTION

The method described in this thesis provides a novel approach on acquiring depth information from images. It uses a single camera as opposed to some of the existing methodologies where two or more cameras are used. Furthermore, comparing it to expensive laser scanning range finders this system is extremely low cost and easy to

implement. Results have shown that the proposed methodology can detect depth of one millimeter and above.

## 1.4 STRUCTURE

Chapter 2 provides an overview of the existing methodologies on depth estimation techniques.

In Chapter 3 the methodology used is being analyzed providing the algorithm implementation as well as results acquired in MATLAB.

In Chapter 4 we describe the architecture and the implementation of the system of the hardware design.

In Chapter 5 results between hardware and software are compared and performance results are presented.

Chapter 6 provides the conclusions and future work that extend the current thesis.

# CHAPTER 2

## RELATED WORK

### 2.1 STEREO VISION

Stereo Vision is one of the most active research areas in machine vision. Depth estimation is calculated for every point in the scene relative to the camera. Depth calculation is achieved using two or more cameras, one next to the other, and they provide two displaced images of the same scene. By finding points in the two images that correspond to the same point in the scene, we can extract the distance of that point.

Stereo Vision simulates the human vision, because the eyes of humans are located at different positions on the head which results in two slightly different images projected to the retinas of the eyes. These positional differences known as binocular disparity are processed by the brain to yield depth perception.

Every point in the scene that is visible by both cameras is projected as a pair of pixels in the two images (corresponding pixels). One image is considered as the reference and the other as the non-reference image. The problem lies in finding those pixels, known as the correspondence problem. The difference in the x-coordinates of the corresponding pixels in the two images is in fact the disparity used to construct an accurate depth map of the scene. The resulting depth map is usually an image, where closer points are brighter.

As we have already mentioned, stereovision uses two cameras, one next to the other separated by a distance  $b$  (baseline). The goal is to find all corresponding pixels and from them to calculate disparity, in order for the depth information to be extracted through triangulation. Disparity is directly connected to depth. More specifically, large disparities correspond to objects in the scene closer to the camera, while small disparities correspond to objects in the scene far from the camera.

Over the years many stereovision algorithms have been developed that probe the correspondence problem. A Taxonomy is presented from Scharstein et al. [16]. In general stereo vision algorithms perform a subset of these steps[16]:

- a) Matching cost computation
- b) Cost aggregation
- c) Disparity computation/optimization
- d) Disparity refinement

Most stereo vision algorithms require rectified images, where the epipolar line constraint is satisfied. The basic geometry of the two cameras is presented in Figure 2.1. Images are considered coplanar: there exists no rotation between the cameras. Epipolar plane is the plane that passes through the center of the cameras and the point in the scene. The intersection of epipolar plane with the projection planes of the two images defines the epipolar line. Finding corresponding pixels is restricted to one line and the search space is reduced from 2D to 1D.

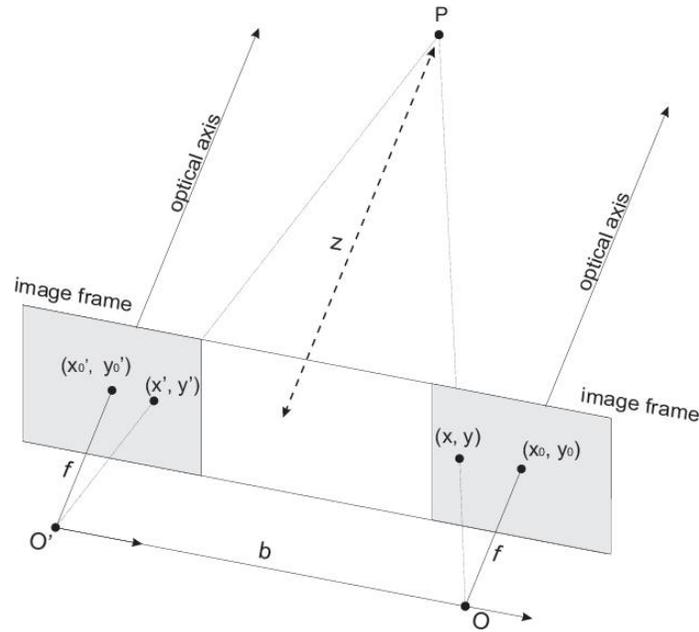


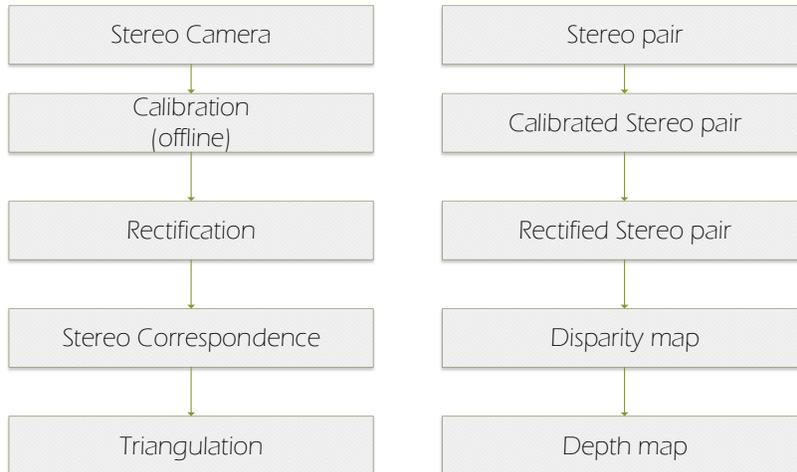
Figure 2.1. Stereo geometry of camera system adopted from [17].

When the epipolar line constraint is not satisfied, spatial transformation are applied on the images to bring them on the same plane. This is known as image rectification and uses the intrinsic and extrinsic parameters obtained from camera calibration. An overview of depth calculation process is presented in Figure. 2.2.

For each point  $(x, y)$ , depth is calculated using equation (2.1) where  $d$  is the disparity calculated from the stereovision algorithm,  $f$  is the focal length of the camera  $b$  is baseline of the stereoscopic camera and  $DP_{hor}$  is the pixel horizontal dot pitch (2.2).

$$Depth = \frac{b * f}{d * DP_{hor}} \quad (2.1)$$

$$DP_{hor} = \frac{Active\ pixel\ array\ area}{Image\ width} \quad (2.2)$$



**Figure 2.2. General description of the depth calculation process**

Applying stereovision techniques for recovering surface height using the correspondence process although in many cases successful, it has many restrictions. Reflection, occlusions, texture-less areas etc. that appear in real world images, makes the task of finding corresponding pixels difficult. Especially in the case of flat one-color surfaces where many pixels in the images have the same value and the image contains too little structure so the disparity cannot be estimated accurately, leading in stereovision algorithms to fail.

## 2.2 3D SCANNERS

In Laser Scanning techniques, 3D reconstruction is based on 3D point clouds provided by laser scanners. Scanners usually emit some kind of light and detect its reflection passing through object in order to calculate the distance. Laser Scanners can be broadly categorized in to Time-Of-Flight (TOF) scanners and triangulation scanners.

In TOF systems distance is usually calculated by measuring the time light propagates through the air and reflected back. The laser range finder finds the distance from the surface by calculating the time a pulse of light needs to travel from the laser to an object and return back to a laser detector. The speed of the pulse of light is known ( $c = 3 \cdot 10^8 \text{m/s}$ ) so the distance is simply calculated by multiplying the speed with half the time measured. If  $t$  is the time measured by the laser scanner then the distance is simply calculated as  $c \cdot t / 2$ . The time measured includes the double distance from the laser to the object and for that reason only half the time measured is considered.

Precision in time measured affects the accuracy of the distance calculated. If for example 1 mm distance is consider then the TOF system must be able to measure a time of approximately 3.3 picoseconds. Standard resolution is usually in the order of 0.5 to 1 cm. Other than pulse TOF different methods have been proposed, improving range resolution that includes amplitude modulation, frequency modulation, hybrid detection

etc. Range is usually measured from the phase variation between the transmitted and received signal [5].

TOF scanners scan only in one direction and detect only one point in this direction. By obtaining a slight change in its direction the entire field of view can be scanned. Recent TOF 3D laser scanners can obtain up to 100,000 points every second.

TOF 3D scanners are only suitable for large structures and are preferred choice for long range measurements. For high precision millimeter measurements this technique is not appropriate, due to the high speed of light and the picoseconds resolution needed in the time measured. Furthermore, TOF scanners require stable platforms for both the scanner and the object and not suitable for incorporating them on a moving platform.

Triangulation based 3D laser scanners are hybrid systems that use digital cameras and infrared lasers. Projecting a laser beam on the object in the scene and capturing the object from a digital camera we can extract distance through triangulation. In this system distance is measured by the triangle formed between the laser beam, the laser source and the camera. Knowing the distance between the digital camera and the laser source (baseline) and the angle between the laser beam and the baseline we can determine the position of laser beam over the object. System Configuration is shown in Figure 2.3.

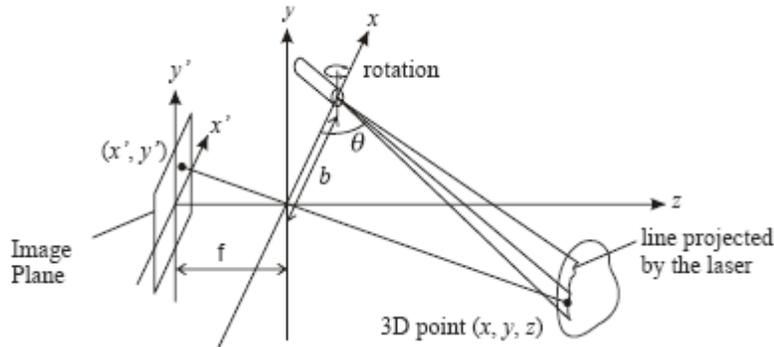


Figure 2.3 System Configuration adopted from [6]

Equations (2.3), (2.4) and (2.5) [6] provide the 3D points coordinates (x,y,z) in real space.

$$x = \frac{bx'}{f \cot \theta - x'} \quad (2.3)$$

$$y = \frac{by'}{f \cot \theta - x'} \quad (2.4)$$

$$z = \frac{bf}{f \cot \theta - x'} \quad (2.5)$$

Where  $x'$ ,  $y'$  are the image pixels,  $b$  is the baseline,  $\theta$  is the angel between the baseline and  $f$  is the focal length [6].

In contrast to TOF 3D scanners triangulation scanners offer very high precision especially in close range measurements and have a small range on the order of meters. Although they offer high precision calculation they are limited due to their high equipment cost.

Range finder scanners described above falls in the non-contact active category of digitally acquiring the 3D shape of an object. They have numerous applications in different fields that include shape acquisition, industrial design, prototyping, computer vision, inspection, robot navigation, medical diagnosis and more. There also other active range image sensor. Besl provides an overview of the capabilities of different sensors [7]. While Blais et al provides a good overview of different laser scanning techniques[5]. Duchoň et al provides an application of laser rangefinders in mobile robotics along with some technique used for environmental mapping and navigation [8].

3D scanning techniques can also be used in road inspection, pavement assessment and road 3D surface modeling. Li et al. provide a real time inspection system for detecting road distress features such as rutting and shoving using structured light triangulation [9]. While Pu et al provides a framework for structure recognition using mobile laser scanning data. Using dense point clouds obtained from mobile laser scanning the data are classified into more detailed classes [10]. Yu et al provide 3D model generation of road surfaces by incorporating 3D laser-range sensors. The 3D models obtained display depth information and geometric details of cracks of a few centimeters wide.

## 2.3 FRINGE PROJECTION TECHNIQUE

### 2.3.1 PRINCIPLE

In Fringe projection techniques a projector is used to project structured patterns, which are then recorded by a camera. Structured patterns are composed by sinusoidal stripes and are called fringe images.

A typical setup of a fringe projection system is shown Figure 2.4. A projector projects vertically or horizontally straight fringe stripes onto the object. Due to objects surface geometry the recorded image is phase modulated. Analyzing the image we can calculate the phase modulation. Knowledge of the correspondence between the camera and the projector is required to retrieve the 3-D information through triangulation[4]. Several fringe analysis methods have been developed for calculating the phase modulation. These methods can be broadly categorized as spatial and temporal. Spatial methods require a sufficiently high frequency spatial carrier while the temporal methods require the acquisition of a number of images by projecting phase-shifted fringe patterns. The most common fringe analysis methods are phase-shifting and Fourier transform.

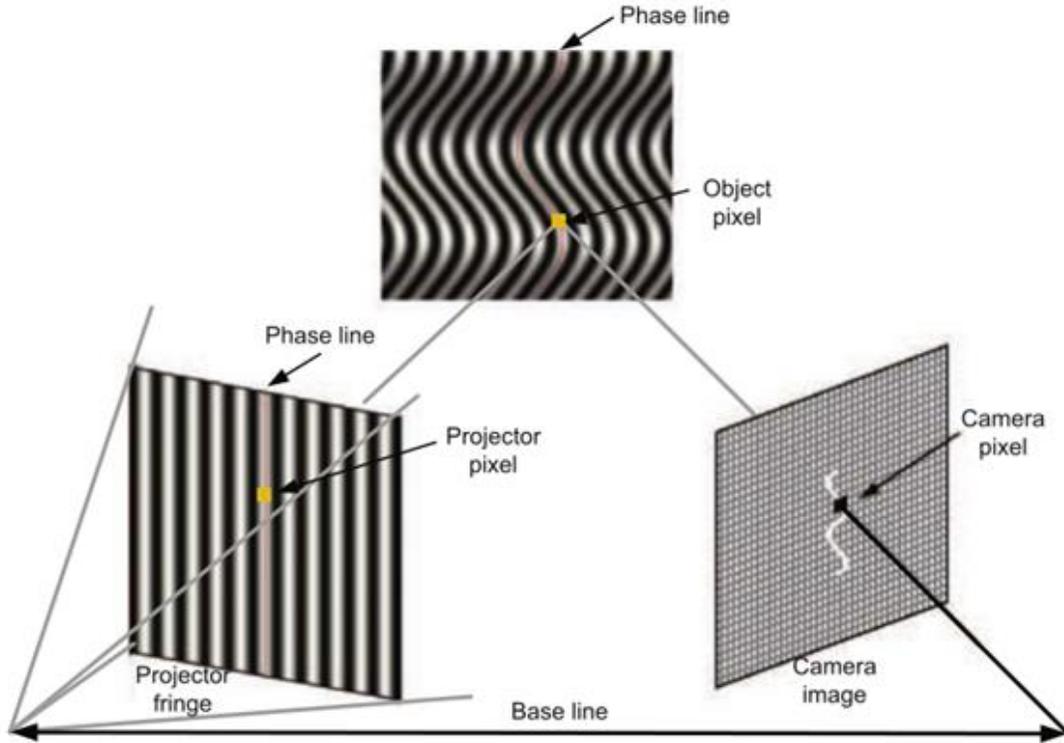


Figure 2.4. Typical setup of a digital fringe projection and phase shifting system adopted from [4].

### 2.3.2 PHASE STEPPING ALGORITHMS

Phase stepping method is a popular technique of temporal Phase Measurement Interferometry (PMI). In these techniques the phase is altered in steps between intensity measurements. A general expression of an interferogram intensity is written as

$$I(x, y) = a(x, y) + b(x, y) \cos(\varphi(x, y) + \alpha) \quad (2.6)$$

here  $a(x, y)$  is the mean intensity,  $b(x, y)$  is the intensity modulation,  $\varphi$  is the difference between the interfering waves and  $\alpha$  is a modulating phase which is controlled experimentally. Equation 2.6 requires a minimum of three intensity measurements to determine the phase as it contains three unknowns. Introducing phase shift  $\alpha_1$ ,  $\alpha_2$  and  $\alpha_3$  we get:

$$I_1 = a + b \cos(\varphi + \alpha_1)$$

$$I_2 = a + b \cos(\varphi + \alpha_2)$$

$$I_3 = a + b \cos(\varphi + \alpha_3)$$

$I_1$ ,  $\alpha$ ,  $b$  and  $\varphi$  are functions of the spatial coordinates.  $\varphi$  is then calculated as.

$$\varphi = \tan^{-1} \frac{(I_2 - I_3) \cos \alpha_1 - (I_1 - I_3) \cos \alpha_2 - (I_1 - I_2) \cos \alpha_3}{(I_2 - I_3) \sin \alpha_1 - (I_1 - I_3) \sin \alpha_2 - (I_1 - I_2) \sin \alpha_3}$$

By choosing a three step phase shifting algorithm and a phase shift of  $2\pi/3$  as stated by Zhang the system is less sensitive to the nonlinearity errors. The phase is then obtained as

$$\varphi(x, y) = \tan^{-1} \frac{\sqrt{3}(I_1 - I_3)}{2I_2 - I_1 - I_3} \quad (2.7)$$

Phase values of equation 2.7 are limited to range  $[-\pi, \pi]$ . A continuous phase map is obtained by using a phase unwrapping algorithm.

### 2.3.3 PHASE AND HEIGHT

The geometry of a general configuration of an imaging system is shown Figure 2.5.  $P_{1,2}$  and  $I_{1,2}$  are the centers of the entrance and exit pupils of the projection and imaging optics, respectively. The intensity variation along  $x$  on the reference plane can be described by the equation

$$I_R(x, y) = A(x, y) + B(x, y)\cos\varphi(x, y)$$

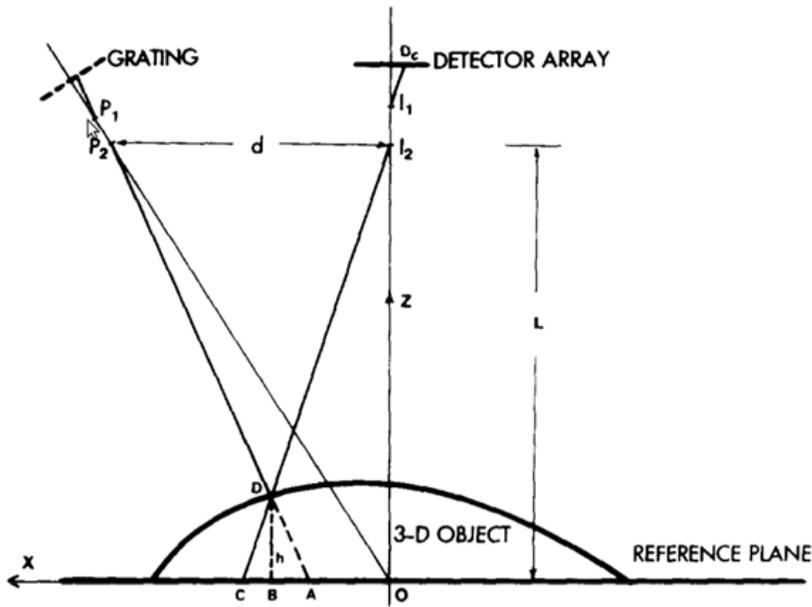


Figure 2.5. Optical geometry of the general configuration adopted from [4].

Every point on the reference plane is characterized by a unique phase value with respect to a reference point such as O. The reference plane as well as the object is recorded by a camera. Point D on the object surface image lies on the same pixel as point C on the reference plane. The phase  $\varphi_D$  on the object surface has the same value as  $\varphi_A$  on the reference plane. We can express  $\varphi_{CD}$  as  $\varphi_{CD} = \varphi_{CA} = \varphi_C - \varphi_A$ .

Due to similar triangles  $P_2DI_2$  and  $ADC$  and because generally  $AC \ll D$  we get

$$Z = h(x, y) = AC(L/d)(2.8)$$

The phase difference  $\varphi_{CD}$  is related to the geometrical distance:

$$AC = (p_0/2\pi)\varphi_{CD}(2.9)$$

Where  $p_0$  is the period of the projected sinusoidal pattern. Combining equations (2.8) and (2.9) we get

$$h(x, y) = \frac{p_0 L}{2\pi d} \varphi_{CD}(2.10).$$

$\varphi_{CD}$  is the phase difference  $\varphi(x, y)$  [4].

### 2.3.4 FOURIER TRANSFORM METHOD

As introduced by Takeda et al. the deformed pattern observed through a CCD camera can be expressed as

$$g(x, y) = r(x, y) \sum_{n=-\infty}^{\infty} A_n e^{2\pi i n f_0 x + n \varphi(x, y)} (2.11)$$

Equation (2.11) can be written as

$$g(x, y) = \sum_{n=-\infty}^{\infty} q_n(x, y) e^{2\pi i n f_0 x} (2.12)$$

Where

$$q_n(x, y) = A_n r(x, y) e^{i n \varphi(x, y)} (2.13)$$

Computing 1-D Fourier Transform of equation (2.13) with respect only to x:

$$G(f, y) = \sum_{n=-\infty}^{\infty} Q_n(f - n f_0, y) (2.14)$$

All the spectra  $Q_n(f - n f_0, y)$  are separated from each other by the carrier frequency  $f_0$ .  $Q_n(f - f_0, y)$  is selected and inverse Fourier Transform is calculated.

$$\hat{g}(x, y) = q_1(x, y) e^{2\pi i f_0 x} = A_1 r(x, y) e^{i[2\pi f_0 x + \varphi(x, y)]} (2.15)$$

When  $h(x, y)=0$  the deformed image  $\hat{g}(x, y)$  is obtained as

$$\hat{g}_0(x, y) = A_1 r(x, y) e^{i[2\pi f_0 x + \varphi_0(x)]} (2.16)$$

Combining equation (2.15) and (2.16) we get

$$\hat{g}(x, y) \hat{g}_0^*(x, y) = |A_1|^2 r(x, y) e^{i \Delta \varphi(x, y)} (2.17)$$

Where

$$\Delta \varphi(x, y) = \varphi(x, y) - \varphi_0(x) (2.18)$$

Computing complex logarithm of Equation (2.17) we get

$$\log(\hat{g}(x, y)\widehat{g}_0^*(x, y)) = \log[|A_1|^2 r(x, y)] + i\Delta\varphi(x, y)$$

$\Delta\varphi(x, y)$  can be obtained by

$$\Delta\varphi(x, y) = \Im\log(\hat{g}(x, y)\widehat{g}_0^*(x, y))$$

Height in a perfect system is then calculated with the formula similar to Equation 2.10

$$h(x, y) = \frac{L\Delta\varphi(x, y)}{\Delta\varphi(x, y) - 2\pi fd} [2.19]$$

The recovered phase  $\Delta\varphi(x, y)$  is mathematically limited to interval  $[-\pi, \pi]$ , consequently the recovered phase contains artificial discontinuities since it is wrapped into this range. These discontinuities are removed by determining the unknown integral multiple of  $2\pi$  to be added at each pixel of the wrapped phase map to make it continues. By comparing the phase at neighboring pixels and adding or subtracting  $2\pi$  brings the relative phase between two pixels into the range of  $-\pi$  to  $\pi$  [13].

Speed limitations in fringe projection techniques are introduced by many reasons. One of the main reasons is that the exposure time of the camera is limited by the projector's projection time. The exposure time should be larger than the single channel projection time [1]. Thus, to reach high 3D shape measurements, the sinusoidal patterns must be captured in a short period of time. Gong et al. introduced a fast digital fringe projection system by defocusing the projector that projected a binary sinusoidal pattern. Although the technique achieved up to 4,000 Hz rate the accuracy is low followed by small measurement range [1]. Other approaches include using an infrared fringe projection system to acquire 3D shape and also capture natural images to acquire texture information [3]. Brauer-Burchardt et al. developed a hybrid system that uses fringe images for finding point correspondences between a stereo image pair. Using stereovision techniques allowed the reduction of fringe sequence length which was also reduced by the omission of the Gray Code sequence. The technique, although is accurate, is limited by the speed of recording the fringe sequence [2].

Fourier method on the other hand reaches a fast 3D measurement rate because it only requires a single fringe pattern, although is not without limitations. Aliasing influence usually the precision of FTP and correct reconstruction cannot be obtained especially in the case of small variations. Furthermore, it will fail to detect depth, due to misalignment of the camera and the projector that would occur from the vibrations induced if ware to be used on a moving platform.

## 2.4 SYNTHETIC APPERTURE RADAR

Synthetic aperture radar (SAR) produces a two-dimensional [2-D] or three-dimensional [3-D] images. The first dimension is a measure of distance from the radar to the target and is called range while the other is perpendicular to range is called azimuth. Range depends from the time required when transmitting a pulse and receiving an echo from a target. The pulse width determines the range resolution. The narrower the pulse the finer range resolution becomes.

Obtaining fine azimuth requires the use of large physical antennas. Large antennas focus the transmitted and received energy into a sharp beam which defines resolution of the azimuth. Since the size of the antenna required is too large to be place on an aircraft, SAR use the motion of SAR antenna over a region. Data can be collected while flying a distance and processed as if it came from a physical antenna. The distance the aircraft flies is known as the synthetic aperture.

For 3D images Tactical Synthetic Aperture Radar (TSAR) is used, which obtains 3D images either by using two antennas where the time delay between radar signals received from them are processed with new techniques, or by flying a region of interest with two offset passes. A detailed description of the theory of operation of SAR is analyzed on [27].

# CHAPTER 3

## SYSTEM MODELING

### 3.1 PROPOSED METHODOLOGY

In this approach, a scene is illuminated by the distinct light sources on each side of a solid-state color camera, which acquires the images, as shown in Figure 3.1. In Figure 3.2 the principles of operation of the system are shown: the slopes “facing” the blue source have a predominantly blue reflection component, the slopes facing the red source have a predominantly red reflection component, whereas “flat” areas have both spectral components and are shown in the diagram as purple. Although in principle the target object might be anisotropic, meaning that the direction of the reflections is not a linear function of the angle of incidence, in practice we have not observed such behavior in laboratory testing of the system on sections of actual road surfaces.

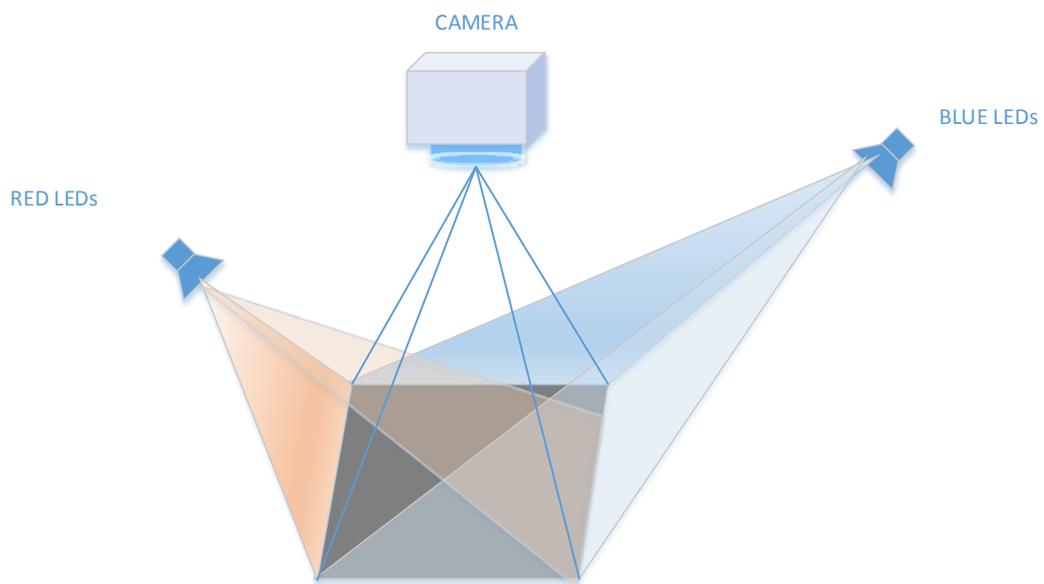


Figure 3.1 System Setup

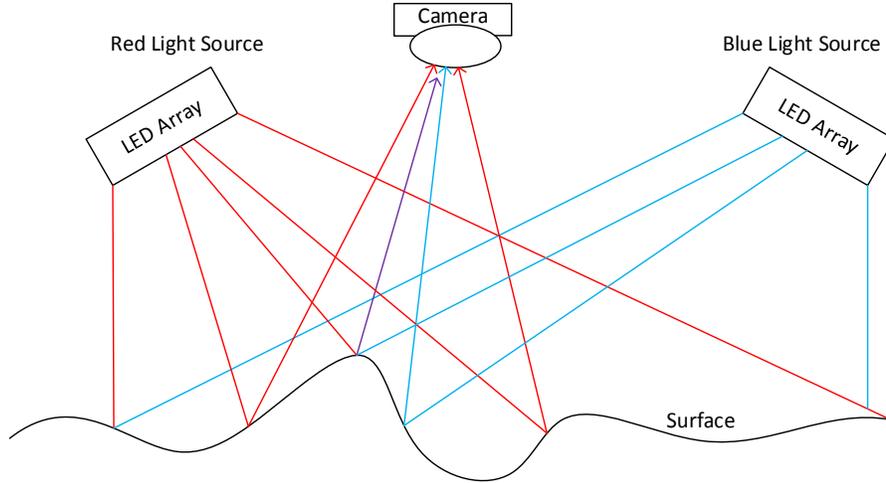


Figure 3.2. System principle of operation

The apparatus has a baffle to shield it from ambient light; however, minor noise would be tolerable in the system, as sunlight would add a DC level to both the blue and the red components of the scene illumination. The acquired image is in RGB form, and very conveniently, the green band (which spectrally is between the red and the blue) is discarded right off and not processed. Each of the R and the B channels is processed separately. The R and B intensities are acquired for each point  $(x, y)$  of the image. Intensities at each channel are multiplied by a normalization factor  $I_{bn}(x, y)$  (3.1) and  $I_m(x, y)$  (3.2) for the blue and the red channel respectively, to account for the attenuation of the light from the LED source at each point. Equations (3.1) and (3.2) will be presented in detail in the calibration section. The result is expressed as:

$$I_1(x, y) = I_R(x, y) \cdot I_m(x, y) \quad (3.1)$$

$$I_2(x, y) = I_B(x, y) \cdot I_{bn}(x, y) \quad (3.2)$$

By obtaining the ratio  $I_1 / I_2$  or  $I_2 / I_1$  we have a quantitative approach of each LED source's contribution to each pixel. The information obtained by this procedure is that a ratio greater than one means that the surface has a slope and it is facing the blue LED whereas a ratio of less than one means that the surface has also a slope in the opposite direction and it is facing the RED LED source. In effect the ratios  $I_1 / I_2$  and  $I_2 / I_1$  are the gradients of the surface.

In our first experiments the scene consists of a piece road-like material, curved, with symbols and a crack. Figure 3.3 shows the original image while Figure 3.4 shows the blue to red ratio obtained by the methodology described above (after normalization); in this figure the ratio is displayed as an image. Figure 3.5 shows a 3D mesh of the ratio.

original image



Figure 3.3. Original image

blue /red ratio



Figure 3.4 Ratio of the Blue and Red Images

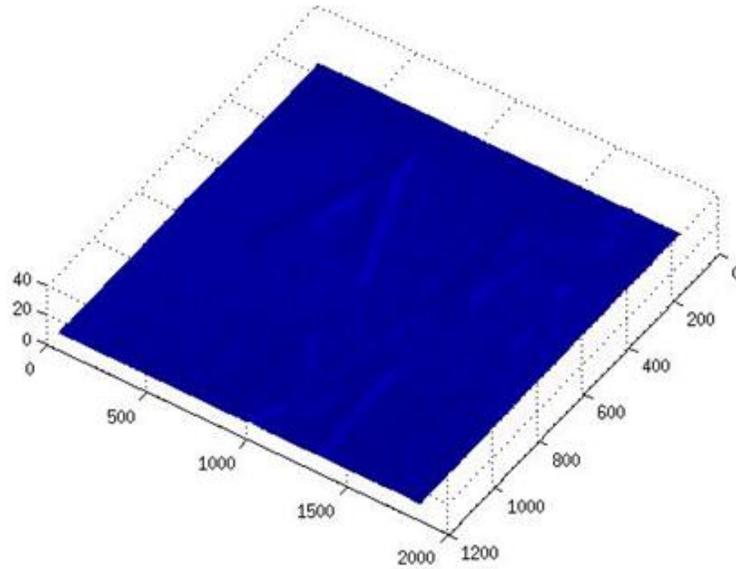


Figure 3.5 3D mesh of the ratio

Figures 3-13 have various 3D meshes from the image, to demonstrate that the 2-3mm indentations of the original tile are discernible, with minimal processing.

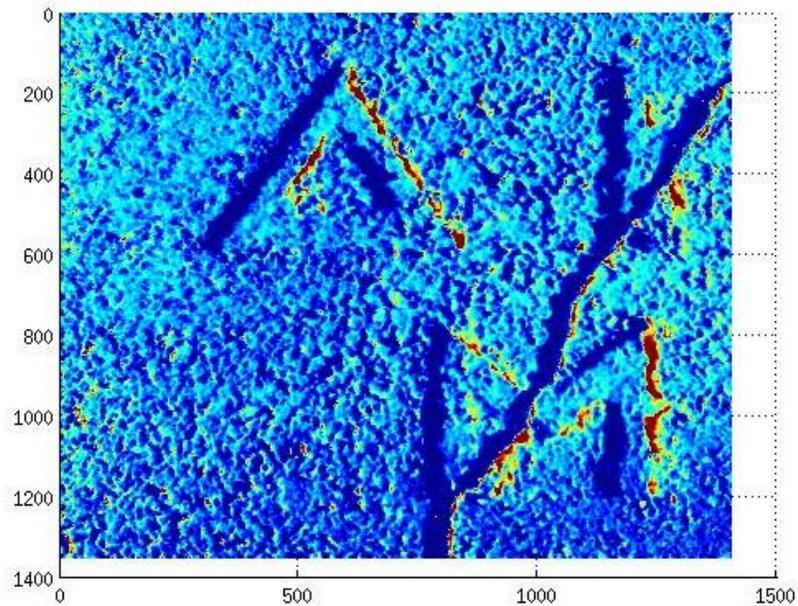


Figure 3.6 Top view of the 3D mesh

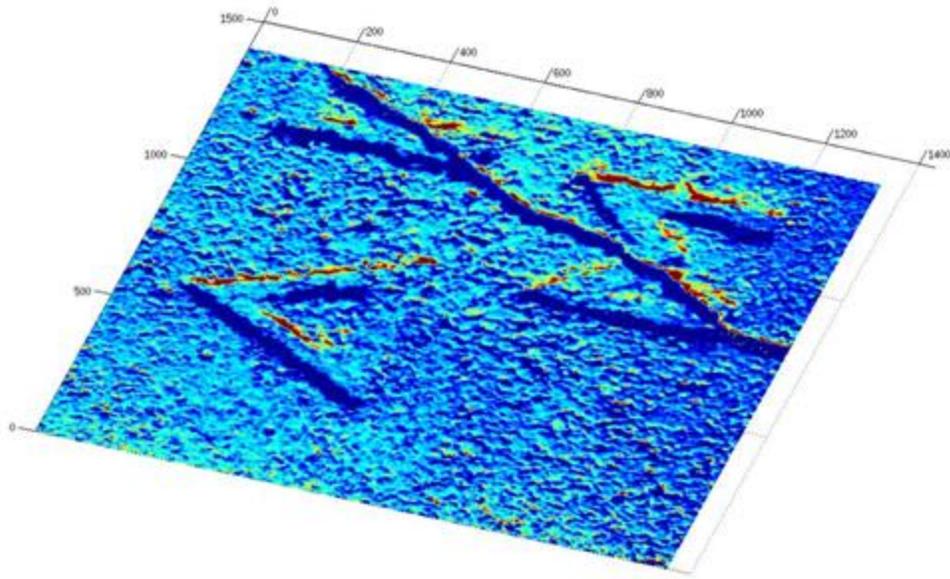


Figure 3.7 Bottom view of the 3D mesh

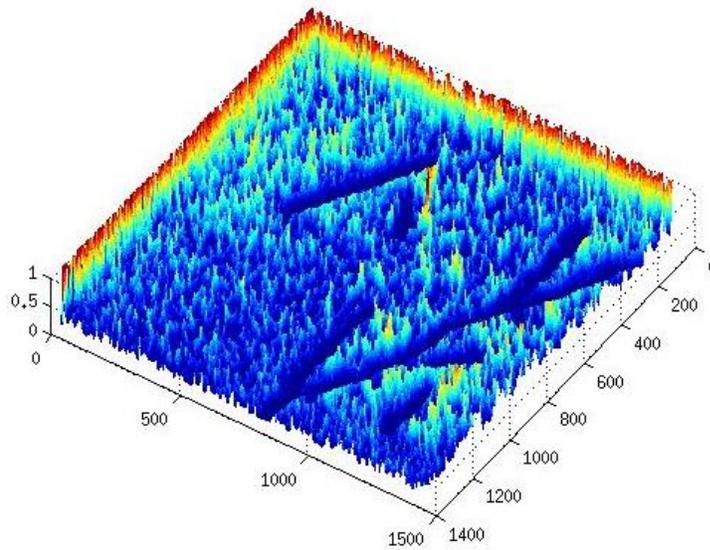


Figure 3.8 3D mesh where the range of the ratio is between 0-1.

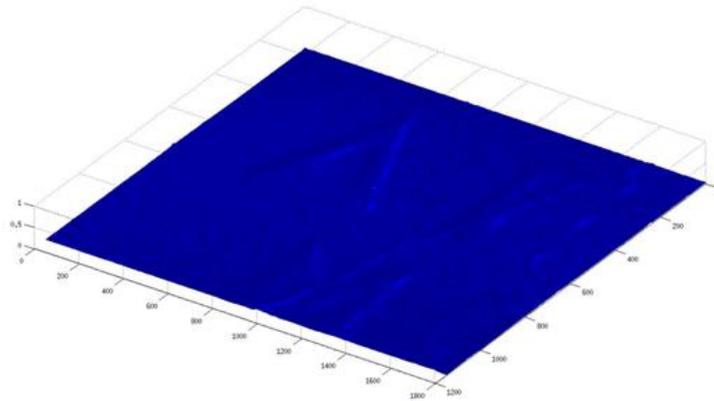


Figure 3.9 Normalized view of the 3D mesh

As mentioned when no slope occurs ratio  $I_2 / I_1$  is one. This is true theoretically but experiments showed that this ratio is close to one and not exactly one. Consequently, a threshold is defined for deciding the slope occurrence. The value of the threshold is explained later in the calibration section. Each line of the ratio is processed separately and the mean value of each line is selected to denote the flat areas and the threshold for slope occurrence all of which are also explained in the calibration section. Detailed z axis information are obtained based on the value of the absolute difference between the flat areas and the ratio. This value after subjected to labeling is assigned to a real slope from which we obtain the distance z. Figure 3.5 shows the road surface of interest with a small section of distress shown enlarged with the actual distance of the areas where slope occurs as obtained from our technique.

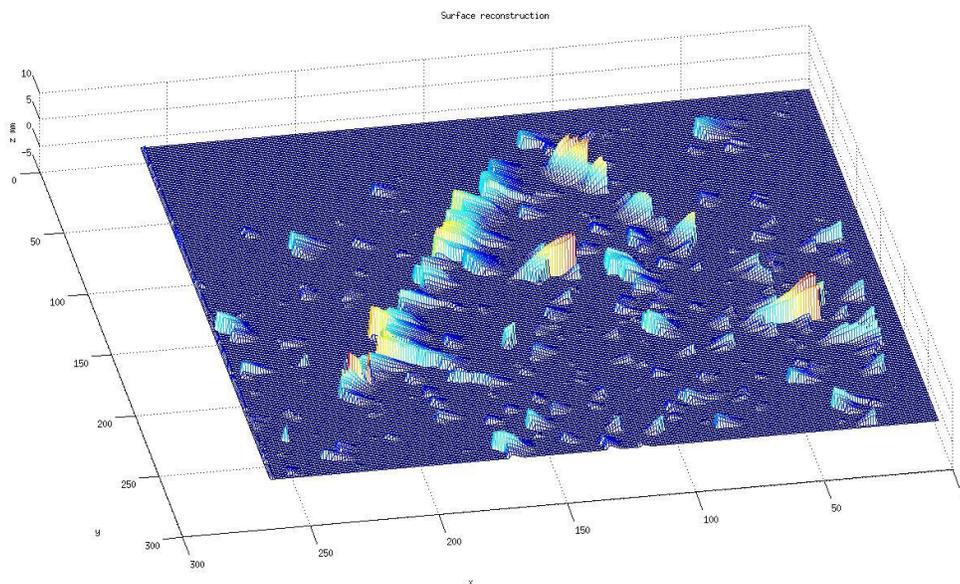


Figure 3.10 Reconstruction of the road surface where slope occurs.

The areas where measured in several points using a caliper and the results obtained are approximately the same.

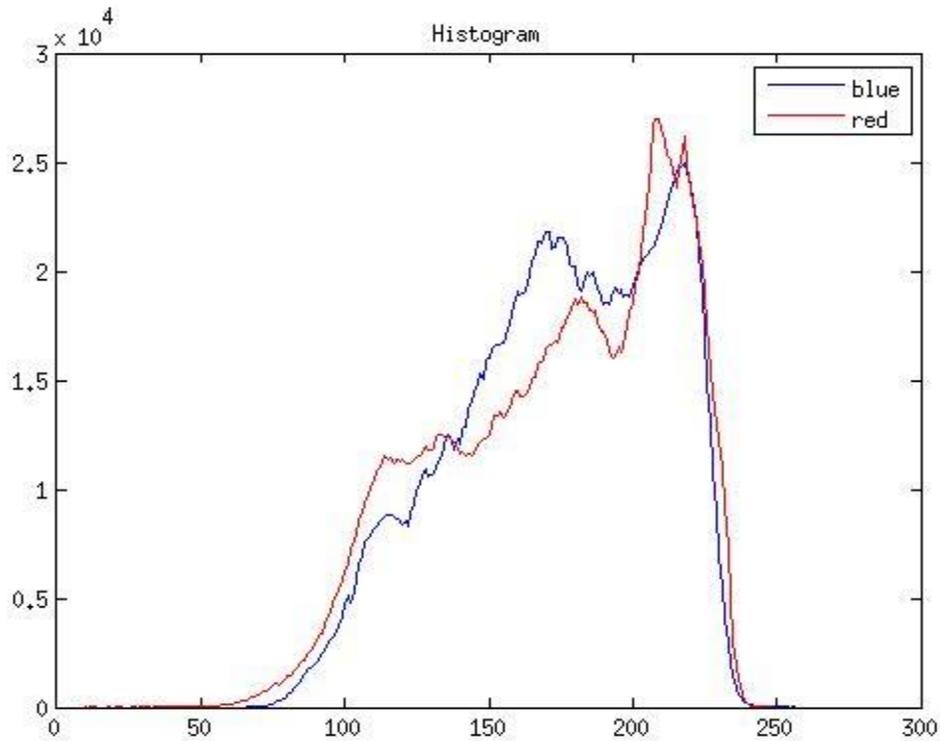
### 3.2 CALIBRATION

Our first requirement is that luminance of both red and blue LED sources should be approximately the same. In order to do that we placed a potentiometer at the blue LEDs and we adjusted it manually. A scene with a white paper placed in front of the camera was then photographed. The scene was illuminated first only by the blue LEDs and then only by the red LEDs. The histograms of both images were compared and the potentiometer was adjusted in order to minimize the difference. We will refer to these images as blue and red image. Figure 3.5 shows the two histograms that were obtained. Compared to the red LEDs, blue LEDs had to be placed at a larger distance, due to the fact that they have a narrower illumination angle (smaller numerical aperture). Intensities of both blue and red image are then normalized at each pixel. We define a normalization factor  $I_m(x, y)$  and  $I_{bn}(x, y)$  such that

$$I_m(x, y) \cdot I_{red}(x, y) = 1$$

$$I_{bn}(x, y) \cdot I_{blue}(x, y) = 1$$

Where  $I_{red}$  and  $I_{blue}$  are the red and blue image respectively. This procedure offers an overview of how the intensities of the LEDs are distributed, the X axis has pixel values and the Y axis has the number of pixels having that value.



**Figure 3.11 Histogram of intensities of red and blue image**

Our method first took place in a controlled environment and results are shown that obtained from MATLAB. The photographs were taken in a dark room so that only blue and red LEDs were illuminating the scene. The histogram of the red channel of the blue image and the blue channel of the red image (i.e. the noise that each light source produces to the other source's results), shown in Figure 3.12 and Figure 3.13 show that the noise in the system due to the LED sources is minimal.

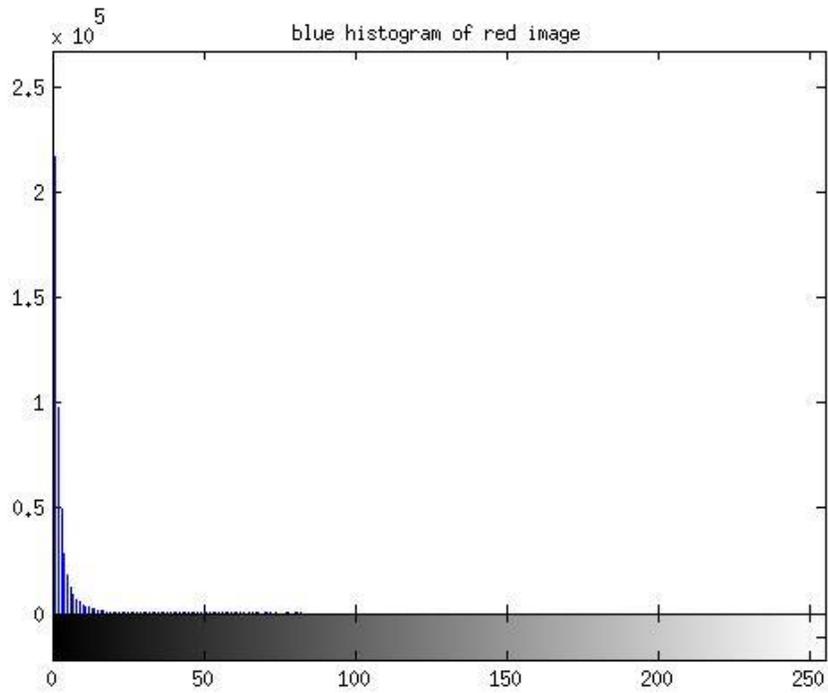


Figure 3.12. Histogram of the intensities in the blue channel of the red image

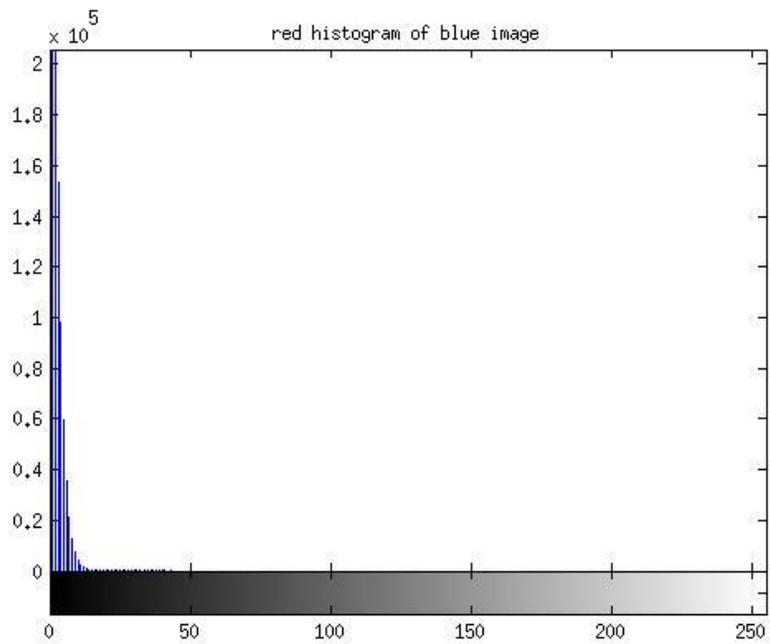
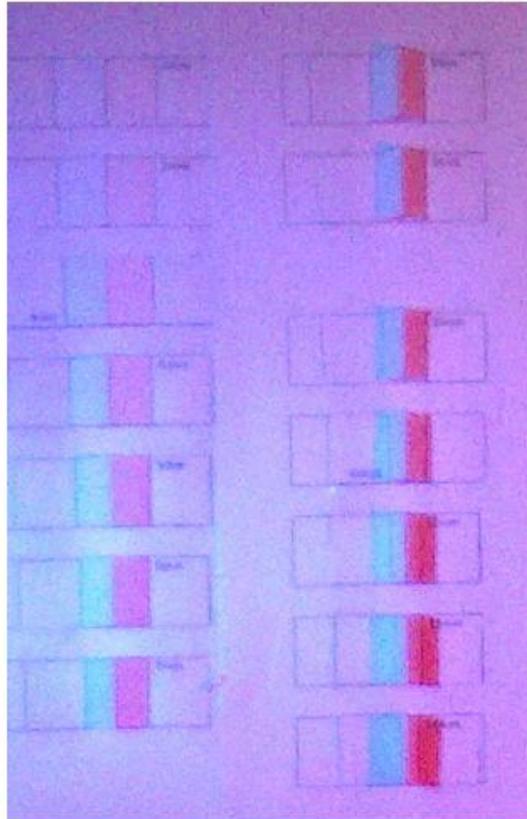


Figure 3.13 Histogram of the intensities in the red channel of the blue image

A new scene is constructed that contains stripes of white paper. Each stripe is formed to shape two triangles of different size. Each side of the triangle in all stripes has a different slope. In all cases the height and the width of the triangle is measured using a calliper. The scene was then illuminated by the blue and the red LEDs. The image of the scene illuminated by the LED arrays shown in Figure 3.14 show that the different sides of the triangles are discernible and the intensities on each side are getting stronger as the height of the triangle and consequently the slope becomes larger.



**Figure 3.14** The scene containing the stripes of white papers shaped with triangles

Intensities at B and R channel are multiplied by the normalization factor  $I_{bn}(x, y)$  and  $I_m(x, y)$  respectively. The result is expressed as:

$$I_{n1}(x, y) = I_R(x, y) \cdot I_m(x, y)$$

$$I_{n2}(x, y) = I_B(x, y) \cdot I_{bn}(x, y)$$

The gradient of the image is considered as the ratio of blue to red image  $I_{n2}/I_{n1}$ . In Figure 3.15 the ratio  $I_{n2}/I_{n1}$  is displayed as an image. We denote the ratio of each line in the image to be

$$\lambda_i(y) = \frac{I_{n2}(i,y)}{I_{n1}(i,y)} \quad 0 < i < H - 1, 0 < y < W - 1$$

Where H and W are the image height and width respectively, of an image of WxH resolution. In Figure 3.16 and 3.17 a 3D mesh of the ratios is shown. As we can see the areas with slope facing the blue LEDs have a ratio greater than one, while the areas facing the red LEDs have a ratio less than one. Instead of depending on a single pixel to explain the behavior of different ratios observed, we filtered each channel with a gaussian smoothing kernel. Specifically we applied a 5x5 gaussian filter. Figure 3.16 and 3.17 shows the 3D mesh of the ratios obtained after applying a 5x5 filter and Figure 3.18 shows the 3D mesh of the ratios obtained after applying a 9x9 filter. Higher order filters have no effect on the observed values of the ratio, therefore they were discarded. Figure 3.19 shows the ratio without applying any filters.



Figure 3.15 .Ratio of blue and red images

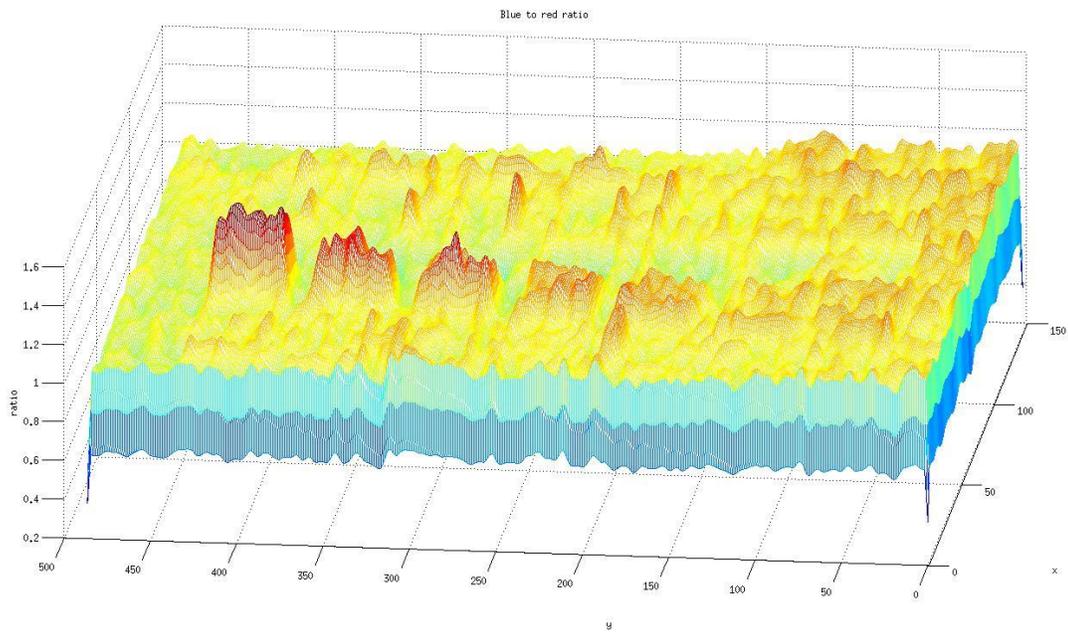


Figure 3.16 Top view of the 3D mesh of the ratios

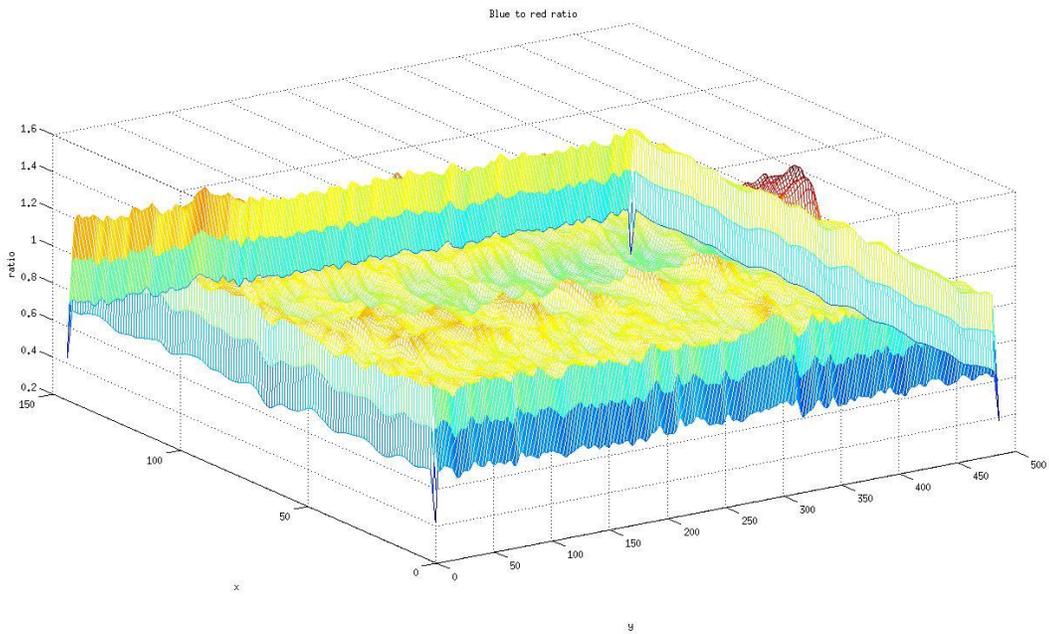


Figure 3.17 Bottom view of the 3D mesh of the ratios

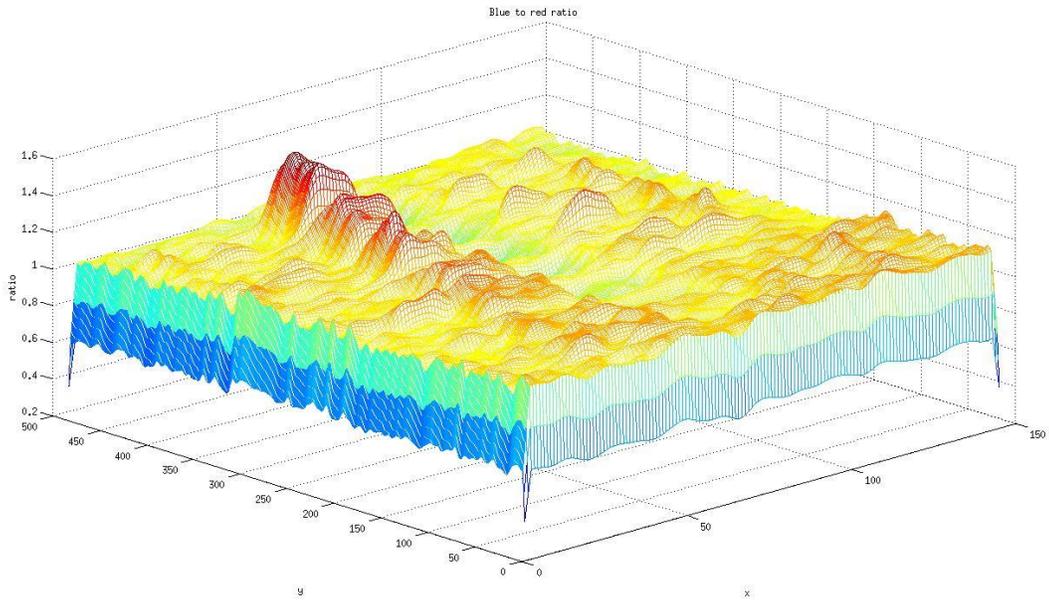


Figure 3.18 Ratio after applying 9x9 gaussian filter to each color channel.

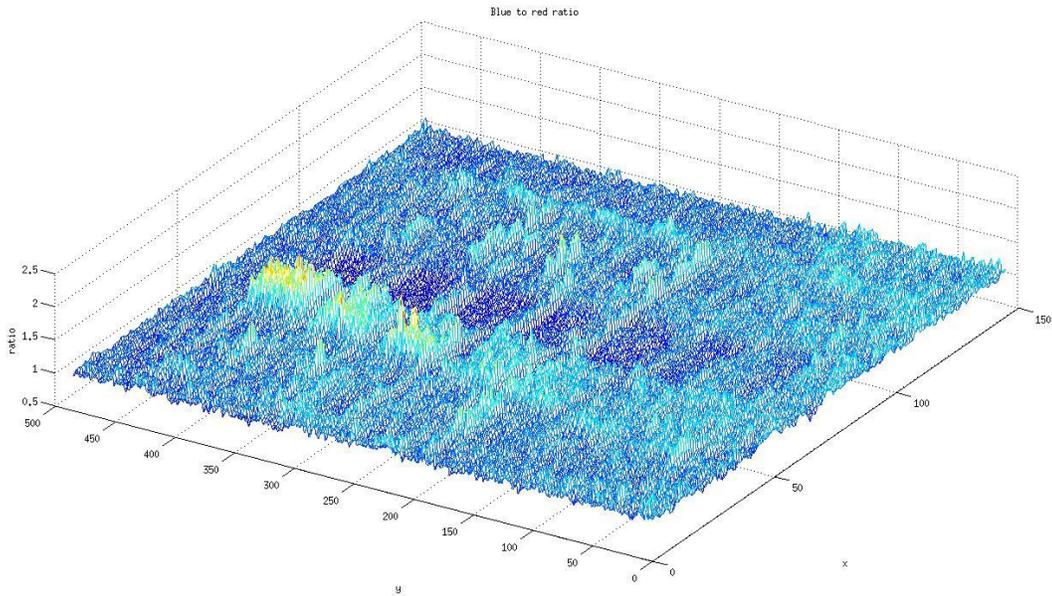


Figure 3.19 Unfiltered Ratio

Slope is calculated by taking the absolute difference between the flat areas and the ratio where slope occurs. Flat areas are calculated for each line by taking the mean value of the line. But, since we can have areas where slope occurs only in one side either the left or the right by taking the mean value for the whole line could lead to a false value representation of the flat areas. To account for this problem we estimate the flat areas

by taking the mean value of specific values of the ratio by not considering large variations of the ratio. So we select a loose value of a threshold  $\tau_1$ , that the values of a line must not exceed, to be calculated in the mean value. More specifically we start by taking the mean value of the line and we add and subtract  $\tau_1$  as formed in equation 3.3.

$$mean(\lambda(i)) - \tau_1 < \lambda(i) < mean(\lambda(i)) + \tau_1 \quad (3.3)$$

Values that lies between these values as shown in equation 3.3 are included in the mean value.

We then calculate the final mean value by adding and subtracting a small value  $l$  to mean value, to define two thresholds  $\tau_3$  (3.4) and  $\tau_4$ (3.5). Values that lie between those threshold are used in calculating the final mean value of each line.

$$\tau_3 = mean\ value - l \quad (3.4)$$

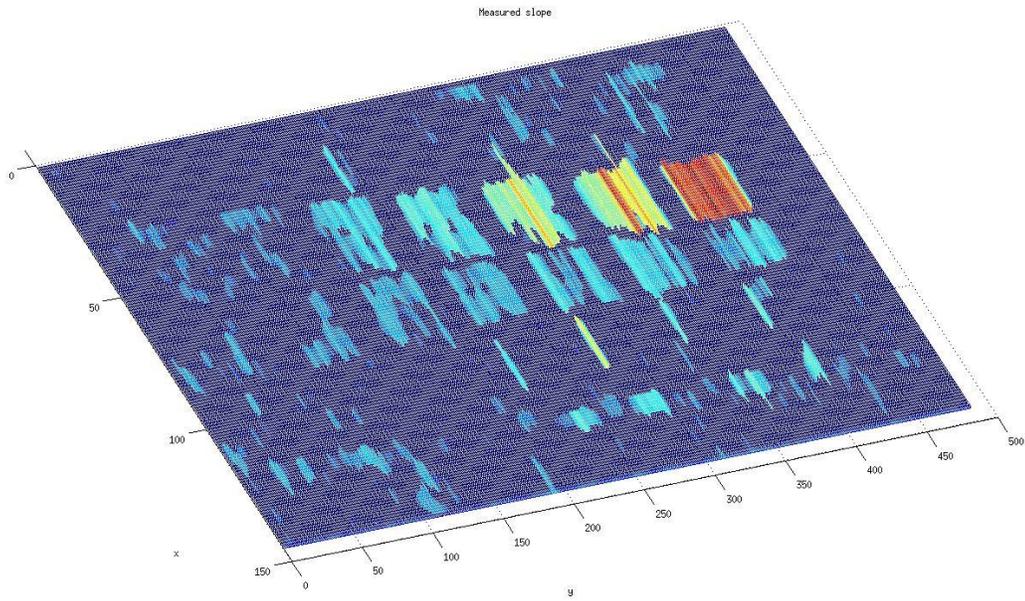
$$\tau_4 = mean\ value + l \quad (3.5)$$

In order to detect the slope occurrence we set a small value  $\epsilon$  that is added and subtracted to final mean value to define two thresholds  $\tau_l$  (3.4) and  $\tau_u$  (3.5) The first threshold is used for detecting slope of the areas that are facing the red LED array, while the second threshold is used for detecting slope of the areas that are facing the blue LED array. If slope detected then we, as already mentioned we calculate the absolute difference.

$$\tau_l = final\ mean\ value - e \quad (3.4)$$

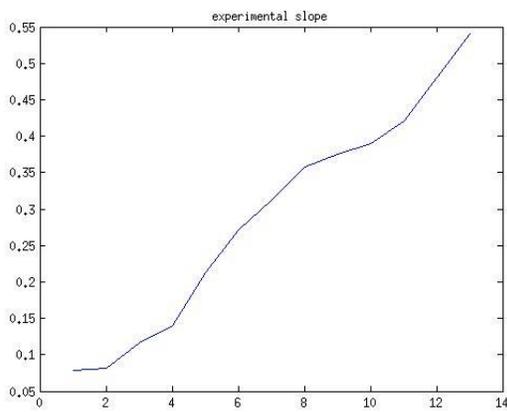
$$\tau_u = final\ mean\ value + e \quad (3.5)$$

Processing  $\lambda$  and setting flat areas to 0 we obtained the slope only where it occurs as shown in Figure 3.20. Difficulty in constructing the triangles to have the same slope along their faces leads in calculating the mean value for each slope to establish a relation between observable slopes and the manual measures of the slopes.

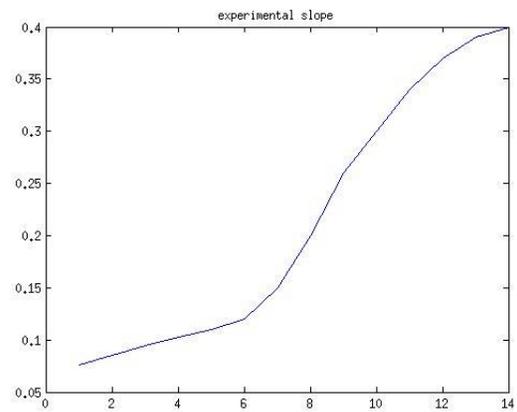


**Figure 3.20 Processed Ratio of blue and red images**

The ratios  $\lambda_i(y)$  are observed and from the H ratios of each line a few are selected covering all stripes in the scene. From this line we sample a point in each different slope occurrence. The slope observed and the real slope based on our results have not the same value, and the slope of the areas facing the red LED array have different values from the slope of the areas facing the blue arrays. Figure 3.21(a) shows the slope due to blue LEDs and Figure 3.21(b) shows the slope due to red LEDs. The angles are not equally distributed among the observable ratios.



(a)



(b)

**3.21 Measured slope : (a) Blue LEDs (b) Red LEDs**

We process ratios of the slopes observed by the blue and red LEDs separately. Based on the observed values and the real slopes we established a relation between them. More specifically depending on the value of the measured slope a label is assigned to each pixel. This label consist the index of a table that holds the real slopes. Two different tables were constructed holding the real values of the slopes that are occurring from the two different light sources.

### 3.2.1 THRESHOLD SELECTION

As we can observe from the different 3D meshes the flat area where no slope occurs is close to one, ranging from 0.9 to 1.1. So the threshold  $\tau$  used for deciding the slope occurrence is set to 0.1. Occurrence of slope is valid when the value of ratio is greater than 1.1 in the case of blue LEDs and smaller than 0.9 in the case of red LEDs. Although this is true when the scene consists of a white paper which is in fact an ideal scene as all colors are reflected back, this may not be true in the real case sceneries such as the road surface. That is the reason why the mean value of each line was selected to denote the flat areas from which the threshold  $\tau$  is added or subtracted to find the slope occurrence due blue and red LEDs respectively.

### 3.2.2 VERIFICATION

To verify our results we reconstructed the triangles by calculated their real distances. To do so we had to find the intrinsic parameters of the camera. We know that the ratio of the size of the object on the sensor and the size of the object in real life is the same as the ratio between the focal length and the distance to the object. Distance from the camera was measured using equation 3.6 and 3.7. To measure the object width a picture of a line was acquired. The size of the line was known and the object width in pixels was easy to calculate. Parameters such as focal length, sensor width and image width are known as shown in table 3.1.

$$\text{object distance from the camera} = \frac{\text{object Size} \cdot \text{focal length}}{\text{object size in image}} \quad (3.6)$$

$$\text{object size in image} = \frac{\text{object width (in pixels)} \cdot \text{sensor width(mm)}}{\text{image width(in pixels)}} \quad (3.7)$$

Parameter	Value
Focal length	3.79 mm
Sensor Width	3.56 mm
Width	1600

Table 3.1 Camera intrinsic parameters

Knowing the distance from the camera we use the same formulas to calculate the object size of one pixel that will be used to estimate the depth. We process the image with the measured slopes and we estimate the depth in each pixel using the following equation.

$$\text{depth in one pixel} = \text{slope} * \text{objectSize of one pixel} \quad 3.8$$

To estimate the whole depth we process the image in the two directions left to right and right to left. When we process the image left to right we examine the slopes that occur when they are facing the blue LEDs and when we process the image right to left we examine the slopes that occur when they are facing the red LEDs. The depth at each pixel is calculated by adding depth in the current pixel with the depth already calculated in the previous pixel. Equation 3.9 describes the value of the distance  $z$  of a line at pixel  $i$ . The parameter  $a$  is the position of the first occurrence of the slope of a specific object.

$$z_i = \sum_{k=a}^{i-1} z_k \quad (3.9)$$

The depth in one pixel can be depicted by the height of the small triangles as shown in Figure 3.22

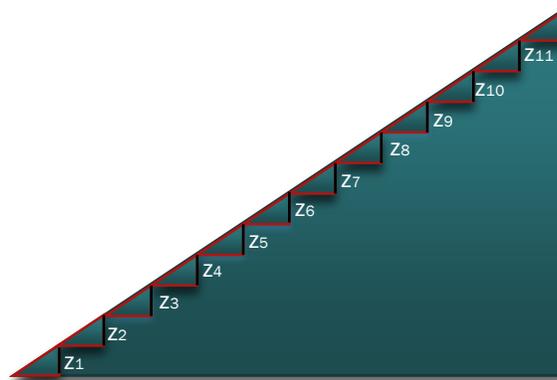


Figure 3.22

Figure 3.23 displays the reconstructed triangles. The real height of each triangle was from 1 to 5 mm as verified from the reconstruction. The height of the first two triangles is less than one and as we can see is not detected from our system. Accurate validation of our technique is difficult to achieve due to difficulty in establishing the ground truth data. i.e the manual measurements of the triangles.

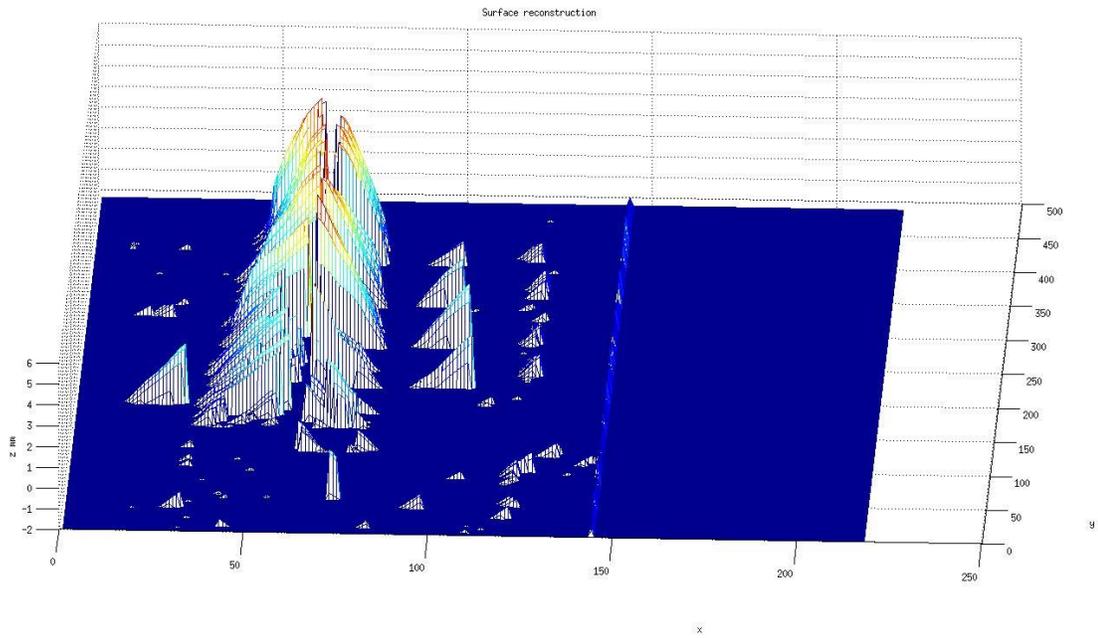


Figure 3.23 Reconstruction of the triangles

# CHAPTER 4

## SYSTEM DESIGN AND IMPLEMENTATION

The system was designed and implemented on a Xilinx Spartan-6 FPGA. The algorithm allows for parallel execution of the image processing tasks. Since each pixel is processed separately, performance depends solely on the pixel stream. Its pixel, provided that can be accessed without any hazards, can be processed in parallel. However, such a behavior is constrained by the amount of the available resources. The system is generic regarding parameters but not reconfigurable during run-time. It can support a range of frame sizes and every parameter discussed in section 3.2 is adjustable.

As already explained in threshold selection section mean value of a line was selected to denote the flat areas of that specific line. Selecting the mean value of the line being processed, would reduce the throughput. Assuming we have an image of size NxM the result would be ready every M clock cycles. To overcome this limitation we chose the mean value of the previous line to denote the flat area based on the assumption that neighboring lines have similar depth. Although, on depth discontinuities the assumption is false, it does not affect the overall outcome.

Considering the fact that ratio results in having floating point numbers we had to chose between floating point and fixed point operations. Selecting floating point operations a significant number of hardware resources are required. Since we already know the range and the precision of the numbers a fixed-point representation is employed for our computations. There is no significant improvement in the ratio results when representing the fractional part with more than 16 bits so we chose this for our implementation.

### 4.1 SYSTEM ARCHITECTURE

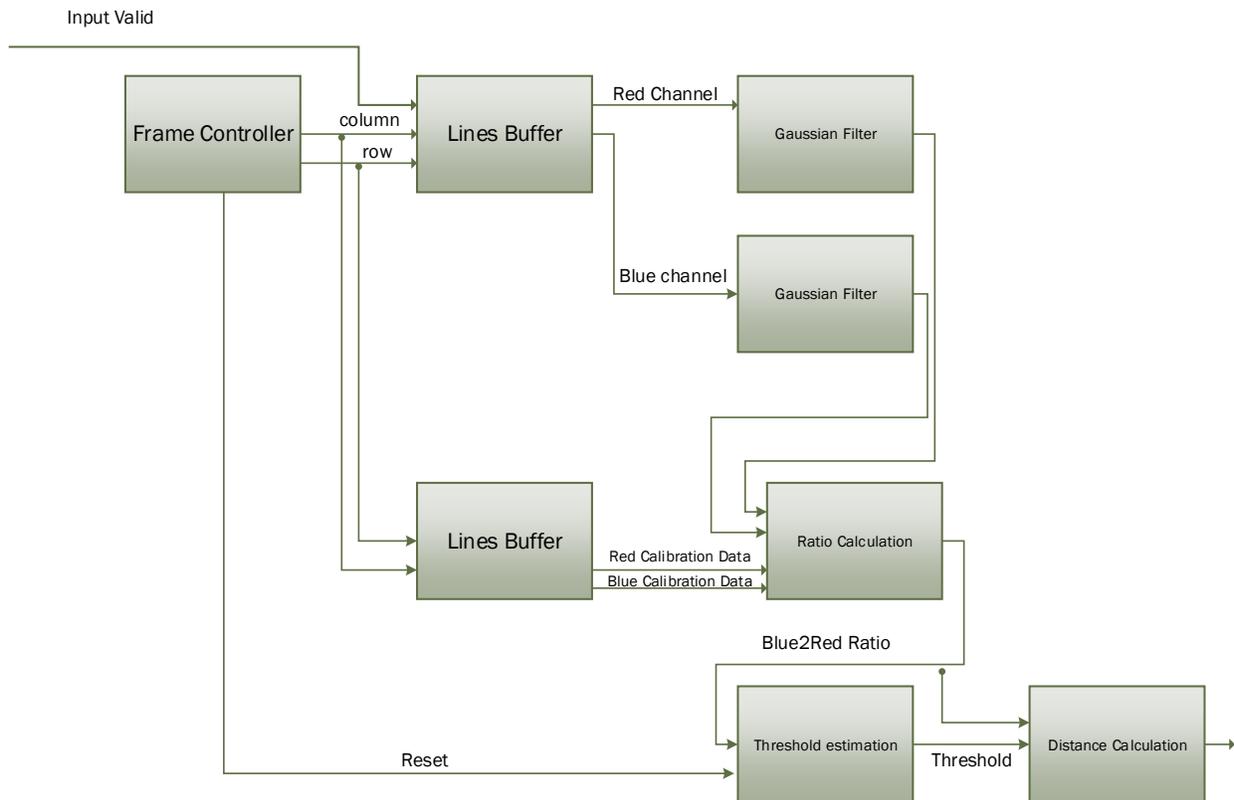
The image generated by the camera is a 16-bit color image of RGB format. Each channel in the RGB format is represented by a specific number of bits. The representation used is 565. This means that out of the 16-bit representation of the pixel value 5 bits are used to represent the value of the red channel, 6 are used for the value of the green channel and 5 for the blue channel as depicted in Figure 4.1. The green channel as already explained in the calibration section is discarded, while red and blue channel are scaled to an 8 bit representation.



Figure 4.1 RGB format

We implemented a pipeline architecture and the general block diagram is shown in Figure 4.2. As Input the system receives one 16-bit value per clock period. The most eight significant bits of

the input value correspond to the red channel and the eight least significant bits correspond to the value of the blue channel. The input value is stored to Lines Buffer, after that the data of the red and blue channel are send to Gaussian filter component where filtering is applied. Except from the input value the calibration data are also stored in Lines Buffer. After filtering the data along with the calibration, data are sent to ratio calculation module where ratio is estimated. Ratio along with a reset value, are sent to threshold estimation module where the different threshold are set. Reset value is used to initialize the different mean values used in the methodology. Distance Calculation calculates the difference between the slopes and the flat areas.



**Figure 4.2 General Block Diagram**

Each Line Buffer is a BRAM that stores a line of the image. The number of Line Buffers depends on the size of the Gaussian filter. Figure 4.3 shows the line Buffer used. Data are then transformed to a window of size  $W^2$  in the Gaussian filter Component. Each value in the window Buffer is multiplied with the coefficients of the Gaussian kernel. The products of each multiplication are then fed to an Adder Tree. Normalization is applied in the end by dividing them with the sum of the coefficients. Filtering is applied to both the red and the blue channel. So both the red and blue channel of the Lines Buffer are fed to the Gaussian filter component. Gaussian filter is shown in Figure 4.4.

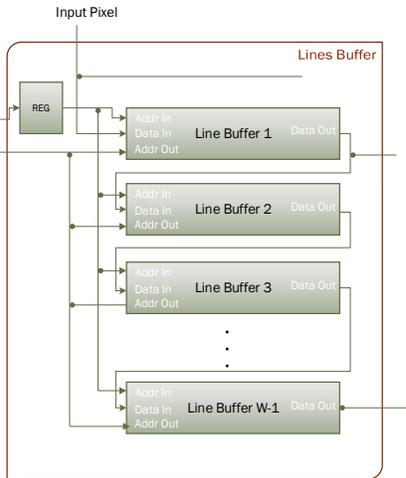


Figure 4.3 Lines Buffer

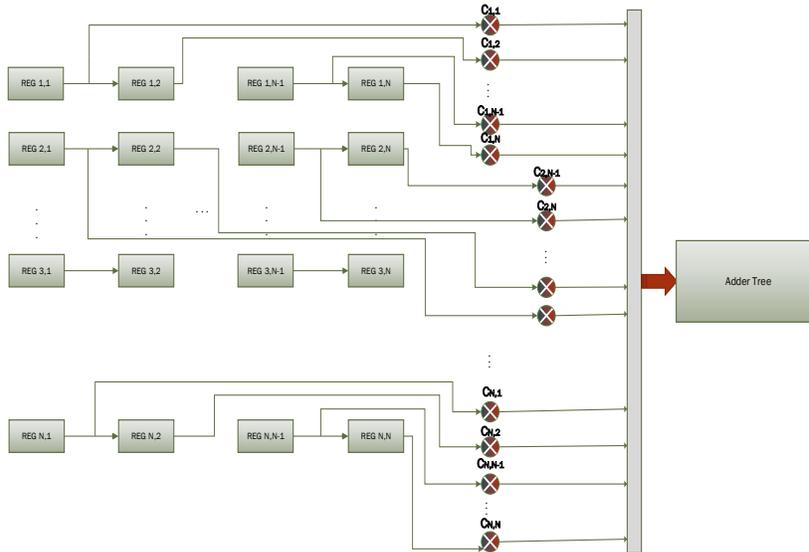


Figure 4.4 Gaussian Filter

After filtering the red and blue channel values are driven to ratio calculation component. The block diagram is shown in Figure 4.5. The equations (2.1) and (2.2) are transformed to the following.

$$ratio = \frac{b \cdot rcal}{r \cdot bcal} \quad (4.1)$$

To implement equation (4.1) we need two multipliers and one divider. Multipliers and divider were generated from Xilinx Core Generator. Data from the blue channel are multiplied with the data of the calibration matrices that are fed to the system along with the image from the camera. The same also applies for the data of the red channel.

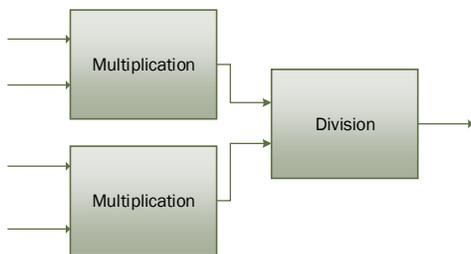


Figure 4.5 Ratio Calculation Component

To establish the threshold we use two components as shown in Figure 4.6. The first define the appropriate threshold of the second component. Specifically it implements equation 3.3. The block diagram of the pre Threshold component is shown in Figure 4.7. We examine the ratio by comparing it to threshold  $\tau_1$  and we count the values that meet those criteria. An Adder and a register are used to sum the values of the ratio along each line and a divider for the calculation of the mean value. More specifically it calculates the  $(\text{ratio} + \text{running sum}) / \text{number of ratios}$ . The register used serves as a feedback node to be used in the next sum.

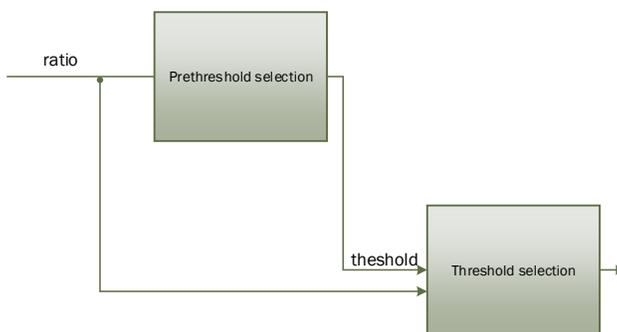


Figure 4.6 Threshold estimation

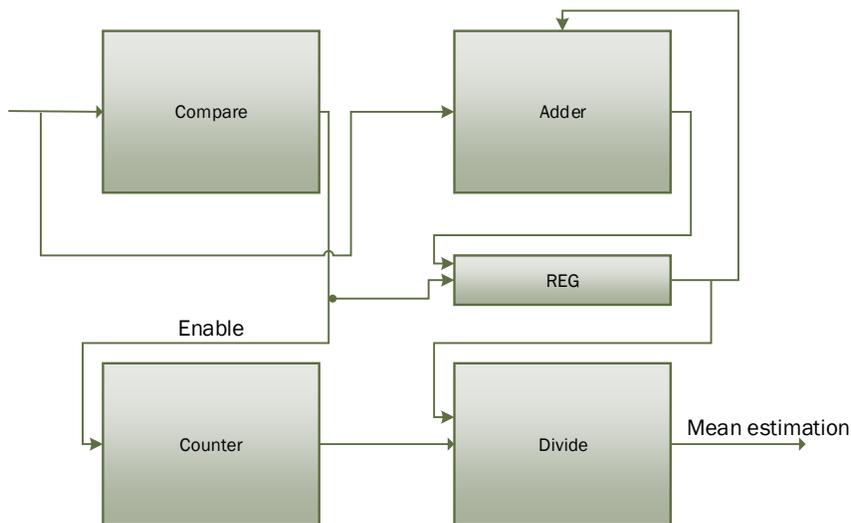


Figure 4.7 Pre Threshold selection Component

Threshold selection component adds and subtracts a threshold from the mean estimation obtained by the previous component. Ratio is examined for being between those thresholds and a new mean value is estimated following the same architecture of the pre threshold selection component as shown in Figure 4.8.

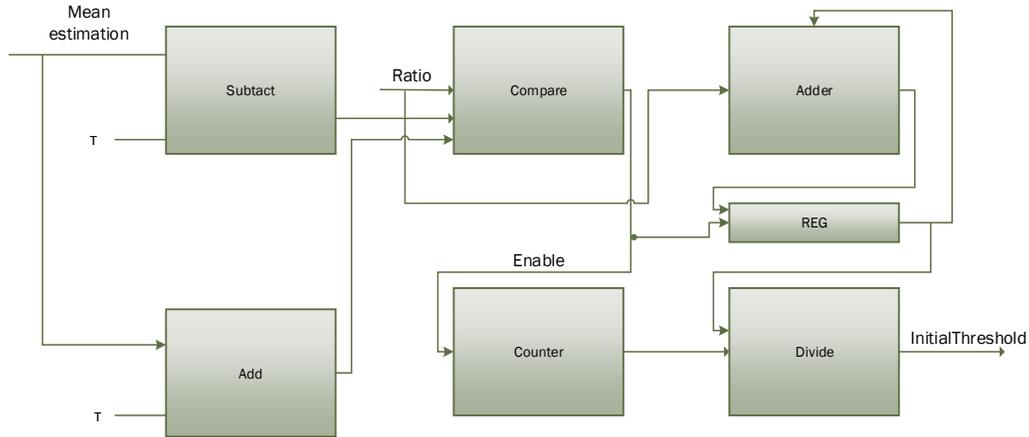


Figure 4.8 Threshold selection component

Distance Calculation Adds and subtracts a threshold from the mean estimation obtaining two new thresholds. These thresholds are compared with the ratio and the results of this comparison are used as a selection to multiplexers that will choose between three results. Either the ratio – mean value or the mean value – ratio or the mean value which of course denotes the flat areas. Block Diagram of the Distance Calculation module is shown in Figure 4.9.

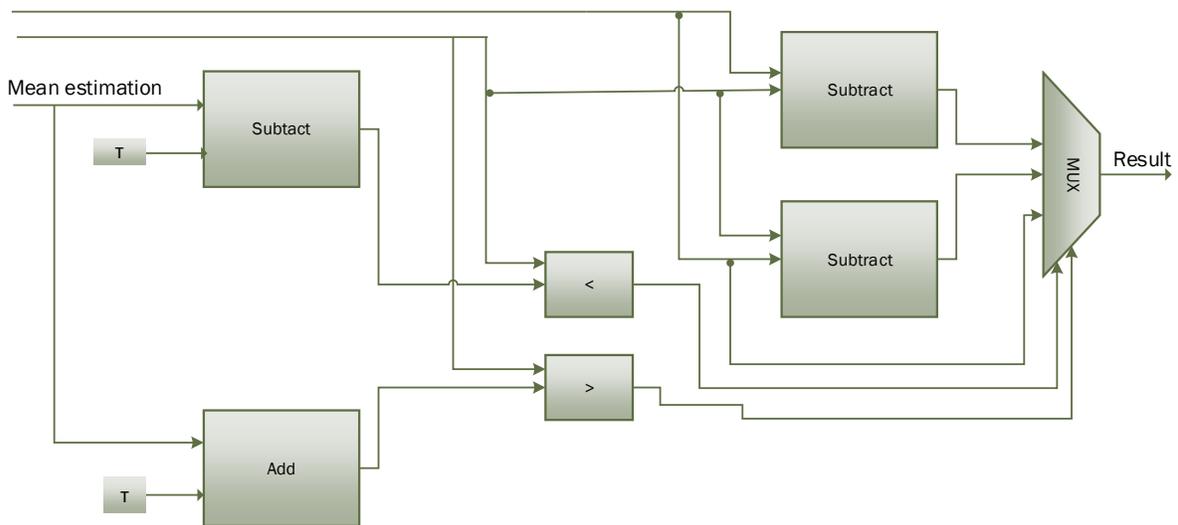


Figure 4.9 Distance Calculation Module

Figure 4.10 displays the system setup. Images are obtained from a Digilent VModCam where the image is buffered DDR2 SDRAM available on the Atlys FPGA board. Image

is read from the memory and is sent to Depth estimation Module where after processing results are written back to memory. Results are read from the memory and are sent to a host PC for verification. On startup the calibration data are written to memory only once.

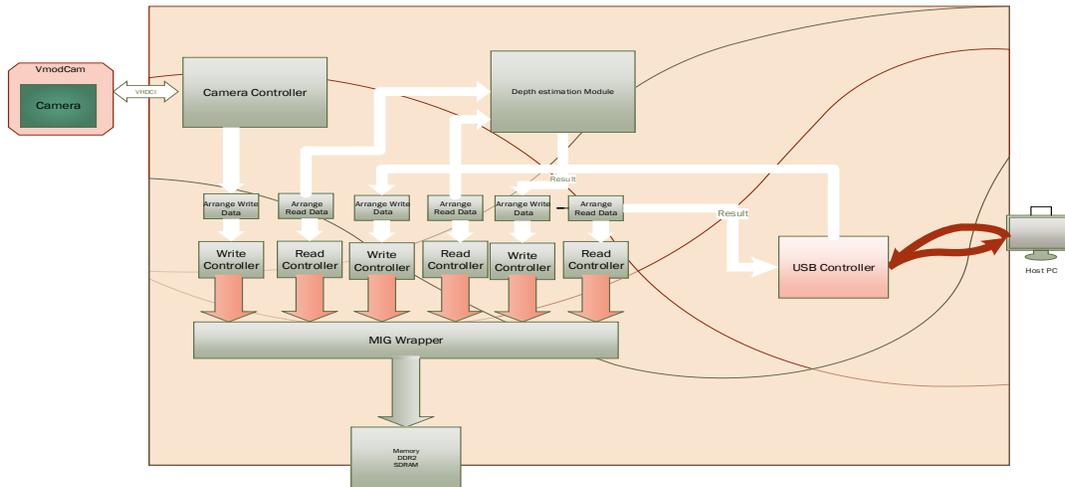


Figure 4.10 System setup

## 4.2 PERFORMANCE RESULTS

Our system receives one pixel per clock and after an initial latency it produces one result per clock. The system implemented is fully pipelined and optimized. Considering the maximum achievable frequency of our system, we present the theoretical maximum throughput in term of frames per second (fps) for different resolutions as shown in Table 4.1.

Resolution	261x273	640x480	720x480	800x600	1280x720	1600x1200	1920x1200
Maximum Clock Frequency	3.185 fps	784 fps	697 fps	404 fps	201 fps	124 fps	98 fps

Table 4.1 The Maximum theoretical throughput of our system

The maximum theoretical clock frequency for each resolution is shown in Table 4.2. As we can see there are variations in the clock frequency between different resolutions.

Resolution	261x273	640x480	720x480	800x600	1280x720	1600x1200	1920x1200
Maximum Frequency	227	241	241	194	186	239	228

Table 4.2 Maximal theoretical frequency for different resolutions

### 4.3 HARDWARE RECOURSES

Hardware recourses required for each resolution are displayed in Table 5.2. From the different results obtained we observe that resources depend on the resolution. The difference in BRAMs is due to the difference in each width, since each BRAM holds a line of the image and the size depends on the width of the image. The difference in LUTs is due to the division cores used. The size of the divisor depends on the width of the image. The implementation of the depth estimation core requires no DSP as opposed to the overall system implementation.

Table 5.4 shows the total hardware recourses required by the system.

Resolution	LUTs(%)	Flip-Flops(%)	BRAMs(%)
Available	27288	54576	116
261x273	9.261 (33%)	12.812(23%)	3(2%)
640x480	9.494(34%)	12.942(23%)	5(4%)
720x480	9.443(34%)	12.942(23%)	5(4%)
800x600	9.443(34%)	12.942(23%)	5(4%)
1280x720	10.084(36%)	13.076(23%)	8 (6%)
1600x1200	10.184(38%)	13.076(23%)	10(8%)
1920x1200	10.386(39%)	13.076(23%)	10(8%)

Table 4.3 Hardware recourses of the depth estimation core

<b>Resolution</b>	<b>LUTs(%)</b>	<b>Flip-Flops(%)</b>	<b>BRAMs(%)</b>	<b>DSPs(%)</b>
<b>Available</b>	27288	54576	116	58
<b>261x273</b>	11.355(41%)	14.197(26%)	9(7%)	11(18%)
<b>640x480</b>	11.588(42%)	14.329(26%)	11(9%)	11 (%)
<b>720x480</b>	11.537(42%)	14.328(26%)	11(9%)	11 (%)
<b>800x600</b>	11.537(42%)	14.330(26%)	11(9%)	11 (%)
<b>1280x720</b>	12.178(44%)	14.465(26%)	14(12%)	11 (%)
<b>1600x1200</b>	12.279(44%)	14.459(26%)	16(13%)	11 (%)
<b>1920x1200</b>	12.483(45%)	14.467(26%)	16(13%)	11 (%)

**Table 4.4 Hardware recourses of the system.**

# CHAPTER 5

## DESIGN VERIFICATION

Our results were verified by providing images to testbench through a library. The images used were downscaled to accelerate the simulation time. The result was written via a library to a txt file in binary form. The binary was subsequently converted to floating point files through a small program written in C. The system was also verified on the Spartan 6 FPGA. The result was sent to a host computer via USB-cable.

To test the result in Atlys Spartan 6 board we modified the block diagram shown in Figure 4.10 by sending the image from USB-port and not from the cameras as shown in Figure 5.1

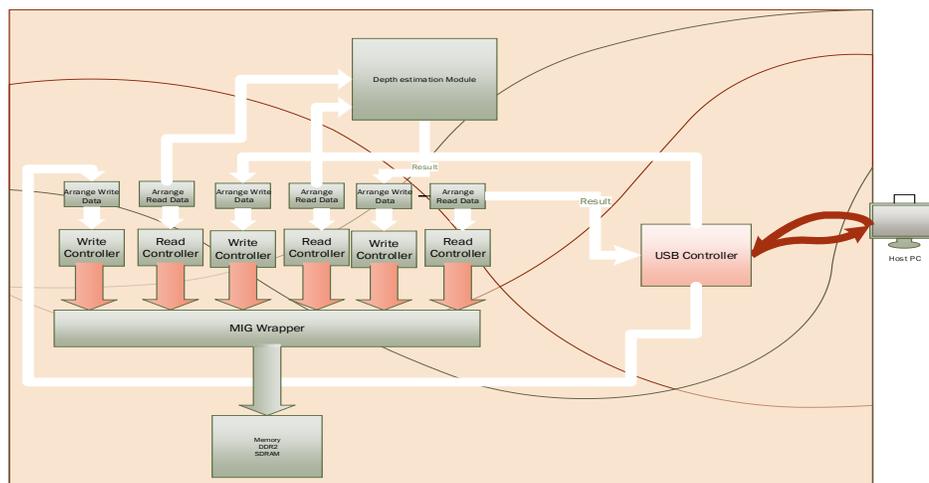


Figure 5.1 System setup for verification

The difference between software and hardware is attributed to fixed point representation used for the ratio results as well as to division cores used to calculate the ratio and the different mean values. Figure 5.2 and 5.3 demonstrates a pixel wise comparison between software and hardware where the black pixels represents deviations larger than 0.1. This difference is estimated between 0.1 to 0.3.

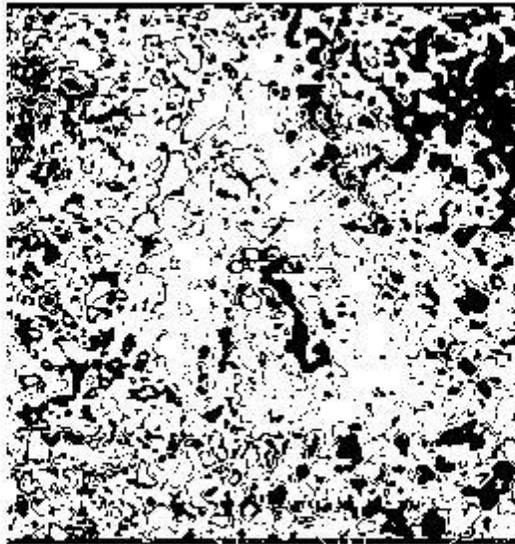


Figure 5.2 Difference between software hardware

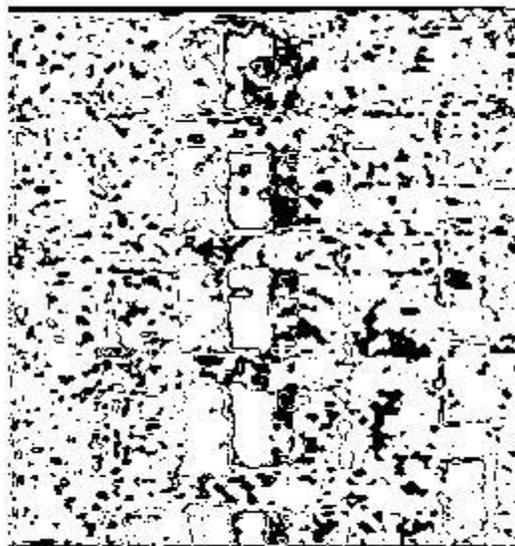


Figure 5.3 Difference between software hardware of the triangles

# CHAPTER 6

## CONCLUSIONS AND FUTURE WORK

The results showed that this technique is very promising and expected to be used as a setup on a moving platform for road inspection and depth estimation of the potholes. Provides information of depth even of one millimeter and above and could be used instead of laser scanners range finders that are currently used. The system though cannot be functional on rainy days as the results will be inconclusive due to water filling the potholes and the unpredicted behavior of the LED arrays when reflected on the water.

The system should be tested more extensively using more scene setups as well as study the behavior on deep potholes. Furthermore a setup for using it on a moving platform should be constructed. This setup most certainly will require more LED arrays and probably two cameras will be required to cover the whole range of the road, which of course will need the synchronization of both.

The most important work that remains is for the system to be used on different FPGA platform for the results to be obtained faster in a host PC for further processing or overview.

## REFERENCES

- [1] Gong, Yuanzheng, and Song Zhang. "Ultrafast 3-D shape measurement with an off-the-shelf DLP projector." *Optics express* 18.19 (2010): 19743-19754.
- [2] Bräuer-Burchardt, Christian, et al. "Fringe projection based high-speed 3D sensor for real-time measurements." *SPIE Optical Metrology*. International Society for Optics and Photonics, 2011.
- [3] Ou, Pan, et al. "Flexible real-time natural 2D color and 3D shape measurement." *Optics express* 21.14 (2013): 16736-16741.
- [4] Zhang, Song. "Recent progresses on real-time 3D shape measurement using digital fringe projection techniques." *Optics and lasers in engineering* 48.2 (2010): 149-158.
- [5] Blais, François. "Review of 20 years of range sensor development." *Journal of Electronic Imaging* 13.1 (2004).
- [6] Peiravi, Ali, and Behrai Taabbodi. "A reliable 3D laser triangulation-based scanner with a new simple but accurate procedure for finding scanner parameters." *Journal of American Science* 6.5 (2010): 80-85.
- [7] Besl, Paul J. "Active, optical range imaging sensors." *Machine vision and applications* 1.2 (1988): 127-152.
- [8] Duchon, Frantisek, et al. "Some applications of laser rangefinder in mobile robotics." *Journal of Control Engineering and Applied Informatics* 14.2 (2012): 50-57.
- [9] Li, Qingguang, et al. "A real-time 3D scanning system for pavement distortion inspection." *Measurement Science and Technology* 21.1 (2010): 015702
- [10] Pu, Shi, et al. "Recognizing basic structures from mobile laser scanning data for road inventory studies." *ISPRS Journal of Photogrammetry and Remote Sensing* 66.6 (2011): S28-S39.
- [11] Yu, Si-Jie, et al. "3D reconstruction of road surfaces using an integrated multi-sensory approach." *Optics and Lasers in Engineering* 45.7 (2007): 808-818.
- [12] Takeda, Mitsuo, and Kazuhiro Mutoh. "Fourier transform profilometry for the automatic measurement of 3-D object shapes." *Applied optics* 22.24 (1983): 3977-3982.
- [13] ] Su, Xianyu, and Wenjing Chen. "Fourier transform profilometry: a review." *Optics and lasers in Engineering* 35.5 (2001): 263-284.

- [14] ] Halioua, Maurice, and Hsin-Chu Liu. "Optical three-dimensional sensing by phase measuring profilometry." *Optics and lasers in engineering* 11.3 (1989): 185-215
- [15] Gåsvik, Kjell J. *Optical metrology*. John Wiley & Sons, 2003.
- [16] D. Scharstein, R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms", in *International Journal of Computer Vision*, Apr. 2002, Vol.47, No. 1-3, pp. 7-42.
- [17] Anderson A. S. Souza, Rosiery Maia and Luiz M. G. Gonçalves (2012). 3D Probabilistic Occupancy Grid to Robotic Mapping with Stereo Vision, *Current Advancements in Stereo Vision*, Dr. Asim Bhatti (Ed.), ISBN: 978-953-51-0660-9, InTech, DOI: 10.5772/49050. Available from: <http://www.intechopen.com/books/current-advancements-in-stereo-vision/3d-probabilistic-occupancy-grid-to-robotic-mapping-with-stereo-vision>
- [18] Thomas, Sotiris, Kyprianos Papadimitriou, and Apostolos Dollas. "Architecture and implementation of real-time 3d stereo vision on a xilinx fpga." *Very Large Scale Integration (VLSI-SoC), 2013 IFIP/IEEE 21st International Conference On*. IEEE, 2013.
- [19] Neoh, Hong Shan, and Asher Hazanchuk. "Adaptive edge detection for real-time video processing using FPGAs." *Global Signal Processing* 7.3 (2004): 2-3.
- [20] Koch, Christian, and Ioannis Brilakis. "Pothole detection in asphalt pavement images." *Advanced Engineering Informatics* 25.3 (2011): 507-515.
- [21] Grace, A. E., et al. "Active shape from stereo for highway inspection." *Machine Vision and Applications* 12.1 (2000): 7-15.
- [21] Koch, C., and I. Brilakis. "Improving Pothole Recognition through Vision Tracking for Automated Pavement Assessment." *Advanced Engineering Informatics in Press*, <http://dx.doi.org/10.1016/j.aci.2011>.
- [22] Axelsson, Peter. "Processing of laser scanner data—algorithms and applications." *ISPRS Journal of Photogrammetry and Remote Sensing* 54.2 (1999): 138-147.
- [23] Yang, Bisheng, Lina Fang, and Jonathan Li. "Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds." *ISPRS Journal of Photogrammetry and Remote Sensing* 79 (2013): 80-93.
- [24] Barber, David, Jon Mills, and Sarah Smith-Voysey. "Geometric validation of a ground-based mobile laser scanning system." *ISPRS Journal of Photogrammetry and Remote Sensing* 63.1 (2008): 128-141.

[25] Jepson, Allan D., and Michael RM Jenkin. "The fast computation of disparity from phase differences." *Computer Vision and Pattern Recognition, 1989. Proceedings CVPR'89., IEEE Computer Society Conference on*. IEEE, 1989.

[26] Rematska, Georgia, Kyprianos Papadimitriou, and Apostolos Dollas. "A low cost embedded real time 3D stereo matching system for surveillance applications." *Bioinformatics and Bioengineering (BIBE), 2013 IEEE 13th International Conference on*. IEEE, 2013.

[27]Cutrona, L. J. "Synthetic aperture radar." *Radar handbook 2* (1990): 2333-2346.

[26] <http://www.digilentinc.com/Products/Detail.cfm?NavPath=2,719,932&Prod=VMOD-CAM>

[27] <http://www.makestuff.eu/wordpress/software/fpgalink/>