POTENTIAL UNDERWATER SENSING USING NARROW BEAM LASERS

by

Sabuj Das Gupta
BSc in Electrical and Electronic Engineering, American International University-Bangladesh, 2009

A Project Submitted in Partial Fulfillment
of the Requirements for the Degree of

MASTER OF ENGINEERING

in the Department of Electrical and Computer Engineering

POTENTIAL UNDERWATER SENSING USING NARROW BEAM LASERS

by

Sabuj Das Gupta
BSc in Electrical and Electronic Engineering, American International University
Bangladesh, 2009

Supervisory Committee

Dr. Adam Zielinski, Supervisor
(Department of Electrical and Computer Engineering)

Dr. Alexandra Branzan Albu, Department Member
(Department of Electrical and Computer Engineering)

Abstract

This report investigates underwater sensing using images created by reflections of narrow laser beams from surface (laser footprints). These images are captured by a tilted camera to form a picture. Such pictures are affected by ocean bottom features and are subjected to perspective distortions. A methodology is developed and verified by some laboratory experiments for underwater sensing. Multitude Laser beams can be created from a single laser by suitable scanning, refraction grid or laser array. A square laser point grid is formed from single laser beam using a diffraction mask and projected over bottom. The plane of the grid is perpendicular to the gravitational vector. The laser beam or laser array is used to project continuous or discrete (point) images (grid) on the bottom. Laboratory experiment showed this system senses object in the floor successfully with dimension and within error rate below 5%. The characteristics of the laser footprints also studied for flat and tilted bottom.

Table of Contents

Contents

List of Tables

List of Figures

Acknowledgments

I would like to express my honest gratitude to my supervisor Dr. Adam Zielinski for supporting towards my Masters of Engineering project, his motivation, guidance and knowledge. His assistance directed me in research, design and implementation of my project.

Besides my supervisor, I would like to be grateful to Dr. Alexandra Braznan Albu for serving my supervisory committee. I would also like to thank University of Victoria for providing me all kind supports towards my graduate studies.

At last, I am deeply thankful to my mother, father and loving wife for their endless compassion and support whom had an outstanding role during completion of my graduate studies.

Dedication

I would like to dedicate my report to my parents Sukhendra Kumar Das Gupta and Khela Rani Das Gupta and wife Papre Sharma Ava for their endless love, support and encouragement in all stages of my life.

Chapter 1

1.1 Introduction

Underwater sensing is critical for many civilian and military applications. Surveys are conducted for the purpose of marine sciences, mine detection, identifying navigation channels and anchorage areas, repairing of undersea structures, such as underwater communication cables and pipelines and etc.

There is a wide variety of technologies available for accurate underwater sensing such as surveillance and inspection of underwater structures with optics cameras, digital sonar data captured by autonomous underwater vehicle, digital close-circuit television (CCTV), lasers etc. Acoustic sonar surveying is an expensive and labour intensive task. This is one of the reasons for the large hydrographical backlogs being experienced by many countries [1]. Light detection and ranging system (LIDAR) is the other possible approach that could offer quick and accurate ocean floor information extraction and underwater sensing for both large and small areas. This method costs less, but is only applicable in shallow and clear waters. The development of LIDAR from the 1970's parallels the development of acoustic bathymetry. Early systems were rather slow and ponderous but advancement in underlying technologies has made LIDAR much more effective.

Underwater sensing normally requires long time processing and analysis by human experts. Side scan automatic processing software packages in object recognition field (which is one of the main functions of hydrographic survey) yield obvious discrepancies [2]. These discrepancies can be referred to the differences in shapes and sizes of submerged and buried objects (such as pipelines, rocks, and wrecked ships). Despite the fact that manual processing techniques became difficult to deal with massive amount of collected data and had a critical time consuming disadvantage, they may lead to a break though by detecting an object which have lost during automatic image quality enhancement process [3].

This report describes a methodology to monitor underwater bottom variation and sensing natural or manmade objects using narrow laser beams. The laser footprints reflected from the surface is captured with a still camera and processed to extract

information about the floor variation. This methodology is also capable of producing a 3-D view of the surface with height.

1.2 Contributions

In this report the complexities and procedures for underwater floor variation capture and sensing techniques of submerged manmade or natural object are studied. Image perspective distortion correction methodology is also analyzed, as this is an important operation while extracting bottom information.

1. A process is presented for correcting the perspective effect from the images formed by narrow laser beam projected perpendicular to the bottom. Experimental results are presented to support this technique.
2. Four parallel lasers are put together to form an array and projected over a flat and tilted surface and observed from different camera positions to analyze the floor variation. Important footprint characteristics are observed and discussed in this report.
3. Finally, an array of laser beams are created where a diffraction mask is used to produce multiple laser beams from single laser source. A still camera is used to take snapshots from different positions. An image-processing algorithm that extracts surface information is also presented.

1.3 Organization of Report

The main objectives of this project are to extract surface variations and sensing natural or manmade objects using laser beams. The first part of the project describes correction of perspective distortion from pictures taken with tilted camera. A complete algorithm to correct the perspective distortion from 2-D image is explained. The second part of the project investigates the extraction of image information for bottom characterization and underwater sensing with the help of the images without perspective distortion.

The report is organized as follows: Chapter 2 gives an overview of the literature survey and a list of similar works that have been done with different technologies. Chapter 3 describes the process of perspective distortion correction in detail. In chapter 4 underwater bottom characterization from laser footprints is explained. Chapter 5 presents the method to extract information from images taken by tilted camera for underwater

sensing along with experimental results, improvements and applications of the proposed method. Chapter 6 provides the concluding remarks.

Chapter 2

2.1 Literature Survey

This project investigates the correction of perspective distortion from images taken with tilted camera shot. In order to produce an accurate variation of bottom floor or sensing underwater objects, it is required to extract information properly from images without perspective distortion. The literature relevant to perspective distortion correction is highlighted in the first part of this section followed by a brief list of research works on underwater object detections.

2.2 Background of Perspective Distortion Correction

Researchers have proposed different methods to correct perspective distortion in a single 2D image. Authors have used the perspective cues, typically vanishing point for corner detection, to predict the depth information and the homography matrix. Murali S et al. [4] have proposed a method based on perspective transformation and plane homography to correct the perspective distortion in an image to actual scale with known camera parameters. Jian Liang et al. [5] estimates shapes from texture flow information obtained directly from the image without requiring additional metric data. Clark and Mirmehdi [7] proposed a perspective distortion correction method based on the use of vanishing points recovery, where these points provide the information required to correct the perspective distortion of the captured document. In Mirmehdi's [7] work, the recovery of the vanishing points is based on the assumption that an image must display some sort of left, right, centered or full formatting. Its main drawbacks are the computational load required by some steps of the correction method including several image transformations and exhaustive searching of parameters.

A different approach is followed by Lu et al. [8] where a method based on applying morphological operators was proposed. This method needs neither high-contrast document boundaries nor paragraph formatting information. Nevertheless, it is constrained to deal with text documents, and even more importantly, it is based on the use of some parameters that require much knowledge about the document contents. For instance the number of characters in the document image and few parameters that are not automatically tuned, such as, the parameters used to discard false detections of top and

bottom lines of the text. Thus small changes on the value of those parameters can make the method fail. More recently, Lu and Tan [8] have presented an extended version of the mentioned work in [9], although it inherits the discussed drawbacks.

Morphological operations are also used in the method proposed by Miao and Peng [10], however less knowledge about the contents of the image is required in comparison with Lu et al.'s [8] approach. Additionally, an adaptive thresholding technique is adopted to binarize the capture, which makes the method capable of dealing with lighting variations. However, the image correction process requires the use of three transformations, which are computationally intensive. Method described in Yin et al. [11, 12] does not depend on the contents of the image, as that method is valid for any kind of image. Article [11] focused on the perspective estimation problem, while article [12] focused on the correction system design. Yin et al. [11] described a method that uses textual information if it is available and also other sources of information, such as image boundaries.

Nevertheless, the major drawback of these methods is the fact that they are not able to recover the original aspect ratio of the image. Iwamura et al. [13] proposed a method that estimates the depth of each area of the image by using measurements of its contents, like the variation of the area of objects with respect to their position. Nevertheless, the proposed approach does not obtain the focal length, being only able to recover an affine distorted version of the original image.

2.3 Background of Underwater Sensing

The conventional method of underwater sensing is to measure the intensity of the reflected signal. With a strong reflected light signal it is possible to ensure the existence of an object and calculate its depth under water from the travel time of the returned signal. It is to be noted that, an underwater object becomes covered due to the current development of counter detection technology. It also does not exhibit reflection and scattering characteristics. For that reason the object detection process that depends on the reflection or scattering poses great challenge to the traditional method. Thus, in certain situations, the traditional methods do not work and a new method is necessary.

Since the 1960s, researchers have used optical scattering technologies, including Brillouin scattering and Raman scattering, to remotely sense physical properties of the

ocean such as sound velocity, temperature, salinity and bulk viscosity [14-16]. Results were obtained [16] with accuracies ranging from ±10 to ±0.75 m⁄s for measuring sound speed using Brillouin scattering and ±2 to ±0.4°C for measuring water temperature using Raman scattering in water. In 1997, Fry et al. [17] presented an elaborate discussion concerning the accuracy limitation on Brillouin lidar measurements of temperature and sound speed in the ocean. Since 2000, many similar works have also been presented by Liu's group [18-20]. In 2004, detecting submerged objects by Brillouin scattering and an edge-filter technique was first shown by Gong et al. [21]. They pointed out several advantages of detecting submerged objects with Brillouin scattering, including the independence of surface reflectivity of the tested submerged objects, real-time measurement, good signal-to-noise ratio (SNR) and less influence from the light reflected from the air-water interface. The principles of all these works were based on spontaneous Brillouin scattering. The spontaneous Brillouin scattering signal is very weak and requires sophisticated filtering and signal processing methodologies.

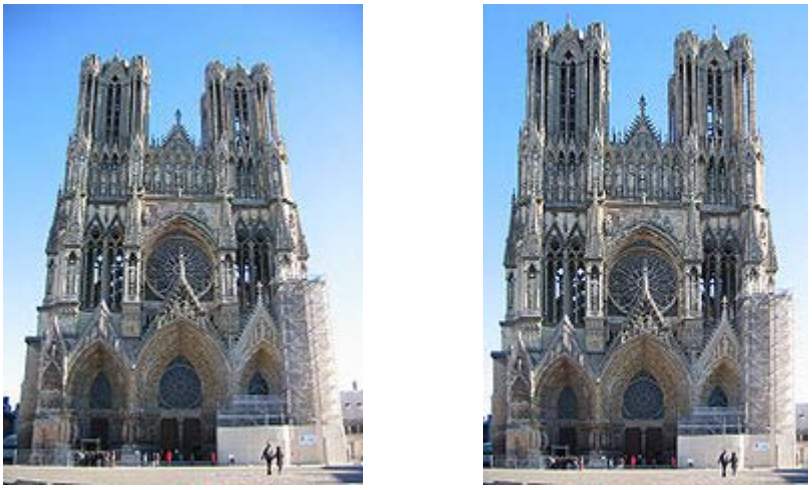
In 1990, Leonard and Sweeney compared stimulated and spontaneous laser-radar methods for the remote sensing of ocean physical properties [22]. In that paper, they pointed out the potential advantage of the stimulated Brillouin scattering (SBS) method, in which the SBS beam is reflected from the water with high efficiency (more than 50%) in exactly the reversed direction of the incident laser beam and this property is especially useful for remote sensing because the scattered returned signal can be collected efficiently at a receiver collocated with the transmitter.

Until 2006, Shi et al. [23] first developed a light detection and ranging (lidar) system based on SBS. Compared with its predecessor mentioned above [21], this new system can greatly increase the SNR and its depth resolution can be improved to 1⁄8 of that with spontaneous Brillouin scattering. In the same year, Lv et al. [24] also designed a system for detecting a submerged object using SBS, which is made different from Liu's system by adopting a polarizer and a quarter-wave plate as the signal extracting subsystem and using a photodiode instead of an intensified CCD (ICCD) camera as the reception subsystem. But their reported detected distance is very short, just 2m.

Chapter 3

3.1 Perspective Distortion Correction

Perspective distortion is one of the common characteristics of images taken by cameras. Perspective distortion confuses user from its original view and often leads to error or failure in analysis. Now-a-days perspective corrected images are required for several applications such as image based rendering, 3D reconstruction, automobile license plate reading methodology and etc. Perspective projection is the projection of actual vision. Many applications would need parallel projection because it facilitates easy extraction. However parallel projection is not parallel in perspective hence its identification is necessary. From this point of view it is essential to correct perspective distortion of images.



(a) **(b)**

Figure 3.1 a) Image with perspective effect, b) Image after perspective effect corrected.

3.2 Methodology

The algorithm that is used in this project includes the following steps to correct the perspective distortion effect. The workflow begins with reading the image previously saved in the image directory. Then the corner points of the image are detected. Corner points are found from the intersection point of the edges. Corner point detection method is described in the later part of the section. The overall workflow is shown in the Figure 3.2.

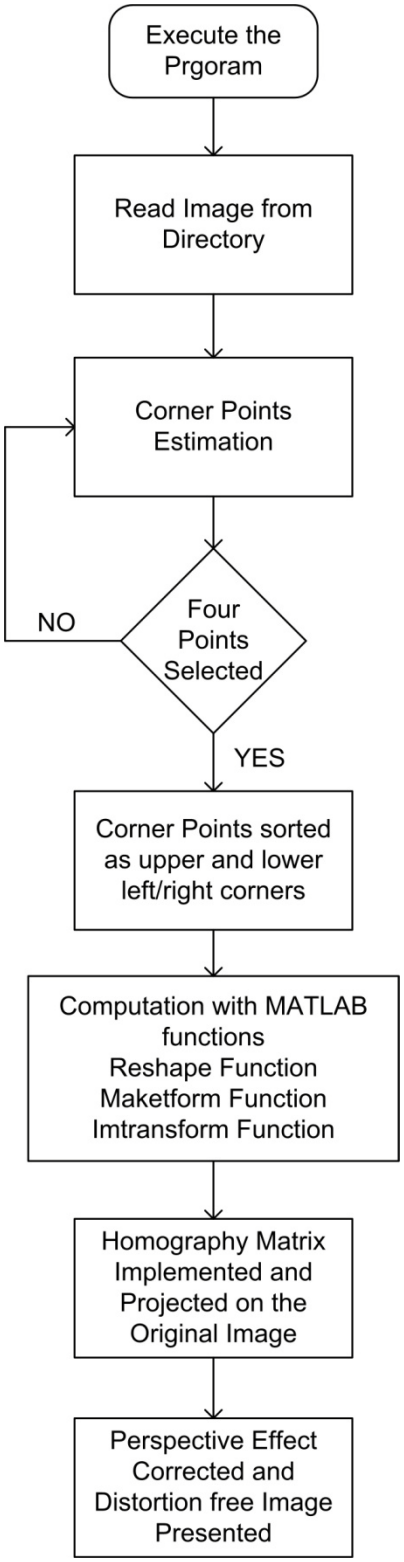


Figure 3.2 Flowchart of the algorithm for correction of perspective distortion.

After the corner points are detected, the perspectives are estimated and homography matrix is implemented. This process is done by using geometric properties of planar

surfaces. Finally, the projection is applied on the original image. This projection is applied with the help of the homography matrix. This algorithm is simple yet powerful in correcting the obvious perspective distortion from most type of images. This approach requires sufficient portion of the picture in the view.

3.3 Corner Detection Technique

There are several methods to detect the corners of an image, such as: Harris corner detector, vanishing point method, Shi & Tomasi's minimum eigen-value method and etc. It is also possible that those corners are detected by the user which offers much flexibility for image crop situations. Therefore, it is possible to work with only a portion of an image that makes the computation faster. Harris corner detector is a widely used method that gives good result and helps to comprehend the detection process properly. Corner detection is necessary as those points will dictate the border of the image. Perspective distortion correction is achieved by multiplying homography matrix with the original image. The homography matrix can only be applied to an image that has at least four corner points selected. For that reason, four corner points have been detected in this algorithm.

Harris corner detector is a mathematical operator that finds a variation in the gradient of an image. It is rotating, scaling and illumination variation independent [28]. It sweeps a window w(x, y) with displacement ‘a’ in the x-direction and ‘b’ in the y-direction and calculates the variation of intensity as following,

$$E(a,b) = \sum_{x,y} w(x,y)[I(x+a, y+b) - I(x,y)]^2 \tag{3.1}$$

where,

w(x, y) is the window function which is equal to ‘1’ inside the window and ‘0’ outside,

I(x, y) is the intensity at (x, y),

I(x + a, y + b) is the intensity at the new shifted frame (x + a, y + b).

The large variation in intensity depicts corners. Hence Equation (3.1) has to be maximized, specifically the term

$$\sum_{x,y} w(x,y)[I(x+a, y+b) - I(x,y)]^2 \tag{3.2}$$

Let's denote,

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} Ix^2 & IxIy \\ IxIy & Iy^2 \end{bmatrix} \tag{3.3}$$

Now Equation (3.1) becomes,

$$E(a,b) = [a\ b] M \begin{bmatrix} a \\ b \end{bmatrix} \tag{3.4}$$

A value is calculated for each window, to determine if it can possibly contain a corner:

$$R = det(M) - k(trace(M))^2 \tag{3.5}$$

where,

$$\det(M) = \lambda_1 \lambda_2;$$

$$\text{trace}(M) = \lambda_1 + \lambda_2;$$

λ_1 and λ_2 are the eigen values of 'M'. 'R' depends on eigen values of 'M'. A window with a value 'R' greater than the threshold is selected and points of local maxima of 'R' are considered as a "corner". Further it is reduced to four points as the requirement of the plane homography. The four points of interest is obtained by selecting the minimum and maximum values in the corner point array as shown in Figure 3.3.

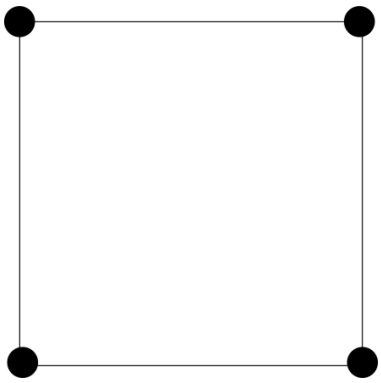


Figure 3.3. Four corner points are detected.

3.4 Perspective Approximation

Images captured by tilted cameras are perspective in nature. In perspective affected cases, objects of same size seem bigger when it is closer to the view point and smaller when it is farther. For that reason it requires less pixels to represent distant objects in the image compare to a closer object in the image. Fewer pixels mean less information for

image processing. Perspective affected images lead to ambiguous results when an individual tends to measure the size of the objects in the picture. The rectangular plane ABCD in Figure 3.4(a) is seen as A1B1C1D1 in a perspective distorted case.

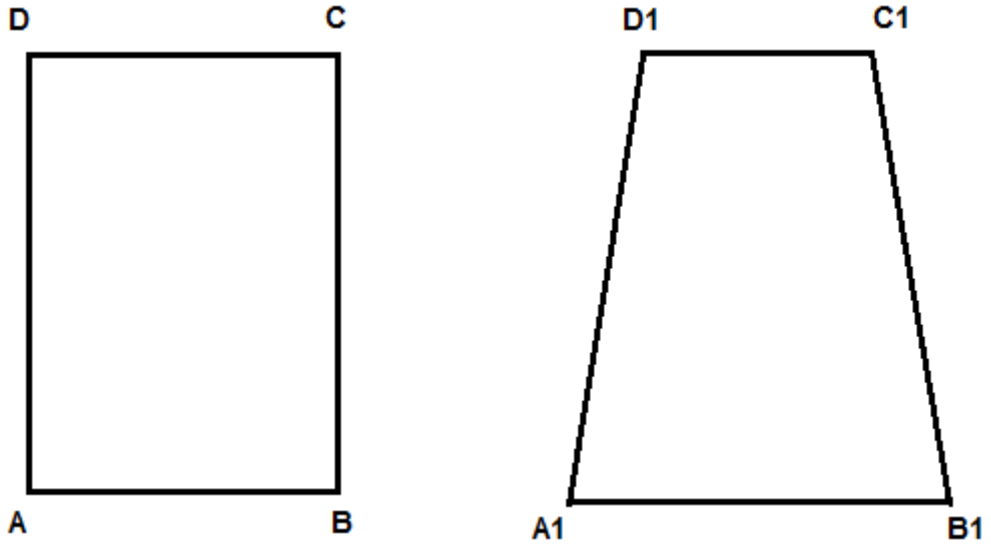


Figure 3.4 a) Original shape of the image plane, b) Image plane with perspective distortion.

3.5 Correction of Perspective Distortion

Perspective transformation turns a perspective projection into a parallel projection. Perspective transformation is used for replacing a view volume into a rectangular shape. The plane homography is used to create the parallel projection of the image from its perspective affect version [11]. The homography can be computed by knowing the relative positions of the four points on the perspective distorted image and the positions of the transformed image to be constructed. The four corner points are required to implement the perspective transformation.

The corner point selection method described in the earlier section of this report detects all the available corner points. The value calculated for each pixel in the corner detector is based on the two eigen values of a matrix. The expression to calculate it is based on observations of variations with different eigen values. The plane homography deals with only four points. For that reason only the four desired corner points are obtained from the minimum and maximum values in the corner point array.

The corner points are further sorted in the clockwise direction. The four corner points of the perspective image (Carrom Board) that is used in the plane homography method as shown in Figure 3.5(a).

a1(X1, Y1),

a2(X2, Y2),

a3(X3, Y3) and

a4(X4, Y4)

The corrected image is obtained having the corners as follows

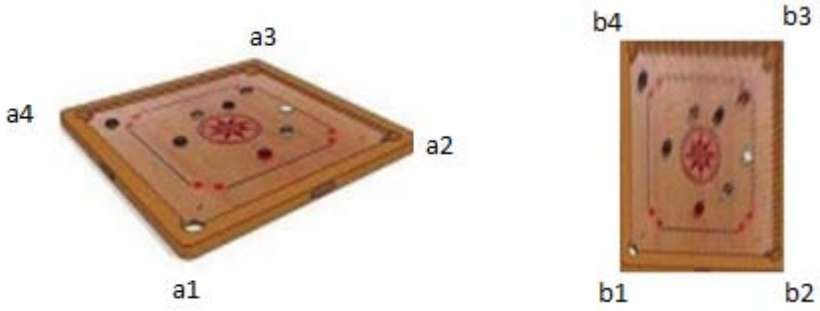
$b1(x1, y1) = b1(\text{Lower Left}_x, \text{Lower Left}_y)$

$b2(x2, y2) = b2(\text{Lower Right}_x, \text{Lower Right}_y)$

$b3(x3, y3) = b3(\text{Upper Right}_x, \text{Upper Right}_y)$ and

$b4(x4, y4) = b4(\text{Upper Left}_x, \text{Upper Left}_y)$

Detail description about detecting these four points is provided in the result section of this article.



(a) **(b)**

Figure 3.5 a) Original Image, b) corrected version of image (a).

Plane homography is widely used for mapping the perspective boundary to a rectangle image, which is a necessary step towards perspective transformation. A 2D homography is defined as a 3×3 homogeneous matrix that maps any point p(x, y) to its corresponding point p′(x′, y′) as,

$$p' = H \,.\, p \tag{3.6}$$

The homography is described by a homographic transformation when p and p′ are converted to homogeneous coordinates.

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.7)$$

It is required to compute ṕ = H • p to apply a homography transformation 'H'. The homography transformation can be applied in correcting the perspective distortion of an image by finding the homography 'H', given p and ṕ The derivation can be attained by first finding the transformation for a single point and then extending it to four points.

Then Equation (3.7) becomes,

$$\begin{bmatrix} wx' \\ wy' \\ w \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (3.8)$$

Solving for $x'$and y':

$$x' = \frac{ax + by + c}{gx + hy + i} \text{ and } y' = \frac{dx + ey + f}{gx + hy + i}$$

The two linear equations with respect to unknown coefficients of matrix H are derived as below:

$$ax + by + c - gxx' - hyy' - ix' = 0 \quad (3.9)$$

$$dx + ey + f - gxy' - hyy' - iy' = 0 \quad (3.10)$$

Similarly four point transformation can be found by extending,

$$p_i(x_i, y_i) \rightarrow p_i'(x_i', y_i') \quad (3.11)$$

Equation (3.9) and (3.10) can be generalized as,

$$ax_i + by_i + c - gx_ix_i' - hy_iy_i' - ix_i' = 0 \quad (3.12)$$

$$dx_i + ey_i + f - gx_iy_i' - hy_iy_i' - iy_i' = 0 \quad (3.13)$$

The above equation can be written in a matrix format as Ai. **h**=0 for i=1, 2, 3, 4 where **h**=[a b c d e f g h i]^T is a vector of unknown coefficients in H and A_i is a 2×9 matrix based on known coordinates x_i, y_i, x_i', y_i' given as,

$$\begin{bmatrix} x_i\ y_i\ 1\ 0\ 0\ 0\ -x_i'x_i\ -y_i'y_i\ -x_i' \\ 0\ 0\ 0\ x_i\ y_i\ 1\ -y_i'x_i\ -y_i'y_i\ -y_i' \end{bmatrix} \quad (3.94)$$

Finally Equation (3.8) can be written as,

$$P' = H\,.\,P \rightarrow \text{Ai}\,.\,\text{h} = 0 \text{ for i} = 1, 2, 3, 4. \quad (3.15)$$

where A_i is a 2×9 matrix, **h** is a 9×1 vector and the result is a 2×1 vector. If all the four matrix equations are put into one equation, it results with the following,

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \\ A_4 \end{bmatrix} . \boldsymbol{h} = 0 \tag{3.106}$$

where the order of the matrix A_i is (4×2) ×9 = 8×9, **h** is a 9×1 vector and the result is a 8×1 vector. The resultant expression can be written as,

$$A_i \, . \, \boldsymbol{h} = 0 \tag{3.17}$$

It implies that there are eight (8) linear equations and nine (9) unknowns. Adding constraints ||**h**||=1 simplifies to A . **h** = 0 subject to ||**h**||=1. The solution obtained using singular value decomposition for the four points transformation between two planes. The elements $A_{1\text{-}8}$ of the transformation can be given as,

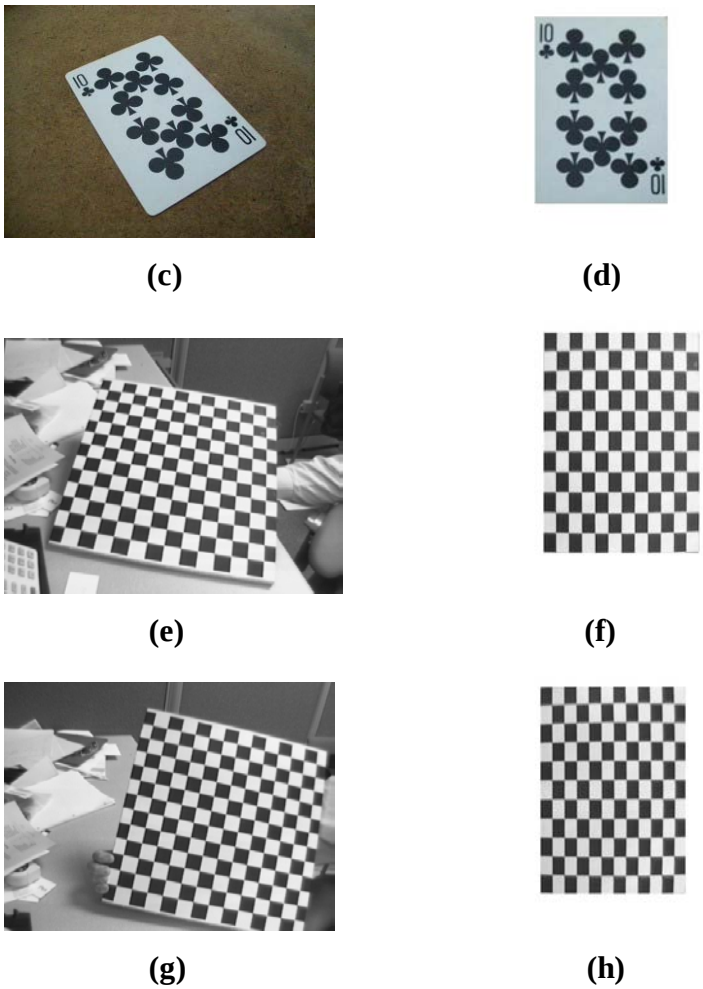
$$\begin{bmatrix} X1 & Y1 & 1 & 0 & 0 & 0 & -X1*x1 & -Y1*x1 \\ 0 & 0 & 0 & X1 & Y1 & 1 & -X1*y1 & -Y1*y1 \\ X2 & y2 & 1 & 0 & 0 & 0 & -X2*x2 & -Y2*x2 \\ 0 & 0 & 0 & X2 & Y2 & 1 & -X2*y2 & -Y2*y2 \\ X3 & Y3 & 1 & 0 & 0 & 0 & -X3*x3 & -Y3*x3 \\ 0 & 0 & 0 & X3 & Y3 & 1 & -X3*y3 & -Y3*y3 \\ X4 & Y4 & 1 & 0 & 0 & 0 & -X4*x4 & -Y4*x4 \\ 0 & 0 & 0 & X4 & Y4 & 1 & -X4*y4 & -Y4*y4 \end{bmatrix}$$

And the vectors are given as $\boldsymbol{h_{1-8}} = [a_{11}\, a_{12}\, a_{13}\, a_{21}\, a_{22}\, a_{23}\, a_{31}\, a_{32}\,]^T$and $\boldsymbol{A_9} = [x_1\, y_1\, x_2\, y_2\, x_3\, y_3\, x_4\, y_4\,]^T$.

This transformation is applied on the perspective images and the perspective distortion is removed as per demonstrated in the results section.

3.6 Results

(a) **(b)**



(c) **(d)**

(e) **(f)**

(g) **(h)**

Figure 3.6 Perspective distortion corrected images by this algorithm. Original images (a, c, e, g) those are used as input of this program, Perspective distortion rectified version (b, d, f, h) of those images from the algorithm.

Above images are examples of this algorithm. Images noted as ‘a, c, e, g’ are the original images with perspective distortion. Once the algorithm is executed it opens the images as per the user command which is saved in the image directory. User is now required to enter the four corner points of the images. Selection of these points is simple and is done by clicking on the image corner points. A function is built inside this algorithm to track the corner points of the images. Once the points are selected the program automatically decides the position of the points in the following way.

Upper Corners: - The first two large values of the y-axis are the top two edges of the image and thus upper two corners are identified. And the rest of the two points are the lower corner points.

Upper Left and Right Corners: - The point which has larger x-axis value from the upper corners belongs to the top-right hand corner and the other one is the top left corner point.

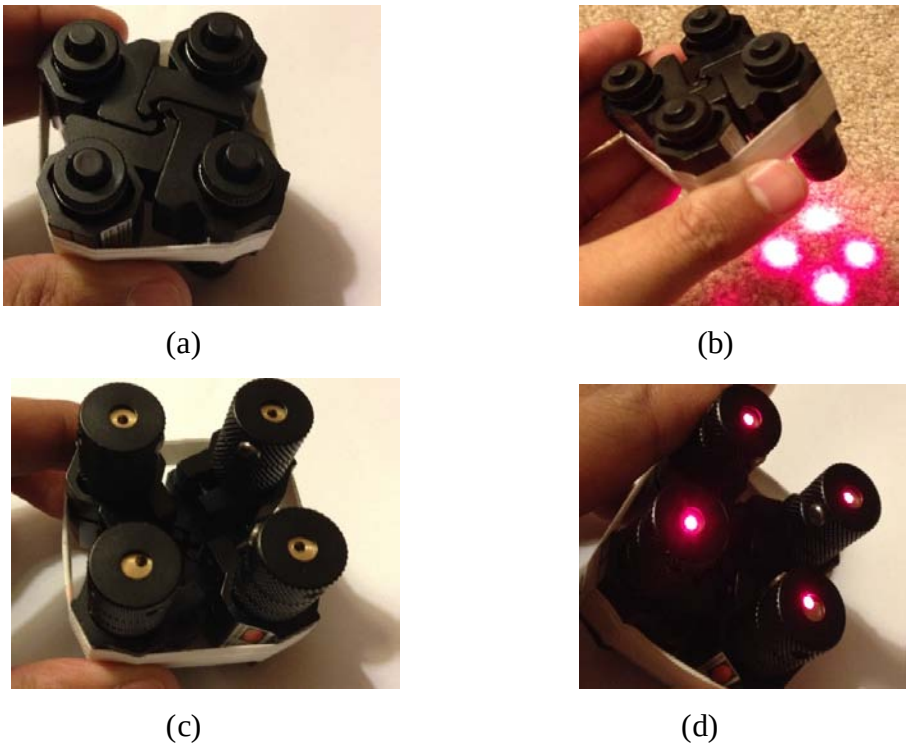
Lower Left and Right Corners: - The point that has larger x-axis value from the lower corner values belongs to the bottom right hand corner and the other one is the bottom left corner point.

Based on these criteria selected points are positioned by this function. This function offers the flexibility over traditional methods that user does not need to select the corners points in clockwise or anti-clockwise direction. Images numbered as ‘b, d, f, h’ are the output images from the algorithm.

Chapter 4

4.1 Underwater Bottom Characterization

The laboratory experiment was conducted initially with four lasers in the ai. Each laser projected a narrow beam aligned with other beams. The beams were reflected from the surface below forming a four dots pattern presenting corners of a multi angle figure. The lasers were adjusted manually to align with vertical axis. Due to the mechanical defects with one laser it was not possible to align perfectly parallel with other beams. In the first scenario, the four lasers module created a normal projection and images were captured with a tilted camera for a flat bottom. This process has been repeated with the same laser projection on a flat bottom but the camera was aligned with the laser array axis. Ideally for absolute alignment with array axis the camera should be placed in the centre point of the four lasers system package. It was not possible to do so with this model and the camera was offset from the array axis slightly. Figure 4.1. (a) & (b) and (c) & (d) are the four lasers system's top and bottom view respectively.



(a) (b)

(c) (d)

Figure 4.1 Top (a & b) and bottom (c & d) view of four parallel laser system.

(a) **(b)**

Figure 4.2 Four parallel laser projection on flat surface with camera aligned with array axis (a) and tilted position (b).

Figure-4.2 (a) & (b) represent the experimental snapshots for scenario one. The bottom surface is flat with normal projection from lasers and camera is aligned with array axis. (figure 4.2 (a)) and tilted (figure 4.2 (b)) position. The laser footprint formed a circular disk shape and four footprints together have created nearly a square shape for this scenario. On the other hand, these four footprints have created a trapezoid shape when the image was captured with camera in a tilted position. The variation in the figure from square to trapezoid is due to the perspective distortion occurred in the image.

It can be concluded from this scenario that the bottom is flat if either condition set-A or condition set-B is satisfied.

Condition Set-A:

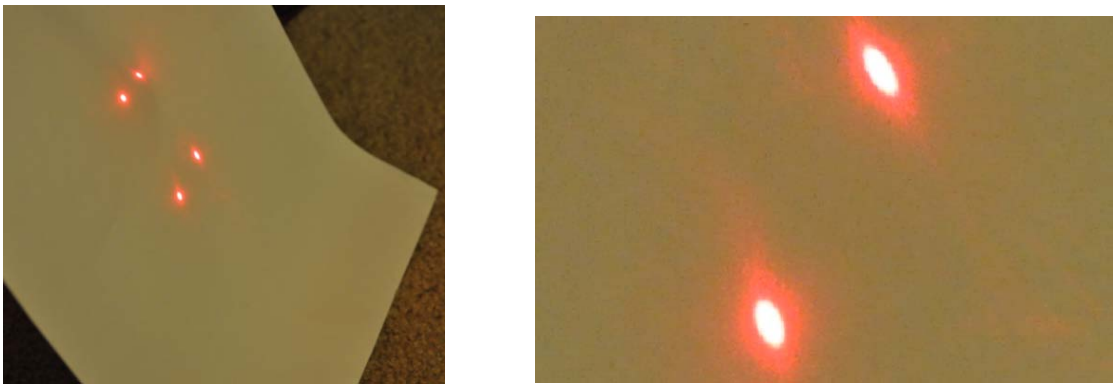
1. Lasers are normally projected over bottom,
2. The camera is aligned with array axis,
3. The laser footprints are circular disk in shape and
4. Four laser footprints together formed a square shape.

Condition Set-B:

1. Lasers are normally projected over bottom,
2. The camera is in tilted position,
3. The laser footprints are circular disk in shape and
4. Four laser footprints together formed a trapezoid shape.

In the second scenario, the four lasers system created a normal projection on a slope bottom and images were captured with a tilted camera. A snapshot of this scenario

have been presented in Figure 4.3 and explained the properties acquired from the laser footprints.



(a) **(b)**

Figure 4.3 (a) Four parallel laser projection on slope bottom, (b) zoomed version of (a).

The figure 4.3 represents the image for the second scenario with a slope bottom. The laser footprint generally transforms to an elliptical shape when there is a slope in the bottom. This is visible in the footprints of Figure 4.3. The size of the elliptical footprints becomes large or small depending on the grazing angle. It is also possible to estimate the bottom slope from the laser footprint [25, 26].

It can be concluded from this scenario that the bottom has a slope if it satisfies condition set-C.

Condition Set-C:

1. Lasers are normally projected over bottom,
2. Camera is in tilted position and
3. Laser footprints are elliptical in shape.

Scenario two has been repeated with same laser position in slope bottom but with camera aligned with array axis. As mentioned in scenario one, the camera position was offset slightly to align with vertical axis as close as possible. The result of this step has been presented in the Figure 4.4.

Figure 4.4 Four parallel laser projections on slope bottom with camera aligned with array axis.

As can be seen from Figure 4.4 the laser footprints are not elliptical in shape although the bottom has a slope. They are circular in shape which describes the property for a flat bottom. With the same laser projection but a tilted camera is placed and the following figure is obtained. As expected from description of Figure 4.3 that, laser footprints are elliptical in shape for a slope bottom with tilted camera, is present in Figure 4.5 below.

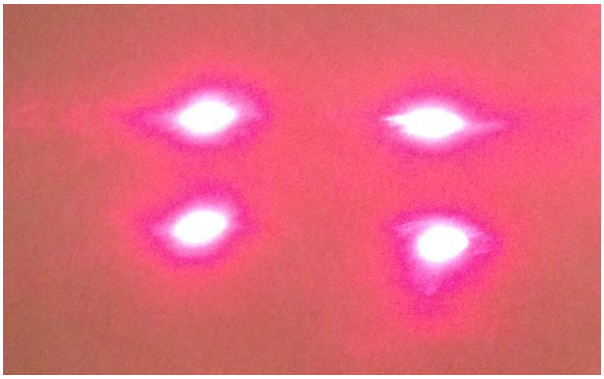


Figure 4.5 Four parallel laser projection on slope bottom with tilted camera.

Therefore from Figure 4.4 it can be said that, this type of arrangement of the camera and lasers is not ideal for investigating the bottom variation.

Further investigation is conducted to extract more information about the bottom characterization and observed that four lasers model has limitation towards this purpose. An ideal model would be a matrix of parallel laser beams attached together with a same distance in between them to form a rectangular or square shape grid when reflected from the bottom. In the next part of the experiment a square grid of laser footprints is formed from a single laser beam using a diffraction mask and projected over bottom. The plane of the grid is perpendicular to the gravitational vector. The camera is almost vertically aligned with the array axis. The idea behind this scenario was to get more footprints thus

more information regarding bottom. A detail discussion about this is presented in chapter-5 where information is extracted from laser footprints to detect natural or manmade objects from the bottom.

Chapter 5

5.1 Underwater Sensing

Underwater sensing process requires information from images without perspective distortion. Correcting these images is necessary as the algorithm fetches all the information from the laser footprints projected on the surface. It is impossible to locate the position of the footprints precisely without perspective correction especially for a large array beams. The flowchart of extracting information for underwater sensing is presented in figure 5.1.

This approach requires at least two images with laser point grid formed by laser beams reflected from the bottom. A matrix of lasers can be put together and reflected from bottom to create the laser footprint grid. Alternatively a diffraction mask can be mounted in front of a single laser beam to produce this grid. A still camera is used to capture pictures. The first image is considered as the reference image with normal projection of laser beams and tilted camera towards the flat surface representing bottom. The second image is treated as the target image formed by laser projection and the camera orientation same as before on the flat surface including an object. The distance from the camera to surface and laser source to surface are kept constant for reference and target images. Pictures taken with a tilted camera from an angle suffers perspective distortion, which is corrected by using the methodology described in Chapter-3 of this report.

Usually pictures taken by cameras are in color format but color information of image does not help in the calculation. For that reason at first it is needed to convert the color image into a gray-level image. A MATLAB function "rgb2gray ()" is used for this purpose. The color of each point consists of three elements: red, green and blue (R, G and B). The transformation is done by this method:

$$Gray\_Level = 0.29900*R + 0.58700*G + 0.11400*B$$

There is no such "correct" conversion from RGB to gray scale, since it depends on the sensitivity response curve of the detector to light as a function of wavelength. The weights used to compute luminance are related to the monitor's phosphors. The explanation for these weights is due to the fact that for equal amounts of color human

eyes are most sensitive to green, then red, and then blue. Thus, the image obtained by the normal averaging of an image's three color components produces a gray scale brightness that is not perceptually equivalent to the brightness of the original color image. The above co-efficient values are used as they are well recognized values and also defined inside MATLAB functions.

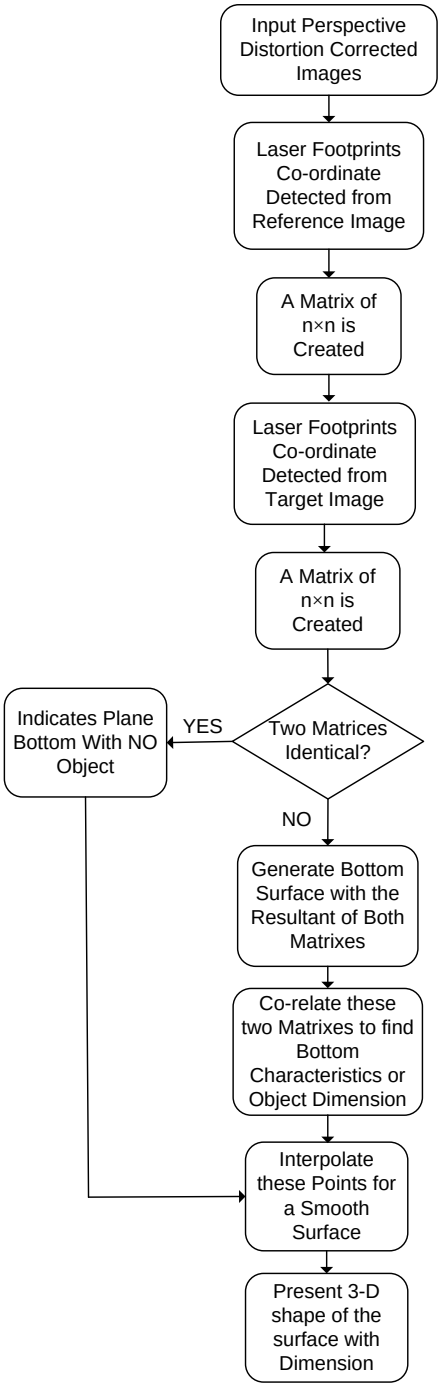


Figure 5.1 Flowchart of underwater sensing algorithm.

The laser footprints on both of these images are identified and two matrices of size n × n are formed. The algorithm compares these two matrices and computes the result which is another matrix containing all the bottom characteristics information. The algorithm produces a flat bottom with little variation when the values of these matrices are nearly equal. These matrix values change in presence of any objects or for wavy variation in the floor. This algorithm is not only capable of producing the contour of the underwater object but also the dimensions of the object. Mathematical explanation for the calculation of the object dimension is as follows.

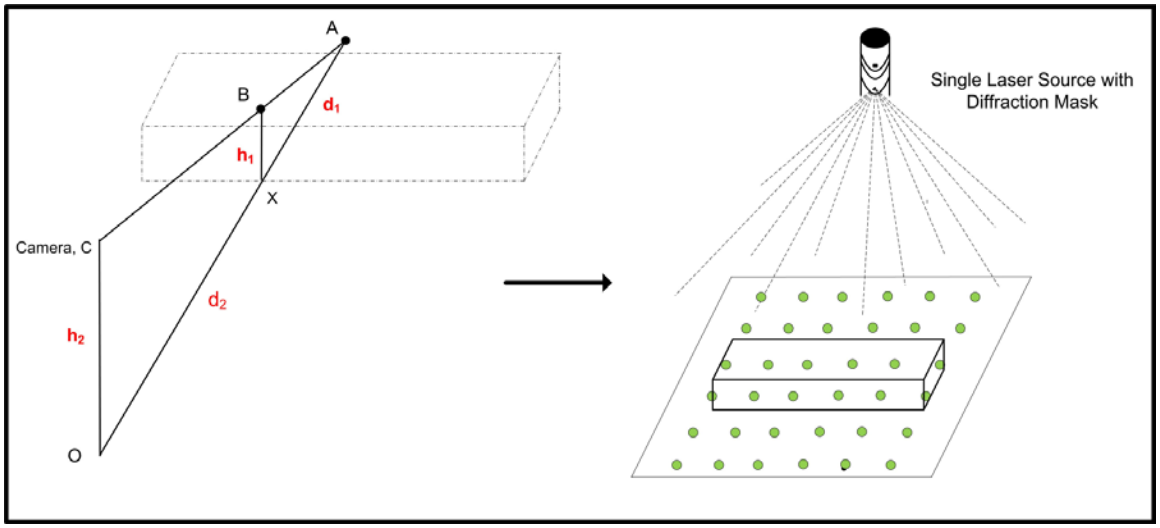


Figure 5.2 View from the observing point towards the laser footprints.

Figure-5.2 is a geometrical representation of the experimental setup while lasers are normally projected over the bottom. Here the camera is located at 'C' and the ground position of the camera is O(x_o, y_o, z_o) where, (x_o, y_o) is found from the image and z_o is found from the height of the camera during the experiment. OC is the height of the camera and denoted by h_2. B(x_1, y_1, z_1) is the location of one of the laser footprints projected on the object. Here (x_1, y_1) is known from the image and z_1 is unknown. BX is the height of point 'B' and denoted by h_1. Finally A(x_2, y_2, z_2) is the projection of point 'B' seen from the camera. All the coordinates of A(x_2, y_2, z_2) is known from the image. The distance from A to X is d_1 and O to X is d_2.

The distances d_1 and d_2 is found with the following two equations,

$$d_1 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (5.1)$$

$$d_2 = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2} \quad (5.2)$$

Now from Figure 5.2 it can be written with the help of the ratio of the distances,

$$\frac{h_1}{d_1} = \frac{h_2}{d_1+d_2} \tag{5.3}$$

Hence the height h_1 is calculated by using,

$$h_1 = \frac{d_1 . h_2}{d_2+d_1} \tag{5.4}$$

Let's consider the reference matrix (extracted from the reference image) is a 5 × 5 matrix of all zeroes to explain the calculation.

$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The target matrix (extracted from the target image) is also a 5 × 5 matrix including some non-zero values (assuming there is an object in the bottom).

$$B = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Then the distance between each pair of points are calculated considering the camera is at the same place and we get the output matrix 'C' as following,

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Each value in this output matrix represents the movement of the laser footprints from its original position in the reference image because of the height of the target object. In other words, each value is 'd_1' at that point. 'd_2' is the distance between the point and the camera in 2-D surface. After that, these values are substituted in Equation (5.4) to get the estimated height of the object 'h_1' which is used to construct the 3-D model of the bottom including object.

Finally, an interpolation methodology is applied to achieve a smooth surface. The MATLAB function is "griddata()". The purpose of this function is to find the intermediate points between two values. For example, the input of this function is $\boldsymbol{X} = [1\ 2\ 3]$ and a range of the output is set as "min(X), max(X), 10". The function will provide a 10 × 10 matrix like below:

$$\boldsymbol{X} = [1\ 2\ 3] \quad \rightarrow \quad X = \begin{bmatrix} 1 & P_{02} & P_{03} & P_{04} & P_{05} & 2 & P_{06} & P_{07} & P_{08} & 3 \\ 1 & P_{12} & P_{13} & P_{14} & P_{15} & 2 & P_{16} & P_{17} & P_{18} & 3 \\ 1 & P_{22} & P_{23} & P_{24} & P_{25} & 2 & P_{26} & P_{27} & P_{28} & 3 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & P_{92} & P_{93} & P_{94} & P_{95} & 2 & P_{96} & P_{97} & P_{98} & 3 \end{bmatrix}$$

These intermediate values (noted as P_{12}, P_{13}…...P_{18}, P_{22},P_{23}…...P_{28}, …...P_{92},P_{93}…...P_{98} in the above matrix) are calculated by interpolation algorithm and the frequency of these points are decided by the number of points defined by the user (in this example it is 10). There are several interpolation methodologies available and different intermediate values will be achieved based on the interpolation algorithm selection.

It is important to calculate the distance between two laser footprints, as it is the bottleneck of this approach. The distance between any two laser footprints is same if perfectly parallel laser sources used to project over the bottom. But when diffraction mask is used in front of single laser beam to produce an array of beams then the laser footprints on the boundary of the grid tend to move far from each other compare to other footprints and create a large distance between them. The distance between two laser footprints on the floor also becomes larger as the laser source moves upward. The underwater sensing approach described in this report is unable to recognize any object smaller than this length (distance between two laser footprints on the bottom). Figure 5.3 is used to describe the calculation to find the distance between two laser footprints from a certain height.

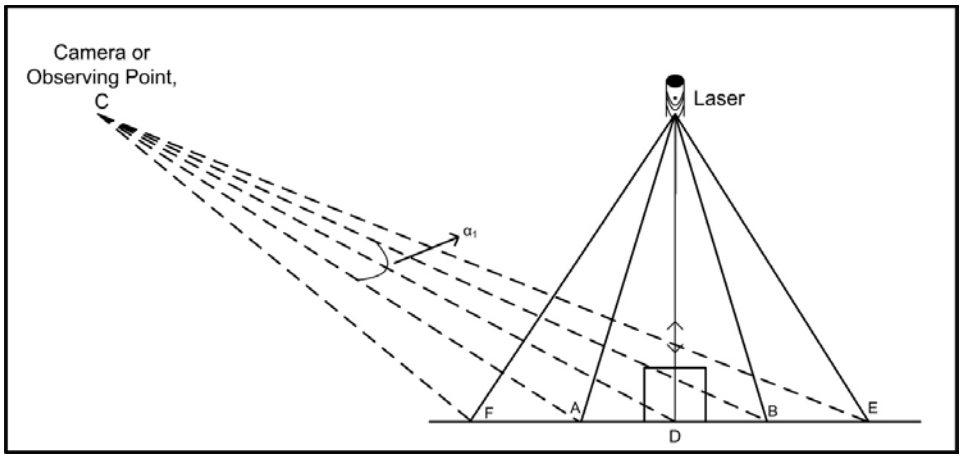


Figure 5.3 Diagram of single laser beam diffracted to form array of beams.

Let's consider the laser source is projected on the flat bottom and the camera is in tilted position. It is proposed from the above diagram, the perceived length 'AB' of this segment is proportional to the angle α_1 between those points as seen from the observation point. This length 'AB' depends on the spatial position of the segment, its length and the distance of 'AB' to the observation point. For instance, if a segment is aligned with the observing direction then 'AB' is zero, i.e., it is seen as a point. In contrast, 'AB' is seen as maximum in its size if the segment is orthogonal to the observing direction. Hence it can be written,

$$AB \propto \alpha_1 \tag{5.5}$$

$$AB = k\,\alpha_1 \tag{5.6}$$

The mathematical explanation is shown below with a simple version of figure 5.3 with experimental values.

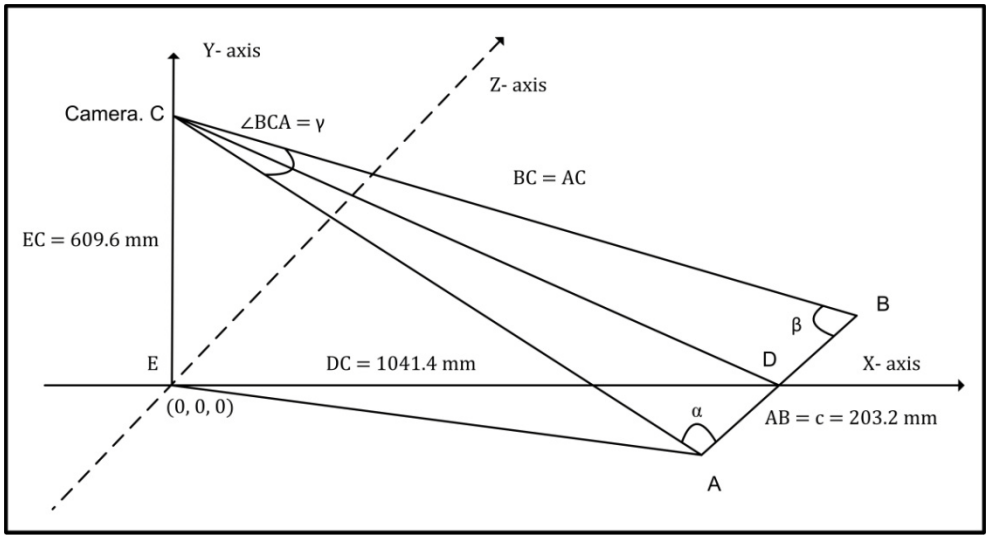


Figure 5.4. Geometrical diagram of the experiment to calculate the observing angle.

ΔACD and ΔBCD are obtained from figure 4.4 and the followings are denoted,

$$AC = b, BC = a \; and \; BC = AC$$

$$\angle BCA = \gamma, \angle DAC = \alpha \; and \; \angle DBC = \beta$$

Hence from ΔACD and ΔBCD it can be written,

$$AD = b * cos\alpha \quad (5.7)$$

$$DB = a * cos\beta \quad (5.8)$$

Applying Pythagoras theorem in the ΔBCD & ΔACD triangles,

$$a^2 = CD^2 + BD^2$$

$$CD^2 = a^2 - BD^2 \quad (5.9)$$

And $$b^2 = CD^2 + AD^2$$

$$\rightarrow CD^2 = b^2 - AD^2 \quad (5.10)$$

From Equation (5.9) and (5.10),

$$a^2 - BD^2 = b^2 - AD^2 \quad (5.11)$$

Also from the Figure 4.4 it can be written,

$$AD + BD = c$$

$$\rightarrow BD = c - AD$$

$$\rightarrow BD = c - b * cos\alpha$$

Now substituting the values of BD and AD in Equation (5.11) we get,

$$a^2 - (c - b * cos\alpha)^2 = b^2 - (b * cos\alpha)^2$$

$$\rightarrow a^2 - c^2 + 2bc * cos\alpha - b^2 * cos\alpha^2 = b^2 - b^2 * cos\alpha^2$$

$$\rightarrow a^2 - c^2 + 2bc * cos\alpha = b^2$$

Hence, $$\cos\alpha = \frac{b^2 + c^2 - a^2}{2bc} \quad (5.12)$$

Similarly, $$\cos\beta = \frac{a^2 + c^2 - b^2}{2ca} \text{ and } \cos\gamma = \frac{a^2 + b^2 - c^2}{2ab}$$

Therefore, $$cos\gamma = cos\angle BCA = cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \quad (5.13)$$

Experiment is conducted to check the validity of this model and is explained in details in the following paragraph. In Figure-5.5 ‘C’ is the location of the camera. It is located at a height of H = 609.6mm from the surface. The ground distance between camera and the object is X = 1041.4mm. The length of the object is AB = 203.2mm.

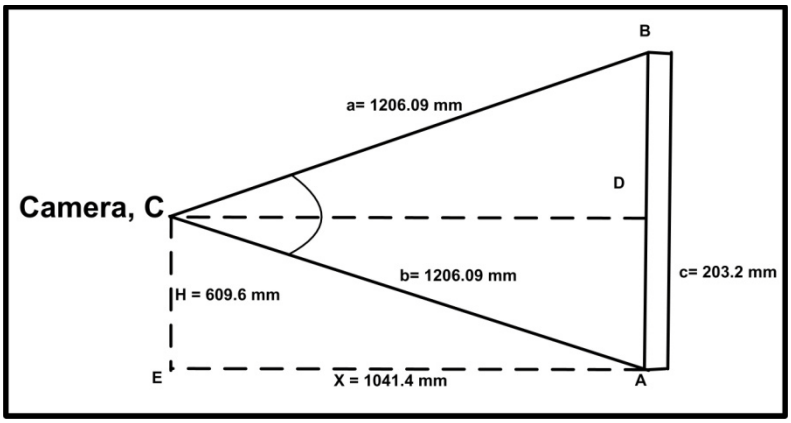


Figure 5.5 Geometrical diagram of the experiment to calculate the observing angle.

It is proposed that the perceived length AB is proportional to ∠BCA as seen from point A and B. It indicates that the length of AB can be perceived if ∠BCA is known or vice versa. As per Equation (5.13)

$$\gamma = \angle BCA = cos^{-1}\left(\frac{a^2 + b^2 - c^2}{2ab}\right)$$

The parameters H, X and c are known from the experimental values. By using Pythagoras theorem in ΔACE,

$$AC^2 = H^2 + X^2$$

$$\rightarrow AC = a = b = \sqrt{H^2 + X^2} \qquad (5.14)$$

By using Equation (5.14), a = b = 1206.9mm. Now substituting the values of a, b and c in Equation (5.13),

$$\gamma = cos^{-1}\left(\frac{1206.9^2 + 1206.9^2 - 203.2^2}{2 \times 1206.9 \times 1206.9}\right)$$

$$\rightarrow \gamma = 9.7\, degree$$

Another representation of equation (5.13) as a function of H, X and c is,

$$\gamma = cos^{-1}\left(\frac{2(H^2+X^2)-c^2}{2(H^2+X^2)}\right) \qquad (5.15)$$

The experimental value of the ∠BCA is found approximately 9.5 degree. A protractor is used to measure the angle, which has measurement accuracy up to 1°. This process has been repeated several times with different camera height, ground distance and matched

with the theoretical values for verification. A table consisting of experimental data for comparing with the theoretical values and percentage of error is presented below.

Table 4.1 Comparison with Experimental and Theoretical Values

SL	Height, H (mm)	Ground Distance, X (mm)	Object size, D (mm)	Experimental Value of ∠BCA	Theoretical Value of ∠BCA	% of Error
1	212.56	1041.40	203.20	11	10.97	0.27
2	359.96	1041.40	203.20	10.5	10.58	0.77
3	548.64	1041.40	203.20	10	9.90	0.98
4	609.60	1041.40	203.20	9.5	9.66	1.65
5	892.52	1041.40	203.20	8.5	8.496	0.04
6	1306.73	1041.40	203.20	7	6.97	0.40
7	609.60	403.46	203.20	16	15.98	0.14
8	609.60	614.94	203.20	13.5	13.48	0.17
9	609.60	759.18	203.20	12	11.98	0.17
10	609.60	1145.54	203.20	9	8.98	0.21
11	609.60	1524.71	203.20	7	7.10	1.34
12	609.60	1844.91	203.20	6	5.99	0.09

From the above table it can be concluded that all the theoretical values and experimental values matched within error of less than 2%. A sophisticated laser source is advised to use in the later part of this section where the angle at a certain height is known. Equation (5.15) can be solved for 'c' while the angle is known and given by,

$$c = \sqrt{2(H^2 + X^2)(1 - cos\gamma)} \tag{5.16}$$

Substituting the values of H, X and cosγ in the Equation (5.16) the size of the segment can be obtained. To verify this method the values of H, X and cosγ are substituted and the size of the segment, 'c' is obtained is 203.2 mm.

A graph is presented in the following figure where, X-axis represents ∠BCA at different heights of camera in degrees and Y-axis represents the ground distance in millimetres (mm). The graph is plotted for different values of H. Red line represents the experimental value of H (609.60mm), purple line is with 10% higher value of H (731.52

mm) from the experimental value and the blue dotted line is with 10% less than the value of H (487.68 mm) from the experimental value.

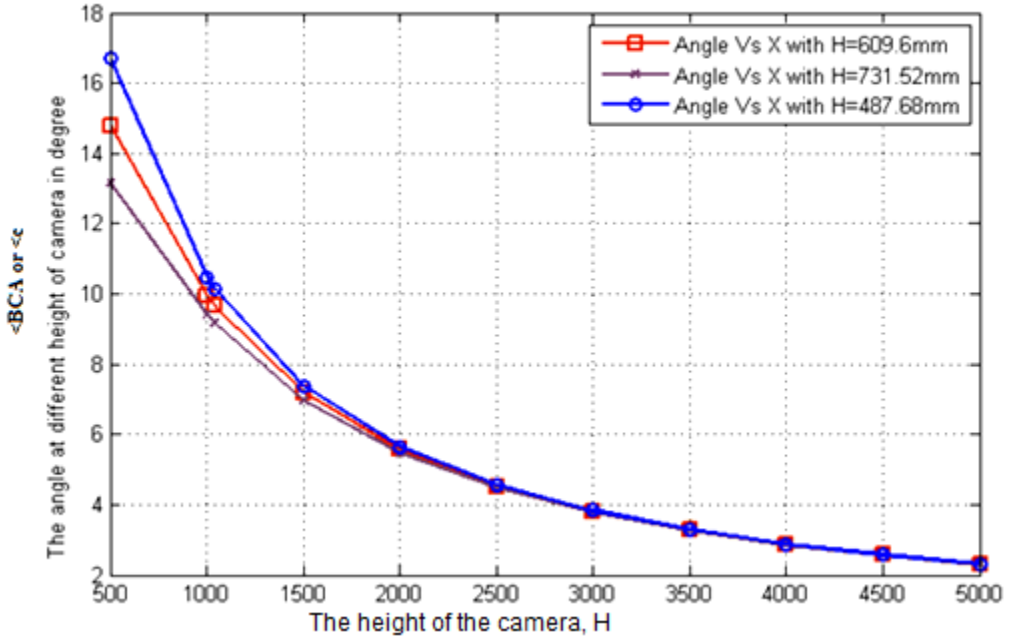


Figure 5.6 Observing Angle Vs. ground distance for three different Heights, H.

Generally for a large height of the camera the observing angle is small as the observing point is placed far from the segment. Similarly, for a large ground distances in between the camera and the segment, observing angle is small as the observing point is located far from the segment. As the ground distance or height of the camera decreases, the observing angle increases for a given fixed segment or vice versa. Experiments are conducted with different camera heights and ground distance to prove this statement. Results for observing angle variation with different ground distances are presented in Figure 5.6.

For the experimental purpose a system of laser grid square in shape is formed from a single laser beam using a diffraction mask and is projected over bottom. A titled camera has been used to capture the surface information. Generally laser footprints on the boundary of the grid tend to move far from each other compare to other footprints for a tilted laser projection over bottom. This leads to a difficult situation to extract information from the pictures correctly. This is the reason to use normal projection of the

laser beam to minimize this effect. A geometrical representation of the project setup is shown in Figure 5.7.

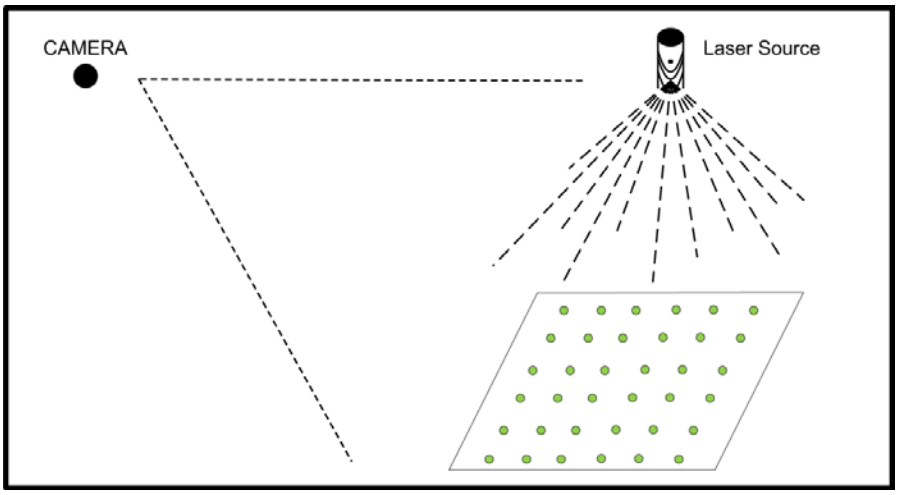


Figure 5.7 Geometrical representation of the experimental setup.

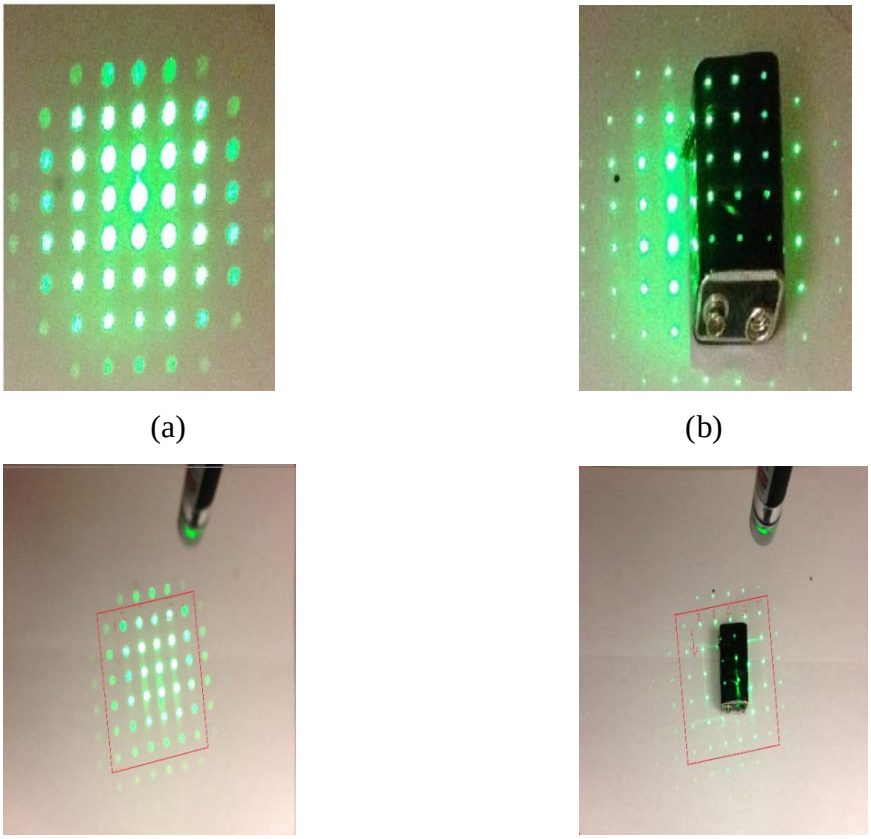
The first image (Figure 5.8 (a)) is taken with the normal projection of the laser beam without any object on the bottom. The second image (Figure 5.8 (b)) is taken with the same laser projection including the object on the bottom. Both of these cases the distance between camera and surface & laser source and surface are kept constant. Figure 5.8 (a) & (b) represent illustration of images taken with the tilted camera.

(a) **(b)**

Figure 5.8 Images taken by titled camera during experiment.

The perspective distortion occurred due to the tilt of the camera is corrected by using homography matrix and projection on the original reference image. Any portion of the image can be selected to correct perspective effect under user flexibility option. It is sufficient to select only the area where the laser beam grid is projected when the objective is to achieve the variation in the bottom. This process truncates the image and

the computation is faster. This algorithm can also find the dimensions of the object and perspective correction is needed with the whole image for that purpose. This is because the dimension of the object is calculated from distance between the laser footprint from the camera and the coordinates of the camera. A crop of the image loses information regarding these points and eventually generates faulty dimensions of the object. In this experiment largest possible area is considered for accurate calculation of the object dimensions. A pictorial representation of these images after correction of the perspective distortion is shown below.



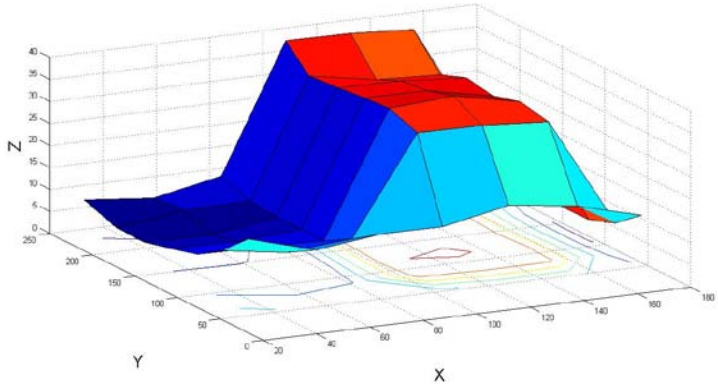
(a) (b)

(c) (d)

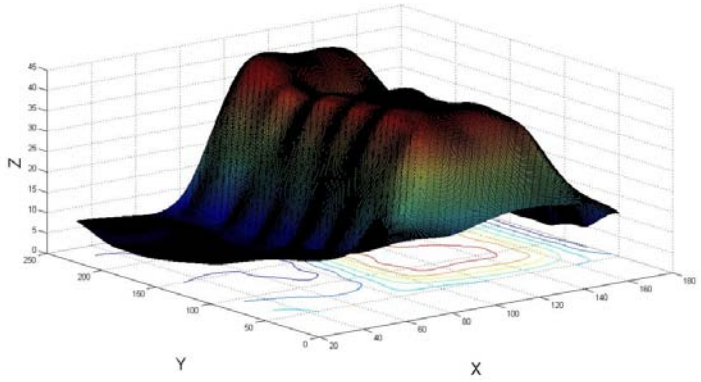
Figure 5.9 (a) & (b) are the output of perspective correction applied on a portion of the images; (c) & (d) are the output of perspective correction applied on the whole images.

Figure 5.9 (a) & (b) are the cases where only the area of the laser beam grid projection is included. Figure 5.9 (a) is used as a reference map and Figure 5.9 (b) is the target image with the object. Figure 5.9 (c) & (d) are the examples where nearly the whole image is included as demonstrated in Figure 5.8. In this case, Figure 5.9 (c) is the reference image and (d) is the target image. Two matrices are created from these images as explained in

the first section of Chapter-5 of this report. These two matrices are compared and computed to estimate the bottom characteristics and detect the object in the bottom. The coordinates of each laser footprints are entered into the algorithm. This information is stored in a '.mat' format file for future uses which reduces the tedious work. Similarly coordinates of each laser footprint from the target image are entered into the algorithm. The overall algorithm stops working along with the matrix calculation to generate a final image if different matrix dimension entered from these two images. Extreme caution is needed to avoid this problem during entering the coordinates of laser footprints from both of the images. The algorithm generates the bottom characteristics once all of these points are entered. Figure 5.10 (a) & (b) shows an example of the output images from the algorithm.



(a)



(b)

Figure 5.10 (a) & (b) Underwater sensing without dimension.

Figure 5.10 (a) is the output image generated by this algorithm using the matrix values from the reference and target images as explained in the first section of this chapter. The algorithm presents a picture of the bottom variation including the object as there is an

object in the target image Figure 5.8 (b). The variation in the bottom due to the object is shown in the output image 5.10 (a) but it doesn't provide any information about the height of the object. Figure 5.10 (b) is the interpolated version of the image in Figure 5.10 (a). Interpolation regime is used to calculate the intermediate points to make the surface smoother as the laser points are not adequately dense. There are four interpolation methods available in MATLAB. They are, linear interpolation, cubic interpolation, nearest interpolation and v4 interpolation. Linear interpolation method is adopted in this experiment as it generates the result faster than the other methods. It is observed from the output images that, this system could not extract the object edges perfectly as the real life object. This is because of the low resolution of the laser footprints.

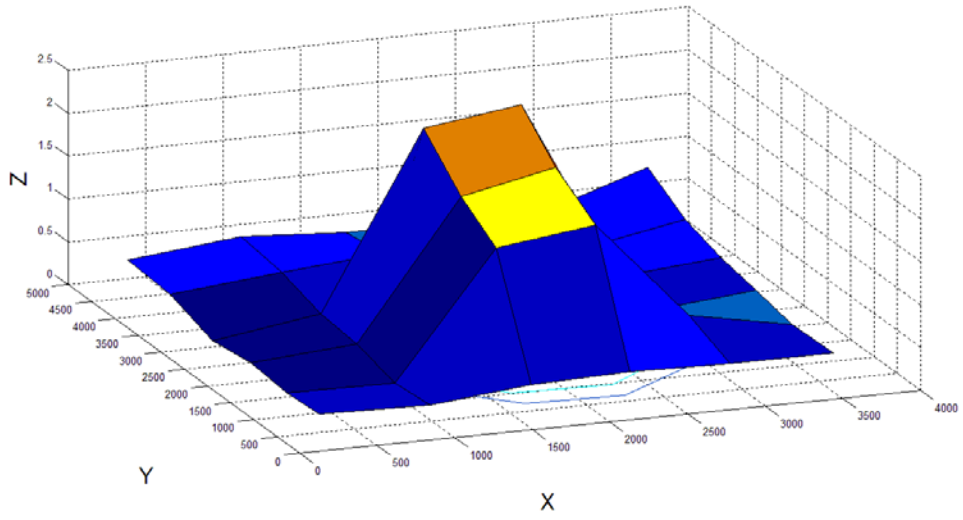


Figure 5.11. Underwater sensing including dimension.

Figure 5.11 is the output image generated by the algorithm with estimated height dimension where nearly the whole image is considered. Here z-axis represents the height. In the following Figure 5.12 the actual height is shown for error calculation.

Figure 5.12 Actual Height of the Object.

From the output image the height of the object is found as 1.45cm and the actual height of the object is 1.5cm. The error in percentage is shown below.

$$\%error = \left|\frac{1.45 - 1.5}{1.5}\right| \times 100\%$$

$$= 3.33\%$$

5.2 Limitation

The only way of extracting information about the surface in this approach is from the laser footprints. For that reason a good amount of laser footprints projected on the bottom is necessary to detect the object. In other words the denser the laser footprints are the better the resolution this algorithm can achieve. Attempts are taken to make this process automated. One of the difficulties faced during that process is explained below. Figure 5.13 is a comparison with two discrete lines of lasers footprints from reference and target images. The left side image of Figure 5.13 is the reference image and the right side one is the target image. Footprints from both of these images are horizontally aligned with each other when there is no object in the surface or for a flat surface. These footprints are shifted when there is a variation in the surface. The algorithm generates matrices from these images with their laser footprints coordinates and estimates the resulting image. This process requires footprint's coordinate from reference image and corresponding footprints coordinate from the target image to compute the resultant matrix. This arise confusion for the computer as shown in Figure 5.13. The computer cannot make the decision to choose correct corresponding points in the target image when there is a shift in the laser footprints due to the presence of an object. The computer faces this ambiguity while choosing the right footprints during the automated process.

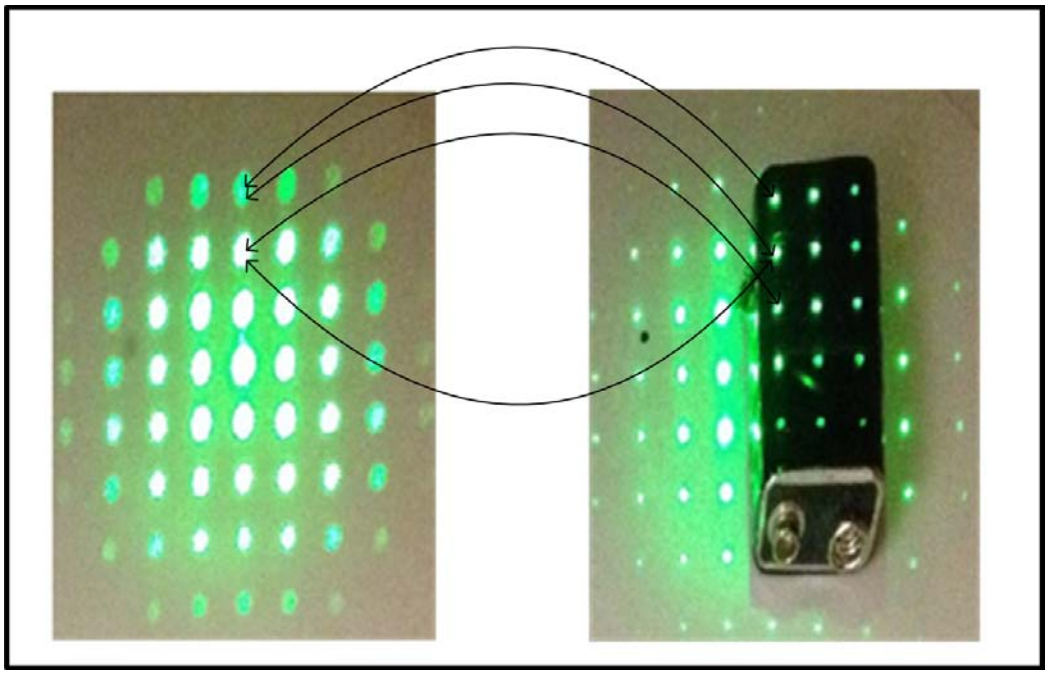


Figure 5.13 Laser footprints comparison from reference and target images.

This system was not able to capture all the edges of the object perfectly due to the low resolution of the laser footprints. Higher number of laser footprints will increase the resolution of the image and hence provides better quality. However, during the experiment process it is important to make sure laser footprints do not overlap with each other.

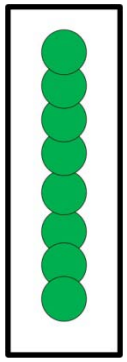
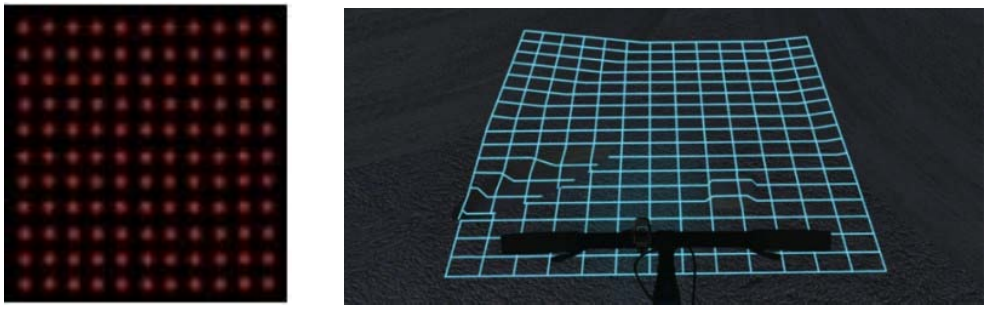


Figure 5.14 Laser footprints overlapping each other.

Figure 5.14 shows the situation with overlapped laser footprints. Images with overlapped footprints create ambiguity during the information extraction process and eventually fail to extract the variation in the bottom.

5.3 Improvement

The proposed algorithm in this project is validated with MATLAB using real life images captured by a still camera. However, there is definitely room for more experiments. It would be very interesting to make this model automated. This will make the overall process less time consuming and useful for analyzing major chunk of data. Another fascinating upgrade will be using a different kind of laser source. The power and features of the lasers used in the laboratory experiment was limited. The only possible way to control the density of the laser beam grid was by moving the source up and down. However, improved results can be obtained by using Diffractive Optical Elements (DOE) for Flexpoint laser modules from the company “Laser Components UK, Ltd". A typical laser array (DOE-258) for this purpose from Laser Components UK, Ltd. is presented below in Figure 5.15 (a). The major advantage of using this laser beam is the constant distance in between all the footprints with higher density.



(a) **(b)**

Figure 5.15 Sample Laser grid created with lasers from Laser Components UK Ltd.

Figure 5.15 (b) is another type of laser source that can be used. One of the major difficulties to automate this algorithm is choosing the right laser footprint. A laser beam projected as Figure 5.16 (b) may make this task simple as the footprints are relatively easy to locate due to the connecting lines in between two points.

5.4 Application

Although the main objective of this approach is to extract the bottom variation and detect underwater natural or manmade objects particularly in shallow or moderately clear water or coastal areas and etc., its uses are not restricted to underwater environment only. Lasers are capable of measuring water depth from 0.2 m down to 60m, depending on the water clarity [27]. Being laser it can also be used in non-underwater environment such as

in dessert or other places to detect objects. In underwater perspective it can be used for dredging operation, oil and gas exploration, underwater construction and bottom object detection like boulders, wracks etc. Laser light has less interference with objects, which makes this technology a unique choice. It is also possible to mount this package in an autonomous underwater vehicle (AUV) or autonomous unmanned undersea vehicle (AUUV). A timer can be set in the camera to capture the photos at predefined intervals and transfer them to the main server remotely for further processing. Although lasers are limited in their operation in the deep sea area however, combining this package (laser source and camera) with AUV will allow fetching bottom information and detecting objects in deep sea bottom.

Chapter 6

6.1 Conclusion

The project proposed a system that extracts information about ocean floor variation and detects underwater natural or manmade objects from the images formed by narrow beam laser projections on the ocean bottom. This system produces accurate results yet reduces computational complexity and it is economical in terms of hardware resources. Lasers themselves are relatively accurate in their ranging capabilities, combined with precise & modern scanning mechanism; this technology can provide accurate information of the bottom. This technology has the ability to capture the floor variation and detect objects where it would be difficult, dangerous, expensive and impossible to use other alternative techniques.

Bibliography

1. United Nations, Status of hydrographic surveying and nautical charting world-wide, *4th United Nations Regional Cartographic Conference for the Americas,* January 23-27, 81/INF/9, 1989.

2. Rebecca Quintal, Jason Infantino, John Shannon Byrne. "Automated side-scan target detection: new tools for hydrography." *Proceedings of the Canadian Hydrographic Conference and National Surveyors Conference,* Vol. 10, No. 03, pp. 01-12, 2008.

3. Alexandru N. Vasile, Richard M. Marino. "Pose-Independent automatic target detection & recognition using 3D laser radar imagery." *Lincoln Laboratory Journal*, Vol. 15, No. 1, pp. 61-78, 2005.

4. Mohan S, Avinash, Murali S. "Rectification of perspective distortion using camera parameters - A perspective geometry based approach", *ICGST International Journal on Graphics, Vision and Image Processing,* Vol. 8, pp.1-7, 2008.

5. Jian Liang, Daniel DeMenthon, David Doermann. "Geometric rectification of camera-captured document images." *IEEE Transactions On Pattern Analysis And Machine Intelligence*, Vol. 30, No.4, pp. 591-605, 2008.

6. Richard I. Hartley. "Theory and practice of projective rectification." *International Journal of Computer Vision*, Vol. 35, No. 2, pp. 115-127, 1999.

7. Paul Clark, Majid Mirmehdi. "Rectifying perspective views of text in 3D scenes using vanishing points." *The Journal of the Pattern Recognition Society,* Vol. 36, No. 11, pp. 2673–2686, 2003.

8. Shijian Lu, Chen B. M., Ko, C. C. "Perspective rectification of document images using fuzzy set and morphological operations," *International Conference on Image Processing*, Vol. 5, pp. 2877-2880, 2005.

9. Shijian Lu, Chew Lim Tan. "The restoration of camera documents through image segmentation," *Proceedings of the Seventh International Association of Pattern Recognition Workshop on Document Analysis Systems*, pp. 484–495, 2006.

10. Ligang Miao, Silong Peng, "Perspective rectification of document images based on morphology," *Proceedings of the International Conference on Computational Intelligence and Security*, Vol. 2, pp. 1805–1808, 2006.

11. Xu-Cheng Yin, Jun Sun, Satoshi Naoi. “Perspective rectification for mobile phone camera-based documents using a hybrid approach to vanishing point detection.” *Proceedings of the Second International Workshop on Camera-Based Document Analysis and Recognition*, pp. 37–44, 2007.

12. Xu-Cheng Yin, Jun Sun, Satoshi Naoi, Fujimoto K., Takebe H., Fujii, Y., Kurokawa, K. “A multi-stage strategy to perspective rectification for mobile phone camera-based document images.” *Proceedings of the Ninth International Conference on Document Analysis and Recognition*, Vol. 2, pp- 574–478, 2007.

13. Masakazu Iwamura, Ryo Niwa, Koichi Kise, Seiichi Uchida, Shinichiro Omachi. “Rectifying perspective distortion into affine distortion using variants and invariants.” *Proceedings of the Second International Workshop on Camera- Based Document Analysis and Recognition*, pp. 138–145, 2007.

14. Donald J. Collins, John A. Bell, Ray Zanoni, I. Stuart McDermid, James B. Breckinridge, Cesar A. Sepulveda. “Recent progress in the measurement of temperature and salinity by optical scattering.” *Proceeding of Photographic Instrumentation Engineers on the Seventh Ocean Optics*. Vol. 489, 1984.

15. Joseph G. Hirschberg, James D. Byrne, Alain W. Wouters, George C. Boynton. “Speed of sound and temperature in the ocean by Brillouin scattering.” *Applied Optics*. Vol. 23, No. 15, pp. 2624–2628, 1984.

16. G. Daniel Hickman, John M. Harding, Michael Carnes, Al Pressman, George W. Kattawar, Edward S. Fry. “Aircraft laser sensing of sound velocity in water: Brillouin scattering.” *Remote Sensing of Environment.* Vol. 36, No. 03, pp. 165–178, 1991.

17. Fry E. S., Emery Y., Quan X. H., Katz J. W. “Accuracy limitations on Brillouin lidar measurements of temperature and sound speed in the ocean.” *Applied Optics*. Vol. 36, No. 27, pp. 6887–6894, 1997.

18. Dahe Liu, Jianfeng Xu, Huaying Wang, Jing Zhou. “Remote sensing to ocean by using Brillouin scattering: Test of sounds speed and submerged object.” *Proceeding of Photographic Instrumentation Engineers on Process Control and Inspection for Industry*. Vol. 4222, No. 114, 2000.

19. Dahe Liu, Jianfeng Xu, Rongsheng Li, Rui Dai, Wenping Gong. “Measurements of sound speed in the water by Brillouin scattering using pulsed Nd:YAG laser.” *Optics Communication.* Vol. 203, No. 3-6, pp. 335–340, 2002.

20. Xu J., Ren X., Gong W., Dai R., Liu D. “Measurement of the bulk viscosity of liquid by Brillouin scattering.” *Applied Optics.* Vol. 42, No. 33, pp. 6704–6709, 2003.

21. Gong W., Dai R., Sun Z., Ren X., Shi J., Li G., Liu D. “Detecting submerged objects by Brillouin scattering.” *Applied Physics B - Laser and Optics.* Vol. 79, pp. 635-639, 2004.

22. Donald A. Leonard, Harold E. Sweeney. “Comparison of stimulated and spontaneous laser-radar methods for the remote sensing of ocean physical properties.” *The Tenth Proceeding of Photographic Instrumentation Engineers in Ocean Optics.* Vol. 1302, pp. 568-582, 1990.

23. Shi J., Li G., Gong W., Bai J., Huang Y., Liu Y., Li S., Liu D. “A lidar system based on stimulated Brillouin scattering.” *Applied Physics B Lasers and Optics.* Vol. 86, pp. 177–179, 2006.

24. Shi J., Ouyang M., Gong W., Li S., Liu D. “A Brillouin lidar system using F–P etalon and ICCD for remote sensing of the ocean.” *Applied Physics B - Lasers and Optics.* Vol. 90, pp. 569–571, 2008.

25. Xiaolu Li, Lijun Xu, Xiangrui Tian, Deming Kong. “Terrain slope estimation within footprint from ICESat/GLAS waveform: model and method.” *Journal of Applied Remote Sensing.* Vol. 6, No. 1, 2012.

26. Xiaolu Li, Lijun Xu, Changwei Wang. “Terrain slope calculation from waveform of airborne LIDAR.*” IEEE International Conference on Geoscience and Remote Sensing Symposium (IGARSS).* pp. 6259-6262, 2012.

27. Chi-Kuei Wang, William D. Philpot. “Using airborne bathymetric lidar to detect bottom type variation in shallow waters.” *Remote Sensing of Environment.* Vol. 106, pp. 123-135, 2007.

28. Chris Harris, Mike Stephens. “A Combined Corner and edge detector.” *The British Machine Vision Association and Society for Pattern Recognition.* Vol. 2, pp. 147-152, 1988.

Appendix

File Name: Main_Sabuj.m

```
% Construct a 3D model from Laser Dot Points

close all;
clear all;
clc;

% % Read Two Images and save them in .mat files:

Reference_Image = imread('D:\3d\SabujV01\photo\05\Ref_1.bmp');      %
Change the File Path
Reference_Image = rgb2gray(Reference_Image);

Reference_Image = uvic('D:\3d\SabujV01\photo\05\Ref_1.bmp');      %
Perspective Correction

figure(1);
imshow(Reference_Image);
set(gcf,'outerposition',get(0,'screensize'));
title('Reference Image','Fontsize',20);

[Map_X,Map_Y] = ginput;
save('d:\3d\SabujV01\Data\ref.mat','Map_X','Map_Y');

Data_Image = imread('D:\3d\SabujV01\photo\05\Target_1.bmp');        %
Change the File Path
Data_Image = rgb2gray(Data_Image);

Data_Image = uvic('D:\3d\SabujV01\photo\05\Target_1.bmp');       %
Perspective Correction
%
 figure(2);
 imshow(Data_Image);
 set(gcf,'outerposition',get(0,'screensize'));
 title('Data Image','Fontsize',20);

 [Data_X,Data_Y] = ginput;
 save('d:\3d\SabujV01\Data\box.mat','Data_X','Data_Y');

%-----------------------------------------------------------
% Create the input matrix - Z:
load('d:\3d\SabujV01\Data\ref.mat','Map_X','Map_Y');      % Get Map_X
and Map_Y
load('d:\3d\SabujV01\Data\box.mat','Data_X','Data_Y');      % Get
Data_X and Data_Y
[row,col] = size(Data_Image);

X_Camera = 0.5 * col;
Y_Camera = row;
Z_Camera = 31;
```

```
d_1 = sqrt(power((Data_X - Map_X),2) + power((Data_Y - Map_Y),2));
d_2 = sqrt(power((Map_X - X_Camera),2) + power((Map_Y - Y_Camera),2));
Z = Z_Camera.*(d_1./(d_1 + d_2));

%-------------------------------------------------------------
% Interpolate the X, Y and Z to get a better resolution

[X_New,Y_New,Z_New] = griddata(Map_X, Map_Y, Z,...

linspace(min(Map_X),max(Map_X),2*(max(Map_X) - min(Map_X) + 1))',...

linspace(min(Map_Y),max(Map_Y),2*(max(Map_Y) - min(Map_Y) + 1)),...
                               'linear');

%-------------------------------------------------------------
% Draw the 3D model without Interpolation
figure(3);

surfc(reshape(Map_X,6,6),reshape(Map_Y,6,6),reshape(Z,6,6));
set(gcf,'outerposition',get(0,'screensize'));

title('3D Model Before Interpolation','Fontsize',20);
xlabel('X','Fontsize',20);
ylabel('Y','Fontsize',20);
zlabel('Z','Fontsize',20);

%-------------------------------------------------------------
% Draw the 3D model after Interpolation
figure(4);

surfc(X_New,Y_New,Z_New);
set(gcf,'outerposition',get(0,'screensize'));

title('3D Model after Interpolation','Fontsize',20);
xlabel('X','Fontsize',20);
ylabel('Y','Fontsize',20);
zlabel('Z','Fontsize',20);

%-------------------------------------------------------------

t1 = toc;

Time_Minute = floor( t1 / 60 );
Time_Second = mod( t1, 60 );

disp(['Calculation Time: ',num2str(Time_Minute),' min
',num2str(Time_Second),' s.']);
```

File Name: Usage_AssistedMode.m (Perspective Distortion Correction)

```
clear all;
close all;
clc;

targetFolder = 'D:/images/';
candidateImages = {'ref.jpg', 'skewed1.gif', 'board.jpg',
'chessboard.jpg', 'chocolate.jpg', 'book.jpg', 'square.jpg',
'card.jpg', 'image3.tif', 'image16.tif'};
candidateChoice = 1;

targetImageFile = strcat(targetFolder ,
char(candidateImages(candidateChoice ))); % gets where the images is
placed and the images name.
targetImageData = imread(targetImageFile); % converts the image file
with codes.

imshow(targetImageData); % it shows the target image.

fprintf('Corner selection must be clockwise or anti-clockwise.\n');
[X Y] = ginput(4); %enables the selection window to open for selection
of four corner points.

[X Y] = sortPolyFromClockwiseStartingFromTopLeft( X, Y );

x=[1;210;210;1];
y=[1;1;297;297];

A=zeros(8,8);
A(1,:)=[X(1),Y(1),1,0,0,0,-1*X(1)*x(1),-1*Y(1)*x(1)];
A(2,:)=[0,0,0,X(1),Y(1),1,-1*X(1)*y(1),-1*Y(1)*y(1)];

A(3,:)=[X(2),Y(2),1,0,0,0,-1*X(2)*x(2),-1*Y(2)*x(2)];
A(4,:)=[0,0,0,X(2),Y(2),1,-1*X(2)*y(2),-1*Y(2)*y(2)];

A(5,:)=[X(3),Y(3),1,0,0,0,-1*X(3)*x(3),-1*Y(3)*x(3)];
A(6,:)=[0,0,0,X(3),Y(3),1,-1*X(3)*y(3),-1*Y(3)*y(3)];

A(7,:)=[X(4),Y(4),1,0,0,0,-1*X(4)*x(4),-1*Y(4)*x(4)];
A(8,:)=[0,0,0,X(4),Y(4),1,-1*X(4)*y(4),-1*Y(4)*y(4)];

v=[x(1);y(1);x(2);y(2);x(3);y(3);x(4);y(4)];

u=A\v;

U=reshape([u;1],3,3)';

w=U*[X';Y';ones(1,4)];
w=w./(ones(3,1)*w(3,:));

T=maketform('projective',U');
```

```
P2=imtransform(targetImageData,T,'XData',[1 210],'YData',[1 297]);

figure,
imshow(P2)
```

File Name: sortPolyFromClockwiseStartingFromTopLeft.m

```
function [X, Y] = sortPolyFromClockwiseStartingFromTopLeft( X, Y )

% The 1st 2 high values for the y-axis are the top 2 edges
% <upper corners identified>
%top_vertices_Y = sort(Y,2);
top_vertices_Y = sortCoordinatesAccordToX(Y);

    for i=1:4
        for j=1:4
            if top_vertices_Y(i) == Y(j)
                top_vertices_X(i) = X(j);
            end
        end
    end

   top_vertices_X = top_vertices_X';
   X = top_vertices_X;
   Y = top_vertices_Y;

   % The larger of the x values for the 1st 2 high values for the y-
axis
   % belongs to the top-right hand corner
   % <upper left and right corners identified>
   if X(1) > X(2)
       top_vertices_X(1) = X(2);
       top_vertices_Y(1) = Y(2);
       top_vertices_X(2) = X(1);
       top_vertices_Y(2) = Y(1);
   end

   X = top_vertices_X;
   Y = top_vertices_Y;

% The larger of the x values for the last 2 high values for the y-axis
% belongs to the bottom-right hand corner
   % <lower left and right corners identified>
   if X(3) < X(4)
       top_vertices_X(3) = X(4);
       top_vertices_Y(3) = Y(4);
       top_vertices_X(4) = X(3);
       top_vertices_Y(4) = Y(3);
   end

   X = top_vertices_X;
   Y = top_vertices_Y;
end
```