University of Kentucky Master's Theses

Graduate School

2008

Motion Correction Structured Light using Pattern Interleaving Technique

Raja Kalyan Ram Cavaturu
University of Kentucky, rajacavaturu@gmail.com

Recommended Citation

Cavaturu, Raja Kalyan Ram, "Motion Correction Structured Light using Pattern Interleaving Technique" (2008). *University of Kentucky Master's Theses*. 551.
https://uknowledge.uky.edu/gradschool_theses/551

ABSTRACT OF THESIS

Motion Correction Structured Light using Pattern Interleaving Technique

Phase Measuring Profilometry (PMP) is the most robust scanning technique for static 3D data acquisition. To make this technique robust to the target objects which are in motion during the scan interval a novel algorithm called ‘Pattern Interleaving’ is used to get a high density single scan image and making Phase Measuring Profilometry insensitive to ‘z’ motion and prevent motion banding which is predominant in 3D reconstruction when the object is in motion during the scan time

KEYWORDS: Structured Light Illumination, Phase Measuring Profilometry, Pattern Interleaving, Motion banding and 3D reconstruction.

Raja Kalyan Ram Cavaturu

September 8th, 2008

Motion Correction Structured Light using Pattern Interleaving Technique

By

Raja Kalyan Ram Cavaturu

Dr. Laurence G Hassebrook
Director of Thesis

Dr. YuMing Zhang
Director of Graduate Studies

September 8th, 2008

RULES OF THE THESIS

Unpublished theses submitted for the Master's degree and deposited in the University of Kentucky Library are as a rule open for inspection, but are to be used only with due regard to the rights of the authors. Bibliographical references may be noted, but quotations or summaries of parts may be published only with the permission of the author, and with the usual scholarly acknowledgments.

Extensive copying or publication of the thesis in whole or in part also requires the consent of the Dean of the Graduate School of the University of Kentucky.

A library that borrows this thesis for use by its patrons is expected to secure the signature of each user.

Name
Date

THESIS

Raja Kalyan Ram Cavaturu

The Graduate School

University of Kentucky

2008

Motion Correction Structured Light using Pattern Interleaving Technique

THESIS

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering in the College of Engineering at the University of Kentucky

By

Raja Kalyan Ram Cavaturu
Lexington, Kentucky

Director: Dr. Laurence G. Hassebrook,
Department of Electrical Engineering
Lexington, Kentucky

2008

Dedicated to

My family

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my sincere heart-felt gratitude to my advisor, Dr.Laurence G. Hassebrook for his encouragement, constant guidance and valuable suggestions and the freedom of thought at every stage of the research work.

I would also like to thank Dr. Kevin Donohue and Dr. Ruigang Yang for serving on my defense committee. I would like to thank Charles Casey and Akshay Pethe for all their cooperation and help. I would like to thank my friends Sandeep Boddapati, Sunil Boddapati, Sudheer Garimella and Vamsi Vankadara for their support.

Finally, I would like to thank my family, without them I would never have made it here. My father, Mr.Madhava Rao Cavaturu for giving me constant encouragement and motivation at every step of my life. My mother, Mrs Vijaya Lakshmi Cavaturu for her caring love, encouragement and support, without her I would not be the person as I am today. My brother, my mentor, Mr. Kiran Cavaturu for his guidance and constant support. I am really fortunate to have a brother like him. My sister-in-law, Mrs.Aparna Cavaturu for treating me like her son. My friend, Swetha Mannepalli for her patience and constant encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FILES

Thesis.pdf (Acrobat Reader file) – 1.7 MB (File Size)

Chapter 1 Introduction

3D data acquisition techniques are broadly classified into two types: active and passive. The necessary condition for both active and passive 3D data acquisition techniques is optical triangulation. The most prominent scanning feature technique in the passive 3D data acquisition is Stereo Vision (StV). In stereo vision technique, the optical triangulation is established between the target object and an array of cameras. In stereo vision techniques, the 3D reconstruction of the target object is achieved by finding the correspondence between the images viewed from two or more points of view (POV). However, the main problem associated with the stereo vision is the one relating to the feature correspondence. This correspondence problem in the stereo vision technique is solved by using active methods [1] but is very dependent on the object having distinctive features. The most widely used scanning feature technique in active methods is structured light illumination (SLI). It has its applications in different fields like biomedical topology [2], quality control [3] and telecollaboration [4]. In the Structured Light Illumination (SLI) technique optical triangulation is established by replacing one of the cameras in the stereo vision technique with a projector. Thus, a geometric relationship between the cameras, projector and the target object is established. In a SLI technique, a coded light pattern is projected by the projector on to the target object which is captured by the camera. Depth information can be obtained by measuring the distortion occurred between the captured image and the reflected image. Various coded light patterns like binary, single spot, stripe or a complex pattern can be used. A commonly used SLI scanning technique is Phase Measuring Profilometry (PMP).

The SLI methodology overcomes the feature correspondence problem associated with the StV by projecting structured patterns. The advantages of using multiple patterns are [6]

1. Depth resolution is determined by number of patterns
2. Non ambiguous depth measurement over long ranges
3. Insensitive to ambient light interference
4. Insensitive to surface shading or color
5. Spatial resolution is determined by camera resolution.

The main drawback of using multi pattern SLI is that it takes more time to scan the target object and thus not suitable for the target objects which are in motion during the scan time. To overcome this drawback, a single pattern technique [7] is used for the dynamic scenes but this technique is not as accurate as the multi pattern technique.

By using multi frequency PMP technique higher accuracy and precision of 3D data can be obtained for a static target object [11]. Chun Guan et al introduced a new technique called composite pattern (CP) by using the multi frequency PMP patterns for scanning dynamic scenes [12]. In this technique, multiple PMP patterns are combined to form a single pattern by using communication theory concepts. However, this technique suffers from low depth resolution and also carrier frequency detection in the captured pattern.

Hassebrook et al [6] proposed a new method, “Lock and Hold strategy” to track the object motion. In this technique, a ‘lock’ state can be obtained by using methods like multiple patterns PMP or single pattern or successive boundary subdivision to determine the snake identity. The phase value of each snake is obtained, thereby getting the phase value of the camera coordinates. In the hold state a single multi frequency sine wave is used and the snake process is used to track the hold state. However it is difficult for certain target objects like human hand to be in standstill state in the ‘lock’ state.

Song Zhang [29] proposed a novel algorithm called ‘Fast three step phase shift algorithm’ which must have a high speed projector and a camera. Hall-Holt [8] proposed a new method for compensating motion. In this technique motion compensation is achieved by tracking the stripe boundaries. Wiese [9] presented a 3D scanning system which uses both structured light and stereo vision and proposed a stereo phase shift method for motion compensation. In this method, the motion compensation is performed on each pixel by analyzing the motion error. Soren Konig [10] proposed a novel motion compensation technique where two additional patterns (on and off patterns) are introduced between the structured light patterns to track the motion.

The main aim of the research work is to compensate for the error that occurred in 3D reconstruction due to the motion of the target object during the scan time. In our method, we interleave a single pattern in between each SLI pattern. We then track motion of these interleaved patterns and correct the SLI pattern set for object motion during the scan. We call this “Pattern Interleaving (PIL)”

This technique uses the traditional PMP technique, a robust algorithm for static 3D scanning, thereby making the traditional PMP insensitive to motion.

1.1 Thesis Organization

This thesis is organized into six chapters, chapter one gives the introduction to the research work and the objective of the research work. Chapter two gives the description of SLI technique, description of multi frequency Phase Measuring Profilometry (PMP) is presented, the importance of the calibration and a mathematical relationship between the camera coordinates and the world coordinates is presented. Chapter three introduces a new technique, Pattern Interleaving (PIL), which is used to compensate the ‘z’ motion during the scan time. Chapter four presents an approach to compensate the lateral movement of the object during the scan time. Chapter five explains the experimental results of the 3D reconstruction of the objects obtained by using PIL technique. Chapter six is a concluding chapter with a few insights of further research.

Chapter 2 Background

We discuss the structured light illumination technique and its advantages. The commonly used SLI technique, Phase Measuring Profilometry, is discussed followed by advantages of calibrating the system and the different calibration techniques used that take distortion models into account and mathematical relationship between the camera coordinates and world coordinates and the projector coordinates is presented.

2.1 Structured Light Illumination Technique

One of the most important scanning methodologies used in the active methods for 3D shape measurement is SLI. Unlike the passive scanning methods like stereovision, SLI overcomes the fundamental ambiguities [12] and it is also simpler and has high precision. And another advantage is the cost of using SLI techniques is low and we can achieve high speeds. A SLI technique consists of a camera and a projector as shown in the Figure 2.1

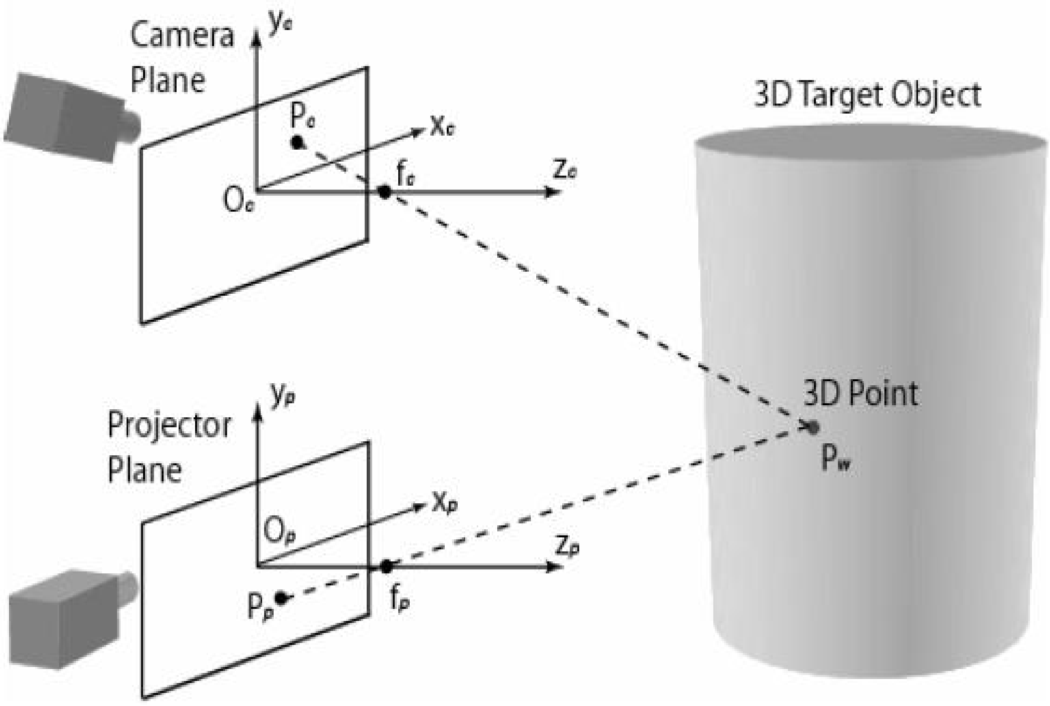


Figure 2.1 Structured Light Illumination (SLI) setup [27]

The projector projects coded light patterns such as stripes, binary codes and gray codes etc. on to the target object and the deformed patterns are captured by the camera. As shown in the Figure 2.1, the camera, the projector and the target object have to form a triangle to detect the deformation. As the captured images are encoded with projector coordinates, by decoding the captured images the correspondence matching can be obtained.

2.2 Multi Frequency Phase Measuring Profilometry

Phase Measuring Profilometry (PMP) is one of the most accurate and robust scanning SLI techniques used for 3D reconstruction. In a SLI PMP technique, the projector projects the shifted sinusoidal patterns on to the target object which are expressed as [5]

$$I_n\left(x^p, y^p\right)=A^p + B^p \cos\left(2\pi f y^p - {2\pi n}/{N}\right) \tag{2.1}$$

where $\left(x^p, y^p\right)$ are the projector coordinates, A^p and B^p are the constants of the projector, f is the frequency of the sine wave and n is the phase shift index and N represents the total number of sine wave patterns.

The deformed projected images are captured by the camera, which is expressed as

$$I_n\left(x^c, y^c\right)= A\left(x^c, y^c\right)+ B\left(x^c, y^c\right)\cos\left(\phi\left(x^c, y^c\right)- {2\pi n}/{N}\right) \tag{2.2}$$

where $\left(x^c, y^c\right)$ are the camera co-ordinates and $\phi\left(x^c, y^c\right)$ is the phase of the pixel location $\left(x^c, y^c\right)$ and can be calculated as

$$\phi\left(x^c, y^c\right)= \arctan\left[\frac{\sum_{n=1}^{N} I_n\left(x^c, y^c\right)\sin\left(\frac{2\pi n}{N}\right)}{\sum_{n=1}^{N} I_n\left(x^c, y^c\right)\cos\left(\frac{2\pi n}{N}\right)}\right] \tag{2.3}$$

The projector co-ordinate can be calculated from the phase obtained in the Eq (2.3) as

$$y^p = {\phi\left(x^c, y^c\right)}/{2\pi f} \tag{2.4}$$

Hence with the help of y^p and $\phi\left(x^c, y^c\right)$ 3D world coordinates can be obtained.

In the single frequency PMP, the accuracy of the depth measurement is directly proportional to the number of the shifted sine wave patterns used and the spatial frequency. However, as the spatial frequency increases the ambiguity error increases. To solve this, Chun and Yalla et al proposed multi frequency PMP [5] [12] which is an extension of the single frequency PMP. In the multi frequency PMP, N number of frequencies can be used and the total number of shifted sine wave patterns projected is constant.

The multi frequency PMP algorithm is described below

1. Project the base frequency PMP and capture it
2. Calculate the phase at each pixel of the captured image using Eq. (2.3). This serves as the base frequency for the higher frequencies
3. Repeat the following steps until all the higher frequencies are projected
   3.1 capture the higher frequency deformed patterns
   3.2 calculate the phase using Eq. (2.3) and the phase value lies in the range of 0-2π
   3.3 unwrap the phase obtained above using the base frequency obtained in step 2 and subtract π to bring it in the range of (-π, π]. This new phase is used to unwrap the phase for the next higher frequency.
4. Calculate y^p from the phase obtained above and find the 3D world coordinates

2.3 Calibration

Camera calibration plays a prominent role in 3D data acquisition process. With the help of camera calibration 3D depth information can be extracted. The main aim of the camera calibration is to form a relationship between the target object, projector and the camera. Thus, a mathematical relationship between the world coordinates, projector coordinates and the camera coordinates have to be established. This mathematical relationship is affected by two parameters intrinsic parameters and the extrinsic parameters. Intrinsic parameters are mainly related to the camera characteristics like focal length, optical center, pixel scale factors and the distortion parameters. Extrinsic parameters are those which describe the position of camera coordinate system with respect to the world coordinate system. Thus, camera calibration is the process of estimating the above said parameters. In general, if the camera is calibrated accurately, the error in 3D reconstruction is minimal.

2.3.1 Lens Distortions:

Lens distortions, which are occurred by optical aberrations, play a prominent role in the 3D reconstruction so it is very important to take these distortions into consideration when performing calibration. Lens distortions are of two types

1. Radial distortion
2. Perspective distortion.

Radial distortion: Radial distortion occurs when the image points are distorted in the radial direction from the optical center. Depending upon the radial direction from the optical center, radial distortion is classified into two types

1. Barrel distortion
2. Pincushion distortion

When the image points move towards the optical center along the radial direction then that distortion is called “Barrel distortion”. If the image points move away from the optical center along the radial direction then that distortion is called “Pincushion distortion”.

The barrel distortion and the pincushion distortion of a rectangular grid are shown in Figure 2.2 .The solid line is the original rectangular object and dotted lines represent the distortion of the rectangular object.

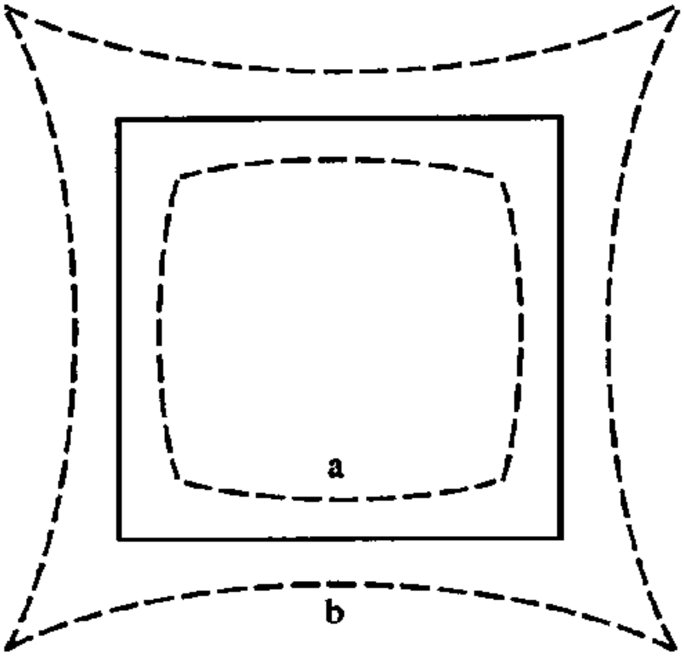


Figure 2.2 Radial Distortion (a) Barrel distortion (b) Pincushion distortion. [19]

Perspective Distortion: Perspective distortion in an image occurs when the distance between the object and the lens is changed. The perspective distortion of a rectangular grid is shown in Figure 2.3

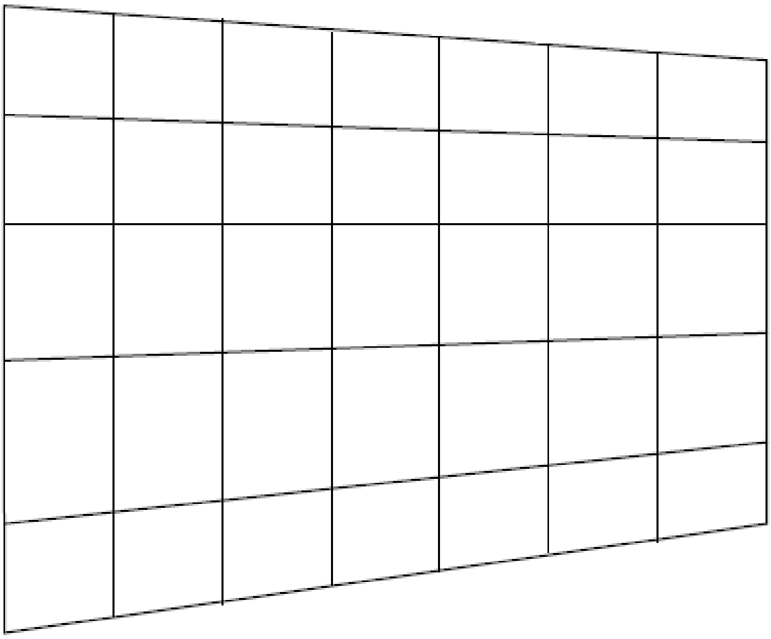


Figure 2.3 Perspective distortion of a rectangular grid [28]

As already mentioned, higher the accuracy of the calibration of camera parameters the better is the 3D reconstruction. So these distortion parameters play a huge role in 3D reconstruction.

Hall [14] proposed a camera calibration technique, in which the transformation matrix is achieved by using linear techniques. O.D.Faugeras and G.Toscani [15] proposed a solution to estimate the camera parameters by considering two cases, with and without the knowledge of 3D world coordinates. In the first case, camera parameters are obtained by using linear least squares approach. In the second case, the camera parameters are obtained by matching features and recursively performing kalman filtering to estimate the parameters. However, the main disadvantage of using these linear techniques is higher accuracy can’t be achieved when the distortion parameters are taken into consideration. Brown [16] proposed the plumb line method to calibrate the lens distortions (radial and tangential). Salvi[17] proposed a non linear optimization technique to calibrate the camera by considering the distortion parameters. Higher accuracy can be achieved by using this technique but the initial guess for the iterative algorithm to achieve convergence is the main limitation to this technique. Tsai [18] proposed a new calibration technique that considers the lens distortion. In this method, camera parameters (intrinsic and extrinsic) are estimated by single view of coplanar and non-coplanar points.

Weng et al [19] proposed a two-step calibration technique, in the first step camera parameters are estimated by using linear methods without considering the distortion and in the second step these estimated camera parameters are iterated through non linear optimized techniques by considering distortion parameters. The initial guess for the second step is obtained from the step one. Zhang [13] presented a new calibration technique, the camera parameters are estimated by observing a planar pattern at different orientations and using a closed form solution. The nonlinear refinement, by considering radial distortion, of the camera parameters is done by maximum likelihood criteria.

Wang et al [20] proposed a new calibration model for lens distortion. The basic idea of this model lies in mathematically expressing the decentering and tilt distortion in a transform consisting of rotation and translation. Thus, this transform is described as two angular parameters and two linear parameters. The two angular parameters describe the pose of the sensor array plane with respect to ideal image plane and the two linear parameters describe location of the sensor array with respect to optical axis.

Guangjun Zhang et al [21] proposed a new calibration algorithm for radial distortion based on cross ratio invariability of the perspective projection. De Xu et al[22] proposed a new calibration method to correct the large lens distortions using a planar grid pattern. In this method, an iterative algorithm is used initially to adjust the distortion parameters and later the camera parameters are estimated when the distortion parameters are adjusted. Zhengyou Zhang [23] presented a new calibration technique by considering the epipolar geometry between the two images having lens distortion. This method is based on the idea that a point in one image and the corresponding point in another image should lie on an epipolar curve instead of straight lines (which is the case for distortionless models). Basing on this epipolar constraint the distortion parameters and the camera parameters are estimated. Lili Ma et al[24] presented a piece wise radial distortion model correction for the camera calibration. In this technique the distortion parameters are solved analytically. Simone Graf and Tobias Hanning [25] also presented a method where the camera parameters can be solved analytically.

Frederic Devernay and Faugeras [26] proposed an automatic calibration for the distortion. This method is based on the idea, that a projection of every line in a space on to the camera is a line if the camera is modeled as a pin hole model. This method doesn’t require any calibration object and it requires images of scenes containing 3D segments. Edge extraction is performed initially on the distorted video sequence followed by a polygonal approximation to extract lines and then finding the distortion parameters that transfers edges to segments.

2.3.2 Mathematical Relationship between the camera coordinates, world coordinates and projector coordinates

As already explained, camera calibration is the process of establishing relationship between the camera coordinates and the world coordinates and also establishing the relationship between the world coordinates and projector coordinates. Camera calibration is achieved by estimating intrinsic and extrinsic parameters. Extrinsic parameters are mainly dependent on the camera position and camera orientation with respect to the world frame and intrinsic parameters are mainly dependent on the camera internal characteristics such as focal length, scale factors etc.

The transformation of the world coordinates and the camera coordinates is achieved by the following four steps [5] [15] [18][27]

1. Rigid body transformation
2. Projective transformation
3. Lens Distortion
4. Mapping from the camera coordinates to image pixel location

Let $\left(x^c, y^c\right)$ be the camera image coordinates, $\left(X^w, Y^w, Z^w\right)$ be the world coordinates and $\left(X^c, Y^c, Z^c\right)$ be the 3D camera coordinates.

Rigid body transformation:

The rigid body transformation is given as

$$\begin{bmatrix} X^c \\ Y^c \\ Z^c \end{bmatrix} = R_{3\times3} \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix} + T_{3\times1} \tag{2.6}$$

where $R_{3\times 3}$ is the rotation matrix and $T_{3\times 1}$ is the translation matrix are the extrinsic parameters given as

$$R_{3\times 3} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$T_{3\times 1} = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}$$

Perspective projection:

Let us consider $\left(X^u, Y^u\right)$ is the undistorted image coordinates and f is the focal length of the camera. By considering the camera as a pin-hole model, the perspective projection is

$$X^u = f\left(\frac{X^c}{Z^c}\right) \qquad (2.7a)$$

$$Y^u = f\left(\frac{Y^c}{Z^c}\right) \qquad (2.7b)$$

Lens Distortion:

Let $\left(X^d, Y^d\right)$ be the undistorted image coordinates, by considering the radial distortion into account we get

$$X^u = X^d + D_X \qquad (2.8a)$$

$$Y^u = Y^d + D_Y \qquad (2.8b)$$

D_X is the distortion in x direction defined as

$$D_X = X^d\left(k_1 r^2 + k_2 r^4 + \ldots\right) \qquad (2.9a)$$

D_Y is the distortion in y direction defined as

$$D_Y = Y^d\left(k_1 r^2 + k_2 r^4 + \ldots\right) \qquad (2.9b)$$

where k_1 and k_2 are the radial parameters and r is given as

$$r = \sqrt{\left(\left(X^{d}\right)^{2} + \left(Y^{d}\right)^{2}\right)} \tag{2.10}$$

Mapping from the camera coordinates to image pixel location:

Finally the camera coordinates measured are mapped to the camera image coordinates in the frame buffer as

$$\begin{bmatrix} x^{c} \\ y^{c} \end{bmatrix} = \begin{bmatrix} S_X & 0 \\ 0 & S_Y \end{bmatrix} \begin{bmatrix} X^{d} \\ Y^{d} \end{bmatrix} + \begin{bmatrix} x^{c0} \\ y^{c0} \end{bmatrix} \tag{2.11}$$

where x^{c0}, y^{c0} are the constant offsets in x and y directions respectively, S_X and S_y are the scale factors in x and y directions respectively

These parameters, focal length, radial parameters, scale factors and constant offsets, are intrinsic parameters.

Thus, the transformation of world coordinates to the camera coordinates is expressed as

$$\begin{bmatrix} s.x^{c} \\ s.y^{c} \\ s \end{bmatrix} = \begin{bmatrix} S_X & 0 & x^{c0} \\ 0 & S_Y & y^{c0} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X^{w} \\ Y^{w} \\ Z^{w} \\ 1 \end{bmatrix} \tag{2.12}$$

where s is the scale factor

Thus Eq. (2.12) can be expressed as

$$\begin{bmatrix} s.x^{c} \\ s.y^{c} \\ s \end{bmatrix} = M_{3\times4} \begin{bmatrix} X^{w} \\ Y^{w} \\ Z^{w} \\ 1 \end{bmatrix} \tag{2.13}$$

where $M_{3\times4}$ is called the perspective transformation matrix and it can be also written as

$$M_{3\times4} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \tag{2.14}$$

Thus, Eq. (2.12) becomes

$$\begin{bmatrix} s.x^c \\ s.y^c \\ s \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_W \\ Z_W \\ 1 \end{bmatrix} \qquad (2.15)$$

Thus, the calibration procedure which is used to estimate the intrinsic and extrinsic parameters is obtained by estimating the perspective transformation matrix.

As the optical models for both projector and the camera are same, the calibration procedure for a structured light illumination system is the calculation of perspective matrices of camera and projector which is given as

$$M^{wc}_{3\times4} = \begin{bmatrix} m^{wc}_{11} & m^{wc}_{12} & m^{wc}_{13} & m^{wc}_{14} \\ m^{wc}_{21} & m^{wc}_{22} & m^{wc}_{23} & m^{wc}_{24} \\ m^{wc}_{31} & m^{wc}_{32} & m^{wc}_{33} & m^{wc}_{34} \end{bmatrix} \qquad (2.16)$$

$$M^{wp}_{3\times4} = \begin{bmatrix} m^{wp}_{11} & m^{wp}_{12} & m^{wp}_{13} & m^{wp}_{14} \\ m^{wp}_{21} & m^{wp}_{22} & m^{wp}_{23} & m^{wp}_{24} \\ m^{wp}_{31} & m^{wp}_{32} & m^{wp}_{33} & m^{wp}_{34} \end{bmatrix} \qquad (2.17)$$

From Eq. (2.15) the camera coordinates cane be written as

$$x_c = \frac{m^{wc}_{11} X_W + m^{wc}_{12} Y_W + m^{wc}_{13} Z_w + m^{wc}_{14}}{m^{wc}_{31} X_W + m^{wc}_{32} Y_W + m^{wc}_{33} Z_W + m^{wc}_{34}} \qquad (2.18)$$

$$y_c = \frac{m^{wc}_{21} X_W + m^{wc}_{22} Y_W + m^{wc}_{23} Z_w + m^{wc}_{24}}{m^{wc}_{31} X_W + m^{wc}_{32} Y_W + m^{wc}_{33} Z_W + m^{wc}_{34}} \qquad (2.19)$$

This perspective matrix has 11 independent variables and it is computed by two most prominent techniques

1. Singular Value Decomposition(SVD) technique
2. Least squares solution technique

Singular Value Decomposition (SVD) technique: [5]

By rearranging the Eq. (2.18) and Eq. (2.19) we can write in a linear form as

$$A^c_{2M\times12} m^c = 0$$

where $A^c_{2M\times12}$ is given as

$$A^c_{2M\times12}=\begin{bmatrix} X^w_1 & Y^w_1 & Z^w_1 & 1 & 0 & 0 & 0 & 0 & -x^c_1X^w_1 & -x^c_1Y^w_1 & -x^c_1Z^w_1 & -x^c_1 \\ 0 & 0 & 0 & 0 & X^w_1 & Y^w_1 & Z^w_1 & 1 & -y^c_1X^w_1 & -y^c_1Y^w_1 & -y^c_1Z^w_1 & -y^c_1 \\ X^w_2 & Y^w_2 & Z^w_2 & 1 & 0 & 0 & 0 & 0 & -x^c_2X^w_2 & -x^c_2Y^w_2 & -x^c_2Z^w_2 & -x^c_2 \\ 0 & 0 & 0 & 0 & X^w_2 & Y^w_2 & Z^w_2 & 1 & -y^c_2X^w_2 & -y^c_2Y^w_2 & -y^c_2Z^w_2 & -y^c_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\cdots\cdots & \cdots\cdots & \cdots\cdots\cdots & \cdots \\ X^w_M & Y^w_M & Z^w_m & 1 & 0 & 0 & 0 & 0 & -x^c_MX^w_M & -x^c_MY^w_M & -x^c_MZ^w_M & -x^c_M \\ 0 & 0 & 0 & 0 & X^w_M & Y^w_M & Z^w_M & 1 & -y^c_MX^w_M & -y^c_MY^w_M & -y^c_MZ^w_M & -y^c_M \end{bmatrix} \tag{2.20}$$

$$m^c=\left[m^{wc}_{11}\quad m^{wc}_{12}\quad m^{wc}_{13}\,\ldots\ldots\ldots\, m^{wc}_{34}\right]^T \tag{2.21}$$

By using SVD technique m^c can be obtained with the help of

$$A^c_{2M\times12}=UDV^T \tag{2.22}$$

Where U is a $2M\times12$ sized matrix and the columns of this matrix are orthogonal vectors D is a positive diagonal eigen value matrix and V is a 12×12 sized matrix whose columns are orthogonal. There exists only one nontrivial solution that corresponds to the last column of V and this is the solution to $M^{wc}_{3\times4}$. Similar procedure is used to obtain $M^{wp}_{3\times4}$.

After the calibration is performed, the reconstruction procedure is obtained using

$$P^w=\left[X^w\quad Y^w\quad Z^w\right]^T=C^{-1}D \tag{2.23}$$

where

$$C=\begin{bmatrix} m^{wc}_{11}-m^{wc}_{31}x^c & m^{wc}_{12}-m^{wc}_{32}x^c & m^{wc}_{13}-m^{wc}_{33}x^c \\ m^{wc}_{21}-m^{wc}_{31}y^c & m^{wc}_{22}-m^{wc}_{32}y^c & m^{wc}_{23}-m^{wc}_{33}y^c \\ m^{wp}_{21}-m^{wp}_{31}y^p & m^{wp}_{22}-m^{wp}_{32}y^p & m^{wp}_{23}-m^{wp}_{33}y^p \end{bmatrix}$$

$$D=\begin{bmatrix} m^{wc}_{34}x^c-m^{wc}_{14} \\ m^{wc}_{34}y^c-m^{wc}_{24} \\ m^{wp}_{34}y^p-m^{wp}_{24} \end{bmatrix}$$

In the above equations, only the vertical direction of the projector is encoded.

Least squares solution method:

The world to camera coordinate transformations are given in the Eqns (2.18) and (2.19) From these equations we can observe that there are infinite number of solutions as there is an unknown coefficient in every term so by making m_{34}^{wc} and m_{34}^{wp} equal to 1 we can obtain a linear transform at the world origin. This assumption holds well because the perspective transformation matrices are defined to scale factor. [30].

Therefore,

$$m^c = \begin{bmatrix} m_{11}^{wc} & m_{12}^{wc} & m_{13}^{wc} & m_{14}^{wc} & m_{21}^{wc} & m_{22}^{wc} & m_{23}^{wc} & m_{24}^{wc} & m_{31}^{wc} & m_{32}^{wc} & m_{33}^{wc} \end{bmatrix}^T \qquad (2.24)$$

is obtained by solving the linear equation $Am^c = B$ where A is given as

$$A_{2i-1} = \begin{bmatrix} X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ -x_i^c X_i^w \\ -x_i^c Y_i^w \\ -x_i^c Z_i^w \\ \end{bmatrix}^T \qquad A_{2i} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ X_i^w \\ Y_i^w \\ Z_i^w \\ 1 \\ -y_i^c X_i^w \\ -y_i^c Y_i^w \\ -y_i^c Z_i^w \\ \end{bmatrix} \qquad (2.25)$$

and B is given as

$$B_{2i-1} = \begin{bmatrix} x_i^c \end{bmatrix} \qquad B_{2i} = \begin{bmatrix} y_i^c \end{bmatrix} \qquad (2.26)$$

The vector m^c is obtained by pseudo inverse solution as

$$m^c = \left(A^T A\right)^{-1} A^T B \qquad (2.27)$$

Similar procedure is used to obtain m^p. After the calibration is performed, the reconstruction procedure is obtained using

$$P^w = \left[X^w \quad Y^w \quad Z^w\right]^T = C^{-1}D \tag{2.28}$$

where

$$C = \begin{bmatrix} m_{11}^{wc} - m_{31}^{wc}x^c & m_{12}^{wc} - m_{32}^{wc}x^c & m_{13}^{wc} - m_{33}^{wc}x^c \\ m_{21}^{wc} - m_{31}^{wc}y^c & m_{22}^{wc} - m_{32}^{wc}y^c & m_{23}^{wc} - m_{33}^{wc}y^c \\ m_{21}^{wp} - m_{31}^{wp}y^p & m_{22}^{wp} - m_{32}^{wp}y^p & m_{23}^{wp} - m_{33}^{wp}y^p \end{bmatrix}$$

$$D = \begin{bmatrix} m_{34}^{wc}x^c - m_{14}^{wc} \\ m_{34}^{wc}y^c - m_{24}^{wc} \\ m_{34}^{wp}y^p - m_{24}^{wp} \end{bmatrix}$$

In the above equations, only the vertical direction of the projector is encoded.

Chapter 3 Pattern Interleaving (PIL) Technique

3.1 Introduction

Pattern Interleaving (PIL) is a novel technique used to obtain high density single scan image when the object is in motion during the scan time. It uses the robust SLI scanning method, Phase Measuring Profilometry (PMP) there by making the traditional PMP insensitive to depth of ‘z’ motion. When the object is in motion during the scan time it is difficult to get high density single scan image because of the movement of the PMP sine wave patterns. In Figure 3.1 we show the 3D reconstruction of a smooth surface when it is in static state and in Figure 3.2 when it is in motion during the scan time.

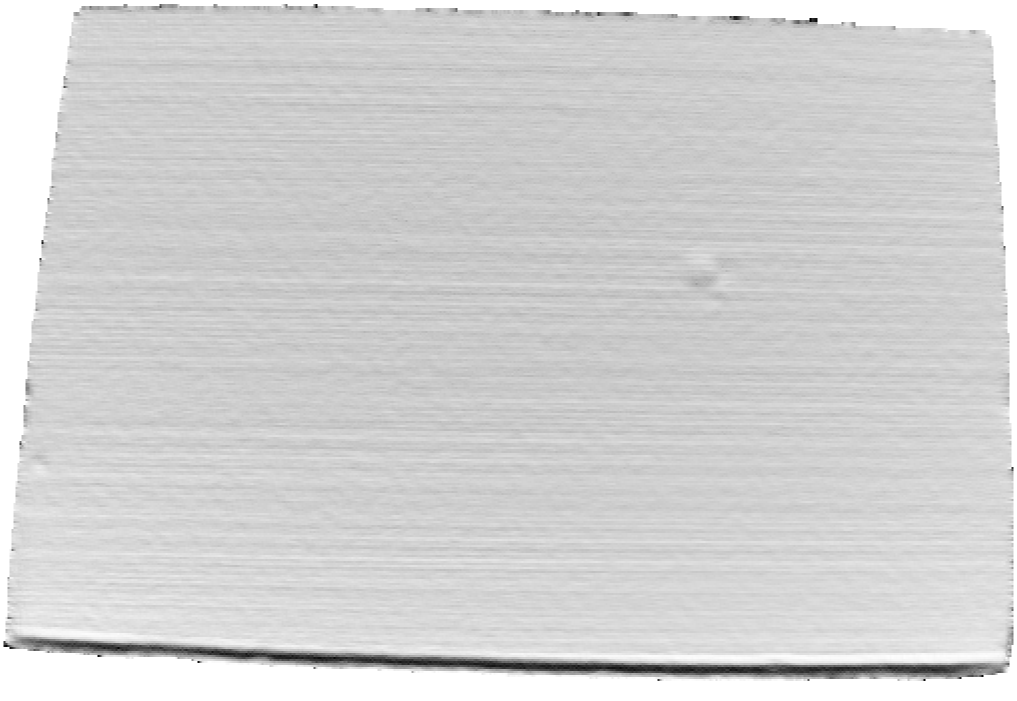


Figure 3.1 Static 3D reconstruction of a static surface

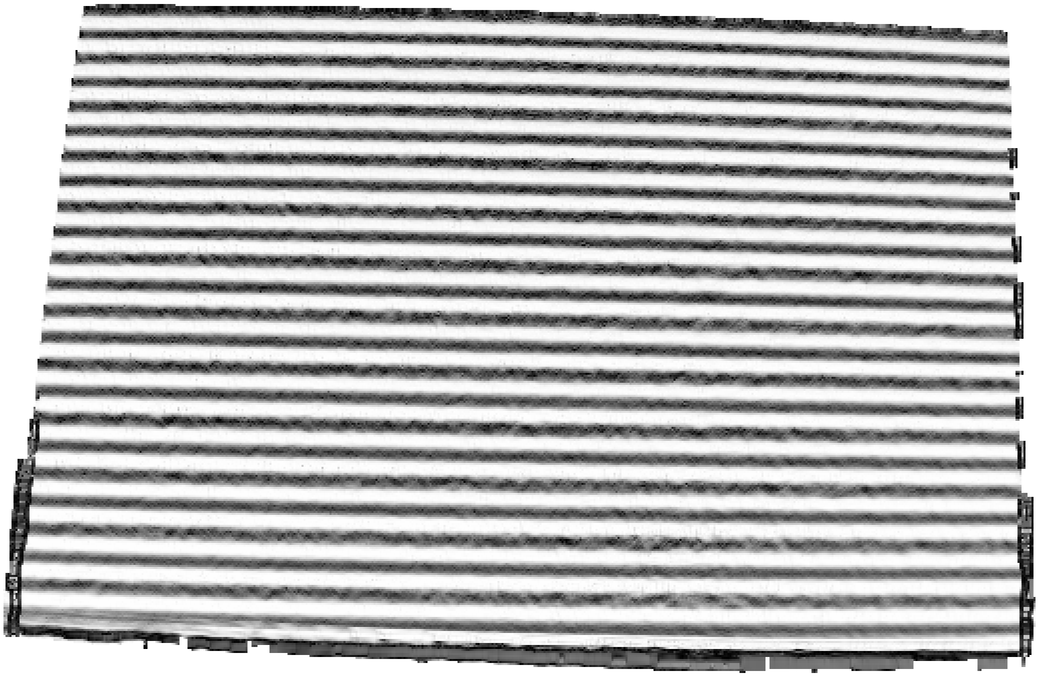


Figure 3.2 3D reconstruction of a smooth surface in motion

The main objective of the PIL technique is to correct the movement of the sine waves that occurs during the scan time and thereby reducing the motion banding. Thus, making traditional PMP insensitive to 'z' motion and thereby preventing motion banding by correcting for the motion.

3.2 Description of Pattern Interleaving (PIL) Technique

In PIL technique, traditional PMP patterns are projected in between the triangular wave patterns (PIL patterns) of constant frequency. The pattern sequence is shown in Figure 3.3

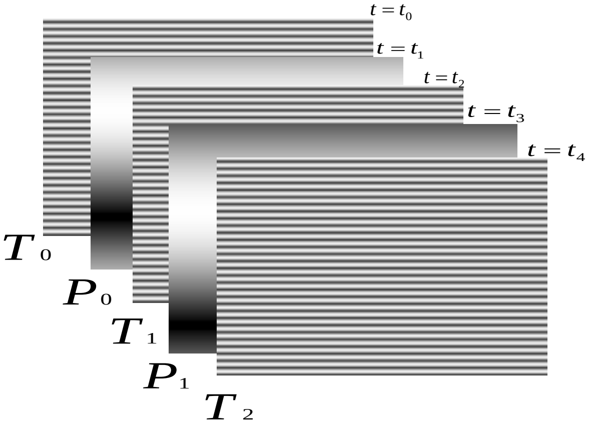


Figure 3.3 Pattern sequence

where T_n is the PIL pattern and n varies from 0 to N where $N+1$ is the total number of PIL patterns and P_n is the PMP pattern and n varies from 0 to $N-1$ and $\sum_{p=0}^{p=n} t_p$ is the scan time where $t_{p+1} > t_p$.

The motion correction is accomplished with the help of two triangular wave patterns (PIL patterns) projected before and after the sine wave pattern. The pictorial representation of the PIL technique is shown in Figure 3.4

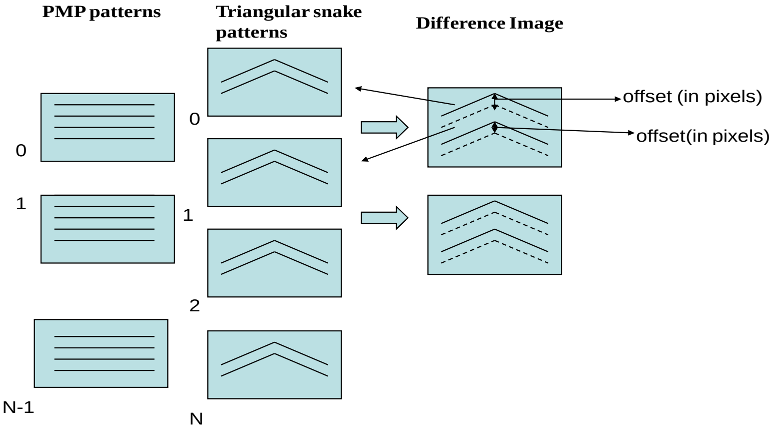


Figure 3.4 Pictorial representation of PIL technique

With the help of two triangular wave patterns the movement of the target object from one time frame to another time frame is tracked. The process of creating a difference image as shown in the Figure 3.4 is the difference of the two respective triangular wave peaks. The peaks are encoded as +1 and non-peaks as 0 so the difference results in the values of $\{+1,0,-1\}$.The lines in the difference image are the 'snakes' where the bold lines in the difference image indicates the positive peak locations and the dotted lines indicates the negative peak locations, the distance between these two (which is measured in pixels) gives the movement of the target object from one PIL pattern to the next PIL pattern. This tracked movement or the offset between the two PIL patterns helps to correct the PMP pattern in between these two PIL patterns. That is, the movement in the sine wave pattern is corrected by shifting the sine waves with half amount of the calculated offset (since PMP pattern is half way between the PIL patterns) plus the accumulated sum of the offsets of the previous PIL patterns. For example, for correcting P_0 PMP pattern the offset between T_0 and T_1 snakes is calculated and the sine waves of the P_0 pattern are shifted by half amount of the offset of T_0 and T_1. For correcting P_4 PMP pattern the offset which is used to correct the sine waves is half of the offset of T_4 and T_5 PIL patterns and sum of the offsets of the previous PIL patterns, that is, T_0 and T_1; T_1 and T_2; T_2 and T_3.

3.3 PIL Algorithm

Let $P_n(x, y)$be the n^{th} PMP pattern which is in motion where x=0, 1,…..($N_x - 1$) and y=0, 1,2…($M_y - 1$) and n is the frame number ranges from 0 to $N-1$where $N+1$ represents the total number of the PIL frames. Let $T_n(x, y)$ be the n^{th}PIL pattern which is also in motion.

Step1: Filtering the PIL patterns

The PIL images are filtered to remove the noise. Let $C_n(x, y)$be the PIL image which is filtered by using a moving average filter such that

$$C_n(x, y) = T_n(x, y) * h(x, y) \tag{3.1}$$

where $h(x, y) = rect\left(\frac{y}{\tau_x}\right) rect\left(\frac{x}{\tau_y}\right)$

and ‘*’ represents convolution

Step 2: Finding the peak to side lobe ratio

Let $PSR_n(x, y)$ be the peak to side lobe ratio for the filtered PIL images and it is calculated as

$$PSR_n(x, y) = \frac{C_n(x, y)}{\max\{C_n(x, y-\tau), C_n(x, y+\tau)\}} \tag{3.2}$$

where τ is the side lobe spacing

Step3: Snake detection and encoding

Snakes detection is performed by looking at the locations where PSR_n is maximum as snakes represent the maximum PSR_n locations. By performing snake detection, we can make snakes visible and thus make it easier to calculate the offset that occurred due to the motion in the PIL pattern images. Let $S_n(x, y)$be the snake image. The snake masking is mainly carried out in two steps

1. Snake Locations
2. Snake Peaking

Snake Locations: For a pixel to be encoded as a peak the PSR at that pixel must be greater than the predetermined threshold and the intensity or the peak value at that pixel must be greater than a predetermined minimum value. The resulting regions encoded with 255 contain the snake center positions. To determine the most likely snake centers, we apply “Snake Peaking” process.

Snake Peaking: After creating low region (zero values) and a high region (255 values) we need to run the snake process in two states

1. Search for the start of high region (state 0)
2. Search for the end of high region (state 1)

First, search for the start of high region (search for value 255) if a value of 255 is found then encode that location as peak and as a start location and then go to state 1. In state 1, search for value 0 and assign the location as an end location. Search for maximum PSR between the start and end locations, if a maximum PSR is obtained encode that location as peak and assign a value of 255 at that location and go to state 0.

A flowchart explaining the process of snake detection is shown in Figure 3.5

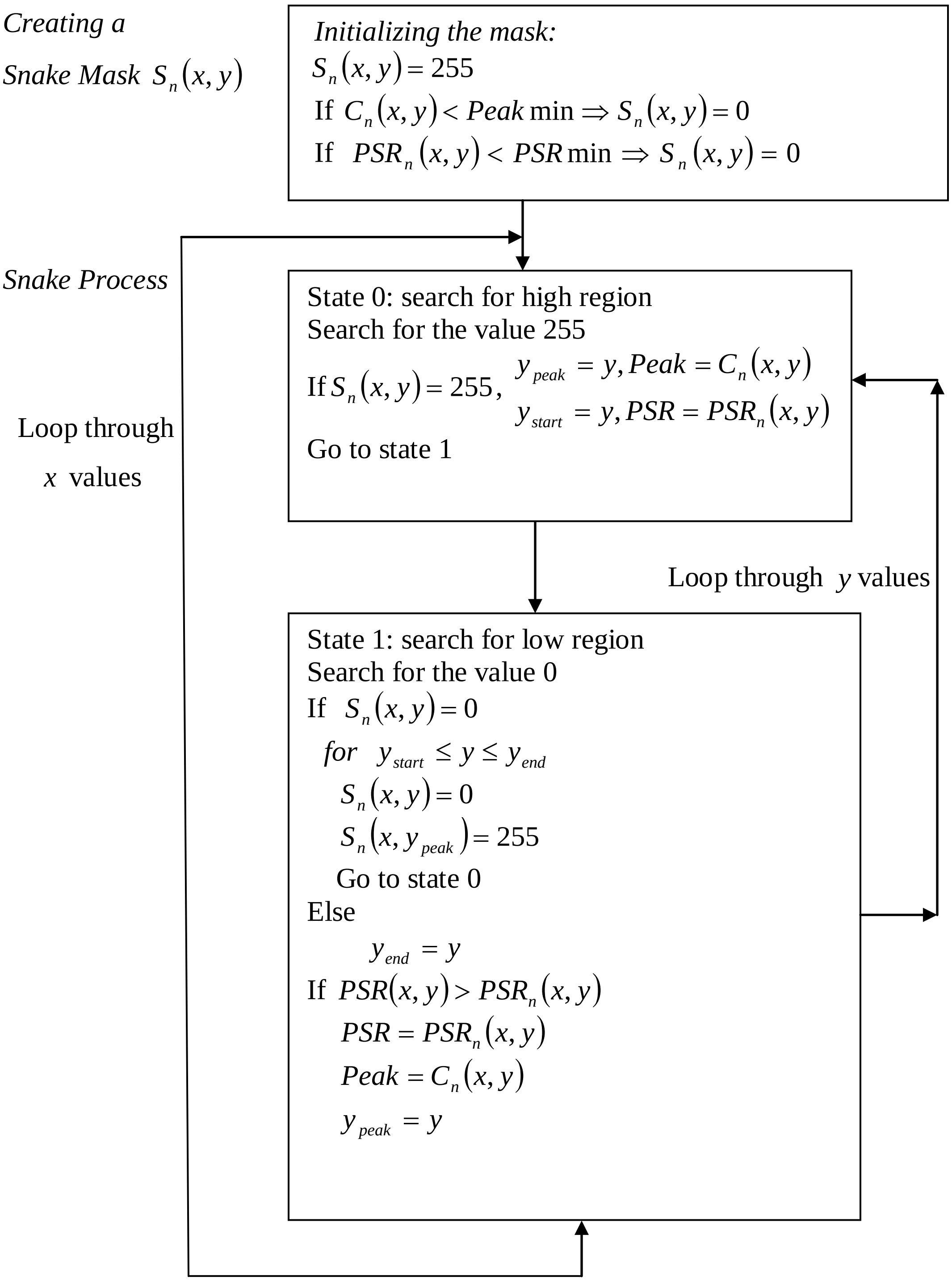


Figure 3.5 Flowchart of snake detection

Step 4: Finding the offset

Let the difference image between the two snake images be

$$D_n(x, y) = S_{n-1}(x, y) - S_n(x, y) \tag{3.3}$$

if the difference between the two snake images is zero, then there is no movement that is, if $D_n(x, y) = 0$ there is no movement otherwise if $D_n(x, y) > 0$ there is movement. Let $\{x, y_p\}$ where p =0, 1, 2…. $P-1$ be the positive peak locations in the difference image, that is, $D_n(x, y_p) > 0$. As we are considering the movement in y direction, the x value can be ignored and P is the total number of positive peak locations in the difference image. Similarly let $\{x, y_q\}$ where q =0, 1, 2…. $Q-1$ be the negative peak locations in the difference image, that is, $D_n(x, y_q) < 0$. Q is the total number of the negative peak locations in the difference image. The offset, due to motion, is calculated as the magnitude of the difference between the positive peak and its associated negative peak Let dy_n be the offset which is calculated as

$$dy_n \equiv y_p - y_q \tag{3.4}$$

Step 5: Compensating the PIL patterns:

The PIL images can be compensated with the help of the offsets obtained in Eq.(3.4) depending on the movement of the object with reference to camera Let us consider y_m and y_{m-1} be the locations of the adjacent snakes in $S_n(x, y)$ where m=0, 1, 2… $M-1$. M is the total number of snakes in the image. From Eq.(3.4) we know that dy_m is the offset occurred due to motion at y_m in $S_n(x, y)$ image with respect to $S_{n-1}(x, y)$ image. Similarly we know that dy_{m-1} is the offset due to motion at y_{m-1} in $S_n(x, y)$ image with respect to $S_{n-1}(x, y)$. These can be expressed as

$$dy_m(x) = dy_n(x, y_m) \tag{3.5}$$

$$dy_{m-1}(x) = dy_n(x, y_{m-1}) \tag{3.6}$$

The offsets between the adjacent snakes can be estimated by the interpolation of the offsets that occurred at the snakes. That is, the offsets between y_m and y_{m-1} can be estimated by interpolating with the help of dy_m and dy_{m-1}.

The offset values between the peaks is interpolated such that

$$dy_n(x,y) = a_m \times y + b_m \qquad \text{for } y_{m-1}(x) \le y \le y_m(x) \tag{3.7}$$

The values of a_m and b_m are obtained with the help of two equations shown below

$$dy_n(x, y_m(x)) = a_m \times y_m + b_m \tag{3.8}$$

$$dy_n(x, y_{m-1}(x)) = a_m \times y_{m-1} + b_m \tag{3.9}$$

As we know $dy_n(x, y_m(x)), y_m(x), dy_n(x, y_{m-1}(x))$ and $y_{m-1}(x)$, a_m and b_m values can be calculated easily

$$a_m = \frac{dy_n(x, y_m) - dy_n(x, y_{m-1})}{y_m - y_{m-1}} \tag{3.10}$$

$$b_m = dy_n(x, y_m) - a_m y_m \tag{3.11}$$

The motion correction of the snake image is

$$S_n(x, y + dy_n(x,y)) = S_{n-1}(x,y) \quad \text{for } y_{m-1}(x) \le y \le y_m(x) \tag{3.12}$$

With the help of Eq(3.7) the PMP images are motion compensated. As the PMP images contributes only half of that of PIL images the offset is taken as $\frac{dy_n(x,y)}{2}$ instead of $dy_n(x,y)$ and we require the previous offsets to accurately align all the PMP patterns

The compensation of the PMP patterns is as follows

$$P_n\left(x, y + \frac{dy_n(x,y)}{2} + \sum_{k=1}^{k=n-1} dy_k(x,y)\right) = P_n(x,y) \quad \text{for } y_{m-1}(x) \le y \le y_m(x) \tag{3.13}$$

3.4 Experiments and results

3.4.1 Estimating offset between the two successive PIL patterns

As explained in the section 3.3, in order to track the movement of the target object during the scan time we need to find the difference between the two successive snake masked PIL patterns. The two successive PIL snake masked patterns are shown in Figure 3.6 and Figure 3.7

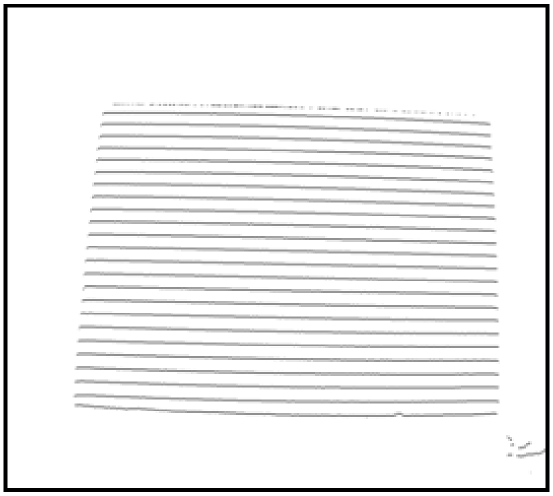


Figure 3.6 Snake masked T_0 PIL pattern

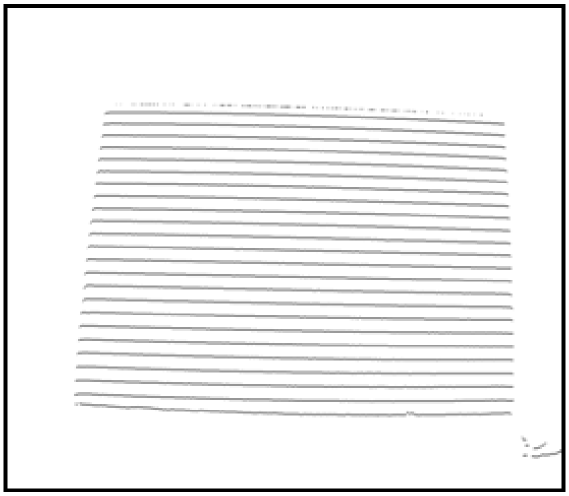


Figure 3.7 Snake masked T_1 PIL pattern

The difference between the two snaked masked PIL patterns (Figure 3.6 and Figure 3.7) is shown in Figure 3.8

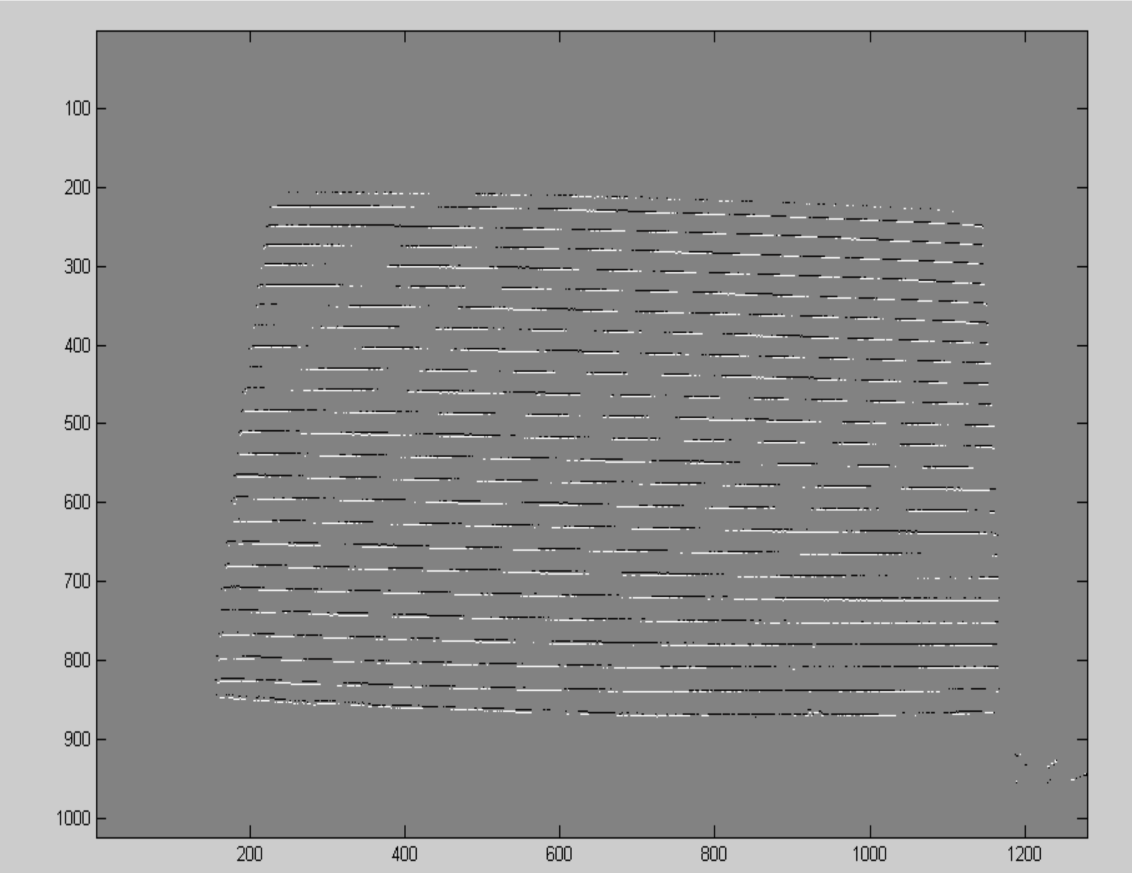


Figure 3.8 Difference image of the snake masked T_0 PIL pattern and T_1 PIL pattern

A cropped out section of Figure 3.8 is shown in Figure 3.9

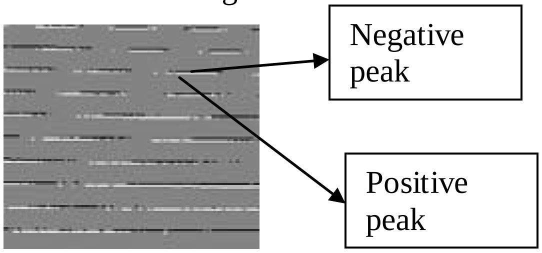


Figure 3.9 Cropped section of the difference image

The white line in the difference image indicates the positive peak locations and the black lines in the difference image indicate the negative peak locations. The difference between these two gives the offset or the movement of the object at that location from one PIL frame to the next PIL frame.

A cross sectional plot of the intensity of the middle column of the difference image is shown in Figure 3.10 for better visualization of the positive and negative locations

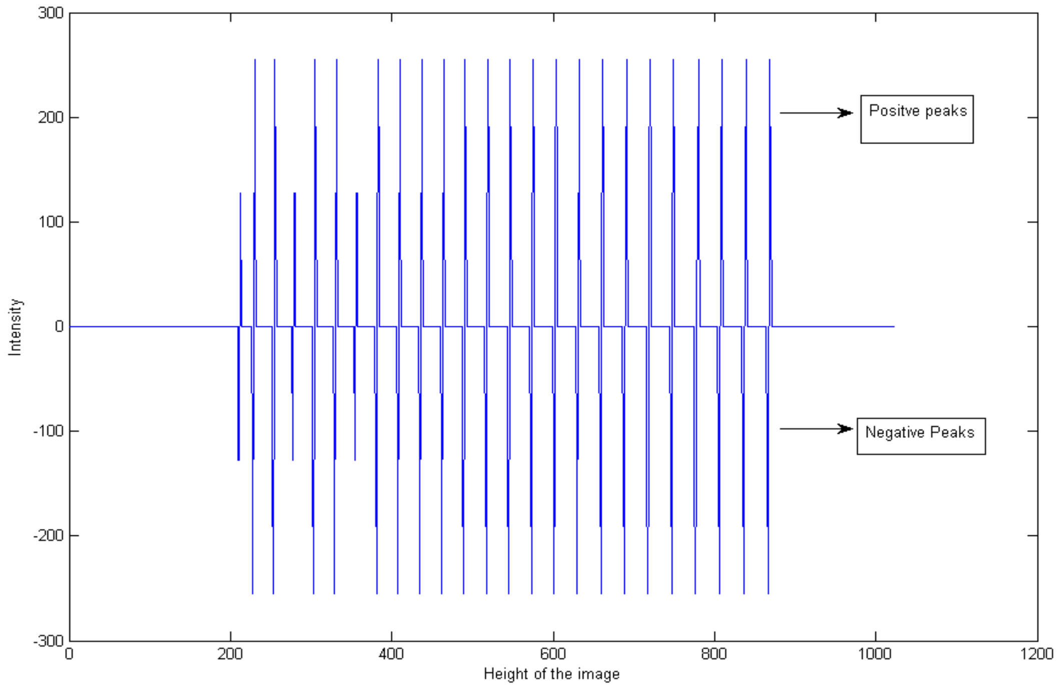


Figure 3.10 Middle column intensity of the difference image

If the object is moving towards the camera the difference between the negative peak and the corresponding positive peak will give the movement or the offset which is the case as shown in the figure above similarly if the target object is moving away from the camera the difference between the positive peak and the corresponding negative peak will give the offset. That is, from Eq(3.4) we have

$dy_n(y_p, y_q) > 0$ When the object is moving away from the camera

$dy_n(y_p, y_q) < 0$ When the obj

ect is moving towards the camera

However, the magnitude of the distance between the negative peak and positive peak (or positive peak and negative peak) will give the offset.

A cross sectional plot of the offsets (the movement of the T_1 PIL pattern with respect to T_0 PIL pattern) for the middle column locations is shown in Figure 3.11

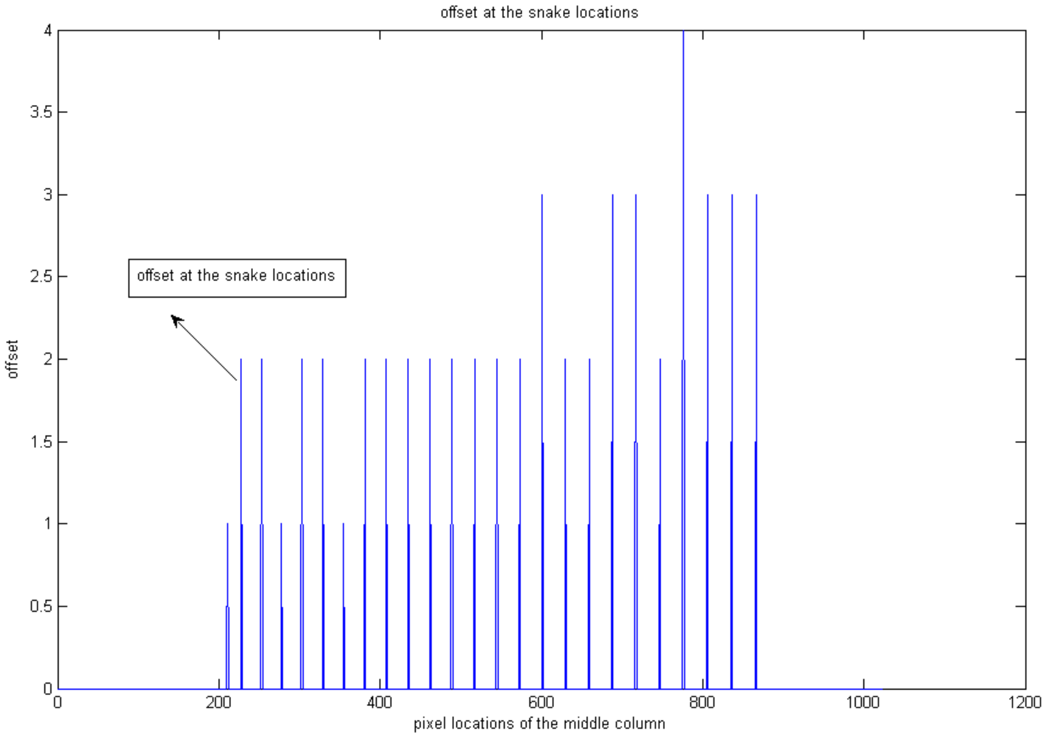


Figure 3.11 dy **for the pixel locations of the middle column at snake locations**

From the Figure 3.11 we can determine the offset at the pixel locations (snake locations) for the middle column of T_1 PIL pattern with respect to the T_0 PIL pattern. Similarly one can find out the offset at the other columns snake locations for T_1 PIL pattern.

3.4.2 Interpolating the offset to correct the PMP pattern

The offset which is calculated above is used only to correct the snake locations, so in order to correct the typical PMP pattern we need to know the offset between the snakes.

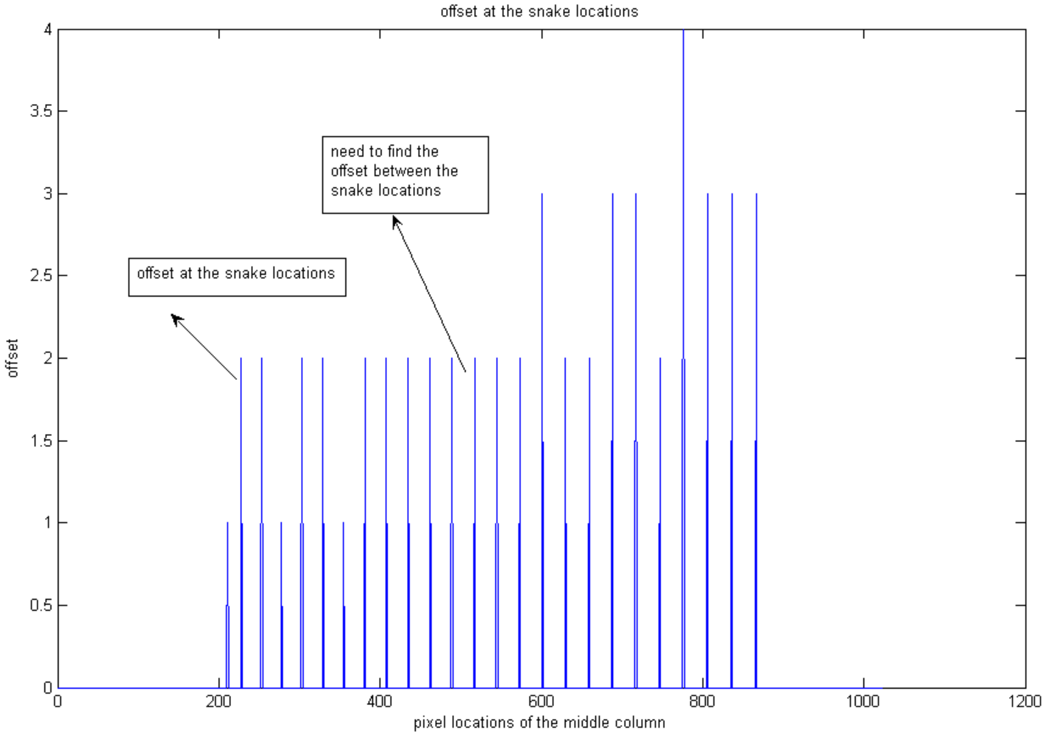


Figure 3.12 Need for interpolating the offset between the snake regions

To calculate the offset between the snake regions, as explained in section 3.3, we use linear interpolation to find out the offsets between the snake regions with the help of known offsets at the snake locations which is explained in the section 3.4.1.

A cross sectional plot of the interpolated offsets for a middle column is shown in Figure 3.13

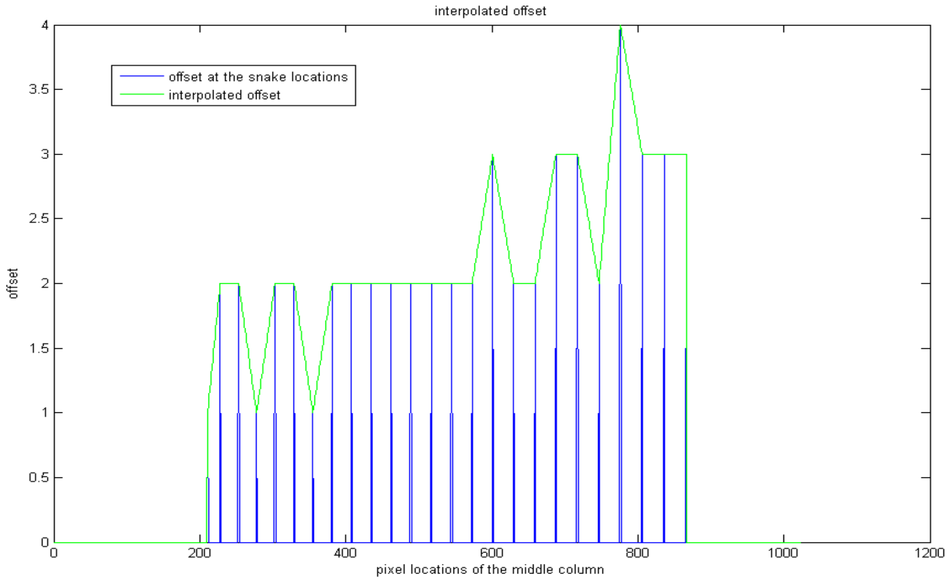
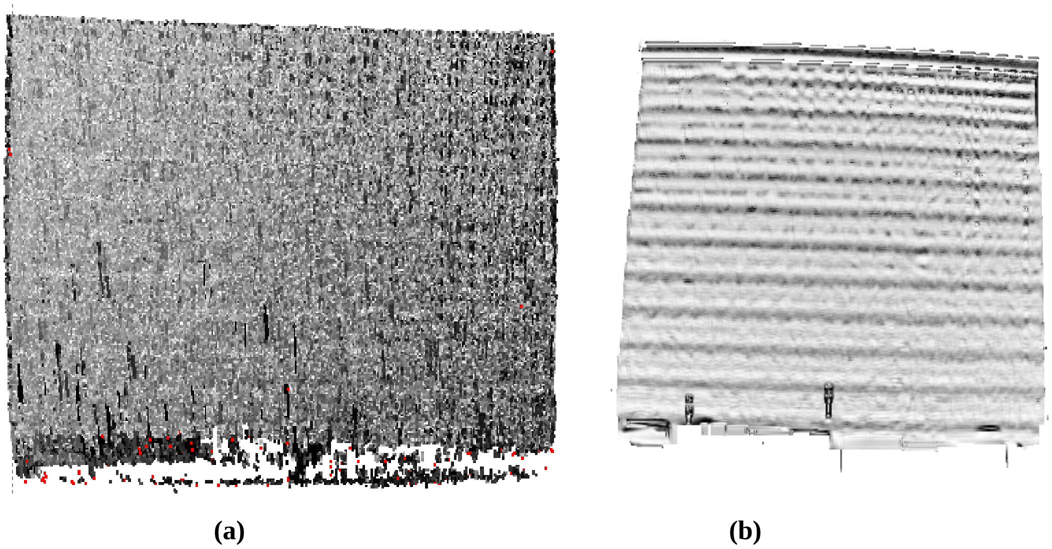


Figure 3.13 Figure showing the offset at the snake locations and the interpolated offset between the snake regions

From the Figure 3.13 we can observe that the blue lines indicate the offsets which are calculated in the section 3.4.1 and the green line indicates the interpolated offsets between the snake locations by using the help of the offsets at the snake locations.

3.4.3 Result of PIL technique

The 3D reconstruction of a smooth surface by using PIL technique is shown in Figure 3.14 (a) and the filtered 3D reconstructed image is shown in Figure 3.14 (b).



(a) **(b)**

Figure 3.14 (a) 3D reconstruction of a smooth surface using PIL technique (b) filtered 3D reconstructed smooth surface using PIL

A cross sectional side view of the 3D reconstruction of static smooth surface, smooth surface in motion and smooth surface in motion corrected by PIL technique are shown in Figure 3.15 for better visualization of the results.

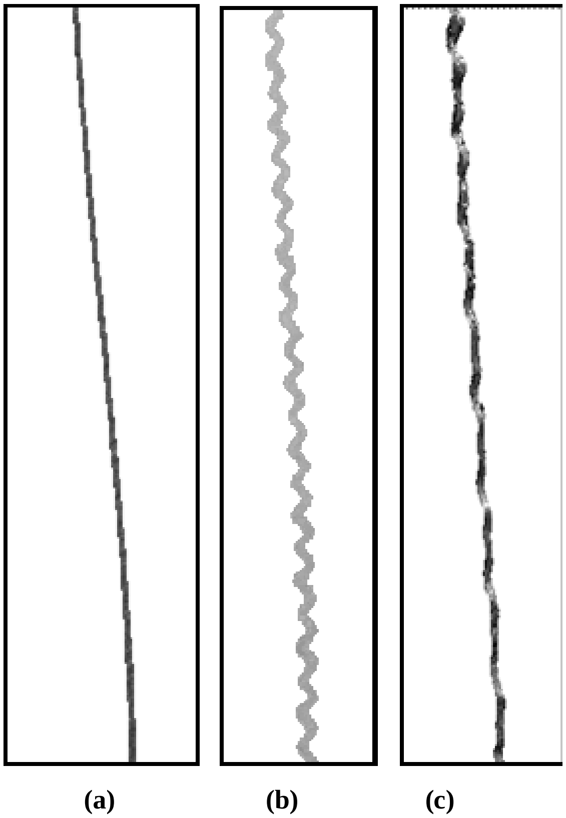


Figure 3.15 (a)Side view of static 3D reconstruction of smooth surface (b) side view of the 3D reconstructed smooth surface in motion (c) side view of the 3D reconstructed smooth surface using PIL technique

From the Figure 3.15 (b), we can clearly see that due to motion on the smooth surface during scan time the 3D reconstruction of smooth surface contains ripples unlike static smooth surface as shown in Figure 3.15 (a). From the Figure 3.15 (c), we can see that by applying the PIL technique the ripples are significantly reduced and the 3D reconstruction of smooth surface using PIL technique looks approximately similar to the static 3D reconstruction.

3.5 Band Ripple Measurement

3.5.1 Calculation of Band Energy for an ideal ripple

To objectively evaluate the performance of PIL, we need a motion banding measure. We introduce a "band energy measure" where band energy is defined as the ratio of the peak to peak distance of ripples to the wavelength of the ripples.

$$b_e = {}^{h}\!/_{\lambda} \qquad (3.14)$$

Where b_e is the average band energy, λ is the wavelength of the ripples and h is the peak to peak distance of ripples.

The calculation of band energy for an ideal ripple (sine wave) is shown in the Figure 3.16

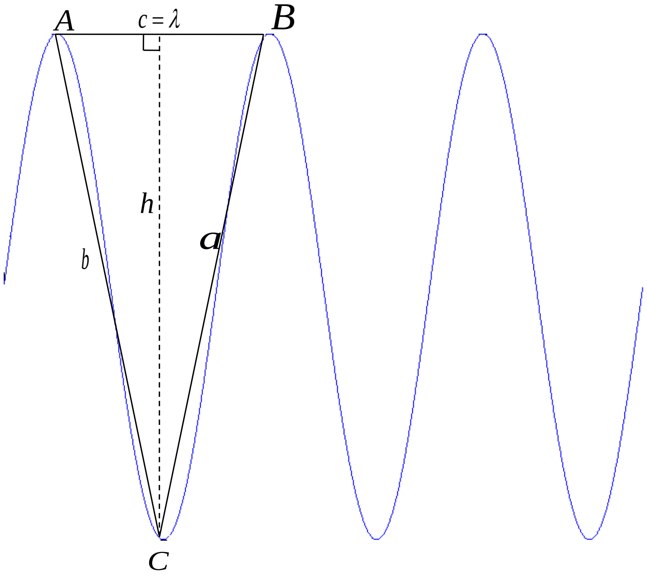


Figure 3.16 Ideal ripple

From the Figure 3.16, we can observe that the points A, B and C form a triangle where A and B are the maximum peaks of a ripple and C is the valley point of a ripple and a, b and c are the corresponding distances or the length of sides of a triangle. Therefore, the wavelength of a ripple is nothing but the distance between the points A and B, which is c.

To calculate the peak to peak distance of a ripple, initially we use Heron's formula to calculate the area of the triangle and after that we can get the peak to peak distance of the ripple from the area calculated.

The area of a triangle is calculated from Heron's formula as follows

$$Area = \sqrt{s(s-a)(s-b(s-c))} \quad \text{where semi-perimeter } s = \frac{a+b+c}{2}$$

The peak to peak distance of the ripple is calculated as

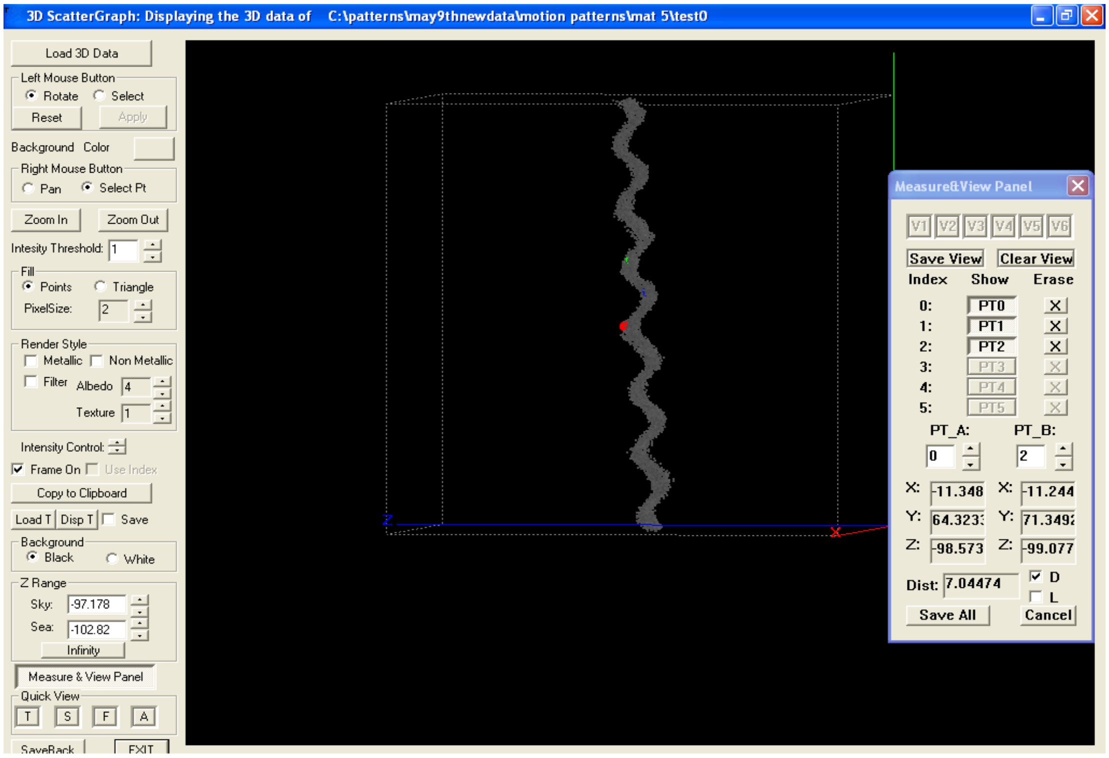
$$h = \frac{2 \times Area}{\lambda}$$

since λ is the base of the triangle.

As the peak points and the valley points are expressed in world coordinates, the band energy measure is independent of the orientation.

3.5.2 Calculation of Band Energy without PIL correction

Band energy for a smooth surface target without PIL correction is calculated with the help of GL3D View software as shown in Figure 3.17



3D ScatterGraph: Displaying the 3D data of C:\patterns\may9thnewdata\motion patterns\mat 5\test0

Left Mouse Button: ● Rotate ○ Select

Right Mouse Button: ○ Pan ● Select Pt

Intesity Threshold: 1

Fill: ● Points ○ Triangle

PixelSize: 2

Render Style: □ Metallic □ Non Metallic

□ Filter Albedo 4

Texture 1

☑ Frame On □ Use Index

□ Save

Background: ● Black ○ White

Z Range

Sky: -97.178

Sea: -102.82

Measure&View Panel

Index	Show	Erase
0:	PT0	X
1:	PT1	X
2:	PT2	X
3:	PT3	X
4:	PT4	X
5:	PT5	X

PT_A: 0 PT_B: 2

	PT_A	PT_B
X:	-11.348	-11.244
Y:	64.3233	71.3492
Z:	-98.573	-99.077

Dist: 7.04474 ☑ D □ L

Figure 3.17 Measuring Band Energy of a smooth surface without PIL correction

From the Figure 3.17, the green and red points denote the maximum peak points and blue point is valley point. The horizontal orientation of ripples is shown in Figure 3.18

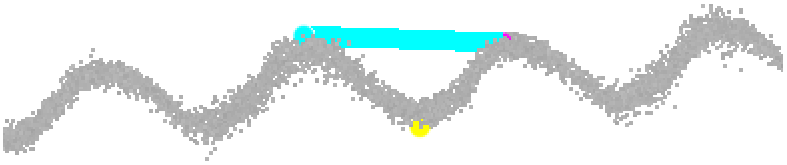


Figure 3.18 Measuring wavelength of ripples of a smooth surface without PIL correction

From Figure 3.18, we can clearly see that these three points form a triangle and the length of the sides can be easily measured with the help of measurement panel. The wavelength of the ripple is nothing but the distance between the red point and the green point as shown in the Figure 3.18. As we know the length of the sides of the triangle we can easily measure the peak to peak distance of the ripple as explained in the section 3.5.1

The wavelength which is measured in world coordinates is obtained as 7.04474 mm and the approximate peak to peak distance calculated is 1.9265 mm.

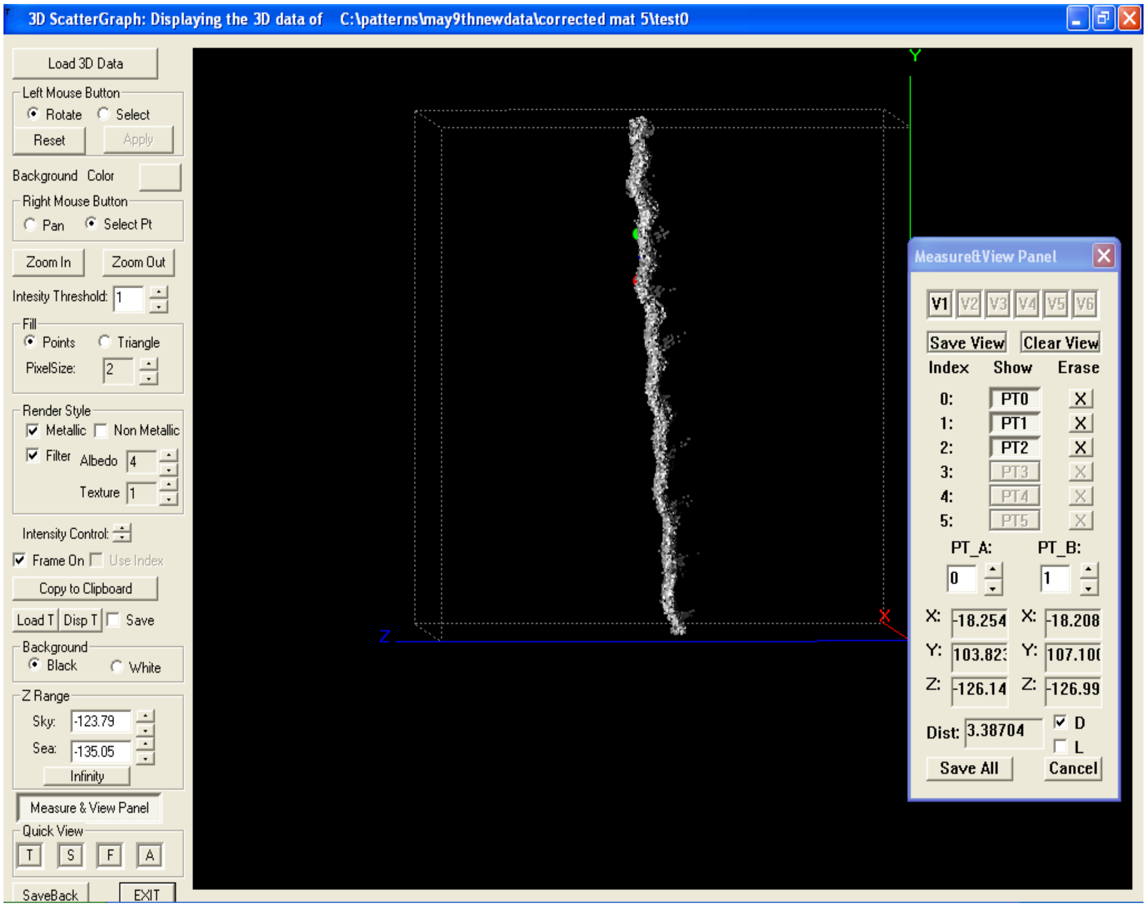
Therefore, average band energy is $b_e = 1.9265$ mm /7.04474 mm

= 0.2735.

The average band energy without PIL correction is 0.2735.

3.5.3 Calculation of Band Energy after PIL correction

The procedure for calculating the band energy after PIL correction is same as calculating the band energy without PIL correction. The 3D reconstruction of a smooth surface after PIL correction is shown in Figure 3.19



3D ScatterGraph: Displaying the 3D data of   C:\patterns\may9thnewdata\corrected mat 5\test0

Left Mouse Button: ● Rotate ○ Select

Right Mouse Button: ○ Pan ● Select Pt

Intesity Threshold: 1

Fill: ● Points ○ Triangle

PixelSize: 2

Render Style: ☑ Metallic □ Non Metallic

☑ Filter Albedo 4

Texture 1

☑ Frame On □ Use Index

□ Save

Background: ● Black ○ White

Z Range

Sky: -123.79

Sea: -135.05

Measure&View Panel

Index	Show	Erase
0:	PT0	X
1:	PT1	X
2:	PT2	X
3:	PT3	X
4:	PT4	X
5:	PT5	X

PT_A: 0   PT_B: 1

X: -18.254   X: -18.208

Y: 103.82:   Y: 107.10(

Z: -126.14   Z: -126.99

Dist: 3.38704   ☑ D □ L

Figure 3.19 Measuring Band Energy of a smooth surface after PIL correction

The horizontal orientation of the ripples after PIL correction is shown in Figure 3.20

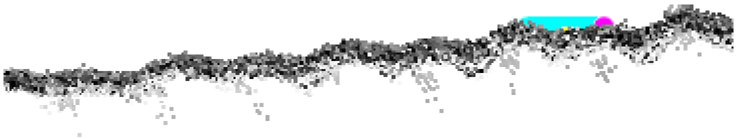


Figure 3.20 Measuring wavelength of the ripples of a smooth surface after PIL correction

By applying the same procedure which is used to find band energy for a smooth surface without PIL correction we get the wavelength of ripples after PIL correction in world coordinates is 6.359 mm and the approximate peak to peak distance is 0.8701

Average band energy b_e = 0.8701 mm/6.359 mm

= 0.1368.

Therefore, band attenuation which is defined as the ratio of the band energy using PIL technique to the band energy without using PIL technique is

b_a = 0.1368/0.2735

= 0.5001

Chapter 4 Lateral Correction Approach

Pattern Interleaving (PIL) Technique is used to correct the ‘z’ motion of the object that occurs during the scan time. In this chapter, an approach to correct the lateral movement of the object during the scan time is presented. In this approach, an object with distinctive features and a uniform black background is considered as shown in the Figure 4.1

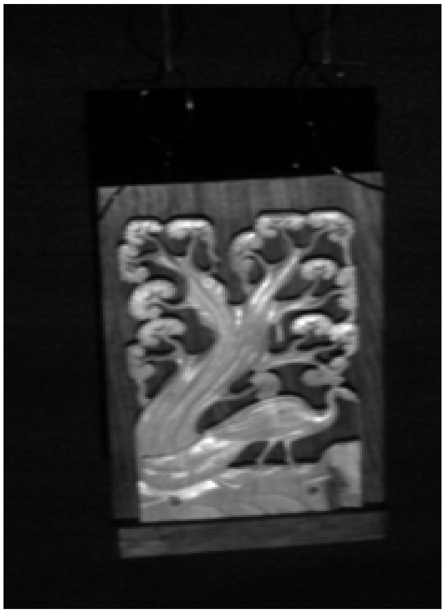


Figure 4.1 Setup arrangement for lateral correction

4.1 Steps to correct the lateral movement

Step 1: Edge enhancement of the PMP patterns:

Two successive PMP patterns are considered to track the lateral or “left to right” motion. These two captured images are edge enhanced by using “sobel” edge enhancement technique so that the distinctive features more visible.

The first captured PMP pattern which is used as a reference to correct the captured second captured PMP pattern is shown in the Figure 4.2

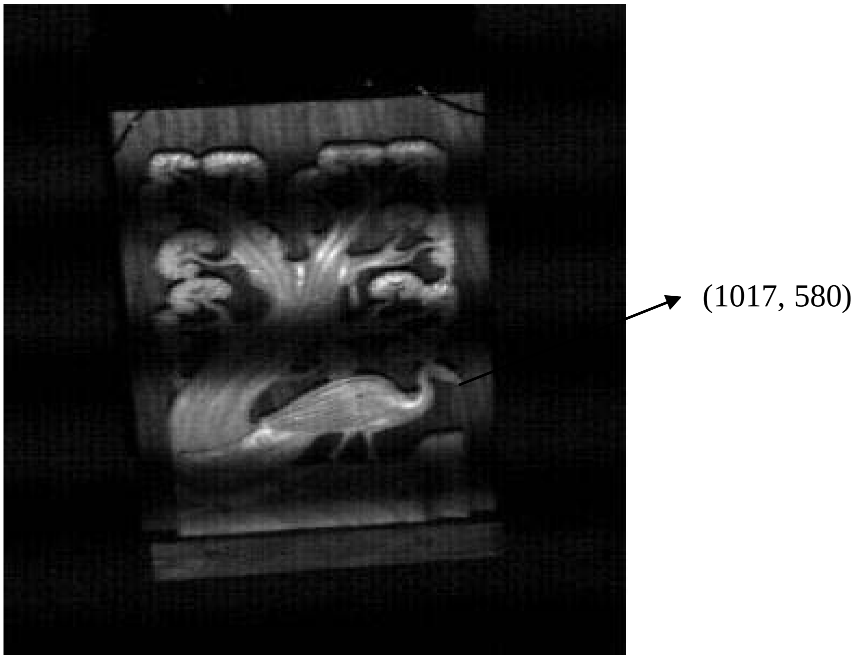


Figure 4.2 First captured PMP pattern

A marker point is used to show the movement from one frame to another frame. The second captured PMP pattern which needs to be compensated in the motion in the lateral direction is shown in the Figure 4.3

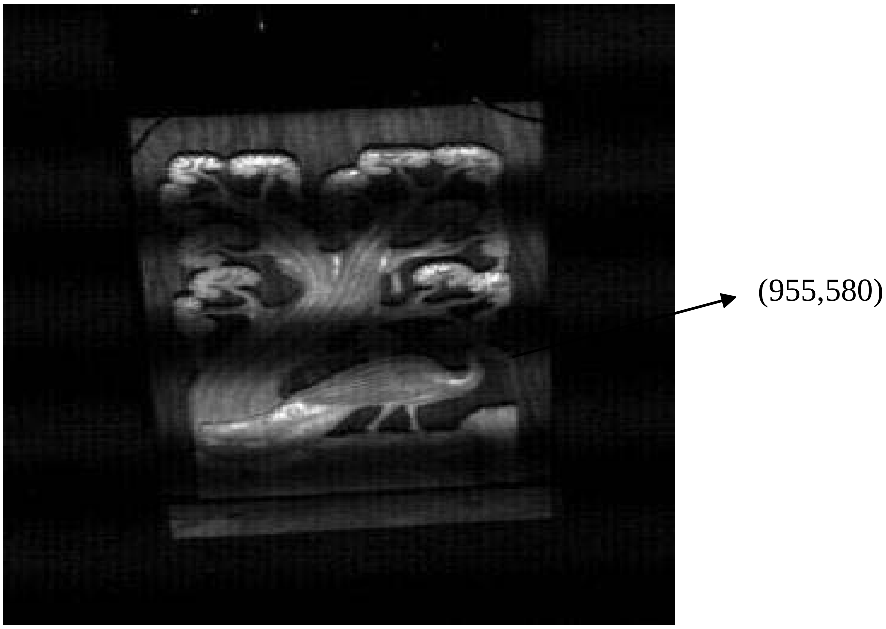


Figure 4.3 Second captured PMP pattern

We can clearly see that there is a movement in the lateral direction on the second captured PMP pattern. That is, the movement from $\{x, y\} = \{1017,580\}$ to $\{x, y\} = \{955,580\}$. The sobel edge enhanced image of the first captured PMP pattern is shown in the Figure 4.4

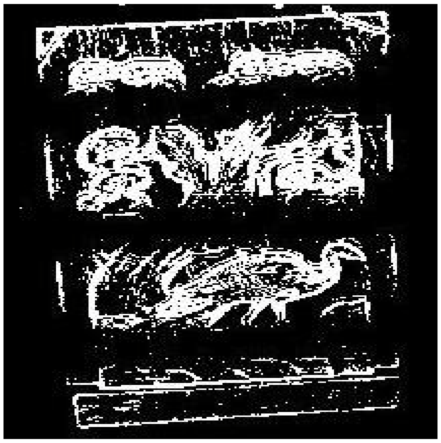


Figure 4.4 Sobel edge enhancement image of the first captured PMP pattern

The sobel edge enhancement of the second captured PMP pattern is shown in the Figure 4.5

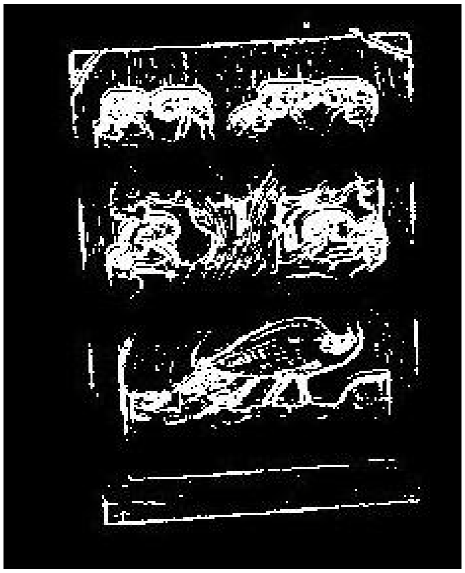


Figure 4.5 Sobel edge enhanced image of the second captured PMP pattern

Step 2: Normalized cross correlation of the edge enhanced images

The two edge enhanced images obtained in the step 1 are cross correlated and normalized to find out the offset or the movement of the PMP pattern from one frame to another frame. 3D cross correlated plot is shown in Figure 4.6

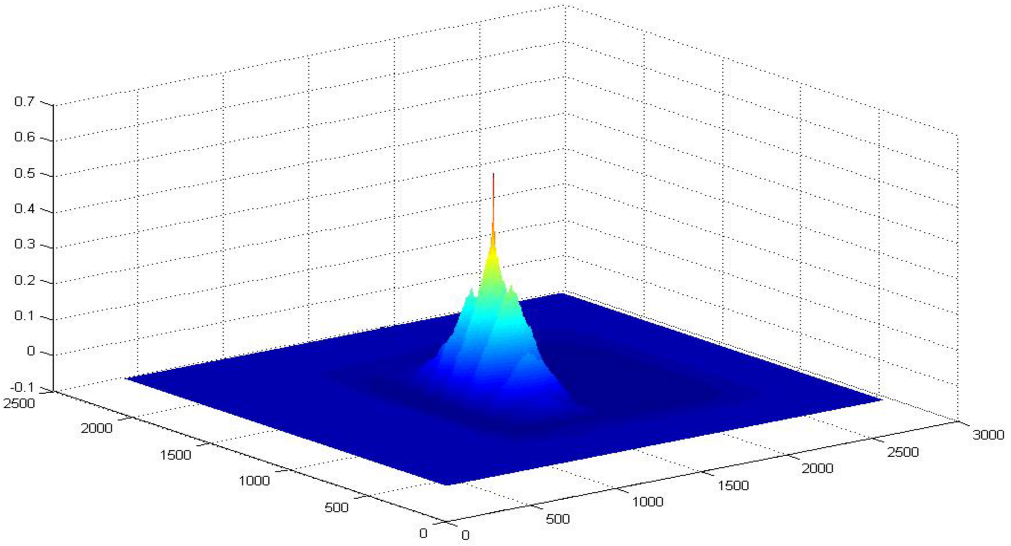


Figure 4.6 3D plot of cross correlation between the two sobel edge enhanced images

Peak location of the plot helps to figure out the offset that needs to be compensated. In general, the offset in ‘x’ direction would be the difference between the peak ‘x’ location and the number of rows in the image.

Step 3: Compensating the PMP patterns

The PMP pattern that needs to be corrected is moved laterally with the help of the offset obtained in the step 2. The PMP pattern which is laterally corrected is shown in the Figure 4.7

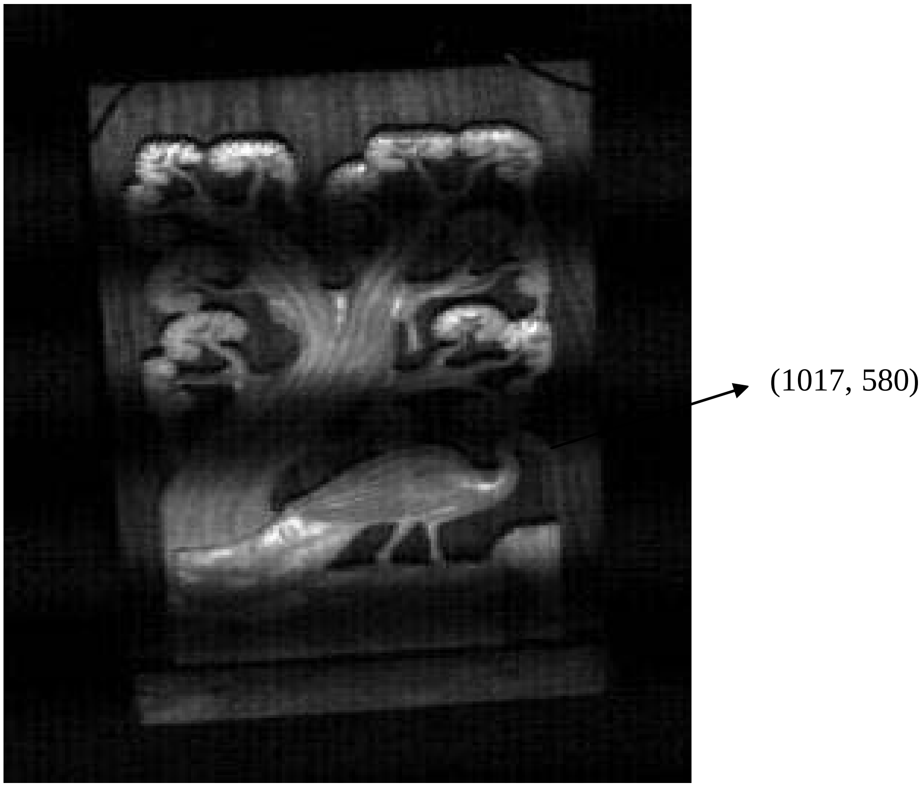


Figure 4.7 Corrected second captured PMP pattern

From the Figure 4.7 and Figure 4.2 we could see that the second captured PMP pattern is motion compensated in the lateral direction. Thus, using this approach we are able to correct the lateral movement in all the PMP patterns.

Chapter 5 Experiments and Results

In this section, different experimental results conducted on different objects by considering two types of ‘z’ motion are presented. The two types of ‘z’ motion are

1. uniform ‘z’ motion
2. non-uniform ‘z’ motion

In a uniform ‘z’ motion, the object moves in the ‘z’ direction in equal increments and in only one direction during the scan time either towards the SLI system or away from the SLI system. In a non-uniform ‘z’ motion, the object moves in the ‘z’ direction in both ways during the scan time that is it moves back and forth (towards the SLI system and away from the SLI system) during the scan time. The comparison of 3D reconstructed results with and without using the PIL algorithm is presented.

5.1 3D reconstruction of objects in uniform ‘z’ motion

We present the comparison of the 3D reconstruction of the objects when they are in static, uniform ‘z’ motion during the scan time and motion corrected using PIL technique The ‘z’ motion towards the camera and ‘z’ motion away from the camera are considered. The side views of the 3D reconstructed images are presented for better visualization of the ripples that occurred due to the motion of the object during the scan time and the side views of the 3D reconstructed images using PIL are also presented to show the correction.

A smooth surface which is subjected to uniform ‘z’ motion towards the camera is considered and the comparison of the 3D reconstruction of the smooth surface when it is in static, ‘z’ motion towards the camera and motion corrected using PIL technique is shown in the Figure 5.1

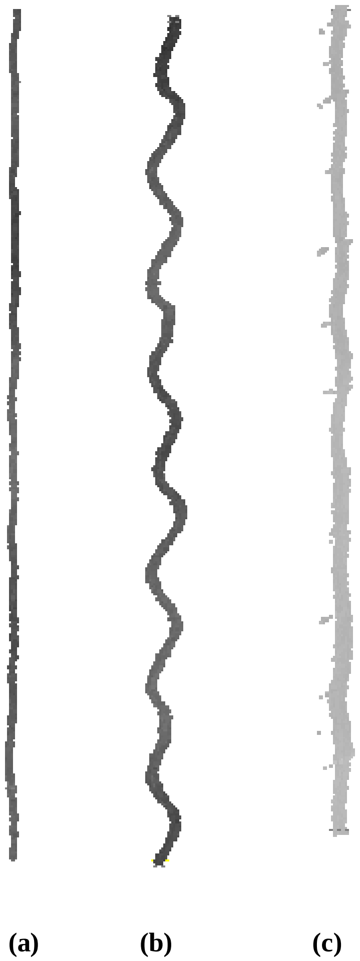


Figure 5.1 (a) side view of the 3D reconstructed smooth surface in static, (b) side view of the 3D reconstructed smooth surface in motion and (c) side view of the 3D reconstructed smooth surface using PIL technique.

From the Figure 5.1 we could clearly see that there is no variation in the ‘z’ direction when the object is in a static position and there is a variation in the ‘z’ direction when the object is subjected to ‘z’ motion during the scan time and we also see the compensation of the ‘z’ motion by using PIL technique. The ripples which appeared in Figure 5.1 (b) are significantly reduced by using PIL technique as shown in Figure 5.1 (c).
Similarly, a face mannequin, ‘Alice’ is subjected to uniform motion in ‘z’ direction during the scan time.

The 3D reconstruction of the Alice in static is shown in the Figure 5.2.

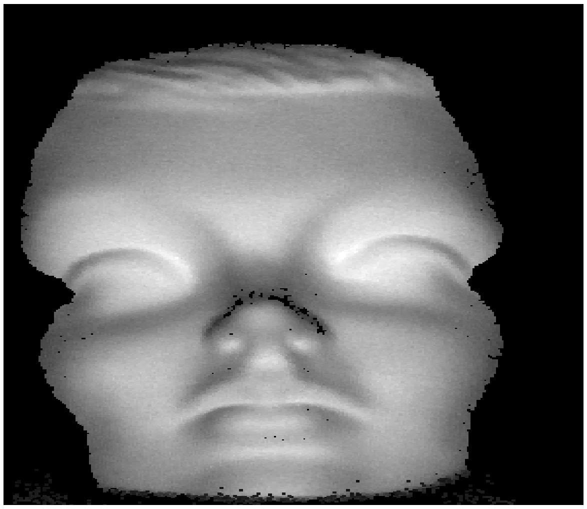


Figure 5.2 3D reconstruction of a face model

The side views of the static, motion and motion corrected (using PIL) 3D reconstructed images are presented to compare the results are shown in the Figure 5.3.

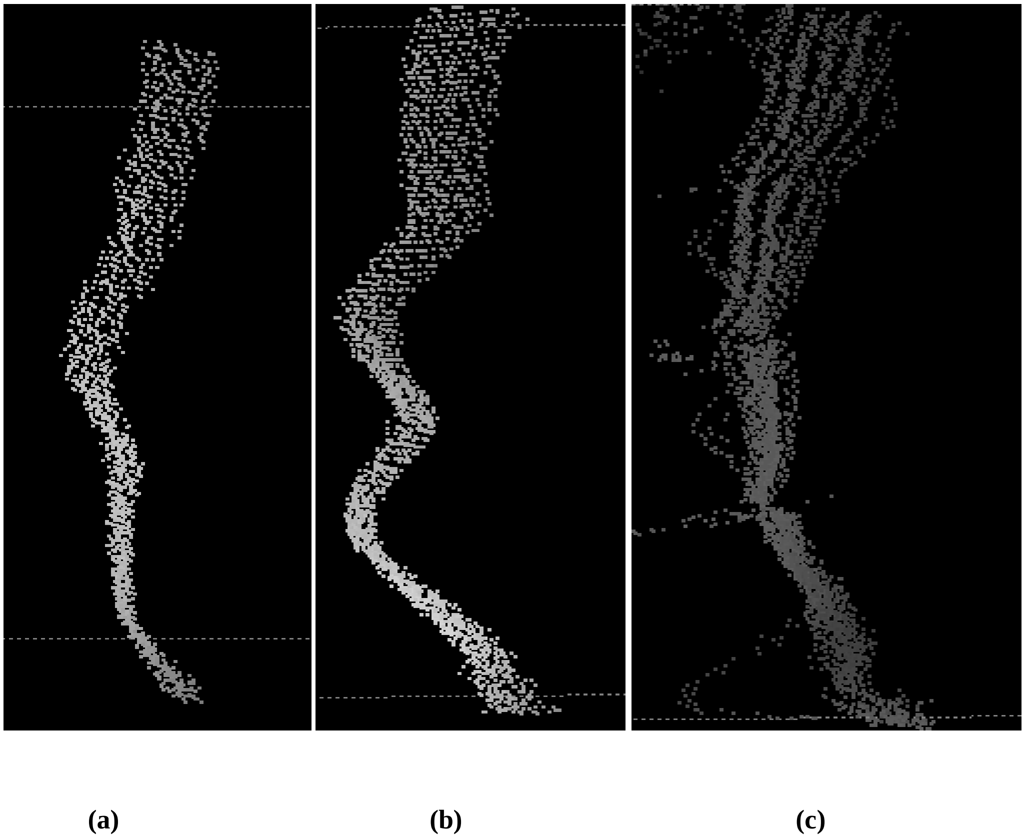


Figure 5.3 (a) cropped side view of the static 3D reconstructed face model (b) cropped side view of the 3D reconstructed face model in motion (c) cropped side view of the 3D reconstructed face model using PIL technique.

Instead of moving towards the camera, a smooth surface is subjected to ‘z’ motion away from the camera during the scan time. The comparison of the side views of the 3D reconstructed smooth surface when it is in static, ‘z’ motion away from the camera and the motion corrected using PIL are shown in the Figure 5.4

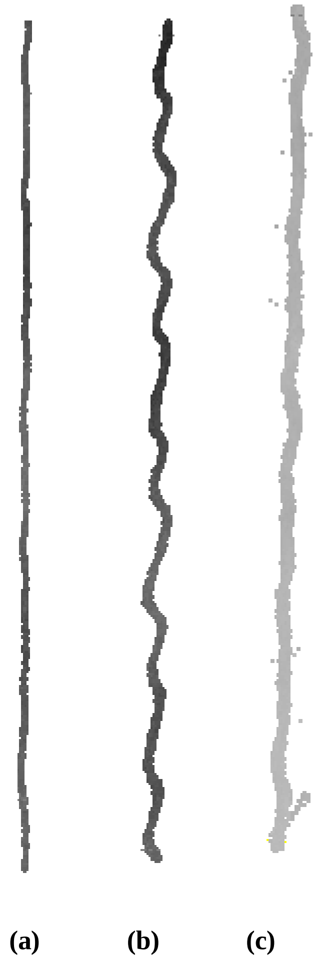


Figure 5.4 (a) side view of the static 3D reconstructed smooth surface (b) side view of the 3D reconstructed smooth surface in ‘z’ motion away from the camera (c) side view of the 3D reconstructed smooth surface using PIL technique.

5.2 3D reconstruction of objects in non-uniform ‘z’ motion

We present the 3D reconstructions of the objects which are subjected to non-uniform ‘z’ motion. Cropped side views of the 3D reconstruction of the objects are presented to show the ripples that occurred due to non-uniform ‘z’ motion during the scan time and also cropped side views of 3D reconstruction of the objects using PIL correction are presented to show the correction of the ripples.

In this section, a smooth surface which is moved back and forth is considered and also a card board hung with strings such that it exhibits free oscillation during the scan time is also considered.

A smooth surface which is moved back and forth in the ‘z’ direction during the scan time is considered and the comparison of the cropped side views of the 3D reconstruction of the smooth surface in static, in non uniform ‘z’ motion and the motion corrected using PIL technique is shown in the Figure 5.5

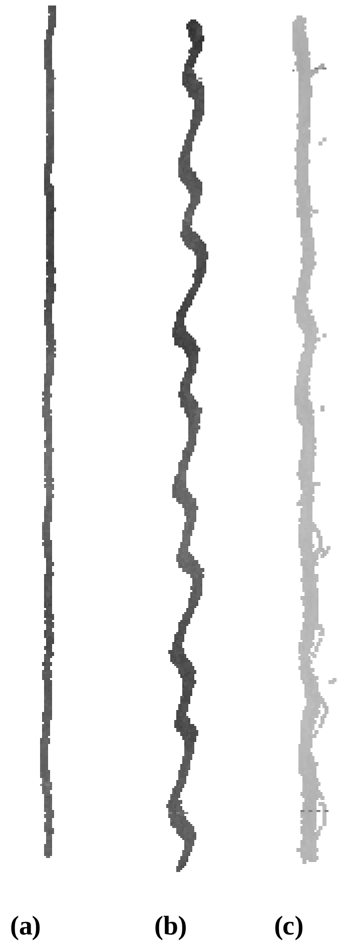


Figure 5.5 (a) cropped side view static 3D reconstruction of a smooth surface (b) cropped side view of the 3D reconstruction of the smooth surface which is subjected to non-uniform motion and (c) cropped side view of the 3D reconstruction of the smooth surface using PIL technique.

Instead of manually moving the object during the scan time, the object that is held with strings is allowed to freely oscillate during the scan time as shown in the Figure 5.6

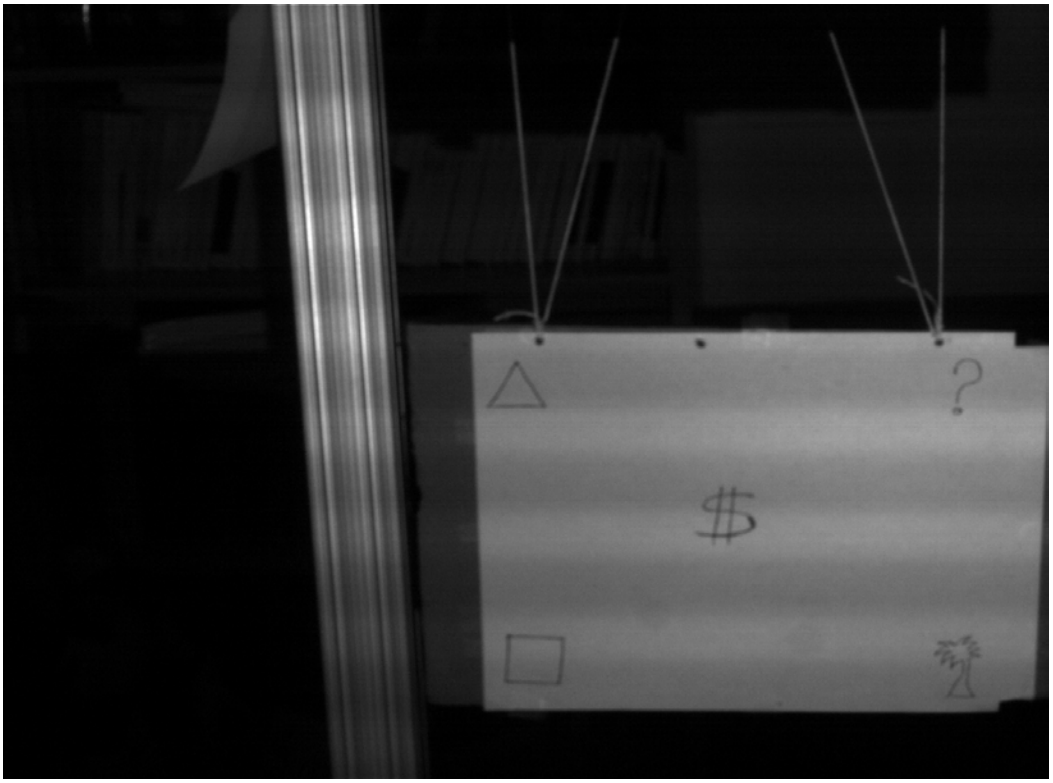


Figure 5.6 object held with strings to freely oscillate during the scan time

Thus, the object is held such a way that it exhibits oscillating motion in ‘z’ direction. The 3D reconstruction of the object when the object is in static, when the object is in non-uniform ‘z’ motion is considered and the 3D reconstruction of the object by using PIL technique is obtained.

The comparison of the 3D reconstruction of the object in static, in non uniform 'z' motion and the motion corrected using PIL is shown in the Figure 5.7

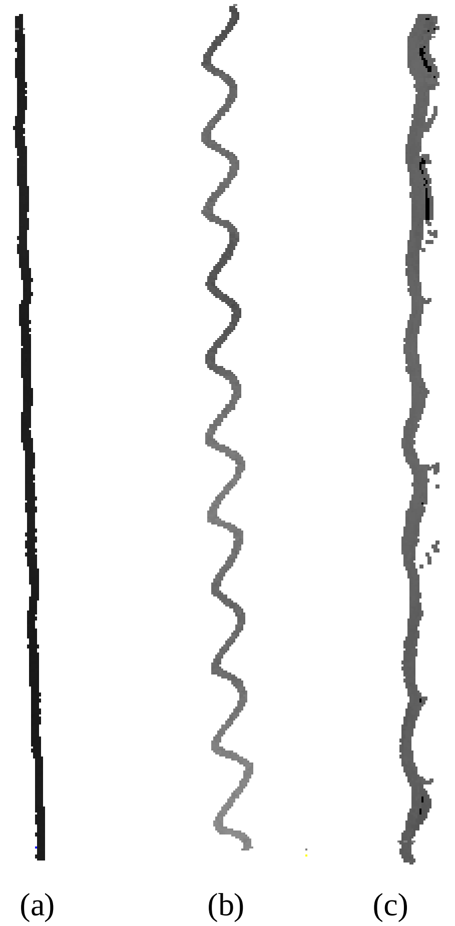


Figure 5.7 (a) cropped side view static 3D reconstruction of the surface (b) cropped side view 3D reconstruction of the surface when it is oscillating during the scan time and (c) cropped side view of the 3D reconstruction of the surface using PIL technique

Chapter 6 Conclusion and Future work

6.1 Conclusion

In this research work, a new technique called “Pattern Interleaving” (PIL) is presented. In PIL technique, an additional pattern (PIL pattern) is introduced to track and correct the ‘z’ motion of the sine waves during the scan time. By correcting the sine wave patterns, we are able to reduce ripples caused by the movement of the object during the scan time. In PIL technique, the offset or the movement of the pixels from one frame to another frame is obtained with the help of tracking ‘snakes’ of the PIL patterns. To correct the PMP patterns, linear interpolation is performed between these snake locations, by taking the offset at the snake locations as the reference. There by using PIL technique we are able to correct the PMP patterns and reduce the motion banding.

Band ripple measurement is performed on the 3D reconstructed object using PIL technique to measure the performance of the PIL technique. The comparison of the 3D reconstruction of the objects in static, in motion and the motion corrected using PIL is performed and we are able to see that the ripples are significantly reduced by using PIL technique.

An approach to correct the lateral movement of the object during the scan time is also presented. In this approach, sobel edge enhancements of the images are obtained and are correlated to track the movement in the lateral direction. We leave additional study of the lateral motion correction to future work.

6.2 Future works

This thesis work is mainly confined to correct the object motion in ‘z’ direction during the scan time. Even though an approach to correct the lateral movement of the object during the scan time is presented, still it needs a robust algorithm to track the lateral movement. There is also a need to correct the movement in the ‘y’ direction. This can be implemented by looking at the phase of the captured PMP patterns. Additionally, we correct the ‘z’ motion by shifting the captured PMP patterns which reduces motion banding but will blur or corrupt high frequency albedo and depth variations. A better approach would be to correct the phase in the Eq. (2.3) and use it to determine the phase map.

Appendix

Assumptions and Limitations of PIL technique:

Though PIL technique is used to correct the ‘z’ motion of the object, there is a limitation on the maximum constant velocity that the surface can move and it is related to triangulation angle, frame rate of the capture and PIL pattern spatial period on the surface. Let us assume a camera with a frame rate of N frames/sec is oriented to have its optical axis parallel to the Z direction. We assume an average triangulation angle of θ degrees, the distance of the surface from the camera is R meters, Δy be the distance between the snakes in the PIL pattern and Δz be the distance of the object in the ‘z’ direction from one frame to another frame just to reach ambiguity that is just before incorrect snake matching will take place as shown in the Figure 7.1

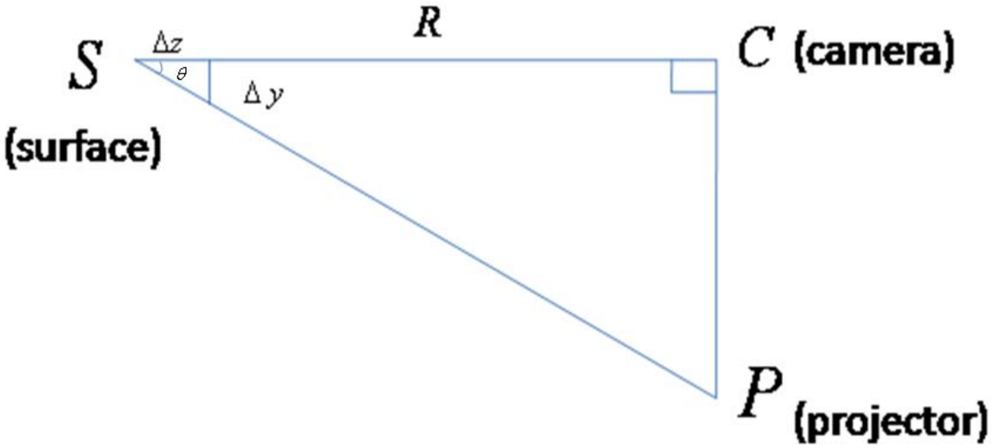


Figure 7.1 SLI arrangement

Calculation of Δz:

From the Figure 7.1, we are able to see two similar triangles as shown in

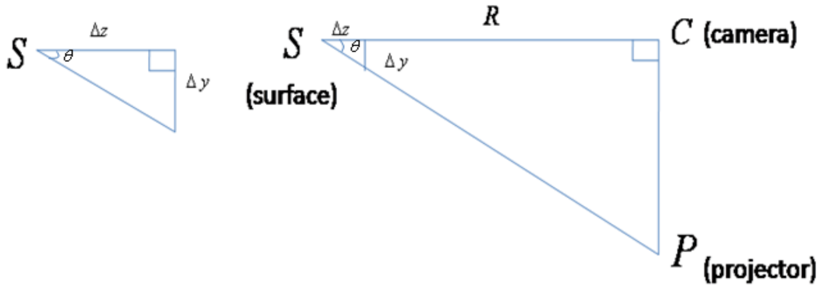


Figure 7.2 Similar Triangles from Figure 7.1

From these two similar triangles, we can write

$$\frac{\Delta z}{R} = \Delta y \big/ CP \tag{7.1}$$

where CP is the distance between the distance between the camera and the projector.

From Eq.(7.1) we get $\Delta z = \left(\frac{\Delta y}{CP}\right) R$ (7.2)

The distance in Eq.(7.2) that would shift one snake to the another. However, we must allow for movement in any direction so the boundary of non-ambiguous movement would be $\frac{\Delta z}{2}$.

Calculation of the maximum ambiguous surface:

Let $v_{\max}$ be the maximum allowed velocity that surface can move between frames. From $\frac{\Delta z}{2}$ and the frame rate we have maximum allowed velocity as

$$v_{\max} = (\Delta z) N / 2 = \frac{\Delta y R N}{2CP} \tag{7.3}$$

Visual C++ code used to correct the object motion in ‘z’ direction

```
void CTrial2Dlg::OnBnClickedCorrect()
{
 //file name variables
 char tfilename[512],tfilereturn1[512],tfilereturn2[512];
 char pfilename[512],pfilereturn[512];
 char filesave[512],filesavereturn[512];
 //open the file
 strcpy_ansi(tfilename,"C:\\Scanner_Settings\\UofKScanner3\\patterns\\tdm#.bmp");
 strcpy_ansi(pfilename,"C:\\Scanner_Settings\\UofKScanner3\\patterns\\pmp#.bmp");
 //get the size of the image
 fileindexall(tfilename,long(0),tfilereturn1);
 bmptest(tfilereturn1,&glbNx,&glbMy);
 //allocation
 glbNindex=(long)glbNx*(long)glbMy;
 pbmpimage=new BMPPIXEL[glbNindex];
 tbmpimage1=new BMPPIXEL[glbNindex];
 tbmpimage2=new BMPPIXEL[glbNindex];
 pcbmpimage=new BMPPIXEL[glbNindex];
 tcbmpimage=new BMPPIXEL[glbNindex];
 //
 pshtimage=new short[glbNindex];
 tshtimage1=new short[glbNindex];
 tshtimage2=new short[glbNindex];
 pcshtimage=new short[glbNindex];
 tcshtimage=new short[glbNindex];
 dimage=new short[glbNindex];
 glboffset=new float[glbNindex];
 //
 pvol=new BMPPIXEL[glbNindex*24];
 int dir;
 short negpoint[20000],pospoint[20000];
 long start,end,n,poffset,toffset,t,p;
 float a,b,offset1,offset2,off;
 double offset;
 n=0;
 t=0;
 p=0;
 short t1negpoint[20000],t2negpoint[20000];
```

```
//intialize the offset to 0
  for(unsigned int i=0;i<glbNx;i++)
  {
          for(unsigned int j=0;j<glbMy;j++)
          {
                  glboffset[j*glbNx+i]=0;
          }
  }
  //get the images
  long inum;
  short iresult;
  for(long inum=0;inum<24;inum++)
  {
          iresult=fileindexall(tfilename,inum,tfilereturn1);
          iresult=fileindexall(tfilename,inum+1,tfilereturn2);
          iresult=fileindexall(pfilename,inum,pfilereturn);
          bmpin(tbmpimage1,tfilereturn1,&glbNx,&glbMy);
          bmpin(tbmpimage2,tfilereturn2,&glbNx,&glbMy);
          bmpin(pbmpimage,pfilereturn,&glbNx,&glbMy);
          //convert to short
          bmp2sht(tshtimage1,tbmpimage1,glbNx,glbMy);
          bmp2sht(tshtimage2,tbmpimage2,glbNx,glbMy);
          bmp2sht(pshtimage,pbmpimage,glbNx,glbMy);
          //nullify the 128's
          for(unsigned int m=0;m<glbNx;m++)
          {
                  for(unsigned int j=0;j<glbMy;j++)
                  {
                          if(tshtimage1[j*glbNx+m]==128){tshtimage1[j*glbNx+m]=0;}
                          if(tshtimage2[j*glbNx+m]==128){tshtimage2[j*glbNx+m]=0;}
                          if(tshtimage1[j*glbNx+m]==255)
                         {
                                t1negpoint[t]=j;
                                t++;
                                if(t>1)
                                {
                                        if((t1negpoint[t-1]-t1negpoint[t-2])>60)
                                        { tshtimage1[t1negpoint[t-2]*glbNx+m]=0;}



                                       if((t1negpoint[t-1]-t1negpoint[t-2])<10)
                                        { tshtimage1[t1negpoint[t-1]*glbNx+m]=0; }
                                }
                  }
                  if(tshtimage2[j*glbNx+m]==255)
```

```
        {
            t2negpoint[p]=j;
            p++;
            if(p>1)
            {
                if((t2negpoint[p-1]-t2negpoint[p-2])>60)
                {
                    tshtimage2[t2negpoint[p-2]*glbNx+m]=0;
                }
                if((t2negpoint[p-1]-t2negpoint[p-2])<10)
                {
                    tshtimage2[t2negpoint[p-1]*glbNx+m]=0;
                }
            }
        }
    }
    t=0;
    p=0;
}
movesht2sht(tcshtimage,tshtimage2,glbNx,glbMy);
movesht2sht(pcshtimage,pshtimage,glbNx,glbMy);
subtractsht(dimage,tshtimage1,tshtimage2,glbNx,glbMy);
//correction code here
for(unsigned int i=0;i<glbNx;i++)
{
    for(unsigned int j=0;j<glbMy;j++)
    {
        if(dimage[j*glbNx+i]==-255)// a negative peak is found
        {
            //look for the direction of the motion
            for(unsigned int k=j+1;k<j+10;k++)
            {
                if(k<=1024)
                {
                 if(dimage[k*glbNx+i]==255)
                 {
                    dir=0; //towards the projector
                    negpoint[n]=j;
                    pospoint[n]=k;
                    n=n+1;
                 }
                }
                }//end of first 'k' for loop
                for(unsigned int k=j-1;k>j-10;k--)
                {
                    if(k<=1024)
```

```
              {
               if(dimage[k*glbNx+i]==255)
               {
                       dir=1;//away from the projector
                       negpoint[n]=j;
                       pospoint[n]=k;
                       n=n+1;
               }
              }
      }//end of second 'k' for loop
      if((dir==0)&&(n>1))
      {
              start=negpoint[n-2];
              end=negpoint[n-1];
              offset1=pospoint[n-2]-negpoint[n-2];
              offset2=pospoint[n-1]-negpoint[n-1];
              a =(offset1-offset2)/float(start-end);
              b =offset1-float(start)*a;
              if(inum==0)//this is the first PMP image
              {
                      glboffset[start*glbNx+i]=offset1;
                      glboffset[(pospoint[n-2])*glbNx+i]=0;
              }//end of 'if' num
              //interpolation
              for(unsigned int p=start;p<=end;p++)
              {
                      off=float(p)*a+b;
                      offset=off+0.5;
                      if(inum==0)//first image
                      {
                              poffset=(off/2)+0.5;
                              toffset=offset;
                      }
                      else
    {
  poffset=((off/2)+0.5)+glboffset[(pospoint[n-2])*glbNx+i];
          toffset=offset+glboffset[pospoint[n-2]*glbNx+i];
    }



 if(((p+poffset)<1024)&&((p+toffset)<1024))
{

    tcshtimage[(p+toffset)*glbNx+i]=tshtimage2[p*glbNx+i];
    pcshtimage[(p+poffset)*glbNx+i]=pshtimage[p*glbNx+i];
```

```
                                  }
                           }//end of interpolation
              tcshtimage[(negpoint[n-2])*glbNx+i]=0;
             if(inum>=1)
           {
               glboffset[start*glbNx+i]=offset1+glboffset[(pospoint[n-2])*glbNx+i];
               glboffset[pospoint[n-2]*glbNx+i]=0;
           }
 }//end of if 'dir=0'
 if((dir==1)&&(n>1))
{
     start=negpoint[n-2];
     end=negpoint[n-1];
     offset1=pospoint[n-2]-negpoint[n-2];
     offset2=pospoint[n-1]-negpoint[n-1];
     a=(offset1-offset2)/float(start-end);
     b=offset1-float(start)*a;
     if(inum==0)//first PMP
     {
         glboffset[start*glbNx+i]=offset1;
         glboffset[pospoint[n-2]*glbNx+i]=0;
      }//end of if 'inum'
      //interpolation
       for(unsigned int p=start;p<=end;p++)
       {
           off=(float(p)*a)+b;
           offset=off-0.5;
           if(inum==0)
           {
              poffset=(off/2)-0.5;
              toffset=offset;
           }
         else
        {
           poffset=(off/2-0.5)+glboffset[pospoint[n-2]*glbNx+i];
           toffset=offset+glboffset[pospoint[n-2]*glbNx+i];
        }



     if(((p+poffset)>0)&&((p+poffset)<1024))
     {
          pcshtimage[(p+poffset)*glbNx+i]=pshtimage[p*glbNx+i];
      }
    if(((p+toffset)>0)&&((p+toffset)<1024))
```

```
            {
                tcshtimage[(p+toffset)*glbNx+i]=tshtimage2[p*glbNx+i];
             }
      }//end of interpolation
     tcshtimage[(negpoint[n-2])*glbNx+i]=0;
         if(inum>0)
       {
           glboffset[start*glbNx+i]=offset1+glboffset[(pospoint[n-2])*glbNx+i];
           glboffset[pospoint[n-2]*glbNx+i]=0;
       }
     }
   }//end of if 'dimage"-2
  }//end of j
    n=0;
 }//end of i
  //convert bmp to sht
  sht2bmp(tcbmpimage,tcshtimage,glbNx,glbMy);
  strcpy_ansi(filesave,"C:\\patterns\\newtdm#.bmp");
  fileindexall(filesave,inum,filesavereturn);
  bmp2bmp24file(filesavereturn,tcbmpimage,glbNx,glbMy);
  sht2bmp(pcbmpimage,pcshtimage,glbNx,glbMy);
  strcpy_ansi(filesave,"C:\\patterns\\newpmp#.bmp");
  fileindexall(filesave,inum,filesavereturn);
   bmp2bmp24file(filesavereturn,pcbmpimage,glbNx,glbMy);
  /*fltzipper(50,3,1,interpoff,glbNx,glbMy);
               short Nsidex,Msidey;
               unsigned char *bimageI;
               long index;
               bimageI=new unsigned char [glbNindex];
               Nsidex=2;
               Msidey=2;
               for(index=0;index<glbNindex;index++) bimageI[index]=10;
                mat5mvfilter(bimageI,interpoff,glbNx,glbMy,Nsidex,Msidey,1);
              seasky(interpoff,glbNx,glbMy,0,255,0);
              flt2bmp(interpoffsetimage,interpoff,glbNx,glbMy);
              strcpy_ansi(filesave,"C:\\interpdy#.bmp");
              fileindexall(filesave,inum+1,filesavereturn);
              bmp2bmp24file(filesavereturn,interpoffsetimage,glbNx,glbMy);
      delete [glbNindex] bimageI;*/

       bmp2volume(pvol,pcbmpimage,glbNx,glbMy,24,inum);

   }//end of inum
delete [] pbmpimage;
delete [] tbmpimage1;
delete [] tbmpimage2;
delete [] pcbmpimage;
delete[] tcbmpimage;
delete[] pshtimage;
delete[] tshtimage1;
delete[] tshtimage2;
delete[] pcshtimage;
delete[] tcshtimage;
delete[] dimage;
delete[] glboffset;

 float* pphase;
 pphase=new float[glbNindex];
 BMPPIXEL*  bmpphase;
 bmpphase=new BMPPIXEL[glbNindex];

 short freq[3],npat[3];
 freq[0]=1;
 freq[1]=8;
 freq[2]=16;
 //npat[0]=8;
 //npat[1]=8;
 //npat[2]=7;
 npat[0]=8;
 npat[1]=8;
 npat[2]=8;
 MFPMP2Phase(pphase,pvol,glbNx,glbMy,24,&freq[0],&npat[0],3);
 flt2bmp(bmpphase,pphase,glbNx,glbMy);
 char phasefile[512];
 strcpy_ansi(phasefile,"C:\\patterns\\phase.bmp");
 //strcpy_ansi(phasefile,"C:\\Scanner_Settings\\UofKScanner1\\Patterns\\phase.bmp");
 bmp2bmp24file(phasefile,bmpphase,glbNx,glbMy);
 //strcpy_ansi(phasefile,"C:\\Scanner_Settings\\UofKScanner1\\Patterns\\phase.byt");
 strcpy_ansi(phasefile,"C:\\patterns\\phase.byt");
 mat5fltfileio(1,phasefile,pphase,glbNx,glbMy);
 delete []pphase;
 delete []bmpphase;
 delete []pvol;
AfxMessageBox("Complete");


}
```

References

1. J. Batlle, E.Mouaddib and J.Salvi “Recent Progress in coded structured light as a technique to solve the correspondence problem”, *Pattern Reorganization*, **31**(7), 963-982, 1998.
2. X.Y.Su and W.S.Zhou, “Complex object profilometry and its application for dentistry” in *Clinical Applications of Modern Imaging Technology II*, L.J.Cerullo, K.S.Heiferman, Hong Liu, H.Podbielska, A.O.Wist and L.J.Eamorano,eds., Proc.SPIE 2132,484-489,1994.
3. G.Sansoni, F.Docchio,U.Miononi and L.Biancardi, “Adaptive Profilometry for industrial applications” in *Laser Applications to Mechanical Industry,* S.Martellucci and A.N.Chester, eds. Kluwer Academic,Norwell,Mass.,1993, 351-365.
4. R.Raskar, G.Welch, M.Cutts, A.Lake, L.Stesin and H.Fuchs, “The office of the future: a unified approach to image-based modeling and spatially immersive displays” presented at SIGGRAPH 98, Orlando, Fla., July 19-24,1998.
5. Veera Ganesh Yalla, Laurence G Hassebrook, “Very high resolution 3-D surface scanning using multi-frequency phase measuring profilometry”, spaceborne sensors II SPIE’s defense and security symposium 2005, Vol 5798-09
6. Laurence G Hassebrook, Daniel L. Lau, “Introduction of Lock and Hold SLI strategy”, Tech. Rep. CSP 06-004, University of Kentucky, Department of Electrical and Computer Engineering, Lexington, KY, USA.
7. M.Maruyama and S.Abe, “Range sensing by projecting multiple slits with random cuts”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15, 1993.
8. O Hall-Holt and S.Rusinkiewicz, “Stripe boundary codes for real time structured - light range scanning of moving objects” in *Proceedings of International Conference on Computer Vision*, 2001.
9. T Weise, B Leibe and L Van Gool, “Fast 3D scanning with Automatic Motion Compensation” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007

10. Soren Konig and Stefan Gumhold, “Image-Based Motion Compensation for Structured Light Scanning of Dynamic Surfaces” in *Proceedings EG Workshop on Dynamic 3D Imaging*, 2007.

11. C.Guan, L.Hassebrook and D.Lau, “Composite Structured Light Pattern for three dimensional video” in *Optics Express*, 11(5), 406-417, 2003.

12. Jielin Li, L.G.Hassebrook and Chun Guan, “Optimized Two-Frequency Phase Measuring Profilometry Light-Sensor Temporal-Noise Sensitivity", *JOSA A*, 20(1), 106-115, 2003.

13. Z.Zhang, “ A flexible new technique for camera calibration”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11), 1330-1334, 2000

14. E.L.Hall, J.B.K.Tio,C.A.McPherson and F.A.Sadjadi, “Measuring Curved Surfaces for Robot Vision”, *IEEE Computer Society* 15(12),42-54, 1982.

15. O.D. Faugeras and G.Toscani, “The calibration problem for stereo”, IEEE Proceedings on Computer Vision and Pattern Recognition,15-20,1986.

16. D.C.Brown, “Close-Range Camera Calibration”, *Photogrammetric Engineering*, 37(8), 855-866, 1971.

17. J.Salvi, X.Armangue, Joan Batlle, “A comparative review of camera calibrating methods with accuracy evaluation”, *Pattern Recognition*, 35, 1617-1635, 2002.

18. R.Y.Tsai, “A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the shelf TV cameras and lenses”, *IEEE Journal of Robotics and Automation*, RA-3, 323-344, 1987.

19. J.Weng, P.Cohen and M.Herniou, “Camera calibration with distorted models and accuracy evaluation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14, 965-980, 1992.

20. J.Wang, F.Shi, J.Zhang and Y.Liu, “A new calibration model of camera lens distortion”, *Pattern Recognition*, 41, 607-615, 2008.

21. G.Zhang, J.He and X.Yang , “Calibrating camera radial distortion with cross-ratio invariability”, *Optics and Laser Technology*, 35, 457-461,2003.

22. De Xu, You Fu Li and Min Tan, “Method for calibrating cameras with large lens distortion”, *Optical Engineering*, 45(4), 043602, April 2006.

23. Z.Zhang, “On the Epipolar Geometry Between Two Images With Lens Distortion”, *ICPR*, vol.I, 407-411, August 1996.

24. L.Ma,Y.Q.Chen and K.L.Moore, “Analytical piecewise radial distortion model for precision camera calibration”, *IEE Proceedings*, no:20045035,2006.

25. Simone Graf and Tobias Hanning, “Analytically solving radial distortion parameters”, *IEEE Conference on Computer Vision and Pattern Recognition*, 2, 1104-1109, 2005.

26. F.Devernay and O.Faugeras, “Automatic calibration and removal of distortion from scenes of structured environments”, *Machine Vision and Applications*, 13, 14-24, 2001

27. C.Guan, “Composite Pattern for Single Frame 3D data acquisition”, *PhD Dissertation*, University of Kentucky, Lexington, KY, USA, December 2004.

28. L.G.Hassebrook and Chun Guan, “Distortion”, *Encyclopedia of Optical Engineering*, edited by R.G.Driggers, published by Marcel Dekker, Inc., New York.

29. Song Zhang and Peisen S. Huang, “Fast three-step phase-shifting algorithm”, *Applied Optics*, **45** (21), 5086-5091, 20 July 2006.

Vita

Raja Kalyan Ram Cavaturu was born at Vijayawada, India on June 3rd 1985. He was awarded Bachelor's of Technology degree from SVHCE, affiliated to Nagarjuna University in April 2006. He is currently working as a Research Assistant at University of Kentucky.