Development of 3-D Surface Data Acquisition System Using Non-Calibrated Laser Alignment Techniques

Kim Chuan Lim

A thesis submitted in partial fulfilment of the requirements of
Sheffield Hallam University
for the award of
Doctor of Philosophy

September 2009

DECLARATION

This is to certify that I am responsible for the work submitted in the thesis, and that neither the thesis nor the original work contained therein has been submitted to this or any institution for a higher degree.

Signature: …..

Name: **Kim Chuan Lim**

Date: **September 2009**

Development of 3-D Surface Data Acquisition Systems Using Non-Calibrated Laser Alignment Techniques

Abstract

The development of cost-effective, but robust desktop-based 3-D surface reconstruction systems have recently been made possible by the increase in computing power of low cost personal computers and the availability of high-resolution but inexpensive webcams. Low cost 3-D surface reconstruction systems can be implemented either by adapting stereo vision or by optical triangulation approaches.

Optical triangulation approaches require the estimation of the parameters of a 'virtual triangle'. This is made up of the centre of a calibrated camera, an external light source and the point on the surface of the object illuminated by the external light source. To achieve this, the traditional optical triangulation based 3-D surface data acquisition systems required the external optical source to be manually aligned to a reference coordinate frame of the system.

The objective of this study was to design a novel optical triangulation based 3-D surface data acquisition system that could accurately determine the relative pose between the camera and the external optical source, without any manual intervention, in real time, using low cost off-the-shelf, readily available components.

The above objective was achieved by designing and implementing a system framework, consisting of a webcam, two planar objects and a freely moveable laser line emitter. A major advantage of the developed freehand scanning system is that the user may repeatedly scan the surface of interest, to generate denser 3-D points to represent the surface, and hence increase the signal to noise ratio by “brushing” across the surface several times.

Another major issue with the traditional triangulation process is that one needs to know the pose of the external plane, for *e.g.* the plane formed by the divergence of a single optical light source passing through a biconcave lens. In the current study the pose of the external plane, the laser plane, was estimated by using the geometric constraints obtained through the process of the intersecting the laser plane with two planar objects. The constraints are the pair of 2-D straight lines formed, on the two planar objects and captured in the camera image. These constraints were projected to the camera 3-D coordinate frame. A key element in determining the projection parameters is to accurately locate the salient features lying on the planar objects. Extensive studies were carried out and an accurate salient features detector was subsequently identified. Since the projected 2-D constraints should lie on a plane in 3-D, novel methods were developed and their performances were critically evaluated. The pose of the laser plane was thus estimated on the fly, during the freehand 3-D surface data acquisition process, without using any extraneous attachments or sensors. This eliminated the need for any specialised and expensive attachments to find its pose.

The developed 3-D surface acquisition system was also applied to another real-world application, *viz.* calibrating the pose of the laser plane in a turntable based 3-D surface reconstruction system. A robust method was developed to automate the turntable pose calibration process. Accurate and robust procedures were implemented to automate the traditional laborious calibration process.

A further real-world application was designed and implemented. It consisted of two vision based respiration monitoring methods; namely mono-vision and stereo-vision. These were developed to estimate the motion of the human chest during the respiration process. The estimated motion was subsequently used to successfully determine the respiration rate.

LIST OF PUBLICATIONS

In the course of completing this study, the following article was published.

1. KC Lim, M. Boissenin, BP Amavasai, and R. Saatchi, "Development of a desktop freehand 3-D surface reconstruction system", *7th IEEE International Conference on Cybernetic Intelligent Systems*, 2008. CIS 2008, pp 16, 2008

ACKNOWLEDGEMENTS

I would like to convey my deepest sincere gratitude to my director of studies, Dr. Reza Saatchi, for encouraging me to write up and completing the thesis, in the final year of the study. My appreciation to him for his interest and enthusiasm, to read, correct, and comment on the thesis.

I take this opportunity to express a deep sense of gratitude for my main supervisor, Mr. Arul Nirai Selvan, who through his continuous guidance, in the last four years, led me to the completion of this study. His encouragement, valuable ideas, and fruitful advice have been constants source of inspiration throughout my project.

I would like to thank my previous and existing supervisors, Dr. Bala Amavasai, Dr. Jon Travis, Dr. Lyuba Alboul, Dr. Christine Ferris, for their various help and support throughout this study. Special thanks to Dr. Lyuba Aboul and Dr. Bala Amavasai for spending their precious time correcting and providing feedback for my work and thesis.

I would also like to thank my laboratory peers in MMVL, Dr. Amir WFW Othman, Dr. Manual Boissenin, Dr. Stephen Mckibbin, Mr. Jan Wedekind, Mr. Kang Song Tan, for all the joys and supports I received from them. It has been a very valuable time and great moments we spent in the lab.

Thanks to the support from Sheffield Hallam University, especially Mr. Brian Didsbury from Precision Workshop. His gracious effort in producing various precise mechanical rigs enabled my research study to be able to carried out smoothly.

Thanks to the government of Malaysia and the University Teknikal Malaysia Melaka for sponsoring my PhD studies at Sheffield Hallam University.

Last but not least, I wish to express my love and appreciation to my family, whom I love so much, for their consistence supports and encouragements.

Glossary

In the case if

A point which is situated in three dimensional space, is referred to as a 3-D point in what follows.

A straight line under consideration, lies in a particular two-dimensional or three-dimensional space, we will refer to this line as a 2-D or 3-D line, respectively.

A surface situated in three dimensional space is commonly called as a 3-D plane.

Pose is referred to as position and orientation in 3-D.

Nomenclature

A three dimensional row vector is denoted by $[x, y, z]$. Its transpose, converts it to a column vector *i.e.* $[x, y, z]^T = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$.

Acronyms/Abbreviations

In the case if

3-D: Three dimensional

BPM: Breath per minute

CCD: Charge-coupled device

GUI: Graphical User Interface

H&S corner detector: Harris & Stephen corner detector

HSV: Hue, Saturation and Value

LSE: Lease Square Estimations

MIA: Moment of Internal Analysis

PC: Personal computer

RANSAC: Random Sample Concensus

RGB: Red, Green Blue

WLSE: Weighted Least Square Estimations

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1: Introduction

1.1 Background

The use of a three dimensional (3-D) surface data acquisition system allows us to obtain 3-D geometry information of an object in digital form. This information can subsequently be processed, analysed and stored digitally. The process of acquisition of 3-D surface data, representing the surface of an object, can be carried out using different types of electronic sensors.

A distance measuring electronic sensor can be used to perform the task of acquiring 3-D surface data. The output of the distance measuring sensor is a single measurement, which is the relative distance of a specific point lying on the object from the origin of the sensor. The 3-D surface information of the object can then be acquired by first moving the scanner to other known locations and then measuring the relative distance to the surface of the object from those locations. To represent the surface of an object, relative distances to a minimum of three different points on the surface of the object, from the corresponding three different known locations, need to be determined. Let the three measured distances to the surface of the object from the three known locations, $(0,0)$, $(\delta_{x1}, \delta_{y1})$, $(\delta_{x2}, \delta_{y2})$, be (z_0, z_1, z_2) respectively. Using the two sets of information, the coordinates of the three 3-D points P_0, P_1 and P_2 on the surface of the object could be found as $[0,0,z_0]^T, [\delta_{x1}, \delta_{y1}, z_1]^T, [\delta_{x2}, \delta_{y2}, z_2]^T$ respectively. Using the estimated 3-D locations of the three points on the object, the surface of the object, enclosed within the three points, can be approximated by interconnecting the points in a triangle as shown in Figure 1.1.

The process could be repeated by systematically moving the distance measuring sensor, with a known displacement, using for example a mechanical rig, around the object. At the end of this scanning process, a set of 3-D locations are estimated on the surface of the object. Connecting this set of 3-D locations with a mesh of polygons yields a cloud of 3-D points representing the surface of the object being scanned.

The major shortcoming in the above approach is the need for an arrangement to systematically move the distance measurer around the object. This could be achieved by, instead of attaching the distance measuring sensor to a mechanical rig and systematically scanning the object, the sensor is removed from the mechanical rig, and used to freely sweep across the surface of interest by hand. This freehand scanning process allows more densely distributed data points to be recorded on the surface of the object. The dense set of data points ensures a more accurate reconstructed surface using polygonal meshing process.

In the current study a freehand scanning process to reconstruct the surfaces of 3-D objects has been designed and implemented.

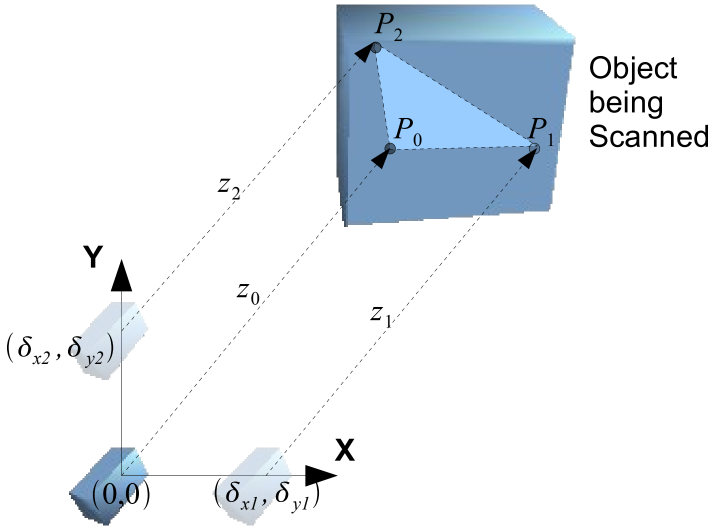


Figure 1.1: Figure illustrating the digitisation of the 3-D surface of an object, which is being scanned, using a single distance sensor.

A 3-D surface data acquisition system can be implemented by using mechanical actuators or by electronic components. The resolution of the mechanical arrangements is limited to the arrangement of their actuators. They also introduce constraints related to the manner the surface data is collected.

It is possible to use large, and expensive, electronic position sensors for collecting the data but the extra weight, introduced by the heavier sensor, would need to be physically carried by the user. Freehand systems provide a facility for the users to freely move the external optical source (*e.g.* a laser emitter) with their hands. The freehand scanning system allows the user to repeatedly scan the same surface of interest, to generate a denser cloud of 3-D points for representing the same part of the surface and hence increase the signal to noise ratio by “brushing” the surface multiple times.

The rational and motivation of the current study was related to the need for objects' surface information in many areas. The following surface information application areas have been investigated in this study:

- Reconstructing the surface of small parts and objects whose surface information cannot be easily captured by other methods.
- Reconstructing the surface of human anatomy.
- Monitoring human respiration rate by determining the changes in the surface of the chest or abdomen.

1.1.1. Reconstructing the surface of objects

Surface information of small objects with intricate details is very hard to capture in digital form using means such as 3-D digitiser [1]. For these situations, the freehand 3-D scanning system designed and implemented in this study is the best suited. Moreover, the freehand 3-D scanning system can handle small as well as larger objects because the system does not require any specific equipment or attachment like a mechanical arm or translator.

1.1.2. Reconstructing the surface of human anatomy

Capturing the surface details of the human anatomy is much more complicated than the surface information of inanimate objects for the following reasons:

- Physical contact needs to be avoided.
- The surface needs to be scanned speedily, since one cannot expect the subject to keep still for long time periods.

1.1.3. Baby Active Breathing Evaluator (BABE)

Respiration rate is a physiological parameter used by clinical experts to monitor various illnesses. An abnormal respiratory rate is indicative of an illness such as cardiac arrest or the need for admission to the intensive care unit (ICU). Respiration rate is better than other physiological measurements, such as pulse and blood pressure for identifying between the stable patients and patients at risk. The ability to accurately measure respiration rate can significantly help in reducing the mortality of the patients and thus makes significant life savings for the National Health Service.

Guidelines were published by the National Institute for Clinical Excellence (NICE), UK, in 2007 stating that measurement of respiratory rate is obligatory in the assessment of children with fever in the hospital setting [2]. At least one assessment of the respiratory rate needs to be carried out for every child visiting hospital's emergency department. The results of the respiratory rate measurements are subsequently used to monitor the changes in the state of health of the patient. The respiration rate of every patient in children's wards and Intensive Care Units is monitored at least once every few hours.

Visual observation is the most basic method for measuring respiration rate. In this method, the back of hand is kept very close to the patient's nose to detect changes in the exhaled air temperature and air flow. This approach is subjective [3] and its accuracy is thus questionable. Equipment like nasal temperatures probes (*e.g.* thermistor) and chest strain gauges require the sensor to be attached to the patient's body. This requirement (especially for children) can cause discomfort and result in the distress. The attachments can can also interfere with breathing. In order to deal with these limitations non-contact methods of respiration monitoring need to be developed. In this study a vision based respiration monitoring system is devised.

1. 2 Aims and objectives of the study

The aim of this study was to design and implement a 3-D surface data acquisition system. The study led to creation of a vision based, freehand 3-D surface data acquisition system for reconstructing the surfaces, in particular of human anatomy, using a personal computer. The overall objectives of this study were to:

- Identify the limitations of existing vision based 3-D surface reconstruction systems.

- Investigate, from the system point of view, the essential elements required to design and implement a vision based, freehand 3-D surface data acquisition system.

- Critically evaluate the accuracy of identifying the corners lying on the planner object. Corner detection is an essential element of the 3-D surface reconstruction system implemented in this study.

- Design and implement an accurate method of determining the pose (*i.e.* the position and orientation) of the laser line emitter, in real time, during the free hand scanning process.

- Based on the above work, implement a position sensor-less, personal computer based, 3-D surface data acquisition system, using off-the-shelf equipment.

- Investigate the effectiveness of the devised 3-D surface reconstruction methods for turntable calibration and real time, non-contact respiration monitoring.

1.3 Research methodologies

A major part of the work carried out in this study consisted of developing and implementing a freehand surface data acquisition system. The basic operations of the system consisted of recording a camera image (from a USB or Firewire camera), low level image processing and displaying the information in real-time. These processes were achieved by using a range of in-house software, *e.g.* MIMAS toolkit [4] and HornetsEye [5], developed at Sheffield Hallam University. Using a basic geometry computer vision feature [6] available in MIMAS toolkit, the possibilities of creating a freehand 3-D surface data acquisition system was explored. The ability to process camera images in real-time with the geometry computer vision techniques was subsequently used to scan surfaces.

1.4 Contribution

The main contributions of this study were:

i. Developing a framework to construct a system to perform a vision based, freehand 3-D surface data acquisition system. This framework had all the critical elements for creating an accurate 3-D surface data acquisition system. The framework is described in Chapter 3.

ii. A critical survey and evaluation of the most suitable salient features detector for a vision based, freehand 3-D surface data acquisition system was carried out. A critical quantitative comparison of the template based corner detector with several other conventional corner detectors was performed. The results of the evaluation are discussed in Section 4.4 and the conclusions of the evaluation are presented in Section 4.5.

iii. Using the principle of RANdom SAmple Consensus (RANSAC) [7], two novel methods for estimating the pose of the laser plane were designed and implemented. These are discussed in Section 5.2.4. The implemented methods were critically evaluated (Section 5.4) and their performances are discussed in Section 5.5.

iv. Using the developed laser plane pose estimation technique in Chapter 5 and the principle of planar homography, the process of turntable calibration, with extensive manual intervention, was improved. An automatic turntable calibration technique using various procedures was proposed. This is described in Chapter 6. Using the developed automatic turntable calibration technique, a turntable based 3-D surface reconstruction system was calibrated. The results of the surface reconstruction with the proposed system are given at the end of Chapter 6.

v. Mono and stereo vision approaches to monitor respiration rate were devised. These are discussed in Chapter 7.

1.5 Organisation of the thesis

A literature review of the existing freehand 3-D surface data acquisition system is provided in Chapter 2. A more focused literature survey, regarding all the essential elements, from the system point of view, is given in Chapter 3.

Using the identified framework, a freehand 3-D surface data acquisition system was constructed. The developed system, along with the result of 3-D surface data acquisition and the reconstructed surfaces, are provided in Chapter 3.

An essential requirement of the designed freehand scanning system was the ability to estimate the salient features lying on the calibration planes. In Chapter 4, three categories of salient features detectors are reviewed and their performances with respect to the requirement of the system are critically evaluated. The results of the evaluation are shown and discussed at the end of the chapter.

Another requirement for the freehand 3-D surface reconstruction system was that it must be able to accurately estimate the pose of the laser plane in real-time. A number of approaches for this purpose are discussed in Chapter 5.

In order to perform a full 3-D scanning of objects, traditional 3-D surface reconstruction systems require rotation of the object to be scanned on the turntable. Using geometric principles, the laser plane and the camera are manually aligned to the coordinate axes of the turntable before the scanning process. This manual operation is laborious, time consuming and error prone. Errors introduced during the alignment of the components result in an inaccurate estimate of the 3-D location of the points lying on the surface of the object. A set of auto-calibration procedures, to estimate the pose of the components of the turntable based surface reconstruction system, are proposed in Chapter 6.

A critical review of the existing respiration monitoring methods are presented at the beginning of Chapter 7. Machine vision based methods, using mono vision and stereo vision are proposed to estimate the breathing rate. The proposed methods were evaluated in the laboratory environment, with a baby mannequin. Chapter 8 concludes this thesis and future research directions are proposed.

Chapter 2: Literature Review and Background Information for 3-D Surface Data Acquisition System

2.1 Introduction

Recently there has been renewed interest in the field of 3-D surface data acquisition [8] [9][10][11]. Some of the main factors contributing to this are:

- Extensive improvement in the price over performance of personal computers. The U.S. Consumer Price Index (CPI) of personal computers is dropped by a ratio of around 731.7% (July 1999 (CPI: 591.5), July 2009 (CPI: 80.838)) [12].

- Vast improvement in the price performance of electronic digital data capturing equipment [12].

- Creation of 3-D surface data information applications in a variety of fields from healthcare [13] to medical education [14], entertainment and gaming [15] to biometric applications [16].

The techniques and technologies to acquire 3-D surface data has improved over the years. In earlier years, one required dedicated equipments such as a linear translator [17], a scanning rig [18], or an articulated arm [19] in order to move the sensor around the scanning object and collect the 3-D surface data. Recently however, the field has developed to a level that one only needs to have just a digital image capturing equipment [20][11]. The remaining parts of this chapter will discuss the background information relevant to 3-D data acquisition and subsequently, outlines the techniques and technologies to perform 3-D data acquisition.

2.2 Background to 3-D surface data acquisition

The techniques and technologies, developed for 3-D data acquisition, mostly use the method of estimating the parameters of a triangle. The triangulation is normally formed by the data capturing mechanism, a point on the surface of the object being scanned and yet another source which could be another data capturing mechanism or a light source [21]. There are other techniques which make use of difference methods like structure from motion [22][23], albedo map and texture height field [20].

Among these mechanisms, optical based systems are very popular. The reasons are, their relatively low cost (a cheaper camera), a greater number of data points on the surface of the object that could be estimated per image, and the ease of their implementation (*i.e.* triangulation).

The process of triangulation provides a measurement of relative distance of a physical point to the optical centre of the camera. The triangulation is formed by using either a single camera with an extra optical source, (*e.g.* laser point/lines emitter, projector) or two cameras (stereo). The model of the camera and the principle of triangulation are explained in the following section.

2.2.1 The camera pinhole model

The camera was modelled by using a camera pinhole model [24]. The camera pinhole model expresses the mathematical relationship between the coordinate of a point in 3-D and its projection onto the camera image plane, in an ideal pinhole camera configuration (see Figure 2.1). The aperture of the camera is described as a point and a lens is not used to focus the light. This first order approximation of the model does not take into account the distortion types, *e.g.* barrier distortion and chromatic distortion caused by the imperfect lens, hence, the validity of the model is completely dependent on how close is the used camera with respect to the ideal pinhole camera model. Equations (2.1) and (2.2) are used by the camera pinhole model to determine the location of a point in the 2-D image plane (u, v) with respect to the corresponding 3-D coordinate point (X, Y, Z) .

$$\lambda u = f X \tag{2.1}$$

$$\lambda v = f Y \tag{2.2}$$

where f is the orthogonal distance between the pinhole to the image plane, and λ is equal to $\frac{1}{Z}$.

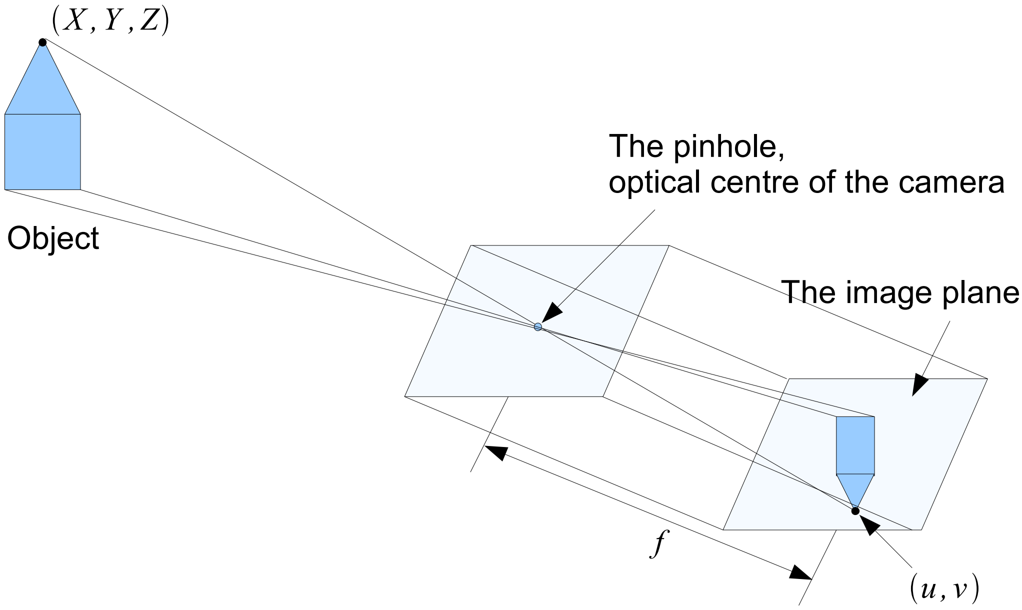


Figure 2.1 The camera pinhole model.

2.2.2 The principle of triangulation

A virtual triangle is formed by three points (C_1, C_2, C_3) as shown in Figure 2.2. A triangle can be defined with two angles (θ_1, θ_2) and known length b. Knowing the parameters of the triangle, the distance l, to the point C_3 from the origin of the camera, could be estimated using the sine law.

$$l = \frac{b \sin(\theta_2)}{\sin(180 - \theta_1 - \theta_2)} \tag{2.3}$$

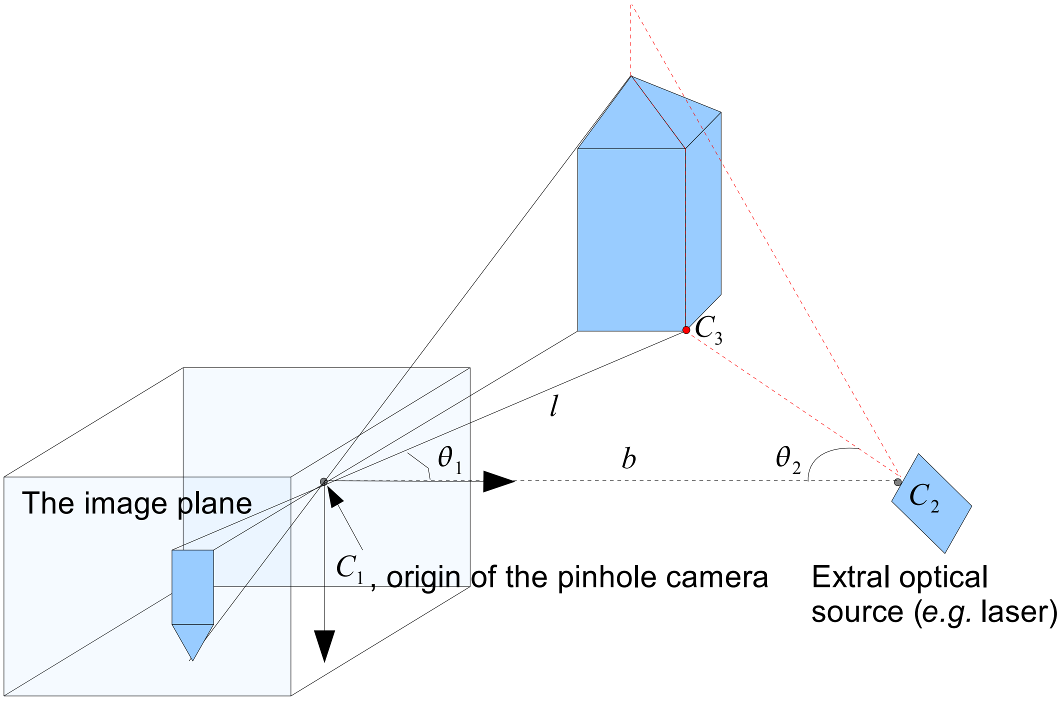


Figure 2.2: Using camera pinhole model with an extra optical source to perform the triangulation.

To perform triangulation, one needs to know the relative location and orientation, (*i.e.* the pose) between the camera and the extra optical source. This can be achieved either by manually aligning them to a defined reference coordinate frame of known pose, or by attaching a position sensor that provides the pose. The position sensor could be either mechanical or electrical. Thus, the optical based 3-D data acquisition system could be categorised as the types with and without the position sensor. Since a major contribution of the current study is to design a method of 3-D data acquisition without the position sensor, the following sections provide a review of these two categories.

2.2.2.1 Three dimensional data acquisition with position sensor

Traditional 3-D reconstruction systems use different mechanical systems to move the camera and the laser plane emitter systematically around the object being scanned. These mechanical systems can use a linear translator [17], a scanning rig [18], or an articulated arm [19]. A mechanical type system can also be a turntable [25][26][27], to rotate the object. The images of the deformed laser line, which is the intersection of laser plane and the scanned object, are captured using the camera. The points on the laser line are extracted and then triangulated to produce a cloud of points in 3-D. These points represent the surface of the scanned object [28].

To obtain the pose of the laser plane the laser line emitter could be attached to a mechanical articulated arm [19], or wireless sensors [29] or optical motion tracker [30] that could be tracked, attached to the laser line emitter and hence, to provide the pose of the laser plane.

There are other non-stereo optical based systems, where the position of the camera needs to be kept static throughout the scanning process. For example in the structured light based methods[21][26][31], several scans are required from different camera view points of the scanned object. The resulting clouds of points in 3-D, from different camera view points, are then registered using a software by correspondence matching of the acquired 3-D points [32].

2.2.2.2 Three dimensional data acquisition without position sensor

The methods of estimating the pose of the laser plane, making use of position sensors, have limitations. For example, the articulated arm might limit the movements of the operator, due to the extra weight of the attached position sensor or the markers may become occluded and thus disturb the scanning process. These limitations are avoided in the freehand scanning system.

Another major advantage of a freehand scanning system is that it is not limited to the resolution of the position sensor. Instead, the user can repeatedly scan the same part of the surface of interest, to generate more densely distributed 3-D points to represent the object's surface and hence increase the signal to noise ratio by “brushing” across the surface several times [32].

For the freehand scanning system, one needs to know the pose of the laser plane for the triangulation process. Many novel methods were proposed to estimate the pose of the laser plane, as discussed below.

Bouguet [21] implemented a 3-D surface reconstruction system using a calibrated camera, a static light source, and a stick to cast a shadow line onto the surface of the object being scanned, lying on a flat surface, *e.g.* a table. A virtual shadow plane was hence formed by the shadow line and the light source. The flat surface was used as one of the reference plane to resolve the pose of the shadow plane. The pose of the surface of the table, with respect to the camera coordinate frame, was initially determined. The process of light source calibration was performed to resolve a 3-D coordinate of a point, with respect to the camera coordinate frame, representing the source of the light. Once the two calibration processes were completed, the stick was slowly waved across the object to be scanned to acquire the 3-D surface data points. A series of camera images, with a shadow line intersecting the flat surface and the object to be scanned, were captured and stored. The shadow points lying on the surface of the object to be scanned, in each image, were extracted and used by the process of triangulation to generate 3-D points representing the surface. The process of triangulation requires knowledge of the pose of the shadow plane in each image. The pose of the shadow plane was determined by using three points; a point representing the source of the light, and two points from the shadow, lying on the flat surface, not occluded by the object to be scanned. To achieve maximum accuracy, the two points from the edge of the shadow should be as separate as possible, *e.g.* at the beginning and the end of the edge of the shadow. A cloud of 3-D points representing the surface was hence produced at the end of the scanning. In the case if the source of light is not static, *e.g.* sunshine, a second flat reference surface can be used to determine the pose of the shadow plane. Some more similar studies can be found from Bouguet' website [33].

In 2008, Winkelbach *et. al.* [32] designed and implemented a very similar system by replacing the shadow mechanism with a laser line emitter. Two planar objects implanted with some markers at known locations were installed as the background planes, one on the left and one on the right. A calibrated camera was installed at a distance such that its field of view covered the two planar objects. The poses of the two planar objects were subsequently estimated. The object to be scanned was placed between the two planar objects and the calibrated camera. A camera image was captured and stored as a background image. The process of 3-D surface data acquisition was started, the laser line emitter was swept across the object to be scanned, intersecting both the background planes, and the images were captured. The laser line in the newly captured image was detected by using image subtraction method, the difference between the newly captured image with laser and the background image provided the location of the laser line in the image. A RANSAC [7] based method was proposed by Winkelbach *et. al.* to estimate the pose of the laser plane, formed by the laser light source and the laser line, on the fly. Three detected laser pixels, lying on the background plane, resulting in three surface points in camera coordinate frame, were randomly selected to estimate the pose of the laser plane.

The result 3-D points cloud of the optical based 3-D surface data acquisition system is limited to the field of view of the camera. To perform a full reconstruction of the object to be scanned, multiple scanning with different view of the object to be scanned were performed and multiple clouds of 3-D points were created. A registration technique was then needed to register all the scanning results. To deal with this issue, a fast surface registration technique was proposed by Winkelbach *et. al.* [32] to register multiple scanning. A free open source surface registration software is provided by MeshLab [34] to perform a similar task.

A real time 3-D model acquisition system with multiple stripes was designed by Rusinkiewicz [10] using a camera and a projector. The relative pose between the projector and the camera was calibrated. Knowing the locations of each projected stripes, 3-D points were generated by using the process of triangulation. The boundary of the stripes lying on the object were tracked between camera image frames and hence the object was localised during the process of scanning. With this tracking ability, the object could be rotated in small increments to create a full 3-D model.

Instead of applying a software registration technique to register the scanning results from multiple views, Zagorchev and Goshtasby [35] proposed the use a hardware reference frame method to perform the registration in their laser based 3-D surface data acquisition system. A reference double-frame was used as the coordinate system of the object to be scanned. The reference double-frame consisted of two frames with different size with known distance in between them. The object to be scanned was placed in between the frames. Only one of the reference frame was used from each camera view. This was the one between the object to be scanned and the camera. The pose of the reference frame was hence estimated. During the scanning process, the pose of the laser plane was estimated by intersecting the laser plane with the reference frame. Since the pose of the reference frame was previously determined, four points (two points from each vertical bar) were used to determine the pose of the laser plane. All the acquired surface data points (previously addressed, *e.g.* the reference plane method) in the camera coordinate frame were transformed into the reference double-frame.

Hebert [8] added an extra pair of camera and laser plane, along with a projector to overcome the disadvantages of using reference planes/frames. Unlike the previously discussed freehand laser scanner, a laser line emitter was physically attached to each camera and the pose with respect to each camera was calibrated. The cameras and the laser emitter were firmly attached to a hand-holdable rig and the relative pose between the two cameras were calibrated. The created self-referenced hand-held laser scanner made use of projector to project light pattern to the scene to create virtual landmarks. Making use of the advantage of stereo camera configuration, the initial landmarks (minimum three are required) visible within the common view of both cameras were firstly triangulated to provide an initial location of the hand-held laser scanner. A cross shape was formed by intersecting the laser lines. This was made visible by the two cameras. The two camera images with the illumination of laser lines on the object to be scanned were captured. Only the laser line belonging to the camera was triangulated to create the 3-D points representing the surface of the object to be scanned. The unused laser line was used by the opposite camera to validate the computed 3-D points. The change of landmarks, entering or leaving the scene, in the new stereo images were used to register the newly triangulated 3-D points to the initial location of the hand-held scanner. In each of the subsequent scanned scene. The study was commercialised and now available in the market [36]

Using the colour data available in the scene being scanned and triangulation method, Popescu *et. al.* [9] took a similar challenge, without using any reference planes/frames to perform the registration, to acquire 3-D points of the scene being scanned. Sixteen low cost laser points pointers and a colour camera were used by Popescu *et. al* to perform the 3-D surface data acquisition. The sixteen laser point pointers along with the camera were mounted to a rig and the locations of each of the laser point pointer, with respect to the camera were calibrated. All the sixteen laser blobs were detected and triangulated during the scanning process. A novel algorithm based on dense colour and sparse depth (result of triangulation) was subsequently used by Popescu *et. al.* to perform fast registration of the scanning result.

2.3 Conclusion

With the publicly available image processing software libraries, established extensive studies of geometry vision algorithms, increasing computing power of personal computer and low cost digital image acquiring system, the implementation of a position sensor-less, desktop based freehand 3-D surface data acquisition system was made possible. A literature survey of different kind of position sensor less 3-D surface acquisition systems, with the know of author, was produced and summarised in this chapter.

Chapter 3: Essential Requirements for a Freehand 3-D Surface Reconstruction System

3.1 Introduction

The essential requirements of a freehand 3-D surface reconstruction system developed in this study are described in this chapter. The system can be divided into two parts, namely the freehand laser scanning part and the 3-D surface reconstruction part. The freehand laser scanner was constructed by using readily available off-the-shelf components, namely a laser line emitter, two planar objects and a webcam (see Figure 3.1). The output of the freehand laser scanning is shown as a cloud of 3-D points. An open source 3-D surface reconstruction software, like PowerCrust [37] *etc.*, was used to reconstruct a 3-D surface from the cloud of 3-D points. An important characteristic of the designed 3-D laser scanner is that it allows the user to hand sweep the laser line across the object to be scanned.

The set up of the freehand laser scanner system is shown in Figure 3.1. The planar object and the object to be scanned are placed at the centre of the horizontal plane. Figure 3.2 illustrates the system block diagram of the designed freehand 3-D scanner.

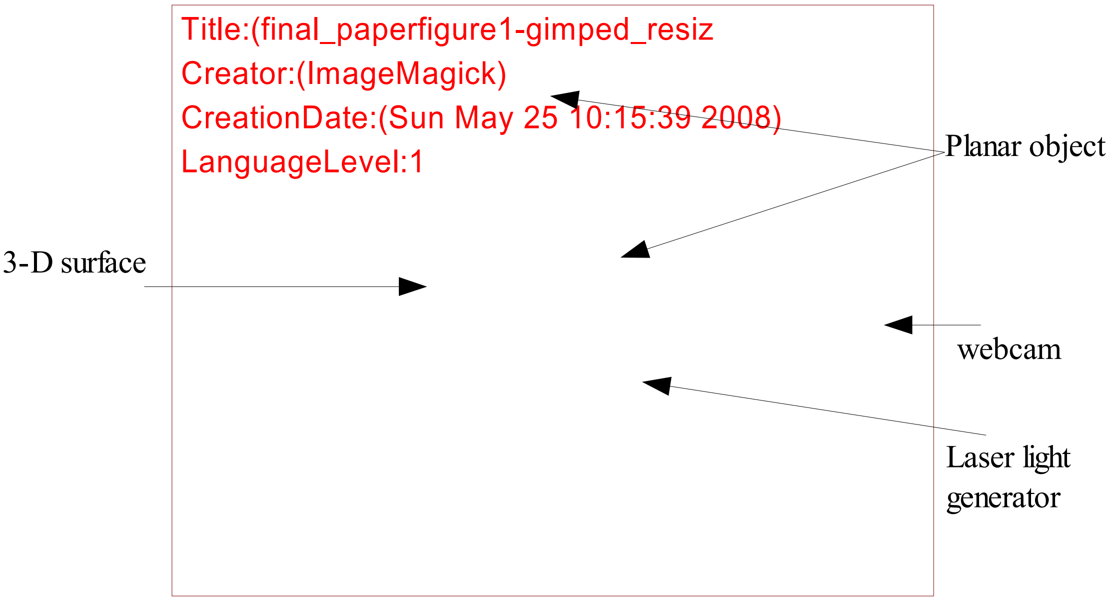


Figure 3.1: Set up of the 3-D reconstruction system.

3.2 Overview of the components of the freehand 3-D scanning process

An overview of the hardware and software components of the freehand 3-D scanning process, shown in the system block diagram Figure 3.2, is provided below.

An off-the-shelf webcam's intrinsic parameters were estimated using a calibration process. The calibrated camera and the MIMAS [4] machine vision software library were then used by the system to acquire images. Then an estimate of the corners of a regular planar pattern were obtained.

Once the camera intrinsic parameters were estimated, the camera's image acquisition properties like focus, gain *etc*. were not adjusted.

The calibrated camera was placed, in a fixed location, in such a way as to be able to view the object to be scanned and the two planar reference objects. The image of the planar object with the checker board pattern on it was captured and stored. This image is referred to as the background image (Img_{bg}). Using Img_{bg} of the planar object, the pose of the horizontal plane $({}^{C}T_{H})$ and the pose of the vertical plane $({}^{C}T_{V})$, with respect to the camera frame as global reference frame, were estimated; see Figure 3.3 for more details.

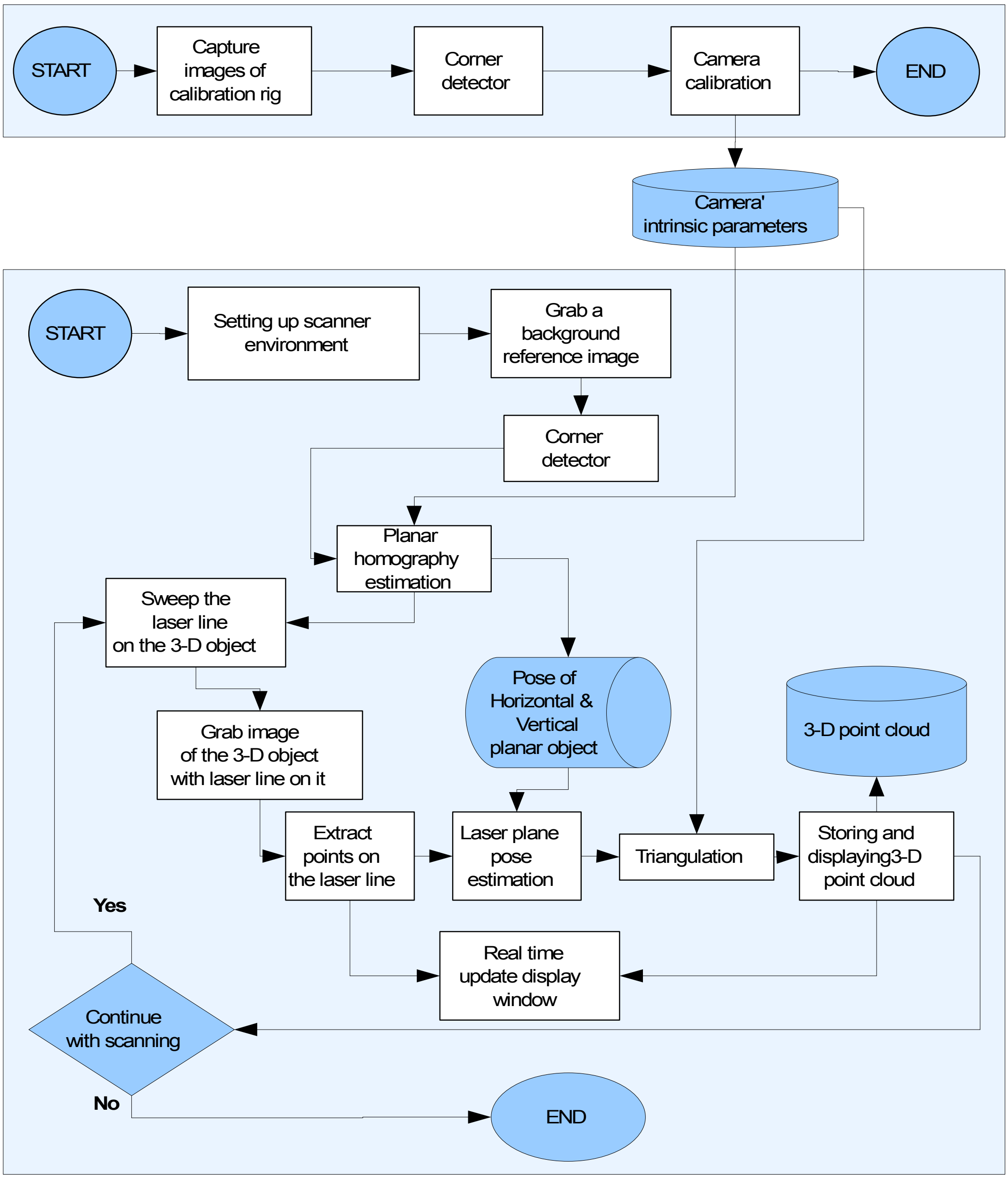


Figure 3.2: The system block diagram of freehand 3-D scanner.

Making use of the camera intrinsic parameters, obtained during the camera calibration process, and the corner features correspondences, provided by the corner detector, the pose of each planar object was estimated using the planar homography estimation process. The pose of each planar objects, the relative rigid transformation between the global reference coordinate frame and the camera coordinate frame, was estimated using planar homography.

Once the pose of each of the two planar objects was estimated, their relative pose between the camera and the planar object was not disturbed.

Subsequently the 3-D scanning operation was performed by sweeping a laser line across the object to be scanned. As the laser line was being swept, images of the laser line were captured and laser points in the camera image were extracted. The laser points on the planar objects were used to determine the pose of the laser plane, with respect to the global reference coordinate frame, during the laser plane pose estimation process. The depths of the laser points on the scanned object, with respect to the camera coordinate frame, were resolved using triangulation process. A cloud of 3-D points were produced at the end of the scanning process. This process is further explained below.

In the scene image, a region of interest was defined to enclose the object to be scanned. Using the laser line emitter, a laser line was swept across the surface of the object. To use the method of triangulation, the angle between the viewing axis of camera and laser plane should be as large as possible. The images (Img_L) of the laser line sweeping across the object were captured. The points of the laser line seen in the image were generated by the laser plane intersecting the horizontal and the vertical planes of the planar object and the surface of the object being scanned. These points were detected and their 2-D co-ordinates were estimated in each of the images. Knowing the pose of the planes, of the planar object, the laser points laying on the two planes were transformed to their corresponding 3-D coordinate points, with respect to the camera coordinate system. By using this estimated 3-D coordinates, of the points on the laser plane, the equation representing the laser plane in each of the images was estimated. Knowing the equation of the laser plane, in each of the images, the 3-D coordinates of all the laser points on the surface of the object were estimated for each of the images. These points represented the points on the surface of the object, that was being scanned.

A cloud of 3-D points representing the surface of the object being scanned was thus created at the end of the scanning process. PowerCrust [37], an open source algorithm for 3-D surface reconstruction, was applied to the cloud of 3-D points to create a set of polygon mesh and thus to approximate the surface of the reconstructed object. MeshLab [34], an interactive open source 3-D viewing and mesh editing program, was then used to display and smooth the generated set of polygonal mesh.

Both the extracted points on the laser line, in the 2-D image coordinate frame and the 3-

D global reference frame, were rendered and displayed as intermediate result, in real time. The real time display is very helpful as it provides to the user an indication, of the part of the object being scanned and the resulting 3-D cloud points. This helps the user to identify the part of the object which needs to be swept further to either acquire new points or to improve the quality of the acquired points.

In the following sections, the techniques used to implement the above discussed steps are provided in detail under the section headings Camera Calibration and 3-D points cloud construction.

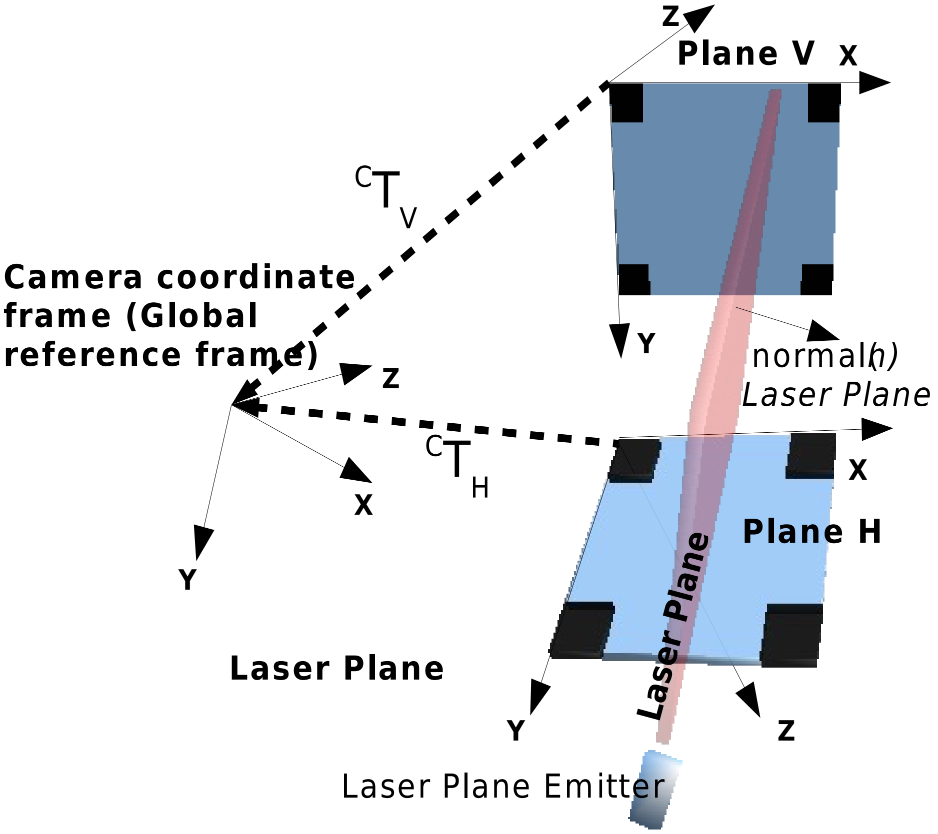


Figure 3.3: System coordinate frames and notation.

3.3 Camera calibration

A basic webcam consists of a lens, an array of photoelectric light sensor and an on board embedded system. The lens directs the ray of lights, from the scene in front of the camera, onto the surface of the array of photoelectric light sensor. The sensor digitises the ray of lights and stores the result onto the internal memory of the on board embedded system. A camera image in the digital format is created according to the specific data format used by the system. Depending on the data interface provided by the embedded system, the created image can be stored on board and transferred to an external computer through a standard interface, *i.e.* USB, Firewire 1394, and *etc*.

The created digital camera image is the output of the digitisation process. This digitisation output depends on the physical property of the system, *e.g.* the dimension and symmetry of each of the photoelectric sensors in the array and the distance of the lens with respect to the array of photoelectric sensors, *i.e.* the camera focal length.

The camera system is modelled as a camera pin-hole projection model [24]. The created digital camera image can be modelled as being dependent on the location of the sensor array with respect to the centre of the projection of ray of lights. The process of camera calibration resolves the relationship of the camera's physical details with the formed digital image.

A camera can be modelled as a pin-hole projection model as shown in equation (3.1).

$$\lambda \underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{:=m'} = \underbrace{\underbrace{\begin{bmatrix} f_{sx} & f_{s\theta} & O_x \\ 0 & f_{sy} & O_y \\ 0 & 0 & 1 \end{bmatrix}}_{:=Intrinsic_{3x3}} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \underbrace{\begin{bmatrix} R & \vec{T} \\ 0 & 1 \end{bmatrix}}_{:={}^{Camera}\Gamma_{Object}}}_{:=P} \underbrace{\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}}_{:=M} \tag{3.1}$$

$$\lambda m' = PM$$

The pixel dimension is expressed as fs_x and fs_y, the skew factor by fs_θ, and the optical centre coordinates by O_x, and O_y. These characterise the camera intrinsic parameters ($Intrinsic_{3x3}$). The relative rigid transformation is expressed by a four by four matrix with three by three rotational matrix R and translation 3-vector $\vec{T}$. A 3-D homogeneous coordinate point $M[X,Y,Z,1]^T$ is projected by the pin-hole model onto the image with coordinate $m'[\lambda u, \lambda v, \lambda]^T$.

The objective of the camera calibration process is to recover the value of camera intrinsic parameters, $Intrinsic_{3x3}$. The relative rigid transformation, ${}^{Camera}\Gamma_{Object}$, is also resolved at the end of the camera calibration process.

In the camera pinhole model, a 3-D point in the real world is mapped to a 2-D point on a camera image by using a three by four projection matrix P as shown by equation (3.2).

$$P=\begin{bmatrix} f_{sx} & f_{s\theta} & O_x \\ 0 & f_{sy} & O_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} R & \vec{T} \\ & \\ 0 & 1 \end{bmatrix} \tag{3.2}$$

$${}^{Camera}\Gamma_{Object}=\begin{bmatrix} R & \vec{T} \\ & \\ 0 & 1 \end{bmatrix}=\left\{\begin{pmatrix}\vec{r}_{c1} \\ \\ 0\end{pmatrix}\begin{pmatrix}\vec{r}_{c2} \\ \\ 0\end{pmatrix}\begin{pmatrix}\vec{r}_{c1}\times\vec{r}_{c2} \\ \\ 0\end{pmatrix}\begin{pmatrix}\vec{T} \\ \\ 1\end{pmatrix}\right\}_{4x4} \tag{3.3}$$

Where, $\vec{r}_{c1}$ and $\vec{r}_{c2}$ are the first and second columns of the rotation matrix R , respectively. $\vec{r}_{c1}\times\vec{r}_{c2}$ is the cross product between the two vector.

Using the relative rigid transformation matrix, referred as ${}^{Camera}\Gamma_{Object}$ in Equation (3.3), a point in the real world, expressed in the object coordinate frame, with homogeneous coordinate $[X,Y,Z,1]^T$ can be transformed into the camera coordinate frame and subsequently projected as an image coordinate point $[u,v,1]^T$. The image coordinate will have a scale factor λ , depending on the location of the image frame from the origin of the camera along the z-axis of the camera coordinate frame as shown in Figure 3.4.

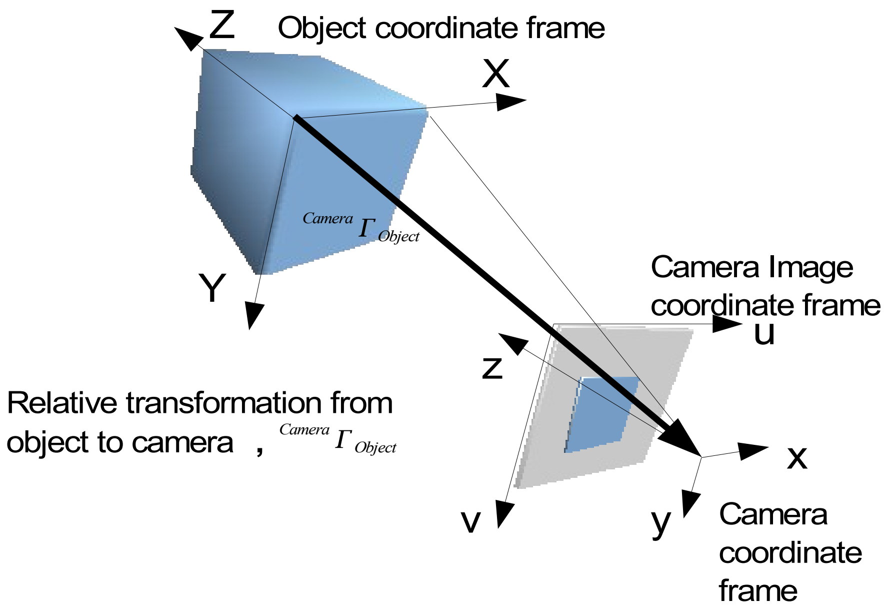


Figure 3.4: Coordinate frames and transformation in details

3.3.1 Camera calibration process

Camera calibration is carried out based on a homogeneous relationship, *i.e.* one to one correspondence between a world 3-D point and it's projected 2-D camera image location. To calibrate the camera, one requires a suitable calibration apparatus to make the correspondence from the world 3-D point to it's projected 2-D camera image location.

There are several types of calibration apparatus discussed in the literature. They can be classified into four categories [38], namely 3-D reference object, 2-D plane, 1-D line and 0-D self calibration.

3.3.1.1 Three-dimensional (3-D) calibration object

For the 3-D object based calibration apparatus, a 3-D object with precisely known dimensions or markers at precise locations, on the 3-D object surface, are used to establish the correspondence.

The cost of building a precise 3-D calibration object, *e.g.* a cube with some precise markers on each side of the cube surface is quite expensive. Instead, a set-up with precise known motion of a plane in the 3-D world coordinate, can provide the 3-D reference points. These points can be used to establish the correspondence.

The sets of 3-D coordinate points with their corresponding detected 2-D image coordinates are subsequently used to calibrate the camera [38].

3.3.1.2 Two-dimensional (2-D) planar object

Camera calibration with a 2-D planar object can be carried out by using planar homography. Planar homography or plane collineation [39] expresses the one to one relationship, *i.e.* the non-singular linear transformation, of a projection of a point in 3-D on a planar object to its corresponding 2-D camera image coordinate. This process can be interpreted as a transformation of coordinates from one projective system (coordinate frame on a 3-D planar surface) to another projective system (coordinate frame on an image), with reference to a fixed coordinate frame of reference (camera coordinate frame).

A 2-D planar object can be easily constructed by marking salient features, like points or squares, with known dimensions onto a planar object. A typical example of a ready to use 2-D planar object is a checker board. A checker board pattern can easily be created by printing the pattern and attaching it to a rigid planar surface.

3.3.1.3 One dimensional (1-D) line

Using a freely moving 1-D object it is, in principle, impossible to calibrate a camera [38]. Zhang has proposed a method to calibrate a camera, using a 1-D object, by rotating, for example a rigid stick or a taut string, with 3 or more markers on it, around a fixed point.

An advantage of using the 1-D calibration object is when the line object is swinging around, the markers will be easily visible to all the cameras in a multi-camera environment. A calibration algorithm is proposed by Wang [40] to calibrate all the cameras in a multi-camera environment using a 1-D calibration apparatus.

3.3.1.4 Self-calibration

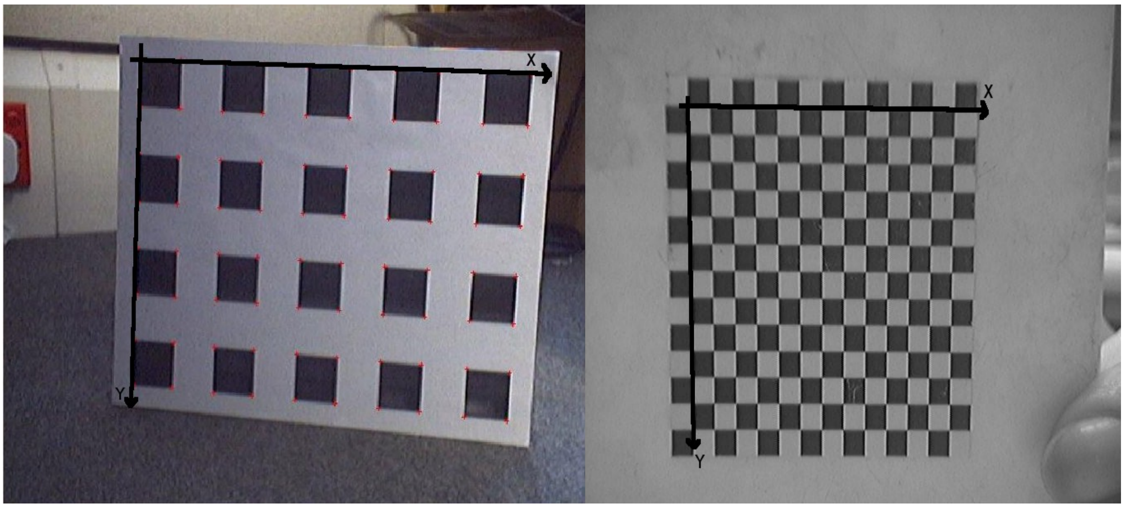
Self-calibration, calibrating camera without any calibration apparatus, was achieved by making use of the correspondences in a sequence of images of a static scene. Pollefeys [41] used the principle of absolute quadric [42] to perform camera self-calibration in the presence of varying internal camera parameters. Self-calibration with the presence of varying internal camera parameters in the sequence of images was achieved by Mendoca [43] by extending the Hartley's self-calibration technique[44].

In this study, we have used the 2-D planar object to perform the camera calibration. Camera calibration using a 2-D planar object is discussed in detail in the next section.

3.3.2 Camera calibration using a 2-D calibration rig

A 2-D camera calibration rig with a planar object is easier to fabricate as compared with a 3-D rig. Moreover, it provides a better accuracy as compared with the 1-D rig and the self calibration methods [38]. For the above reasons a 2-D calibration rig was considered best suited to our work.

Zhang proposed a flexible method to calibrate camera with a few images of a planar object captured in different poses [45]. Each planar homography induced by the planar object on each camera image was extracted and used to resolve the camera intrinsic parameters. The planar object used in our study is a checker board shown in Figure 3.5. The process of planar homography estimation and its subsequent usage to estimate the camera intrinsic parameters are explained in the following sections.



(a) *(b)*

Figure 3.5: The planar objects used in the study.(a) L-corner checker board pattern. (b) X-corner pattern.

3.3.2.1 Estimation of planar homography

It has been shown in reference [39] that the projection of a planar 3-D point with homogeneous coordinates $[X, Y, 0,1]^T$ onto an image plane with homogeneous coordinate $[u, v, 1]^T$ can be modelled by a planar homography ($Homography_{3x3}$) as shown by equation (3.4).

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}}_{:= Homography_{3x3}} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{3.4}$$

By using equation (3.5), the planar homography (eight unknowns, with h_9 normalised to one) can be estimated using four correspondences (each non-collinear correspondence gives two independent equations). The eigenvector corresponding to the smallest eigenvalue of matrix **M** provides the solution of the linear least square system.

$$\underbrace{\begin{pmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -u'_1X_1 & -u'_1Y_1 & -u'_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -v'_1X_1 & -v'_1Y_1 & -v'_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -u'_2X_2 & -u'_2Y_2 & -u'_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -v'_2X_2 & -v'_2Y_2 & -v'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & 1 & 0 & 0 & 0 & -u'_NX_N & -u'_NY_N & -u'_N \\ 0 & 0 & 0 & X_N & Y_N & 1 & -v'_NX_N & -v'_NY_N & -v'_N \end{pmatrix}}_{:=M} \underbrace{\begin{pmatrix} h_{11} \\ h_{12} \\ \vdots \\ h_{31} \\ h_{32} \\ h_{33} \end{pmatrix}}_{:=\vec{h}} = \underbrace{\begin{pmatrix} \epsilon'_{11} \\ \epsilon'_{12} \\ \epsilon'_{21} \\ \epsilon'_{22} \\ \vdots \\ \epsilon'_{N1} \\ \epsilon'_{N2} \end{pmatrix}}_{:=\vec{E}} \qquad (3.5)$$

In each of the images, the first approximated locations of the four outer most corners of the checker board were marked by the user. Using the approximated locations of the corners in the four images, and knowing the actual dimensions of the checker board pattern, the programme estimates the first approximation of a 3×3 homogeneous matrix. The accuracy of the estimated homogeneous matrix could be improved by using more corners on the checker board image. An error-minimisation algorithm that can handle more than four corners had been implemented and integrated into MIMAS [4]. The algorithm applies singular value decomposition (SVD) and provides the first approximation of a homography matrix.

Using the first approximation of the homography matrix, the corners of the checker board were re-estimated. Then, a corner detection algorithm was used to locate the actual corners of the checker board. The most recent fine tuned estimated position of the four corners formed a homography for each set of the images.

3.3.2.2 Zhang' camera calibration algorithm

The process of using homographies to resolve the camera intrinsic parameters has been explained in detail in the paper by Zhang [45]. The process in briefly described below.

Each set of the estimated homography provides two constraints (or conditions), as illustrated by equations (3.10) and (3.11). From equation (3.6), it can realised that the planar homography is the composition of camera intrinsic parameters and a modified notation of the rigid transformation matrix ${}^{Camera}\Gamma_{Object}$ [45].

Vectors r_{c1} and r_{c2} are two orthonormal vectors from a rotational matrix hence their dot product is equal to zero (see equation3.10). Moreover, the vectors $\vec{r_{c1}}$ and $\vec{r_{c2}}$ are unit vectors, hence their magnitudes are equal to one. (see equation (3.11)). Zhang [45] demonstrated a flexible method to calibrate a pin-hole model camera by making use of the above two constraints (see equations (3.10) and (3.11)) as shown in equations (3.15).

Let $Homography_{3x3}=[h_{c1} h_{c2} h_{c3}]$, where h_{ci} is i^{th} column of matrix $Homography_{3x3}$ *e.g.* $h_{c1}=[h_1 h_4 h_7]^T$ and $h_{c2}=[h_2 h_5 h_8]^T$.

We know,

$$Homography_{3x3}=\lambda Intrinsic_{3x3} AbusedExtrinsic_{3x3} \tag{3.6}$$

with

$$AbusedExtrinsic_{3x3}=\{a_{c1} \quad a_{c2} \quad a_{c3}\}=\{r_{c1} \quad r_{c2} \quad t_{3x1}\} \tag{3.7}$$

and

$$B=\underbrace{\begin{bmatrix} f_{sx} & f_{s\theta} & O_x \\ 0 & f_{sy} & O_y \\ 0 & 0 & 1 \end{bmatrix}}_{Intrinsic_{3x3}} \tag{3.8}$$

Hence from equation (3.6)

$$[h_{c1} h_{c2} h_{c3}]=\lambda B[r_{c1} r_{c2} t_{3x1}] \tag{3.9}$$

With the first constraint $r_{c1}\cdot r_{c2}=0$, hence $h_{c1}^T B^{-T} B^{-1} h_{c2}=0$ (3.10)

With the second constraint $\|r_{c1}\|=\|r_{c2}\|=1$, hence

$$h_{c1}^T B^{-T} B^{-1} h_{c1} = h_{c2}^T B^{-T} B^{-1} h_{c2} \tag{3.11}$$

Let

$$S = B^{-T} B^{-1} = \underbrace{\begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{21} & S_{22} & S_{23} \\ S_{31} & S_{32} & S_{33} \end{bmatrix} = \begin{bmatrix} S_{11} & S_{12} & S_{13} \\ S_{12} & S_{22} & S_{23} \\ S_{13} & S_{23} & S_{33} \end{bmatrix}}_{\text{symmetric matrix}} \tag{3.12}$$

Hence

$$\underbrace{\left|\begin{matrix} h_1^{(1)} h_2^{(1)} & h_1^{(1)} h_5^{(1)} + h_2^{(1)} h_4^{(1)} & h_4^{(1)} h_5^{(1)} & h_2^{(1)} h_7^{(1)} + h_1^{(1)} h_8^{(1)} & h_5^{(1)} h_7^{(1)} + h_4^{(1)} h_8^{(1)} & h_7^{(1)} h_8^{(1)} \\ h_1^{(1)} h_1^{(1)} - h_2^{(1)} h_2^{(1)} & 2(h_1^{(1)} h_4^{(1)} - h_2^{(1)} h_5^{(1)}) & h_4^{(1)} h_4^{(1)} - h_5^{(1)} h_5^{(1)} & 2(h_1^{(1)} h_7^{(1)} - h_2^{(1)} h_8^{(1)}) & 2(h_4^{(1)} h_7^{(1)} - h_5^{(1)} h_8^{(1)}) & h_7^{(1)} h_7^{(1)} - h_8^{(1)} h_8^{(1)} \\ h_1^{(2)} h_2^{(2)} & h_1^{(2)} h_5^{(2)} + h_2^{(2)} h_4^{(2)} & h_4^{(2)} h_5^{(2)} & h_2^{(2)} h_7^{(2)} + h_1^{(2)} h_8^{(2)} & h_5^{(2)} h_7^{(2)} + h_4^{(2)} h_8^{(2)} & h_7^{(2)} h_8^{(2)} \\ h_1^{(2)} h_1^{(2)} - h_2^{(2)} h_2^{(2)} & 2(h_1^{(2)} h_4^{(2)} - h_2^{(2)} h_5^{(2)}) & h_4^{(2)} h_4^{(2)} - h_5^{(2)} h_5^{(2)} & 2(h_1^{(2)} h_7^{(2)} - h_2^{(2)} h_8^{(2)}) & 2(h_4^{(2)} h_7^{(2)} - h_5^{(2)} h_8^{(2)}) & h_7^{(2)} h_7^{(2)} - h_8^{(2)} h_8^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ h_1^{(N)} h_2^{(N)} & h_1^{(N)} h_5^{(N)} + h_2^{(N)} h_4^{(N)} & h_4^{(N)} h_5^{(N)} & h_2^{(N)} h_7^{(N)} + h_1^{(N)} h_8^{(N)} & h_5^{(N)} h_7^{(N)} + h_4^{(N)} h_8^{(N)} & h_7^{(N)} h_8^{(N)} \\ h_1^{(N)} h_1^{(N)} - h_2^{(N)} h_2^{(N)} & 2(h_1^{(N)} h_4^{(N)} - h_2^{(N)} h_5^{(N)}) & h_4^{(N)} h_4^{(N)} - h_5^{(N)} h_5^{(N)} & 2(h_1^{(N)} h_7^{(N)} - h_2^{(N)} h_8^{(N)}) & 2(h_4^{(N)} h_7^{(N)} - h_5^{(N)} h_8^{(N)}) & h_7^{(N)} h_7^{(N)} - h_8^{(N)} h_8^{(N)} \end{matrix}\right|}_{:=SYS} \underbrace{\begin{pmatrix} S_{11} \\ S_{12} \\ S_{22} \\ S_{13} \\ S_{23} \\ S_{33} \end{pmatrix}}_{:=\vec{s}} = \underbrace{\begin{pmatrix} \epsilon'_{11} \\ \epsilon'_{12} \\ \epsilon'_{21} \\ \epsilon'_{22} \\ \vdots \\ \epsilon'_{N1} \\ \epsilon'_{N2} \end{pmatrix}}_{:=\vec{E}}$$

N , the number of homographies. (3.13)

The eigenvector corresponds to the least eigenvalue of SYS and is the solution for the linear equations. Once the value of S is obtained by the operation of Singular Value Decomposition, the inverse of the camera intrinsic, B^{-1} , can be recovered by using the Cholesky decomposition, as shown in equation (3.14). The camera intrinsic parameters, B , is subsequently resolved by inverting the result of equation (3.14), as shown in equation (3.15).

$$B^{-1} = chol(S) \tag{3.14}$$

Where $chol(S)$ is a function to perform the Cholesky decomposition of S .

and

$$Intrinsic_{3x3} = B = inv(B^{-1}) \tag{3.15}$$

Where $inv(B^{-1})$ is a function returning the inverse of B^{-1}

To resolve the five unknown camera intrinsic parameters (fs_x, fs_y, fs_θ, O_x, and O_y), at least three sets of homographies ($N=3$, equation (3.13)) are needed. If we want to take into account the radial distortion of the lens, it introduces another two extra unknown variables. Hence, at least seven constraints are to be provided to the camera calibration algorithm. This requirement is satisfied by using four sets of homographies estimated from the checker board images at four different poses.

Starting with the very first estimate of the camera intrinsic parameters, the final values are estimated using Levenberg-Marquardt optimisation process as suggested by Zhang [45]. The cost function used by the Levenberg-Marquardt optimisation process is the difference between the locations of the re-projected corners, obtained by making use of the calculated camera intrinsic parameters, and the actual camera image corners.

The above described camera calibration method is implemented by Bouguet [46] and also available in the MIMAS toolkit, a C++ real-time computer vision library [6].

3.4 Three-dimensional (3-D) points cloud creation

As shown in the system block diagram in Figure 3.2, the 3-D points cloud creation process consists of the three major operations namely, planar object pose estimation, laser plane pose estimation and 3-D points cloud estimation. The requirements of the above listed three operations are discussed briefly below.

The planar object pose estimation provides us the position and orientation of a stable reference plane with respect to the fixed camera coordinate frame. Making use of these reference planes one can subsequently estimate the position and orientation of the laser plane by detecting the points on the laser line. This is performed by taking into account the fact that the laser line is formed by the intersection of the laser plane with the planar object plane. Once the laser plane pose is estimated, the location of the 3-D points, illuminated by the laser line, on the surface of the object being scanned can be approximated. The points on the surface of the 3-D object are illuminated by the laser line through the process of sweeping the laser line on the surface of the object being scanned.

The details of the three operations are discussed briefly in the following sections.

3.4.1 Planar object pose estimation

Once the camera intrinsic parameters have been determined, the pose of the planar object, *i.e.* the rigid transformation matrix ${}^{Camera}\Gamma_{Object}$ given in equation (3.3) can be resolved by using equation (3.17).

From the equations (3.6) and (3.7), we get

$$AbusedExtrinsic_{3x3} = Intrinsic_{3x3}^{-1} \cdot Homography_{3x3} \tag{3.16}$$

Thus, the rigid transformation matrix, given by equation (3.3) can be resolved by

$$\begin{gathered} r_{c1} = \lambda_1 \cdot Intrinsic_{3x3}^{-1} \cdot a_{c1} \\ r_{c2} = \lambda_2 \cdot Intrinsic_{3x3}^{-1} \cdot a_{c2} \\ r_{c3} = r_{c1} \times r_{c2} \\ t_{3x1} = \lambda_3 \cdot Intrinsic_{3x3}^{-1} \cdot a_{c3} \\ \lambda_1 = 1/\|Intrinsic_{3x3}^{-1} \cdot a_{c1}\| \\ \lambda_2 = 1/\|Intrinsic_{3x3}^{-1} \cdot a_{c2}\| \\ \lambda_3 = (\lambda_1 + \lambda_2) * 0.5 \end{gathered} \tag{3.17}$$

Where $r_{c1} \times r_{c2}$ is the cross product between r_{c1} and r_{c2} .

3.4.2 Laser plane pose estimation

To estimate the 3-D location of the points on the surface of a 3-D object, we needed to perform the following operations. First the pose of the laser plane is needed to be estimated, in order to find the relative position and orientation of the laser plane with respect to the calibrated camera reference coordinate frame. By knowing the relative position and orientation of the laser plane, any image pixel illuminated by the laser plane can be transformed from the image coordinate point to a 3-D coordinate point with respect to the calibrated camera reference coordinate frame.

Normally a 3-D reconstruction process starts with adjusting the position and orientation of the calibrated camera to make sure the object to be digitised is within the viewpoint. Once the camera viewpoint is decided, the laser plane will be projected to the scene and some fine adjustments between the camera and laser plane will be carried out to get the best image. The laser plane calibration process had to be carried out before the 3-D reconstruction process (turntable based, linear translator based 3-D surface reconstruction systems) or during the scanning (freehand 3-D surface reconstruction system). The laser plane calibration process is only needed to be carried out once, if both the camera and the laser emitter are not going to be moved after this calibration process, *e.g.* turntable based 3-D surface reconstruction system [25,26].

The laser plane pose estimation involved the detection and extraction of the points on the image illuminated by the laser and fitting a line to the detected set of points. Using the fitted line, the laser plane pose was estimated.

3.4.2.1 Laser pixel extraction process

The intersection of a laser plane with an object produces a set of illuminated pixels in the camera image which is referred as a laser line in the following discussion. The laser line can be defined by using two points on the line. But at least three non-collinear points are needed to define the plane in the 3-D space. Hence, two sets of non-collinear laser lines are needed to be generated in the camera image. To achieve this, the laser plane is made to intersect two non collinear planes.

Any planar object could be used for the above discussed operation for acquiring two non collinear lines in a single image. In this study, a folded checker board (see Figure 3.1), which had been previously used for camera calibration, was used for this purpose.

The illuminated laser points are needed to be detected to estimate the pose of the laser plane. The 3-D reconstruction process is normally accomplished in different environments, for example in a dark room, where the ambient light is under control, or in an environment where the ambient light is affected by the flickering of fluorescent light (if it is indoors), or by variation in sun light (if it is outdoor). The intensity of the image pixel illuminated by the laser vary depending upon the environment. Hence, suitable robust methods are needed to detect the pixels on the laser line for the laser plane pose estimation process.

Depending on the environment, one can use either of the two methods, namely Thresholding or Background subtraction, to estimate the pixel locations illuminated by the laser.

3.4.2.1.1 Estimation of pixel locations, illuminated by the laser, using thresholding

The algorithm to detect the laser line on a camera image in a controlled environment is easier than uncontrolled environment. Because the knowledge about the properties of the pixels illuminated by the laser in the image, *e.g.* the RGB (Red, Green Blue colour space) or HSV (Hue, Saturation, Value colour space) value of the laser illuminated pixels can be known beforehand. This eliminates the requirement of the user to provide the range of values for the thresholding process.

Also, in a controlled environment the ambient light can be adjusted (*e.g.*, its intensity reduced) such that an image with high quality information (pixel with maximum intensity) of the pixels illuminated by the laser can be produced without using the maximum power of the laser generator.

The known HSV range of values, which adequately represent the pixels illuminated by the laser better than RGB colour space, were used in the thresholding process to filter the image.

Figure 3.6 shows the result of the thresholding process by using a predefined thresholding RGB colour space value. Figure 3.7 shows the plot of the HSV colour space information in one of the row of the same image. The result of the thresholding by using the observed HSV colour space value for the pixels illuminated by the laser is shown in Figure 3.8. The result of the thresholding provides the information regarding the position of the pixel locations illuminated by the laser in the image. Depending on the threshold values used, the width of the line, formed by the extracted pixels, was approximately two to five pixels.

The extracted pixel locations illuminated by the laser provide the spatial information to form the laser line which is basically the intersection of the laser plane and the planar object namely the checker board. The following section provides the details of the process of fitting a line to the extracted pixel locations.

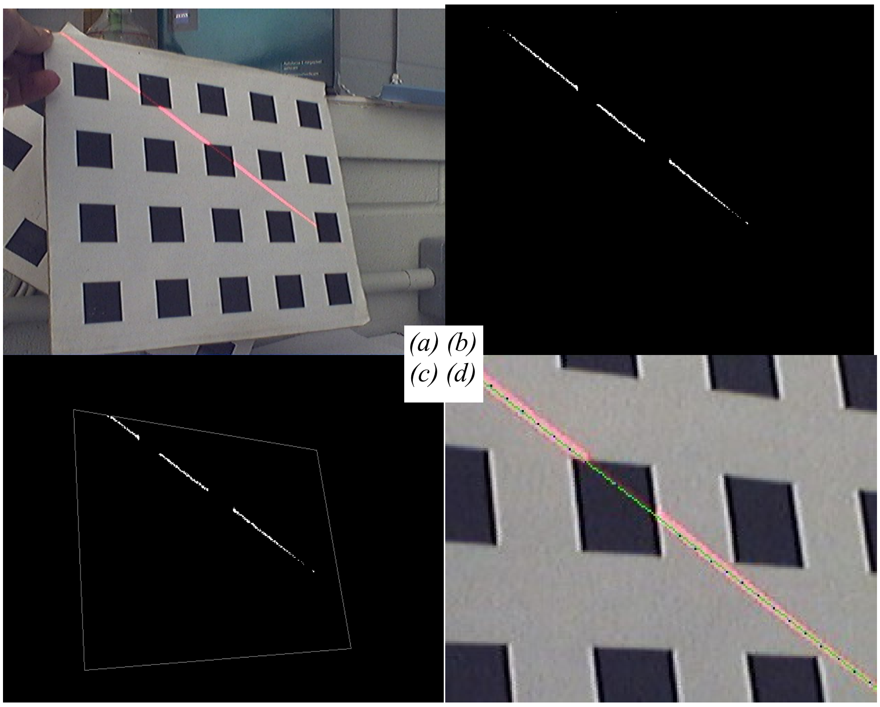


Figure 3.6: Images showing laser line detection using RGB.

(a) The laser plane intersected by the checker board.

(b) The detected points on the laser line by thresholding the image using RGB channel information(any pixel with red channel value =240+/-10, the maximum value is 255).

(c) Image showing only the detected points on the laser line which are inside the checker board frame.

(d) The fitted line (green colour,with polar space, theta =61 and radius=85) and some of the original thresholded points marked on the fitted line (little dot in blue colour)

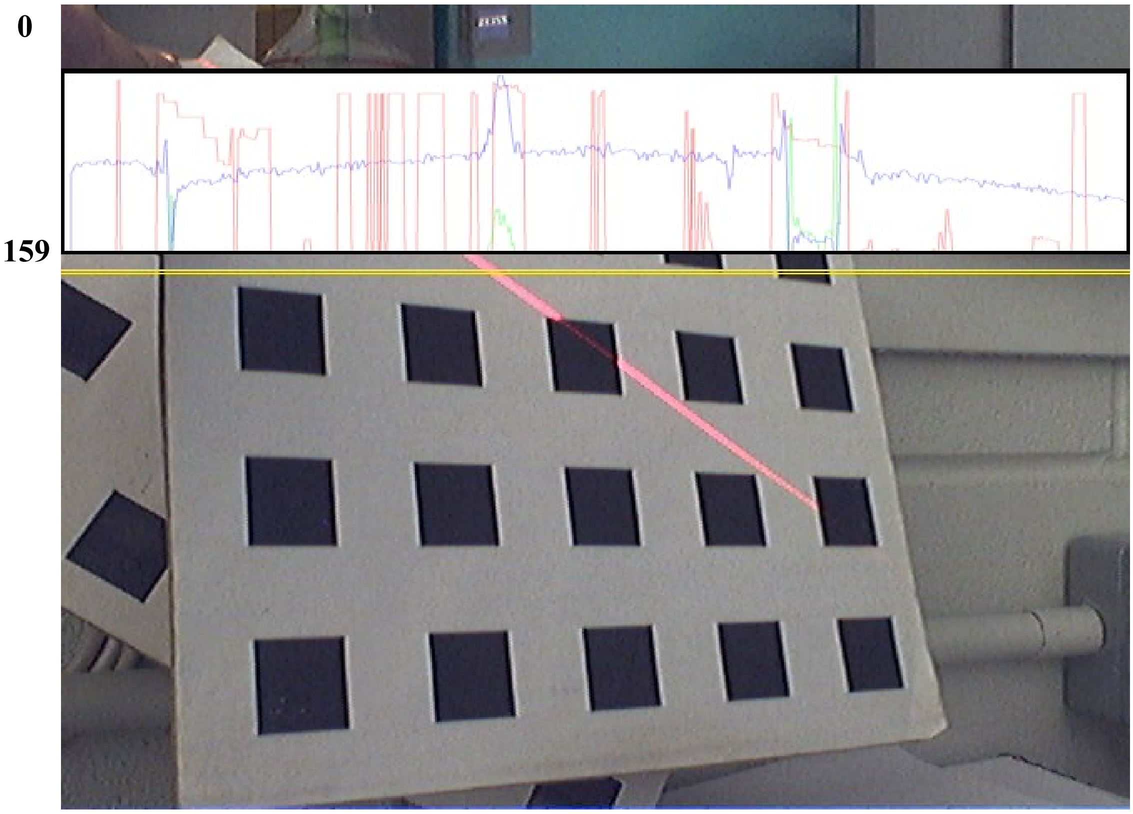
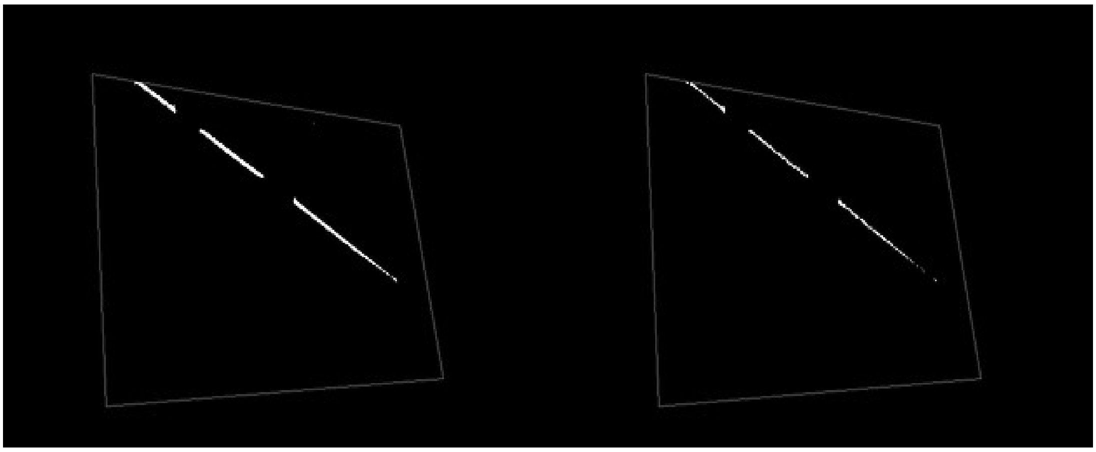


Figure 3.7: Image showing the pixels' HSV values plot along the row number 159 (the row between two yellow colour lines). The Blue colour plot, which indicates the V channel of the HSV, peaks (100%, minimum value 0%, maximum value 100%) at the location of the pixels illuminated by the laser.

The Red colour plot is for the Hue channel (minimum value 0 degree, maximum value 359 degree) and the Green colour plot is for the saturation channel (minimum value 0%, maximum value 100%).



(a) *(b)*

Figure 3.8: Image showing the thresholded pixel locations illuminated by the laser using different threshold value.

(a) V=0.90, (b) V=0.99

The width of the extracted line formed by the pixel locations line (left), using threshold value V=0.9, is thicker (3 to 4 pixels wide) than the line (right) extracted using V=0.99 (2 to 3 pixels wide).

3.4.2.1.2 Estimation of pixel locations, illuminated by the laser, using background subtraction

If one does not have a control on the environment like lighting condition, an inappropriate threshold value will result in the false detection of the pixels illuminated by the laser. Different types of noise will be always present in any image, hence a thresholding process with hard coded values may fail to detect the the pixels illuminated by the laser. Hence instead of using specific threshold values, a better way of detecting the pixels illuminated by the laser was designed and is explained herein.

The alternative approach involved background subtraction. Briefly, the background subtraction can be carried out by first taking an initial image with the laser emitter turned off and then powering on the laser emitter and pointing it at the scene and capturing another image (Figure 3.9). The additional component in the second image, that is the pixels illuminated by the laser, can be detected by finding the difference between the two images. The main objective of using background subtraction process is to avoid the use of any thresholding. The process is discussed in detail below.

The pixels illuminated by the laser can be detected by applying a general image processing technique, the background subtraction method. Firstly an image without laser lines (background image) was captured. The background subtraction method was performed with each of the subsequent images (laser image). The difference between each image, the projected laser line, was detected. A range of grey values from 0 to 255 (8-bits) have been commonly used to represent the intensity of a pixel in an image. A brighter component has a higher intensity value, *e.g.* white component is represented by value 255 and black is 0. A pixel illuminated by the laser in an image will have a higher intensity value in the red channel and a lower intensity value in the rest of the colour channels. Therefore, the result of the subtraction, will produce a resulting RGB image, where the pixels with the highest intensity value in red channel can be safely assumed as the pixels illuminated by the laser.

The following process is used to collate all the pixels illuminated by the laser. Firstly along each row of the image the pixel which has the highest intensity value in the Red channel will be marked as a possible pixel illuminated by the laser and the image coordinate was recorded. To eliminate extraneous points, the fact that the the pixels illuminated by the laser will be closer to one another was made use of and the nearest neighbourhood method was applied and explained as follows. A window with size of 5×5 was placed around a marked possible pixel illuminated by the laser. If there were any other marked pixel locations within the window, the detected pixel was most likely to be a pixel illuminated by the laser. Those marked points which were far away from other marked points, were removed in this process. Figure 3.13 shows the output of the thresholding, of the difference image, with different relative rates. The process of thresholding, of the difference image, was performed using only the value of the red channel in the RGB image. This avoided the noise contamination which might have occurred if the other colour channels were used.

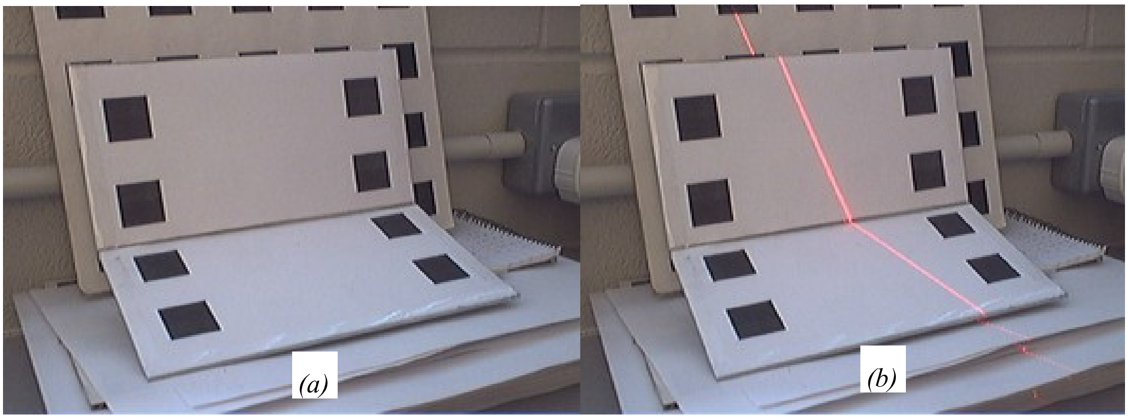
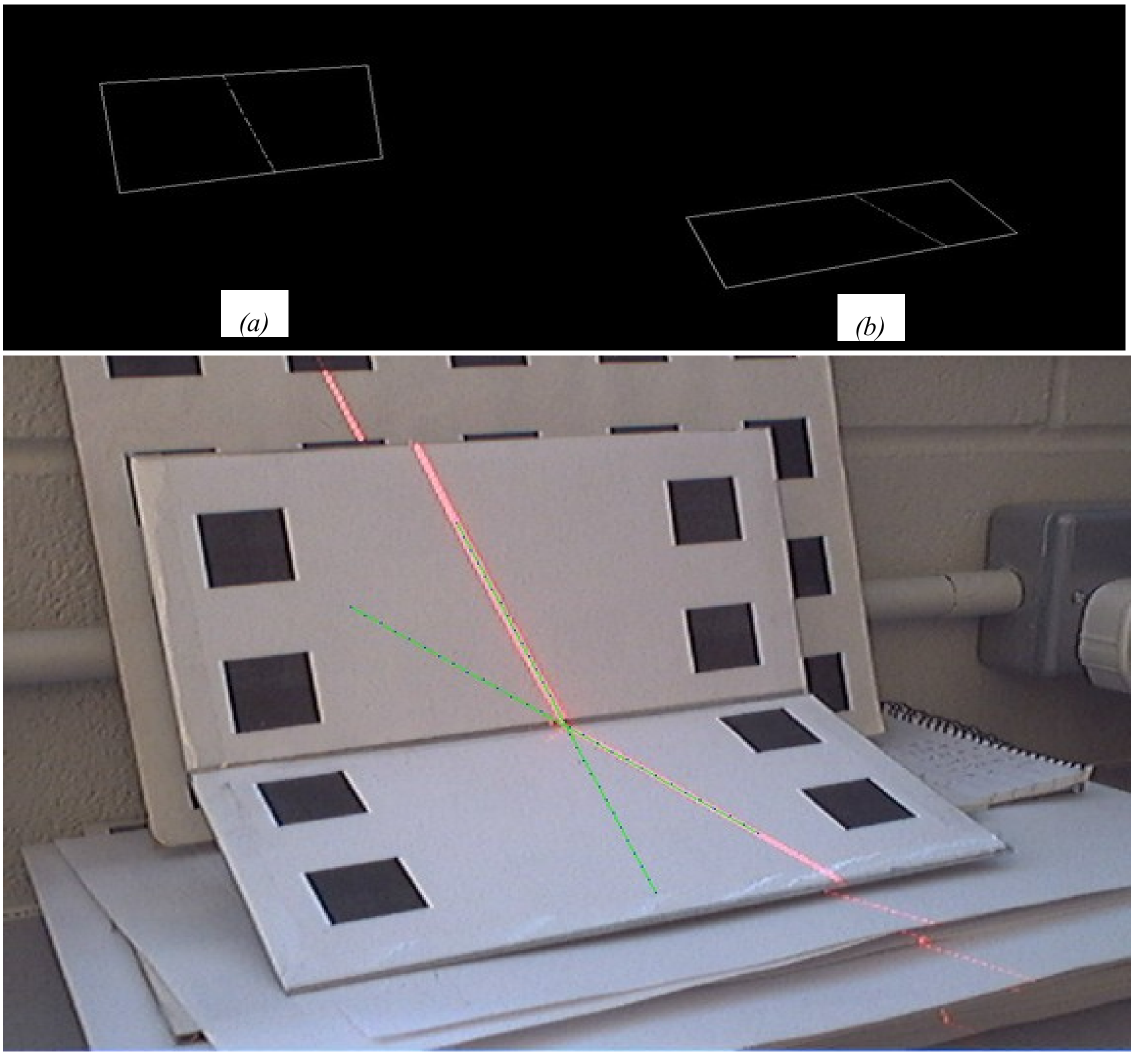


Figure 3.9: Two images for background subtraction method. The background image(a), The image with laser(b)



(a)

(b)

(c)

Figure 3.10 Result of Hough Transform line detection by using the output of background subtraction method, the marked blue colour laser pixel (c) within the defined checker board region(a) and (b) The two detected laser line are plotted in green colour (c).

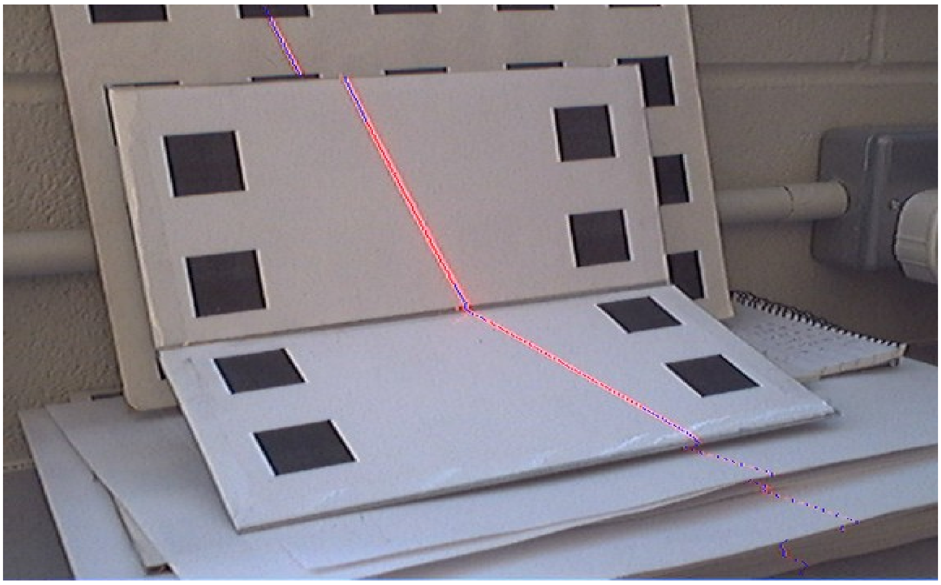


Figure 3.11: Resulting image of laser pixel detection. The points in blue colour are the pixels with highest intensity across each row of the image. The dots marked with bright red colour are the detected laser pixels inside the defined checker board boundary (Figure 3.10 a,b).

3.4.2.2 Laser line fitting process

To estimate the location of the laser line and hence the pose of the laser plane, a 2-D line was fitted through the earlier estimated image pixel locations, illuminated by the laser. In this study, the Hough transform [47] 2-D line fitting process was used to estimate the equation of the line on the image from the detected set of points illuminated by the laser. The advantage of this method is its spatial property and robustness against false pixel location detection. The false detection of pixel location illuminated by the laser could happen due to reasons like the threshold value giving extraneous locations due to the specular reflection of the laser or due to the misbehaviour of the camera CCD.

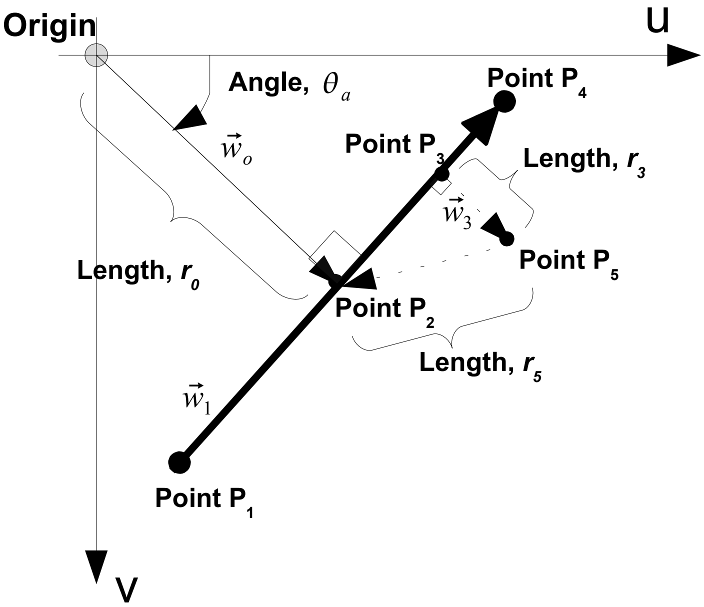


Figure 3.12: The polar parameters of a line passing through the two points P_1 and P_4.

As mentioned above, the model that needed to be fitted, to the set of pixel locations, is the function representing a straight line. The equation of the straight line is expressed in polar space. One needs at least two points, in 2D space, to define a unique line. A line $\vec{w}_1$, passing through these two points (say P_1 to P_4), can be defined by the two polar parameters, namely the orthogonal distance r_o, of the line to the origin, and the angle, θ_a, between the orthogonal line and one of the Cartesian axis, u (Figure 3.12). The values of r_o and θ_a can be resolved as follows :

Let $\vec{w}_0$ be the normal to the line $\vec{w}_1$ drawn from the origin.

Let $\vec{w}_0$ intersects $\vec{w}_1$ at the location $P_2(u,v)$.

The polar co-ordinates for the intersecting point P_2, can be resolved by using a general point equation (3.20).

Then, the values of r_o and θ_a can be resolved by using equations (3.18) and (3.19) respectively.

$$r_o = \sqrt{P_2(u)^2 + P_2(v)^2} \tag{3.18}$$

$$\theta_a = \cos^{-1}\left(\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T \times \vec{w}_0\right) \tag{3.19}$$

Where

$$\vec{w}_0 = \begin{bmatrix}-\|\vec{w}_1(v)\| & \|\vec{w}_1(u)\| & 0\end{bmatrix}^T$$

$$\|\vec{w}_1\| = \begin{bmatrix}\|\vec{w}_1(u)\| \\ \|\vec{w}_1(u)\| \\ 0\end{bmatrix} = \left\| \begin{bmatrix} P_4(u) - P_1(u) \\ P_4(v) - P_1(v) \\ 0 \end{bmatrix} \right\|$$

$$P_l = P_s + \lambda \|\vec{w}_l\| \tag{3.20}$$

Where

P_l , any point on a line

P_s , a starting point of the line

$\|\vec{w}_l\|$, the unit vector of the line

λ , the length of the line from P_s to P_l .

3.4.2.2.1 Effect of threshold values used to estimate pixel locations illuminated by the laser

It is important to note that the accuracy of the laser line fitting process is completely dependent on the output of earlier thresholding process, extracting the image pixels illuminated by the laser. Each thresholded image pixel will contribute equal weight for a possible line in the image. An erroneous thresholding value will result in the false fitting of a line due to the weight of the noise pixel in the image accidentally forming a false line. Hence, a proper range of thresholding value is needed for this approach to guarantee the accuracy of the fitted line.

Figures 3.13 and 3.14 illustrate the effect of choosing different threshold values while detecting the pixel points illuminated by the laser and the effect it has in the line fitting process. In Figure 3.13, when a threshold value $0.9 < v < 1.0$ was used, for Value in the

HSV model, one can see that many false points being detected. But, because of the robustness of the Hough transform based line fitting process, the correct laser line is fitted. In Figure 3.14, because of the choice of a different threshold value (0.99<v<1.0), Value in the HSV model, the number of extraneous points has greatly reduced. The property of the fitted line is slightly different from the earlier line shown in Figure 3.13.

Table 3.1 lists the properties of lines fitted through the points detected using different threshold values.

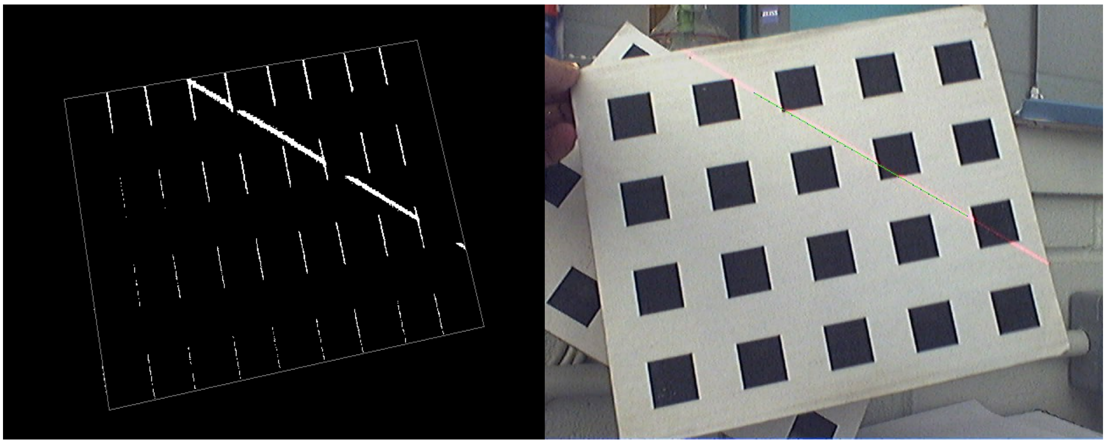


Figure 3.13: (a) The thresholded image using the threshold 0.9<v<1.0 in the HSV colour space.

(b) The fitted line (θ =61 and r =86 in the polar space) drawn in green colour.

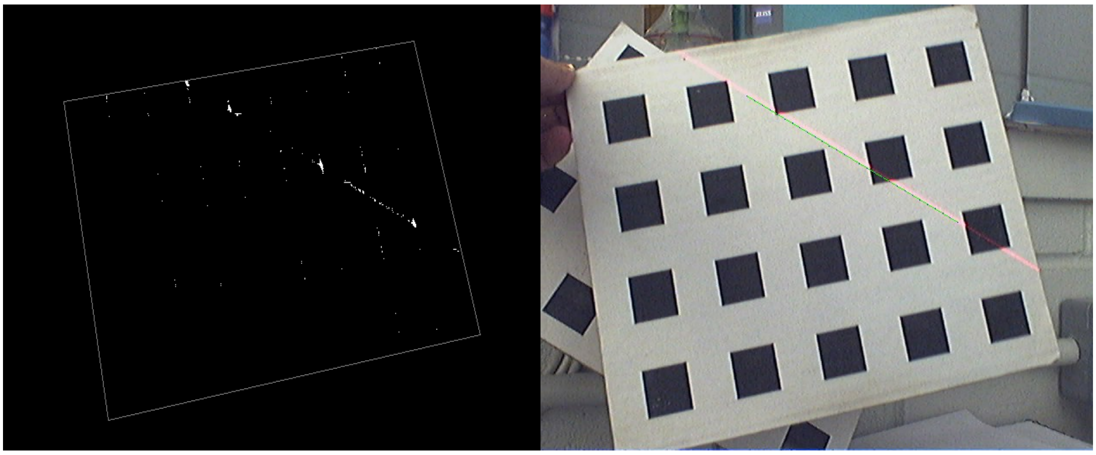


Figure 3.14 (a) The thresholded image using the threshold 0.99<v<1.0 in the HSV colour space.

(b) The fitted line (θ =61 degree and radius, r =85 in the polar space) drawn in green colour.

The width of the laser line varied from two to five pixels, depending on the thresholding value, for the Value in the HSV model, used.

Table 3.1: Result of the Hough transform line fitting process through points detected using different threshold value range.

Threshold value for the parameter Value in the HSV model	Number of line detected	Polar co-ordinates of the fitted line. Theta(θ , degree) and Radius(r)
$0.90<v<1.0$	1	$\theta=61$, $r=86$
$0.92<v<1.0$	1	$\theta=61$, $r=86$
$0.94<v<1.0$	1	$\theta=61$, $r=86$
$0.95<v<1.0$	2	$\theta=61$, $r=86$ $\theta=61$, $r=87$
$0.96<v<1.0$	1	$\theta=61$, $r=86$
$0.98<v<1.0$	1	$\theta=61$, $r=86$
$0.99<v<1.0$	1	$\theta=61$, $r=85$

3.4.2.2.2 Sub-pixel operator

Along with the value of Value in the HSV model, the intensity value of each pixel illuminated by the laser pixel can further be used to find the actual location of the laser line, in each row of the image with up to sub-pixel accuracy. It is assumed that the laser line will never be horizontal to the image. The performances of different sub-pixel operators had been evaluated by Trucco [48] and found that Blais and Rioux (1986) [49] fourth order operator gave the best performance for their application of extracting pixels illuminated by a laser, in their study of laser stripe profiler.

This section will expand on how the locations of the points illuminated by the laser were estimated with sub-pixel-accuracy. Both the previous thresholding and background subtraction processes provided a range of pixel locations, in each row of the image, illuminated by the laser. The process of identifying the centre of the laser line with sub-pixel-accuracy starts (Figure 3.15) with the reading of the intensity values of the pixels around the peak value of the identified pixel. Based on the theoretical property of the Gaussian energy dispersion, the intensity values of the laser pixel should fit into a Gaussian curve and hence the centre of the laser line lying on the peak of the Gaussian curve (Figure 3.16). The sub-pixel detector will be used to detect the peak of the signal with up to sub-pixel accuracy. The operation of the sub-pixel operator is expressed next.

The values of Value of the HSV model at row number 159 (Figure 3.7), around the pixels illuminated by the laser were extracted.

V	0.862	0.878	0.960	0.976	1.000	1.000	0.976	0.964	0.929	0.909
Pixel #	253	254	255	256	257	258	259	260	261	262

The location of the pixel with highest value was identified. Three columns of values, left and right, around the peak were selected. An index was assigned to each value and the location of the sub-pixel was obtained as discussed below.

V	0.862	0.878	0.960	0.976	1.000	1.000	0.976	0.964	0.929	0.909
Pixel #	253	254	255	256	257	258	259	260	261	262
		c_0	c_1	c_2	c_3	c_4	c_5	c_6		
					Peak					

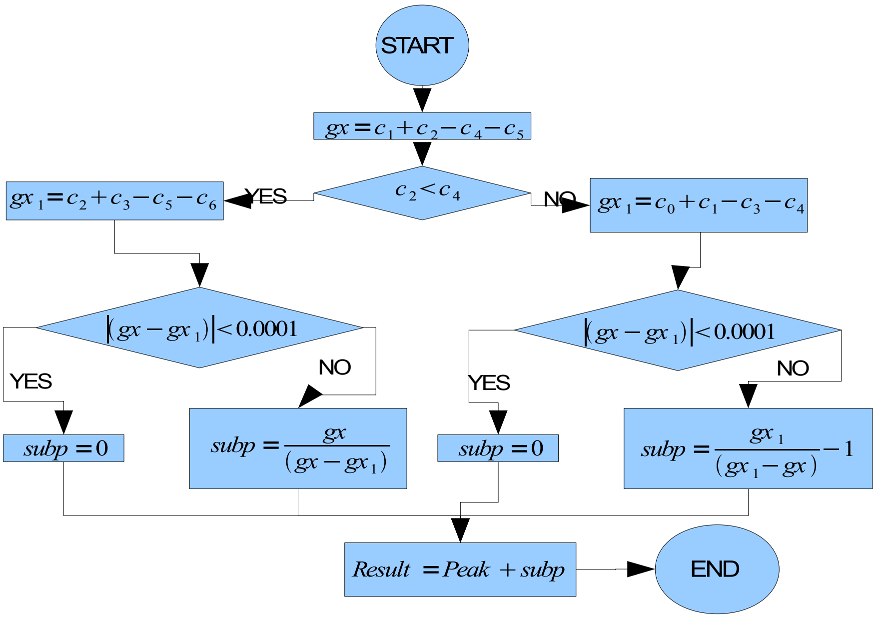


Figure 3.15. The flow chart of the sub-pixel operator

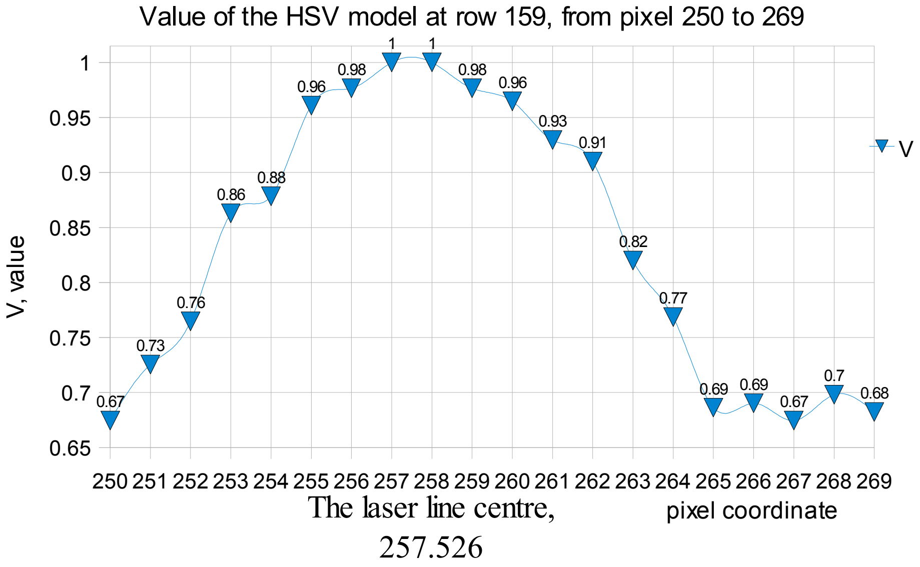


Figure 3.16 Close up plotting of the Value of the HSV model from image in Figure 3.7. The location of the centre of the laser line was obtained using the Blais and Rioux forth order [49] sub-pixel operator.

The result of the sub-pixel accuracy estimation of the pixels illuminated by the laser is shown in Figure 3.17. The algorithm performed very well, except where part of the laser line was absorbed by the checker board pattern, *i.e.* black colour square of the checker board. The false detection (Figure 3.18) was due to the row operation filtering process on an image subjected to chromatic aberration. The effect of chromatic aberration, which is an artifact of the webcam poor lens quality, created an edge of saturated pixels along the border between dark and light shades of the checker board patterns. When the pixels around those borders were illuminated by the laser line the pixel location with the largest illumination value was not any more at the centre of the laser line but rather at the border between the dark and light shade of the checker pattern. Hence the algorithm to estimate the location with sub-pixel accuracy got applied around the wrong location, in the same row, having the highest illumination. Figure 3.18 shows the wrongly detected pixel location with high intensity value (marked in Red) and the actual pixel location in the laser line (marked in White).

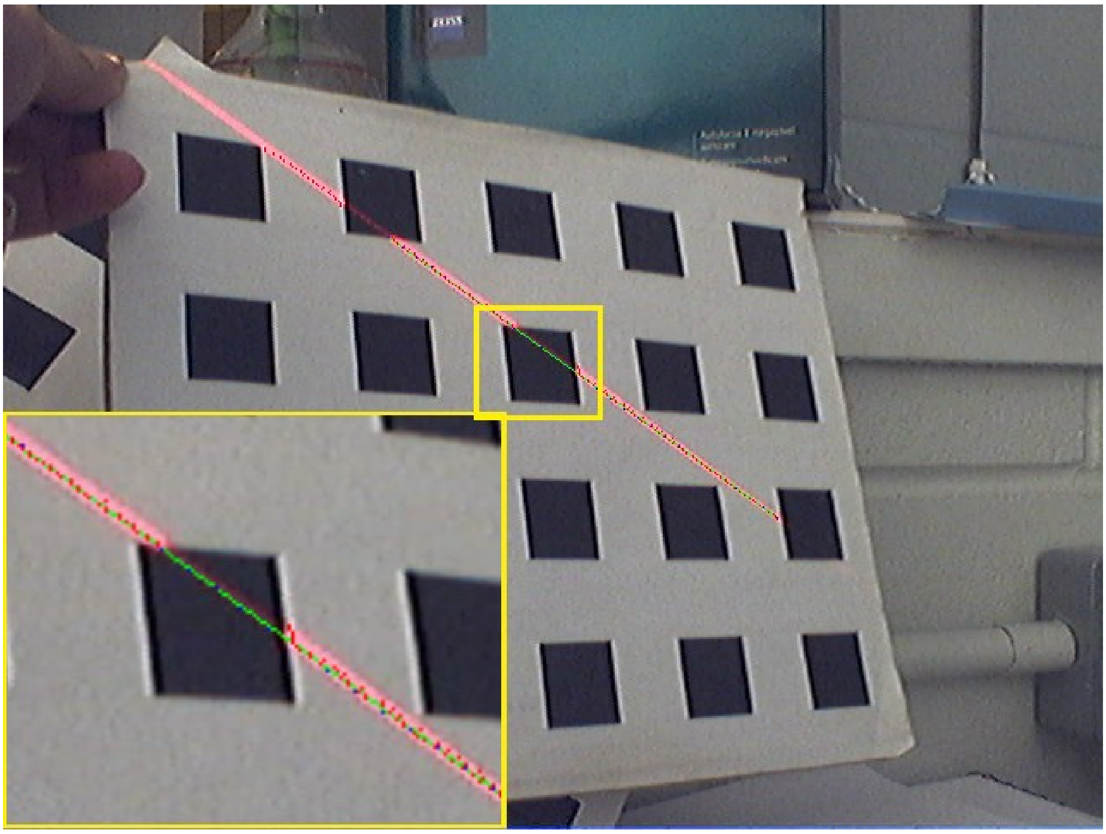


Figure 3.17: The result(red dot in bright colour) of thresholding with the intensity value of HSV which belong to laser line with sub-pixel accuracy are plot on top with the result of Hough Transform line detector(green line) for comparison.

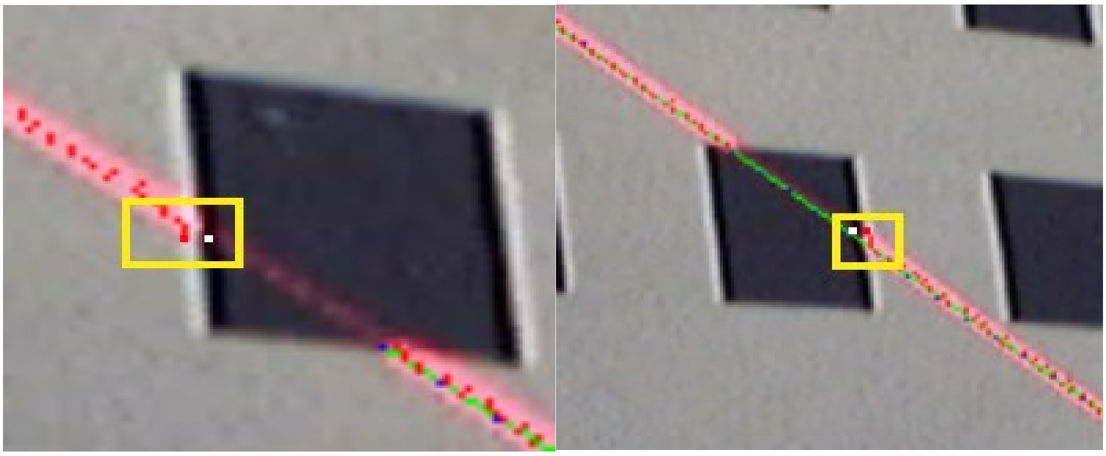


Figure 3.18: False laser pixel detection in a row of the image. The detected pixel (bright red) and the actual position (white colour) at the same row.

3.4.2.3 Inverse projection of the 2-D coordinates of the image pixels illuminated by the laser onto the checker board plane

The coordinate of the image pixel $([u \quad v \quad 1]^T)$ illuminated by the laser line was back projected to the planar object by using equation (3.21). The inverse of the estimated planar homography was used to transform the coordinates of the pixels illuminated by the laser to the surface point ($[X \quad Y \quad 1]^T$) on the checker board.

From equation (3.4) we inverses the $Homography_{3x3}$ to obtain the equation (3.21)

$$\lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}}_{:=Homography_{3x3}}$$

$$\lambda_X \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}^{-1} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{:=Homography_{3x3}} \tag{3.21}$$

3.4.2.4 Rigid transformation of 3-D points on the checker board coordinate frames to the camera coordinate frame

There are more than one coordinate frames in the system, namely one coordinate frame with respect to the camera, one coordinate frame with respect to each of the horizontal and vertical checker board planes. Since we have more than one coordinate frames in the system, all the detected points have to be transformed to a global reference coordinate frame. The camera coordinate frame has been chosen as our global reference frame (Figure 3.19). Equation 3.22 shows how a point in a coordinate frame is transformed to the camera coordinate frame.

$$P_{camera4\mathrm{x}1} = {}^{camera}\Gamma_{object4\mathrm{x}4} \cdot P_{checkerboard4\mathrm{x}1} \tag{3.22}$$

Where

$$P_{checkerboard4\mathrm{x}1} = \begin{bmatrix} X & Y & 0 & 1 \end{bmatrix}^T$$

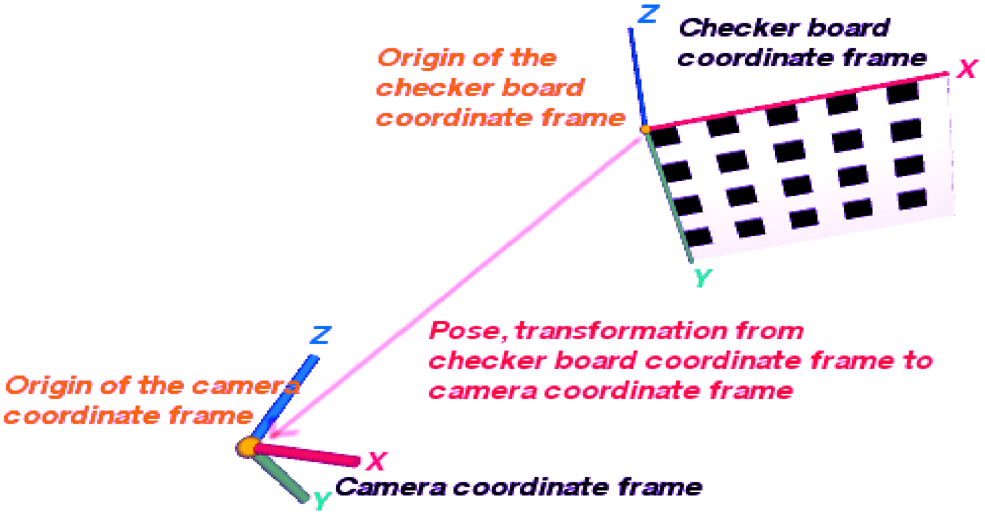


Figure 3.19: Transformation from the checker board coordinate frame to the camera coordinate frame.

3.4.2.4 Laser plane pose equation

As mentioned earlier, the laser plane pose estimation involves the detection and extraction of the points on the image illuminated by the laser and fitting a line to the detected set of points. Using the fitted line information, the laser plane pose was estimated.

The previous sections (3.4.2.1 and 3.4.2.2) elaborated the detection and extraction of the points on the image illuminated by the laser and fitting a line to the detected set of points. This section will give a brief outline of how the pose of the laser plane can be estimated by using the 2-D coordinates of the image pixels illuminated by the laser line. Chapter 5 will discuss in detail the laser plane pose estimation.

A 3-D plane (Figure 3.20) can be defined by any 3-D point on the plane (m) and the unit normal vector of the plane $(\vec{n})$. Then, any 3-D point X on the plane will satisfy the equation (3.23).

$$\vec{n} \cdot (\overrightarrow{X - m}) = 0 \tag{3.23}$$

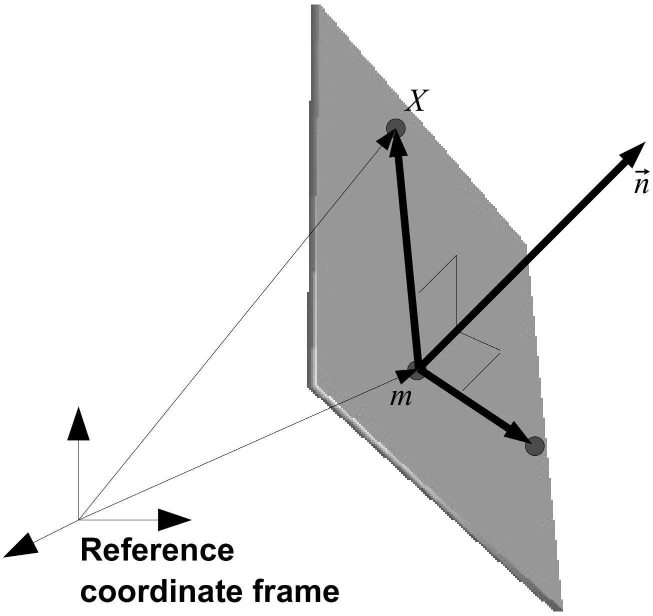


Figure 3.20: A plane in 3-D space (shaded).

Using the 3-D points locations, estimated through the detection of the points illuminated by the laser, the equation of the laser plane and hence $(m, \vec{n})$ of the laser plane can be estimated. This process is elaborated in more detail in Chapter 5.

$$\theta = \frac{180(\cos^{-1}(n_1 \cdot n_2))}{\pi} \tag{3.24}$$

3.4.3 Three dimensional (3-D) points cloud estimation

The final component which is required for 3-D surface acquisition is the three dimensional coordinate points estimation. Once the pose of the laser plane has been estimated (Section 3.4.2) the 3-D points on the surface of the object can be estimated by finding the intersection of the laser plane with the surface of the object. This could be achieved by the process of triangulation. The triangulation process is discussed in detail as follow.

3.4.3.1 Triangulation process

The 3-D reconstruction process was completed with the generation of a cloud of 3-D points by extracting the laser line and transforming the 2-D points on the laser line to the 3-D space. This process was performed as described below.

The laser line seen in an image is caused by the intersection of the laser plane and the surface of the object being scanned. Hence, the points on the laser line seen in an image are the points on the surface of the object. Using the image of the laser line, captured by the camera, the 3-D co-ordinates of the laser line can be computed. From the camera pinhole projection model, a point on an image will lie on a line which passes through the centre of the camera and the point on the scanned object which formed the image. Using this model the 3-D coordinates of the point on the scanned object could be resolved by finding the intersection of the camera projection ray with the laser plane line. A set of generated 3-D coordinates were then transformed to the 3-D surface points by resolving the position of the turntable during the time the image was captured. Thus, a cloud of points, in 3-D, representing the surface of the scanned object was generated at the end of the 3-D reconstruction process.

The “depth” of a laser point on an object, that appeared on an camera image was resolved (Figure 3.21) by using the method of triangulation (equation (3.25)). A ray (equation 3.26) was formed starting from the origin of the camera coordinate frame, penetrating the camera image and terminating on the object. The length of the ray (equation 3.27) was resolved by finding the intersection of the ray and laser plane.

$$Q_i = k\vec{r}_i \tag{3.25}$$

$$k = \frac{m_x n_x + m_y n_y + m_z n_z}{r_x n_x + r_y n_y + r_z n_z} \tag{3.26}$$

$$\vec{r}_i = Intrinsic_{3x3}^{-1}[u_i \quad v_i \quad 1]^T \tag{3.27}$$

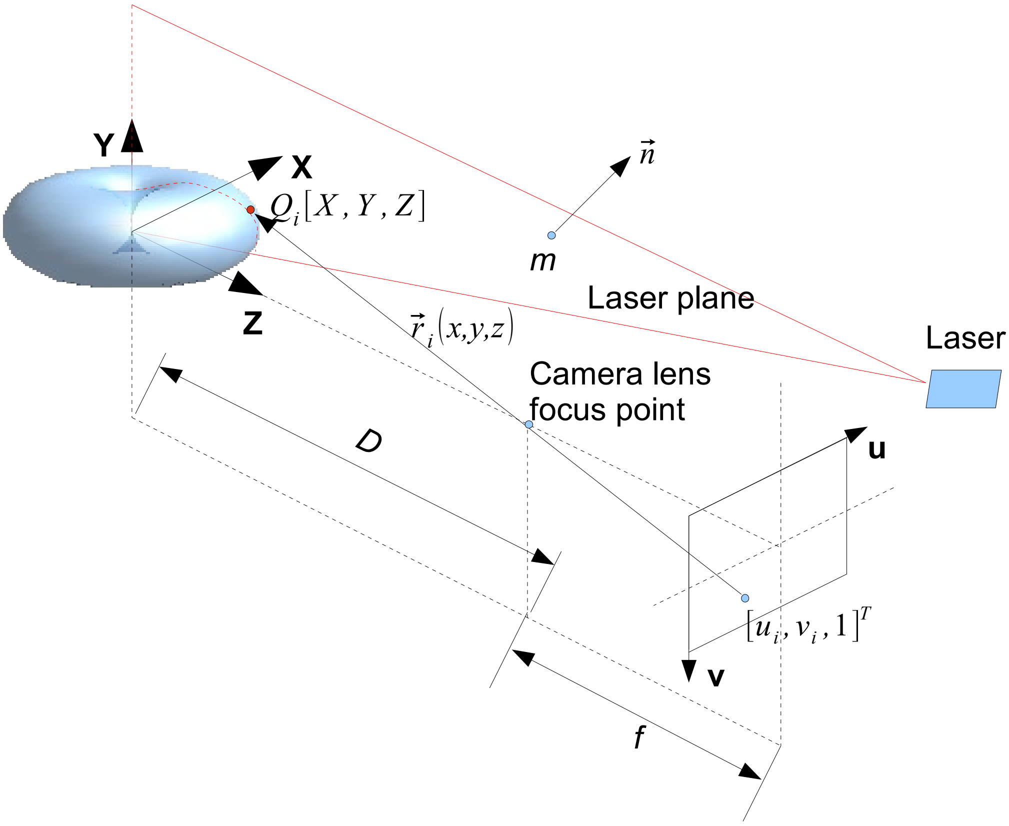


Figure 3.21: Principle of triangulation.

3.5 The implemented freehand 3-D surface data acquisition system

Based on the identified essential requirements for a Freehand 3-D surface reconstruction system discussed in Section 3.2, a simple 3-D surface data acquisition software with Graphical User Interface (GUI) was designed and implemented. The GUI was created with Qt [50] and the machine vision algorithms were implemented with MIMAS Toolkit. The designed GUI was shown in Figure 3.22.

A tool bar (with four buttons namely, 'C', 'O', 'B', 'L') was provided to perform the basic operations of the scanning system. The following provides a summary of the operations performed by each buttons.

- 'C': performed corner detection.
- 'O':performed object segmentation.
- 'B'; captured a camera image and saved it as a background reference image.
- 'L': started/stopped the laser scanning process.

Two canvases, namely the camera view and 3-D view were provided to display in real time the camera view and the result of triangulation respectively (Figure 3.23). The detected laser pixels (blue colour in Figure 3.23 left), which were the result of the background subtraction, were superimposed onto the camera image. The pixels lying in the segmented object boundary were rendered in red colour. The pixels used for laser plane pose estimation were rendered in yellow colour.

Figure 3.23 illustrates an screen shot of the process of acquiring 3-D surface data points of a deformable breast phantom for medical training (Figure 3.24). The camera was positioned around 300mm from the top of the breast phantom (Figure 3.25). The result of the 3-D surface data point acquisition is shown in Figure 3.26. The 3-D point cloud was subsequently processed by the PowerCrust to create the surface mesh (Figure 3.27). An smoothing filter provided by Meshlab was used to smooth the surface. The result of smoothing process, along with the colour rendering are shown in Figure 3.28.

The implemented software was running with a Intel Core2 Duo 2.10 GHz CPU (single process, not multi-threaded) that could perform at around 3 frames per second.

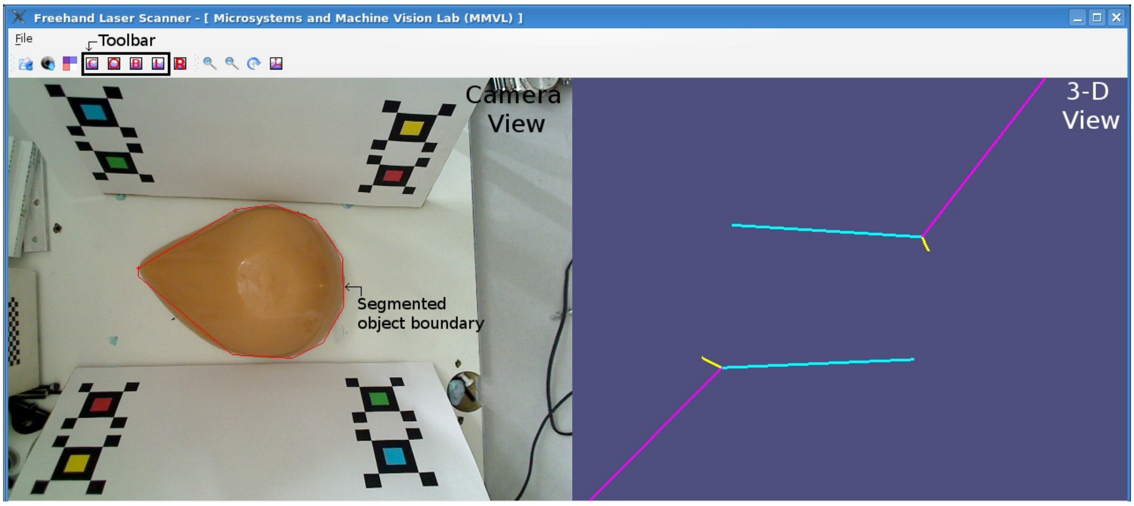


Figure 3.22: Figure illustrated the designed and implemented GUI for 3-D surface data acquisition system. A tool bar and two canvas, namely, Camera view and 3-D view were provided.

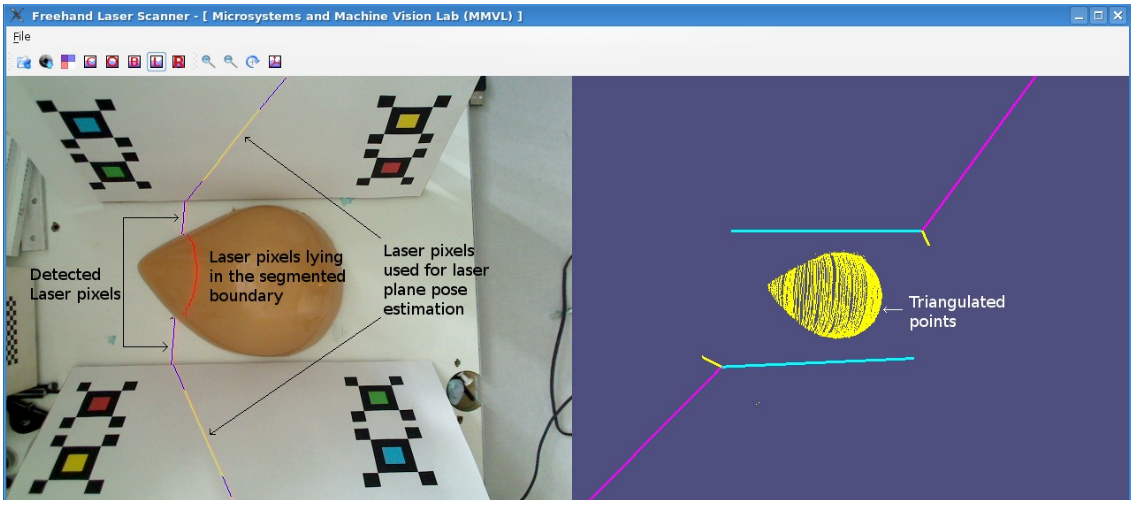


Figure 3.23: A screen shot of the GUI during the process of scanning a breast phantom. The result of background subtraction operation was superimposed onto the capture camera image and displayed in the camera view canvas (Left). The result of triangulation was shown, in live, in the 3-D view canvas (Right).

Figure 3.24: The object scanned, breast phantom.

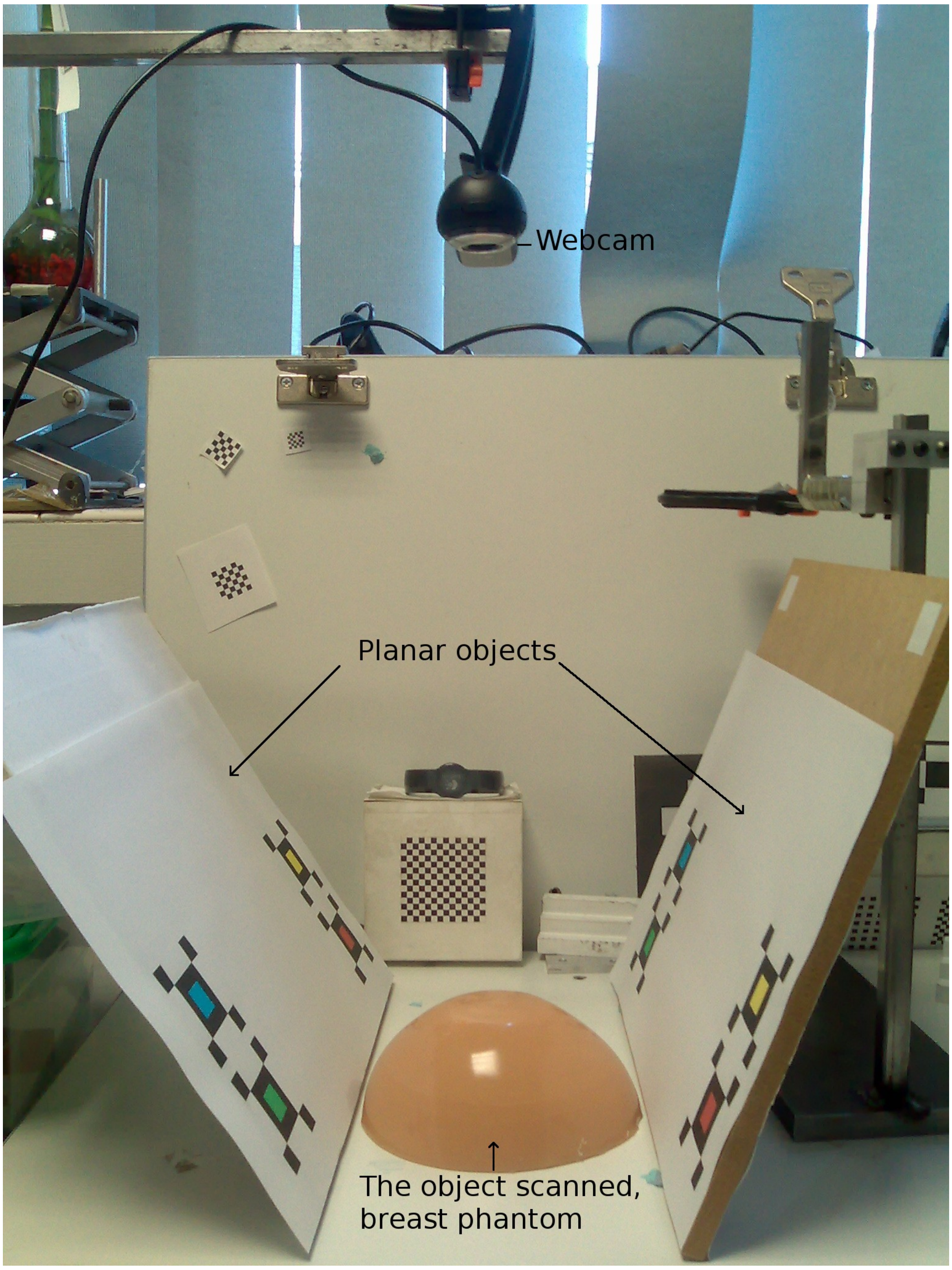


Figure 3.25: The setting for scanning the deformable breast phantom.

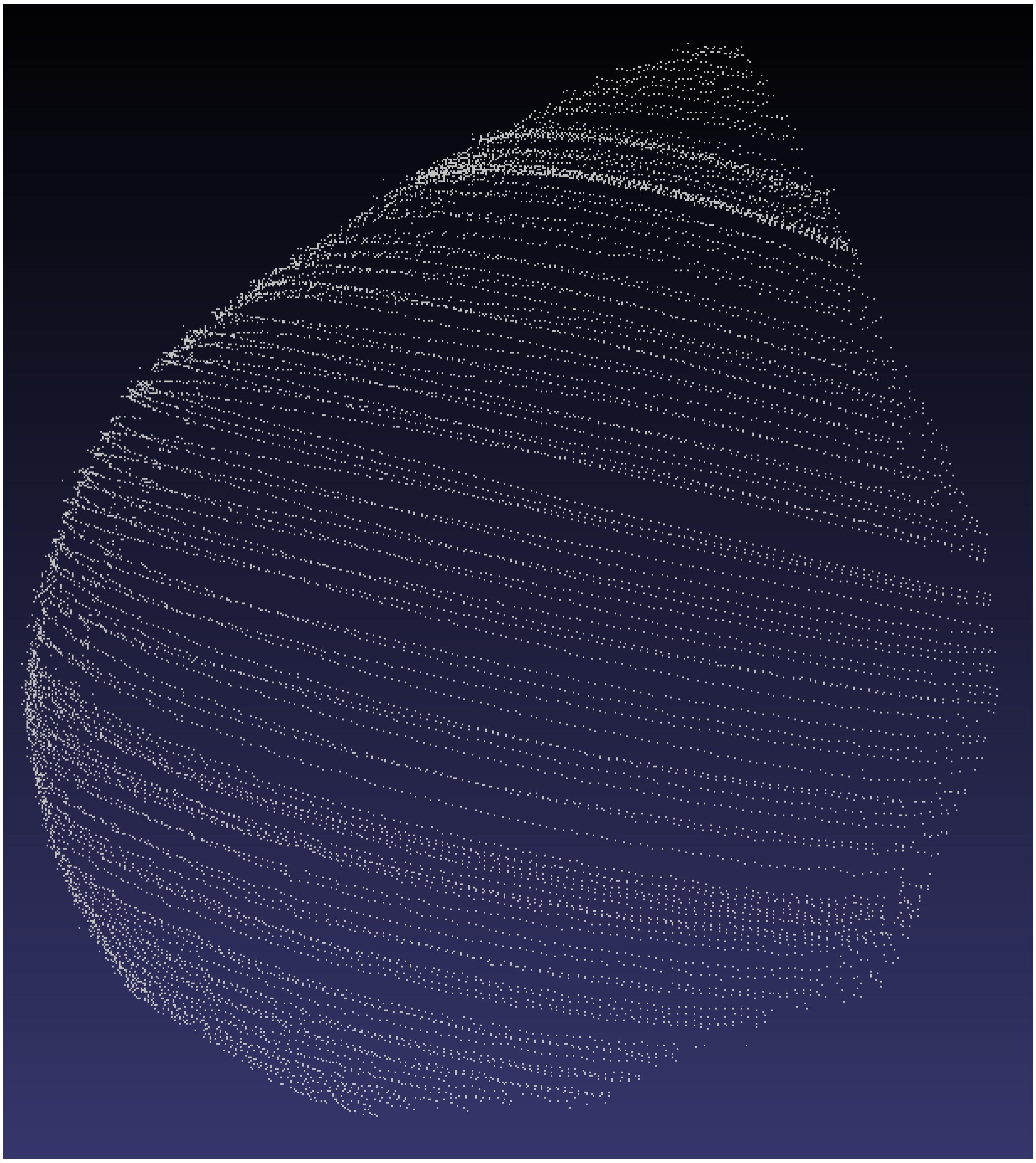


Figure 3.26: Rendering the scanning result, a cloud of 3-D points, of the breast phantom with Meshlab.

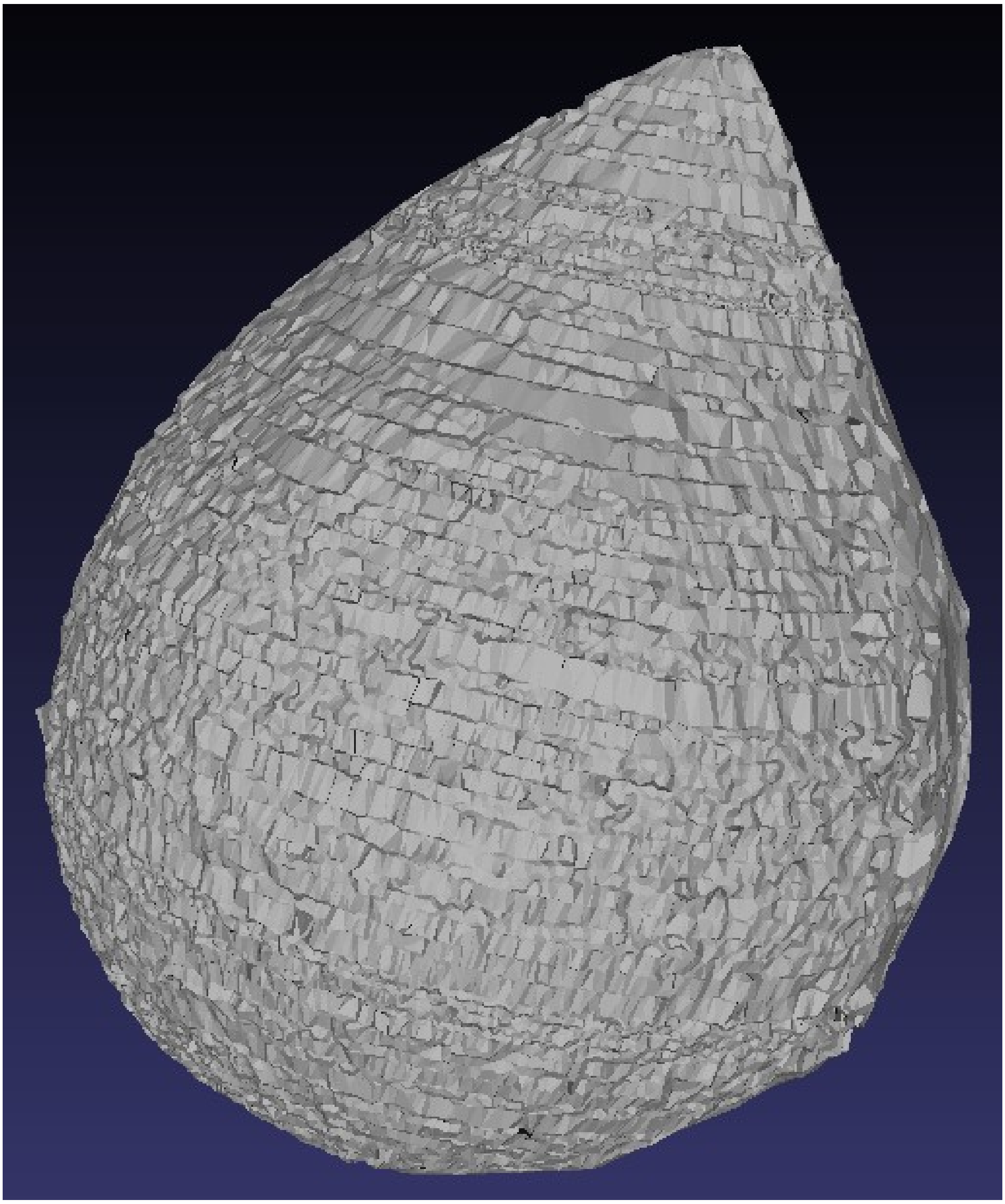


Figure 3.27: The surface meshes generated by PowerCrust.

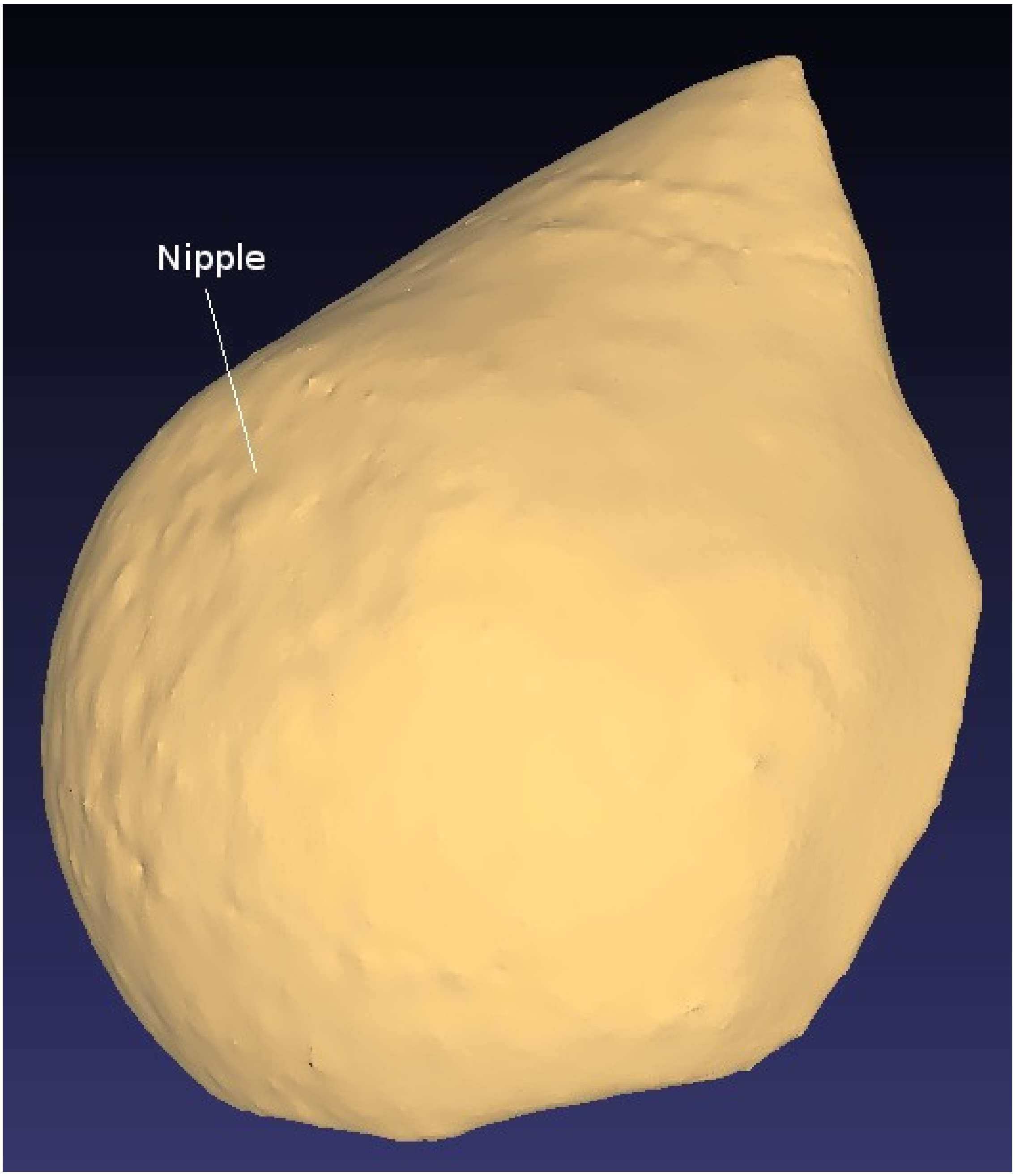


Figure 3.28: The result of smoothing the surface meshes with Meshlab. The nipple was able to visualised after the smoothing operation.

3.6 Conclusion

A system framework, consist of all the essential elements of a position sensor-less, freehand 3-D surface data acquisition system was designed and proposed. The framework clearly identify the processes flow between the essential elements in the system, right from the begining process of camera calibration to the end process of real time updating the display window with the acquired 3-D surface data points.

Difference camera calibration rigs were found from the literatures. The principle of 2-D planar object based camera calibration algorithm was discussed and adopted in this project. The key element of 2-D planar object based calibration method, the estimation of planar homography and subsequently determining the pose of the planar object was used to carry out the laser plane pose estimation process.

An accurate method of extracting the image pixel illuminated by the laser line is the fundamental of the laser plane pose estimation process. Background subtraction technique was found to perform better compared to the colour thresholding technique. A scan line based sub-pixel estimator found from the literature was adapted to extract the location of the image pixel, illuminated by the laser, with up to sub-pixel accuracy.

Using the explained triangulation method, the image pixels illuminated by the laser line, lying on the surface of the scanned object, were triangulated to produce a cloud of 3-D points. A real-time updating window was implemented to allow the visualisation of the acquired 3-D data points during the scanning process.

Based on the proposed framework, a position sensor-less, freehand 3-D surface data acquisition system was successfully implemented. The implemented system was managed to perform the 3-D surface data acquisition on the fly, with around 3 frames per second.

Chapter 4: Salient Features Detection and Evaluation

4.1 Introduction

As discussed in chapter 3 (section 3.3.2), camera calibration using 2-D planar object based calibration method requires planar homography estimation. Planar homography is also needed for planar pose estimation. To estimate planar homography, one needs to determine the correspondence between the camera image of the planar calibration object and the actual object itself (section 3.3.2.1). To estimate this correspondence, one needs to know the locations of the salient features of the planar object. Different types of markers or patterns can be implanted, with known locations, on the surface of a planar object to suit this task. In deciding the best type of salient features, three basic requirements were proposed; namely, detecting the locations of the salient features in the image with up to sub-pixel accuracy, measuring the length and the angle of a well calibrated pattern (black square, say) on a planar object. It is a normal practice to use corners, formed by high contrasting colours, as the salient features.

The pattern used in the current study was black squares of known dimension, repeated along the X and Y directions with a known distance between them. The features which were used to establish the extremities of the squares (and thus the locations, lengths and angles of the squares) were the corners of the squares.

In this study, three different categories of corner detection algorithms were selected for benchmarking. The first category was image intensity variation based. This category of algorithms uses the variation of the image intensity values for corner feature detection. The second category of method used was template-based corner features detector. In this category a template based matching algorithm was used to correlate and detect the corner features within the image. In the last category, a model-based corner features detector was used. In this method the static templates, which were used in the second method, were replaced by different models of corners to match and detect the corner features in the image.

The performance of the selected corner detectors were evaluated and are described in the final section of this chapter.

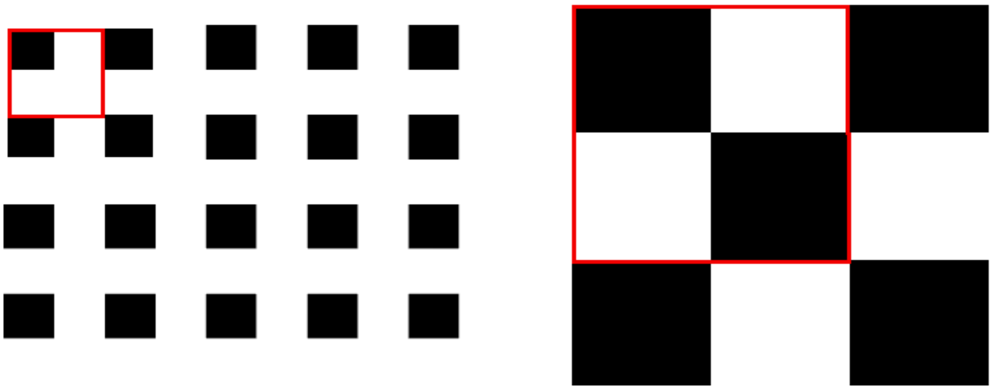
4.2 Introduction to corner detector

The pattern which was used in the current study was black squares of known dimension repeated along the X and Y directions with a known distance between them. In this section, first the types of corners produced with different types of arrangement of the black squares are discussed. Then, the basic criterion for an accurate corner detector is discussed. Sub-pixel operator was incorporated by the different corner detection algorithms to refine the result of corner detection. Two different sub-pixel operators are discussed at the end of this section.

4.2.1 Introduction to different types of corners namely L-corner and X-corner

In this study, two different types of corner features were investigated and evaluated for their suitability. The two different types of corner features were L-corner and X-corner. Figure 4.1 shows the two different types of corner features, namely L-corner and X-corner.

L-corner and X-corner are widely used in camera calibration applications. L-corner and X-corner patterns were used because they exhibit hight contrast in the image. The high contrast is achieved by the dark colour of the square against a white background. The squares used were of the same dimension repeated at regular interval. Since the size of all the squares and the interval between them are the same for each of the corners of the squares, a grid coordinate could be assigned. This is illustrated in Figure 4.2. In this study both the L-corner and X-corner were used to create the test images which were used for calibration.



(a) *(b)*

Figure 4.1: High contrast checker board pattern used for calibration.

(a) The L-corner where a pattern is formed with one black square surrounded by three white squares (outlined in red)

(b) The X-corner where a pattern is formed with a pair of black squares and a pair of white squares diagrammatically opposite (outlined in red)

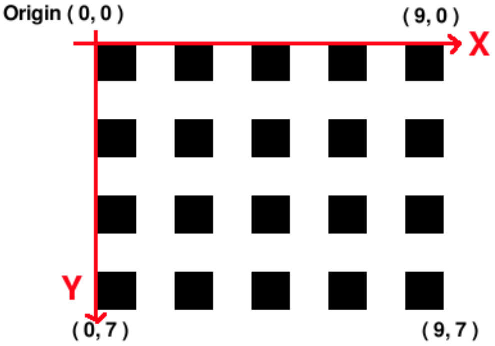


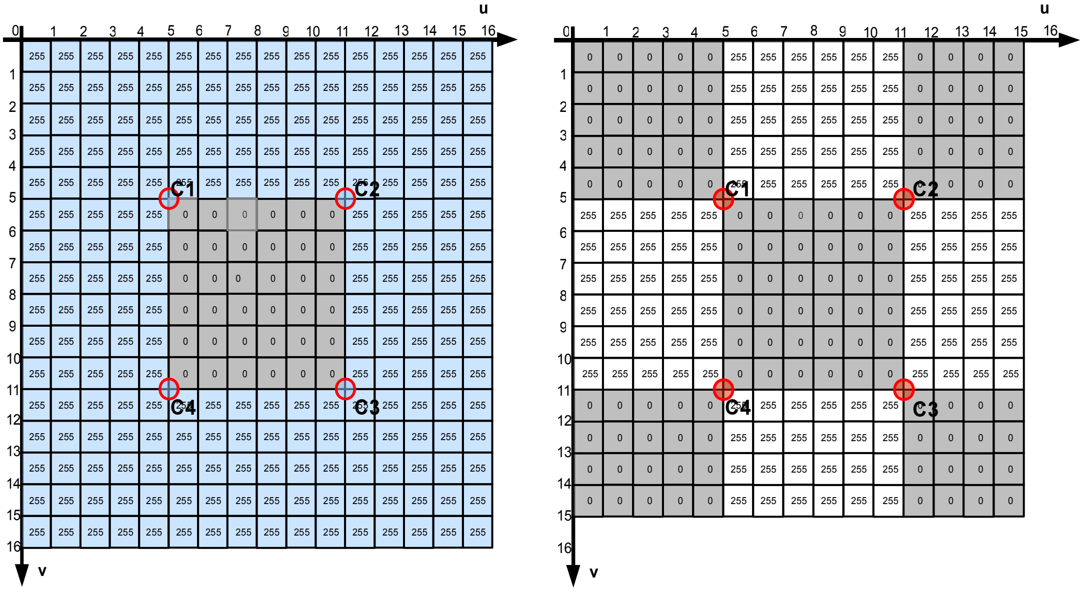
Figure 4.2: Pattern formed with dark squares of the same size repeated at regular interval. The repetitive planar pattern forms a 2-D grid coordinate system on a plane using which each of the corners are assigned with a fixed coordinate value.

4.2.2 Basic criteria for an accurate corner detector for detecting corner on a planar checker board pattern

An image is formed when the checker board pattern is captured by a camera. Figure 4.3 shows the sample images of two different checker board patterns when the checker board model is placed perpendicular to the camera viewing axis. Figure 4.3a depicts the L-corner that was formed by surrounding a black square (pixel value 0) all around by white squares (pixel value 255). Figure 4.3b depicts the X-corner that was formed when a black square is surrounded by white squares on its four sides and black squares along the two diagonals.

The image coordinate system is defined as shown in Figure 4.3. The position of a corner in an image is defined by a Cartesian coordinate (u , v). If the side lengths of each square of each side of the black square is 6 pixel units, *i.e.* the euclidean distance between C1 and C2 is 6 pixel unit, then the coordinates of the corners (L-corner in the left image and X-corner in the right image) in the images are C1(5,5), C2(11,5), C3(11,11), and C4(5,11), correspondingly. The angle at each corner location are 90 degrees. The value of the angle at any corner could be obtained by the dot product between the unit vectors of the two sides which forms the corner. For example, the dot product between the unit vector formed by C1C2 andC1C4 gives the value of the angle of the corner formed by the sides C1C2 and C1C4 which should be a right angle (Equation 4.1).

$$\theta_{C1}=\cos^{-1}(\|\vec{C1C2}\|\cdot\|\vec{C1C4}\|)*180/\pi \qquad (4.1)$$



(a) (b)

Figure 4.3: The figures show the Cartesian coordinate system of two pseudo images. Each image pixel is represented by a grey scale value ranging from from 0 (darkest) to 255 (brightest).

6x6 dark square is shown at the centre of the image. The four corners of the dark square, at the centre, are marked by circling them in red.

(a) The L-corner. (b) The X-corner.

4.2.3 Processes to refine the detected corner locations measured at discrete image locations

The process of detecting corner location in an image can be divided into two stages, namely firstly by applying the corner detector algorithm to detect the coarse location of the corner and subsequently refining the result to sub-pixel accuracy. Since both stages are processing intensive, for real time machine vision applications, we need to find a trade off between performance and accuracy.

A digital image captured from a camera, is the result of sampling an analogue scene into a two dimensional discrete set of values. The information in the original analogue domain can be reconstructed from the discrete digital image domain by using suitable interpolation techniques.

The result of the corner detection process yields a set of measurements of the strength of the corner at discrete pixel locations. In order to get the precise location of the corner using these measured values, of the strength of the corner at discrete locations, one can make use of a surface fitting technique. One of the main requirements for the type of chosen surface is that it has only one peak. Therefore, we could get the only location where the corner measure is at its maximum. Quadratic surface fitting is best suited for this purpose, because it is the lowest order surface which has got a single peak and thus it is less processing intensive, as compared with the higher order functions.

To fit a quadratic surface to a set of values measured at discrete locations, two general quadratic surface fitting techniques, namely the linear least squares estimator and the weighted linear least squares estimator were adapted to fit a set of two dimensional array digital values, measured at discrete locations. Both methods are discussed in detail in the following sections.

4.2.3.1 Quadratic fit with Least Squares Estimation

As described above in order to estimate the precise location of the corner, with up to sub-pixel accuracy, one needs to fit a quadratic surface to the fitting values estimated at discrete locations. The fitting of a quadratic surface to a set of values estimated at discrete locations, around the possible corner location, P^x, P^y, is carried out as follows.

The coordinates of the neighbouring discrete image locations from which the fitting values are considered to fit the quadratic function is indicated by S_{img} in Figure 4.4a. Figure 4.4b shows the coordinate system used to define the quadratic surface given by the equation (4.2).

$$Q_S(x,y)=C_1x^2+C_2xy+C_3y^2+C_4x+C_5y+C_6 \tag{4.2}$$

Making use of the fitting values from the discrete locations, the quadratic surface is estimated using the linear least squares estimation as detailed below.

From a three by three window, S_{img}, (Figure 4.4), centred at the location of a probable corner, $S_{img}[1,1]$ and the neighbourhood digital values within the three by three window are obtained. The origin of the quadratic fit coordinate system is located at the probable corner, *i.e.* $S_{img}[1,1]$. For example, the value of the quadratic surface at location $[0,0]$ was obtained from the three by three window location $S_{img}[1,1]$.

$$Ac = Q_S$$

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & -1 & -1 & 1 \\ 0 & 0 & 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 1 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{bmatrix}}_{c} = \underbrace{\begin{bmatrix} Simg[0,0] \\ Simg[1,0] \\ Simg[2,0] \\ Simg[0,1] \\ Simg[1,1] \\ Simg[2,1] \\ Simg[0,2] \\ Simg[1,2] \\ Simg[2,2] \end{bmatrix}}_{Q_S}$$

Solving the above set of equations by using linear least squares estimation results $A^T Ac = A^T Q_s$. Hence

$$c = (A^T A)^{-1} A^T Q_s \tag{4.3}$$

Subsequently the peak of the surface, $(\delta x, \delta y)$, is estimated. This is taken as the best estimate of the corner location. The location of the peak of the quadratic surface is found by partially differentiating the quadratic function, i.e. equation (4.2), with respect to x an y and equating them to zero as shown by equations (4.4) and (4.5). By solving the equations (4.6) and (4.7), the values of δx and δy were obtained. Making use of the location of the peak of the quadratic function, one can estimate the location where the maximum fitting occurs using Equation (4.8) which is the precise location of the corner.

$$\frac{\partial Q_S}{\partial x}(x,y)=2C_1 x+C_2 y+C_4=0 \tag{4.4}$$

$$\frac{\partial Q_S}{\partial y}(x,y)=C_2 x+2C_3 y+C_5=0 \tag{4.5}$$

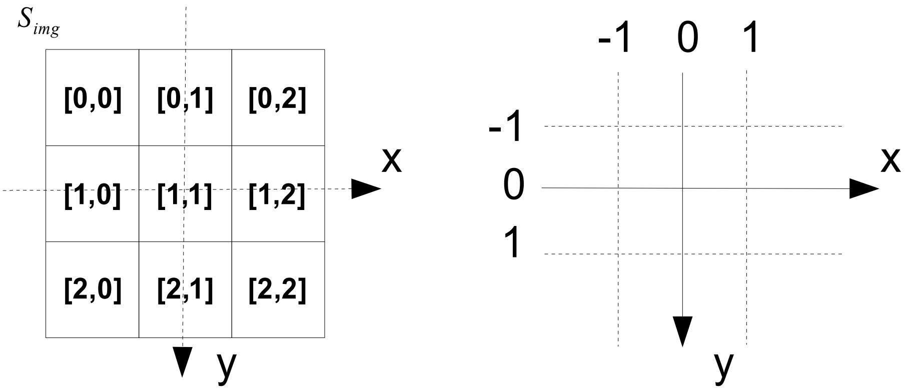
Deriving from (4.4)

$$\delta x=x=\frac{-C_2 y-C_4}{2C_1} \tag{4.6}$$

Substituting the value of x from equation (4.6) into equation (4.5) we get

$$\delta y=y=\frac{2C_1 C_5-C_2 C_4}{C_2^2-4C_1 C_3} \tag{4.7}$$

$$(P^x+\delta x, P^y+\delta y) \tag{4.8}$$



(a) *(b)*

Figure 4.4: The quadratic fit coordinate system (x,y) with least square estimator. (a) ***S_{img}*** *indices. (b) The quadratic coordinate system (x,y).*

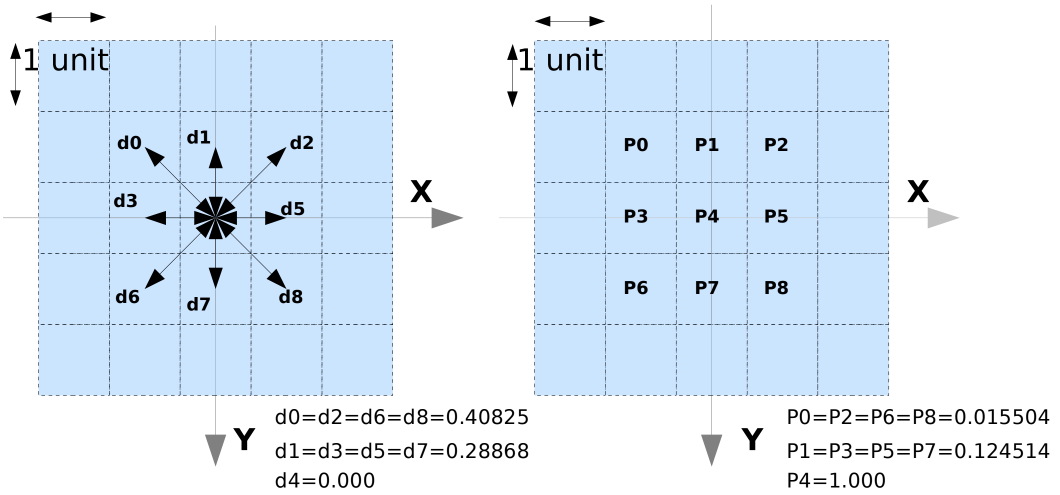
4.2.3.2 Quadratic fit with weighted least squares estimation

Weighted least squares estimator could be used to determine the equation of the best fitting quadratic surface instead of using quadratic surface with least squares estimator [51]. To assign weights to each of the locations considered, within the window S_{img} , a Gaussian function was used. The values of the weights depends on the normalised distance, (Figure 4.5a), of each pixel away from the centre pixel and were estimated using equation (4.9), where k a constant value. This process ensures that the further the location of the measured fitting digital value away from the centre, the lower the assigned weight , (Figure 4.5b), in the quadratic surface fitting process.

$$P_i = e^{-(d_i^2/k^2)}; i = \{0,1, ...,8\} \qquad (4.9)$$

Where

$$d_i = \frac{\sqrt{(x_i^2 + y_i^2)}}{\sum_{i=0}^{8} (\sqrt{(x_i^2 + y_i^2)})^2} \qquad (4.10)$$



(a) *(b)*

Figure 4.5: Figure illustrating the weight parameters equation (4.9) and distance measure equation (4.10) used to assign weighting to the fitting digital value measured around a pixel location.

(a) The distance values d_i *(b) The weight parameters* P_i

The process of quadratic fitting with weighted least squares estimation was performed as follows.

First, the most probable location of the corner was identified by locating the pixel location, (P_x, P_u) ,with highest fitting digital value. Then, a three by three window, S_{img} , (Figure 4.4), was centred at the location of the most probable corner location $S_{img}[1,1]$.

The origin of the quadratic fit coordinate system is located at the centre of S_{img} , *e.g.* the surface value of $Q_s[0,0] = S_{img}[1,1]$. To estimate the initial six coefficients, x_0 , of the quadratic function equation (4.2), six equations were formed by making use of the fitting digital values at the neighbourhood of the five pixels (

$S_{img}[0,1], S_{img}[0,2], S_{img}[1,0], S_{img}[1,2], S_{img}[2,1]$) locations and at the centre pixel $S_{img}[1,1]$ as indicated in equation (4.11).

Since the estimated fitting digital values could become corrupted by noise, the quadratic surface estimated using the fitted digital values may not be the actual surface, we are interested. To estimate the actual quadratic surface, one needs to estimate the noise value. Equation (4.14) gives the relationship between the quadratic surface $\hat{X}$ estimated using the fitting digital values, the actual quadratic surface X_0 and the noise parameter $\hat{x}$. The noise vector $\hat{x}$ was estimated by using equation (4.13). The peak of the actual quadratic surface $(\delta x, \delta y)$ was then estimated using the equation (4.15). Using $(\delta x, \delta y)$ the location of the actual corner was found as $(P_x+\delta x, P_y+\delta y)$. Thus, the refined corner location was found with sub-pixel accuracy.

$$A_0 X_0 = Q_S^0$$

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 & -1 & 1 \\ 1 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \end{bmatrix}}_{X_0} = \underbrace{\begin{bmatrix} S_{img}[1,0] \\ S_{img}[0,1] \\ S_{img}[1,1] \\ S_{img}[2,1] \\ S_{img}[0,2] \\ S_{img}[1,2] \end{bmatrix}}_{Q_S^0} \qquad (4.11)$$

$$\hat{x}=(A^T P A)^{-1} A^T P l \tag{4.13}$$

$$\hat{X}=X_0+\hat{x} \tag{4.14}$$

$$\begin{aligned} den &= \hat{X}_2 \hat{X}_2 - 4 \hat{X}_3 \hat{X}_1 \\ \delta x &= \frac{(2 \hat{X}_3 \hat{X}_4 - \hat{X}_2 \hat{X}_5)}{den} \\ \delta y &= \frac{(2 \hat{X}_1 \hat{X}_5 - \hat{X}_2 \hat{X}_4)}{den} \end{aligned} \tag{4.15}$$

4.3 Corner detection algorithms

In this section three different categories of corner detection algorithms are discussed. The first category is based on image intensity variation. This category of algorithms makes use of the variations in the image intensity values for corner feature detection. The second category is template-based corner features detector. In this category, a template based matching algorithm to correlate and detect the corner features within the image is used. In the last category, model-based corner features detector is used. In this method the static templates, which are used in the second method, are replaced by different models of corners to match and detect the corner features in the image.

4.3.1 Corner detection based on image intensity variation

Dutta *et. al.* in their survey of corner detection detection algorithms showed that 114 corner detection algorithms were implemented between 1977 to 2006 [49]. Of the various corner detection algorithms, Harris and Stephens (H&S) corner detector [52] is a well known corner detection algorithm. The H&S corner detector built upon an earlier corner detector known as Moravec' corner detector [53]. The major enhancement of H&S corner detector over Moravec' corner detector is the way the initial estimate of corners is performed. Also, H&S introduces a measure to estimate the cornerness of any detected corner locations. Where cornerness is a measure using which one can objectively measure the appropriateness of a pixel location being a corner [54].

The H&S corner detector works by determining the local maxima changes of pixel intensity within a small shifting window. The result of this initial estimate of the corner location is subsequently refined up to sub-pixel accuracy by applying a sub-pixel operator centred at the pixel location of each of the local maxima. The entire process is explained in detail in the following section.

4.3.1.1 Harris and Stephens corner detection method

A method to determine the cornerness of an image pixel location is by evaluating the changes of image intensity value, *I*, when moved from a particular image pixel location *(x,y)* in all the directions. Moravec [53] achieved this by evaluating the change in the image intensity value, within a small shifting square window. To start with a three by three window S was centred at the image pixel location (*x, y*) which needs to be evaluated for being a corner. The pixel values at the nine locations of the 3x3 window were obtained from the underlying image. Next, the 3x3 window was shifted in the eight directions (east, south east, south, south west, west, north west, north and north east). In a two dimensional image, this along the horizontal axis, *u*, and the vertical axis, *v.* The sum of the difference between each of the pixel values within the 3x3 window, centred at the image location, and the pixel values within the window in the shifted location were found using equation (4.16). The minimum of these eight values at the image pixel location (*x, y*) was stored into the cornerness map, $C(x,y)$ as indicated by (Equation 4.17).

$$V_{u,v}(x,y)=\sum_{a=-1}^{1}\sum_{b=-1}^{1}(I(x+u+a,y+v+b)-I(x+a,y+b))^2 \tag{4.16}$$

Where direction (*u,v*) are east (1,0), south-east (1,1) south (0,1), south-west (-1,1), west

$$C(x,y)=min(V_{u,v}(x,y)) \tag{4.17}$$

(-1,0), north-west (-1,-1), north (0,-1)and north east (1,-1)

A limitation of Moravec's approach is the strong cornerness response along the edges of an object in the image. This is due to the pixelisation, caused by the low resolution of the camera, and due to white noise. These limitations of Moravec's approach were addressed by Harris *et. al.* [52] as follows. To minimize the noise, Harris *et. al.,* convolved the image with a small square window of Gaussian weights. This smoothing operation was performed before processing the image to measure the cornerness at each pixel location in the image.

The problem of strong cornerness response along the edges was addressed, by Harris *et. al.,* as explained below. A matrix $G(x,y)$, made up of the variation of the intensity values along the image horizontal axis, *u*, and the vertical axis *v* at the (*x, y*) location was created as indicated by equation (4.18). The direction of the variation was analysed. The eigenvalues (λ_1, λ_2) of the matrix G, representing the direction of the variation were compared. For the edges there is a strong variation perpendicular to the direction of the edges. Hence, one of the eigenvalues is significantly larger compared to the others. At corner locations are large variations in both the directions. Hence the values of both λ_1 and λ_2, are large at the corner locations. The division of the eigenspace of matrix G into distinct features is shown in Figure 4.6 [52].

$$G(x,y)=\begin{bmatrix} I_u^2 & I_u I_v \\ I_u I_v & I_v^2 \end{bmatrix} \tag{4.18}$$

Where

$$I_u = I_w * [-1, 0, 1]$$
$$I_v = I_w * [-1, 0, 1]^T$$

$$I_w = I * w$$

and *, the convolution operator

w, the Gaussian weight

I, the intensity value of the pixel at the location (*x, y*)

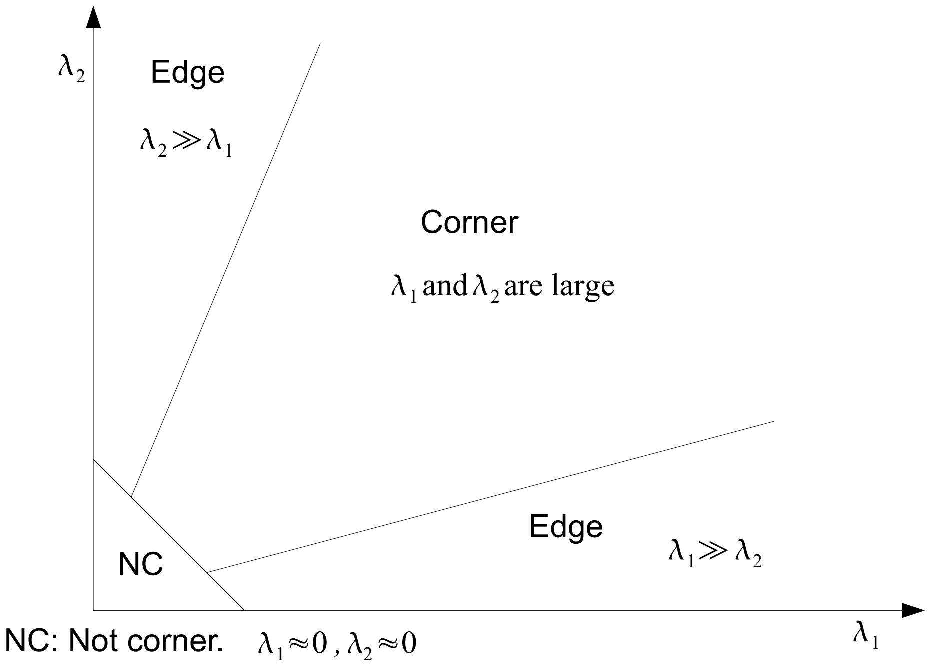


Figure 4.6: The division of eigenvalue space into distinct feature regions.

The determinant (Det) of matrix G, is the value of the product of the eigenvalues λ_1 and λ_2; while the trace (Tr) of the matrix is the value of the summation of the eigenvalues λ_1 and λ_2. Using the determinant and the trace values of the matrix G Harris *et. al.* provided a cornerness measurement from the determinant and trace value (see Equation (4.3)). This equation gives a higher strength to the corner at an image pixel location with higher values of λ_1 and λ_2. The result of the cornerness map is varied according to the chosen ***k*** value in the equation. Empirical testing [54] indicated that the range of *k* that yields the best result is 0.04 < ***k*** < 0.06. Figure 4.7 shows the contour plot of the cornerness map. Figures 4.8 and 4.9 show the H&S cornerness map of the image in Figure 4.3 for the value of ***k***=0.04. The pixel locations with local maxima, (P^x, P^y) ,are highlighted by a small circle.

$$R = Det - k\,Tr^2 \tag{4.19}$$

Where

R is the cornerness measure.

$$Tr = I_u^2 + I_v^2$$
$$Det = I_u^2 I_v^2 - (I_u I_v)^2$$

and $k \in \mathbb{R}$ is usually a very small value.

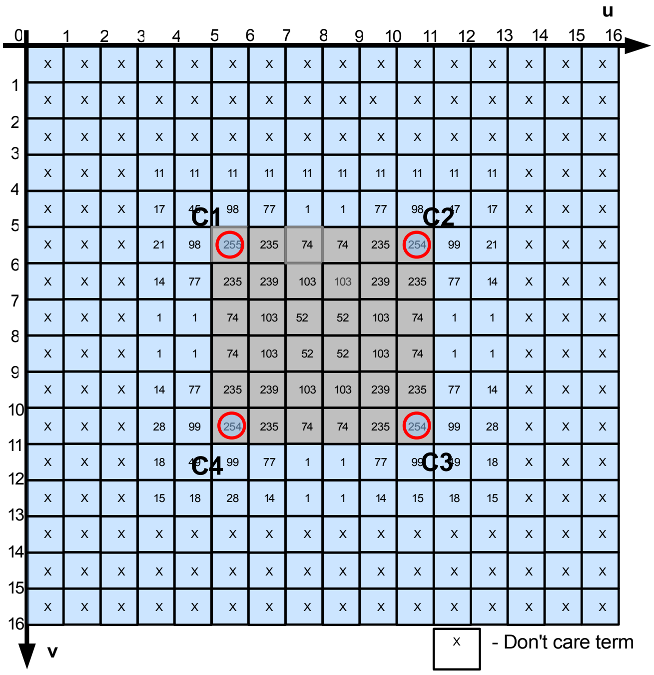
Title:/home/song/matlabout1.eps
Creator:MATLAB, The Mathworks, Inc.
CreationDate:06/04/2009 15:22:55
LanguageLevel:2

Figure 4.7: The contour plot of cornerness map C(x,y) with k=0.04

The process of detecting all the corner locations of the L-corners, from the cornerness map produced by H&S corner detector algorithm, required careful selection of the thresholding value. A thresholding value of 253 was used to threshold the cornerness map. Any pixel location with a cornerness value larger than the thresholding value was identified as a possible corner location. This is illustrated in the Figure 4.8.

The X-corner the thresholding process produces four two by two regions, with very high cornerness response, around the location of the X-corner (Figure 4.9). Hence, the process of finalizing the corner locations using H&S corner detector requires one more step, namely, the non-maxima suppression, to mark the pixel location with maximum cornerness value within a small window. Using a three by three window the pixel locations with comparatively lesser cornerness value are eliminated. The result of the non-maxima suppression is shown in Figure 4.10.

Once the local maxima, (P^x, P^y), indicating the coarse location of the corner, is estimated, a sub-pixel operator is applied, centred at the coarse location of the detected corner to estimate the actual location of the corner at sub-pixel accuracy.

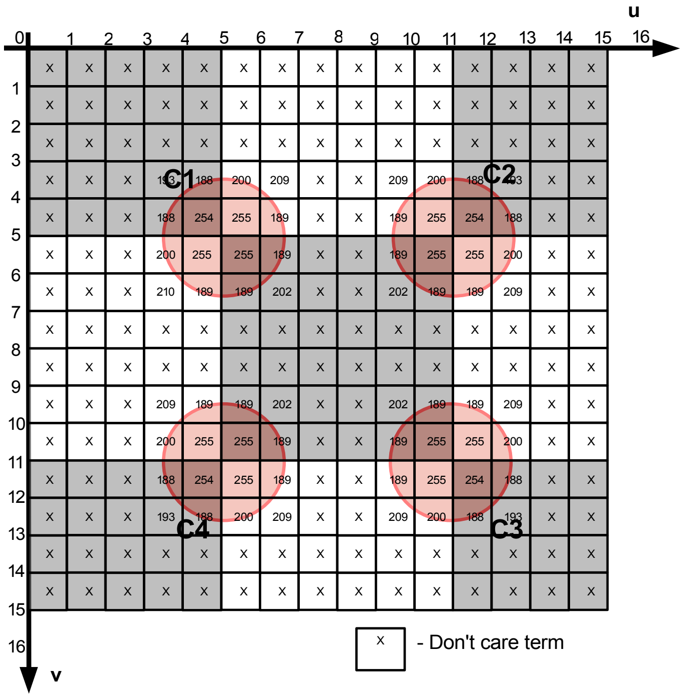


X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	11	11	11	11	11	11	11	11	11	11	X	X	X
X	X	X	17	45	98	77	1	1	77	98	47	17	X	X	X
X	X	X	21	98	255	235	74	74	235	254	99	21	X	X	X
X	X	X	14	77	235	239	103	103	239	235	77	14	X	X	X
X	X	X	1	1	74	103	52	52	103	74	1	1	X	X	X
X	X	X	1	1	74	103	52	52	103	74	1	1	X	X	X
X	X	X	14	77	235	239	103	103	239	235	77	14	X	X	X
X	X	X	28	99	254	235	74	74	235	254	99	28	X	X	X
X	X	X	18	49	99	77	1	1	77	99	49	18	X	X	X
X	X	X	15	18	28	14	1	1	14	15	18	15	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 4.8: Cornerness map produced by the H&S corner detector on the L-corner image (Figure 4.3a).

For each image pixel location the rounded normalised value of the cornerness at that location is given. The normalised cornerness value ranges from 0(weakest) to 255(strongest).

The image pixel locations (P_i^x, P_i^y) *with the highest H&S cornerness value are circled in red.*



X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	193	188	200	209	X	X	209	200	188	193	X	X
X	X	X	188	254	255	189	X	X	189	255	254	188	X	X
X	X	X	200	255	255	189	X	X	189	255	255	200	X	X
X	X	X	210	189	189	202	X	X	202	189	189	209	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	209	189	189	202	X	X	202	189	189	209	X	X
X	X	X	200	255	255	189	X	X	189	255	255	200	X	X
X	X	X	188	254	255	189	X	X	189	255	254	188	X	X
X	X	X	193	188	200	209	X	X	209	200	188	193	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 4.9: Cornerness map produced by the H&S corner detector on the X-corner image (Figure 4.3b).

For each image pixel location the rounded normalised value of the cornerness at that location is given. The normalised cornerness value ranges from 0(weakest) to 255(strongest).

The region of image pixel locations (P_i^x, P_i^y) *having the highest H&S cornerness values are circled in red.*

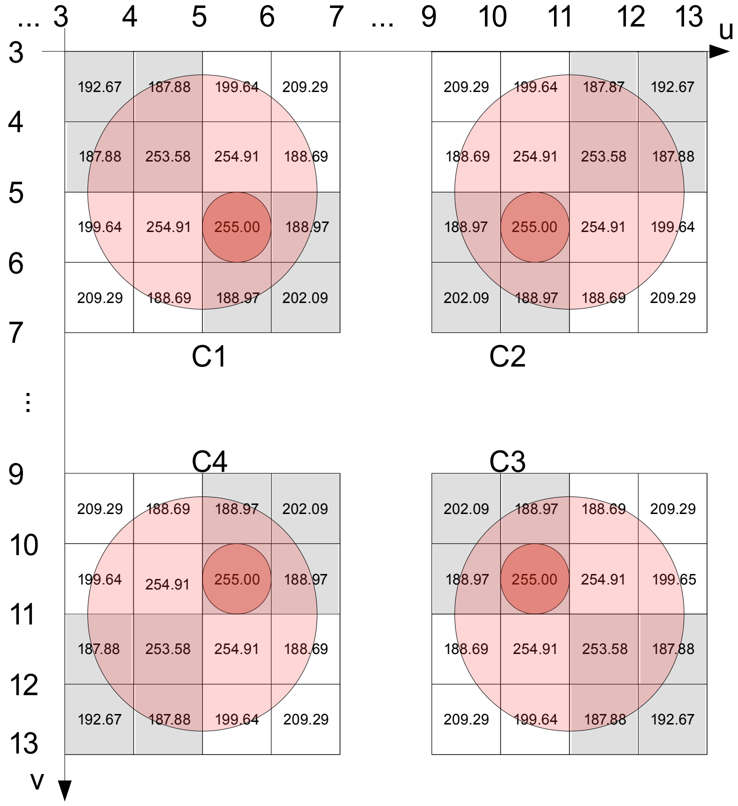


Figure 4.10: Zoomed in version of the cornerness map of Figure 4.9 .

The location of the local maxima, (P_i^x, P_i^y) *which was detected using non maxima suppression is the location of the corner.*

The location of the corner is marked with a smaller circle.

4.3.1.1.1 Process of refining cornerness map

It is assumed that the cornerness values obtained by measuring the cornerness at the discrete locations around a probable corner location form a paraboloid surface profile. By determining the peak of the paraboloid surface, the precise location of the corner can be resolved. Making use of any one of the quadratic surface fitting methods, described in the earlier sections, one can fit a quadratic surface to a set of values measured at discrete locations. If one uses the cornerness values, evaluated at discrete locations, by fitting a quadratic surface to the set of values one can estimate the precise location of the corner up to sub-pixel accuracy. This process is explained below.

To fit a set of two dimensional array of cornerness values, measured at discrete locations, to a paraboloid surface, the two methods discussed at sections 4.2.3, namely Least Squares-Estimator and Weighted Least Squares estimator are used

4.3.1.1.1.1 Fitting a quadratic surface function with Linear Least Squares Estimator

Using the location of the peak of the quadratic function one can estimate the location where the maximum cornerness occurs by linear least squares estimator (Section 4.2.3.1), which is the precise location of the corner. Figures 4.11 and 4.12 show a surface plot of the cornerness map around the corner C1 for L-corner and X-corner respectively.

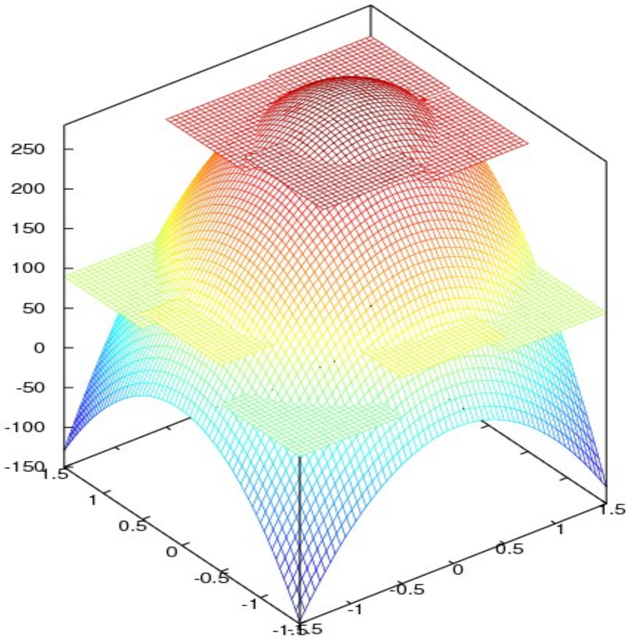


Figure 4.11: Quadratic surface fit of the cornerness map produced by H&S corner detector at a L-corner(C1 Figure 4.8).

The horizontal planes indicate the cornerness value around the neighbourhood. The peak of the quadratic surface is at (δx = 0.5413, δy = 0.5413).

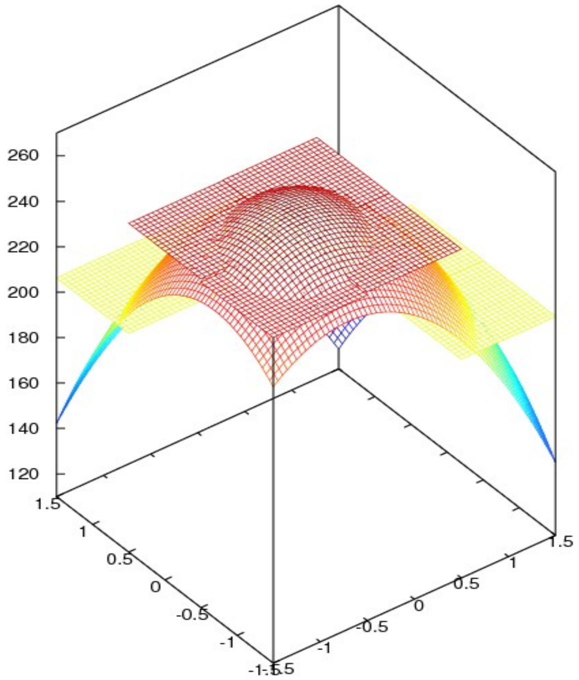


Figure 4.12: Quadratic surface fit of the cornerness map produced by H&S corner detector at a X-corner(C1 Figure 4.9).

The horizontal planes indicate the cornerness value around the neighbourhood. The peak of the quadratic surface is at (δx = -0.5957, δy = -0.5957).

4.3.1.1.1.2 Fitting a quadratic surface function with Weighted Linear Least Squares Estimator

Similar to the process of fitting quadratic surface function with linear least square estimator that is mentioned in section 4.3.1.1.1.1, the process was repeated with weighted least square estimator. A similar surface profiles were obtained. The peak of the quadratic surface of L-corner was found at (δx =0.4942, δy =-0.4942). The peak of the quadratic surface of X-corner was found at (δx =-0.5864, δy =-0.5864).

4.3.2 Template based corner detector

In a template based corner detection, initially one designs a template which will match the shape and size of the corner one wants to detect, and then the image is convolved with the designed template.

To detect corners on a planar object, one needs to create a specific template for each of the four corners. We consider four different types of L-corner(Figure 4.13).

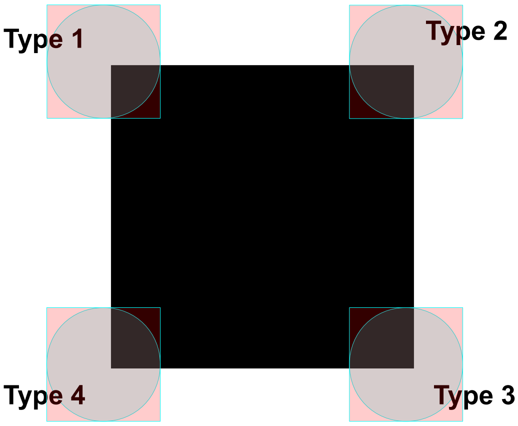


Figure 4.13: Four different types of L-corner.

4.3.2.1 Detecting the L-corner with cross shape templates

To detect L-corner, in a binary image, Pachidis *et. al.*[55] proposed the use of cross shaped corner templates of size 11×11 pixel. They designed two cross shaped templates (Template$_{\text{CenterS}}$(Figure 4.14a) and Template$_{\text{CenterC}}$(Figure 4.14b)) for convolving the image and located the corners. The difference between the two templates is that at the centre, one template has the pixel value of the background (white), while the other has the pixel value of the corner (black). The two templates were rotated at the centre by 90, 180, and 270 degrees to detect the L-corners at Type2, Type3, and Type4 (see Figure 4.13) respectively.

To find the cornerness response at a pixel location, the template is centred at the pixel location, the template pixel values are convolved with the image pixel values and the correlation coefficient is found (see equation 4.20). For each type of the corner (see Figure 4.13) convolving with the corresponding pair templates yields four pixel locations with high cornerness response. To estimate the actual location of the corner, one needs to find the mean of the locations having the highest cornerness response. For example Figure 4.15 shows the pixel locations having the highest cornerness response.

In Figure 4.15 the pixels locations with the highest cornerness response, to the pair of templates, are (4,4),(4,5),(5,4) and (5,5), hence the corner location is the mean of the set of four pixel locations which is (4.5, 4.5).

The estimated corner location, by the above process, is an offset to the actual corner location. Since the offset is uniform, one can subtract the offset from the estimated corner location to find the actual corner location.

Since the error of all detected locations of the corner is constant, the result is offseted to get back the actual corner location.

For binary images with a regular pattern (*e.g.* Checker board) Pachidis [55] has also proposed to scan along the slope of the line, formed by the current detected centre point ((C1+C2+C3+C4)/4) with the previous detected centre point, to reduce the number of operations required for scanning through the entire image, hence increasing the performance by reducing the computational time of the template-based corner detector.

$$corr(Img, Template) = \frac{Cov(Img, Template)}{\sqrt{Cov(Img, Img) * Cov(Template, Template)}} \tag{4.20}$$

S	x	x	x	x	x	x	x	x	x	S
x	S	x	x	x	x	x	x	x	S	x
x	x	S	x	x	x	x	x	S	x	x
x	x	x	S	x	x	x	S	x	x	x
x	x	x	x	S	x	S	x	x	x	x
x	x	x	x	x	S	x	x	x	x	x
x	x	x	x	S	x	C	x	x	x	x
x	x	x	S	x	x	x	C	x	x	x
x	x	S	x	x	x	x	x	C	x	x
x	S	x	x	x	x	x	x	x	C	x
S	x	x	x	x	x	x	x	x	x	C

(a)

S	x	x	x	x	x	x	x	x	x	S
x	S	x	x	x	x	x	x	x	S	x
x	x	S	x	x	x	x	x	S	x	x
x	x	x	S	x	x	x	S	x	x	x
x	x	x	x	S	x	S	x	x	x	x
x	x	x	x	x	C	x	x	x	x	x
x	x	x	x	S	x	C	x	x	x	x
x	x	x	S	x	x	x	C	x	x	x
x	x	S	x	x	x	x	x	C	x	x
x	S	x	x	x	x	x	x	x	C	x
S	x	x	x	x	x	x	x	x	x	C

(b)

Figure 4.14: Two different cross shape templates. The two templates vary by the template's pixel value at the centre of the window(highlighted in grey colour). 'S' indicates the pixel value of background colour, white. 'C' indicates the pixel value of the black square. 'x' indicates don't care.

(a) $Template_{CenterS}$. *(b)* $Template_{CenterC}$.

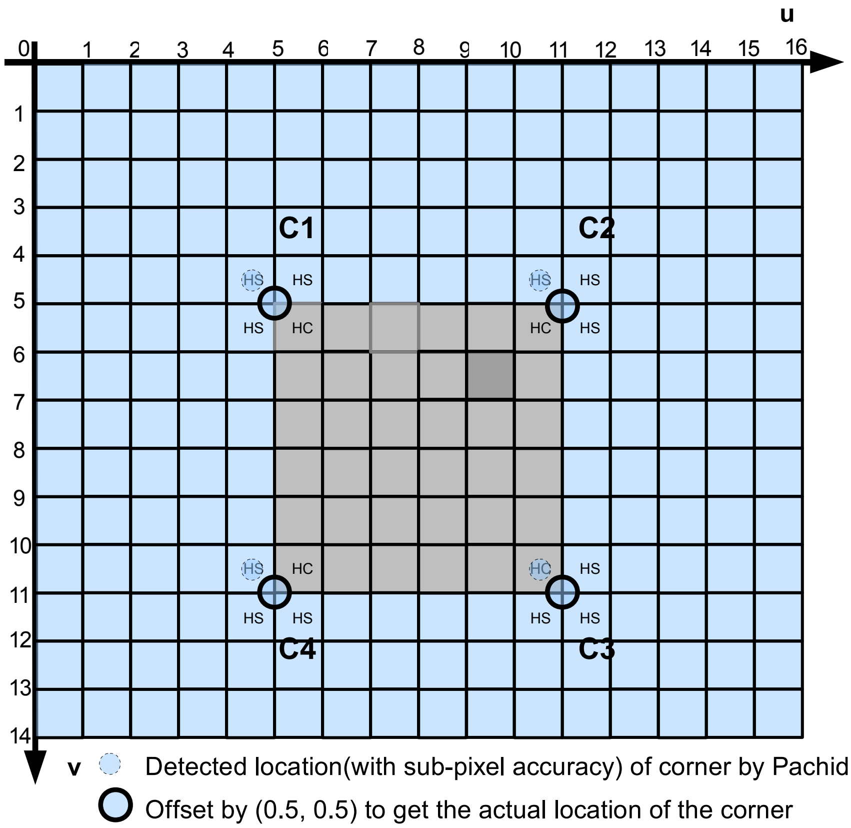


Figure 4.15 - shows the Correlation coefficient response map produced by the cross shape templates on a L-corner.

'HS' and 'HC' indicate the pixels with high response to template Template$_{CenterS}$ and Template$_{CenterC}$ respectively.

The planar object used for camera calibration was a checker board made up of black squares on a white object. For the camera calibration process, one needs to detect the locations of the corners of the black squares in the camera image. Using Pachidis template matching method [55], described in Section 4.3.2.1, one should be able to detect the L-corners in the binary image of the planar object. But during the image capturing process the camera records the binary source image as a colour image. The resulting scene image of the checker board is no more a binary image. But Pachidis template matching method [55], can only handle binary images. Faucher [6] had extended Pachidis template matching method to detect L-corners, with up to sub-pixel accuracy, in colour images. Faucher [6] integrated his method into the Mimas vision tool kit [4].

For this study the author used Faucher's [6] method through the Mimas vision tool kit [4] to test its capability to detect the four L-corners of each black coloured square. In Faucher [6] method instead of using the standard binary grey scaled values, *i.e.* 255 for white and 0 for black, the actual RGB values of the pixels in the white background and in the black squares were used. To get the actual RGB values of the pixels in the camera image, the user was given a graphical user interface facility to choose representative locations to get the actual intensity (RGB) values of the background and the black square. These RGB values were used as cell values to form the template. A sample template formed using the RGB values of the pixels to represent the “white” background and the “Black” square is shown in Figure 4.16. To detect the corners of type C2, C3 and C4 (Figure 4.13), the template was rotated by 90, 180, and 270 degrees respectively (Figure 4.16b,c,d). The Blais and Rioux sub-pixel estimator (Section 3.4.2.2.2) was subsequently used by Faucher to refine the result of the corner detection. The sub-pixel estimator was centred at the pixel location with the highest local correlation value, and the neighbour pixels along the image horizontal and vertical axis, were used to estimate the sub-pixel location along the horizontal and vertical axis, respectively.

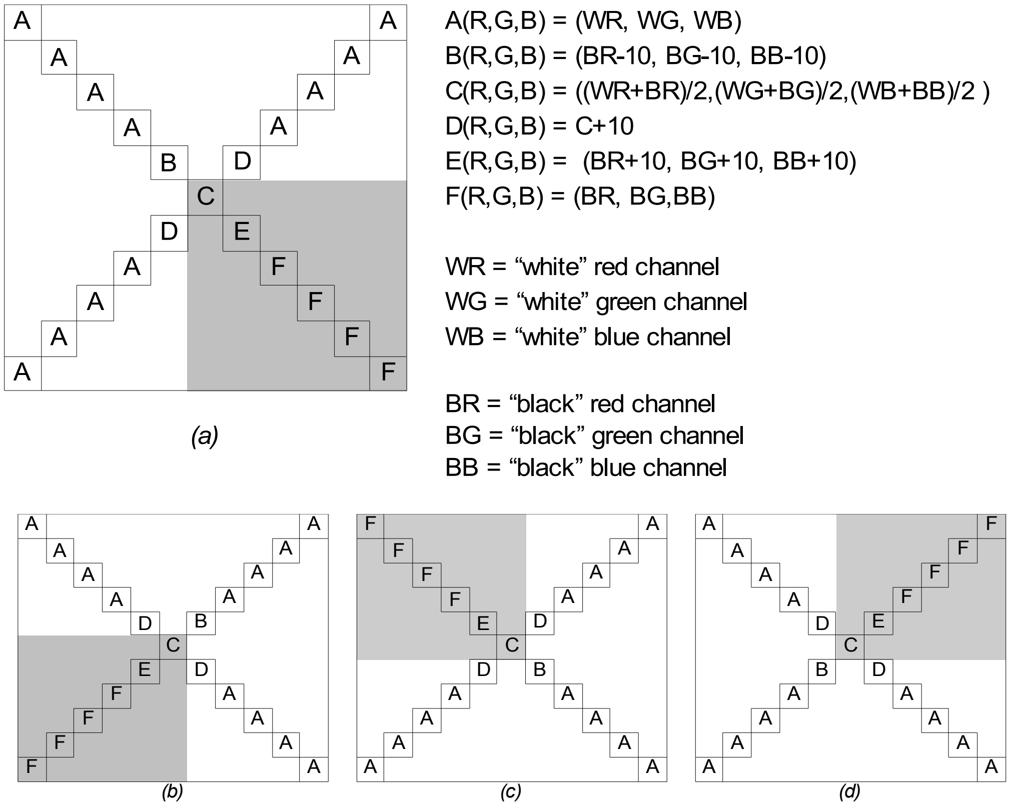
The way Faucher estimated the sub-pixel is directional dependant. A directional independent way of estimating the sub-pixel location, the quadratic surface fitting based sub-pixel detector was adapted. The surface fitting operation, estimates the peak of the surface made up of the correlation measures obtained through template matching.

This operation estimates the actual location of the corner much more precisely than just using the corner location obtained by only template matching based corner detection.

The Pachidis' template-based corner detector, produces high response values distributed around the actual location of the corner, when the template is positioned centred at the actual corner. In contrast, the template detector designed by Faucher creates a high response at the internal of the L-corner (see Figure 4.17). The subsequent quadratic surface fitting, making use of the correlation measures, does not change the location of the corner. Hence, the final estimation of the corner location, making use of Faucher template, has a larger displacement from the actual corner location when compared to H&S corner detector.

It was found that the Faucher's set of templates could not detect the corners when they were presented in a rotated fashion (see Figure 4.18). The designed templates were rotated by 45 degree (see Figure 4.19) but this did not work well due to aliasing. Trying to design templates to match the rotated corners did not produce satisfactory results.

In conclusion, for camera calibration, if the checker board is presented normal to the camera axis, and if the checker board corners are not subjected to large perspective distortion, then the template matching based corner detection provides satisfactory results.



A(R,G,B) = (WR, WG, WB)
B(R,G,B) = (BR-10, BG-10, BB-10)
C(R,G,B) = ((WR+BR)/2,(WG+BG)/2,(WB+BB)/2)
D(R,G,B) = C+10
E(R,G,B) = (BR+10, BG+10, BB+10)
F(R,G,B) = (BR, BG,BB)

WR = “white” red channel
WG = “white” green channel
WB = “white” blue channel

BR = “black” red channel
BG = “black” green channel
BB = “black” blue channel

(a) *(b)* *(c)* *(d)*

Figure 4.16: Figure illustrating the colour(R,G,B) distribution (Top right) for each pixel within the template. (a) Template to detect corner type C1. (b) Template C1 was rotated by 90. to detect corner type C2. (c) Rotated by 180 to detect corner type C3. (d) Rotated by 270 degrees to detect corner type C4.

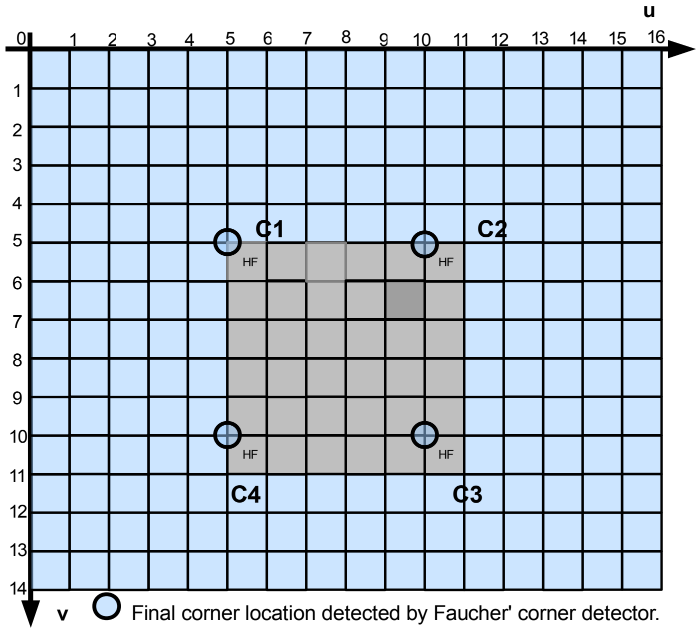


Figure 4.17: Correlation coefficient response map produced by Faucher' cross shaped templates on the L-corners.

'HF' indicates the pixel with the highest response when convolved with the designed templates.

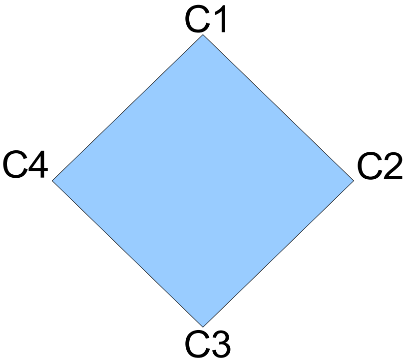


Figure 4.18: Figure illustrating a L-corner rotated by 45 degrees.

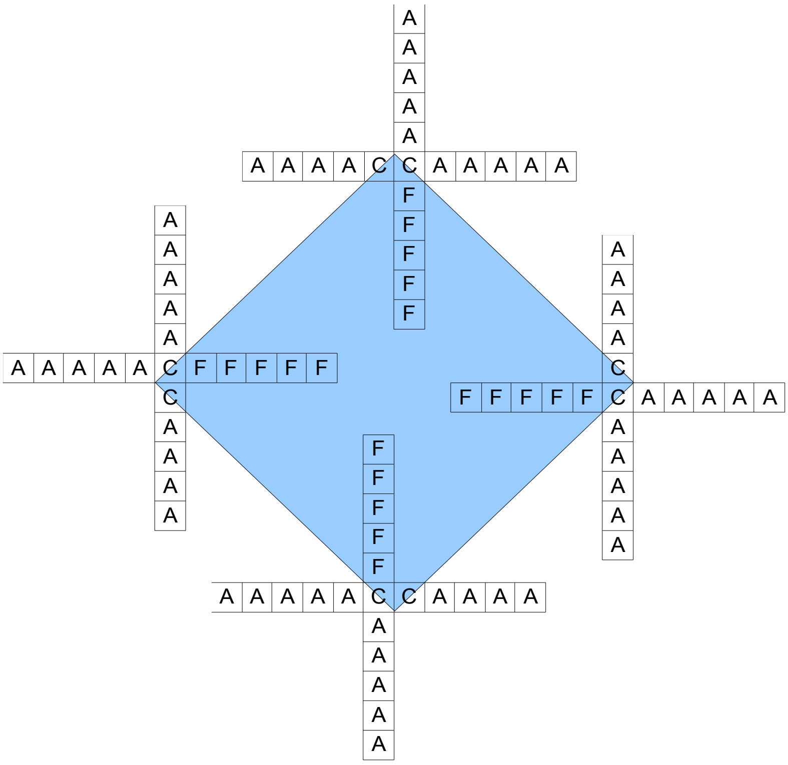


Figure 4.19: Simply rotating the template by 45 degree was not able to match the corner location due to aliasing issue.

4.3.2.2 Detecting the X-corners

As detailed in section 4.2.1, there is another type of corners referred to as X corners. In this study, a cross shaped template was again used to design an X-corner detector. To detect all the four types of X corners (see Figure 4.20), just a pair of templates were needed.

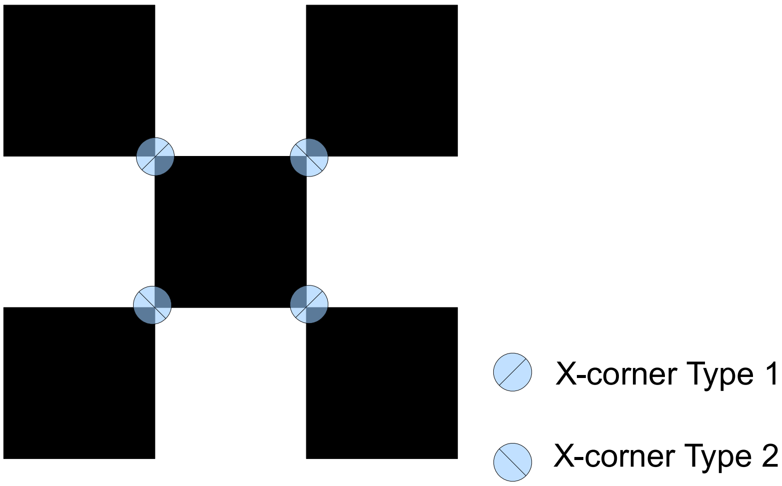


Figure 4.20: Two difference types of X-corner.

The pair of templates (T21(0), T21(1)) are indicated in Figure 4.21. They needed to detect the four types of X corners superimposed on the top of the X corner types C1 and C2. The X-corner of type C3 and C4 have the same properties as the X-corners C1 and C2 respectively. Hence to detect the X corner of types C3 and C4, the same pair of templates used for the corner types C1 and C2 respectively were found to be adequate.

To detect X-corners when the checker board was rotated by forty five degrees (see Figure 4.22), the templates T21(0) was rotated clockwise by forty five degrees to generate a corner template that generated a high response to the rotated X-corner. The rotated template T21(2), to detect the rotated X corner is shown in the Figure 4.22. To generate the rotated version of the template T21(1), the pixel values of 'F' and 'A', in the template T21(2), were interchanged. The rotated version of the template T21(1) is template T21(3) is shown in Figure 4.22.

When the X-corner was rotated by 45 degrees, the corners might not span through the entire pixel (see Figure 4.23). Hence the pair of templates T21(2) and T21(3) needed to be modified such that the templates created a high cornerness response.

The template T21(2) was modified as described next.

The three pixels at the centre of the template T21(2), were given a value 'C', which was the average value of 'A' and 'F' (see Figure 4.23). The modified template is T21c(2).

The modified version of the template T21(3), *i.e.* T21c(3), was generated by interchanging the pixel values of 'F' and 'A', in the template T21c(2).

The response measure, when the template was centred at the corner location, could be increased by convolving the template with more of the pixels around the corner. This was achieved by incorporating more pixels around the centre pixel of the template. Figure 4.24 illustrates this issue. In Figure 4.24 two corner templates, namely T25(0) (to detect corner C1) and T25(1) (to detect corner C2) were formed by adding four more pixels around the centre pixel in the templates T21(0) and T21(1) respectively. The created templates, T25(0) and T25(1), were rotated by 45 degree (see Figure 4.25) to create T25(2) and T25(3), respectively. Using T25(0) and T25(1), along with a similar modification to the template set T21c(2) and T21c(3), a new set of templates T25c (see Figure 4.26) was created.

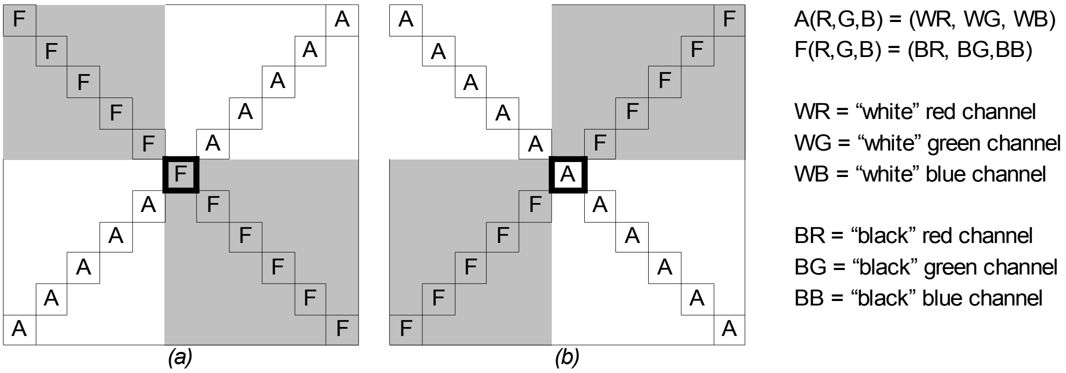
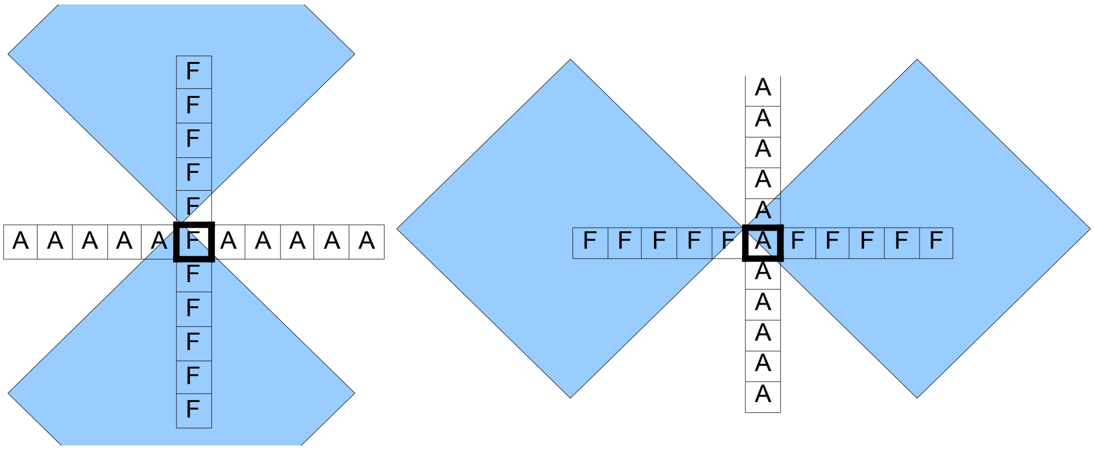


Figure 4.21: Designed templates (a) (T21(0) and (b) T21(1) to create high response on X-corners. The pixel location with a high response is highlighted with a bold box.

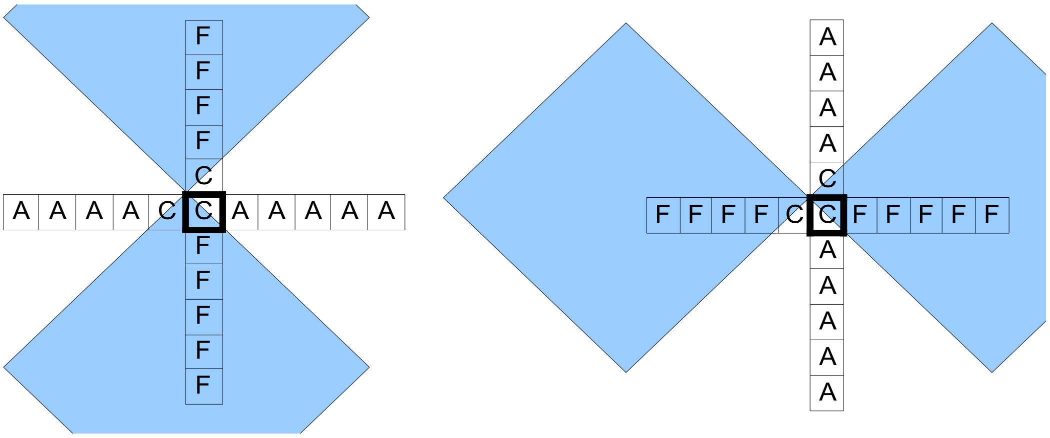


(a) *(b)*

Figure 4.22: (a) The template (T21(2)) designed to create high response on a rotated X-corner. The template (T21(2)) is got by rotating the template T21(0) (Figure 4.21a) by 45 degree clockwise. The pixel with high response is highlighted with a bold box.

(b) The pixel location of 'F' and 'A' were interchanged to generate template T21(3) to detect the same corner type with 90 degrees rotation.

The symbols used to indicate the template pixel values were the same as stated in Figure 4.21.

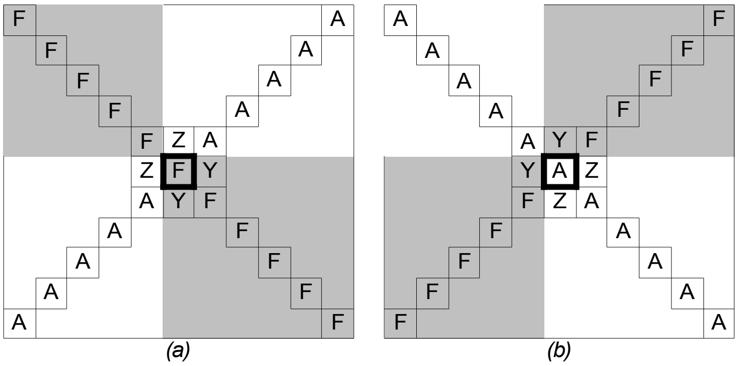


(a) *(b)*

Figure 4.23: (a) The template (T21c(2)) designed to create high response on a rotated X-corner. The template (T21c(2)) is got by replacing three templates values of the template T21(2) (Figure 4.22a). The template pixel value at locations marked by 'C' are the average value of the template pixel values at locations 'A' and 'F'. The pixel with high response is highlighted with a bold box.

(b) The pixel location of 'F' and 'A' were interchanged to generate template T21c(3) to detect the same corner type with 90 degrees rotation.

The symbols used to indicate the template pixel values were the same as stated in Figure 4.21.



A(R,G,B) = (WR, WG, WB)
F(R,G,B) = (BR, BG,BB)
Y(R,G,B) = (BR, BG, BB)
Z(R,G,B) = (WR, WG, WB)

WR = "white" red channel
WG = "white" green channel
WB = "white" blue channel

BR = "black" red channel
BG = "black" green channel
BB = "black" blue channel

Figure 4.24:

Figure illustrating the designed templates (a) T25(0) and (b) T25(1) to create high response on a X-corner. These templates are almost the same as the two templates T21(0), T21(1) (Figure 4.21a,b) respectively except that an addition set of five pixel values ("Z" and "Y") were added around the central pixel of the template. This modification was done to increase the cornerness response for the template.

The pixel location with the highest response is highlighted with a bold box.

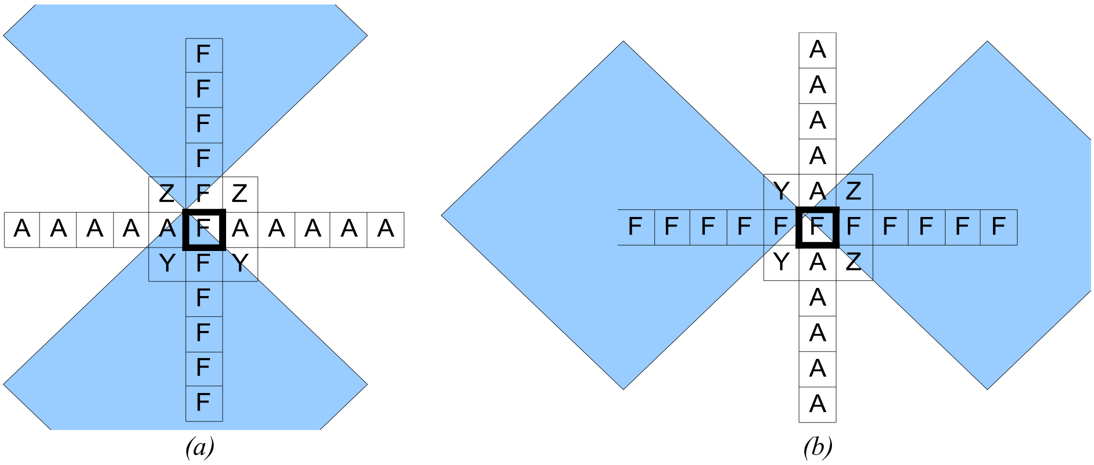


Figure 4.25: (a) The template (T25(2)) designed to create high response on a rotated X-corner. The template (T25(2)) is got by rotating the template T25(0) (Figure 4.24a) by 45 degree clockwise. The pixel with high response is highlighted with a bold box.

(b) The pixel location of 'F' and 'A' were interchanged to generate template T25(3) to detect the same corner type with 90 degrees rotation.

The symbols used to indicate the template pixel values were the same as stated in Figure 4.21.

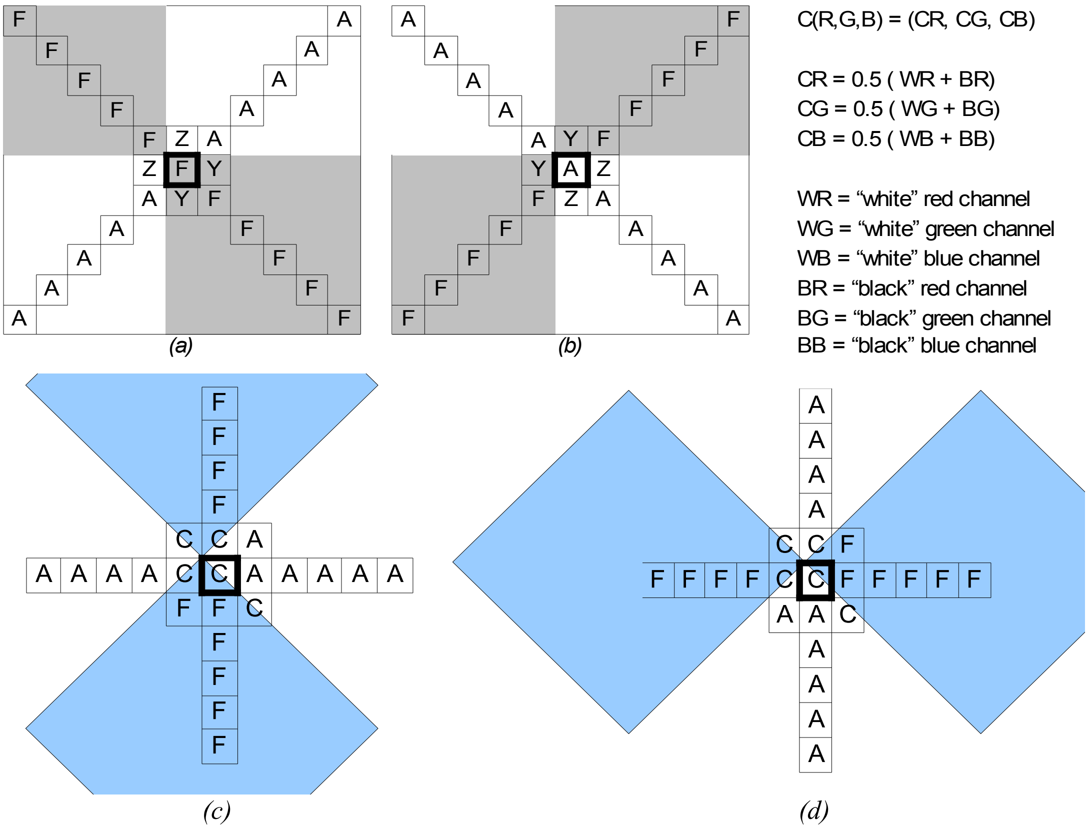


Figure 4.26: Templates in T25c.

(a) T25c(0) was created by using template T25(0).

(b) T25(1) was created by using template T25c(1).

(c) The template (T25c(2)) designed to create high response on a rotated X-corner. The template (T25c(2)) is got by using average RGB values 'C' as the template pixel value for the intersection of template and the corner. The template pixel value at locations marked by 'C' are the average value of the template pixel values at locations 'A' and 'F'. The pixel with high response is highlighted with a bold box.

(d) The pixel location of 'F' and 'A' were interchanged to generate template T25c(3) to detect the same corner type with 90 degrees rotation.

The symbols used to indicate the template pixel values were the same as stated in Figure 4.21.

4.3.2.3 Refining the result of the template based corner detector with sub-pixel operator

Two of the previously discussed sub-pixel operators, namely quadratic fit with least squares estimator (Section 4.2.3.1) and weighted least squares estimator (Section 4.2.3.2) were used as the sub-pixel operator to refine the result of corner detection to sub-pixel accuracy. A window of correlation coefficient values (Equation (4.20)), produced during the process of convolving the template pixels with the image pixels, centred at the highest correlation coefficient values, was used to fit a quadratic surface.

The process of refining the corner detection up to sub-pixel accuracy was achieved by estimating the peak of the quadratic surface as the final result of the corner detection.

4.3.3: Model-based feature detector

Model-based feature detectors, detect salient features, like corners, in an image making use of a model of the feature that needs to be detected. For example to detect corner features in an image, the feature models that could be used are L-corner/L-junction model or X-corner/X-junction model or Y-junction model *etc*., depending upon whether one needs to detect L-corner, X-corner or Y-corner respectively in the image.

The process of detecting features in an image, making use of feature models, starts with forming the model. The feature model is a two dimensional array of intensity values. The intensity values which form the feature model could be extracted from a sample image of the feature that needs to be detected subsequently. For example, Figure 4.27a and Figure 4.29a illustrate the intensity values of the 21x21 size feature models for the L-corner and the X-corner respectively. The pixels values of the feature models were extracted from the actual image intensity values of the L-corner and the X-corner values (Figure 4.27b and Figure 4.29b respectively). The 3-D plotting of the L-corner and X-corner are shown in Figure 4.28 and Figure 4.30 respectively.

The size of the feature model is chosen according to the noise level of the image, the larger the noise level in the image larger will be the size of the feature model.

In this section, firstly different model-based corner detectors are discussed. Next the salient features of the model based detectors uniquely designed to detect L-corner and X-corner, formed by the checker board pattern, are discussed. Lastly the results, demonstrating the efficiency of the model based corner detectors are discussed.

4.3.3.1 Past related work on the model-based L-corner detector

Different model-based feature detectors had been implemented to locate corners in a digital image. Deriche [56] proposed an efficient, model-based, approach to precisely locate the location of edges, corners, and vertices. His proposal uses a non-linear minimisation process to efficiently fit a model to the local image grey level intensity values. He has discussed the relevant issues, *e.g.* different type of models and their efficiency and the size of the fitting window with respect to the noise level in the image. Olague [57] proposed a revolutionary algorithm that used affine transformation to automatically detect L-corners in an image. Olague [57] proposed method does not need any guessing of the initial set of parameter values for the optimization process. His algorithm, used a Unit Edge Function Model and a L-corner model. It was tested on both synthetic and real images and good results were reported. Olague [58] proposed another model based L-corner detector. His method uses a Unit Step Edge function model, a distribution function model, optical and physical characteristics.

Due to the complexity of the algorithms (*i.e.* their requirement for non-linear fitting and their possible non convergence) and lack of availability of their original source codes, the above discussed L-corner model-based detectors were not implemented in this study.

4.3.3.2 Past related work on the model-based X-corner detector

Wedekind [59] proposed steerable filters to detect the X-corner by making use of hyper complex dual tree wavelet transform. Muhlich and Aach [60] designed multi-steerable filters to locate the X-corner and further refined the results up to sub-pixel accuracy by using paraboloid fitting. The orientation of the filter was estimated with high accuracy by using non-linear Lavenberg-Marquard optimization. The multi-steerable filters gave three times lower error as compared with the corner detector implemented by Bouguet [46].

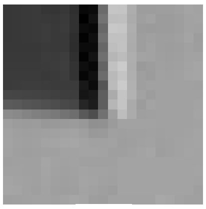
The surface profile formed by the image intensities values around a X-corner can be represented by a hyperbolic paraboloid. The location of the corner, is the surface's saddle point. Hence to locate the corner one needs to first linearly fit the image intensities values to a quadratic function (equation 4.2). The saddle point, the critical point of the quadratic surface function is actually the intersection of two lines, defined by the partial derivative of the quadratic function with respect to the x and y axis, as shown in equation (4.21) [61].

$$\begin{aligned} 2C_1 x + C_2 y + C_4 &= 0 \\ 2C_3 y + C_2 x + C_5 &= 0 \end{aligned} \tag{4.21}$$

Before fitting a quadratic surface to the image intensities, Lucchese [61] suggested to use cubic interpolation method to create an interpolated values of the original image pixel values. A similar saddle point based X-corner detector was implemented by Bouguet [46] This is discussed in the next section.

55	55	55	55	56	57	58	59	59	70	91	140	159	162	163	162	159	161	164	165	159
56	56	56	56	56	57	58	59	60	69	89	137	156	159	160	159	160	161	162	164	160
57	57	57	57	56	57	58	59	61	68	85	132	155	158	159	158	160	160	161	161	161
58	58	58	58	56	57	58	59	61	67	83	128	156	159	160	159	160	160	160	159	162
54	57	60	60	58	60	59	57	60	66	80	125	152	160	163	160	165	164	163	162	162
53	56	57	57	57	59	60	58	59	64	77	121	153	161	164	160	162	161	160	159	161
56	58	59	57	57	61	62	61	59	62	73	115	154	162	165	162	161	160	159	158	163
47	49	49	46	45	48	51	51	59	61	70	110	155	163	166	162	162	161	160	159	166
8	0	11	3	11	3	14	6	24	21	38	89	152	159	163	163	160	163	164	163	164
11	18	8	15	11	19	8	16	11	25	28	78	144	153	161	163	160	162	163	161	158
107	115	104	112	103	111	100	108	85	94	94	113	145	154	161	162	160	162	161	159	157
197	190	200	193	190	182	193	185	187	171	182	159	155	160	162	159	160	161	160	157	162
196	191	198	193	194	188	203	196	203	196	196	189	162	159	158	159	163	163	163	163	162
173	171	174	172	171	174	173	175	177	178	170	171	162	160	159	161	163	163	163	163	162
157	159	156	158	158	163	154	159	159	162	156	159	162	160	161	163	163	163	163	163	162
157	162	155	160	162	162	157	156	161	157	163	160	162	161	162	165	163	163	163	163	162
158	156	156	158	158	159	160	161	161	161	161	161	161	160	159	158	163	163	163	163	168
158	156	156	158	157	158	159	160	161	161	161	161	161	161	160	160	163	163	163	163	166
160	158	158	160	158	159	160	161	161	161	161	161	161	161	162	162	163	163	163	163	162
164	162	162	164	161	162	163	164	161	161	161	161	161	162	163	164	163	163	163	163	160
162	163	161	157	158	162	165	165	160	161	163	164	162	164	164	163	163	164	162	159	160

(a)



(b)

Figure 4.27: (a) The image intensities value around a L-corner.

(b).The camera image of image intensities values shown in Figure 4.27(a).

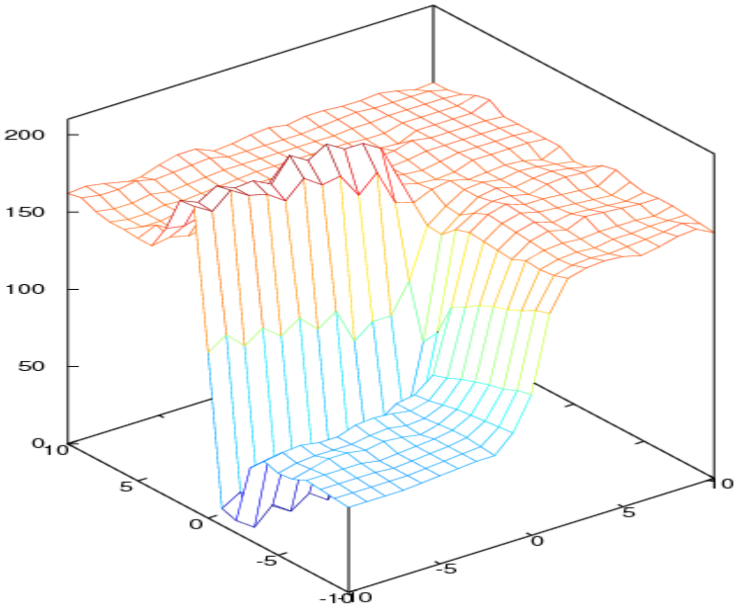
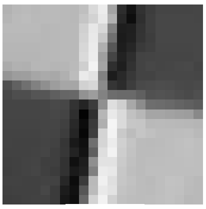


Figure 4.28: Mesh plot of the surface intensities in Figure 4.27a.

178	181	186	184	182	181	171	149	80	71	69	69	68	67	67	66	64	63	66	64	62
184	186	186	184	182	181	177	157	97	78	69	69	68	67	66	66	65	65	66	65	63
186	188	186	184	182	181	176	158	112	78	69	69	68	67	66	66	67	67	65	65	65
184	185	186	184	182	181	182	166	129	85	69	69	68	67	66	67	68	69	65	65	66
186	186	183	183	183	183	183	170	143	86	70	69	69	69	68	69	63	64	67	71	66
186	186	184	184	184	184	181	168	157	95	73	71	69	68	66	63	63	60	66	59	59
185	185	185	185	185	185	182	176	166	102	71	68	64	61	54	55	50	50	39	28	23
184	184	185	185	185	185	186	188	166	104	51	47	41	37	25	35	16	26	12	17	14
191	197	208	211	215	218	207	234	209	146	36	3	12	7	15	19	29	33	59	86	103
217	210	231	233	235	236	243	226	215	157	60	47	58	62	111	95	143	128	185	187	211
236	232	215	199	175	159	164	129	139	130	144	182	175	212	213	222	226	235	240	233	242
163	177	130	112	87	69	47	28	53	109	222	233	245	243	236	242	226	232	223	226	209
45	39	29	27	29	34	30	41	44	106	207	218	218	201	208	205	202	199	198	195	192
19	13	31	32	38	46	50	56	61	106	168	193	198	188	196	194	192	190	193	191	189
50	47	51	52	58	67	67	67	73	96	140	183	197	197	187	187	187	187	188	188	188
61	63	63	61	63	68	71	67	75	81	110	165	185	192	186	187	189	190	186	187	189
69	71	71	71	70	69	69	71	77	83	108	165	186	188	187	188	189	190	190	190	191
69	71	71	71	70	69	70	71	76	81	105	162	185	188	189	189	190	190	194	193	193
69	71	71	71	70	69	72	71	73	77	100	159	184	189	191	191	190	190	193	192	190
69	71	71	71	70	69	73	71	72	74	96	157	183	190	193	192	191	190	188	187	184
70	70	71	71	71	71	68	70	74	79	91	137	181	185	190	190	190	190	185	185	185

(a)



(b)

Figure 4.29: (a) The image intensities value around a X-corner.

(b) The camera image of image intensities values shown in Figure 4.29(a).

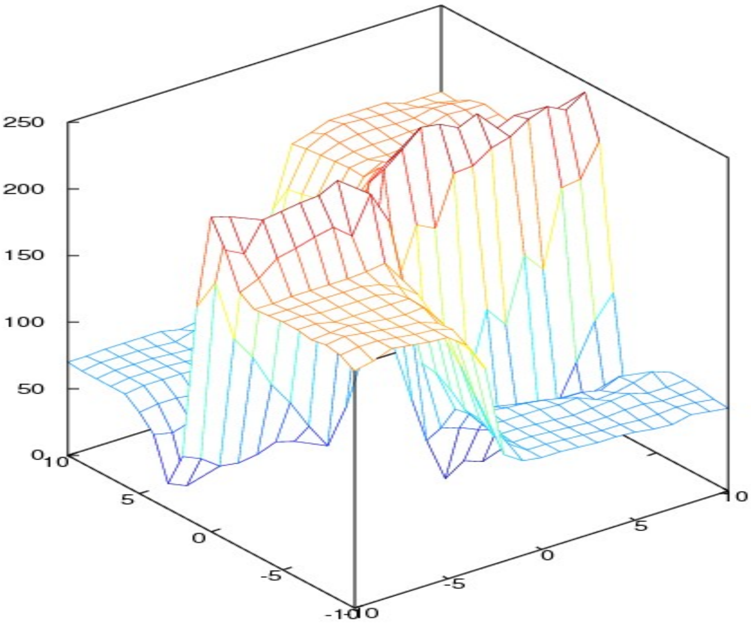


Figure 4.30: Mesh plot of the surface intensities in Figure 4.29a..

4.3.3.2.1 Saddle point based X-corner detector

The major operations of a saddle point based X-corner detector implemented by Bouguet [46] are to first detect the location of the corner approximately *i.e.* coarse corner detection and then refining the coarse location of the corner.

During the coarse corner detection, the approximate location of the X-corner was estimated by making use of one of the corner detectors. The robust corner detector, H&S was chosen for this purpose. This was because it was found that the error by H&S in locating this type of corner was usually less than two pixels.

The refining of the estimated coarse location of the corner is shown in Figure 4.31 and discussed in detail in the following section.

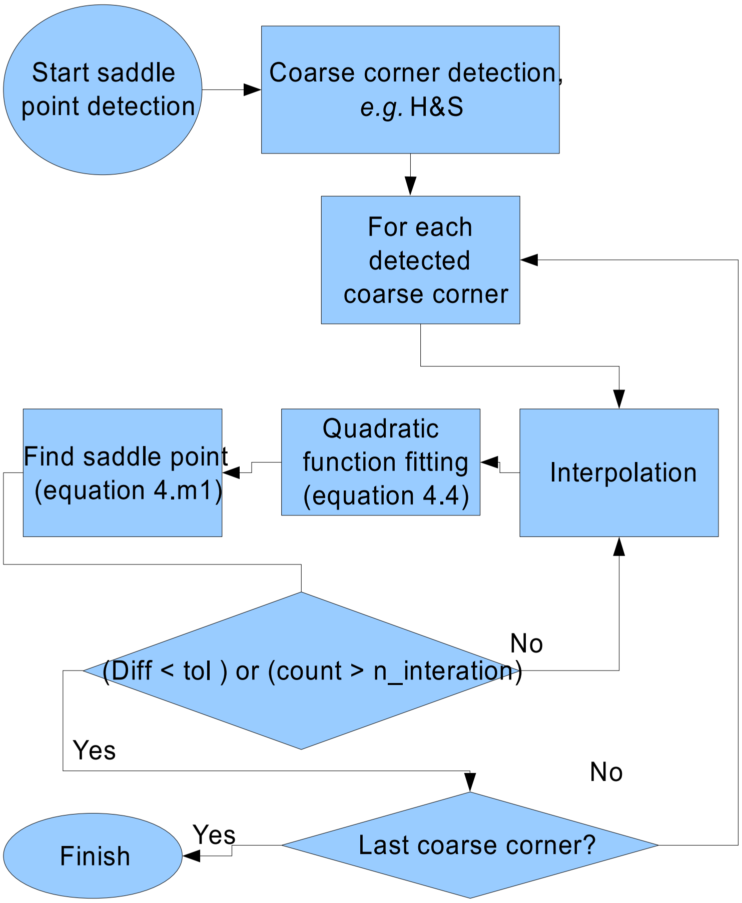


Figure 4.31:Flow chart illustrating the steps involved in the saddle point based corner detection process.

4.3.3.2.1.1 Refining of the estimated coarse corner location

The process of refining the located corners is explained in this section.

The refinement of the initial coarse corner location was performed by repeatedly generating a new set of interpolated image pixel values, around the coarse corner location. This was achieved by correlating the original pixel values with the interpolation mask (for details see below).

The estimated coarse corner location was used to initialise the interpolation method,. For subsequent refinements, the just refined corner location was input to the method. The output of the method was the refined corner location with up to sub-pixel accuracy. The operation of interpolation and thus the refinement of the input value was performed, as described next.

Let the input value of the corner location be (*cIx*, *cIy*,).

Let *rup_cIx* and *rup_cIy* be the rounded up values of *cIx*, *cIy* respectively.

Let *rcIx* be the difference between *cIx* and *rup_cIx* and *rcIy* be the difference between *cIy* and *rup_cIy*.

1) Use correlation process to get the interpolated pixel values of the image

First, two correlation masks, **vIx** and **vIy**, were constructed, using *rcIx, rcIy* respectively, as shown in Figure 4.32. The values in the correlation mask, summed up to one, provided the information of the weight of each source pixel to the result of the interpolated. The process of correlation, sum of multiply of the weight with the corresponding source image pixel value, produced the result of interpolation. The process of interpolation required the use of these masks to correlate with the source image pixel values, within a window centred at the image location (*rup_cIx*, *rup_cIy*).

The end of the above correlation process was a two dimensional array of interpolated pixel values. The size of the array was *wintx* by *winty*, and was centred at *rup_cIx*, *rup_cIy* (see Figure 4.33). This array of values was subsequently correlated with a Gaussian kernel to produce a continuous two dimensional function.

vlx

If (*rup_clx* > 0)

rclx	1-rclx	0

else

0	1+rclx	-1*rclx

vly

If (*rcly* > 0)

rcly
1-rcly
0

else

0
1+rcly
-1*rcly

Figure 4.32: Figure illustration the two correlation masks used for interpolation.

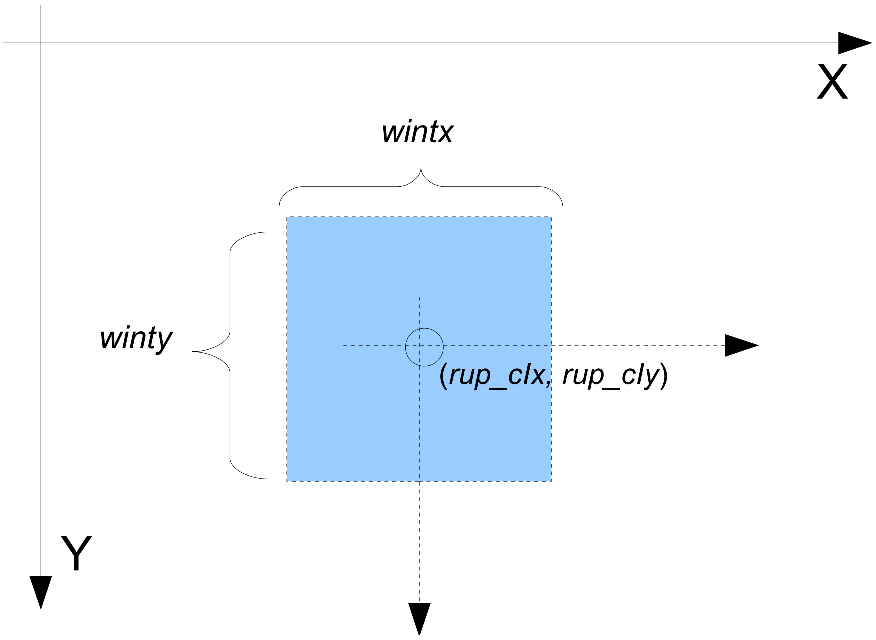


Figure 4.33: Figure illustrating the window location for the saddle point detection process.

2) Perform quadratic function fitting

The set of values obtained at the end of the correlation process (see above) were used to fit a quadratic function. The linear least squares estimation method (Section 4.2.3.1) was used to fit the quadratic function. This quadratic function fitting process produced a set of coefficients $(C_1, C_2, C_3, C_4, C_5, C_6)$ of the fitted quadratic function.

3) Estimate the saddle point

The saddle point (SP), *i.e.* the critical point of the quadratic surface function was the intersection of two lines, produced by the partial derivative of the quadratic function. The two lines are in fact the slope of the quadratic. This pair of lines is defined by the equation (4.21); The saddle point was estimated by making use of the set of coefficients obtained at the end of the quadratic function fitting process. This latest located saddle point value is given by the equation (4.24).

From 4.21

$$\begin{bmatrix} 2\mathrm{C}_1 & C_2 \\ C_2 & 2\mathrm{C}_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -d \\ -e \end{bmatrix} \tag{4.22}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2\mathrm{C}_1 & C_2 \\ C_2 & 2\mathrm{C}_3 \end{bmatrix}^{-1} \begin{bmatrix} -d \\ -e \end{bmatrix} \tag{4.23}$$

$$SP = (rup\_cIx + x\,,\, rup\_cIy + y) \tag{4.24}$$

3) Refine the process:

For each coarse corner location, the above three steps were, repeated at least ten times (*n_iteration* = 10) or until the Euclidean distance between the latest estimated saddle point and the earlier estimated saddle point was very small, *i.e.* less than the defined tolerant/threshold value (*tol*)

4.3.3.2.2 Chen and Zhang X-corner detector

Instead of using H&S corner strength to locate the initial coarse location. For any type of corner, Chen and Zhang [62] proposed to use the principle of the Hessian matrix (4.25) to locate the X-corner. The Hessian matrix is the matrix of second order partial derivatives of a function, in this case, the image function. The elements of the Hessian matrix, *i.e.* the second order derivative of the image, were found by correlating the image pixel values with the three by three mask shown in the figure 4.34.

1	-2	1
1	-2	1
1	-2	1

1	1	1
-2	-2	-2
1	1	1

Correlation mask to produce rxx ryy

Figure 4.34: The second order derivative mask used to correlate with the image.

For a two dimensional image at any location, there will be two eigenvectors. At the saddle point of a two dimensional image, one of the eigenvalue will be the maximum and the other the minimum. Hence, an X-corner point is located at the image pixel location where the Hessian matrix (4.25) has its the maximum eigenvalue λ_1 and minimum eigenvalue λ_2 . Hence, the location of the X-corner (x_0, y_0) can be located using the equation (4.26), where S is the local negative extrema.

$$H=\begin{bmatrix} r_{xx} & r_{xy} \\ r_{xy} & r_{yy} \end{bmatrix} \tag{4.25}$$

Where r_{xx}, r_{yy}, r_{xy} are the second order partial derivatives of the image function $r(x, y)$.

$$S=\lambda_1 \cdot \lambda_2 = r_{xx} r_{yy} - r_{xy}^{2} \tag{4.26}$$

$$\begin{aligned} s=subpx(x_0, y_0)&=\frac{r_y r_{xy} - r_x r_{yy}}{r_{xx} r_{yy} - r_{xy}^{2}} \\ t=subpy(x_0, y_0)&=\frac{r_x r_{xy} - r_y r_{xx}}{r_{xx} r_{yy} - r_{xy}^{2}} \end{aligned} \tag{4.27}$$

Where r_x, r_y are the first order partial derivatives of the image function $r(x, y)$.

For sub-pixel refinement, Chen and Zhang [62] proposed to use a second order polynomial equation to describe the local intensity profile at the coarse approximate location of the corner. It is assumed by Chen and Zhang that the sub-pixel location of the X-corner is located at (x_0+s, y_0+t) , where (s,t) is within a one pixel by one pixel window, centred at (x_0, y_0) . The saddle point of the local intensity profile can be extracted by the equation (4.27).

4.3.3.2.2.3 Sojka corner detector

Another model based corner detector was proposed by Sojka [63]. Sojka's corner detector was designed based on measuring the variance of the directions of the gradient of brightness or image intensity values. An L-corner model was introduced by Sojka and the location of the corner in the corner model was defined by two theorems. Sojka compared the results of his implementation against several famous corner detectors (Beaudet [64], Deriche-Giraudon [65], Harris-Stephen [52], Kitchen-Rosenfeld [66], Noble [67], SUSAN [68]). His approach was effective for not only on L-corners, but also on X,Y,T type corners. Sojka's corner detector was selected as one of the benchmarking candidate for evaluating the model-based corner detector. Sojka method had been compared against other corner detectors and had been shown to provide promising results. It implementation source code is also publicly available [69].

4.4 Performance analysis of the salient features detector

In this section the performance of the three different methods, in detecting and localising the salient features, is evaluated and discussed in detail. Also at the end, the performance of the three methods is compared with each other. The evaluation of the different methods was carried out by making use of both synthetic images as well as real world images, captured by the camera.

4.4.1 Evaluation of the image intensity variation based corner detector algorithm H&S corner detector

The performance of Harris & Stephen corner detector, which is a image intensity based, was evaluated and the results are shown in this section.

The H&S corner detection algorithm (Section 4.3.1.1) was applied to detect and localise the L-corners and X-corners in the pseudo image, shown in Figure 4.3a and Figure 4.3b, respectively. The results of the L-corner detection and localisation process with H&S corner detector, with least squares estimator (LSE) and weighted least squares estimator (WLSE) sub-pixel operator were listed in Table 4.1. The results of X-corner detection are listed in Table 4.2.

Table 4.1: Location of the detected L-corner in Figure 4.3 (left), Using H&S corner detection process

	C1(u,v)		C2(u,v)		C3(u,v)		C4(u,v)		Sum of square error
Actual location	5.00	5.00	11.00	5.00	11.00	11.00	5.00	11.00	
Detected by H&S (k=0.04)	5.00	5.00	10.00	5.00	10.00	10.00	5.00	10.00	
Error	0.00	0.00	-1.00	0.00	-1.00	-1.00	0.00	-1.00	4.00
With sub-pixel									
LSE	5.54	5.54	9.46	5.54	9.46	9.46	5.54	9.46	
Error	0.54	0.54	-1.54	0.54	-1.54	-1.54	0.54	-1.54	10.68
WLSE	5.49	4.51	9.51	4.51	9.51	10.49	5.49	10.49	
Error	0.49	-0.49	-1.49	-0.49	-1.49	-0.51	0.49	-0.51	5.95

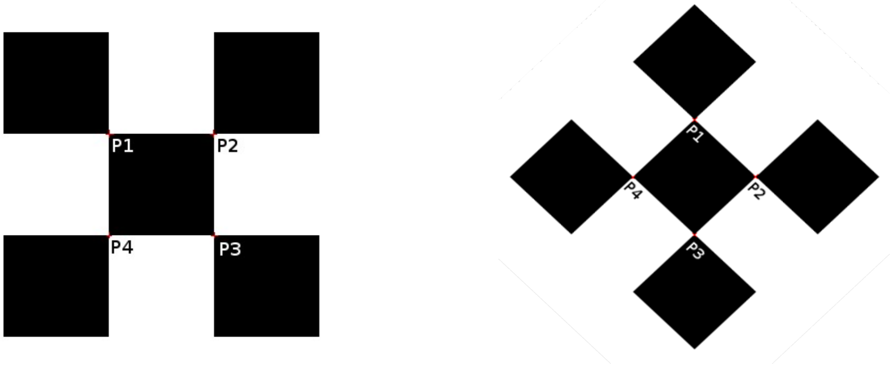
Table 4.2: Location of the detected X-corner in Figure 4.3 (right), Using H&S corner detection process

	C1(u,v)		C2(u,v)		C3(u,v)		C4(u,v)		Sum of square error
Actual location	5.00	5.00	11.00	5.00	11.00	11.00	5.00	11.00	
Detected by H&S (k=0.04)	5.00	5.00	10.00	5.00	10.00	10.00	5.00	10.00	
Error	0.00	0.00	-1.00	0.00	-1.00	-1.00	0.00	-1.00	4.00
With sub-pixel									
LSE	4.40	4.40	10.60	4.40	10.60	10.60	4.40	10.60	
Error	-0.60	-0.60	-0.40	-0.60	-0.40	-0.40	-0.60	-0.40	2.07
WLSE	4.41	5.59	10.59	5.59	10.59	9.41	4.41	9.41	
Error	-0.59	0.59	-0.41	0.59	-0.41	-1.59	-0.59	-1.59	6.75

The results in Table 4.1 show that H&S corner detector picks up the location of the L-corners with offsets at corner locations C2, C3, and C4. The results were plotted and shown in Figure 4.8. The performance of sub-pixel operators on the cornerness map produced by the H&S corner detector on the L-corner did not improve the result of detection.

The results in Tables 4.2 indicate that H&S corner detector picks up the location of the X-corners with offsets at corner locations C2, C3, and C4. The results were plotted and shown in Figure 4.10. The performance of the least squares sub-pixel operator on the cornerness map produced by the H&S corner detector on the X-corner improved the results of detection at X-corner locations C2, C3, and C4, but an error was introduced at corner location C1.

The performance of H&S corner detector was evaluated on a synthetic image of X-corners (see Figure 4.35) rotated by 15, 30, 45, 60, and 75 degrees (refer to appendix B.1 for the method of image generation). The inner angles of each of the black square was right angle and the length of the edge, *i.e.* (P1 to P2), was 100 pixel units. The result is shown in Tables 4.3 and 4.4. The performance of H&S corner detector on X-corner was found to suffer more than one pixel unit of error in detecting the corner location with least squares quadratic fitting based sub-pixel operator. More than two pixels unit of error where found with weighted least squares quadratic fitting based sub-pixel operator.



(a) *(b)*

Figure 4.35: Two synthetic images with checker board pattern(X-corner) were generated by using postscript. The accuracy of corner detector algorithms are tested using these images. Please find Appendix B.1 for the example script for using postscript to generate the image. The size of each black square is 100x100 pixels. (a) The generated image.

(b) Rotating Figure 4.35(a) by 45 degrees at the image centre.

Table 4.3: Error in detecting the location of the X-corner in Euclidean space (pixel unit), with H&S

Rotation	Corner			
0 degree	P1	P2	P3	P4
H&S LSE	0.72853	0.72853	0.72853	0.72853
H&S WLSE	1.70966	0.72130	1.70966	0.72130
15 degree				
H&S LSE	1.14960	1.51767	1.00209	0.14985
H&S WLSE	1.40859	2.19741	1.56232	0.98694
30 degree				
H&S LSE	1.38278	1.58516	0.88100	0.41894
H&S WLSE	1.94849	1.99544	0.58326	0.86981
45 degree				
H&S LSE	1.09592	1.19870	1.19870	1.09592
H&S WLSE	1.64732	2.41577	1.86868	2.40240
60 degree				
H&S LSE	0.41894	0.88100	1.58516	1.38278
H&S WLSE	1.29654	1.14126	1.56710	1.85361
75 degree				
H&S LSE	0.14985	1.00209	1.51767	1.14960

Table 4.4: The overall performance (sum of all the euclidean errors) of H&S corner detector

Overall	0 degree	15 degree	30 degree	45 degree	60 degree	75 degree	**total**
H&S LSE	2.91410	3.81921	4.26788	4.58923	4.26788	3.81921	23.67750

In subsequent sections, we will evaluate the other two types of algorithms to find out whether these results could be further improved.

4.4.2 Evaluation of the template based corner detection algorithms

The evaluation of the template based corner detection algorithms on the L-corner was not carried out because the designed corner template produced false cornerness responses (Section 4.3.2.1, Detecting the L-corner with cross shape template).

The template-based corner detector was applied on the synthetic images of X-corners (see Figure 4.35). Four different versions (indicated by the number 0 to 3) of the templates T21, T21c, T25 and T25c were applied to each of the corners P1 to P4 and the best correlated version was considered to estimate the location of the corner. Six different versions (indicated by the number 0 to 5) of template T25m were formed and applied to each of the corners P1 to P4 and the best correlated type was considered as to have estimated the location of the corner. The six versions were the first four, *i.e.* T25m(0 to 3) were exactly the same as T25c (0 to 3) and the last two, T25m(4 to 5) were T25(2) and T25(3).

The convolution of template T21, with the synthetic image, produced multiple responses with the generated synthetic image with rotations 0^o and 45^o. Figure 4.36 shows the produced multiple response.

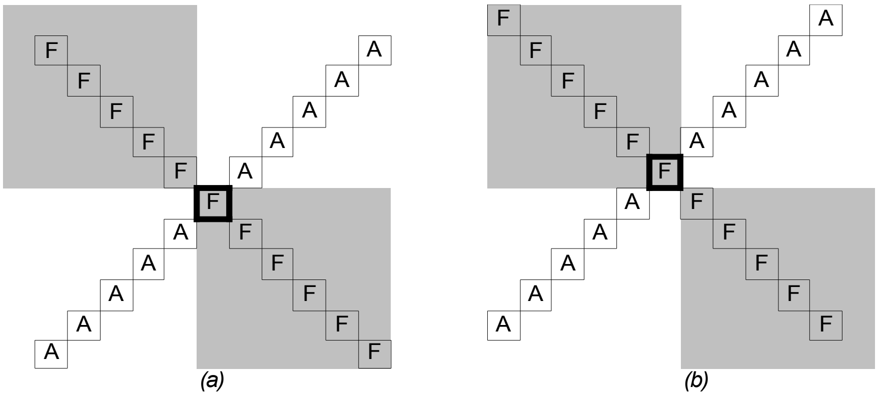


Figure 4.36: The multiple respond produced by T21(0) on the generated synthetic image with rotation 0^o. (a) The first respond produced by T21(0). (b) Another respond.

The convolution of the template T21c, with the synthetic image, produced multiple responses for the generated synthetic image rotated to 0^o.

The convolution of the template T25, with the synthetic image, produced multiple responses with the generated synthetic image having rotation 45^o.

The convolution of the template T25c and T25m, with the synthetic images, did not produce any multiple responses. Hence, the results obtained by the convolution of the template T25c and T25m, with the synthetic images, and the corresponding corner location are discussed in detail below.

The performance of the T25c and T25m templates based corner detector, using least squares estimator as the sub-pixel operator (Section 4.3.2.3), is shown in Tables 4.5 and 4.6. The first column of Table 4.5 indicates the different types of templates used, namely, T25c and T25m. The four numbers, 0 to 6, within the brackets, indicate the versions of the respective templates, which gave the highest correlation value corresponding to the four corners P1 to P4. For example T25c(0101), stands for the Template T25c versions T25c(0), T25c(1), T25c(0) and T25c(1) (see Figure 4.26) applied to the corners P1, P2, P3 and P4 respectively to get the best correlation. The overall performance is summarised in Table 4.6.

Table 4.5: Error of detecting the location of x-corner in Eulidean (pixel unit), with LSE

Rotation	Corner			
0 degree	P1	P2	P3	P4
T25c (0101)	0.03152	0.03152	0.03152	0.03152
T25m (0101)	0.03152	0.03152	0.03152	0.03152
15 degree				
T25c (0101)	0.38974	0.32138	0.29928	0.52535
T25m (0101)	0.38974	0.32138	0.29928	0.52535
30 degree				
T25c (2323)	0.38972	0.23256	0.27058	0.28245
T25m (2323)	0.38972	0.23256	0.27058	0.28245
45 degree				
T25c (2323)	0.07613	0.06177	0.06177	0.07613
T25m (2323)	0.07613	0.06177	0.06177	0.07613
60 degree				
T25c (2323)	0.28373	0.27058	0.25857	0.39039
T25m (2343)	0.28373	0.27058	0.70410	0.39039
75 degree				
T25c (1010)	0.52535	0.29928	0.32138	0.38974

Table 4.6: The overall performance(sum of all the euclidean errors) of T25c and T25m with LSE.

Overall	0 degree	15 degree	30 degree	45 degree	60 degree	75 degree	**total**
T25c	0.12609	1.53574	1.17531	0.27579	1.20327	1.53574	5.85194

The performance of the T25c and T25m templates based corner detector, using weighted least squares estimator as the sub-pixel operator (Section 4.3.2.3), is shown in Tables 4.7 and 4.8.

Table 4.7: Error in detecting the location of x-corner, in Eulidean space (pixel unit), with WLSE

Rotation	Corner			
0 degree	P1	P2	P3	P4
T25c (0101)	0.00389	0.00389	0.00389	0.00389
T25m (0101)	0.00389	0.00389	0.00389	0.00389
15 degree				
T25c (0101)	0.45488	0.62394	0.54463	1.15127
T25m (0101)	0.45488	0.62394	0.54463	1.15127
30 degree				
T25c (2323)	1.05676	0.78458	0.34047	0.35039
T25m (2323)	1.05676	0.78458	0.34047	0.35039
45 degree				
T25c (2323)	0.50477	0.05017	0.53354	0.10905
T25m (2323)	0.50477	0.05017	0.53354	0.10905
60 degree				
T25c (2323)	0.54093	1.27809	0.64383	0.51645
T25m (2343)	0.54093	1.27809	0.94626	0.51645
75 degree				
T25c (1010)	0.73221	0.44097	0.26433	0.99343

Table 4.8: The overall performance(sum of all the euclidean errors) of T25c and T25m with WLSE

Overall	0 degree	15 degree	30 degree	45 degree	60 degree	75 degree	**total**
T25c	0.01556	2.77472	2.53219	1.19752	2.97931	2.43095	11.93026

The error in estimating the length of the sides and the inner angles of the squares (Section 4.2.1), using the four detected corner locations, are shown in Table 4.9.

Table 4.9: Error in the estimated length and inner angle using the detected corner locations

	The error of measured length (pixel unit) of each side of the black square(x-corner).				The error of angle measurement (degree) at each corner of the black square(x-corner).			
Rotation	Length				Angle			
0 degree	L1	L2	L3	L4	P1	P2	P3	P4
T25c (0101)	0.00000	0.00000	0.00000	0.00000	-0.00070	-0.28347	-0.00070	0.28488
T25c (0101)w	0.00000	0.00000	0.00000	0.00000	-1.37140	1.14058	1.38156	-1.15074
T25m (0101)	0.00000	0.00000	0.00000	0.00000	-0.00070	-0.28347	-0.00070	0.28488
T25m (0101)w	0.00000	0.00000	0.00000	0.00000	-1.37140	1.14058	1.38156	-1.15074
15 degree								
T25c (0101)	0.10292	-0.03815	0.16483	0.13551	0.00000	0.00000	0.00000	0.00000
T25c (0101)w	0.11498	-1.00377	-0.43143	0.74539	0.69524	-0.69524	0.69524	-0.69524
T25m (0101)	0.10292	-0.03815	0.16483	0.13551	0.00000	0.00000	0.00000	0.00000
T25m (0101)w	0.11498	-1.00377	-0.43143	0.74539	0.69524	-0.69524	0.69524	-0.69524
30 degree								
T25c (2323)	-0.02556	-0.00344	-0.07430	0.11546	0.00000	0.00000	0.00000	0.00000
T25c (2323)w	0.71852	-1.01503	0.01993	0.66346	-0.66784	0.66784	-0.66784	0.66784
T25m (2323)	-0.02556	-0.00344	-0.07430	0.11546	0.00000	0.00000	0.00000	0.00000
T25m (2323)w	0.71852	-1.01503	0.01993	0.66346	-0.66784	0.66784	-0.66784	0.66784
45 degree								
T25c (2323)	0.10723	-0.01445	0.10723	0.12153	0.42785	-0.42785	-0.42785	0.42785
T25c (2323)w	0.31808	0.40986	0.50152	0.40934	1.81493	-1.22918	-1.21008	0.62433
T25m (2323)	0.10723	-0.01445	0.10723	0.12153	0.42785	-0.42785	-0.42785	0.42785
T25m (2323)w	0.31808	0.40986	0.50152	0.40934	1.81493	-1.22918	-1.21008	0.62433
60 degree								
T25c (2323)	-0.07186	0.01978	-0.03527	0.11523	0.00000	0.00000	0.00000	0.00000
T25c (2323)w	0.66924	-1.00809	-0.75507	0.02479	-0.66784	0.66784	-0.66784	0.66784
T25m (2343)	-0.07186	0.36905	-0.32075	0.11523	0.00000	0.00000	0.00000	0.00000
T25m (2343)w	0.66924	-0.55370	-1.02246	0.02479	-0.66784	0.66784	-0.66784	0.66784
75 degree								
T25c (1010)	0.16483	-0.03815	0.10292	0.13551	0.00000	0.00000	0.00000	0.00000
T25c (1010)w	-1.15790	-0.07915	0.73364	-0.40086	0.69524	-0.69524	0.69524	-0.69524
T25m (1010)	0.16483	-0.03815	0.10292	0.13551	0.00000	0.00000	0.00000	0.00000
T25m (1010)w	-1.15790	-0.07915	0.73364	-0.40086	0.69524	-0.69524	0.69524	-0.69524

Anotation:

T25c (0101)w. The 'w' at the end indicating the result produced with weighted least squares estimator.

From the Tables 4.6 and 4.8, the following conclusions could be inferred:

The template T25m, obtained by combining templates, T25c and T25 did not yield the best result due to the false template giving the highest correlation measurement within the correlation window. For example, while detecting the corner rotated to 60 degrees, corner P3, template T25m(4) (equivalent to T25(2)) gave the highest correlation value compared to T25c(2) even though the result of detection using T25c(2) is closer to the ground truth. The false template affected measurement of the length and angle. Hence, Template T25c was found to perform better than T25m.

The total error produced by the template T25c with LSE (5.85914) was found to be smaller as compared with WLSE (11.93026). Hence, quadratic fit with the least squares estimator based sub-pixel operator was found to provide better refined results when compared to the weighted least squares estimator.

The overall result of detecting X-corners using template-based approach (T25c) was not found to be sufficiently accurate. This X-corner detection algorithm suffers around half pixel unit error (Table 4.3, 0.52535) in detecting the location of the corner (Table 4.3, 0.52535). The error in the measured length along each of the edges is around 0.16 pixels unit (Table 4.7 left) and the measured inner angles of the square at each corner (Table 4.7 right) also suffers around half a degree error.

4.4.3 Evaluation of model based corner detection algorithms

Three model based X-corner detection algorithms namely, saddle point X-corner detection algorithm (Section 4.3.3.2.1), Chen and Zhang's X-corner detection algorithms (Section 4.3.3.2.2) and Sojka corner detection algorithm were evaluated and the corresponding results are discussed in the following section.

4.4.3.1 Performance of the Saddle point X-corner detector

The Saddle point based X-corner detector, discussed in Section 4.3.3.2.11, was implemented. A quadratic fitting window of size of 15x15 used in Bouguet' implementation [46] was adapted. Due to the requirement of large sized fitting window, the implemented corner detection algorithm was directly evaluated on the series of generated synthetic images (Figure 4.35). The results of the corner detection is shown in Table 4.10 and Table 4.11.

Table 4.10: Error in detecting X-corner location (pixel unit) using Saddle point based corner detector

Rotation (degree)	Corner							
	P1(x,y)		P2(x,y)		P3(x,y)		P4(x,y)	
0	-0.00002	-0.00002	-0.00002	-0.00002	-0.00002	-0.00002	-0.00002	-0.00002
15	0.00181	0.01059	-0.01066	0.00177	-0.00179	-0.01064	0.01059	-0.00189
30	0.00922	0.01037	-0.01036	0.00922	-0.00923	-0.01035	0.01037	-0.00922
45	0.00003	-0.06039	0.06040	0.00005	-0.00005	0.06040	-0.06039	-0.00007
60	-0.00913	0.01039	-0.01038	-0.00913	0.00913	-0.01039	0.01039	0.00913
75	-0.00181	0.01063	-0.01068	-0.00171	0.00168	-0.01070	0.01064	0.00173

Table 4.11: Error in the estimated length and the inner angle estimated by Saddle point based corner detector

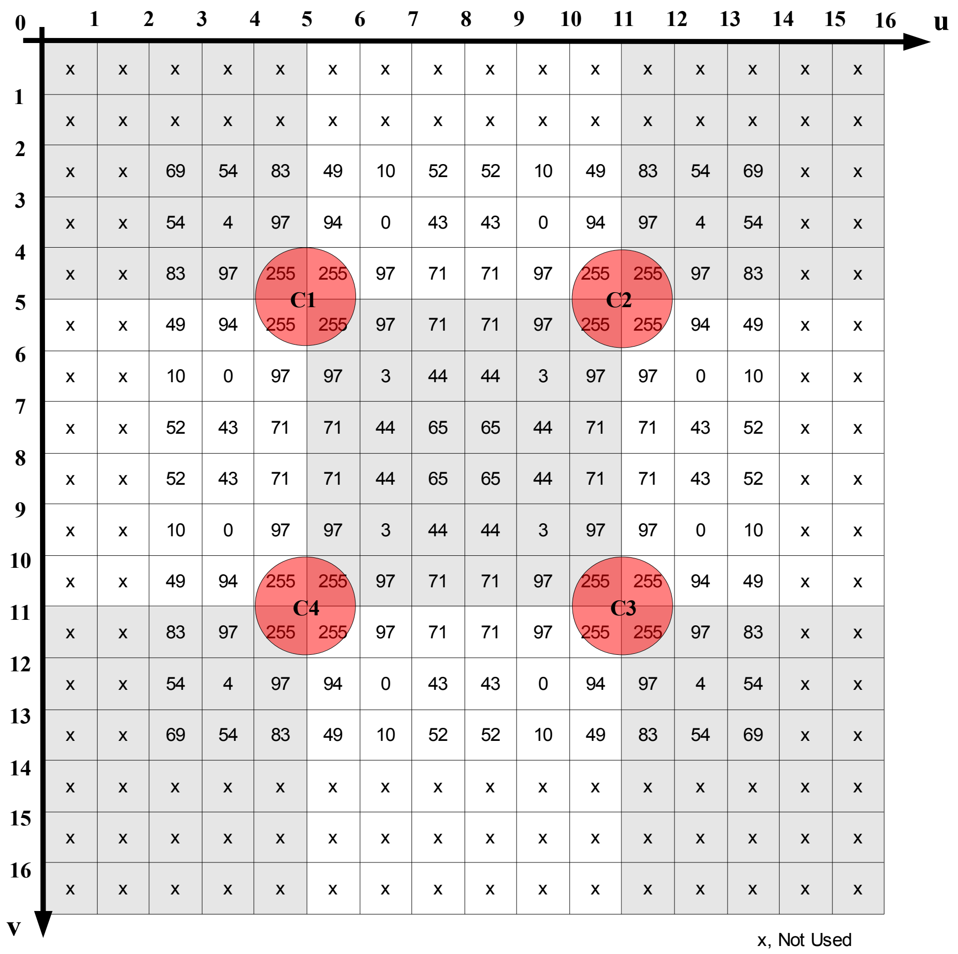
	The error in the estimated length (pixel unit) of each side of the black square(x-corner).				The error in the angle estimated (degree) at each corner of the black square(x-corner).			
Rotation (degree)	Length				Angle			
	L1	L2	L3	L4	P1	P2	P3	P4
0	0.00000	0.00000	0.00000	0.00000	-0.00008	-0.00008	-0.00008	-0.00008
15	0.01432	0.01427	0.01422	0.01433	-0.00011	-0.00010	-0.00001	-0.00008
30	0.01753	0.01752	0.01753	0.01753	-0.00007	-0.00007	-0.00007	-0.00008
45	-0.08543	-0.08542	-0.08543	-0.08538	-0.00009	-0.00007	-0.00007	-0.00009
60	0.01753	0.01752	0.01753	0.01753	-0.00008	-0.00007	-0.00007	-0.00007
75	0.01422	0.01427	0.01432	0.01433	-0.00008	-0.00001	-0.00010	-0.00011

The accuracy of the implemented Saddle point based algorithm was found to be very good. The maximum error of the detected corner locations was found to be only 0.06040 pixel unit.

4.4.3.2 Performance of Chen and Zhang' X-corner detector

The Chen and Zhang' X-corner detector, discussed in Section 4.3.3.2.2, was implemented. The implemented corner detection algorithm was subsequently used to detect and localise the X-corners in the 16x16 pseudo image as shown in Figure 4.3b. The results of the X-corner detection were plotted and is shown in Figure 4.37.

Figure 4.37 shows the multiple strong responses produced by Chen and Zhang' X-corner detector around the location of the X-corner. The result of the sub-pixel detector is shown in Table 4.12.



x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	69	54	83	49	10	52	52	10	49	83	54	69	x	x
x	x	54	4	97	94	0	43	43	0	94	97	4	54	x	x
x	x	83	97	255	255	97	71	71	97	255	255	97	83	x	x
x	x	49	94	255	255	97	71	71	97	255	255	94	49	x	x
x	x	10	0	97	97	3	44	44	3	97	97	0	10	x	x
x	x	52	43	71	71	44	65	65	44	71	71	43	52	x	x
x	x	52	43	71	71	44	65	65	44	71	71	43	52	x	x
x	x	10	0	97	97	3	44	44	3	97	97	0	10	x	x
x	x	49	94	255	255	97	71	71	97	255	255	94	49	x	x
x	x	83	97	255	255	97	71	71	97	255	255	97	83	x	x
x	x	54	4	97	94	0	43	43	0	94	97	4	54	x	x
x	x	69	54	83	49	10	52	52	10	49	83	54	69	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Figure 4.37: Cornerness map produced by Chen and Zhang's X-corner detector on the X-corner shown in Figure 4.3b. Each image pixel was filled with rounded normalised value from 255 (strongest) to 0 (weakest) cornerness, superimposed on the original image.

Table 4.12: Location of the detected X-corner in Figure 4.3b

	C1(u,v)		C2(u,v)		C3(u,v)		C4(u,v)	
Actual location	5.00000	5.00000	11.00000	5.00000	11.00000	11.00000	5.00000	11.00000
Detected by Chen	5.00000	5.00000	10.00000	5.00000	10.00000	10.00000	5.00000	10.00000
Error	0.00000	0.00000	-1.00000	0.00000	-1.00000	-1.00000	0.00000	-1.00000
With sub-pixel								
coordinate	(4,4)		(10,4)		(10,10)		(4,10)	
	4.38240	4.38240	10.38240	4.38240	10.38240	10.38240	4.38240	10.38240
Error	-0.61760	-0.61760	-0.61760	-0.61760	-0.61760	-0.61760	-0.61760	-0.61760
coordinate	(5,4)		(11,4)		(11,10)		(5,10)	
	4.61760	4.38240	10.61760	4.38240	10.61760	10.38240	4.61760	10.38240
Error	-0.38240	-0.61760	-0.38240	-0.61760	-0.38240	-0.61760	-0.38240	-0.61760
coordinate	(5,5)		(11,5)		(11,11)		(5,11)	
	4.61760	4.61760	10.61760	4.61760	10.61760	10.61760	4.61760	10.61760
Error	-0.38240	-0.38240	-0.38240	-0.38240	-0.38240	-0.38240	-0.38240	-0.38240
coordinate	(4,5)		(10,5)		(10,11)		(4,11)	
	4.38240	4.61760	10.38240	4.61760	10.38240	10.61760	4.38240	10.61760
Error	-0.61760	-0.38240	-0.61760	-0.38240	-0.61760	-0.38240	-0.61760	-0.38240
Averaging the result of sub-pixel	4.50000	4.50000	10.50000	4.50000	10.50000	10.50000	4.50000	10.50000
Error	-0.50000	-0.50000	-0.50000	-0.50000	-0.50000	-0.50000	-0.50000	-0.50000

From the errors listed in Table 4.12 one could see that the Chen and Zhang' X-corner detector produced different errors at different corner locations. Taking the average of all the detected locations, within a nearby region, produced a constant offset/error of -0.5 pixel unit along the image coordinate axis. The constant offset/error could be improved by adding 0.5 pixel unit to each of the axis of the final result of the corner detection.

Although the algorithm was implemented with double floating point precision, but due to the rounding up issue, the number of strong responses produced by the implemented algorithm was slightly different from the results shown above. The actual number of responses produced by the implementation are shown in Table 4.13.

Table 4.13: Number of strong corner respond produced by the implemented Chen' X-corne

X-Corner	C1(u,v)	C2(u,v)	C3(u,v)	C4(u,v)
Number of respo	2	1	2	1

Due to the multiple responses issue, the performance of the Chen and Zhang' X-corner detector was not evaluated on the series of rotated synthetic images .

4.4.3.3 Performance of the Sojka corner detector in locating X-corners

The original implementation of Sojka' corner detector was downloaded from Sojka' website [69]. The downloaded implementation was applied on the generated synthetic images (see Figure 4.35). The results are as shown in Tables 4.14 and 4.15.

Table 4.14: Error in detecting the location of the X-corner (pixel unit) with Sojka' corner detector

Rotation (degree)	Corner							
	P1(x,y)		P2(x,y)		P3(x,y)		P4(x,y)	
0	0.00000	-1.00000	-1.00000	0.00000	0.00000	-1.00000	-1.00000	0.00000
15	0.35500	-0.76300	-0.23700	0.35500	0.64500	-0.23700	-0.76300	0.64500
30	-1.69900	-0.83434	0.42999	-0.61864	-0.35953	1.20188	-1.06115	0.98436
45	0.00000	0.71100	-1.71100	0.00000	-1.00000	-1.71100	0.71100	0.00000
60	0.69900	-0.69900	-0.30100	0.69900	-1.69900	-0.30100	-0.69900	-1.69900
75	-1.35500	-0.76300	-0.23700	0.64500	-1.64500	-0.23700	-0.76300	-1.64500

Table 4.15: Error in the estimated length and inner angle estimated using the detected X-corner locations

	The error of the estimated length (pixel unit) of each side of the black square(x-corner).				The error of the inner angle estimated (degree) at each corner of the black square(x-corner).			
Rotation (degree)	Length				Angle			
	L1	L2	L3	L4	P1	P2	P3	P4
0	-0.99000	-0.99000	1.00000	1.00000	-0.01100	1.15700	-0.01100	-1.13500
15	-0.27500	-0.79800	1.14000	1.65000	-0.30500	1.11200	0.28600	-1.09200
30	0.72300	0.72000	0.71700	0.72000	0.00000	0.00000	0.00000	0.00000
45	-1.71000	-1.71000	-2.41900	-1.00500	-0.41200	0.00000	0.41200	0.00000
60	0.72000	0.72000	0.72000	0.72000	0.00000	0.00000	0.00000	0.00000
75	-1.30920	-1.53778	-1.30181	-1.54289	0.80700	-1.09200	0.00000	0.28600

The performance of the Sojka' corner detector in locating the corners of the generated synthetic X-corner images was found to be sub-optimal since it had more than one pixel error (Table 4.14) while detecting the location of the actual corner and the length of the edges (Table 4.15). Also the measured angles of the black square were found to suffer having more than one degree error.

4.5 Discussion and conclusion

After extensive experimentation and careful analysis it was found that saddle point based X-corner detector was the best candidate. This is because it was observed that the saddle point based X-corner detector, inspired by Lucchese [61], implemented by Bouguet [46], out performed the other two types of corner detector algorithms (*i.e.* image intensity variation based and template based). This is due to the use of the suitable X-corner model where the location of the corner is located at the saddle point of the model. Last but not least, the use of very large window (fifteen by fifteen) and repeating interpolation increased the accuracy of the corner localisation.

Unfortunately, the use of large window and repeating interpolation are very processing intensive and required longer time to complete the process of corner detection. However, the process of corner detection is only needed to be performed once, at the initial state of the designed freehand 3-D surface data acquisition system. An accurate corner detector, rather than a fast corner detector, is more suitable for this application.

The fundamental of an accurate corner detector is to identify a good corner model.

Several corner models were discussed in this literature survey. L-corner was found to be much difficult to model (as it required non-linear operation) with respect to X-corner. Hence in this study X-corners (instead of L-Corners) were used as the salient features and Saddle point corner locator was used for locating the corners.

Chapter 5: Design, Implementation and Evaluation of Laser Plane Pose Estimation Methods

5.1 Introduction

The estimation of the laser plane pose was needed for the process of triangulation (Chapter 3, Section 3.4.3.1). In a 3-D reference coordinate frame, a plane in 3-D space can be represented by two parameters consisting of a 3-D point on the plane (m) and the unit normal vector of the plane $(\vec{n})$. Once these two parameters for a laser plane are resolved, any pixel, in a calibrated camera image, illuminated by the laser can be re-projected to the 3-D reference coordinate frame by using the triangulation process. The above process was used to estimate the 3-D locations of the points on the surface of any object by illuminating them with the intersection of the laser plane with the surface of the object. This is performed by sweeping a laser plane across the object and capturing the images of the object during the sweeping process. Thus, the laser pose plane estimation is an important component of the 3-D object surface estimation.

In this chapter, the process of estimating the pose of the laser plane is explained in detail. Three different methods were investigated to estimate the laser plane pose parameters.

5.2 Estimating the pose of the laser plane

From the geometry point of view, the result of intersection of a laser plane with a planar object is a laser line. When this information was digitised into a camera image, a 2-D line was formed by a series of image pixels illuminated by the laser plane. Different approaches can be used to estimate the pose of the laser plane. These different methods could either make use of the fact that the pixels, illuminated by the laser, lay on the laser plane or the laser line, formed by the pixels illuminated by the laser, lay on the laser plane.

The three different methods used in this study to estimate the pose of the laser plane are outlined below.

In the camera image captured while sweeping the laser light across the surface of the object, the line formed by the pixels illuminated by the laser light is formed by the intersection of the laser plane with the surface of the 3-D object. The first method makes use of the above fact to estimate the pose of the laser plane. Firstly, using the Hough Transform [47] a best fitting line was fitted to the pixels illuminated by the laser. Then, transforming the image coordinates of couple of points on the fitted line to the 3-D world coordinates, the parameters of the laser plane was estimated.

In the second method, the laser plane equation was estimated in 3-D space, by fitting the best fitting plane to all the points detected to be lying on the laser line.

In the third method, the concept of RANdom SAmple Consensus [7] (RANSAC) method was applied to solve the problem.

The above three methods, of estimating the pose of the laser plane, are described in detail below.

5.2.1 Estimation of the location of the pixels and the lines on the laser plane

To define the equation of a plane (equation 3.23), a minimum of three non-collinear points or two non coincided lines are needed. Hence the non-collinear points or lines that are lying on the laser plane need to be found first.

To estimate the locations of the pixels, and thus the location of the line lying on the laser plane, one needs to first detect those pixels in the image illuminated by the laser.

The estimation of the locations of the pixels illuminated by the laser is explained in detail in Section 3.4.2.1 (Laser Pixel Extraction Process).

Figure 5.1 illustrates the steps of extracting two non-collinear laser lines lying on the laser plane. The lines could be subsequently used for the laser plane pose estimation. The steps involved are as follows: A planar object, comprising of two non-collinear planes, was formed by attaching two planes at an angle. A background image (Img_{bg}) of the planar object comprising of the two planes, (referred as horizontal plane and vertical plane), was first obtained with a stationary camera (Figure 5.1a). The laser line generator was then turned on to illuminate a set of pixels along the vertical and the horizontal planes in the scene. The scene image, referred as laser image (Img_{L}), having the points illuminated by the laser was obtained by the same camera (Figure 5.1b).

Since the scene and the camera were keep static while capturing the above two images, the pixels illuminated by the laser can be extracted by finding the difference between the background image and the laser image (Figure 5.1c). The pixels illuminated by the laser on the vertical and the horizontal planes were grouped separately. The process of grouping the laser pixels is explained below.

The outermost corners of the black squares in the checker patterns, on the horizontal and the vertical planes were detected (Figure 5.2). The detected corners, that were four corners for each plane, were connected to form the 2-D boundaries around the horizontal and the vertical planes. The boundary of the 3-D object was manually outlined by the user.

To group the pixels, illuminated by the laser, as belonging to the vertical plane, or the horizontal plane or on the object, the pixels were tested to determine whether they laid within one of the three boundaries. If a pixel, illuminated by the laser line, fell within the object boundary as well as the vertical or horizontal plane boundaries, the pixel was grouped as a pixel belonging within the boundary of the object.

Using the pixels, illuminated by the laser, grouped within the vertical plane and the horizontal plane two lines were fitted (Figure 5.1d).

For the laser plane pose estimation, the pixels, illuminated by the laser, or the fitted lines on both the horizontal and the vertical planes (Figure 5.1d) were than re-projected on to the global reference 3-D coordinate system. Using the detected two non-collinear lines, the equation of the laser plane, with respect to the global reference 3-D coordinate system, was then estimated. All the pixels, illuminated by the laser, grouped within the object boundary, were used by the triangulation process (Section 3.4.3.1) to generate the 3-D points cloud.

The user had been provided with the facility to sweep with the laser line generator to generate a 3-D points cloud of the object during the free hand 3-D scanning process. Hence, the pose of the laser plane needed to be re-estimated in real time, every time the laser line generator was moved during the scanning process.

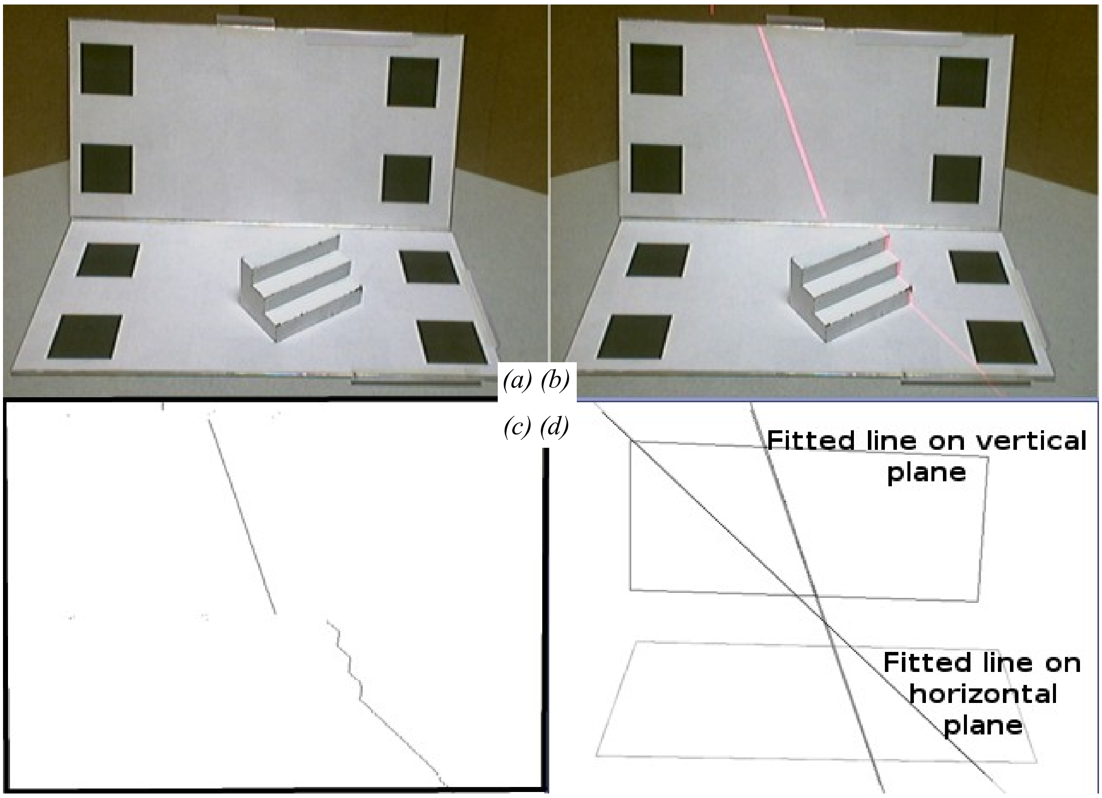


Figure 5.1: Process of extracting laser lines for the laser plane pose estimation.

(a) Background image. (b) Laser image. (c) Result of background subtraction. (d) Two detected laser lines one on the horizontal plane and one on the vertical planes.

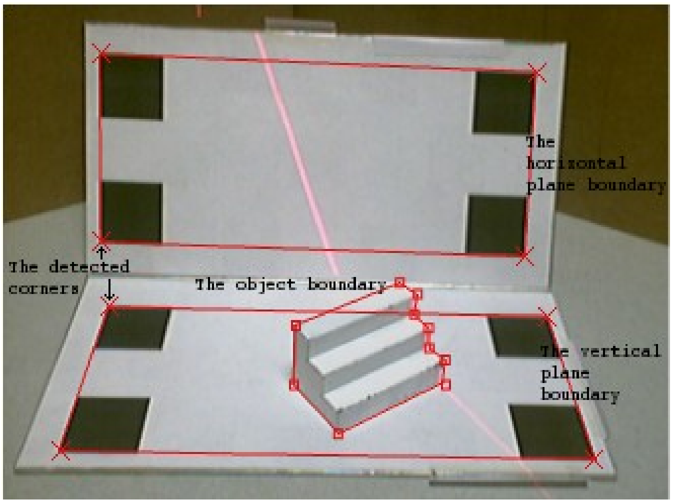


Figure 5.2: Image showing the boundaries of the horizontal plane and the vertical plane outlined automatically by connecting the detected corners.

The boundary of the object was manually segmented by the user with the provided mouse interface.

5.2.2 Estimating the pose of the laser plane using lines fitted along the laser illuminated pixels

A method used to estimate the pose of the laser plane was by using the two non-collinear lines, one on the vertical plane and the other on the horizontal plane, fitted to the laser illuminated pixels

The Hough transform (HT) [47] line detector was used to fit the best fitting line along the laser illuminated pixels.

Using the Hough transform, two 2-D lines, one from the horizontal plane and one from vertical plane, were detected. Two points from each of the two lines were chosen. Using the back projection process, the homogeneous image coordinates of these two pairs of points (x_{Laser}) , were transformed to the corresponding planar coordinate point $(X_{\{V,H\}plane})$ of the vertical or the horizontal plane by using the respective planar homography $(H_{\{V,H\}3x3})$ of the corresponding plane (5.1). The homogeneous planar coordinate points were transformed to the homogeneous point (X_{Laser}) in the camera coordinate frame, which is the world reference frame (5.2), (refer also to Section 3.4.2.4).

$$X_{\{V,H\}plane} = H_{\{V,H\}3x3}^{-1} \cdot x_{Laser_i} \tag{5.1}$$

Where $H_{\{V,H\}3x3}^{-1}$ is the inverse of $H_{\{V,H\}3x3}$

$$X_{Laser_i} = {}^{C}T_{\{V,H\}} \cdot X_{\{V,H\}plane} \tag{5.2}$$

Where $$X_{Hplane_i} = \begin{bmatrix} X_{plane(x)_i} / X_{plane(z)_i} \\ X_{plane(y)_i} / X_{plane(z)_i} \\ 0 \\ 1 \end{bmatrix}$$

A 3-D plane is defined by a 3-D point on the plane (m) and the unit normal vector of

From equation (3.23) $\vec{n} \cdot (X - m) = 0$

the plane $(\vec{n})$ such that any 3-D point X on the plane will satisfy equation (3.23).

The unit normal vector of the plane $(\vec{n})$, with respect to the world coordinate frame, can be resolved using equation (5.3).

$$\vec{n} = \frac{(\vec{v}_1 \times \vec{v}_2)}{\sin(\theta)} \tag{5.3}$$

Where

$$\begin{gathered} \vec{v}_1 = (X_{Laser1} - X_{Laser2}) / \|(X_{Laser1} - X_{Laser2})\| \\ \vec{v}_2 = (X_{Laser3} - X_{Laser4}) / \|(X_{Laser3} - X_{Laser4})\| \\ \theta = \cos^{-1}(\vec{v}_1 \cdot \vec{v}_2) \end{gathered}$$

Resolving for $\vec{n}$, we obtain (5.3). The corresponding point of the normal was obtained by using equation (5.4). This provides the average of the four points. The flow chart of the explained method is shown in Figure 5.3.

$$m = \frac{\sum_{(i=1)}^{4} X_{Laser_i}}{4} \tag{5.4}$$

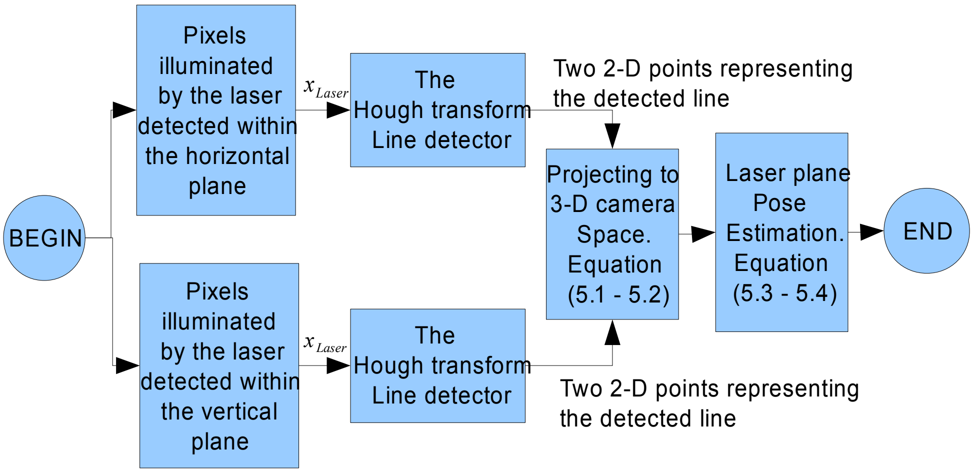


Figure 5.3: Flow chart for method explained in section 5.2.2 (Estimating the pose of the laser plane using lines fitted along the laser illuminated pixels)

5.2.3 Estimating the pose of the laser plane using the laser illuminated pixels through Moment of Inertial Analysis (MIA)

In the second method, the laser plane parameters were computed in the 3-D space. The reason for choosing this approach was that it was hypothesised that it was better to estimate the pose of the laser plane directly in 3-D space, because the error produced during the re-projection step could be reduced by using more pixels illuminated by the laser. The details of the process is given in detail below.

Firstly, all the pixel locations detected as illuminated by the laser were transformed as 3-D points with respect to the camera frame. The transformation was performed by using the equations (5.1) and (5.2) as explained in Section 5.2.2. Secondly, a plane was fitted to best fit all the 3-D points. In this study, to obtain the best fitted plane through a 3-D georeferenced data, and thus to estimate the plane parameters (m and $\vec{n}$), the Moment of Inertia Analysis (MIA) method [70][71] was used. MIA has the advantage of being able to provide the quantitative measure of the shape of the trace and also gives a measure of the reliability of the data set. For more detailed information about how to obtain the above measures refer to the paper by Fernandez [70].

The estimation of the plane parameters (m and $\vec{n}$), of the best fitting plane, using the MIA method is explained below.

For N 3-D laser points in the 3-D camera reference frame, $X_{Laser_{1...N}}$, projected from the 2-D image pixel location illuminated by the laser (5.2), it was assumed that the point, m , lying on the best-fitting laser plane, is the mean of those points ($m=mean(X_{Laser_{1...N}})$). Hence the normal of the best fitting laser plane can be resolved in least-square sense. The system of equations for the least-square estimation was formed as shown in equation (5.5).

From (3.23) $\vec{n}\cdot(X-m)=0$

$$\underbrace{\begin{bmatrix} X_{Laser(x)_1}-m_x & X_{Laser(y)_1}-m_y & X_{Laser(z)_1}-m_z \\ X_{Laser(x)_2}-m_x & X_{Laser(y)_2}-m_y & X_{Laser(z)_2}-m_z \\ \vdots & \vdots & \vdots \\ X_{Laser(x)_i}-m_x & X_{Laser(y)_i}-m_y & X_{Laser(z)_i}-m_z \end{bmatrix} \begin{bmatrix} n_{(x)} \\ n_{(y)} \\ n_{(z)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{:=M} \tag{5.5}$$

To solve the system of equation (5.5) in the least square sense, Singular Value Decomposition (SVD) was used. Since the input of the SVD must be in square matrix format, the matrix **M** (equation 5.5) was multiplied with it's transpose. The orientation matrix T , $T=M^T M$ **,** was obtained as shown in (5.6).

The eigenvector corresponding to the smallest eigenvalue of the matrix **T** was taken as the normal $\vec{n}$ of the laser plane. The flow chart of the explained method is shown in Figure 5.4.

$$T=\left\{\begin{matrix} \sum(a_i^2) & \sum(a_i b_i) & \sum(a_i c_i) \\ \sum(b_i a_i) & \sum(b_i^2) & \sum(b_i c_i) \\ \sum(c_i a_i) & \sum(c_i b_i) & \sum(c_i^2) \end{matrix}\right\} \tag{5.6}$$

Where

$$\left\{\begin{matrix} a_i \\ b_i \\ c_i \end{matrix}\right\} = \left\{\begin{matrix} X_{Laser(x)_i}-m_x \\ X_{Laser(y)_i}-m_y \\ X_{Laser(z)_i}-m_z \end{matrix}\right\} \tag{5.7}$$

For *i* = *1..N*

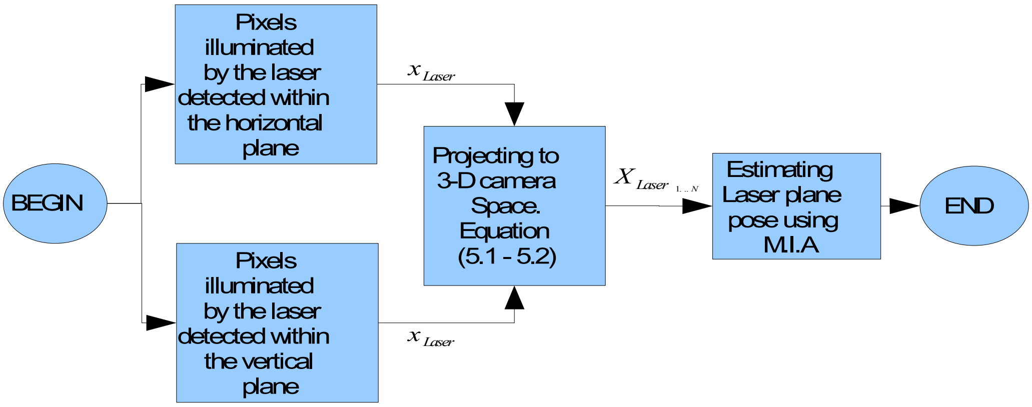


Figure 5.4: Flow chart for method explained in Section 5.2.3 (Estimating the pose of the laser plane using the laser illuminated pixels through Moment of Inertial Analysis (MIA))

5.2.4 The RANdom SAmple Concensus (RANSAC)

The third method to estimate the laser plane parameters makes use of the RANdom SAmple Concensus (RANSAC) process. RANSAC is a robust statistical method for estimating single or multiple models from a dataset. The method is robust due to it's ability to discard outliers. In the RANSAC process a minimum number of measured data were randomly selected and were used to compute the unknown parameters of the best fitting model. The accuracy or the validity of the instantiated model was evaluated by counting the number of consistence samples from all the measured datasets. The compared sample was added into a consensus set, if the error produced during the evaluation process was less than a defined threshold error value. A good model might have been found if the number of samples inside the consensus set was more or equal than the defined probability of number of good samples in the measured dataset. For example if the percentage of the samples inside the consensus set was 80% while the rate of possible good measured data was 70%, then it could be taken that a good model had been found.

The process of randomly selecting minimum number of points and instantiating the model fitting model was repeated until the number of trials exceeded a pre-defined confidence limit of the probability of obtaining a good model. Thus, at the end of the process one resolves the best model along with it's best consensus data set, with the least mean error.

The advantages of RANSAC clearly highlight the weaknesses of the other two methods namely the Hough transform line detector based method and the MIA method. Unlike

RANSAC, the Hough transform line detector computes all the possibilities of the unknown 2-D line parameters. The computational time increases proportional to the resolution of the model, the finer the resolution, the better the accuracy but more computational time will be required. For example, by increasing the resolution for θ from 180 steps (0 to 179, 1 degree per step) to 360 steps (0.5 degree per step), the total computation time will be increased by a factor of two. The second method, Moment of Inertia Analysis, assumes that the best fitting model passes through the centre of the mass of the projected 3-D dataset $(X_{Laser_{1...N}})$, and hence the Moment of Inertia Analysis (MIA) process will get distracted by outliers.

There are two possible ways of using RANSAC to estimate the pose to the laser plane. In the first method the estimated pixel locations, illuminated by the laser, are made use of by RANSAC to estimate the best fitting line model's parameters. The laser plane pose estimation is subsequently performed by using either the lines or the consensus pixel locations. This method is named as the RANSAC 2-D straight line model fitting.

In the second method, RANSAC process was used directly to estimate the best fitting plane by using all the estimated pixel locations, illuminated by the laser lying on both of the horizontal and vertical boundaries. The second approach, is named as the RANSAC 3-D plane model fitting.

The above two possible ways of applying RANSAC to estimate the pose of the laser plane were illustrated in Figure 5.5.

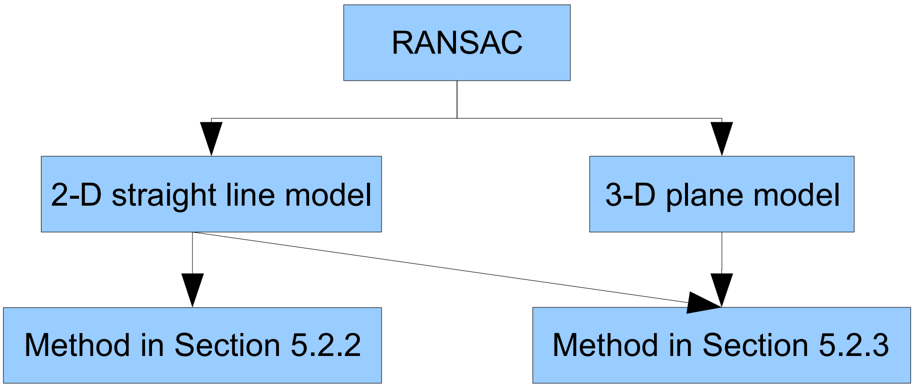


Figure 5.5: The possible ways of using RANSAC to determine the pose of the laser plane.

5.2.4.1 The RANSAC 2-D straight line model fitting

RANSAC had been used in many different applications to estimate the unknown parameters of a model to be fitted to a set of data. In the current application, the model that needs to be fitted is the equation of a line in polar space to a set of locations in 2-D space.

The 2-D locations are the estimated image pixels illuminated by the laser. Since the estimation process, namely thresholding or background subtraction, might yield sub-optimal or inaccurate locations, we need to find the best possible set of 2-D locations which were actually illuminated by the laser. To achieve this, we make use of the RANSAC process to find the best fitting line which will represents the intersection of the laser plane with the scene which was projected onto the image plane as the set of pixels illuminated by the laser.

To carry out the above RANSAC process, the best fitting line model's parameters (r and θ) were estimated in polar space. The parameters of a line in polar space passing through two locations, given in the Cartesian space, can be derived as discussed below.

Following is the process of expressing the function, in polar space, of a line passing through two locations given in the Cartesian space. One needs at least two points, in 2-D space, to define a unique line. A line $\vec{w}_1$, (Figure 5.6) passing through minimum of two points (say P_1 to P_4), can be defined by the two polar parameters, namely the orthogonal distance r_o of the line to the origin, and the angle, θ_a , between the orthogonal line and one of the Cartesian axis, u (Figure 5.6). The values of r_o and θ_a can be resolved as follows:

Let $\vec{w}_0$ be the normal to the line $\vec{w}_1$ drawn from the origin.

Let $\vec{w}_0$ intersects $\vec{w}_1$ at the location $P_2(u,v)$.

The polar co-ordinates for the intersecting point P_2, can be resolved by using a general point equation (5.10).

Then the values of r_o and θ_a can be resolved by using the equations (5.8) and (5.9) respectively.

$$r_o = \sqrt{P_2(u)^2 + P_2(v)^2} \tag{5.8}$$

$$\theta_a = \cos^{-1}\left(\begin{bmatrix}1 & 0 & 0\end{bmatrix}^T \cdot \vec{w}_0\right) \tag{5.9}$$

Where

$$\vec{w}_0 = \begin{bmatrix}-\|\vec{w}_1(v)\| & \|\vec{w}_1(u)\| & 0\end{bmatrix}^T$$

$$\|\vec{w}_1\| = \begin{bmatrix}\|\vec{w}_1(u)\| \\ \|\vec{w}_1(v)\| \\ 0\end{bmatrix} = \left\|\begin{bmatrix}P_4(u) - P_1(u) \\ P_4(v) - P_1(v) \\ 0\end{bmatrix}\right\|$$

$$P_l = P_s + \lambda \|\vec{w}_l\| \tag{5.10}$$

Where

P_l , any point on a line

P_s , a starting point of the line

$\|\vec{w}_l\|$, the unit vector of the line

λ , the length of the line from P_s to P_l .

Since P_2 was lying on both lines ($\vec{w}_0, \vec{w}_1$), the two equations of lines (5.11 and 5.12) were defined. By solving these equations, the value of intersection point, P_2 , can be resolved by (5.16).

$$P_2 = [0 \quad 0 \quad 0]^T + \lambda_0 \|\vec{w}_o\| \tag{5.11}$$

$$P_2 = P_1 + \lambda_1 \|\vec{w}_1\| \tag{5.12}$$

$$\lambda_0 \|\vec{w}_o\| = P_1 + \lambda_1 \|\vec{w}_1\| \tag{5.13}$$

By performing cross product of both sides with $\|\vec{w}_1\|$ yields equation (5.14)

$$\lambda_0 (\|\vec{w}_o\| \times \|\vec{w}_1\|) = P_1 \times \|\vec{w}_1\| \tag{5.14}$$

the value of λ_0 can hence be resolved by only taking the magnitude part of right hand side, *i.e.*

$$\lambda_0 = magnitude(P_1 \times \|\vec{w}_1\|) = P_1(v) \|\vec{w}_1(u)\| - P_1(u) \|\vec{w}_1(v)\| \tag{5.15}$$

From equation (5.11), $P_2[u \quad v \quad 0]^T$ can be resolved by equation (5.16)

$$P_2 = \lambda_0 \begin{bmatrix} -\|\vec{w}_1(v)\| \\ \|\vec{w}_1(u)\| \\ 0 \end{bmatrix} \tag{5.16}$$

5.2.4.1.1 Determine the consensus samples of a line model

The instantiated line model, through the randomly selected pair of locations, had the possibility of becoming the best fitting line model if the number of the inlier pixel locations, drawn from the locations illuminated by the laser, exceeded a minimum count. The criteria for a pixel location being an inlier of the instantiated line model is defined as the maximum orthogonal distance, r_3 , of the pixel, P_5 , away from the line (Figure 5.6). If r_3 is smaller than a pre-defined threshold value, *e.g.* two pixel units away from the instantiated line, the pixel location P_5 will be considered as an inlier. The value of r_3 can be resolved using the equation (5.17).

$$r_3 = r_5(\|\vec{w}_o\| \cdot \|\vec{w}_5\|) \tag{5.17}$$

Where

$$\|\vec{w}_5\| = \left\| \begin{bmatrix} P_2(u) - P_5(u) \\ P_2(v) - P_5(v) \\ 0 \end{bmatrix} \right\|$$

$$r_5 = \sqrt{(P_2(u) - P_5(u))^2 + (P_2(v) - P_5(v))^2}$$

$(\|\vec{w}_o\| \cdot \|\vec{w}_5\|)$ is the dot product between the two unit vector.

All the possible line models along with their consensus data set were stored. When the number of instantiated models exceeded a predefined threshold value, all the previously stored line models were evaluated to find the best fitting line model. The stored line model with the least mean error, m_{err} , (equation 5.18) was selected as the best line model. The flow chart of the process of RANSAC 2-D straight line model fitting is shown in Figure 5.7.

$$m_{err} = \frac{\sum_{(i=1)}^{n} OD_i}{n} \tag{5.18}$$

Where OD_i is the measured orthogonal distance of a point away from line model, *e.g.* r_3 in Figure 5.6 , Equation (5.17), and n is the number of 2-D laser points in the consensus set.

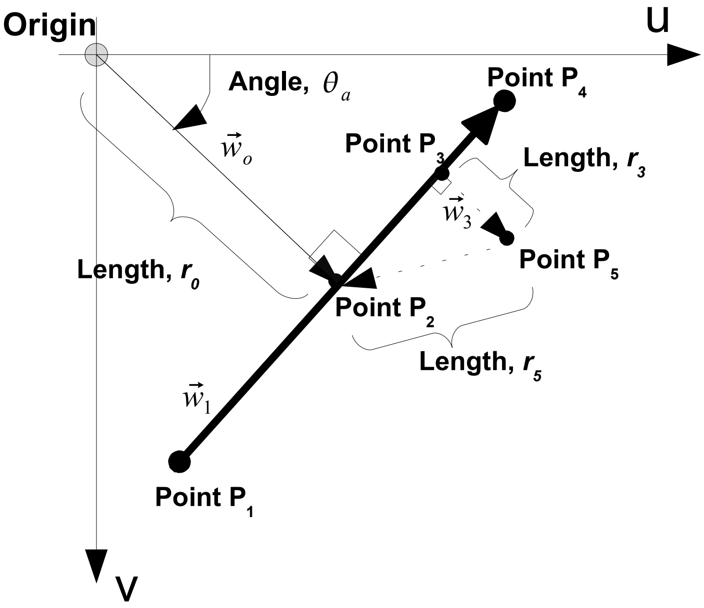


Figure 5.6: The polar space of a line. A line is formed by two points P_1 and P_4.

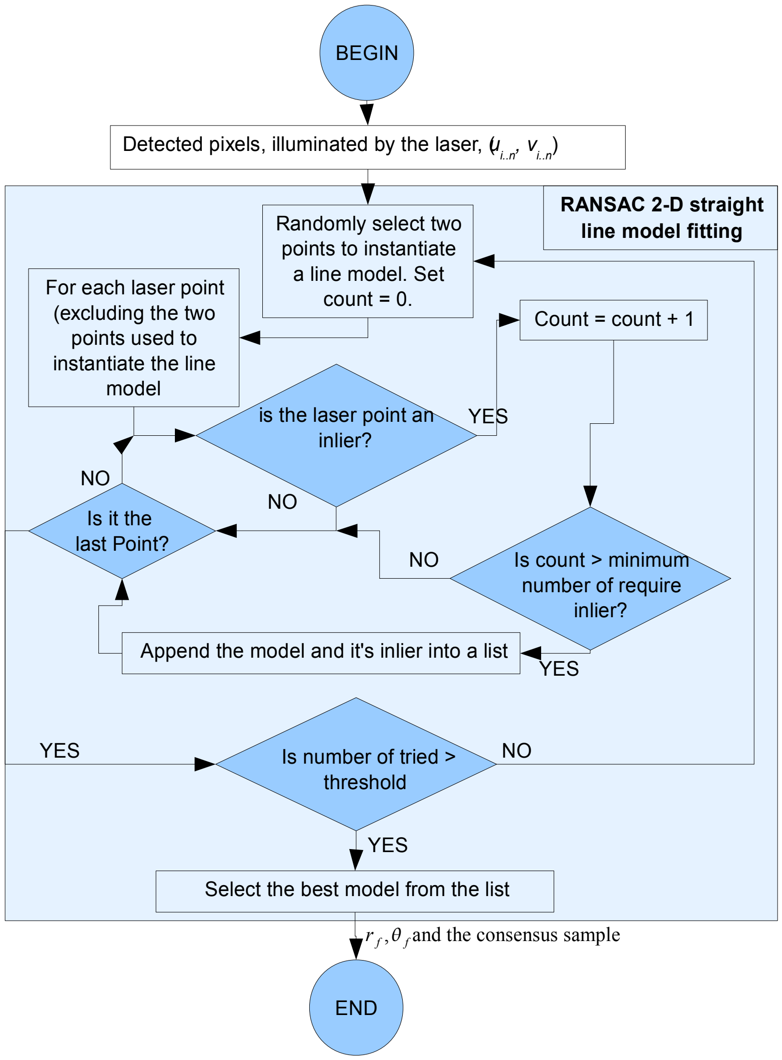


Figure 5.7: Flow chart of the process of RANSAC 2-D straight line model fitting.

5.2.4.1.2 Estimating laser plane pose with the result of RANSAC 2-D straight line model fitting

The result produced at the end of the RANSAC process contained two sets of information, namely the fitted line model parameters $(r_f \text{ and } \theta_f)$ of the best fitting line passing through the pixels illuminated by the laser and the consensus samples (pixel locations illuminated by the laser), of the line model. Using these two information the pose of the laser plane can be estimated in two different ways. These ways are explained in the following sections.

5.2.4.1.2.1 Laser plane pose estimation with the best fit 2-D straight line model

Two best fitting line models, through the pixels illuminated by the laser, one from the horizontal plane and one from vertical plane were estimated using the RANSAC process. Four locations, two from each of the instantiated best fitted line model, were then used and the pose of the laser plane was estimated as explained in Section 5.2.2, using equations (5.3) and (5.4). The process is illustrated in Figure 5.8.

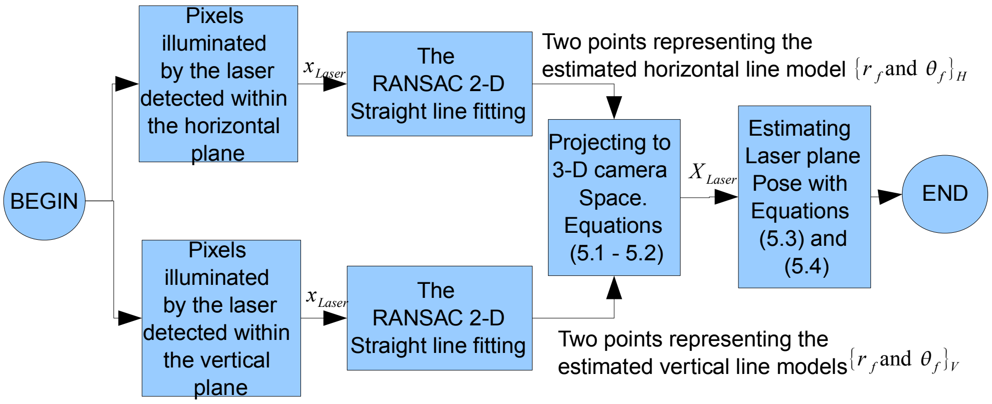


Figure 5.8: Flow chart for method explained in Section 5.2.4.1.2.1 (Laser plane pose estimation with the best fit 2-D straight line model).

5.2.4.1.2.2 Laser plane pose estimation with the consensus samples of the best fit 2-D straight line model

Another way to estimate the pose of the laser plane is to make use of all the pixel locations which were the consensus samples (Figure 5.9). Using all the consensus samples to estimate the laser plane pose will be more accurate for the following reason.

The process of projecting 2-D information (either the best fit 2-D lines and or the consensus samples) requires two set of input parameters, namely the pose of the planar object and the camera intrinsic parameters. Error might be introduced during the process of estimating these two sets of parameters. If we assume that the noise introduced in the estimating processes had a Gaussian distribution with zero mean, it is recommended to use more locations (that is the consensus samples), where the errors might cancel out each other, to estimate the laser plane pose.

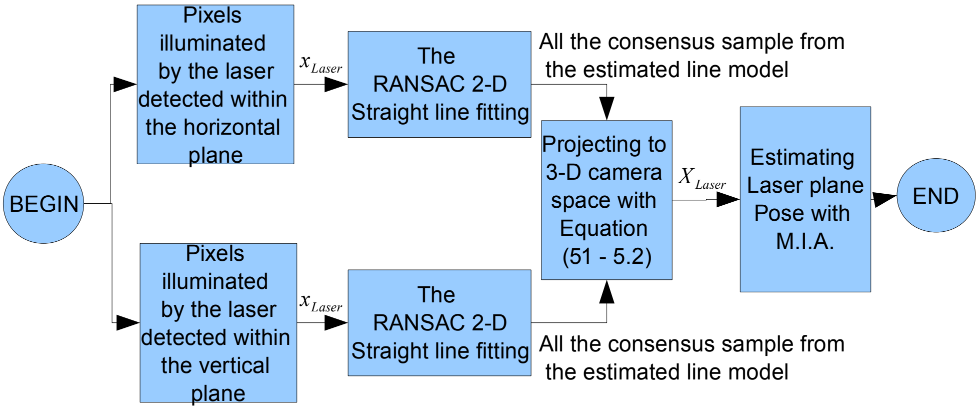


Figure 5.9: Flow chart for method explained in Section (5.2.4.1.2.2 Laser plane pose estimation with the consensus samples of the best fit 2-D straight line model).

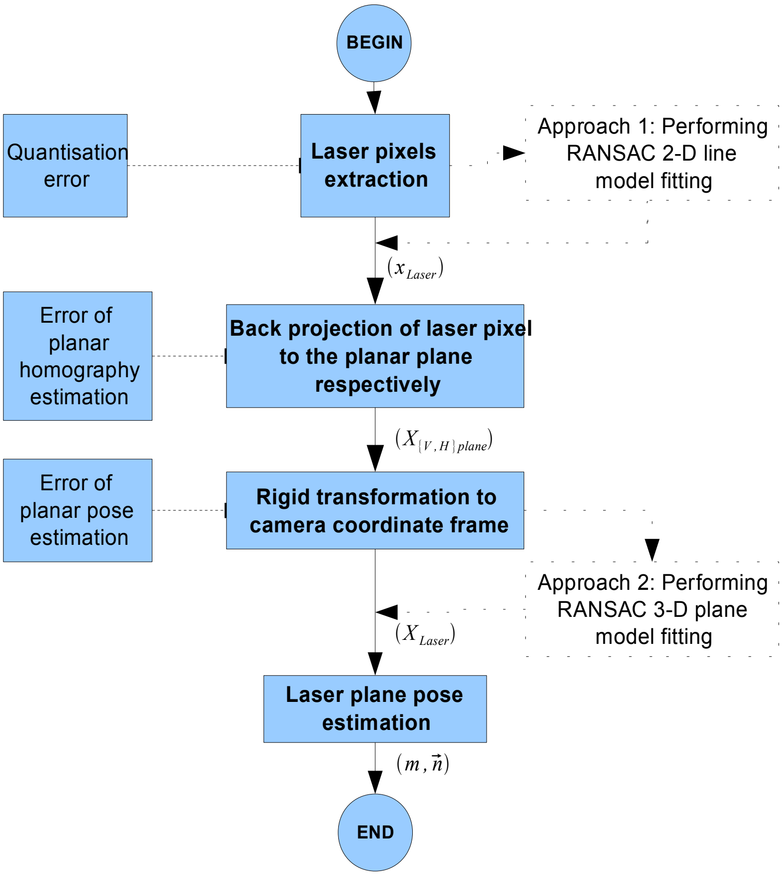


Figure 5.10: The process flow of the laser plane pose estimation. The error source during each process is shown in the left hand side of each process. Two possible scenarios for applying different RANSAC fitting process were identified and shown on the right hand side.

5.2.4.2 The RANSAC 3-D plane model fitting

In the RANSAC 3-D plane model fitting approach, using RANSAC to estimate the best fitting plane, all the estimated pixel locations, illuminated by the laser lying on both of the horizontal and vertical boundaries were made use of. This process of RANSAC 3-D plane model fitting was performed as explained below.

The process started by converting all the located image pixels illuminated by the laser to 3-D points in the camera space using equation (5.2). Next the RANSAC process was initiated by randomly selecting three 3-D points in the camera space. Using these three 3-D points, the unit normal vector of the plane $(\vec{n})$, with respect to the world coordinate frame, was resolved using equation (5.3) and the corresponding point of the normal m_i, was obtained by equation (5.4). Now, from the rest of the 3-D points a set of consensus 3-D point samples, were selected. These inlier of the initiated plane model, was selected by measuring the orthogonal distance, d (equation 5.19), to the initiated plane model. If the measured orthogonal distance of the 3-D point was smaller than a predefined threshold value, an inlier was identified and appended into the consensus sample.

A set of consensus 3-D point sample, that is the inlier of the initiated plane model, could be determined from the list of 3-D points by measuring the orthogonal distance, d (equation 5.19), of a 3-D point to the initiated plane model. If the measured orthogonal distance of the 3-D point was smaller than a predefined threshold value, then that 3-D point was identified as an inlier and appended into the consensus sample.

From (3.23)

$$\vec{n} \cdot (X - m) = 0$$

$$d = \vec{n}_i \cdot (X - m_i) \tag{5.19}$$

After iterating through the list of 3-D points, the initiated plane model was considered as a good model if and only if the number of inlier points was found to be more than a predefined value (of number of inlier points) and the sum of all the orthogonal distances was the minimum when compared to the other possible instantiated plane models. The flow chart shown in Figure 5.11 illustrates the process of the RANSAC 3-D plane model fitting.

5.2.4.2.1 Laser plane pose estimation with RANSAC 3-D plane model fitting

To recapitulate, in the laser plane pose estimation using RANSAC 3-D planes model fitting process, the set of pixels, in the horizontal and the vertical planes, identified as illuminated by the laser, were first transformed to the camera 3-D space using equation (5.2). Subsequently, RANSAC 3-D plane model fitting was used to identify the inlier points for the best fitting plane model. Finally, the identified inlier points were used to estimate the pose of the laser plane with the M.I.A method. The flow chart in Figure 5.12 outlines the laser plane pose estimation with RANSAC 3-D plane model fitting process.

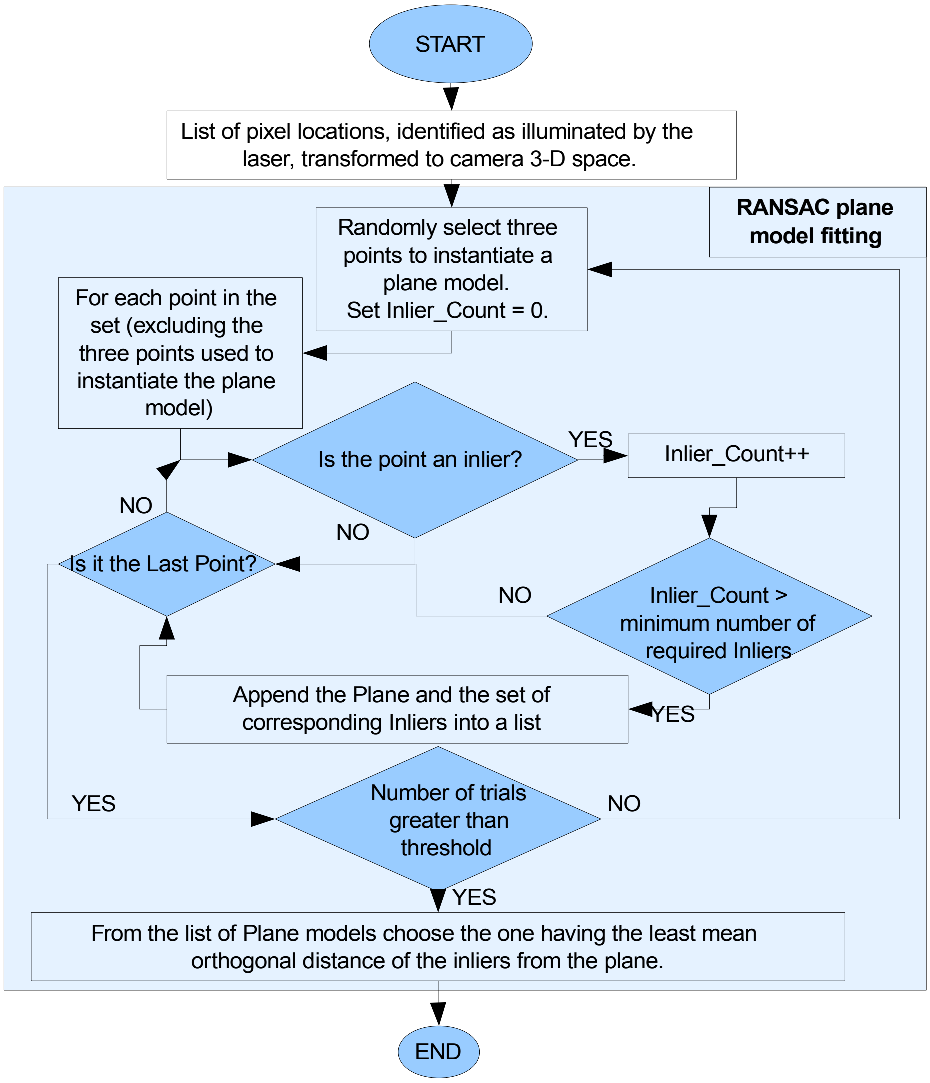


Figure 5.11: Flow chart of the RANSAC plane model laser plane pose estimation process.

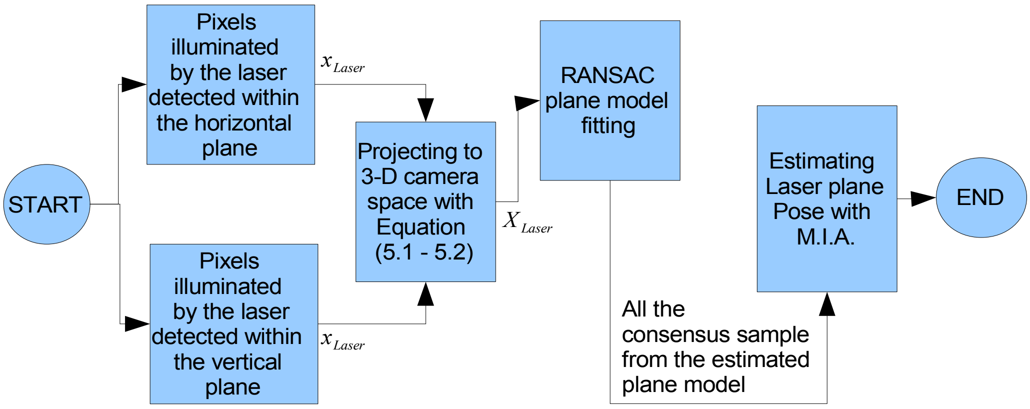


Figure 5.12: Flow chart for method explained in Section 5.2.4.2.1 Laser plane pose estimation with RANSAC 3-D plane model fitting.

5.3 Critical reflection on the different RANSAC based laser plane pose estimation methods devised in this study

The process of estimating the pose of the laser plane is subjected to different types of errors and noise sources. The errors and noise sources could be the quantisation noise, planar homography estimation error, and planar pose estimation error. Figure 5.10 shows the error sources through out the process of laser plane pose estimation.

The quantisation noise is introduced during the process of forming the camera image through digitisation, using a digital camera.

Planar homography is needed to project the detected pixel, illuminated by the laser, on to the planar space. The planar homography estimation error might be introduced during the process of identifying corners' location in the digitised image.

The pose of the planar object is needed to transform the planar coordinate points to the camera 3-D coordinate space. The plane pose estimation error, is introduced due to the error in estimating planar homography and due to the error in estimating the camera intrinsic parameters.

Other less significant error sources, during planar homography estimation, are the non-rigidity of the planar object and the lack of precision in marking the corners points.

To minimize the effect of the above mentioned errors, during the estimation of the pose of the laser plane, one needs to employ a robust method such as RANSAC. The input to the RANSAC process is the pixel locations illuminated by the laser. As discussed in the earlier sections, following are the three different ways one could use the RANSAC process to estimate the pose of the laser.

For the first two methods the estimated pixel locations, illuminated by the laser, are made use of by RANSAC firstly to estimate the best fitting line model's parameters. This RANSAC estimation is subsequently used, to estimate the laser plane, in the following two different ways.

- The laser plane pose estimation is performed by using the best fitted lines.

- The laser plane pose estimation is performed by using the the consensus pixel locations.

The above two methods are categorised as the RANSAC 2-D straight line model fitting.

In the third method the RANSAC process was used directly to estimate the best fitting plane by using all the estimated pixel locations, illuminated by the laser, lying on both of the horizontal and vertical boundaries. This third approach, is named as the RANSAC 3-D plane model fitting.

To choose the best possible method, among the above three different methods, one can perform the following thought experiment; (since much of modern physics is built not upon measurement but on thought experimentation [72]).

To choose the RANSAC 2-D straight line model fitting as the best method, one can propose the following argument. This method must be much more robust since this method utilises the following two different geometry constraints. Firstly, the two-dimensional straight line constraints, that the valid points should lie on the line produced by the intersection of the laser plane with the other two planar objects. The second, three dimensional geometrical constraint is that the chosen points should lie on the same plane which is chosen as the laser plane. The input of the RANSAC 3-D plane model fitting is only the data scattering around two regions, intersection of the laser plane with the two planar objects, but not fully across the entire laser plane.

To choose RANSAC 3-D plane model fitting as the best method, one can propose the following argument. In this method RANSAC was performed, to find the consensus sample of the best 3-D laser plane model fitting, by using all the 3-D projected points, rather than finding the best fitting 2-D line model through extracted pixels illuminated by the laser and subsequently using the points on the line to resolve the plane.

Based on the above two arguments the following were concluded:

Firstly, for the RANSAC 3-D plane model fitting, even though the method makes use of all the 3-D projected points, those points are only along the line formed by the intersection of the laser plane with the two planar objects. To fit a plane, it would have been more appropriate, if one makes use of a set of points sampled from across the entire plane.

In the RANSAC 2-D straight line model fitting, all the inlier pixels illuminated by the laser in the fitted 2-D line models are subsequently projected into the camera 3-D space. The linear least square 3-D plane fitting method (MIA) which makes use of the three dimensional constraint and the law of Truly Large Numbers [73], determines the pose of the laser plane.

Hence the RANSAC 2-D straight line model fitting is more appropriate since the method fully makes use of the two dimensional constrains.

The conclusion reached above, through the thought experiment, was validated by a practical experiment. This practical experiment is discussed in detail in the following section.

5.4 Practical experiment to validate the hypothesis

The experimental set up was as follows. Two planar objects, having X-corner patterns, were used as the horizontal and the vertical plane. The planar object having sixteen X-corners is shown in the Figure 5.13. A webcam was calibrated using the camera calibration toolbox [46]. The Saddle point based corner detector algorithm (Section 4.3.3.2.1) was used to estimate the location of the sixteen X-corners on the planar objects with up to sub-pixel accuracy.

To evaluate the accuracy of the 3-D location of points, using the different laser pose estimation methods, a 3-D object with precise measurements was used. The white 3-D object used was the shape of a stair case. Each tread of the staircase was of ten millimetres depth and height and fifty millimetres in length (Figure 5.13).

To scan the object, the laser line emitter was attached to a linear translator having an accuracy of 0.02 millimetre. The configuration of the set up is shown in Figure 5.13.

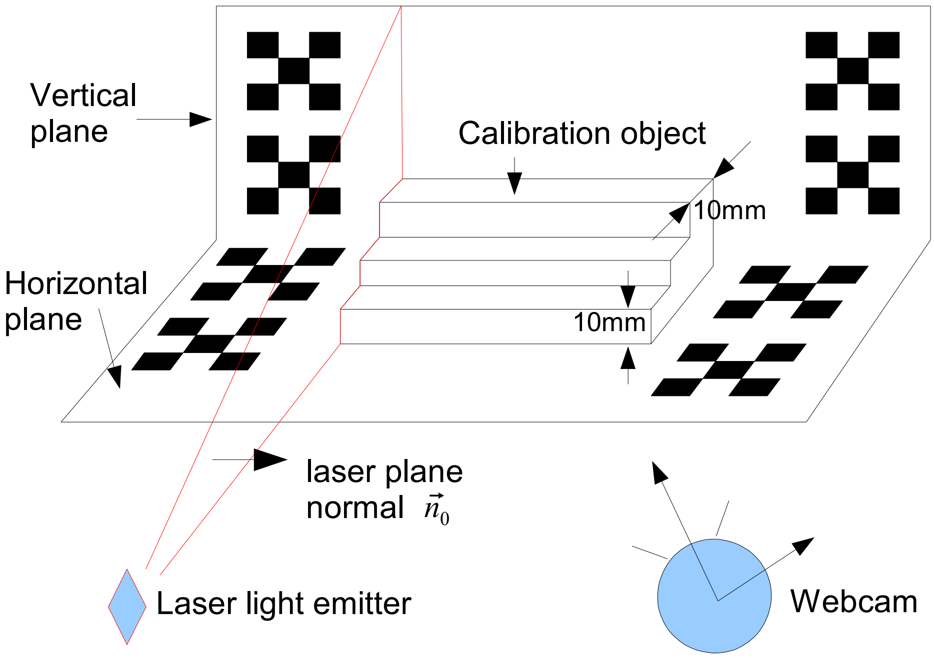


Figure 5.13: Figure illustrating the experimental set up.

The 3-D object was scanned by translating the laser line across the surface of the 3-D object. The laser plane was translated linearly by five millimetres along the normal of the laser plane, $\vec{n}_0$. A total of eleven images, one for each of the scan locations, were captured at different locations (the intermediate images of the experiment setting can be found in Appendix C). The pose of the laser plane was estimated by using the following three methods:

- Fitting the best fitting line model and using the best fitted lines; as explained in the Section 5.2.4.1.2.1 (R2D1).
- Fitting the best fitting line model and using the the consensus pixel locations; as explained in the Section 5.2.4.1.2.2 (R2D2).
- Directly estimating the best fitting plane by using all the estimated pixel locations; as explained in the Section 5.2.4.2.1 (R3D).

To evaluate the performance of the above three different methods of estimating the pose of the laser plane, the following two measurements were used:

Measurement of the normal to the laser plane at each scan plane. Since the laser plane was translated linearly, along the normal to the laser plane, the measured norm should be the same at different locations. The angular difference θ , between the normal of the laser planes at two different scan locations was estimated using equation (3.24). This difference should be equal to zero.

The distance between each scan plane needed to be measured. Since the laser scan plane was linearly translated precisely five millimetres between each scan, the distance between the scan planes should be five millimetres. The distance between two estimated scan planes , $(m_1, \vec{n}_1)$ and $(m_{2,} \vec{n}_2)$, was calculated as follows. One of the estimated scan plane $(m_1, \vec{n}_1)$ was taken as a reference plane. The relative distance, d , (see equation 5.20) between the two estimated planes was then measured by finding the shortest distance of point m_1 , along the normal of the reference plane, $\vec{n}_1$, and the intersecting point, m_i (see equation 5.21) , at the second plane $(m_2, \vec{n}_2)$.

$$d=\sqrt{(m_i(x)-m_1(x))^2+(m_i(y)-m_1(y))^2+(m_i(z)-m_1(z))^2} \tag{5.20}$$

where

$$m_i=m_1+t(n_1) \tag{5.21}$$

$$t=\frac{(m_2-m_1)^T \vec{n}_2}{\vec{n}_1^{\,T} \vec{n}_2} \tag{5.22}$$

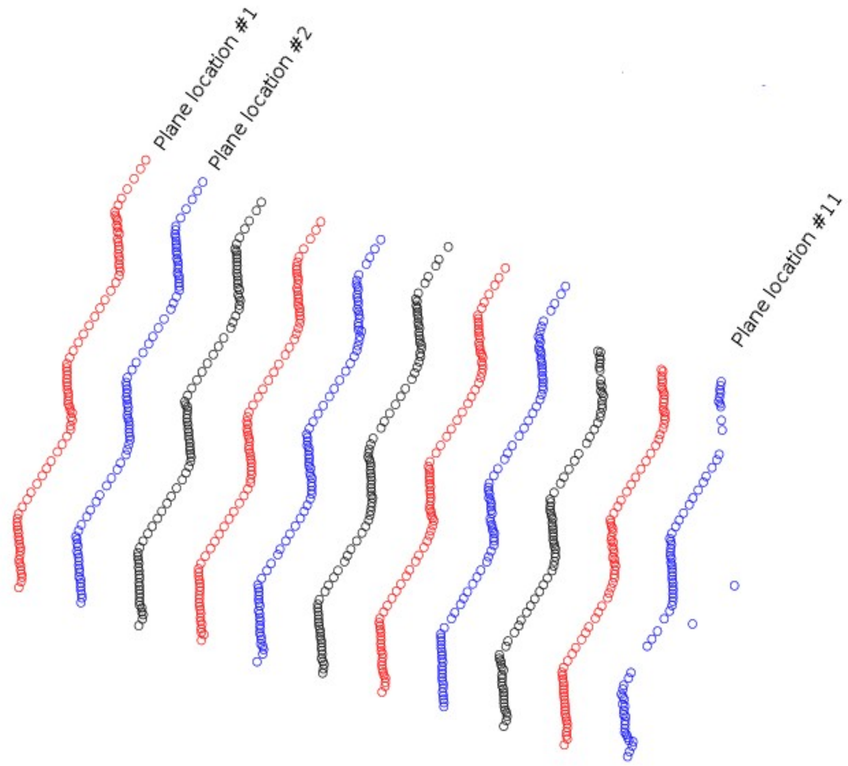


Figure 5.14: Figure illustrating locations of the points illuminated by the laser on the eleven scan planes.

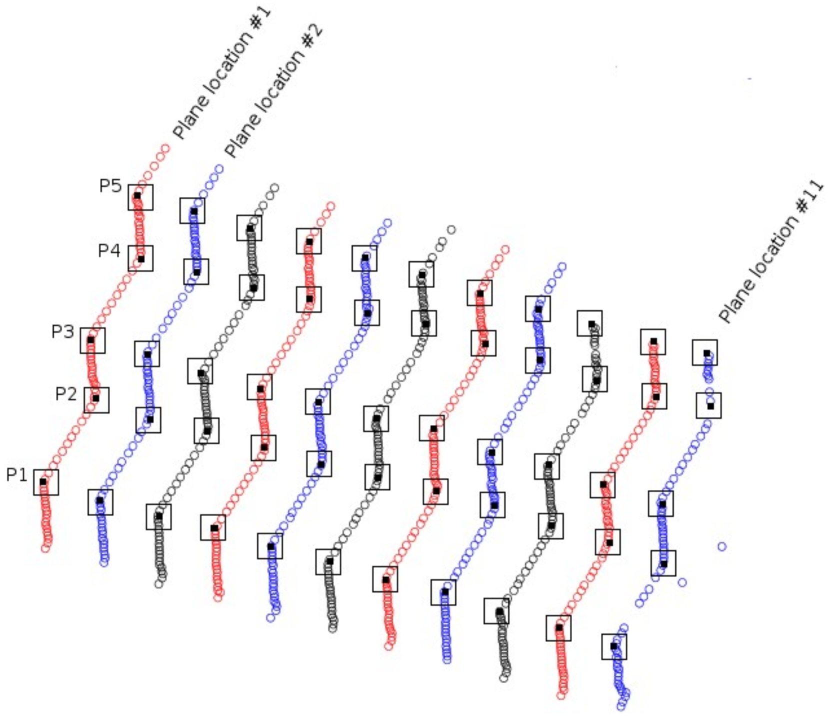


Figure 5.15: The five major points, on the surface of the staircase illuminated by the laser plane at eleven different locations, selected by the user to estimate the depth and height of the tread.

5.5 Results and discussions

In this section the results obtained to evaluate the accuracy of the laser plane pose estimation methods and the the quality of the acquired 3-D points of a calibrated 3-D object are presented and discussed.

5.5.1 Effectiveness of the proposed laser plane pose estimation methods

Table 5.1 lists the difference for each of the three different methods of laser plane pose estimation. The standard deviation of the difference for each of the method is plotted and shown in Figure 5.16.

The results of the estimated distance between the different scan planes are shown in Table 5.2. The difference between the actual distance and the estimated distance between the scan planes are shown in Table 5.3. The standard deviation of the difference between the actual distance and the estimated distance is plotted and shown in Figure 5.17.

From the results of measuring the norm of the scan planes and the distance between the scan planes following observations were made:

The performance of estimating the laser plane pose by the method, namely fitting the best fitting line model and using the best fitted lines (R2D1) is the least accurate method. Large standard deviations were obtained in the second set of measurement (Figure 5.17).

The performance of estimating the laser plane pose by the two methods, namely fitting the best fitting line model and using the the consensus pixel locations (R2D2) and directly estimating the best fitting plane by using all the estimated pixel locations (R3D) are almost the same. Except that the standard deviation of the angular difference between the norm of the scan planes, for the laser plane pose estimation using R3D is slightly larger than using R2D2 (Figure 5.16). This might due to the inlier points obtained by R3D are more scatter, subjected to both the planar homography and planar pose estimation errors (Figure 5.10) and do not fully utilise the available 2-D straight line constraints to filter them. However, the contribution of this little angular different (standard deviation of 0.5 degree) is too small to be observed, as one can see in the second measurement (distance between each plane), across the 50mm length (from plane number one to the plane number eleven).

5.5.2 The quality of the acquired 3-D points of a calibrated 3-D object

The performance of the three methods to estimate the pose of the laser plane were also compared by reconstructing the 3-D object. This comparison was done as follows:

The pixels illuminated by the laser, lying on the surface of the calibration object in each of the camera image were extracted and triangulated. The result of the process of triangulation provided the 3-D location of the pixels illuminated by the laser on the surface of the object (see Figure 5.14). Five points of interest, as shown in Figure 5.15, were manually selected and the depths (P1 to P2 and P3 to P4) and the heights (P2 to P3 and P4 to P5) of the treads of the staircase were evaluated at each of the scan planes. All together forty four measurements were estimated. The angular values at locations P2, P3 and P4 of the staircase were estimated (three measurements per plane, all together thirty three measurements).

The results are listed in Table 5.4. From the results shown the following, could be deduced

- the results obtained through the methods R2D2 and R3D are better than that of R2D1.
- both the method (R2D2 and R3D) produced the same standard deviation of the measurement errors. The standard deviation of both the methods were found relatively small as compared to the R2D1 method.

Table 5.1: The angular difference, measured in degrees between the norm of the laser plane at each scan location

Location		1	2	3	4	5	6	7	8	9	10	11
	R2D1											
1		0.00	0.39	0.48	0.17	0.19	0.14	0.27	0.37	0.24	0.32	0.45
2		-0.39	0.00	0.11	0.31	0.28	0.47	0.60	0.66	0.58	0.66	0.75
3		-0.48	-0.11	0.00	0.41	0.38	0.55	0.67	0.72	0.65	0.72	0.80
4		-0.17	-0.31	-0.41	0.00	0.03	0.30	0.43	0.54	0.41	0.49	0.62
5		-0.19	-0.28	-0.38	-0.03	0.00	0.32	0.46	0.55	0.43	0.51	0.64
6		-0.14	-0.47	-0.55	-0.30	-0.32	0.00	0.13	0.23	0.11	0.19	0.31
7		-0.27	-0.60	-0.67	-0.43	-0.46	-0.13	0.00	0.12	0.02	0.06	0.18
8		-0.37	-0.66	-0.72	-0.54	-0.55	-0.23	-0.12	0.00	0.14	0.09	0.09
9		-0.24	-0.58	-0.65	-0.41	-0.43	-0.11	-0.02	-0.14	0.00	0.08	0.21
10		-0.32	-0.66	-0.72	-0.49	-0.51	-0.19	-0.06	-0.09	-0.08	0.00	0.14
11		-0.45	-0.75	-0.80	-0.62	-0.64	-0.31	-0.18	-0.09	-0.21	-0.14	0.00
	MAX	0.00	0.39	0.48	0.41	0.38	0.55	0.67	0.72	0.65	0.72	0.80
	MIN	-0.48	-0.75	-0.80	-0.62	-0.64	-0.31	-0.18	-0.14	-0.21	-0.14	0.00
	STD	0.15	0.35	0.41	0.36	0.37	0.30	0.30	0.32	0.28	0.29	0.28

Location		1	2	3	4	5	6	7	8	9	10	11
	R2D2											
1		0.00	0.23	0.24	0.09	0.20	0.26	0.41	0.54	0.39	0.45	0.52
2		-0.23	0.00	0.03	0.15	0.38	0.47	0.61	0.75	0.59	0.66	0.74
3		-0.24	-0.03	0.00	0.16	0.39	0.48	0.62	0.76	0.60	0.67	0.75
4		-0.09	-0.15	-0.16	0.00	0.23	0.31	0.46	0.60	0.44	0.50	0.59
5		-0.20	-0.38	-0.39	-0.23	0.00	0.09	0.23	0.37	0.21	0.28	0.37
6		-0.26	-0.47	-0.48	-0.31	-0.09	0.00	0.15	0.28	0.13	0.19	0.28
7		-0.41	-0.61	-0.62	-0.46	-0.23	-0.15	0.00	0.14	0.02	0.05	0.14
8		-0.54	-0.75	-0.76	-0.60	-0.37	-0.28	-0.14	0.00	0.16	0.09	0.05
9		-0.39	-0.59	-0.60	-0.44	-0.21	-0.13	-0.02	-0.16	0.00	0.07	0.16
10		-0.45	-0.66	-0.67	-0.50	-0.28	-0.19	-0.05	-0.09	-0.07	0.00	0.09
11		-0.52	-0.74	-0.75	-0.59	-0.37	-0.28	-0.14	-0.05	-0.16	-0.09	0.00
	MAX	0.00	0.23	0.24	0.16	0.39	0.48	0.62	0.76	0.60	0.67	0.75
	MIN	-0.54	-0.75	-0.76	-0.60	-0.37	-0.28	-0.14	-0.16	-0.16	-0.09	0.00
	STD	0.17	0.34	0.35	0.30	0.29	0.29	0.29	0.34	0.26	0.27	0.28

Location		1	2	3	4	5	6	7	8	9	10	11
	R3D											
1		0.00	0.05	0.08	0.05	0.63	0.72	0.88	1.04	1.11	1.14	1.13
2		-0.05	0.00	0.04	0.03	0.66	0.74	0.91	1.07	1.14	1.17	1.16
3		-0.08	-0.04	0.00	0.05	0.69	0.77	0.94	1.10	1.17	1.20	1.19
4		-0.05	-0.03	-0.05	0.00	0.64	0.72	0.89	1.05	1.12	1.15	1.14
5		-0.63	-0.66	-0.69	-0.64	0.00	0.09	0.26	0.42	0.49	0.52	0.51
6		-0.72	-0.74	-0.77	-0.72	-0.09	0.00	0.17	0.33	0.40	0.43	0.42
7		-0.88	-0.91	-0.94	-0.89	-0.26	-0.17	0.00	0.16	0.23	0.26	0.25
8		-1.04	-1.07	-1.10	-1.05	-0.42	-0.33	-0.16	0.00	0.07	0.10	0.09
9		-1.11	-1.14	-1.17	-1.12	-0.49	-0.40	-0.23	-0.07	0.00	0.03	0.02
10		-1.14	-1.17	-1.20	-1.15	-0.52	-0.43	-0.26	-0.10	-0.03	0.00	0.02
11		-1.13	-1.16	-1.19	-1.14	-0.51	-0.42	-0.25	-0.09	-0.02	-0.02	0.00
	MAX	0.00	0.05	0.08	0.05	0.69	0.77	0.94	1.10	1.17	1.20	1.19
	MIN	-1.14	-1.17	-1.20	-1.15	-0.52	-0.43	-0.26	-0.10	-0.03	-0.02	0.00
	STD	0.48	0.52	0.54	0.53	0.52	0.52	0.52	0.52	0.52	0.52	0.51

Table 5.2: The measured distance between the scan planes

Location	1	2	3	4	5	6	7	8	9	10	11
R2D1											
1	0.00	4.23	9.33	14.00	18.83	25.90	31.82	37.86	41.63	47.31	53.50
2	4.26	0.00	5.14	9.78	14.63	21.75	27.70	33.82	37.53	43.26	49.51
3	9.41	5.14	0.00	4.63	9.46	16.56	22.50	28.62	32.32	38.04	44.28
4	14.05	9.80	4.66	0.00	4.88	12.03	18.01	24.10	27.86	33.59	39.85
5	18.93	14.68	9.54	4.88	0.00	7.15	13.14	19.23	22.99	28.72	34.99
6	25.74	21.52	16.45	11.79	7.00	0.00	5.87	11.86	15.64	21.28	27.41
7	31.43	27.23	22.19	17.52	12.76	5.82	0.00	5.95	9.74	15.34	21.42
8	37.17	32.97	27.95	23.29	18.57	11.70	5.92	0.00	3.78	9.35	15.38
9	41.22	37.01	31.97	27.30	22.53	15.58	9.75	3.79	0.00	5.61	11.70
10	46.68	42.48	37.45	32.78	28.03	21.12	15.32	9.38	5.59	0.00	6.06
11	52.37	48.18	43.19	38.52	33.81	26.98	21.23	15.33	11.54	5.99	0.00

Location	1	2	3	4	5	6	7	8	9	10	11
R2D2											
1	0.00	5.05	10.02	14.97	19.97	24.92	29.91	34.90	39.84	44.82	49.62
2	5.03	0.00	4.98	9.90	14.90	19.86	24.84	29.82	34.76	39.73	44.53
3	10.01	4.98	0.00	4.92	9.91	14.88	19.86	24.84	29.78	34.75	39.54
4	14.95	9.92	4.95	0.00	4.99	9.95	14.94	19.92	24.86	29.84	34.64
5	19.94	14.94	9.95	4.98	0.00	4.97	9.95	14.93	19.89	24.87	29.68
6	24.92	19.91	14.93	9.94	4.96	0.00	4.99	9.97	14.93	19.91	24.72
7	29.92	24.91	19.94	14.93	9.95	4.99	0.00	4.98	9.95	14.94	19.75
8	34.91	29.91	24.93	19.90	14.93	9.97	4.98	0.00	4.98	9.97	14.79
9	39.86	34.86	29.88	24.87	19.90	14.94	9.95	4.97	0.00	4.99	9.80
10	44.87	39.86	34.89	29.87	24.90	19.93	14.94	9.96	4.99	0.00	4.81
11	49.71	44.69	39.72	34.70	29.72	24.76	19.77	14.78	9.80	4.81	0.00

Location	1	2	3	4	5	6	7	8	9	10	11
R3D											
1	0.00	5.01	9.97	14.97	19.93	24.90	29.89	34.87	39.84	44.82	49.61
2	5.00	0.00	4.97	9.97	14.92	19.89	24.87	29.85	34.83	39.81	44.60
3	9.96	4.97	0.00	5.00	9.94	14.91	19.90	24.88	29.85	34.83	39.62
4	14.96	9.97	5.00	0.00	4.95	9.92	14.90	19.88	24.86	29.84	34.63
5	19.90	14.90	9.94	4.92	0.00	4.98	9.99	14.98	19.98	24.97	29.77
6	24.90	19.89	14.93	9.91	4.98	0.00	5.01	10.00	15.01	19.99	24.80
7	29.93	24.92	19.95	14.93	10.00	5.02	0.00	4.99	10.01	14.99	19.80
8	34.93	29.91	24.95	19.92	15.00	10.01	4.99	0.00	5.03	10.02	14.82
9	39.98	34.96	29.99	24.97	20.03	15.05	10.02	5.03	0.00	4.98	9.79
10	44.97	39.95	34.98	29.96	25.02	20.03	15.01	10.02	4.98	0.00	4.81
11	49.78	44.76	39.79	34.77	29.83	24.84	19.82	14.83	9.79	4.81	0.00

Table 5.3: The difference between the actual distance and the estimated distance betwee each of the scan planes

Location		1	2	3	4	5	6	7	8	9	10	11
	R2D1											
1		0.00	-0.77	-0.67	-1.00	-1.17	0.90	1.82	2.86	1.63	2.31	3.50
2		-0.74	0.00	0.14	-0.22	-0.37	1.75	2.70	3.82	2.53	3.26	4.51
3		-0.59	0.14	0.00	-0.37	-0.54	1.56	2.50	3.62	2.32	3.04	4.28
4		-0.95	-0.20	-0.34	0.00	-0.12	2.03	3.01	4.10	2.86	3.59	4.85
5		-1.07	-0.32	-0.46	-0.12	0.00	2.15	3.14	4.23	2.99	3.72	4.99
6		0.74	1.52	1.45	1.79	2.00	0.00	0.87	1.86	0.64	1.28	2.41
7		1.43	2.23	2.19	2.52	2.76	0.82	0.00	0.95	-0.26	0.34	1.42
8		2.17	2.97	2.95	3.29	3.57	1.70	0.92	0.00	-1.22	-0.65	0.38
9		1.22	2.01	1.97	2.30	2.53	0.58	-0.25	-1.21	0.00	0.61	1.70
10		1.68	2.48	2.45	2.78	3.03	1.12	0.32	-0.62	0.59	0.00	1.06
11		2.37	3.18	3.19	3.52	3.81	1.98	1.23	0.33	1.54	0.99	0.00
	MAX	2.37	3.18	3.19	3.52	3.81	2.15	3.14	4.23	2.99	3.72	4.99
	MIN	-1.07	-0.77	-0.67	-1.00	-1.17	0.00	-0.25	-1.21	-1.22	-0.65	0.00
	STD	1.29	1.46	1.46	1.67	1.86	0.69	1.23	2.02	1.39	1.56	1.85

Location		1	2	3	4	5	6	7	8	9	10	11
	R2D2											
1		0.00	0.05	0.02	-0.03	-0.03	-0.08	-0.09	-0.10	-0.16	-0.18	-0.38
2		0.03	0.00	-0.02	-0.10	-0.10	-0.14	-0.16	-0.18	-0.24	-0.27	-0.47
3		0.01	-0.02	0.00	-0.08	-0.09	-0.12	-0.14	-0.16	-0.22	-0.25	-0.46
4		-0.05	-0.08	-0.05	0.00	-0.01	-0.05	-0.06	-0.08	-0.14	-0.16	-0.36
5		-0.06	-0.06	-0.05	-0.02	0.00	-0.03	-0.05	-0.07	-0.11	-0.13	-0.32
6		-0.08	-0.09	-0.07	-0.06	-0.04	0.00	-0.01	-0.03	-0.07	-0.09	-0.28
7		-0.08	-0.09	-0.06	-0.07	-0.05	-0.01	0.00	-0.02	-0.05	-0.06	-0.25
8		-0.09	-0.09	-0.07	-0.10	-0.07	-0.03	-0.02	0.00	-0.02	-0.03	-0.21
9		-0.14	-0.14	-0.12	-0.13	-0.10	-0.06	-0.05	-0.03	0.00	-0.01	-0.20
10		-0.13	-0.14	-0.11	-0.13	-0.10	-0.07	-0.06	-0.04	-0.01	0.00	-0.19
11		-0.29	-0.31	-0.28	-0.30	-0.28	-0.24	-0.23	-0.22	-0.20	-0.19	0.00
	MAX	0.03	0.05	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	MIN	-0.29	-0.31	-0.28	-0.30	-0.28	-0.24	-0.23	-0.22	-0.24	-0.27	-0.47
	STD	0.09	0.09	0.08	0.08	0.08	0.07	0.07	0.07	0.09	0.09	0.14

Location		1	2	3	4	5	6	7	8	9	10	11
	R3D											
1		0.00	0.01	-0.03	-0.03	-0.07	-0.10	-0.11	-0.13	-0.16	-0.18	-0.39
2		0.00	0.00	-0.03	-0.03	-0.08	-0.11	-0.13	-0.15	-0.17	-0.19	-0.40
3		-0.04	-0.03	0.00	0.00	-0.06	-0.09	-0.10	-0.12	-0.15	-0.17	-0.38
4		-0.04	-0.03	0.00	0.00	-0.05	-0.08	-0.10	-0.12	-0.14	-0.16	-0.37
5		-0.10	-0.10	-0.06	-0.08	0.00	-0.02	-0.01	-0.02	-0.02	-0.03	-0.23
6		-0.10	-0.11	-0.07	-0.09	-0.02	0.00	0.01	0.00	0.01	-0.01	-0.20
7		-0.07	-0.08	-0.05	-0.07	0.00	0.02	0.00	-0.01	0.01	-0.01	-0.20
8		-0.07	-0.09	-0.05	-0.08	0.00	0.01	-0.01	0.00	0.03	0.02	-0.18
9		-0.02	-0.04	-0.01	-0.03	0.03	0.05	0.02	0.03	0.00	-0.02	-0.21
10		-0.03	-0.05	-0.02	-0.04	0.02	0.03	0.01	0.02	-0.02	0.00	-0.19
11		-0.22	-0.24	-0.21	-0.23	-0.17	-0.16	-0.18	-0.17	-0.21	-0.19	0.00
	MAX	0.00	0.01	0.00	0.00	0.03	0.05	0.02	0.03	0.03	0.02	0.00
	MIN	-0.22	-0.24	-0.21	-0.23	-0.17	-0.16	-0.18	-0.17	-0.21	-0.19	-0.40

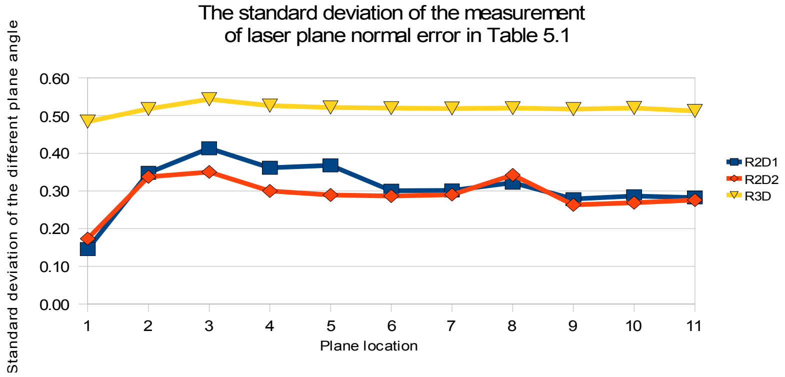


Figure 5.16: The plotting of the standard deviation of the measurement of laser plane normal error in Table 5.1.

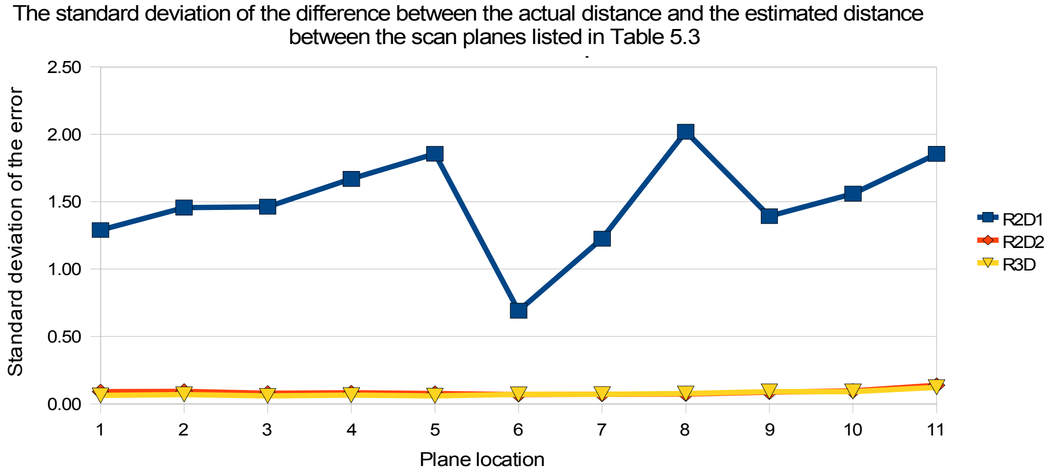


Figure 5.17: The plot of the standard deviation of the difference between the actual distance and the estimated distance between the scan planes listed in Table 5.3.

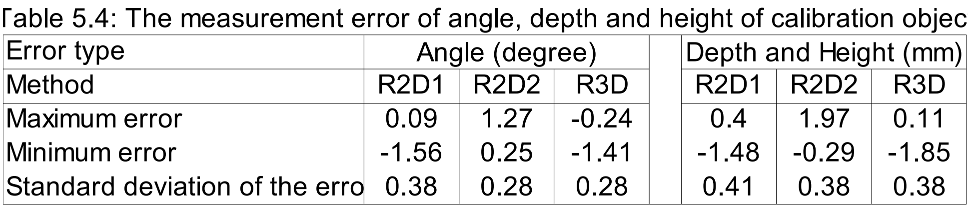


Table 5.4: The measurement error of angle, depth and height of calibration objec

Error type	Angle (degree)			Depth and Height (mm)		
Method	R2D1	R2D2	R3D	R2D1	R2D2	R3D
Maximum error	0.09	1.27	-0.24	0.4	1.97	0.11
Minimum error	-1.56	0.25	-1.41	-1.48	-0.29	-1.85
Standard deviation of the erro	0.38	0.28	0.28	0.41	0.38	0.38

5.6 Conclusion

RANSAC based laser plane pose estimation methods were designed, implemented and critically evaluated with two measurements. It was found that both the R2D2 and R3D methods are almost the same (similar standard deviation) in determining the translated distance between each translated planes in the second measurement of the experiment.

Method R2D2 was experimentally proved to be better compare to R3D due to a smaller standard deviation of measured plane normal produced in the first experiment method. Although method R3D produced a larger standard deviation (~0.5 degree) than method R2D2 (~0.35 degree), however the influence is too insignificant to be observed in other experiments through out the length of 50mm workspace. Method R2D2 is experimentally proved to be more appropriate, in determining the pose of the laser plane, since the method fully makes use of the two dimensional constrains.

Chapter 6: Three Dimensional Reconstruction of Objects Using a Turntable

6.1 Introduction

In our method of reconstructing the surface of an object one needs to illuminate the surface of the object with a laser light source. This process can be termed as scanning. Scanning of an object surface could be either performed by keeping the object stationary and moving the laser light across the object's surface or by keeping the laser light source stationary and moving the object surface across the laser light source. To accomplish the second method one can place the object at the centre of a turntable and rotate the object around the object's axis such that the object's surface is illuminated by the stationary laser light source.

Turntable based 3-D surface reconstruction system consists of three major components, namely a turntable, a camera, and a laser light source. To perform a 3-D surface reconstruction, one needs to know the relative pose between each of these three major components. One way to know the relative pose between them is to manually align them to one another, with respect to a fixed coordinate frame [74].

Figure 6.1 shows the configuration of the components of a turntable based 3-D surface reconstruction system. To know the relative pose between each of the components the following manual operations were performed [74]:

- The camera lens axis was aligned to the Z-axis of the turntable coordinate frame, hence, producing the image plane which was parallel to the XY plane of the turntable coordinate frame.

- The distance between the focal point of the camera and the centre of the turntable, *D*, was measured.

- On the surface of the turntable a line at an angle θ_l to the YZ plane of the turntable was marked and the laser plane was aligned with this line.

- Also the laser plane was made perpendicular to the XZ plane and correlating with the Y-axis of the turntable coordinate frame.

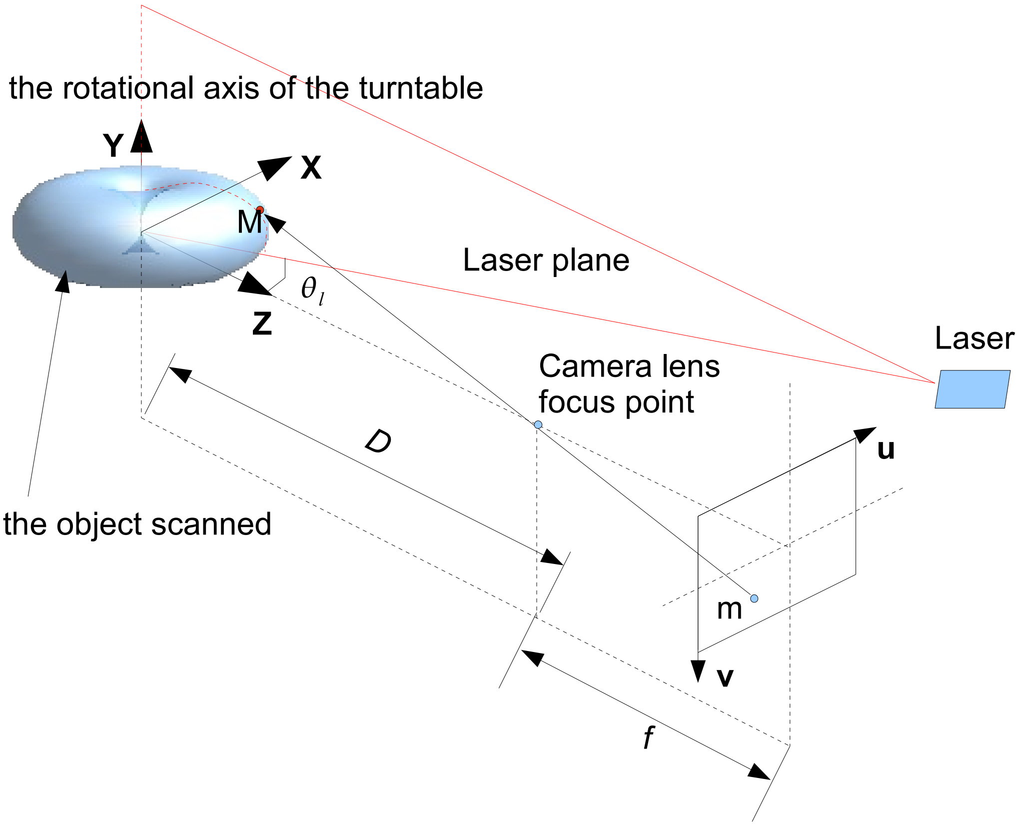


Figure 6.1: The general configuration of the apparatus used in turntable based 3-D surface reconstruction system.

Using the above set up, the location $M(X,Y,Z)$ on the surface of the scanned object, illuminated by the laser, was estimated from the camera image coordinate $m(u,v)$, by using equations (6.1), (6.2) and (6.3).

$$Z=\frac{D\lambda_u u}{u\lambda_u\cos\theta_l-\sin\theta_l} \tag{6.1}$$

$$Y = \lambda_v v (Z \cos\theta_l - D) \tag{6.2}$$

$$X = \lambda_u u (Z \sin\theta_l - D) \tag{6.3}$$

Where λ_u and λ_v is the size of a pixel, in metric, along the image axis u and v, respectively.

The above described method required extensive manual intervention. The manual operation of aligning the camera and the laser light source with each other was laborious, time consuming and error prone. Error introduced during the alignment of the components resulted in inaccurate estimate of the 3-D location of the points lying on the surface of the object. In order to eliminate the error prone laborious manual alignment of the components, a set of auto-calibration procedures, to estimate the pose of the components of the system, were investigated. This chapter elaborates these procedures and demonstrates their accuracy

6.2 The need to know the relative pose of the laser plane and the relative pose of the turntable with respect to the camera

For the turntable based 3-D reconstruction along with a calibrated camera one needs to know the relative pose of the laser plane and the turntable with respective to the camera. The need for this information is explained in detail below.

To reconstruct the 3-D surface of an object, the object was placed on the top of the turntable. Using a laser light emitter a laser plane was shone at the object. Where the laser plane intersected with the surface of the object, being scanned, a laser line of locations illuminated by the laser was formed. A camera image of these locations, illuminated by the laser was captured by the camera. In the camera image, the 2-D locations of the pixels illuminated by the laser were found. These 2-D locations were converted into 3-D coordinates with respect to the camera. This conversion was achieved by means of a triangulation process. For this process one needs to know the camera calibration parameters and the relative pose of the laser plane with respect to the camera.

To estimate the 3-D coordinates of the rest of the locations on the surface of the object, the laser plane was made to intersect on those locations. This was achieved by keeping the camera and the laser plane static while the object was rotated by rotating the turntable. As the turntable was rotated, different locations on the surface of the object were illuminated by the laser. Using the camera images of the newly illuminated locations, the 3-D coordinates of these locations with respect to the camera were estimated using the triangulation process. Making use of these set of 3-D coordinates, the 3-D surface of the object was reconstructed.

For this reconstruction, it was necessary to know the relative angular displacement between the set of 3-D coordinates. To estimate this angular displacement, the relative pose of the turntable with respect to the camera, the location of the centre of the turntable and the rotational axis were needed to be known.

Hence, for the 3-D reconstruction of the surface of the object using the turntable, the relative pose of the laser plane and the relative pose of the turntable with respect to the camera were required. The relative pose of the laser plane with respect to the camera was estimated as discussed in the chapter *Laser Plane Pose Estimation* (Chapter 5). In this chapter, the estimation of the relative pose of the turntable with respect to the camera, the location of the centre of the turntable and the rotational axis are discussed in detail in the following sections.

The estimation of the relative pose of the turntable with respect to the camera, the location of the centre of the turntable and the rotational axis is termed as "*turntable calibration*". In this study two different methods were used for the turntable calibration. In the first method, a checker pattern planar object was used and in the second method the turntable calibration was performed by using the features or markers on the surface of the turntable. These two methods of calibrating the turntable is discussed in detail in the following two sections.

6.3 Turntable calibration using checker pattern planar object

The calibration planar object comprised of two planes having a common intersecting side. The calibration object was carried out by folding a checker pattern planar object along the common intersecting side. This calibration object is referred as folded checker board (Figure 6.2a).

In this method, to calibrate the turntable, first the folded checker board was placed, on the top of the turntable, such that the common intersecting side of the two planes was vertically at the top of the centre of rotation of the turntable. It was assumed that the axis of rotation of the turntable runs along the common intersecting side of the two planes. So, to resolve the rotational axis of the turntable only the common intersecting side of the two planes needed to be resolved.

The orientation and the location of the common intersection line side of the two planes were estimated as follows. The folded checker board was represented by two planes, left hand side plane (P1) and the right hand side plane (P2) (Figure 6.2b).

The plane P1 was represented by a point $A[a_1,a_2,a_3]^T$ and its plane unit normal vector $\vec{u}$ $[u_1,u_2,u_3]^T$ and the plane P2 were defined by a point $B[b_1,b_2,b_3]^T$ and its plane unit normal vector $\vec{v}$ $[v_1,v_2,v_3]^T$. The intersection of the two planes was represented by the equations (6.4) and (6.5); where, the *x, y* and *z* in equations (6.4) and (6.5) are the coordinates of the point *p(x,y,z)* along the intersection of the two planes.

$$u_1(x-a_1)+u_2(y-a_2)+u_3(z-a_3)=0 \tag{6.4}$$

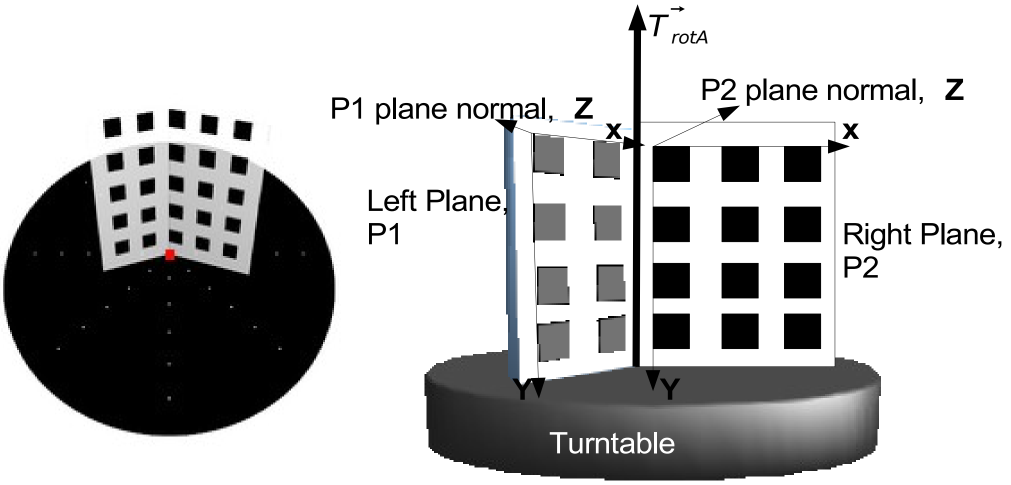
$$v_1(x-b_1)+v_2(y-b_2)+v_3(z-b_3)=0 \tag{6.5}$$

Where the point *p(x,y,z)* is any point along the intersection of the two planes, there is a specific point, where the XY plane of the camera intersects the axis of rotation (Figure 6.3). If we resolve this point of intersection with respect to the camera coordinate system then the value of *y* at this location will be equal to zero. Making the value of *y* equal to zero in the equations (6.4) and (6.5), the *x* and *z* values are determined using the equations (6.6) and (6.7).

$if\ y=0$

$$z=\frac{v_1(u_1a_1+u_2a_2+u_3a_3)-u_1(v_1b_1+v_2b_2+v_3b_3)}{v_1u_3-u_1v_3} \tag{6.6}$$

$$x=\frac{v_1b_1+v_2b_2+v_3(b_3-z)}{v_1} \tag{6.7}$$



(a) *(b)*

Figure 6.2: Figure (a) illustrating the calibration object, folded checker board, placed at the centre of the turntable.

Figure (b) illustrating the notations for the different planes of the folded checker board and the location of the rotational axis of the turntable.

Thus, the rotational axis of the turntable was characterised by the 3-D point *p* and the unit vector $\vec{T}_{rotA}$ of the intersection of the two planes. The value of $\vec{T}_{rotA}$ is given by the equation (6.8). In the equation 6.8 the normal of the two intersecting planes, $\vec{u}$ and $\vec{v}$ was estimated by estimating the pose of the two planes using equation 3.13 (Refer Section 3.4.1, Planar object pose estimation). The third column of the rotation matrix of the estimated planar pose is the plane normal of the planar object.

$$\vec{T_{rotA}}=\frac{CrossProduct(\vec{u},\vec{v})}{\sin(\theta)} \tag{6.8}$$

Where

$$\theta=\cos^{-1}(\vec{u}\cdot\vec{v})$$

$$\vec{u}=[u_1 u_2 u_3]^T,\ [{}^{camera}pose_{P1}]_{4x4}=\begin{bmatrix}(1,1) & (1,2) & (u_1) & (1,4)\\(2,1) & (2,2) & (u_2) & (2,4)\\(3,1) & (3,2) & (u_3) & (3,4)\\(4,1) & (4,2) & (4,3) & (4,4)\end{bmatrix}_{4x4}$$

$$\vec{v}=[v_1 v_2 v_3]^T,\ [{}^{camera}pose_{P2}]_{4x4}=\begin{bmatrix}(1,1) & (1,2) & (v_1) & (1,4)\\(2,1) & (2,2) & (v_2) & (2,4)\\(3,1) & (3,2) & (v_3) & (3,4)\\(4,1) & (4,2) & (4,3) & (4,4)\end{bmatrix}_{4x4}$$

The accuracy of estimating the orientation and location of the rotational axis of the turntable using the above described method was verified. The verification was performed by simulating a turntable by creating an artificial image in POV-ray [75] with the known turntable rotational axis ($\vec{T_{rotA}}$) and the points on the rotational axis (*p).* The estimated results were subsequently compared with the actual values as shown in Table 6.1. The following paragraph explains the process of the simulation.

The scene in the POV-ray was created with the virtual camera, a virtual turntable and a virtual folded checker board. The creation of the virtual objects in the scene is detailed below.

The properties of the virtual camera are listed in the Table 6.2. The 3-D coordinate frame of the camera was taken the same as the world 3-D coordinate frame.

The turntable 3-D coordinate system was centred at the centre of the turntable with the XY plane on the surface of the turntable and the Z-axis perpendicular to the turntable. Initially, the turntable 3-D coordinate frame was aligned with the camera coordinate frame. Subsequently, the virtual turntable was rotated around the X-axis of the camera coordinate frame by 50 degrees and translated by 32 units along the camera Z-axis. Thus, the turntable normal *i.e.* its the rotational axis, was oriented [*0.0000, 0.7660, 0.6428*]^T with respect to the world coordinates which was the same as the camera coordinate frame. And the turntable rotational axis intersected the camera XY plane at

$[0.0, 0.0, 32.0]^T$.

Two planar checker board objects (P1 and P2) were created to simulate the virtual folded checker board for the turntable calibration process. To ease the creation of the virtual scene, the P1 plane was created by attaching a three dimensional coordinate frame to the bottom right corner of the plane, with the XY plane along the surface of the planar object and the Z-axis perpendicular to it. To start with, the planar object coordinate frame was aligned with the camera coordinate frame. Subsequently, the P1 plane was firstly rotated around the Y-axis by 20 degrees and -40 degree around the camera X-axis. After the rotation the planar object was translated by 32 units along the camera Z-axis. This placed the planar object perpendicular to the surface of the turntable (*i.e.* form a right angle between the turn-table and the planar object).

The P2 plane was created similar to the P1 planar object except for the following differences :

- The three dimensional coordinate frame was located to the bottom left corner of the plane
- The P2 plane was firstly rotated around the Y-axis by -20 degrees

To ease the process of determining the poses of the planar planes, the coordinate frames of the two planar object (P1 and P2) were re-located from the initial locations (bottom right and bottom left) to the locations shown in Figure 6.2b.

Correspondences of four known locations on the planar objects were used to estimate the pose of each of the planar object. The four known locations were the four corner locations. Both the ground truth locations of the corners as well as the corners' locations estimated by the corner detection processes were used. The difference between the ground truth corners' location with the detected corners are shown at the beginning of the table. The difference between the estimated pose of the planar object (found by using the estimated corner locations with their corresponding grid coordinate, Section (3.4.1), Planar object pose estimation) and the ground truth poses are listed at the centre of the table. The difference between the estimated and the ground truth turntable pose parameters (p, $\vec{T}_{rotA}$) are listed at the bottom of the table.

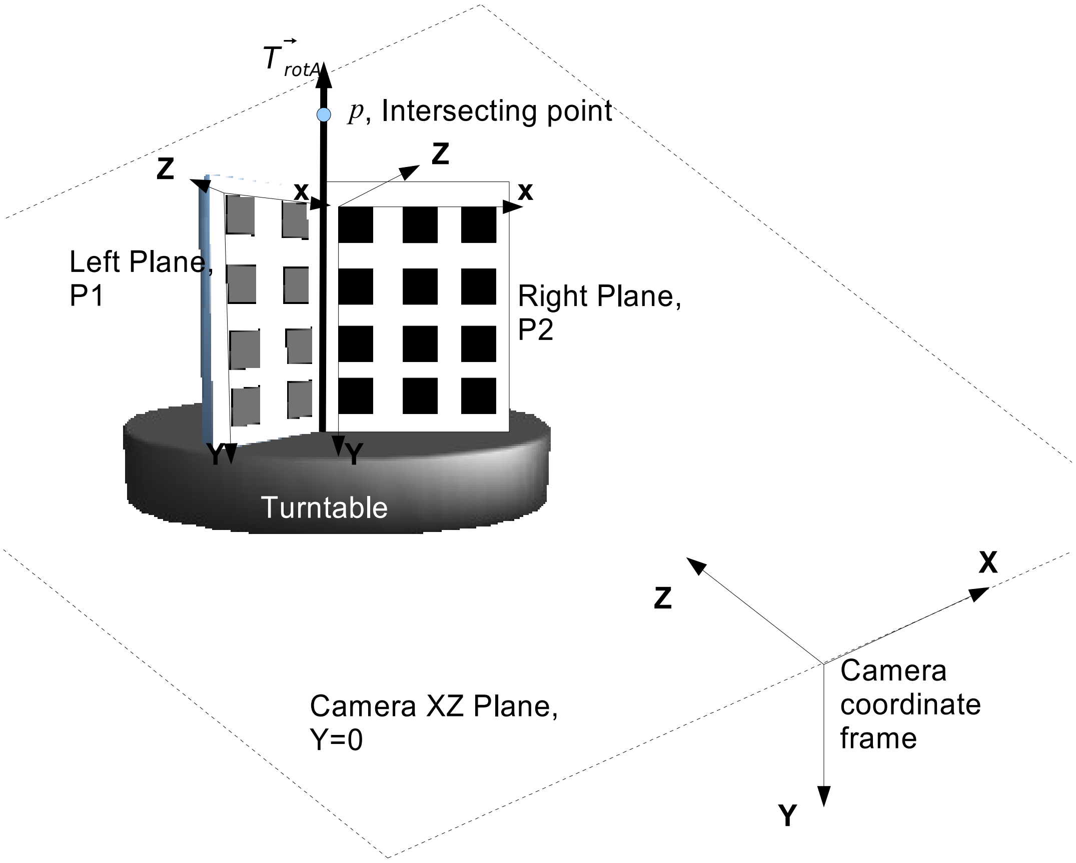


Figure 6.3: Intersection between camera plane at Y=0 with turntable rotational axis.

Table 6.1: Comparison of turntable calibration result between the ground truth and detected from artificial images.

Plane	**Corners,(grid coordinate)**							
	(0,0)		(3,0)		(3,7)		(0,7)	
Left Plane, P1								
Ground Truth	250.096	142.381	303.040	133.363	305.494	231.569	260.473	241.795
Detected	250.335	142.019	303.031	133.783	305.162	231.942	260.203	241.403
Error	0.239	0.362	0.009	0.420	0.332	0.373	0.270	0.392
Right Plane, P2								
Ground Truth	354.257	136.309	408.277	145.511	395.057	245.323	349.258	234.919
Detected	354.003	136.741	407.953	145.079	395.358	244.781	349.639	235.434
Error	0.254	0.432	0.324	0.432	0.301	0.542	0.381	0.515

Plane Pose	**P1**				**P2**			
Ground Truth	0.9397	0.0000	-0.3420	-3.7588	0.9397	0.0000	0.3420	1.8794
	-0.2198	0.7660	-0.6040	-5.2490	0.2198	0.7660	-0.6040	-5.6887
	0.2620	0.6428	0.7198	25.8097	-0.2620	0.6428	0.7198	26.3337
	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
Detected								
	0.9445	-0.0001	-0.3286	-3.7769	0.9437	0.0004	0.3308	1.8885
	-0.2092	0.7710	-0.6015	-5.3003	0.2095	0.7731	-0.5987	-5.7040
	0.2534	0.6368	0.7282	26.0126	-0.2560	0.6343	0.7295	26.4891
	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000	1.0000
Information from Pose:								
Rotation at X-axis (degree)								
Ground Truth	-40.000				-40.000			
Detected	-39.559				-39.375			
Rotation at Y-axis								
Ground Truth		20.000				-20.000		
Detected		19.182				-19.317		
Rotation at Z-axis								
Ground Truth			0.000				0.000	
Detected			-0.006				0.026	
Translation (unit)								
Ground Truth								
X				-3.759				1.879
Y				-5.249				-5.689
Z				25.810				26.334
Detected								
X				-3.777				1.888
Y				-5.300				-5.704
Z				26.013				26.489
Rotation Error	-0.441	0.818	0.006		-0.625	-0.683	-0.026	
Translation Error								
X				0.018				-0.009
Y				0.051				0.015
Z				-0.203				-0.155

	$\|\vec{u}\|$			$\|\vec{v}\|$		
Ground Truth	-0.3420	-0.6040	0.7198	0.3420	-0.6040	0.7198
Detected	-0.3286	-0.6015	0.7282	0.3308	-0.5987	0.7295
Error	-0.0134	-0.0025	-0.0083	0.0112	-0.0053	-0.0096
	$A(a_1,a_2,a_3)$			$B(b_1,b_2,b_3)$		
	a_1	a_2	a_3	$b1$	$b2$	$b3$
Ground Truth	-3.7588	-5.2490	25.8097	1.8794	-5.6887	26.3337
Detected	-3.7769	-5.3003	26.0126	1.8885	-5.7040	26.4891
Error(same as erro of translation)	0.0181	0.0514	-0.2029	-0.0091	0.0153	-0.1554
Rotational axis	$\vec{T}_{rotA}$			$p(x,y,z)$		
				x	y	z
Ground Truth	0.0000	0.7660	0.6428	0.0000	0.0000	32.0000
Detected	-0.0020	0.7727	0.6348	-0.0831	0.0000	32.0677
Error	0.0020	-0.0066	0.0080	0.0831	0.0000	-0.0677

Table 6.2: The properties of the virtual camera

Camera Type		Perspective		Render Mode			
	x	y	z	Size			
Location	0.0000	0.0000	0.0000		Width:		640
Sky	0.0000	1.0000	0.0000		Height:		480
Direction	0.0000	0.0000	1.0000	Quality			
Right	1.3333	0.0000	0.0000		Quality:	Mode 9	
Up	0.0000	1.0000	0.0000		Anti-aliasing:	Enabled	
Look at	0.0000	0.0000	1.0000			Method	Non-recursive
						Threshold	0.3
						Depth	2

6.4 Turntable calibration by using the features marked on the turntable

The major short coming of the turntable calibration using folded checker pattern planar object is the need to place the common axis of the two planes vertically above the centre of the turntable. This to ensure that the rotational axis of the turntable lies along the line of intersection of the two planes (See Section 6.3). This manual process is time consuming, laborious and error prone. One way to assist the human operator, in placing the common axis of the two planes vertically above the centre of the turntable, is to mark turntable with concentric circles and radial lines passing through the centre of the turntable. The human operator could make use of the grooves of the radial lines as guides to place the common axis of the two planes vertically above the centre of the turntable.

A much better solution is to completely eliminate the need for the planar object. This could be achieved by making use of the turntable itself as a planar object for the calibration of the turntable. The following sections (Section 6.4 and 6.5) discusses two such methods where the turntable is used as the calibration planar object.

6.4.1 Turntable calibration by marking regular grid points on the turntable

In this method, regular grid points are marked on the surface of the turntable. Making use of these known locations the axis of rotation and the centre of rotation of the turntable were estimated.

In this process, it is assumed that the centre of rotation of the turntable does not change while the turntable rotates. But the pose of the rotational axis might change as the turntable rotates. That is, the surface of the turntable might not rotate on the same plane. To address this issue, the pose of the rotational axis is estimated for every angular interval (every 20 degrees say)

Knowing the actual locations of the regular grid points marked on the planar surface of the turntable and their corresponding image coordinates, a planar homography was estimated (See section 3.3.2.1). Making use of this planar homography, the pose of the planar plane, *i.e.* the surface of the turntable, was estimated (See section 3.4.1). Once the pose of the turn table was estimated, the centre point of the turntable (p), on the estimated plane, could be found by projecting the assigned grid coordinate of the centre point to the camera coordinate system by using equation (3.22).

The regular grid points on the turntable were the points on the concentric circles marked at regular angular interval. The grid points were marked on the turntable by making use of standard carpentry tools (calliper/protractor) commonly available in general mechanical workshop. An image of the real turntable with grid points marked on it is shown in Figure 6.4.

Figure 6.4: An image of the surface of the turntable. White markers were systematically installed on the platform. Lines were intersecting at the centre of the turntable.

To find the grid coordinates of the points on the turntable, the inner most concentric circle was used. The points were marked at a regular angular interval of 45 degrees. Using a calliper, the radius of the innermost concentric circle was measured accurately and was found to be 40mm.

Planar homography can be formed by considering any set of four points on the regular grid. Figure 6.5 shows the grid coordinate frame marked in green on top of the turntable. The planar homography was formed (Section 3.3.2.1) making use of the four points ($pp1, pp2, pp3, pp4$) as shown in Figure 6.5. The homogeneous grid coordinates of the four points in camera calibration unit are given in equation (6.9).

Once the planar homography was determined, the pose of the virtual plane, created by the four markers, was estimated (Section 3.4.1). The 3-D coordinate of the centre of the circle, p was determined from the last column of the estimated virtual plane pose, which is the translation of the origin of the turntable coordinate frame to the origin of the camera coordinate frame. The rotational axis which is perpendicular to the surface of the turntable, parallel to the Z-axis of the homography coordinate frame was extracted from the third column of the detected pose of the coordinate frame.

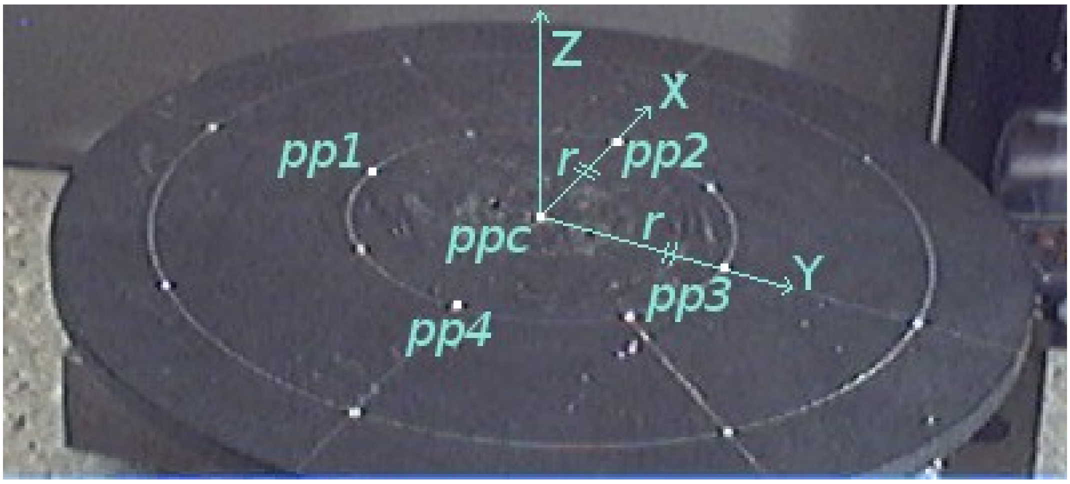


Figure 6.5: A chosen grid coordinate system of the turntable.

$$\begin{aligned} pp1 &= [0, -r, 0, 1]^T \\ pp2 &= [r, 0, 0, 1]^T \\ pp3 &= [0, r, 0, 1]^T \\ pp4 &= [-r, 0, 0, 1]^T \end{aligned} \tag{6.9}$$

Where r is the radius of the circle used.

Using a simulated image, as a scene in the POV-ray, with known parameters of the turntable, the above implementation was tested. The simulated environment for the turntable was the same as the one discussed in Section 6.3 (See Section 6.3 Turntable calibration using checker pattern planar object), except when the folded checker board was removed and instead the grid locations were marked on the turntable. The centre of the turntable expressed in the camera coordinate frame is point *p(x,y,z)*.

The validation was carried out as follows:

Using four known grid locations and the correspondences of the points with the locations in the camera image (found by blob locations, the centre of gravity of the blob of white pixels), the homography, was found (Using equation (3.5)).

Using the homography, the pose of the virtual plane having the four marker locations was estimated (equation (3.17)). This was the pose of the turntable.

From the pose estimation, the centre and the axis of rotation of the turntable were extracted (explained previously, five paragraphs above, in this section).

The point *p(x,y,z)* with *y* is equal to zero was subsequently resolved using equation (5.20) by projecting the point *p* along the rotational axis vector $\vec{T}_{rotA}$ to the plane with the origin coordinate $[0.0,0.0,0.0]^T$ at default plane normal,Y-axis $[0.0,1.0,0.0]^T$. The results obtained making use of the above implementation are provided in Table 6.3.

Table 6.3: Comparison of turntable calibration result between the ground truth and detected from artificial images by using markers on checker board.

Corners,(grid coordinate)

Plane	*pp1*		*pp2*		*pp3*		*pp4*	
Homography plane on turn table								
Ground Truth	245.267	191.963	394.733	191.963	418.143	303.085	221.857	303.085
Detected	244.500	191.500	394.500	191.500	417.500	302.500	221.500	302.500
Error	0.767	0.463	0.233	0.463	0.643	0.585	0.357	0.585

Plane Pose	P1			
Ground Truth	1.0000	0.0000	0.0000	-5.6569
	0.0000	0.6428	0.7660	-3.6362
	0.0000	-0.7660	0.6428	36.3334
	0.0000	0.0000	0.0000	1.0000
Detected				
	1.0000	0.0004	0.0000	-5.7287
	-0.0003	0.6499	0.7600	-3.6800
	0.0003	-0.7600	0.6499	36.4207
	0.0000	0.0000	0.0000	1.0000
Information from Pose:				
Rotation at X-axis				
Ground Truth	50.000			
Detected	49.466			
Rotation at Y-axis				
Ground Truth		0.000		
Detected		0.000		
Rotation at Z-axis				
Ground Truth			0.000	
Detected			0.023	
Translation				
Ground Truth				
X				-5.6569
Y				-3.6362
Z				36.3334
Detected				
X				-5.7287
Y				-3.6800
Z				36.4207
Rotation Error	0.534	0.000	-0.023	
Translation Error				
X				0.072
Y				0.044
Z				-0.087

Rotational axis	$\vec{T}_{rotA}$			$p(x,y,z)$		
				x	y	z
Ground Truth	0.0000	0.7660	0.6428	0.0000	0.0000	32.0000
Detected	0.0000	0.7600	0.6499	-0.0626	-0.0381	31.9737
Error	0.0000	0.0060	-0.0071	0.0626	0.0381	0.0263

$p(x,y,z)$ along vector the Rotational axis with y equal to zero

	x	y	z
Ground Truth	0.0000	0.0000	32.0000
Detected	-0.0623	0.0000	32.0057
Error	0.0623	0.0000	-0.0057

6.4.2 Turntable calibration using X-corner features on a planar object of irregular checker pattern

An issue faced in the turntable calibration using features marked on the turntable was that the detection of the markers suffered from perception distortion. That is the size of the marker, appearing in the image, varied depending upon how far it was away from the camera. That was one of the main reason it was found that the accuracy of the marker based turntable pose estimation, discussed in section 6.4.1, in an idle simulated camera image, was slightly less compared to the folded checker board method discussed in section 6.3.

Hence, instead of markers on the surface of the turntable it was decided to make use of a patterned checker board struck flat on the surface of the turntable. The X-corner detection methodology, made use of the detected corner features (Section 4.6.3), to handle the perspective distortion (of the corner sizes). To eliminate the process of manually select the extreme corners, an irregular checker board pattern was designed (Figure 6.6a). Using this an algorithm can automatically estimate the extreme corners (Figure 6.14).

A regular planar pattern, *e.g.* a symmetric checker board pattern (Figure 6.2) does not give a unique corner orientation when the pattern was rotated around its axis. Hence, to determine the pose of the checker board, some initial information, *e.g.* setting the top left corner as the origin of the pattern, need to be provided to the machine vision algorithm. In contrast an irregular planar pattern, with a unique orientation, makes it possible for the machine vision algorithm to automatically determine the correspondence of the features in the camera image (namely corners) with the planar object (Figure 6.14). Planar homography can be estimated by making use of this identified correspondences. Once the planar homography was estimated, the pose of the planar object was found by using the method discussed in planar object pose estimation in Section 3.4.1.

Still the camera image of a irregular planar pattern can be distorted due to perspective distortion. Before one tries to get a match of the camera image, with the actual pattern, it is necessary to rectify the camera image for perspective distortion. The rectification was performed by estimating the perspective distortion parameters; using which the camera image was rectified. The actual perspective distortion might be resolved by repeatedly rectifying the camera image. Using the rectified camera a set of accurate correspondences were subsequently identified. The planar homography can be estimated by making use of this more correctly identified correspondences. Once the planar homography was estimated, the pose of the planar object was found by using the method discussed in planar object pose estimation in Section 3.4.1.

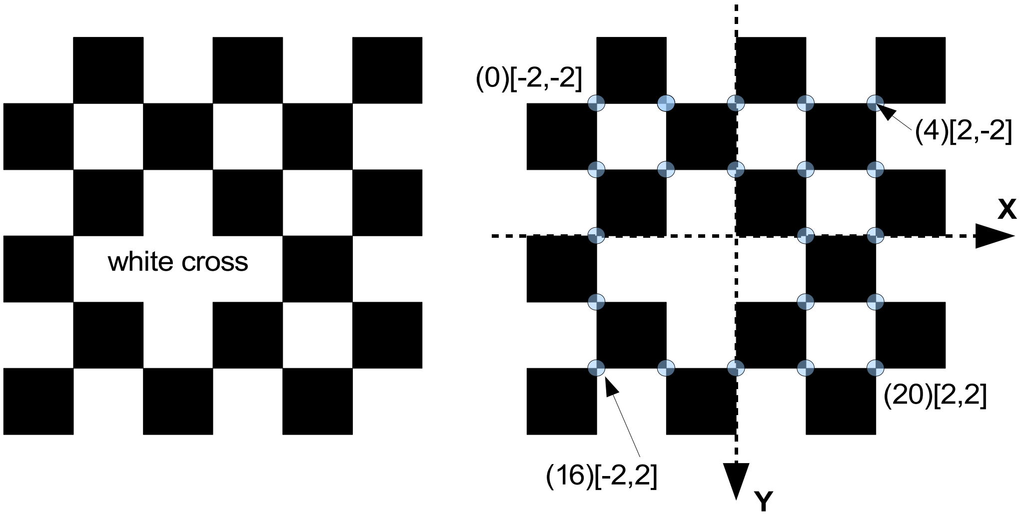
The following sections will discuss in detail the process of perspective distortion rectification (Within Section 6.4.2.2) and the robust estimation of the planar homography (Section 6.4.2.1).

Making use of the irregular pattern, the pose of the turntable was estimate as described below:

The irregular planar object was placed flat on the turntable. The turntable was rotated at regular angular interval and the pose of the turntable was estimated. The reason for estimating the pose of the turntable at different angular orientation was to handle the possibility that the turntable might not rotate at a perfect plane due to non-centric rotational axis and/or non-vertical rotational axis. The estimated poses of the planar object at multiple locations help in handling these errors. The design of non-self-similar planar pattern and the process of automatic turntable pose calibration are discussed in detail below.

6.4.2.1 Design of the irregular planar pattern having X-corner features

A minimum of four correspondences are needed to create a planar homography. But the more the number of correspondences are used, the more robust will be the estimation of the planar homography. Hence, an irregular planar pattern with twenty one X-corners was created as shown in Figure 6.6a. A grid coordinate frame was assigned, to the irregular planar object, with the origin of the coordinate frame centred at the centre of the pattern. To ease the process of turntable pose estimation, the origin of the grid coordinate frame was aligned to the centre of rotation of the turntable (Figure 6.6b).



(a) *(b)*

Figure 6.6: (a) Figure showing the designed irregular planar pattern.

(b) Figure showing the location of the grid coordinate frame assigned to the planar object and the grid coordinate assigned to each of the X-corners. All the X-corners are circled.

6.3.2.2 The process of automatic turntable pose calibration

To accurately estimate the pose of the planar object, and hence the pose of the turntable, one need to know the perspective distortion parameters. To determine these parameters, the correct grid coordinates of each of the salient features, on the planar object, need to be corresponded with the correct projected image coordinates of the features. The salient features of the planar object, in this case are the twenty one X-corner locations.

The correspondence of the grid coordinates of the X-corner locations with their corresponding projected image coordinates were performed as follows:

Initially, all the prominent features in the image (Figure 6.7) were identified with a general feature detector, such as the Shi-Tomasi [76] feature detector. The result of this detection process is shown in Figure 6.8. Next, to pick up only those features that are of interest (*i.e.* the X-corners), inter-image subtraction technique was used. The inter-image subtraction technique made use of another image where the features of interest had changed to another location. For example, this could be the camera image of the turntable platform rotated by a constant step angle (Figure 6.9). The inter-image subtraction was performed by firstly subtracting the next image with the current image. In the resulting image (Figure 6.10), if the change of the pixel intensity value was larger than a threshold value, it was considered that there was a possibility that the feature of interest was located at that pixel location. This way all the extraneous features points were eliminated from consideration. The result of thresholding, where the image pixel location with very little difference between the current image and the next image were thresholded and binarised into white colour is provided in Figure 6.11. The image pixels with high difference were subsequently thresholded and binarised into black colour. The detected features, produced by using Shi Tomasi feature detector, were superimposed into the binarised image and shown in Figure 6.12. The features points lying at the white region of the binarised image were subsequently removed.

The designed irregular planar pattern consisted of both the L and X-corners which were successfully detected by the Shi-Tomasi feature detector. These corner features remained detected even after the process of inter-image subtraction (since all the planar features points were rotated together). Within a small window of interest centred at each of the location of the features, detected above, an X-corner detector proposed by Chen (Section 4.6.4), was subsequently used to pick up the actual twenty one X-corners in the image of the irregular planar pattern. At each feature location, the X-corner detection process yielded two results, namely the refined X-corner image coordinate and the X-cornerness of the X-corner. From this list of X-corners with their corresponding X-cornerness twenty-one of the X-corners, with highest X-cornerness measurements, were identified as the X-corners. The result of this operation is shown in Figure 6.14.

Once the twenty one X-corners were successfully identified, in the image, the next step was to identify the corresponding grid coordinate of each of the identified X-corner in the image. This was performed by identifying the corners at the top left most and right most and bottom left most and right most corners. These corners, located at the extreme points, were identified by finding those four corners which form the largest area spammed by the polygon.

Once the extreme four X-corners were identified, the four X-corners were sorted to be located in a clock-wise direction/orientation. Making use of these four extreme corners, a trial and error method was adapted to estimate the perspective distortion parameters. The trial and error process was carried out as follows. Making use of the correspondence between the four sorted X-corner image coordinate points and the four grid coordinate points arranged in clock wise ([-2,-2], [2,-2], [2,2], [-2,2]) (Figure 6.6b), the planar homography was estimated together with the corresponding perspective distortion parameters. Applying the estimated perspective distortion parameters the image coordinates of the planar object was corrected. The corrected image was binarised and a black and white image of the perspective distortion corrected planar object image was formed. To find out whether the estimated planar homography was accurate, the area within the white cross (Figure 6.6a) was estimated, by counting the number of white pixels. If the number of white pixels were within a thresholded value, (say 80% of the expected value), then the estimated planar homograhy was accepted. Else the process was repeated by cycling the four sorted points, with maximum up to four trials. The four possible results of perspective distortion corrected planar object images were shown in Figure 6.13.

Before proceeding with the process of estimating the pose of the turntable, each of the detected X-corners needs to be assigned the corresponding grid coordinate. This was performed by projecting the grid coordinates onto the image making use of the estimated planar homography. The nearest X-corner with respect to the projected grid coordinate was assigned the grid coordinate. An index, corresponding to each of the grid coordinate was assigned to each of the corresponding X-corner as shown in Figure 6.14.

Once the correspondences were identified, the estimation of the perspective distortion was refined by estimating a planar homography by making use of all the twenty one correspondences.

Thus, the pose of the planar pattern lying on the surface of the turntable was estimated. Hence, the pose of the turntable (p, $\vec{T}_{rotA}$), was estimated by using the same method as discussed in Section 6.4.1 (Turntable calibration by marking regular grid points on the turntable).

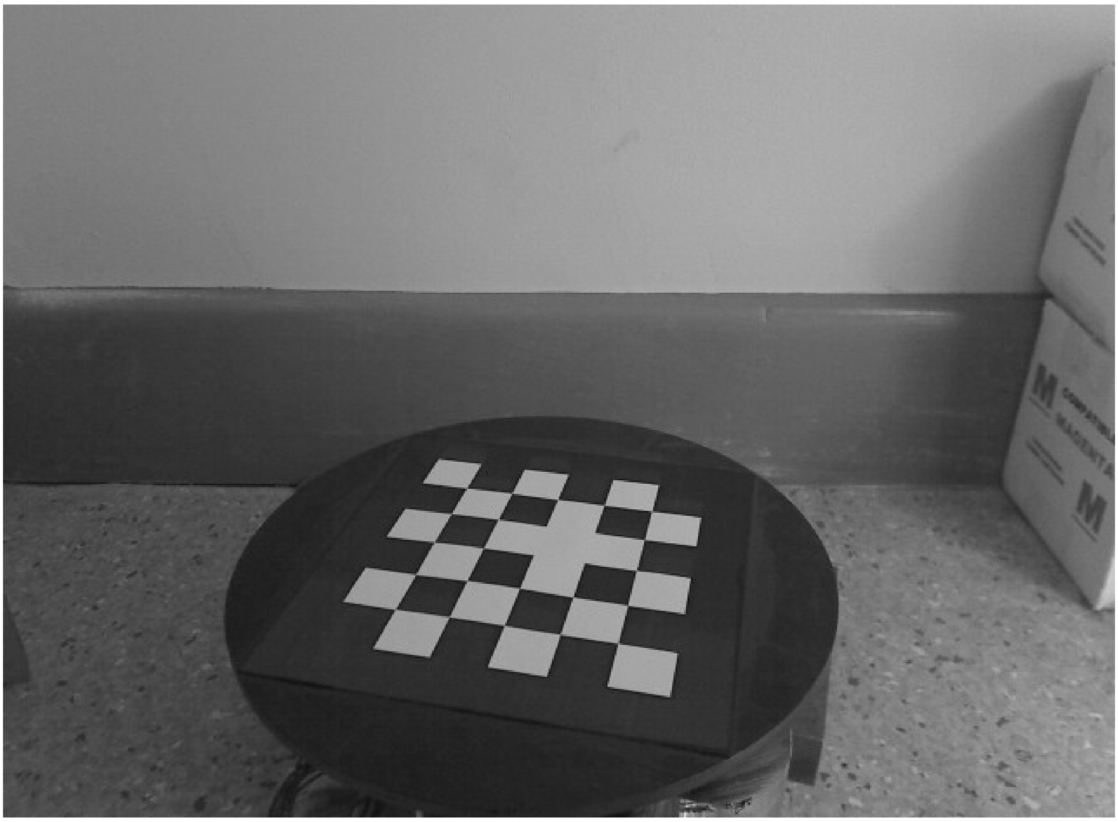


Figure 6.7: Figure showing the camera image of the irregular pattern attached to the surface of the turntable.

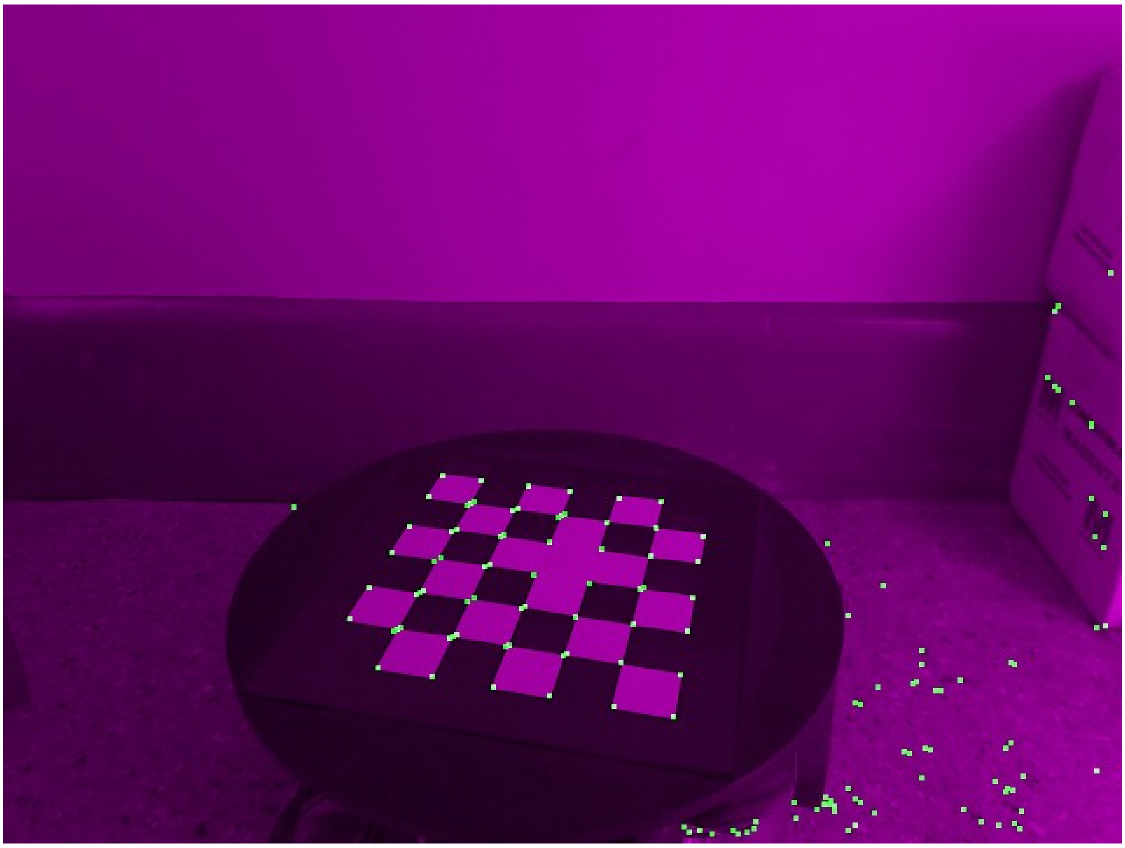


Figure 6.8: The result of Shi-Tomasi feature detector.

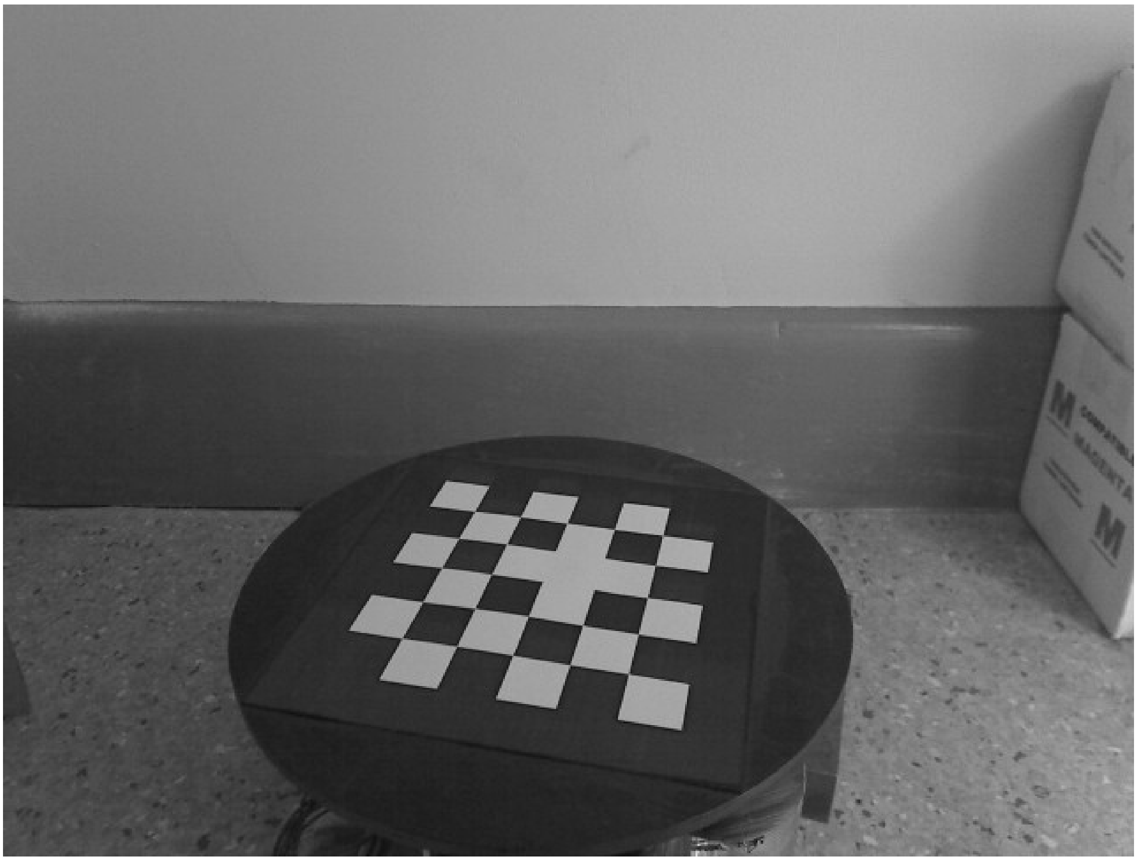


Figure 6.9: The image of the turntable rotated to difference location.

Figure 6.10: The result of the image subtraction between Figure 6.7 and Figure 6.9.

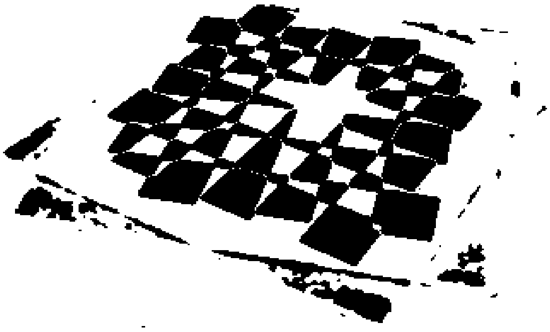


Figure 6.11: Result of binarisation of Figure 6.10.

Figure 6.12: The result of Shi Tomasi feature detector (Figure 6.8) was superimposed onto the binarised image. The background features lying at the white region were subsequently removed.

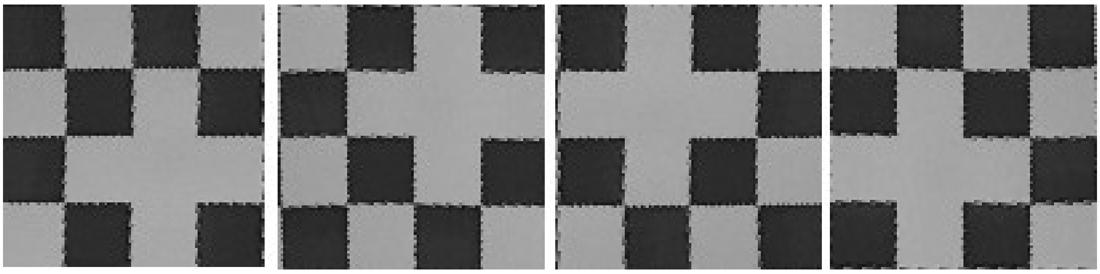


Figure 6.13: The cropped images four possible perspective distortion corrected planar pattern. The right most image is the correct detected planar pattern (Figure 6.6a).

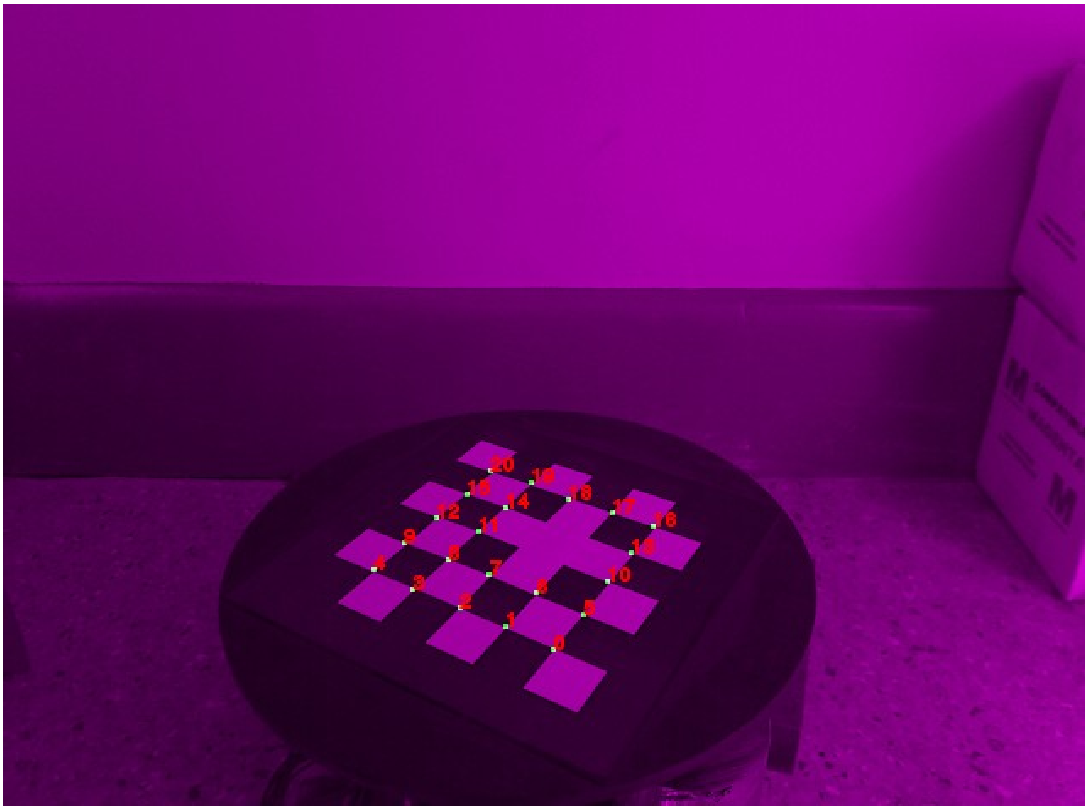


Figure 6.14: Image showing the detected salient features. All the detected twenty one X-corners are highlighted in yellow.

A unique index (from 0 to 20) is assigned to each of the X-corner, corresponding to the X-corner's location in the grid coordinate.

6.5 Robust turntable pose estimation based on multiple estimates

To estimate the turntable pose robustly, the turntable pose was estimated at regular angular intervals and subsequently global optimisation was applied to the estimated parameters to arrive upon a consensus value of the pose parameters. This process is explained in detail in this section as follows.

Using the process "*Turntable calibration using X-corner features on a planar object of irregular checker pattern*" (Section 6.4.2), the pose of the turntable was estimated at regular angular interval as the turntable was rotated around its central axis. From the list, of the estimated turntable pose parameters, the locations of the centre of rotation, was extracted. From this list of the location of the centre of rotation the euclidean distance between each of the location with the rest of the locations were found. For each turntable, angular placement of the mean of the euclidean distances between its location of the rotational axis, with the rest of the locations, of the rotational axis, for other angular placements of the turntable, was found [Table 6.4].

The turntable was rotated and the pose of the turntable was estimated at a regular angular interval of eighteen degrees. Table 6.4 lists the euclidean distance difference between the location of the centre of axis of rotation for each of the turntable angular position and the rest of its angular position. The mean difference along with their standard deviation were plotted and shown in Figure 6.15. From this plot, the pose estimation at angular position 260^0 (20x13) and 340^0(20x17), having the largest mean difference, were found to be outliers. Removing the pose parameters of these two angular positions, of the turntable, the pose parameters from the rest of the angular positions were considered as a list of consensus poses.

Removing the outlier poses, that had the largest mean difference, yielded a list of consensus poses. The process of global optimisation was carried out as follows.

Firstly, for each of the poses (in the list of consensus poses) the four extreme grid coordinates ([-2,-2],[-2,2],[2,2],[2,-2]), of the designed planar pattern, were projected to the camera coordinate frames. The result of this process was a set of 3-D coordinate points distributed across the plane of rotation of the turntable.

Assuming that the turntable was a perfect planar object, a plane was fitted to the set of 3-D coordinate points by using Moment of Inertia Analysis (MIA) global optimisation process (Section 5.5.2.3). The result of Moment of Inertia Analysis yielded two information about the best fitted plane, namely the normal of the plane and a 3-D point lying on the plane.

Since the planar object was firmly fitted onto the turntable, the plane normal yielded by MIA was the global optimised plane normal of the turntable.

The 3-D point was the average of all the four extreme grid coordinates, of the planar object, at different rotational pose of the turntable. This average was in fact the centre of the planar object. Since the centre of the planar object was physically made to correlate with the centre of the turntable, thus the centre of the turntable, *p*, was estimated.

The estimated centre of rotation, *p*, of the turntable, was subsequently multiplied with the corresponding metric unit for a single unit of the designed planar pattern, in millimetre.

Table 6.4 Euclidean distance different between each measured turntable centre point

LOC	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	0.00	6.46	3.51	1.58	0.24	3.79	5.60	3.98	2.20	1.58	2.98	0.42	10.80	2.96	0.77	2.24	9.96	5.35
2	6.46	0.00	2.95	4.89	6.25	10.25	12.07	10.44	8.65	8.03	3.52	6.17	17.26	9.40	5.74	8.68	3.50	1.14
3	3.51	2.95	0.00	1.94	3.30	7.30	9.11	7.49	5.71	5.08	0.65	3.23	14.31	6.46	2.80	5.73	6.45	1.85
4	1.58	4.89	1.94	0.00	1.36	5.37	7.18	5.55	3.77	3.15	1.42	1.32	12.38	4.52	0.89	3.80	8.39	3.78
5	0.24	6.25	3.30	1.36	0.00	4.01	5.82	4.19	2.41	1.79	2.75	0.25	11.02	3.17	0.57	2.46	9.75	5.14
6	3.79	10.25	7.30	5.37	4.01	0.00	1.82	0.30	1.61	2.23	6.76	4.10	7.01	0.92	4.52	1.62	13.75	9.14
7	5.60	12.07	9.11	7.18	5.82	1.82	0.00	1.67	3.43	4.05	8.57	5.91	5.21	2.71	6.34	3.42	15.57	10.95
8	3.98	10.44	7.49	5.55	4.19	0.30	1.67	0.00	1.79	2.41	6.93	4.27	6.83	1.08	4.70	1.81	13.93	9.33
9	2.20	8.65	5.71	3.77	2.41	1.61	3.43	1.79	0.00	0.62	5.15	2.49	8.61	0.78	2.92	0.30	12.15	7.54
10	1.58	8.03	5.08	3.15	1.79	2.23	4.05	2.41	0.62	0.00	4.53	1.86	9.23	1.38	2.29	0.70	11.53	6.92
11	2.98	3.52	0.65	1.42	2.75	6.76	8.57	6.93	5.15	4.53	0.00	2.67	13.76	5.90	2.25	5.18	7.00	2.42
12	0.42	6.17	3.23	1.32	0.25	4.10	5.91	4.27	2.49	1.86	2.67	0.00	11.10	3.23	0.46	2.52	9.66	5.06
13	10.80	17.26	14.31	12.38	11.02	7.01	5.21	6.83	8.61	9.23	13.76	11.10	0.00	7.86	11.52	8.59	20.76	16.14
14	2.96	9.40	6.46	4.52	3.17	0.92	2.71	1.08	0.78	1.38	5.90	3.23	7.86	0.00	3.66	0.74	12.89	8.28
15	0.77	5.74	2.80	0.89	0.57	4.52	6.34	4.70	2.92	2.29	2.25	0.46	11.52	3.66	0.00	2.94	9.23	4.62
16	2.24	8.68	5.73	3.80	2.46	1.62	3.42	1.81	0.30	0.70	5.18	2.52	8.59	0.74	2.94	0.00	12.17	7.55
17	9.96	3.50	6.45	8.39	9.75	13.75	15.57	13.93	12.15	11.53	7.00	9.66	20.76	12.89	9.23	12.17	0.00	4.62
18	5.35	1.14	1.85	3.78	5.14	9.14	10.95	9.33	7.54	6.92	2.42	5.06	16.14	8.28	4.62	7.55	4.62	0.00
MAX	10.80	17.26	14.31	12.38	11.02	13.75	15.57	13.93	12.15	11.53	13.76	11.10	20.76	12.89	11.52	12.17	20.76	16.14
STD	3.10	4.18	3.44	3.09	3.11	3.76	4.00	3.79	3.41	3.24	3.31	3.10	4.80	3.57	3.09	3.41	4.80	3.91
MEAN	3.58	6.97	4.88	3.96	3.58	4.69	6.08	4.82	3.90	3.74	4.58	3.60	10.69	4.22	3.68	3.91	10.07	6.10

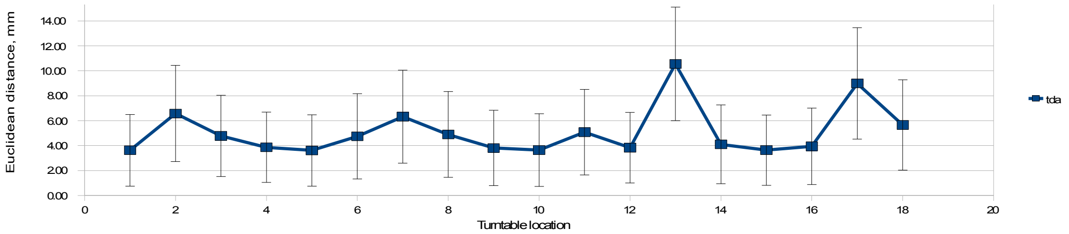


Figure 6.15: The plotting of mean different between the measured location of the turntable centre point and the standard deviation.

6.6 Verifying the accuracy of the turntable pose estimation

The pose of the turntable, (p, $\vec{T}_{rotA}$), was initially estimated with the designed method (Section 6.5). To verify the above estimation the following physical verification was performed. Firstly the platform of the turntable was replaced with a calibration object having a very sharp tip (Figure 6.16a). The calibration object, was placed in such a way that the tip physically lie along the rotational axis of the turntable (Figure 6.16b). The laser plane was made to intersect the sharp tip and the pose of the laser plane was estimated (Section 5.2.4.1.2.2).

Using both the estimated laser plane pose parameters and the image pixel location of the tip, illuminated by the laser, the 3-D location of the tip, p_t, (Figure 6.16b), was determined using the triangulation method (Section 3.4.3.1). Using the two locations p (from the pose of the turntable estimation) and p_t, (3-D location of the tip) a vector, $\vec{V}_t$, was estimated. Theoretically the angle, a_t, between this estimated vector $\vec{V}_t$ and the vector, $\vec{T}_{rotA}$, got from the turntable pose estimation should be equal to zero. Using equation (3.24) it was found that a_t is very close to zero (0.1 degree).

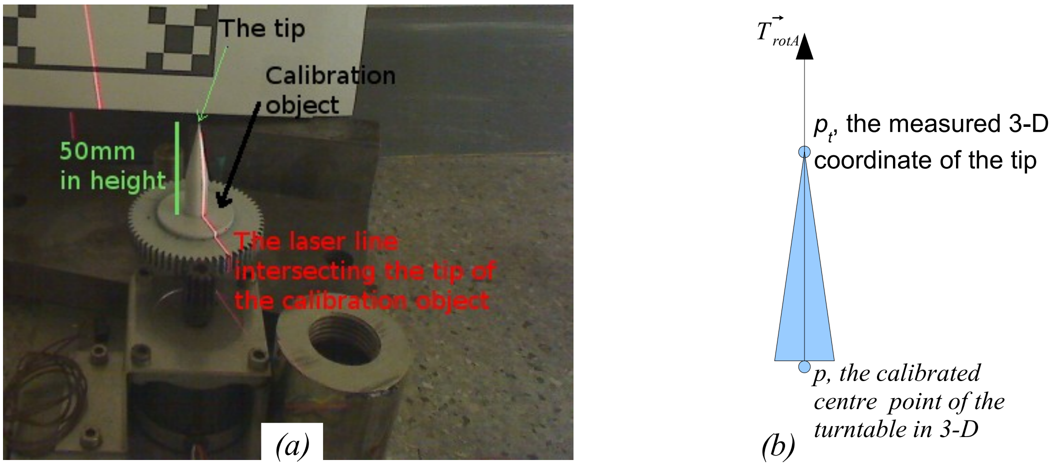


Figure 6.16: (a) The laser plane was made intersected at the tip of the calibration object. (b) The location of the measured 3-D coordinate of the tip should lie on the calibrated rotational axis, $\vec{T}_{rotA}$.

6.7 Three-dimensional surface reconstruction using the estimated turntable pose parameters

The object whose surface needed to be reconstructed was placed on the turntable. The object surface was scanned by making use of a laser strip light as the turntable was rotated at regular angular intervals. At each rotational step, an image of the laser line illuminated points on the object was captured. The image location of the pixels illuminated by the laser, with very high red colour image intensity values, were segmented by thresholding the red channel of the RGB colour space. The segmented image pixels, illuminated by the laser, were subsequently triangulated (Section 3.4.3.1) to create the list of 3-D coordinate points with respect to the camera coordinate frame. These set of 3-D coordinate points with respect to the camera coordinate frame were resolved/rotated to their corresponding angular location (θ_{tt}), on the surface of the object. This transformation was performed as follows: The set of 3-D locations were first translated to the centre of location of the camera, as the origin. This was carried out by deducting the location of the estimated turntable centre point, *p,* from each of the 3-D coordinate points. Subsequently the 3-D coordinate points were rotated by the angular displacement by which the turntable was rotated. The rotation was performed by multiplying the translated set of 3D coordinate points with a three by three rotation matrix, using the known Rodrigues' rotation formula [77], as shown in equation (6.10). The parameters of the three by three rotation matrix was generated from the estimated parameters of the turntable rotational axis, $\vec{T}_{rotA}$, and the angle, (θ_{tt}), to which the turntable was rotated. After the 3-D points rotation, they were translated back to the camera coordinate frame by adding the estimated turntable centre point, *p,* to each of the location.

$$\begin{bmatrix} \cos\theta_{tt}+w_x^2(1-\cos\theta_{tt}) & w_x w_y(1-\cos\theta_{tt})-w_z\sin\theta_{tt} & w_y\sin\theta_{tt}+w_x w_z(1-\cos\theta_{tt}) \\ w_z\sin\theta_{tt}+w_x w_y(1-\cos\theta_{tt}) & \cos\theta_{tt}+w_y^2(1-\cos\theta_{tt}) & -w_x\sin\theta_{tt}+w_y w_z(1-\cos\theta_{tt}) \\ -w_y\sin\theta_{tt}+w_x w_z(1-\cos\theta_{tt}) & w_x\sin\theta_{tt}+w_y w_z(1-\cos\theta_{tt}) & \cos\theta_{tt}+w_z^2(1-\cos\theta_{tt}) \end{bmatrix} \tag{6.10}$$

Where

$$\vec{T}_{rotA}=\begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix}$$

6.8 Result and discussion

The implemented turntable pose estimation algorithm using X-corner features on a planar object of irregular checker pattern (Section 6.4.2) was used to calibrate a turntable. The calibrated turntable was subsequently used to perform 3-D surface reconstruction of a plastic cap in white colour (Figure 6.17). The cap was installed around the centre of the turntable. A laser line emitter was cast to the object and the pose of the laser plane was estimated (Section 5.2.4.1.2.2). Table 6.5 provides the system calibration parameters. The turntable was designed with 1000 steps per revolution. 100 images were captured by systematically rotating the turntable with 10 steps per image, 3.6 degree angular interval. The captured images (Figure 6.17) were thresholded (Figure 6.18a) and sub-pixel operator (Section 3.4.2.2.2) was applied to each row of the detected pixels illuminated by the laser (Figure 6.18b). The result of the sub-pixel operator, the detected pixel illuminated by the laser with up to sub-pixel accuracy was triangulated (Section 3.4.3.1) and subsequently rotated to the corresponding location (Section 6.6). All the acquired 3-D points of the scanning object were loaded into Meshlab for visualisation (Figure 6.19). The cloud of 3-D points was subsequently processed by Powercrust to generate a polygonal mesh and plotted in Meshlab (Figure 6.20a). A close up look of the object being scaned was shown in Figure 6.21b.

Table 6.5 System parameters for 3-D surface reconstruction system

Camera Intrinsic parameters		
537.752	0.000	325.406
0.000	534.896	219.324
0.000	0.000	1.000

Laser plane pose estimation					
$\vec{n}$	[	0.928	-0.196	0.318	$]^T$
m	[	-6.397	-29.222	372.188	$]^T$

Turntable pose estimation					
$\vec{T}_{rotA}$	[	-0.019	0.879	0.477	$]^T$
p	[	22.917	43.819	329.955	$]^T$

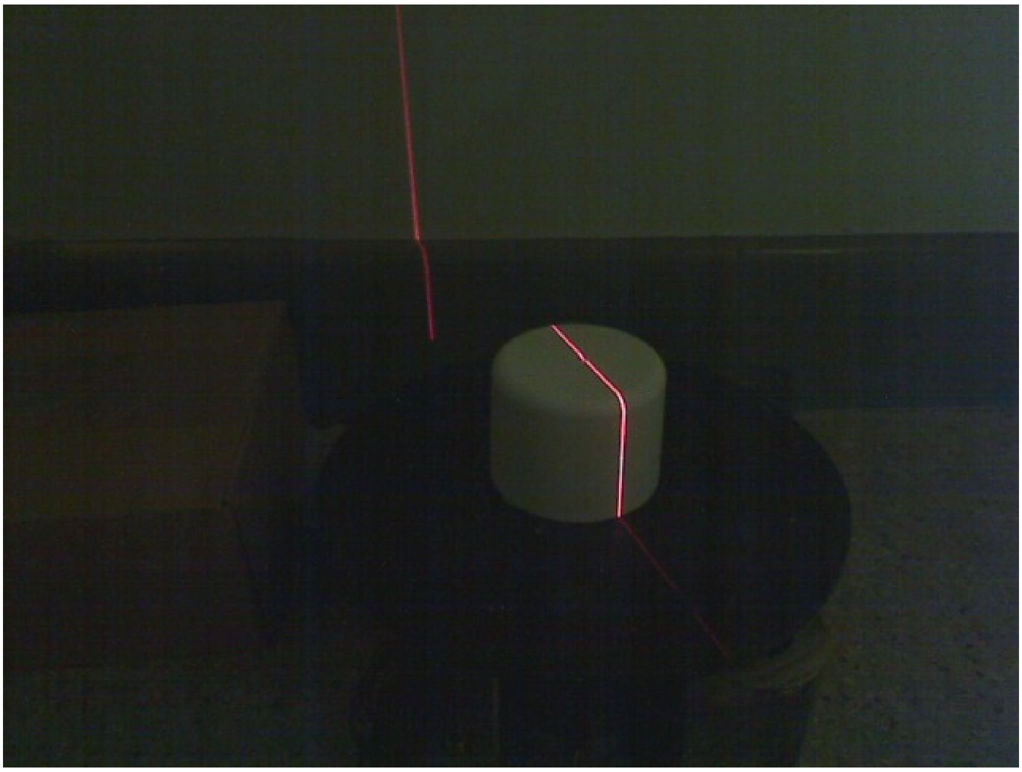
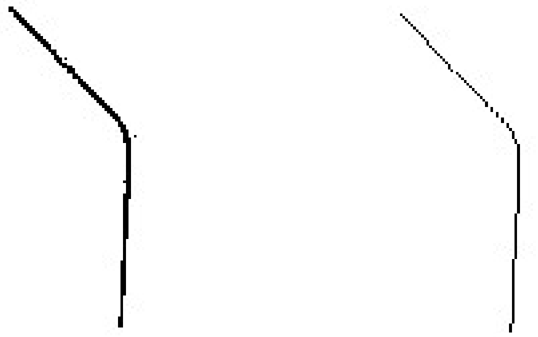


Figure 6.17: The white colour cap illuminated by the laser.



(a) *(b)*

Figure 6.18:(a) The cropped image of the process of thresholding applied on Figure 6.17. (b) The result of sub-pixel operator was rounded and rendered.

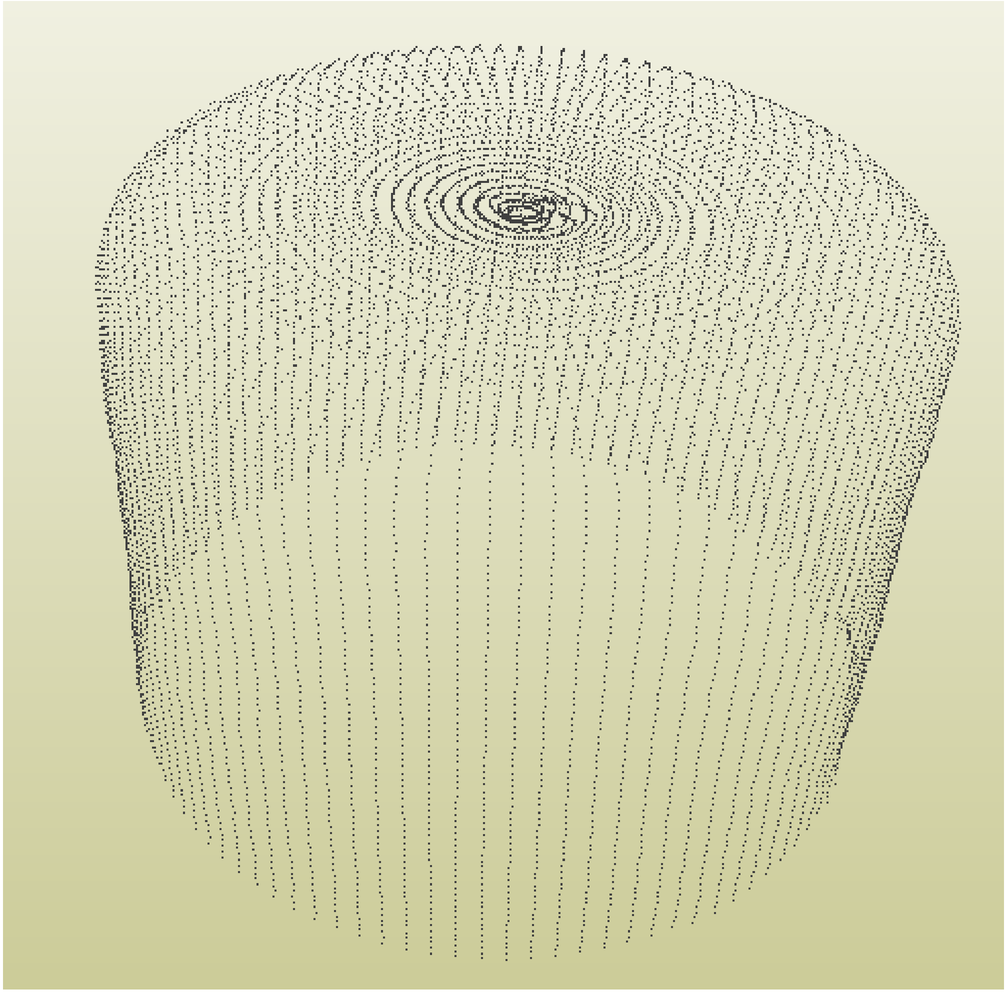


Figure 6.19: Visualising the acquired 3-D points with MeshLab.

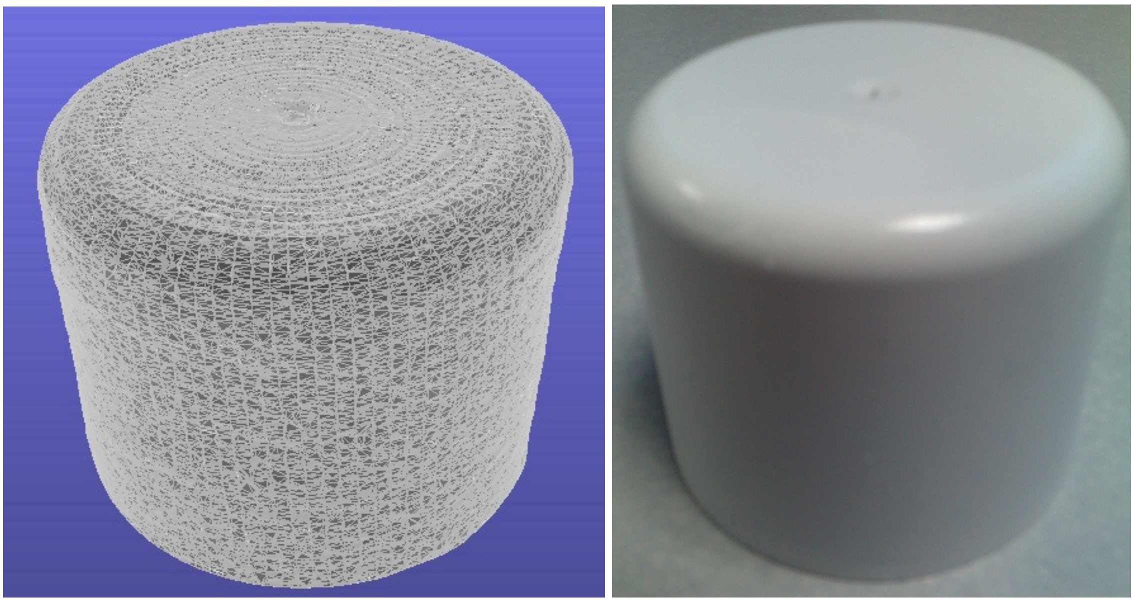


Figure 6.20: (a) The created polygonal mesh with Powercrust [37].(b) The object scanned (container cap in white colour).

6.9 Conclusion

A number of turntable pose estimation techniques were proposed in this study and they were simulated with POV-ray [75]. Of these techniques the irregular checker planar pattern with X-corner features was chosen as the best (Section 6.4.2).

Using the chosen technique, the pose of the turntable (p, $\vec{T}_{rotA}$) was estimated as follows. Multiple estimates of the pose parameters of the turntable, was estimated by rotating the turntable at multiple angular steps, around the centre of the turntable. The estimated poses that had a large difference with the mean of the other estimations were discarded. The final pose parameters of the turntable was subsequently estimated with the remaining poses, using a global optimisation method (Section 6.5).

The estimated pose parameters of the turntable was evaluated for its accuracy as outlined below.

Using a calibration object, the rotational axis of the turntable was estimated (Section 6.6). The rotational axis, estimated using the calibration object, and the rotational axis, derived from the pose estimation $\vec{T}_{rotA}$, was found almost aligned. The angular difference between the rotational axis, estimated using the calibration object, and the rotational axis, derived from the pose estimation $\vec{T}_{rotA}$, was found to be 0.1 degrees.

Using the result of the turntable calibration, the 3-D locations on the surface of a regular object (cylinder) was estimated. The plotted 3-D locations was found to be visually accurate (multiple concentric circles were formed).

Using the developed method, the turntable was calibrated with minimal manual intervention. The only manual effort involved, in the calibration, was to precisely stick the designed irregular planar pattern onto the turntable, such that it's origin correlates with the centre of the turntable.

Chapter 7: Development of a Vision Based Respiration Monitoring System

7.1 Introduction

In this chapter an overview of different types of non-contact approaches to monitor respiration rate in infant is provided. The objective of this work was to develop a vision based, non-contact, multi sensory respiration monitoring system to determine the rate of breathing in infants in a clinical environment.

A number of physiological changes occur during breathing. These include: chest and abdomen movements and changes in the surface skin temperature around the tip of the nose. The major considerations in developing the system were its patient safety, measurement accuracy and reliability, cost effectiveness and ease of use.

A commonly used method to monitor respiration rate is by attaching thermistors under the subject nose to measure temperature changes caused by inhalation and exhalation of air. However, as this method requires the sensor to be attached to the subject, it causes discomfort. Also, the thermistors can be used only once for hygiene reasons (cost implication) and can also interfere with the process of breathing. Infants can also detach the sensors causing the system to malfunction. Therefore, methods that allow respiration rate to be measured remotely have distinct advantages.

In two different studies, chest movements, produced during respiration process were measured with infra-red and ultrasonic sensors [78,79]. A basic non-contact respiration monitoring system, consisting of an embedded system with either the ultrasonic or infra-red sensor was developed. The system's software displayed the measured distance of the sensor to the chest (thus chest movements), transferred the captured data in real-time through the embedded system connected to a PC via Universal Serial Bus (USB). The ultrasonic sensor measured the time difference for the sound signal to travel from the sensor to the chest and its reflection back to the sensor. As the speed of sound in air is known, and the delay time can be accurately measured, then the distance of the sensor to the chest can be computed using, the measure, distance = 0.5*(delay time*signal speed). The infra-red based distance sensor, consisted of three major units, namely an infra-red light-emitting diode (LED), a Position Sensitive Device (PSD), and a digital signal processing unit. The PSD composed of a linear array of infra-red photodiode. The infra-red signal emitted by the infra-red LED, was reflected by the chest, and was measured by the PSD. Based upon the location of the incidence of the reflected signal, on the PSD, the distance of the chest, away from the sensor, was measured by the digital signal processing unit, using the principle of triangulation. A number of laboratory tests using manikins were carried out to evaluate these two types of respiration monitoring systems. These indicated that the infra-red sensor based approach was less effective than the ultrasonic sensor based approach [78, 79]. The main problem with the infra-red sensor based approach was its high sensitivity to optical noise. The ultrasonic sensor could detect chest movements to within 0.5 mm accuracy, but required the subject under test to be still.

In order to overcome the limitations of the infra-red and the ultrasonic based sensors, vision based approaches to directly detect chest movements were investigated. A monocular vision system was implemented by using one webcam. This required a small marker acting as a planar pattern to be attached to a manikin's chest. Manikins had to be used during the development stages as they facilitated laboratory developments. They could be manually pumped, making their chest to move, thus simulating a respiration process. The respiration rate of the manikins was monitored in real time. This study highlighted that the vision algorithm needed to be sufficiently fast to track the chest movements.

Another vision based approach, involved using the addition of a second camera. This eliminated the need for the chest marker. The aim was to use the principle of triangulation to monitor the chest movements. Based on this approach, a pilot system was developed [80]. This created a stereo-vision system, consisting of a USB web-cam, one off-the-shelf Firewire camera, and the necessary software that ran on a PC. The system was able to track and estimate the 3-D location of a single salient feature of the cloth worn by the manikin and then determined the respiration rate. The operation was performed in real-time and involved a manual thresholding method of the captured images to detect the feature [80]. The details of the system are provided in the following sections.

7.2 Motivation

Respiration rate is a physiological measure used by the clinicians to detect and monitor a varied range of medical conditions. According to respiratory experts, ninety five percent of the patients can be identified as high risk, with up to twenty four hours in advance by using the measurement of the change of respiratory rate. Hence, the ability to accurately measure respiratory rate can significantly help reducing the morbidity or mortality of the patients and significantly reduce financial cost for the NHS.

Guidelines were published by the National Institute for Clinical Excellence (NICE) in 2007, stating that measurement of respiratory rate is obligatory in the assessment of children with fever in the hospital setting [2]. At least one assessment of respiratory rate needs to be carried out for every child visiting the emergency department. The result of the respiratory rate measurements were subsequently used to monitor the changes in the state of health of the child. The respiratory rate of every child on the wards and intensive care units (ICUs) needs to be measured at least every few hours [2]

7.3 Existing methods

A critical review of non-invasive respiratory monitoring system was produced by Folke *et. al.* (2003) [81] Folke *et. al* (2003), categorised respiration monitoring system into three types: (i) approaches based on detection of movement, volume and tissue composites, (ii) airflow sensing approaches, (iii) blood gas content measurements. Folke *et. al* (2003) have recommended the use of miniature gas sensors to monitor the exchange gas during respiration. A study to measure the carbon dioxide in expiratory air with a canola based carbon dioxide sensor system was conducted by Folke *et. al.* (2005) [82]. The disturbance created by the canola was reported to affect subjects' breathing.

A review of both contact and non contact based approaches to monitor respiration rate is provided in the next sections.

7.3.1 Contact based respiratory measurement systems

Thermistor is a heat sensor, that is attached very near to the nostril to measure the change of temperature caused by inhaling and exhaling air. A thermistor is made from the process of sintering of sulphides, solenoids or oxides of nickel, manganese, cobalt, copper, iron or uranium [83]. The resistance of the created mixture is changed with respect to the temperature. The corresponding resistance (R_T) is calculated with respect to temperature (T) , in Kelvin, a reference temperature of the material (T_{ref}), and the reference resistance of the material (R_{ref}) is shown in Equation 7.1.

$$R_T = R_{ref} e^{(-\beta(1/T - 1/T_{ref}))} \tag{7.1}$$

where β is a factor that depend upon the composition of the mixture, normally supplied by the manufacture.

The strain gauge is an extensometer, in a belt-like form, strapped around the thorax and abdomen, measuring the change of strain of the thorax and abdomen during its expansion (inhaling air) and contraction (exhaling air). It is made from extendible/deformable conducting material, either a very fine wire or thin foil, such that the conductivity can be maintained during the stretching process. The principles of the strain gauge sensor are the conductivity and geometry of the conductor. The area of the conductor is increased during the process of stretching, causing an increase in the resistance of the conductor. The change of resistance (ΔR) is subsequently converted into the measured strain (ε), as shown in equation (7.2). The belt-like strain gauge has to be strapped properly around the thorax and abdomen to maximise the sensitivity of the device.

$$\varepsilon = \frac{\Delta R}{R_G} \tag{7.2}$$

Where R_G is the resistance of the strain gauge without any deformation.

The change of resistance of the sensor can be measured by using a Wheatstone bridge as shown in Figure 7.1. A Wheatstone bridge consists of four resistors $(R_1, R_2, R_3 \text{ and } R_s)$. The resistance of the resistors $R_1, R_2 \text{ and } R_3$ is constant. R_s has a variable resistance and that can be either the thermistor or strain gauge. The output voltage (V_{out}) of the bridge varies according to the change of the resistance of the sensor and hence a small electronic signal is generated. This signal is amplified and converted to digital format by an Analogue to Digital Converter (ADC).

The digital signal is plotted versus time on a computer screen. By looking at the plot, a clinician estimates the time-dependent respiratory parameters (*e.g.* respiratory rate per minute). A combination of two strain gauges, one strapped at the rib cage and one at the abdomen, allows measurement of independent compartment of rib cage and abdomen [84]. This provides the volume-dependant respiratory parameters (*e.g.* volume of respiratory at a particular time, V_T , and the rate of volume changing with respect to time, V_T/T_1 [85]).

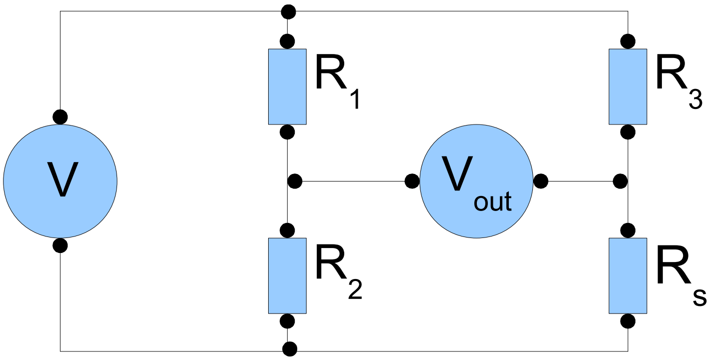


Figure 7.1: The configuration of the Wheatstone bridge and the sensor.

Photoplethysmograph (PPG) has also been used for measuring the respiratory rate. It was suggested that during the process of inhaling air, the pressure within the thoracic cage decreases and hence, the right atrium expands allowing faster filling from the vena cave. The ventricular pre-load increases and the amount of blood pumped during the cardiac contraction increases. This effect can be observed during exhaling process, where the heart is compressed and the cardiac efficiency is reduced, hence, reducing the stroke volume. Nevertheless, the overall net result of respiration, the frequency and the depth of respiration, can be considered as a pump to the entire cardiovascular system. The cardiac output increases proportionally to the rate and depth of respiration [86].

The PPG signal can be collected by attaching a probe, consisting of a infra-red light emitting diode (LED) and a photo-detector, to a finger, ear, or forehead. Infra-red wavelength generated by the infra-red LED illuminates the skin. The emitted infra-red wavelength is more strongly absorbed by blood and weakly by the tissue. Hence, the change of blood volume, affecting the constant rate of emitted infra-red energy from the emitter to the photo-detector, can be observed by the photo-detector.

Ogawa *et. al.*(1999) [87] implanted a probe with one photo-detector and four infra-red LED (wavelength 880nm), waterproof with silicon rubber, into the bottom of the bathtub for collecting the PPG signal during shower. The PPG signal was subsequently filtered with an analogue low-past filter with a cut-off frequency of 0.5Hz. Both the respiratory and heart rates were successfully measured simultaneously. Depending on the probe’s contact surface area with the skin, the photo-detector could be configured to receive the reflective signal (forehead) or transmitted signal (finger) [88].

Electrocardiogram (ECG), represents the electrical activity of the heart. It is recorded using skin electrodes. This signal provides another mean to monitor respiration rate. The ECG-Derived-Respiration (EDR) technique makes use of an existing ECG recording device, to derive the respiratory rate. A technique to obtain the respiratory rate from ECG is by measuring the fluctuation in the mean cardiac electrical axis [89]. An open source free software to measure the respiratory rate derived from the ECG signal is available from the reference site [90].

Surface Electromyography (sEMG) has also been examined as a mean to monitor respiration rate. The sEMG was used to measure the activation signal of the respiratory muscle (*e.g.* intercostal muscle). The ECG artifact in the collected signal was removed by an adaptive filter to enhance the signal to noise ratio of the measured signal [91].

An idea to create a miniaturised, wearable, and battery-operated monitoring system to monitor breathing, using microphone was proposed by Corbishley and Rordriguez-Villegas (2008) [92]. The omnidirectional microphone was mounted to an aluminium conical bell and was attached to the skin to measure the acoustic breathing signal produced from a human body. A low power consumption system was proposed to discriminate the respiration signal from artifacts and noise. The designed sensor was tested in domestic environment and was reported to be 91.3% accurate .

7.3.2 The non-contact based respiratory measurement systems

A prototype respiration monitoring based for detecting the carbon dioxide has been reported [93]. The system was tested fifty centimetres away from the subjects. Several carbon dioxide sensors were mounted around four sides of a baby's crib, away from the baby to avoid uneasiness. The system was not able to measure respiration rate, but it was possible to integrate the carbon dioxide sensors to an existing respiration monitoring system to reduce the false alarm.

Chest movements produced during the process of respiration can be measured in a non-contact manner using the Doppler radar. The ultrasound sensor emits a continuous waveform using an antenna toward the direction of the moving chest. The reflected waveform, from the chest is subsequently picked up by the antenna and the phase different is measured. The location of the chest, $x(t)$, at a particular time is determined using Equation (7.3). Based on this principle, different breathing measurement systems or Radar Vital Signs Monitor Technology were reported [94],[95],[96].

$$x(t)=\frac{\lambda\theta(t)}{4\pi} \tag{7.3}$$

Where

$\theta(t)$ is the measured phase shift of the reflecting continuous wave at time t .

λ is the wavelength of the continuous waveform

Machine vision systems, using a static camera to detect thorax motion has been be used to monitor respiration [97]. The projection of the surface of thorax was presented as a region with a range of brightness/intensity values in the images. Respiration rate monitoring was made possible by detecting the variation of the location of those image intensity values across different time values.

The relative motion between the camera and the moving thorax produced a pattern of apparent motion namely, the optical flow. To measure the optical flow, the image intensity value at each pixel location (x, y) captured at time t, the point on the surface of thorax was projected onto the camera image. The system was reported to correctly classify 99.4% of the movements over 61 hours of monitoring [97].

Infra-red cameras were also used to monitor respiration rate [98][99]. A thermal infra-red camera, measuring the respiratory rate passively, consists of a focal plane array of mid-wave infra-red sensor. It detects the change of infra-red energy around the nostril region of the subject. The minor change in detected infra-red energy due to the absorption of emitted carbon dioxide around the nostril region during the process of inhaling and exhaling air was picked up by a sensor array and digitised. In order to correctly determine the location of measuring region, image processing algorithms were applied.

The process of tracking required identification of salient features. Zhen Zhu *et. al.* (2005) [98] selected the two areas between the bridge of nose and the inner corner of the eyes (*i.e.* the periorbital regions which are the face warmest areas) as two salient features. Another salient feature was the apex of nose which has lowest temperature around the face region. This is due to the lower blood supply to this region and it is widely exposed to ambient environment. To correctly identify the measuring region, the flow of carbon dioxide, these three salient features were used to manually form a triangle and the perspective distortion of the triangle was estimated. Hence, a quadrilateral region, *e.g.* in rectangle shape with zero perspective distortion, right under the apex of nose was formed as the measuring region. The mean temperature of the measuring region was subsequently used to estimate the breathing signal.

Instead of using mid-wave infra-red energy, Chekmenev *et. al* (2005) [100] made use of focal plane array of long-wave infra-red sensor to measure the temperature change around the neck region, carotid vessel complex, and the nasal region. A wavelet-based method, using a kernel with second derivative of the Gaussian function, was used to decompose the image to difference scale. A three-scale decomposition was used to decompose the image of manually selected carotid vessel complex and nasal regions. The mean values of the regions, along with the time stamps, where subsequently used by the continuous wavelet analysis to reconstruct the breathing signal.

Instead of using the projective distorted information, *e.g.* the projected three dimensional movement of thoracic into the two dimensional camera image, directly measuring the three dimensional movement could help to provide a better measurement of respiratory rate in a non-contact based environment. Measuring the three dimensional information for respiration monitoring was carried out by Aoki *et. al* [101] using a fibre grating vision sensor. Their system consisted of two units, namely the fibre grating projecting an array of invisible infra-red light spots (wavelength 810nm) and a unit of CCD camera with and optical band-pass filter (light of wavelength 760-900nm). The CCD camera was used to capture the scene of light spots and an image processing algorithm was subsequently used to detect the barycentric of each of the light spots. Since the distance between the camera and the fibre grating was known, the principle of triangulation could be used to determine the relative 3-D location of each light spot. The thoracic movements, displacing the location of light spots appeared in each inter image frame, was detected using inter image subtraction techniques. The moving distances of bright spots in each image were extracted and used to classify as respiration (periodic displacement), or rolling over (large amount of displacement) of the subject.

Multiple infra-red sensors were used to monitor respiration rate by detecting chest movements [78]. The type of infra-red sensor used was a low cost distance sensor, SHARP GP2Y0A21YK. The sensor operated based on the principle of triangulation where the incidence of the reflected, modulated infra-red signal on one of the position sensitive device (PSD) was a function of the distance of the object. This relationship was used to estimate to determine the distance to the chest (and thus its movements). The layout of the sensor is shown in Figure 7.2.

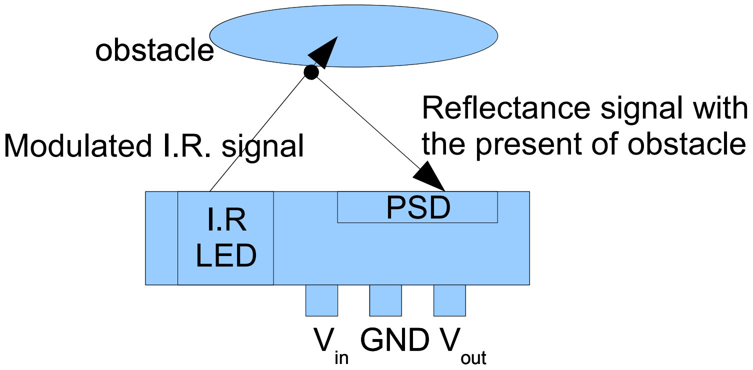


Figure 7.2: The operation of SHARP GP2Y0A21YK infra-red distance sensor.

The sensitivity of the sensor was about three millimetres. This was found to be insufficient to accurately detect finer chest movements.

Another approach involved the use of an ultrasonic distance sensor. It measured the phase difference of the transmitted and received ultrasound echo signals to determine the distance of the chest (thus chest movements) [79]. An accurate ultrasonic sensor, Mini-A series [102], provided by Senscomp was used. The distance measurement accuracy of the sensor was 0.5 mm. The sensor was connected to an analogue to digital converter, provided by a Microchip's PICDEM FS USB board. The measured distance, along with the time the measurement was taken and were transferred to the PC via USB port. A software module read the transferred data and displayed it in real time. The respiration rate was estimated by setting a threshold value, the average value between the maximum and minimum distance during the process of inhaling and exhaling. The implemented system was further improved by Mr. Stephen Tan Kang Song (a Ph.D. research student at Sheffield Hallam University). He added a web-cam to record the subject's movements and the respiration rate was automatically estimated from the measured distance by using the Fast Fourier Transform (FFT) of the recorded data.

7.4 Implemented computer vision based system

Vision systems provided provided another route to monitor respiration rate.

A vision based respiratory rate monitoring system works by sampling the location of the area of interest, *e.g.* the chest or abdomen, into images. These images are subsequently processed to measure the generated motion during the process of respiration. The operation to estimate the motion can be performed in the two dimensional projective space and the re-projective, three dimensional, space. The optical flow method discussed previously is an example of motion estimation in two dimensional projective space. The re-projective space method, makes use of the principle of computer vision geometry, by projecting back the information available in the sampled image to the three dimensional space to perform the required motion estimation.

A monocular vision system, using the images captured by a calibrated camera and a small planar marker attached to the position of the body generating the motion during breathing, was implemented and is discussed in the following section. Another respiratory rate measuring system, using the principle of triangulation and stereo vision system to measure the generated motion in three dimensional space, was also constructed and will be discussed in the following section.

7.4.1 Monocular vision system

The use of monocular vision system for measuring the respiration rate was inspired by the planar object pose estimation method discussed in the section 3.4.1. Using the estimated planar homography (Section 3.3.2.1) and the calibrated camera (Section 3.3.1), the pose (location and orientation) of the planar object with respect to the camera coordinate frame could be estimated. To accommodate the application of respiratory rate measurement, a small planar object (about 20mm by 20mm) was implanted with black and white checker board pattern (size of each square was 3.75mm by 3.75mm each) as shown in Figure 7.4. Grid coordinates were assigned to each X-corner, formed by the intersection of the corner of each black square, to create a planar homography. The set-up of the respiratory rate measurement system is shown in Figure 7.5.

The process of estimating the respiratory rate involved attaching a checker board marker to the position of the body where maximum displacement occurred during respiration. A software with graphical user interface (GUI) running in a PC was created to display both the camera image and the measured respiratory rate. A web camera was connected to the PC and was adjusted in such a way so that the marker appeared around the centre of the image. The process of determining the respiratory rate started by requiring the user to click once in the centre of the the black square in the middle of the the marker. Once this centre position was identified, a window region, centred at this point was used by the X-corner detector algorithm to determine the location of four corners of that square, with up to sub-pixel accuracy. When the four corners of the square were successfully found, the window was saved as a reference template and could be used for recovering the location of marker within the image, with any template based tracking method.

A coarse planar homography was formed using the equation (3.5) by employing the detected four corners. Using the estimated coarse planar homography the remaining X-corners were detected, using the similar technique described in Appendix A1. A planar homography was subsequently estimated using all the detected sixteen corners. The pose of the planar marker was estimated using equation (3.13). The 3-D position of the centre of the marker, with respect to the camera coordinate frame, was hence estimated. The estimated 3-D coordinate was used as the initial location of the measurement. The image location at the centre of gravity of the marker was estimated by using the projection of planar homography by using equation (3.4).

The image location previously provided by the user was replaced with the image location of the centre of gravity of the marker. The process of detecting four nearest corners around the centre of gravity of the marker was repeated in a newly captured camera image and a coarse homography was estimated. Due to the intensive processing required for the corner detection, the number of corners to form the refined planar homography was reduced from sixteen to eight (*i.e.* the four corners nearest to the centre of gravity of the marker and another four extreme diagonal corners). The pose of the planar homography was again estimated and the second location of the 3-D coordinate of the centre of marker was estimated.

The process of monitoring the breathing was initialised as follows. To monitor the breathing process one needs to locate the centre of gravity of the planar pattern in a pair of consecutive images. The process is initialised if the Euclidean distance between the locations was larger than a threshold value, 1.5mm was used in this experiment. The value 1.5mm was decided based on the accuracy of the calculation of the 3-D location of a point on the planar object which is around plus or minus 1mm.,

It was assumed that the position of the camera was always right in front of the marker. Hence, the generated motion was very much aligned to the viewing axis, the Z-axis of the camera. The motion generated along the other two axes (X-axis and Y-axis) was neglected. Based on the movements of the location along the Z-axis only, the respiration was estimated.

To further improve the detection of the chest movements caused by respiration, the average of several estimated Z-axis values (Z_a) was used. A cyclic buffer with a size of three was used to find the average value of movement which is used and updated through the monitoring process. Since the movement values from the initial first three frames needed to be known before the monitoring could be started, there will be a two frames delay at the starting of the monitoring process.

The state of respiration, inhaling or exhaling, was identified by determining the direction of the motion of planar marker. During the process of inhaling, the volume of thoracic increased and move toward the camera. Hence, the planar marker moved toward the camera and the measured average Z-axis value of the marker reduced. The difference (Z_d) between the latest Z-axis value ($Z_{a(i)}$) and the previous Z-axis value ($Z_{a(i-1)}$) was measured as shown in equation (7.4). If this Z_d was negative, the state of respiration was quantified as inhaling, otherwise, exhaling. The state of respiration was displayed on the GUI. The subsequent task was to make use of the highest peak and lowest peak of the measured 3-D coordinate point of the centre of the marker to estimate the respiratory cycle.

$$Z_d = Z_{a(i)} - Z_{a(i-1)} \tag{7.4}$$

Respiration transition state from inhaling to exhaling, or the other way round, was determined by detecting the change of sign of the value Z_d. Once a complete breathing cycle was detected, the time taken to complete the breathing cycle was determined and the rate of respiration per minute (bpm) was measured by using equation (7.5). The measured respiration per minute was displayed on a GUI. The complete flowchart of the process is shown as a flow chart in Figure 7.3.

$$bpm = \frac{60 * c}{t} \tag{7.5}$$

Where

c , the number of breath count,

t , the measured time, in second.

The developed system was tested using manikins that could be manually pumped to perform a simulated respiration (Figure 7.6). In order to obtain maximum accuracy of estimating the location of the marker, the webcam was positioned to only monitor the torso area (140mm width, 110mm height) of the manikin. This required the webcam to be placed very close (around 130mm to 140mm) to the torso of the manikin. To obtain the same accuracy without placing the camera very close to the manikin, the optical range of the camera could be increased by attaching a lens to it. This could be achieved by using a lens attached to a commercial camera by means of a C-mount (Figure 7.7).

The results obtained from measuring respiration waveform were plotted and shown in Figure 7.8. The snapshot of the designed GUI with qt-ruby [103] is shown in Figure 7.9. The machine vision algorithms and process of estimating the rate of respiration was coded with Hornetseye [5], a Ruby real-time computer vision extension. An average of 6.5 frame per second was achieved by using a Pentium 4 3GHz machine with a web cam with resolution 640x480 at 10 frames per second (fps). The actual setting of the system is shown in Figure 7.6.

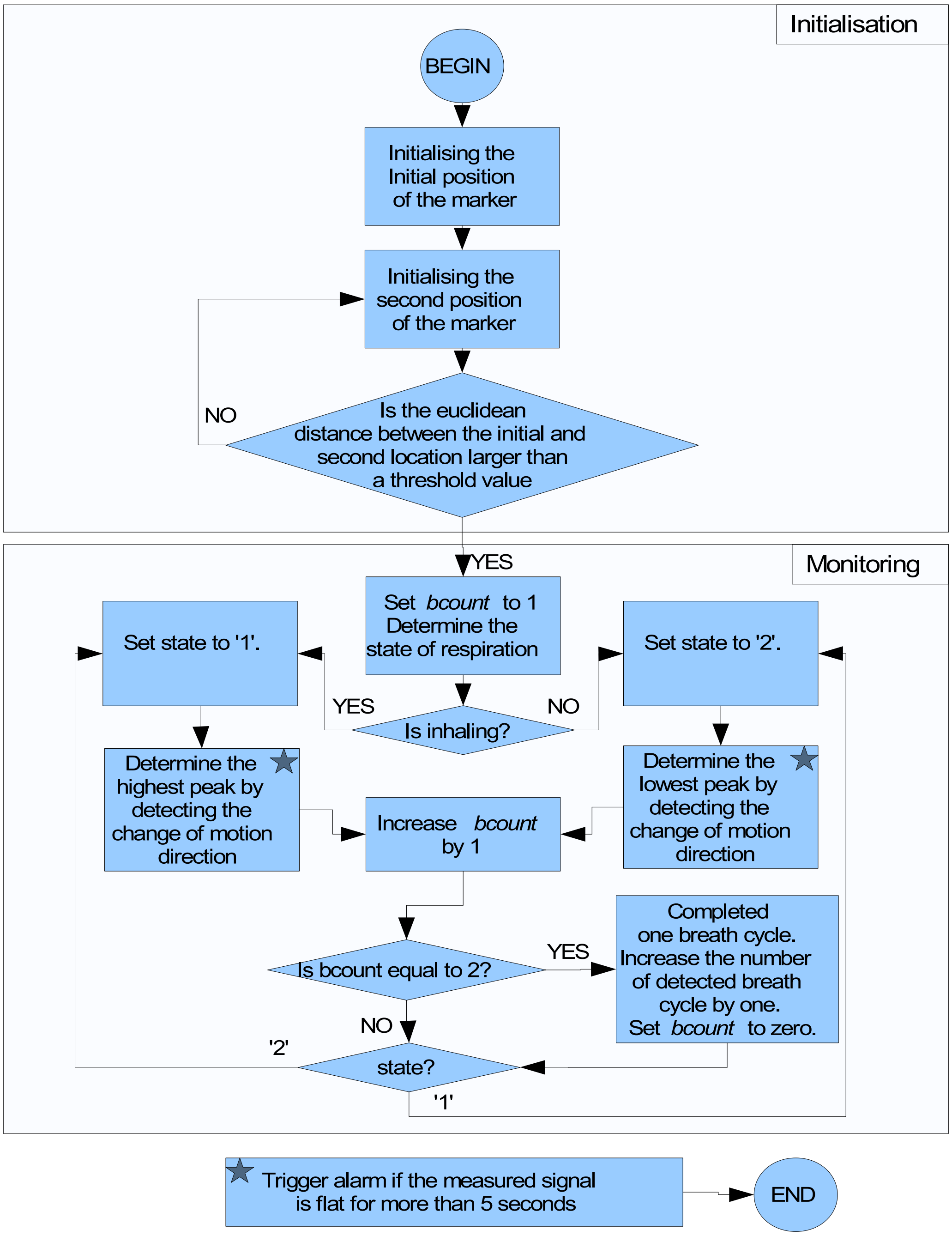


Figure 7.3:The flow chart of the monocular vision system for measuring breathing rate.

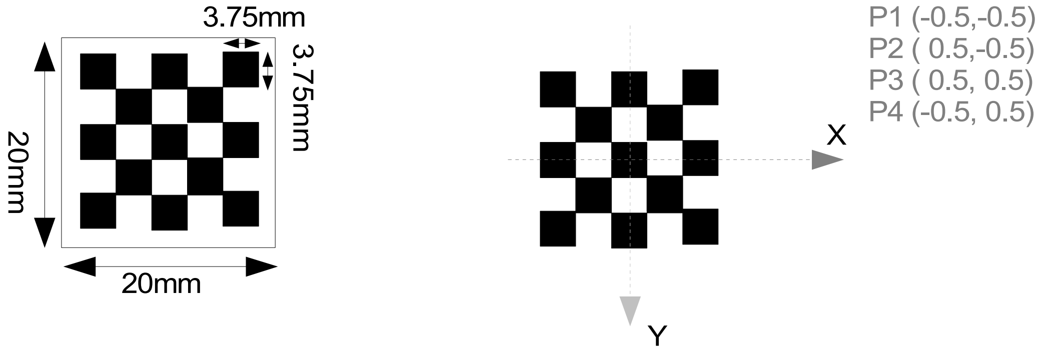


Figure 7.4: The dimension of the planar object marker (left). Grid coordinates were assigned to four X-corners (right).

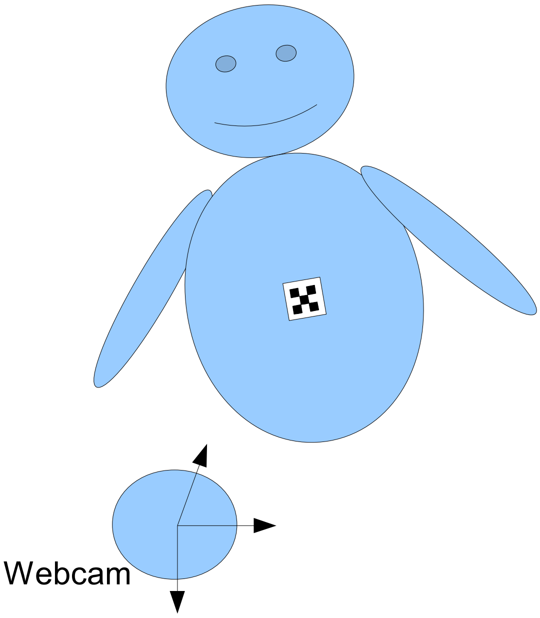


Figure 7.5: The setup of the monocular respiratory rate measuring system.

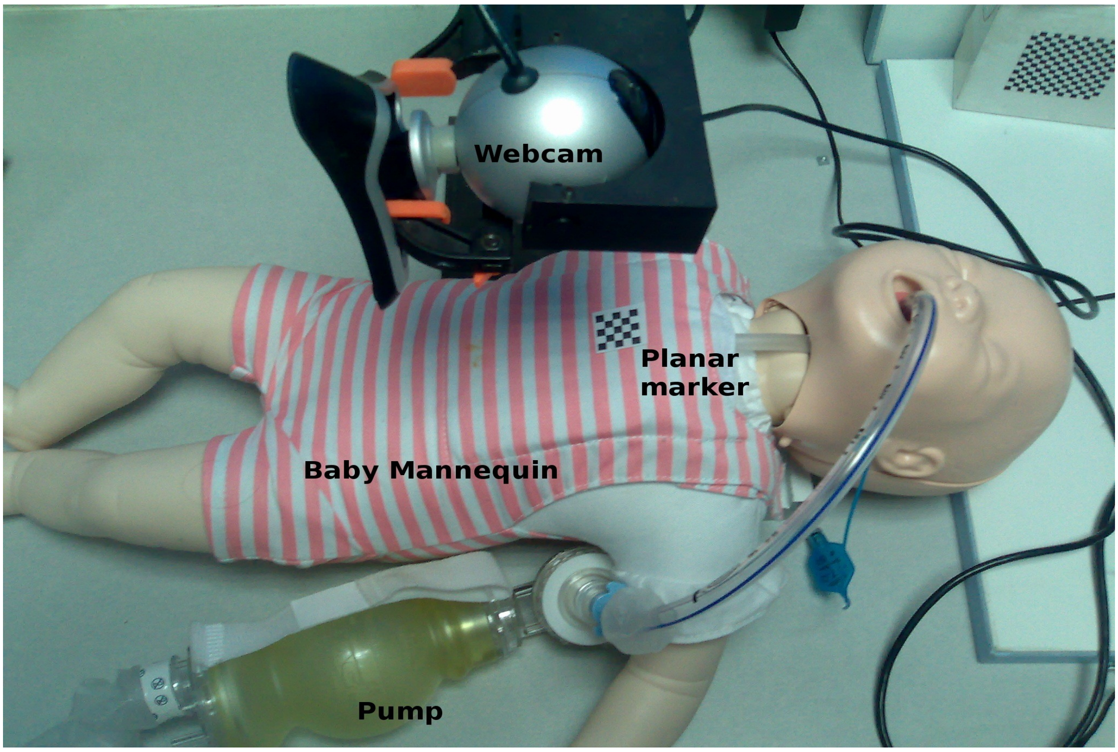


Figure 7.6: The actual testing scenario with the webcam, planar marker, baby mannequin, and pump, to simulate the breathing action.

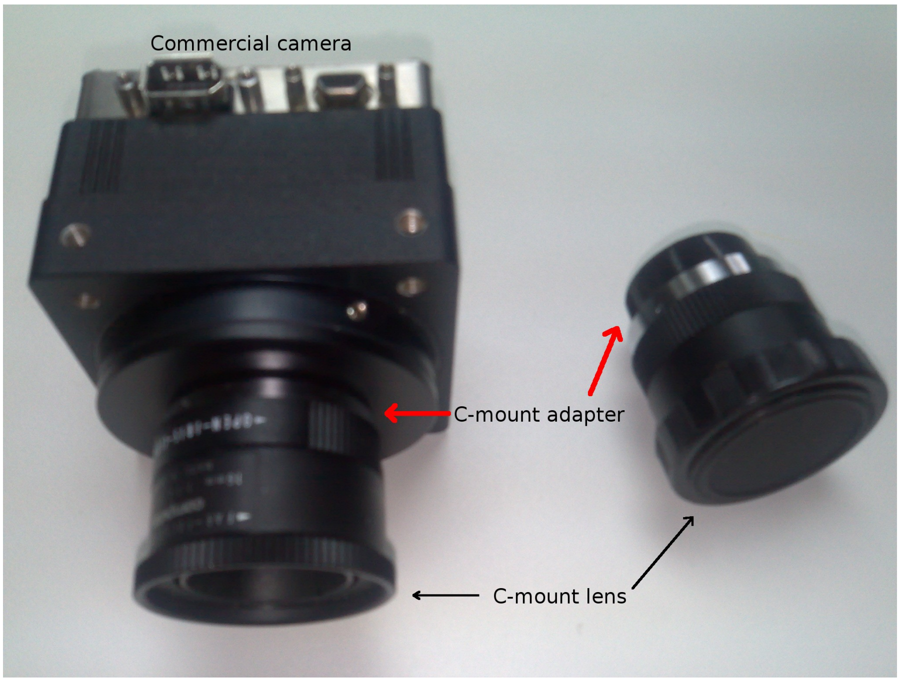


Figure 7.7: Using the commercially available c-mount camera with the monocular respiratory rate measuring system.

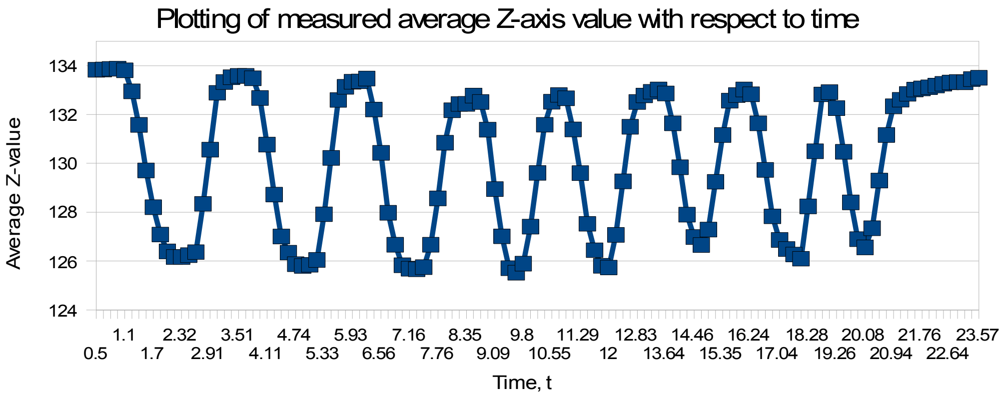
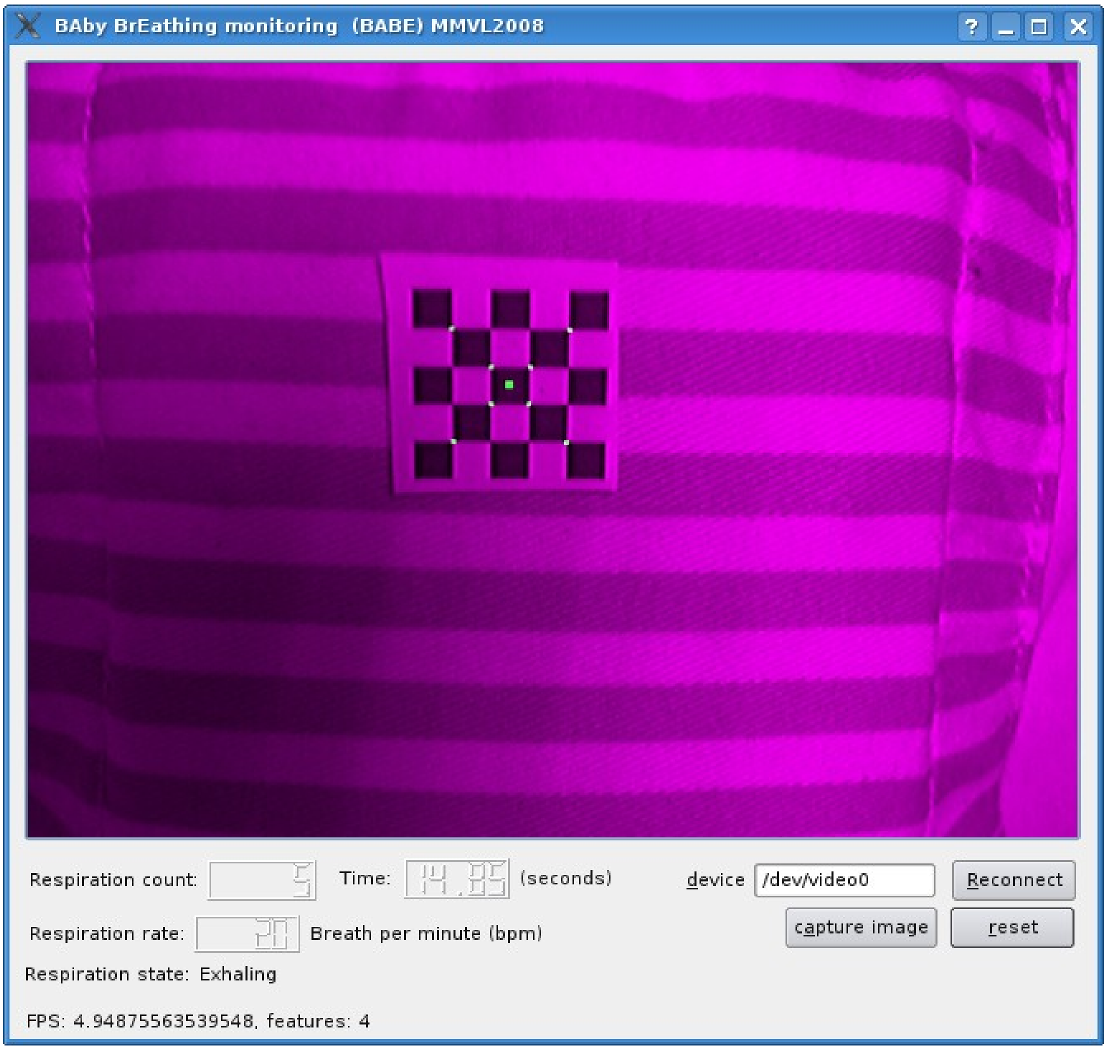


Figure 7.8: The plotting of waveform of respiration, produced by using the average measured Z-axis value, Z_a *, with respect to the system time.*



Respiration count: 5

Time: 14.85 (seconds)

device /dev/video0

Respiration rate: 20 Breath per minute (bpm)

Respiration state: Exhaling

FPS: 4.94875563539548, features: 4

Figure 7.9: The snapshot of the designed GUI, with Qt-Ruby and Hornetseye, for measuring the rate of respiration. The eight detected corners and the centre of gravity were highlighted.

7.4.2 Stereo vision system

A physical point location in the real world can be expressed with it's 3-D coordinates, with respect to a coordinate frame in the Cartesian coordinate system. The use of a single camera, attaching to a computer system, enables the use of the computer system to estimate multiple 3-D locations or points with up to a scale. Camera pin-hole model is a commonly used model used by the geometry computer vision community to perform the measurement. A three dimensional coordinate frame, with the origin at the centre of the projection of the camera viewing scene into the camera image, was assigned as a global reference coordinate frame. Any physical point in the viewing scene of the camera, can subsequently be expressed with a 3-D coordinates with respect to the camera coordinate frame, up to a scale. By introducing some constraint, *e.g.* four points lying on a planar object or the planar homography, allowed the computer system to recover the metric measurement of a plane, in the calibrated monocular vision system.

Adding another calibrated camera into the computer system allowed the metric measurement of the physical scene to be recovered without any constrain. Once the relative location and orientation between the calibrated cameras were estimated, the 3-D coordinates of any physical point in the viewing scene accommodated by the two calibrated cameras could be estimated by using the method of triangulation. Hence, the planar marker introduced in the previous section, was replaced by the stereo vision system. In the following section initially the steps of determining the rigid body transformation in between two cameras is discussed, the method of triangulation, and the process of using the calibrated system to measure the breathing rate is also outlined.

7.4.2.1 To estimate rigid body transformation between two cameras

A stereo rig, consisting of two cameras firmly mounted onto a rig, was created to perform the stereo vision system. The process of calibrating the stereo rig determined the relative location and orientation, the rigid body transformation, between the coordinate frame of the two cameras.

A rigid body transformation consist of two components, namely a three by three rotation matrix (R) and a three dimensional translation vector ($\vec{T}$) . These two components were used to transform any 3-D coordinate point in a coordinate frame (C) into another coordinate frame (C'). The rigid body transformation preserved the rigidity, *e.g.* the Euclidean distance in between the two coordinate points in a coordinate frame (C) remained the same in the transformed coordinate frame (C'). The four by four transformation matrix (${}^{C'}\Gamma_C$), representing the rigid body transformation of a homogeneous point $P=[x \quad y \quad z \quad 1]^T$ in C into $P'=[x' \quad y' \quad z' \quad 1]^T$ in C' is shown in equation (7.6).

$$p' = {}^{C'}\Gamma_C\, p \qquad (7.6)$$

Where

$${}^{C'}\Gamma_C = \begin{bmatrix} R & \vec{T} \\ \vec{0}^T & 1 \end{bmatrix}$$

$$\vec{0} = [0 \quad 0 \quad 0]^T$$

A pose of a planar object (${}^{C}Pose_O$), transforming the grid coordinates representing in a planar object coordinate frame (O) into a camera coordinate frame (C), was estimated by using equation (3.13). Using the two cameras (see Figure 7.10), namely camera left (CL) and camera right (CR) looking at the same planar object, two sets of poses, ${}^{CL}Pose_O$ and ${}^{CR}Pose_O$, were estimated, respectively. Using the two estimated poses and the principle of rigid body transformation, the relative orientation (R) and locations ($\vec{T}$) of the two cameras, from the right camera to the left camera, could be estimated with equation (7.8) and (7.13), respectively.

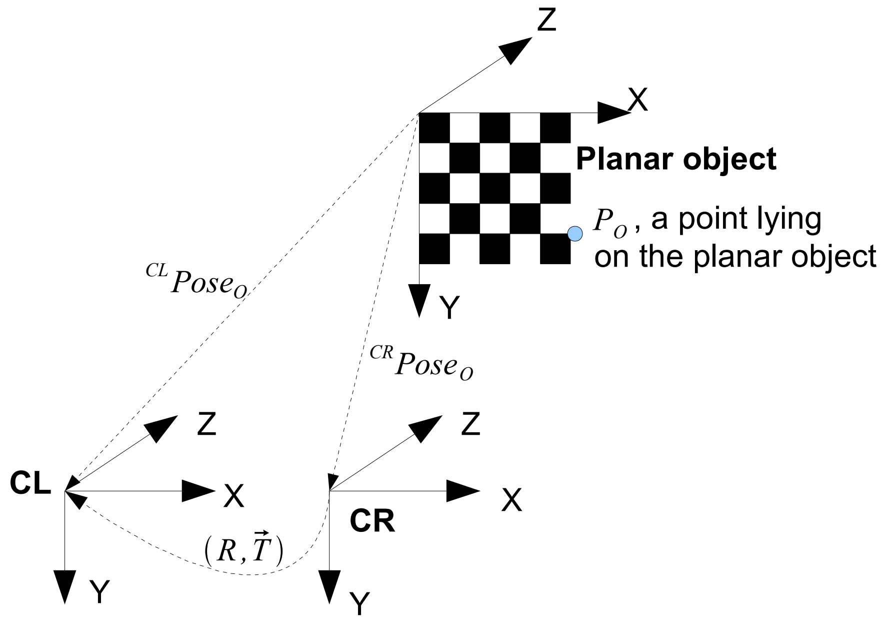


Figure 7.10: Three coordinate frames and their transformation.

Let $^{CL}Pose_O = \begin{bmatrix} R_L & \vec{T}_L \\ 0 & 1 \end{bmatrix}$, $^{CR}Pose_O = \begin{bmatrix} R_R & \vec{T}_R \\ 0 & 1 \end{bmatrix}$

Using the rigid body transformations method

$$R_R = R\, R_L \tag{7.7}$$

Hence,

$$R = R_L^{-1} R_R \tag{7.8}$$

A point expressed in the planar object coordinate frame, P_O , can be expressed in both the left and right camera coordinate frame, using equations (7.9) and (7.10), respectively.

$$P_L = R_L X_O + \vec{T}_L \tag{7.9}$$

$$P_R = R_R X_0 + \vec{T}_R \tag{7.10}$$

Using the same principle, a point expressed in left camera coordinate frame can subsequently be expressed in right camera coordinate frame as shown in equation (7.11).

$$P_R = R P_L + \vec{T} \tag{7.11}$$

Substitute equation (7.9) and (7.10) into (7.11), we get (7.12)

$$R_R X_0 + \vec{T}_R = R(R_L X_0 + \vec{T}_L) + \vec{T} \tag{7.12}$$

Solving equation (7.12), the relative translation between two camera coordinate frames, $\vec{T}$, is resolved as shown in equation (7.13).

$$\vec{T} = T_R - R\vec{T}_L \tag{7.13}$$

To achieve a better accuracy in determining the relative orientation and location between the two camera coordinate frames, multiple left and right images of the planar object at different locations were captured to perform the calibration. A stereo rig calibration software toolbox was created by Bouquet (2008) [104] and used to perform the calibration in this project.

7.4.2.2 Using the principle of triangulation in a stereo vision rig to recover the physical 3-D location of a point

A physical point in three dimensional space, $P = [X \quad Y \quad Z]^T$ (inhomogeneous), within the line of sight of both the cameras in a stereo vision rig, was projected as a correspondence, (x_l, x_r) , in camera images left and right, respectively. Using the camera pin hole model, the process of projection was assumed centred at the origin of the camera coordinate frame. Hence, the location of the correspondence in the camera image was actually the intersection of the line formed by the physical point of the origin of the camera coordinate frame with the CCD/focal plane array sensor of the camera. Two unit vectors, $\vec{v}_l$ and $\vec{v}_r$, representing the original line of sight of the physical point to the left and right camera, is shown in Figure 7.11. With two of the lines of sight and the transformation in between the two cameras, a virtual triangle was formed.

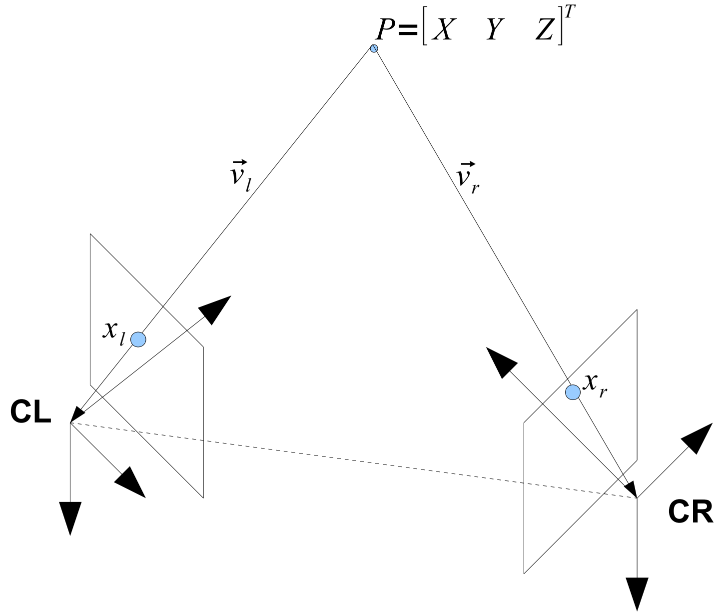


Figure 7.11: The virtual triangle formed by the stereo rig with a 3-D point.

As discussed previously in section 3.4.3.1, the vector representing the line of sight was determined, in their corresponding coordinate frame, by using an image coordinate (x_l, x_r) and the camera intrinsic parameters (I_l, I_r) (see Equation (7.14) and (7.15)).

$$\vec{v}_l = I_l x_l \tag{7.14}$$
$$\vec{v}_r = I_r x_r \tag{7.15}$$

The point of interest, in this case, the physical point producing the correspondence (x_l, x_r) in both the left and right camera image, was determined by finding the intersection of two lines of sight. The intersection of two 3-D straight lines. lines of sight, in 3-D space forms a 3-D point. To resolve the 3-D coordinate of the intersection point, the line of sight, $\vec{v}_r$, had to be transformed into the left camera coordinate frame, $\vec{v}_r{}'$, the reference coordinate frame of the stereo rig, as shown in equation (7.16).

$$\vec{v}_r{}' = R\,\vec{v}_r + \vec{T} \tag{7.16}$$

Using the two lines of sight, $\vec{v}_l$ and $\vec{v}_r{}'$, the point of interest, P, expressing in the left camera coordinate frame, was determined as follow:

Let the point, expressed in left camera coordinate frame, be

$$\lambda_l v_l \tag{7.17}$$

where λ_1 is the length of the line, measured from origin of the left camera coordinate frame to the point of interest.

Let the same point, expressed in right camera coordinate frame, be

$$\lambda_r v_r \tag{7.18}$$

where λ_r is the length of the line, measured from origin of the right camera coordinate frame to the point of interest. Using equation (7.16), the point expressed in right camera coordinate frame was transformed into left coordinate camera frame, as shown in equation (7.19).

$$\lambda_r R \vec{v}_r + \vec{T} \tag{7.19}$$

Hence,

$$\lambda_l \vec{v}_l = \lambda_r R \vec{v}_r + \vec{T} \tag{7.20}$$

Arranging Equation (7.20) in matrix form, as shown in Equation (7.21).

$$\underbrace{\begin{bmatrix} \vec{v}_l & -R\vec{v}_r \end{bmatrix}}_{:=A} \begin{bmatrix} \lambda_l \\ \lambda_r \end{bmatrix} = \vec{T} \tag{7.21}$$

The length of each line was determined by solving the non-homogeneous system in the least square sense as shown in Equation (7.22).

$$\begin{bmatrix} \lambda_l \\ \lambda_r \end{bmatrix} = (A^T A)^{-1} A^T \vec{T} \tag{7.22}$$

Once the values of λ_l and λ_r were obtained, the 3-D location of the physical point, the point of interest, was determined using either equation (7.17) or (7.19) in left camera coordinate frame, or equation (7.18) in right camera coordinate frame.

Various sources of errors such as uncertainties in camera intrinsic parameters and error in determining the exact location of the correspondence (x_l, x_r), might make the lines do not intersect at all. A general approach, taking the average of the measured result with equations (7.17) and (7.19) was used as shown in equation (7.23).

$$0.5(\lambda_l v_l + \lambda_r R \vec{v}_r + \vec{T}) \tag{7.23}$$

7.4.2.3 Using the stereo vision system to measure the breathing rate of a baby mannequin

Using the strategy described in monocular vision system (section 7.4.1), the respiration rate of a manikin was measured by using the depth value if the Z-axis of the reference camera coordinate frame was almost aligned to the direction of the movement induced during the process of respiration. If this was not the case, the measurement was substituted with the change of Euclidean distance of the physical point. As discussed in the previous section, using the principle of triangulation in a stereo vision rig to recover the physical 3-D location of a point required a correspondence, (x_l, x_r). This was performed by providing the user with a GUI to select the point of interest to be measured x_l and x_r, in both left and right camera image, respectively. Once the inputs (x_l, x_r) were provided, a window of interest, centred at each of the provided input point, and a tracking templates, one in each image, was initialised. Two important issues, namely, the selection of tracking feature and the accuracy of the triangulation with the detected correspondence feature are discussed following the following section.

The aim was to use the feature provided by the user during the process of initialisation was to perform the tracking using the image tracking algorithms, *e.g.* template matching based (normalised cross-correlation) or image intensity value based (colour segmentation, mean-shift) method. Unfortunately, non of these two dimensional image tracking algorithm worked well on a texture less object/surface, *e.g.* a plain wall with same colour or texture less part of the thoracic. Hence, the user needed to provide the machine vision algorithm with unique high contrast feature.

The process of triangulation was carried out by using the result of tracking algorithm.

For the template matching based method, the centre of the template window, with highest matching value, was used for triangulation. The advantage of template matching based method is its high accuracy of detection, but it is processing intensive. When the image intensity value based method was applied, the centre of gravity of the segmented colour region was used for triangulation. The advantages of this tracking method is that it is less processing intensive, but at the same time it is less accurate compared to template matching and the selected segmentation colour must be unique within the image region.

To handle the two issues described above, a colour segmentation technique was used to track a small unique colour marker. A unique colour was chosen for the marker and the shape of the marker was made to provide a sharp feature, *e.g.* the shape of a water drop. The shape of water drop allowed the feature detector (*e.g.* Shi-Tomasi [76])to be applied around the colour segmented region, centred at the centre of gravity of the result of the colour segmentation. Hence, the triangulation could be applied accurately on the distinct feature. The designed marker was shown in Figure 7.12.

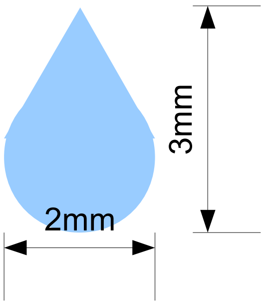


Figure 7.12: The designed marker.

The discussed system was used by a student Mr. Chong Chi Wei in his Msc project “Development of stereo vision based respiratory rate measurement system” to measure the simulated respiration of a baby manikin [80]. The simulated respiration state and rate of respiration were correctly determined by the designed system, with two off the shelf cameras, Hornetseye and Qt-Ruby. A sample measured respiration wavefrom, estimated by measuring the change in the Euclidean distance, of a single tracked feature, with respect to time, was recorded and plotted (Figure 7.13). The baby mannequin was manually pumped at a constant rate. The pumping count was manually recorded and was correlated well with the estimated rate of the respiration. A visual correlation between the estimation and the physical breathing state of the mannequin was also correlated. More result and discussion regarding this work can be found in the master's student thesis [80].

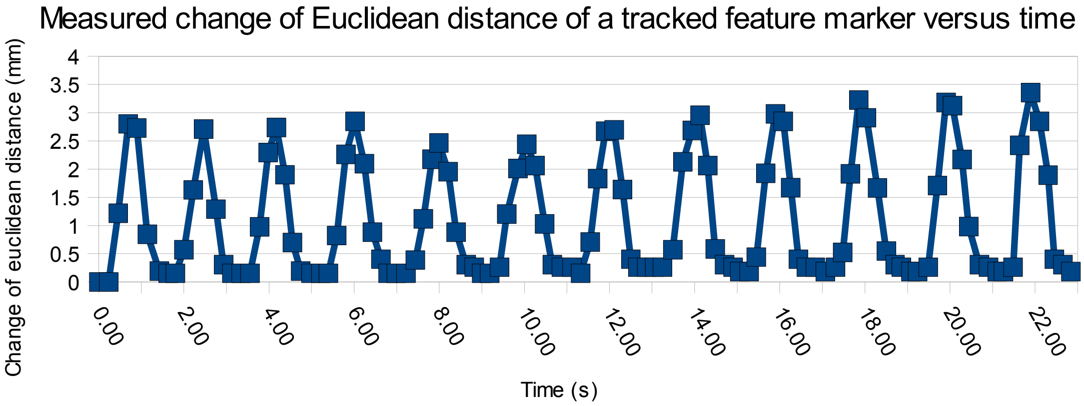


Figure 7.13: Measured change of Euclidean distance of a tracked feature marker versus time using the calibrated stereo vision rig [80].

7.5 Conclusion

A real time mono vision and a stereo vision based non contact breathing monitoring systems, for pilot studies, were designed and implemented. The systems were tested on a baby mannequin and the state of the estimated respiration, *e.g.* inhaling or exhaling, were successfully estimated and visually correlated well with the simulated pumping action. The designed and implemented mono vision based breathing monitoring system demonstrated the possibility of using a camera, with the help of a planar marker, to measure the breathing rate. This could subsequently be ported to any embeded device, *e.g.* mobile phone with a camera, or a laptop/netbook. The designed stereo vision rig, implemented as part of a masters thesis [80], under the supervision of the author, demonstrated the possibility of marker-less breathing monitoring system. However, a very small custom made marker (2mm x 3mm), with a unique colour, was used by the designed stereo vision based breathing monitoring system to provide better accuracy. Colour segmentation method was used to locate the location of the same feature in both the camera images (left and right). The pointed feature of the marker, in both the camera images, were picked up by Shi-Tomasi feature detector [76] and subsequently used by the process of triangulation to accurately locate the 3-D location of the feature. The rate of respiration was successfully measured by using the estimated 3-D location of the feature, in a series of camera image, across the time.

Chapter 8: Conclusion and Future Works

8.1 Conclusion

This research work has resulted in a number of major contributions, namely novel vision based 3-D surface data acquisition system, an automatic calibration for turntable based 3-D surface reconstruction system, and a vision based respiration monitoring system. The contributions are summarised in the following sections.

8.1.1 Novel vision based 3-D surface data acquisition system

For this thesis a novel vision based 3-D surface data acquisition system was designed and implemented. Off-the-shelf equipment, mainly a webcam, a laser line emitter, and two planar objects implanted with 'X'-type corners, along with a personal computer, were used to develop this system. Using of off-the-shelf equipment, avoided the need for expensive hardware and enabled the system to be designed for use in any desktop PC environment. The novelty of the designed system is that it does not require the use of positional sensors to provide the pose of the laser plane and allows the 3-D surface data to be acquired using freehand. The results achieved in this work are summarised in the following sections.

8.1.1.1 The imaging framework

An imaging framework was developed to achieve 3-D reconstruction. The essential elements of this framework were; camera calibration, corner detection, planar homography estimation, planar object pose estimation, laser points detection, laser plane pose estimation, triangulation, and real time update display window. A number of camera calibration techniques were explored and the most effective technique, *i.e.*

camera calibration using planar objects, was chosen. Another two essential elements, namely the planar homography estimation and planar object pose estimation, were provided by the chosen camera calibration technique.

Two different 2-D laser points detection techniques, colour segmentation and background subtraction, were proposed to detect the camera image pixel illuminated by the laser. It was found that the background subtraction technique performed better than the colour segmentation technique. The reason for this effect being that the colour values of the pixels change under different imaging environmental conditions. The accuracy of the 2-D laser points detection was improved, by incorporating a known Blias and Rioux sub-pixel operator.

The emitted laser line was made to intersect with the calibrated planar objects during the process of 3-D surface data acquisition. Using the parameters provided by the calibrated planar objects, the detected 2-D laser points lying on both the calibrated planar objects, were transformed into 3-D laser points with respect to the camera 3-D coordinate frame. Using the non-collinear 3-D coordinate points, the pose of the laser plane at the time captured by the camera was subsequently estimated. The estimation methods are explained in the next section.

8.1.1.2 The novel laser plane estimating methods

Using the geometry constraints obtained through the process of intersecting the laser plane with two planar objects, a number of novel laser plane pose estimation methods were designed and implemented. The first constraint is the pair of straight lines, formed by the intersection of the laser plane with the two non-collinear planar objects, which is captured in the camera image. This constraint was projected onto the camera 3-D coordinates frame. The second geometry constraint is that the two projected lines belong to the laser plane. The pose of the laser plane was subsequently estimated using the above two geometry constraints. This process is outlined in the following paragraph.

An effective way to accurately determine the pose of the laser plane is to make use of as much input data as possible, discarding the outliers, and subsequently forming a laser plane model with the inlier data. The rejection of the outliers was achieved using RANdom SAmple Consensus (RANSAC) based method either in the 2-D or 3-D space. RANSAC randomly selected the minimum number of input sample data and formed a model. It subsequently tested the model to determine the percentage of samples conforming to the model.

In the 2-D space, the rejection of the outliers was achieved by using RANSAC line model fitting process. Subsequently two different methods were implemented and compared to estimate the pose of the plane, using the projected inlier points. The first method, given the name R2D1 (see Section 5.2.4.1.2.1), was designed to make use of the two pairs of points, each pair from each of the two best fitted line models, (totalling four points in the camera 3-D coordinate frame). The second method, given the name R2D2 (see Section 5.2.4.1.2.2), was designed to make use of the all the inlier points. The method of least square plane fit was used to approximate the best fitting plane to the set of the projected points identified in R2D1 or R2D2.

In the 3-D space, the rejection of the outliers was achieved using RANSAC plane model fitting process. This was the third developed method, named R3D (see Section 5.2.4.2.1). Based on this method, all the 2-D laser points were projected to the camera 3-D coordinates and the inlier points were identified using the RANSAC plane model fitting process. The coordinates of these inlier points were subsequently used by the least square plane fitting method to determine pose of the laser plane.

The above developed methods were critically evaluated empirically. The experimental results indicated that multiple 3-D laser points were needed to accurately estimate the pose of the laser plane. Based on these, it was established that the methods R2D2 and R3D provided a better estimate of the pose of the laser plane.

Both the R2D2 and R3D methods, made use of all the inlier points to estimate the pose of the laser plane and provided good estimation. However, R3D which only made use of the 3-D plane geometry constraint, without considering the 2-D line geometry constraint, produced a larger standard deviation on the errors as compared with R2D2 in determining the orientation of the computed laser plane. The effect of the larger standard deviation (0.5 degree for R3D and 0.35 degree for R2D2) was not obvious within the 50mm length scanning workspace.

The experimental results (Section 5.5) showed that the R2D2 method was able to estimate the laser plane pose parameters in 3-D (described by a point in 3-D lying on the plane and the normal of the plane) with the following minimal uncertainty. For the normal of the plane, the variation in the value was found to be from 0.15(min) to 0.41(max) degrees and, for the point in 3-D, the variation between the values was found to be between 0.07mm(min) to 0.14mm(max) (Figure 5.16 and Figure 5.17).

The accuracy of the acquired 3-D points, using the R2D2 method, was assessed by scanning a staircase-like calibration object. The standard deviation of the estimated dimensions of the calibration object was found to be 0.38mm (linear dimension) and 0.28 degrees (angular dimension, i.e. the estimated angle formed by each stair).

Using the estimated laser plane pose parameters and the calibrated camera parameters, the 2-D laser points lying on the scanned object were triangulated. The triangulation result produced a cloud of points in 3-D. These were subsequently rendered in a real time operation to update the display window. The real time feedback allowed the user to keep track of the scanning progress. This assisted the user in filling up the sparsely populated areas with cloud of points in 3-D.

8.1.1.3 Critical evaluation of Salient feature detectors

The result of detecting corners lying on the planar object directly influences the accuracy of the laser plane pose estimation. A set of correspondences was formed by the implanted L-type and X-type corners on the planar object and their projected image coordinates. These were needed to calibrate the planar object. There are a variety of corner detection algorithms that have been developed during the last twenty years and these can be characterised into three types; image intensity variation based, template-based, and model based. Candidates from each category were selected and their accuracy in detecting the location of L- type and X-type corners in the synthetic images were critically evaluated. The result of the experiments showed that the Saddle point based X-type corner detector, one of the candidate from the model based corner detector, performed the best. This was considered to be due to the use of a suitable X-type corner model, where the corner is located at the saddle point of the model. Also, the use of a very large window (15x15) and recursive interpolation increased the accuracy of the corner localisation. The basis of an accurate corner detection is to identify a good corner model. Several corner models were examined and the L-type corners were found to be difficult to model (as they required non- linear operation) as opposed to X-type corners. Hence in this study, X-type instead of L-type corners were used as the salient features and the Saddle point corner locator was used for locating the corners.

8.1.1.4 A basic graphical user interface for the developed freehand 3-D surface data acquisition system

A Graphical User Interface (GUI) for the vision based 3-D surface data acquisition system, using the critically evaluated essential elements, was subsequently designed and implemented. This software was executed on an Intel Core2 Duo 2.10 GHz CPU (single process, single threaded) that could process approximately 3 frames per second. Using this system, a deformable breast phantom was scanned and a visual inspection of the surface confirmed that an accurate scanning of the phantom's surface had been obtained. The details of the phantom were visible in the reconstructed surface.

8.1.1.5 The system performance

Using the most accurate plane pose estimation method, *i.e* R2D2, the designed system was able to estimate the 3-D surface points at a rate of 3 frames per second (frame size 640 by 480 pixels). Hence the system was able to provide an accurate laser plane pose estimation as well as display the scanned 3-D points sufficiently rapidly.

8.1.2 Automatic calibration for turntable based 3-D surface reconstruction system

An immediate application for the novel laser plane pose estimation method was to calibrate the laser plane in a turntable based 3-D surface reconstruction environment. Traditional turntable based methods required extensive manual intervention. The manual operation of aligning the camera and the laser light source is laborious, time consuming and error prone. Errors introduced during the alignment of the components resulted in an inaccurate estimation of the 3-D location of the points lying on the surface of the object. Different ways of calibrating the pose of the turntable were proposed and simulated. Based on the results of simulation, a set of auto-calibration procedures, to estimate the pose of the components of the system, were designed and implemented. The auto-calibration procedures consisted of two major components, namely the laser plane pose estimation method and automatic turntable pose calibration. In the developed turntable pose calibration process, the only required task was to mount an irregular planar pattern to the surface of the turntable, such that its origin correlated with the centre of the turntable. Using the planar object pose estimation method, the poses of the turntable, rotated with multiple angular steps around the centre of the turntable, were estimated several times. The estimated poses that had a large difference with the mean of the other estimations were discarded. The pose of the turntable was subsequently estimated with the remaining poses using a global optimisation method.

Using a calibration object, the rotational axis of the turntable was estimated (Section 6.6). The rotational axis estimated using the calibration object, and the rotational axis derived from the pose estimation $\vec{T}_{rotA}$, were found almost aligned. The angular difference between the rotational axis estimated using the calibration object and the rotational axis derived from the pose estimation $\vec{T}_{rotA}$, was found to be 0.1 degrees.

Using the result of the turntable calibration, the 3-D locations on the surface of a regular object (cylinder) were estimated. The plotted 3-D locations were found to be visually accurate (multiple concentric circles were formed).

8.1.3 Vision based respiration monitoring system

Inspired by the accuracy and the efficiency of the implemented planar homography based 3D surface acquisition system, a monocular vision based baby respiration monitoring system was proposed and implemented. The monocular vision based respiration monitoring system consisted of a webcam and a small planar marker. The planar marker was attached to the chest of a baby mannequin. The respiratory chest movement was simulated by means of a chest pump. Using the computed 3-D coordinates provided by the centre of gravity of the planar marker, the torso motion was obtained and subsequently used to generate a respiration waveform.

The regular pattern based marker method that was used in this study had the problem of not being able to be reinitialised once tracking of the pattern was lost. This was due to the regularity of the pattern which prevented the designed method from resolving whether a location had moved or not. Making use of an irregular pattern was not practical due to the complexity of locating the pattern using brute force methods.

Inspired by stereo-vision technique, the marker was removed and a second camera was added to the system. The proposed stereo vision based baby respiration monitoring system had been implemented independently in an Msc degree project supervised any the author [80]. A simulated feature was attached to the torso of the baby mannequin and the 3-D locations were tracked during the simulated respiration. The movements of the baby mannequin's chest were estimated and the respiration rate was computed.

8.2 Future research

The performance of the implemented 3-D surface data acquisition system can be further improved by utilising multi-core processors available in most personal computer nowadays. A possible optimisation would be to concurrently run the RANSAC algorithm of each the planar objects on separate CPUs.

The current system requires manual intervention to segment the object being scanned and could benefit from a background subtraction technique to automate this process. Finding the difference between two image scenes, a background image and an image with the object to be scanned inserted to the scene, could provide a region of interest, with both the object and the shadow of the object. The shadow region could be detected and subsequently removed from the region of interest by using the property of how shadow influences the image pixel values, represented in HSV colour space [105][106].

A line scan operation was carried out to extract a single image pixel illuminated by the laser, from each row of the image. Instead of only having a single image pixel per row to represent the points lying on the object being scanned, it is possible to use an interpolation method, such as polynomial curve fitting [107], to provide more densely distributed data points. The required order of the polynomial curve fitting function could be decided based on the complexity of the shape of the object appearing in the camera image.

At the moment, the process of turntable calibration is performed using a ruby script. A GUI can be developed to enable the users to visualise the processes. The camera used in the turntable based 3- D surface reconstruction system was calibrated with the Matlab camera calibration toolbox [46]. Investigation could be carried out, using the planar object based calibration technique [45] with the collected images, from the turntable pose and laser plane pose calibration process, to automate the entire process.

Due to the performance issue of ruby scripting language, the developed mono vision based respiration monitoring system was limited to only 6.5 frames per second. The developed prototyping system could be ported to C/C++ programming language to increase its performance. Finally, an efficient planar marker tracking algorithm can be included to handle the localisation of the marker in the camera image.

References

[1] GHOST 3D LLC, “Microscribe solutions - integrated 3d digitizer systems,” Retrieved August 28, 2009, from http://microscribe.ghost3d.com/gt_microscribe.htm.

[2] National Institute for Health and Clinical Excellence, “Feverish illness in children,” Retrieved August 17, 2009, from http://www.nice.org.uk/nicemedia/pdf/CG47Guidance.pdf.

[3] P. Woodrow and J. Roe, *Intensive care nursing: a framework for practice.* Routledge, 2000.

[4] B. Amavasai, “Mimas computer vision toolkit,” Retrieved August 17, 2009, from http://vision.eng.shu.ac.uk/mediawiki/index.php/Mimas.

[5] J. Wedekind, B. Amavasai, K. Dutton, and M. Boissenin, “A machine vision extension for the Ruby programming language,” in *Information and Automation, 2008. ICIA 2008. International Conference on*, 2008, pp. 991–996.

[6] J. Faucher, “Camera calibration and 3-D reconstruction,” Master’s thesis, Sheffield Hallam University, UK, 2006.

[7] M. Fischler and R. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography.” ACM New York, NY, USA, 1981.

[8] P. Hébert, “A self-referenced hand-held range sensor,” in *Proc. 3DIM*, 2001, pp. 5–12.

[9] V. Popescu, E. Sacks, and G. Bahmutov, “The modelcamera: a hand-held device for interactive modeling,” in *3-D Digital Imaging and Modeling, 2003. 3DIM 2003. Proceedings. Fourth International Conference on*, 2003, pp. 285–292.

[10] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy, “Real-time 3d model acquisition,” vol. 21, no. 3. ACM New York, NY, USA, 2002, pp. 438–446.

[11] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch, “Visual modeling with a hand-held camera,” vol. 59, no. 3. Springer, 2004, pp. 207–232.

[12] U. Government, "CPI: U.S. City average; Personal computers and peripheral equipment," Retrieved August 17, 2009, from http://www.economagic.com/em-cgi/data.exe/blscu/CUUR0000SEEE01.

[13] P. Kühmstedt, C. Bräuer-Burchardt, C. Munkelt, M. Heinze, M. Palme, I. Schmidt, J. Hintersehr, and G. Notni, "Intraoral 3D scanner," in *Proceedings of SPIE*, vol. 6762, 2007, p. 67620E.

[14] J. Tan, "Medical Ultrasound Training Simulation Using Haptic Force Feedback Mechanism," Master's thesis, Sheffield Hallam University, UK, 2008.

[15] M. LLC, "Mova : About Us : Overview," Retrieved August 17, 2009, from http://www.mova.com/aboutus.php.

[16] L. Akarun, B. Gokberk, and A. Salah, "3D face recognition for biometric applications," in *Proc. of the 13th European Signal Processing Conference (EUSIPCO)*, 2005.

[17] F. Isgro, F. Odone, and A. Verri, "An open system for 3D data acquisition from multiple sensor," in *Computer Architecture for Machine Perception, 2005. CAMP 2005. Proceedings. Seventh International Workshop on*, 2005, pp. 52–57.

[18] G. Power and K. Xue, "A non-linear transform technique for a camera and laser 3-D scanner," in *Aerospace and Electronics Conference, 1995. NAECON 1995., Proceedings of the IEEE 1995 National*, vol. 2, 1995.

[19] GHOST 3D LLC, "Rsi 3d laser for microscribe g2 and mx digitizers," Retrieved August 17, 2009, from http://microscribe.ghost3d.com/G3D_Systems_MS-Laser_e.pdf.

[20] M. Glencross, G. Ward, F. Melendez, C. Jay, J. Liu, and R. Hubbold, "A perceptually validated model for surface depth hallucination." ACM New York, NY, USA, 2008.

[21] J. Bouguet, "Visual methods for three-dimensional modeling," Ph.D. dissertation, Citeseer, 1999.

[22] P. Torr and A. Zisserman, "Feature based methods for structure and motion estimation." Springer, 1999, pp. 278–294.

[23] L. Linsen, "Point cloud representation." Citeseer, 2001.

[24] Y. Ma, *An Invitation to 3-D Vision: From Images to Geometric Models.* Springer Verlag, 2004.

[25] Y. Vershinin, "3D Digital Surface Reconstruction Scanner for Medical Application," in *Proceedings of the Fourth IASTED International Conference Signal and Image Processing*, 2002, p. 496.

[26] H. Kawasaki and R. Furukawa, "Entire model acquisition system using handheld 3D digitizer," in *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, 2004, pp. 478–485.

[27] Yale School of Architecture, "Turntable calibration," Retrieved August 17, 2009, from http://www.architecture.yale.edu/dmonline/Tutorials/LargeLaserScanner/turntable.html.

[28] Endors.com, "3D Laser Scanner," Retrieved August 17, 2009, from http://www.sestosenso3d.com/home/en/3d-laser-scanner.html.

[29] G. Welch, G. Bishop, L. Vicci, S. Brumback, and K. Keller, "The HiBall tracker: High-performance wide-area tracking for virtual and augmented environments," in *Proceedings of the ACM symposium on Virtual reality software and technology.* ACM New York, NY, USA, 1999.

[30] P. S. Technologies, "Personal Space Tracker," Retrieved August 17, 2009, from http://www.personalspacetechnologies.com/product/pst-nf/.

[31] T. Pajdla, "Laser plane range finder: The implementation at the CVL." Citeseer, 1995, p. 98.

[32] S. Winkelbach, S. Molkenstruck, and F. Wahl, "Low-cost laser range scanner and fast surface registration approach," vol. 4174. Springer, 2006, p. 718.

[33] J. Bouguet and P. Perona, "3D photography on your desk," Retrieved August 17, 2009, from http://www.vision.caltech.edu/bouguetj/ICCV98/index.html.

[34] P. Cignoni, M. Corsini, and G. Ranzuglia, "Meshlab: an open-source 3d mesh processing system," 2008, pp. 45–46.

[35] L. Zagorchev and A. Goshtasby, "A paintbrush laser range scanner," vol. 101, no. 2. Elsevier, 2006, pp. 65–86.

[36] Creaform, "Handyscan 3D Handheld Laser Scanners," Retrieved August 17, 2009, from http://www.creaform3d.com/en/handyscan3d/default.aspx.

[37] N. Amenta, S. Choi, and R. Kolluri, "The power crust," in *Proceedings of the sixth ACM symposium on Solid modeling and applications.* ACM New York, NY, USA, 2001, pp. 249–266.

[38] G. Medioni and S. Kang, *Emerging topics in computer vision.* Prentice Hall PTR Upper Saddle River, NJ, USA, 2004.

[39] J. Semple and G. Kneebone, *Algebraic projective geometry.* Oxford University Press, USA, 1998.

[40] L. Wang, F. Wu, and Z. Hu, "Multi-camera calibration with one-dimensional object under general motions," in *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, 2007, pp. 1–7.

[41] M. Pollefeys, R. Koch, and L. Gool, "Self-calibration and metric reconstruction inspite of varying and unknown intrinsic camera parameters," vol. 32, no. 1. Springer, 1999, pp. 7–25.

[42] B. Triggs, "The absolute quadric," in *Proc. 1997 Conference on Computer Vision and Pattern Recognition*, 1997, pp. 609–614.

[43] P. Mendonça and R. Cipolla, "A simple technique for self-calibration," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, 1999, pp. 500–505.

[44] R. Hartley, "Estimation of relative camera positions for uncalibrated cameras," in *Proc. European Conference on Computer Vision*, vol. 92. NLCS, 1992, pp. 579–587.

[45] Z. Zhang, "A flexible new technique for camera calibration," vol. 22, no. 11, 2000, pp. 1330–1334.

[46] J. Bouguet, "Camera Calibration Toolbox for Matlab," Retrieved August 17, 2009, from http://www.vision.caltech.edu/bouguetj/calib_doc/.

[47] R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures." ACM New York, NY, USA, 1972.

[48] E. Trucco, R. Fisher, A. Fitzgibbon, and D. Naidu, “Calibration, data consistency and model acquisition with laser stripers,” vol. 11, no. 4. Taylor & Francis, 1998, pp. 293–310.

[49] F. Blais and M. Rioux, “Real-time numerical peak detector,” vol. 11, no. 2. Elsevier North-Holland, Inc. Amsterdam, The Netherlands, The Netherlands, 1986, pp. 145–155.

[50] T. Qt, “GUI library,” Retrieved August 17, 2009, from http://qt.nokia.com/products.

[51] Q. Zhu, B. Wu, and N. Wan, “A sub-pixel location method for interest points by means of the Harris interest strength,” vol. 22, no. 120. Blackwell Publishing Ltd, 2007, pp. 321–335.

[52] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Alvey vision conference*, vol. 15. Manchester, UK, 1988, p. 50.

[53] H. Moravec, “Obstacle avoidance and navigation in the real world by a seeing robot rover.” Stanford University Stanford, CA, USA, 1980.

[54] D. Parks and J. Gravel, “Harris/Plessey Operator,” Retrieved August 17, 2009, from http://www.cim.mcgill.ca/~dparks/CornerDetector/harris.htm.

[55] T. Pachidis, J. Lygouras, and V. Petridis, “A novel corner detection algorithm for camera calibration and calibration facilities,” in *Proc. 2nd WSEAS International Conference on Signal Processing and Computational Geometry and Vision*, pp. 6911–6916.

[56] T. Blaszka and R. Deriche, “Recovering and characterizing image features using an efficient model based approach.” Institut national de recherche en informatique et en automatique, 1994.

[57] G. Olague and B. Hernández, “Autonomous Model Based Corner Detection using Evolutionary Algorithms,” in *American Society for Photogrammetry and Remote Sensing, Annual Conference Proceedings, St. Louis, MO*, vol. 12, pp. 23–27.

[58] G. Olague and B. Hernández, “A new accurate and flexible model based multi-corner detector for measurement and recognition,” vol. 26, no. 1. Elsevier, 2005, pp. 27–41.

[59] J. Wedekind, B. Amavasai, and K. Dutton, “Steerable filters generated with the hypercomplex dual-tree wavelet transform,” in *IEEE International Conference on Signal Processing and Communications*, 2007, pp. 1291–1294.

[60] M. Muhlich and T. Aach, “High accuracy feature detection for camera calibration: A multi-steerable approach,” vol. 4713. Springer, 2007, p. 284.

[61] L. Lucchese and S. Mitra, “Using saddle points for subpixel feature detection in camera calibration targets,” in *Circuits and Systems, 2002. APCCAS’02. 2002 Asia-Pacific Conference on*, vol. 2, 2002.

[62] D. Chen and G. Zhang, “A new sub-pixel detector for X-corners in camera calibration targets,” in *Proceedings of the WSCG International Conference in Central Europe on Computer Graphics and Visualization*, 2005, pp. 97–100.

[63] E. Sojka, “A new algorithm for detecting corners in digital images,” in *Proceedings of the 18th spring conference on Computer graphics.* ACM New York, NY, USA, 2002, pp. 55–62.

[64] P. Beaudet, “Rotationally invariant image operators,” in *International Joint Conference on Pattern Recognition*, vol. 579, 1978, p. 583.

[65] R. Deriche and G. Giraudon, “A computational approach for corner and vertex detection,” vol. 10, no. 2. Springer, 1993, pp. 101–124.

[66] L. Kitchen and A. Rosenfeld, “Gray-level corner detection,” 1980.

[67] J. Noble, “Finding corners,” vol. 6, no. 2. Butterworth-Heinemann Newton, MA, USA, 1988, pp. 121–128.

[68] S. Smith and J. Brady, “SUSAN—A new approach to low level image processing,” vol. 23, no. 1. Springer, 1997, pp. 45–78.

[69] E. Sojka, “A New Algorithm for Detecting Corners in Digital Images,” Retrieved August 17, 2009, from http://www.cs.vsb.cz/sojka/cordet/presentation.html.

[70] O. Fernández, “Obtaining a best fitting plane through 3D georeferenced data,” vol. 27, no. 5. Elsevier, 2005, pp. 855–858.

[71] N. Woodcock, “Specification of fabric shapes using an eigenvalue method,” vol. 88, no. 9. Geological Soc America, 1977, pp. 1231–1236.

[72] M. Cohen, *Wittgenstein’s beetle and other classic thought experiments.* Blackwell Pub, 2005.

[73] E. W. Weisstein, “Law of Truly Large Numbers,” Retrieved August 17, 2009, from http://mathworld.wolfram.com/LawofTrulyLargeNumbers.html.

[74] A. Robinson, “Surface scanning with uncoded structured light sources,” Ph.D. dissertation, Sheffield Hallam University, UK, 2004.

[75] P. Team, “POV-Ray–the persistence of Vision Raytracer,” 2000.

[76] J. Shi and C. Tomasi, “Good features to track,” in *1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1994. Proceedings CVPR’94.*, 1994, pp. 593–600.

[77] S. Belongie, “Rodrigues’ Rotation Formula,” Retrieved August 17, 2009, from http://mathworld.wolfram.com/RodriguesRotationFormula.html.

[78] K. Ng, “Infrared Based Quantification of Breathing Rate,” Master’s thesis, Sheffield Hallam University, UK, 2008.

[79] L. Chan, “Development of an Ultrasound System to Monitor Breathing in Infants,” Master’s thesis, Sheffield Hallam University, UK, 2008.

[80] C. Chong, “Development of Stereo Vision Based Respiratory Rate Measuring System,” Master’s thesis, Sheffield Hallam University, UK, 2008.

[81] M. Folke, L. Cernerud, M. Ekström, and B. Hök, “Critical review of non-invasive respiratory monitoring in medical care,” vol. 41, no. 4. Springer, 2003, pp. 377–383.

[82] M. University, “Respiratory monitoring and other clinical applications of measurements of carbon dioxide in expiratory air ,” Retrieved August 17, 2009, from http://www.iss.mdh.se/index.php?choice=projects&id=0070.

[83] O. Bishop, *Practical Electronic Control Projects.* Bernard Babani Publishing, 1996.

[84] K. Konno and J. Mead, “Measurement of the separate volume changes of rib cage and abdomen during breathing,” vol. 22, no. 3. Am Physiological Soc, 1967, pp. 407–422.

[85] J. Cacioppo, L. Tassinary, and G. Berntson, *Handbook of psychophysiology.* Cambridge Univ Pr, 2007.

[86] K. Shelley, D. Jablonka, A. Awad, R. Stout, H. Rezkanna, and D. Silverman, “What is the best site for measuring the effect of ventilation on the pulse oximeter waveform?” vol. 103, no. 2. IARS, 2006, p. 372.

[87] M. Ogawa and T. Tamura, “Monitoring of heart and respiratory signals with PPG in bathing,” in *Engineering in Medicine and Biology, 1999. 21st Annual Conf. and the 1999 Annual Fall Meeting of the Biomedical Engineering Soc.] BMES/EMBS Conference, 1999. Proceedings of the First Joint*, vol. 2, 1999.

[88] L. Nilsson, T. Goscinski, S. Kalman, L. Lindberg, and A. Johansson, “Combined photoplethysmographic monitoring of respiration rate and pulse: a comparison between different measurement sites in spontaneously breathing subjects,” vol. 51, no. 9. Blackwell Publishing Ltd, 2007, pp. 1250–1257.

[89] G. Moody, R. Mark, M. Bump, J. Weinstein, A. Berman, J. Mietus, and A. Goldberger, “Clinical validation of the ECG-derived respiration (EDR) technique,” vol. 13, 1986, pp. 507–510.

[90] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” vol. 101, no. 23, 2000 (June 13), pp. e215–e220, circulation Electronic Pages: http://circ.ahajournals.org/cgi/content/full/101/23/e215.

[91] K. Raoof, P. Gumeiy, P. Mahul, and G. Quezel, “A Real Time Signal Processing System for respiratory,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, p. 369.

[92] P. Corbishley and E. Rodriguez-Villegas, “Breathing detection: towards a miniaturized, wearable, battery-operated monitoring system,” vol. 55, no. 1, 2008, pp. 196–204.

[93] S. Neaves, “Cradle of ingenuity Wireless sensors aim to prevent sudden infant death syndrome,” Retrieved August 17, 2009, from http://www.uta.edu/publications/utamagazine/winter_2008/index.php?id=565.

[94] A. Droitcour, S. University, and D. of Electrical Engineering, *Non-contact measurement of heart and respiration rates with a single-chip microwave Doppler radar.* Citeseer, 2006.

[95] J. Geisheimer and E. Greneker III, “Remote detection of deception using radar vital signs monitortechnology,” in *IEEE 34th Annual 2000 International Carnahan Conference on Security Technology, 2000. Proceedings*, 2000, pp. 170–173.

[96] A. Vergara and V. Lubecke, “Data Acquisition System for Doppler Radar Vital-Sign Monitor,” in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 2269–2272.

[97] K. Nakajima, Y. Matsumoto, and T. Tamura, “Development of real-time image sequence analysis for evaluating posture change and respiratory rate of a subject in bed,” vol. 22, no. 3. IOP PUBLISHING, 2001, pp. 21–28.

[98] Z. Zhu, J. Fei, and I. Pavlidis, “Tracking Human Breath in Infrared Imaging,” in *Proceedings of the Fifth IEEE Symposium on Bioinformatics and Bioengineering.* IEEE Computer Society Washington, DC, USA, 2005, pp. 227–231.

[99] R. Murthy, I. Pavlidis, and P. Tsiamyrtzis, “Touchless monitoring of breathing function,” in *Engineering in Medicine and Biology Society, 2004. IEMBS’04. 26th Annual International Conference of the IEEE*, vol. 1, 2004.

[100] S. Chekmenev, H. Rara, and A. Farag, “Non-contact, wavelet-based measurement of vital signs using thermal imaging,” vol. 2, no. 2, p. 2.

[101] H. Aoki, Y. Takemura, K. Mimura, and M. Nakajima, “Development of non-restrictive sensing system for sleeping personusing fiber grating vision sensor,” in *Micromechatronics and Human Science, 2001. MHS 2001. Proceedings of 2001 International Symposium on*, 2001, pp. 155–160.

[102] Senscomp, “MINI-A PB Ultrasonic Transducer,” Retrieved August 17, 2009, from http://www.senscomp.com/specs/Mini-A%20PB%20spec.pdf.

[103] Korundum, “RubyForge: Korundum/QtRuby,” Retrieved August 17, 2009, from http://rubyforge.org/projects/korundum/.

[104] J. Bouguet, “Fifth calibration example - Calibrating a stereo system, stereo image rectification and 3D stereo triangulation,” Retrieved August 17, 2009, from http://www.vision.caltech.edu/bouguetj/calib_doc/htmls/example5.html.

[105] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, and S. Sirotti, “Improving shadow suppression in moving object detection with HSV color information,” in *Proc. IEEE Int’l Conf. Intelligent Transportation Systems*. Citeseer, 2001, pp. 334–339.

[106] NavDyn, “Shadow detection,” Retrieved August 28, 2009, from http://dali.mty.itesm.mx/~autonomos/Navdyn/node11.html.

[107] The MathWorks, Inc, “Polynomial curve fitting,” Retrieved August 28, 2009, from http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/math/brfaisd-17.html&http://www.google.co.uk/url?sa=t&source=web&ct=res&cd=3&url=http%3A%2F%2Fwww.mathworks.com%2Faccess%2Fhelpdesk%2Fhelp%2Ftechdoc%2Fmath%2Fbrfaisd-17.html&ei=_LKwSuenJNGi4QavrYCtCg&rct=j&q=Polynomial+curve+fitting&usg=AFQjCNG69UAeIxTxWclJTzqfae6tEX0FfQ.

Appendix A

A1: Automatic detection of the initial X-corner locations in a planar object made of a regular pattern

Traditionally a planar object is made up of a checker board pattern. To locate the location of the corners of the black and white squares the user is expected to mark the outer corners of the pattern and give the interval between the squares of the regular pattern. Based on these two user inputs the corner detection algorithm will initialise the initial coarse location of the corners and then proceed with the estimation of the actual location of the corners.

There had been prior studies which had come out with recommendations to automate the process of the initial estimation of the corners of the checker board pattern.

In this study following method was used to automate the initialisation of the corner location process. The regular pattern made use of as the planar object is shown in the Figure A1. As it could be seen in the figure the planar object consists of colour square patterns embedded within some of the black squares (*i.e.* within the bigger out corners).

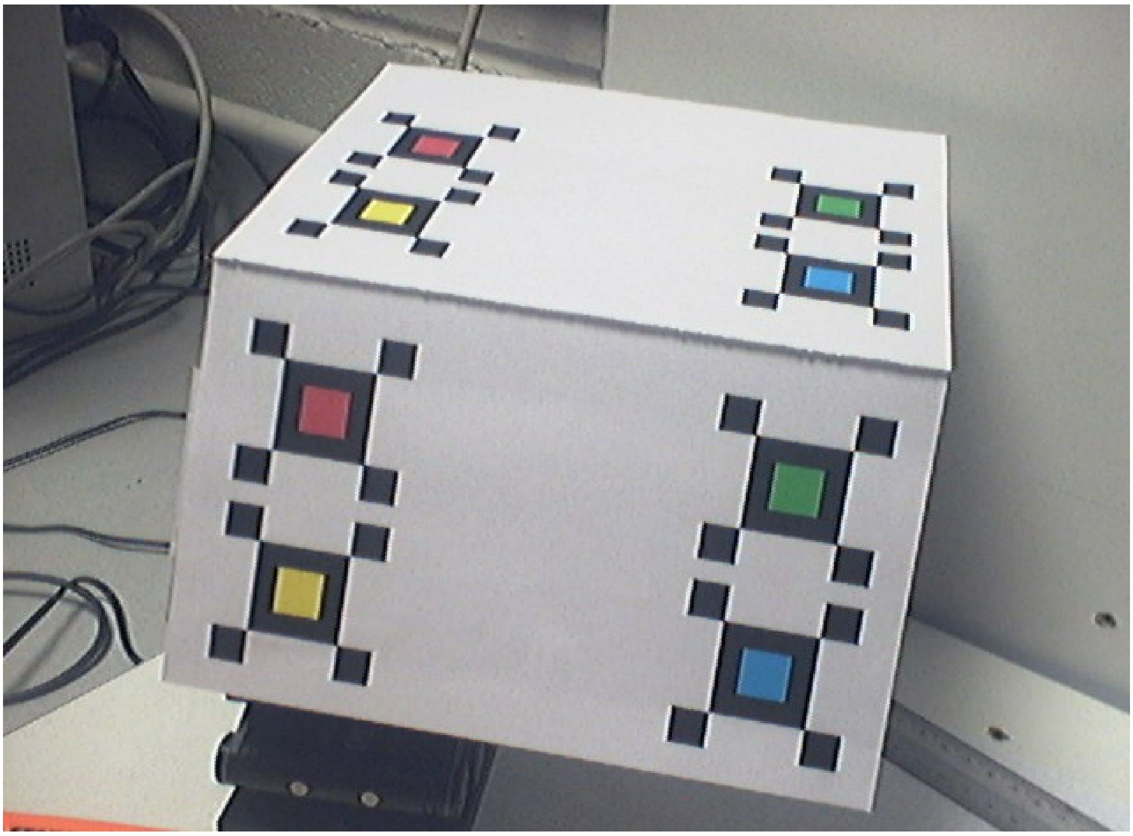


Figure A1: Figure showing the image of the planar object having X-corner type corners.

Figure A2 illustrates the assumed direction of *u*, the horizontal axis and *v* the vertical axis with the origin at the top left corner of the image. The location of the sixteen X-corners (4 corners around each colour blob) are marked by dark circles in the figure.

Following steps are involved to automatically locate the initial location of the sixteen corners.

Colour blob segmentation: Four colour blobs were segmented using the colour segmentation method described in Appendix A2.

Estimation of the centre of the Colour blob: Using the segmentation output, from the previous step, the pixel co-ordinates (u_{colour}, v_{colour}) of the pixels belonging to each of the different colours were collected in their own respective list of *u* and *v* values. The centre of each of the colour blob was estimated by finding the median values, u_{m_colour}, $v_{m\_colour;}$ of the pixel locations in each of the corresponding *u* and *v* list. The median value ensures that outlier pixel co-ordinates due to noise will have minimal influence on estimating the centre of the colour blob.

Estimation of the location of the sixteen corners in the grid co-ordinate system:

The estimated median value (u_{m_colour}, v_{m_colour}) in each of the colour list was assumed as the corresponding image coordinates of the projected centre point (Figure A2) of each coloured square on to the planar object. For *e.g.* (u_{m_red}, v_{m_red}) in the image co-ordinate, is the projected centre point (0.5, 0.5) in the grid co-ordinate system.

These four correspondences were used to estimate the projective transformation matrix (eight unknowns, with h_9 normalised to one) by solving the set of equations (A.1). The system of equations was solved by using the linear least square estimation which is the eigenvector corresponding to the smallest eigenvalue of the matrix **M**

Estimation of the pixel co-ordinates of the sixteen corners: Using the projective transformation matrix the grid coordinates of the sixteen corner locations were transformed into the corresponding image pixel locations using the equation (A.2).

These sixteen corner locations are used as the initial coarse location of the corners and corner detection was applied around these locations within a ten by ten region.

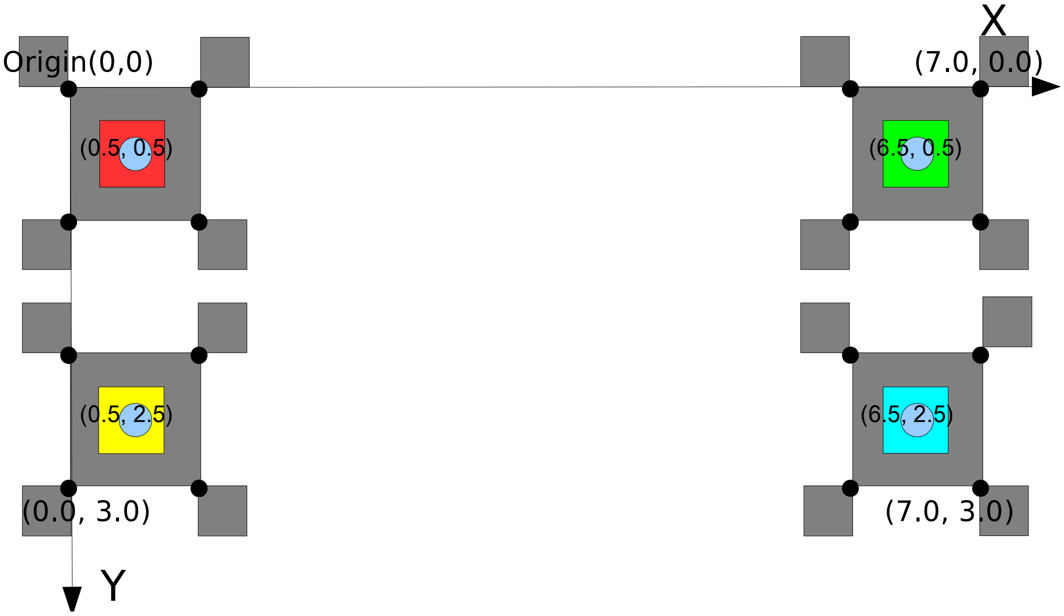


Figure A2 : Figure showing the assumed coordinate system and the location of the sixteen X-corners, marked with dark coloured circles.

$$\underbrace{\begin{pmatrix} 0.5 & 0.5 & 1.0 & 0.0 & 0.0 & 0.0 & 0.5*u_{\mathrm{m\_red}} & 0.5*u_{\mathrm{m\_red}} & u_{\mathrm{m\_red}} \\ 0.0 & 0.0 & 0.0 & 0.5 & 0.5 & 1.0 & 0.5*v_{\mathrm{m\_red}} & 0.5*v_{\mathrm{m\_red}} & v_{\mathrm{m\_red}} \\ 6.5 & 0.5 & 1.0 & 0.0 & 0.0 & 0.0 & 6.5*u_{\mathrm{m\_green}} & 0.5*u_{\mathrm{m\_green}} & u_{\mathrm{m\_green}} \\ 0.0 & 0.0 & 0.0 & 6.5 & 0.5 & 1.0 & 6.5*v_{\mathrm{m\_green}} & 0.5*v_{\mathrm{m\_green}} & v_{\mathrm{m\_green}} \\ 6.5 & 2.5 & 1.0 & 0.0 & 0.0 & 0.0 & 6.5*u_{\mathrm{m\_blue}} & 2.5*u_{\mathrm{m\_blue}} & u_{\mathrm{m\_blue}} \\ 0.0 & 0.0 & 0.0 & 6.5 & 2.5 & 1.0 & 6.5*v_{\mathrm{m\_blue}} & 2.5*v_{\mathrm{m\_blue}} & v_{\mathrm{m\_blue}} \\ 0.5 & 2.5 & 1.0 & 0.0 & 0.0 & 0.0 & 0.5*u_{\mathrm{m\_yellow}} & 2.5*u_{\mathrm{m\_yellow}} & u_{\mathrm{m\_yellow}} \\ 0.0 & 0.0 & 0.0 & 0.5 & 2.5 & 1.0 & 0.5*v_{\mathrm{m\_yellow}} & 2.5*v_{\mathrm{m\_yellow}} & v_{\mathrm{m\_yellow}} \end{pmatrix}}_{:=M} \underbrace{\begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{pmatrix}}_{:=\vec{h}} = \begin{pmatrix} \varepsilon'_{u\_red} \\ \varepsilon'_{v\_red} \\ \varepsilon'_{u\_green} \\ \varepsilon'_{v\_green} \\ \varepsilon'_{u\_blue} \\ \varepsilon'_{v\_blue} \\ \varepsilon'_{u\_yellow} \\ \varepsilon'_{v\_yellow} \end{pmatrix} \tag{A.1}$$

$$\begin{bmatrix} \lambda u \\ \lambda v \\ \lambda \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{A.2}$$

A2: Simple colour segmentation

Let *p* be the rgb pixel value of each pixel in the image and *pr*, *pg*, *pb* be the corresponding value of red, green and blue channel.

Let *ratio_red*, *ratio_blue, ratio_yellow* and *ratio_green* be the threshold value for the ratio of red, blue, yellow, and green colour, with respect to the other colours, respectively. These thresholding values are from 0.60 to 1.10

Let $red_1 = 100$, $blue_1 = 100$, $yellow_1 = 130$, $yellow_2 = 120$, $yellow_3 = 120$, $green_1 = 90$, $green_2 = 150$.

be the threshold values for each of the colour channels pixel values.

All these values are getting from manually observation of image.

The above threshold values were found by manually selecting the respective colour pixels in the image and observing the values of the different channels.

Following pseudo code outlines the the colour segmentation process which was used in the automatic coarse location of the corners (Appendix A1).

Using the following conditions one can decide to which of the four different colour blobs (red, green, blue, yellow) a pixel, *p*, belongs.

if($pr/(pb+pg) > ratio\_red \;\&\&\; pr > red_1$) //this pixel might belong to the red colour blob

if($pb/(pr+pg) > ratio\_blue \;\&\&\; pb > blue_1$) //this pixel might belong to the blue colour blob

if ($pr/pg > ratio\_yellow \;\&\&\; pr > yellow_1 \;\&\&\; pg > yellow_2 \;\&\&\; pb < yellow_3$) //this pixel might belong to the yellow colour blob

if($pg/(pb+pr) > ratio\_green \;\&\&\; pg > green_1 \;\&\&\; pg < green_2$) //this pixel might belong to the green colour blob

Appendix B

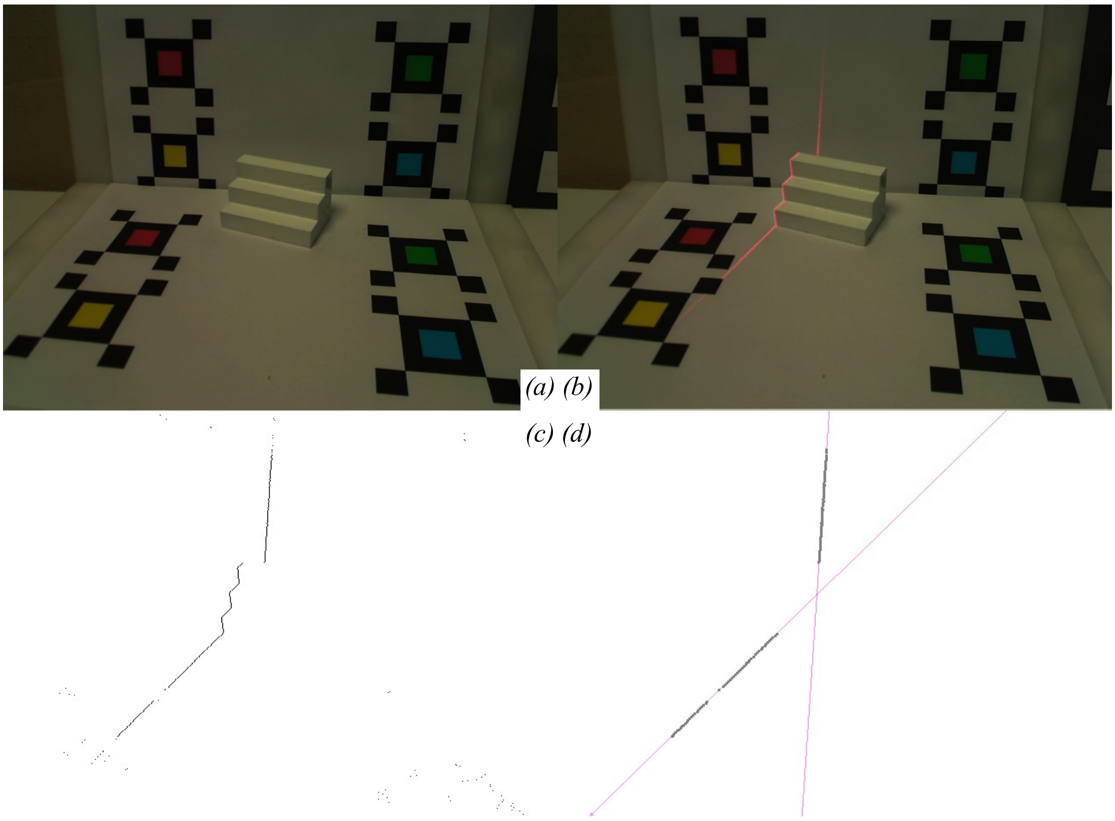
B.1 Code in Postscript scripting language to generate synthetic images. The sample image created with the code was shown in Figure 4.35a

```
%begining of code to create the sample image
/inch {25 mul} def      % Convert inches->points (1/72 inch)
/image {
0 setlinewidth
0 setgray
newpath                         % Start a new path
-2 inch 2 inch moveto
4 inch 0 inch rlineto % an inch in from the lower left
0 inch -4 inch rlineto
-4 inch 0 inch rlineto
0 inch 4 inch rlineto
closepath               % Automatically add left side to close path
fill                    % Fill in the box on the paper
newpath                         % Start a new path
-6 inch 6 inch moveto
4 inch 0 inch rlineto % an inch in from the lower left
0 inch -4 inch rlineto
-4 inch 0 inch rlineto
0 inch 4 inch rlineto
closepath               % Automatically add left side to close path
fill                    % Fill in the box on the paper
newpath                         % Start a new path
2 inch 6 inch moveto
4 inch 0 inch rlineto % an inch in from the lower left
0 inch -4 inch rlineto
-4 inch 0 inch rlineto
0 inch 4 inch rlineto
closepath               % Automatically add left side to close path
fill                    % Fill in the box on the paper
newpath                         % Start a new path
-6 inch -2 inch moveto
```

```
4 inch 0 inch rlineto % an inch in from the lower left
0 inch -4 inch rlineto
-4 inch 0 inch rlineto
0 inch 4 inch rlineto
closepath                  % Automatically add left side to close path
fill                       % Fill in the box on the paper
newpath                            % Start a new path
2 inch -2 inch moveto
4 inch 0 inch rlineto % an inch in from the lower left
0 inch -4 inch rlineto
-4 inch 0 inch rlineto
0 inch 4 inch rlineto
closepath                  % Automatically add left side to close path
fill                       % Fill in the box on the paper
} def
gsave
12 inch 16 inch translate
0 rotate   %set the degree of rotation here
0 0 image stroke
grestore
%%BoundingBox: 75 170 525 630
showpage                   % We're done... eject the page
%end of code
```

Appendix C:

C1: Intermediate results of the experiments carried out in chapter 5



(a) (b)

(c) (d)

Figure C.1: The intermediate images of the experiment setting.

(a) Background image. (b) Laser image. (c) Result of background subtraction. (d) Two detected laser lines one on the horizontal plane and one on the vertical planes.