



University Library

Author/Filing Title OGUNDANA, O.O.

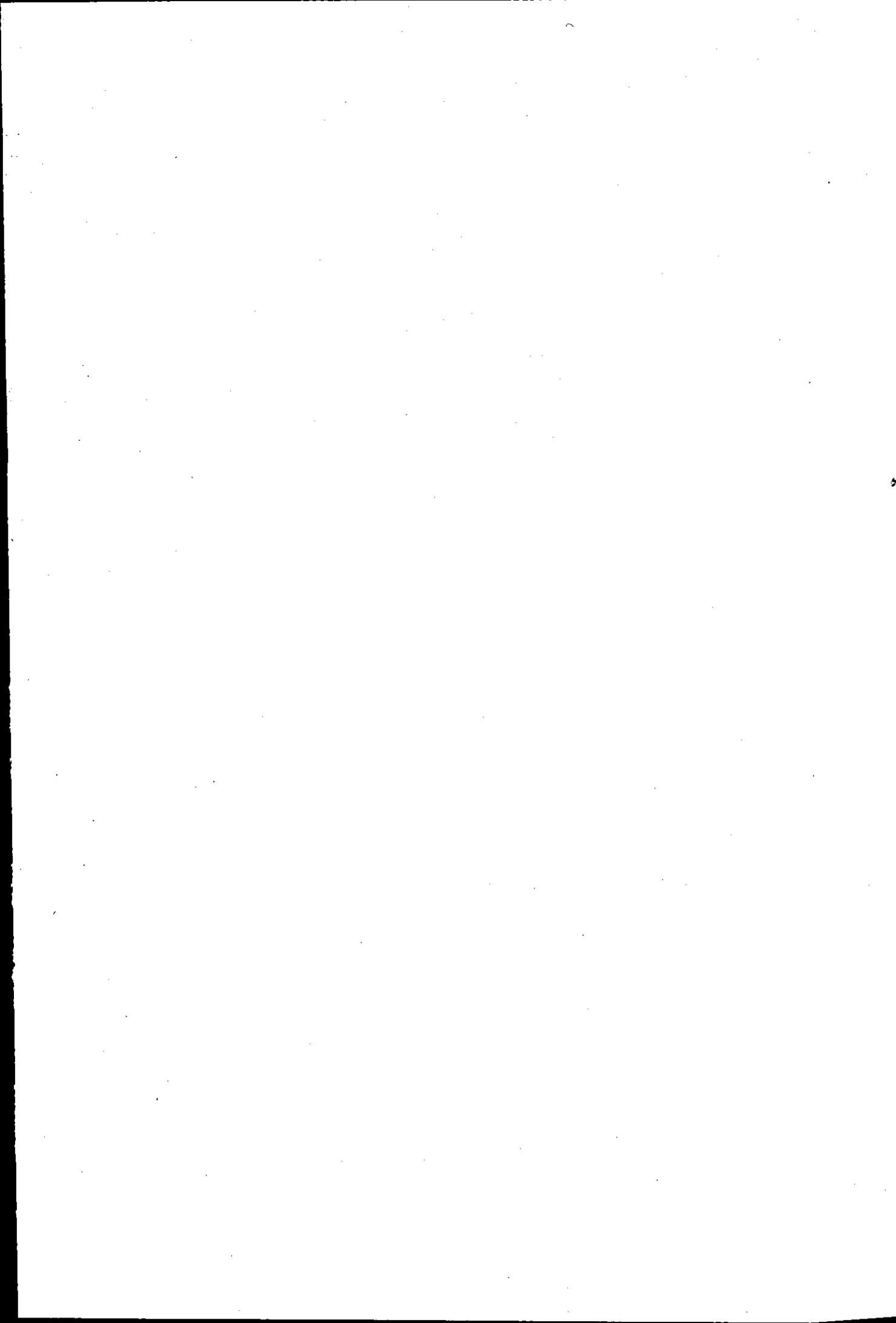
Class Mark T

**Please note that fines are charged on ALL
overdue items.**

--	--	--

0403604265





Automated Calibration of Multi-Sensor Optical Shape Measurement System

By

Olatokunbo Omodele Ogundana

A Doctoral Thesis

Submitted in partial fulfilment of the requirements

for the award of

Doctor of Philosophy of Loughborough University

April, 2007

© by Olatokunbo Omodele Ogundana, 2007



Loughborough
University
Pilkington Library

Date Nov '08

Class T

Acc No. 0403604265

Acknowledgements

Financial support through a Scholarship from Loughborough University and Airbus UK is gratefully acknowledged. My sincere thanks and appreciation go to my supervisors - Prof. Jon Huntley, Dr. Russell Coggrave (Phase Vision Ltd) and Dr. Richard Burguete (Airbus UK), for their technical input, academic support, constant encouragement and friendship during the course of the research project.

Also appreciated is Dr. Jim Chandler of the Department of Civil Engineering for useful discussions on calibration techniques in photogrammetry. Many thanks to Jagpal Singh of the Metrology Laboratory for making artefacts available, and also for providing assistance and support whenever I had to make any of my numerous measurements with the mechanical coordinate measuring machine (CMM). Members of the Structural Integrity Research group are appreciated for sharing their knowledge and experience, and also for their valuable advice and friendship.

To my friends, both at Loughborough and beyond, who have always been a constant source of encouragement, I say thank you.

Finally, I would like to thank my parents and members of my family for their patience, understanding and consistent support through several years of study.

Abstract

A multi-sensor optical shape measurement system (SMS) based on the fringe projection method and temporal phase unwrapping has recently been commercialised as a result of its easy implementation, computer control using a spatial light modulator, and fast full-field measurement. The main advantage of a multi-sensor SMS is the ability to make measurements for 360° coverage without the requirement for mounting the measured component on translation and/or rotation stages. However, for greater acceptance in industry, issues relating to a user-friendly calibration of the multi-sensor SMS in an industrial environment for presentation of the measured data in a single coordinate system need to be addressed.

The calibration of multi-sensor SMSs typically requires a calibration artefact, which consequently leads to significant user input for the processing of calibration data, in order to obtain the respective sensor's optimal imaging geometry parameters. The imaging geometry parameters provide a mapping from the acquired shape data to real world Cartesian coordinates. However, the process of obtaining optimal sensor imaging geometry parameters (which involves a nonlinear numerical optimization process known as bundle adjustment), requires labelling regions within each point cloud as belonging to known features of the calibration artefact. This thesis describes an automated calibration procedure which ensures that calibration data is processed through automated feature detection of the calibration artefact, artefact pose estimation, automated control point selection, and finally bundle adjustment itself.

The process of calibration artefact selection is discussed, with the objective of developing a low cost artefact, with appropriate geometric and material properties such as unobstructed viewing by sensors, low coefficient of thermal expansion and non-specular surface finish. Automated detection of calibration artefact features is investigated, for enhancing the ease, speed, and accuracy of calibration. A novel 3-D Hough transform based on an optimised sparse 3-D matrix model is described, including methods developed for efficient peak detection in the Hough accumulator space.

The calibration results of a two-camera and two-projector optical SMS based on multiple poses of the respective calibration artefacts developed, are discussed. A comparison of usage of the calibration artefacts is also made in order to assess their practicable use in an industrial environment. Based on acquired shape data of one of the artefacts, calibration accuracy of about one part in 5,000 was achieved.

In applications for product inspection and quality assessment, the measured data needs to be presented in a form that provides for visualisation on a computer. A method for efficiently tessellating the measured point cloud using sensor pixel neighbourhood information is described. This method provides for the presentation of the measured point cloud data in industry accepted file formats. In addition, the measured data of a component may need to be compared against an idealised model of the component e.g. a computer-aided design (CAD) model. Methods for matching the measured data to a CAD model are therefore also discussed.

The results of calibrating a multi-sensor SMS at an industrial site are presented. In spite of a less well controlled environment, a calibration accuracy of about one part in 1,600 was achieved, with the SMS subsequently used as a valuable tool for measuring out-of-plane displacement during a series of structural tests.

Table of contents

Acknowledgements.....	ii
Abstract.....	iii
Table of contents.....	v

Section I - Introduction and literature survey

1	Introduction.....	1
1.1	Background.....	1
1.2	Project aims and objective.....	2
1.3	Fundamental concepts of measurement.....	5
1.3.1	Accuracy and precision	5
1.3.2	Measurement standards	6
1.4	Coordinate measuring machines.....	8
1.5	Measurement principle of optical SMS.....	9
1.5.1	Phase measurement.....	9
1.5.2	Calibration	11
1.5.3	Data processing and presentation	11
1.5.4	Sensor design.....	11
1.6	Thesis outline.....	12
1.7	Figures	15
2	Sensor Calibration Methods	19
2.1	Introduction	19
2.2	Sensor calibration techniques	20
2.2.1	Calibration in close range photogrammetry	20
2.2.2	Calibration in computer vision	25
2.2.3	Classifications of sensor calibration techniques	27
2.2.4	Calibration methods for structured light measurement systems.....	33
2.3	Multiple sensor calibration methods.....	36
2.4	Calibration artefact design.....	38
2.4.1	Geometric considerations	38
2.4.2	Material considerations.....	42
2.5	Summary.....	43
2.6	Figures	45

Section II - Development of automated calibration method

3	Description of Model for New Calibration Process	50
3.1	Introduction	50
3.2	Calibration concept.....	50
3.2.1	Model for bundle adjustment.....	50
3.2.2	Computing a point cloud	52
3.2.3	Description of the 'pose' of artefact	53
3.3	Bundle adjustment in object space	55
3.3.1	Description of expression for minimisation	55
3.3.2	Optimisation of parameters	56
3.4	New calibration process.....	57

3.5	Summary.....	59
3.6	Figures	61
4	Sphere Artefacts and Sphere Detection.....	65
4.1	Introduction	65
4.2	Multiple-sphere artefacts	66
4.2.1	Arrangement of spheres.....	66
4.2.2	Manufacturing Process	66
4.3	Ball bar	67
4.4	Automated Feature Detection.....	69
4.4.1	Objective.....	69
4.4.2	Finding Circles using the Hough transform	69
4.4.3	Finding sphere centres using the Hough Transform.....	73
4.4.4	Sparse matrix representation	75
4.4.5	Peak detection in Hough space	78
4.4.6	Identification of non-viable peaks	82
4.4.7	Post-processing of sphere centres.....	83
4.4.8	Experimental Results.....	87
4.5	Summary.....	89
4.6	Tables.....	91
4.7	Figures	93
5	Plane Artefacts and Plane Detection	108
5.1	Introduction	108
5.2	Artefact geometry	109
5.3	Artefact component parts.....	109
5.4	Weight of artefact	110
5.5	High precision planar parts.....	111
5.5.1	Gauge blocks	111
5.5.2	Surface plates.....	111
5.5.3	Optical flats.....	112
5.5.4	Precision ground blocks.....	112
5.5.5	Precision ground plates.....	113
5.6	Selection of parts	113
5.7	Plane detection using the 3-D Hough transform	113
5.7.1	Finding Lines using the Hough Transform.....	113
5.7.2	Parameterization for Plane Detection	115
5.7.3	Voting in 3-D Hough space and parameter estimation.....	117
5.7.4	Voting in 1-D Hough space	120
5.8	Results of 3D Hough Transform for Planes	122
5.8.1	Introduction	122
5.8.2	Simulated data	122
5.8.3	Experimental data.....	125
5.9	Summary.....	127
5.10	Tables.....	129
5.11	Figures	131
6	Implementation of Calibration Process	144
6.1	Introduction	144
6.2	Shape Data Acquisition	146

6.3	Sensor parameter initialisation	147
6.3.1	2-D DLT method	147
6.3.2	Calibration artefact for sensor parameter initialisation	151
6.4	Shape data post-processing.....	153
6.4.1	Initialising pose parameters for artefacts.....	153
6.4.2	Control point selection.....	154
6.5	Bundle adjustment in object space	156
6.5.1	Relationship between pose parameters and control point	156
6.5.2	Parameter constraints for multiple-sphere or ball bar artefact	160
6.5.3	Parameter constraints for two-plane artefact	161
6.6	Software implementation.....	162
6.6.1	Extensible markup language (XML) description	162
6.6.2	Description of calibration XML document	163
6.6.3	Structure of software	164
6.7	Experimental results	166
6.7.1	Description of experiments.....	166
6.7.2	Initialisation of sensor parameters using the 2-D DLT method	168
6.7.3	Results for calibration using multiple-sphere artefact.....	169
6.7.4	Results of calibration using ball bar artefact	172
6.7.5	Results of calibration using two-plane artefact	175
6.7.6	Comparison of results from calibration using different types of artefacts.....	177
6.8	Summary.....	180
6.9	Tables.....	184
6.10	Figures	191

Section III - Post-processing of measured coordinates

7	Matching of Measured Coordinate Data to CAD Models.....	226
7.1	Introduction	226
7.2	Meshing of point cloud from optical SMS	228
7.2.1	Description of algorithm for tessellating point cloud.....	228
7.2.2	Data representation in CAD Software (CAD file formats)	230
7.2.3	STL file format	231
7.2.4	OBJ file format	232
7.2.5	Results for exporting measured data for visualisation.....	234
7.3	Methods for registration of point clouds	235
7.4	Iterative Closest Point Algorithm (ICP)	236
7.4.1	Description of the Conventional ICP.....	236
7.4.2	Efficient Variants of the ICP	238
7.5	Registration using a 3-D look-up table.....	241
7.5.1	Description of method	241
7.5.2	Description of C++ implementation.....	245
7.6	Summary.....	246
7.7	Tables.....	248
7.8	Figures	249
8	Application of Multi-Sensor Optical Shape Measurement System	259

8.1	Introduction	259
8.2	Structural testing in the Aerospace industry.....	260
8.2.1	Types of structural tests	260
8.2.2	Compression panel tests	261
8.3	Set up of SMS for measuring out-of-plane displacement.....	264
8.4	Description of calibration process at Airbus' site and results	266
8.4.1	Initialisation phase – shape data acquisition and sensor parameter initialisation	266
8.4.2	Refinement phase – shape data acquisition, shape data processing and bundle adjustment.....	267
8.5	Procedure for making measurements during test	270
8.6	Results from structural test.....	271
8.7	Summary.....	272
8.8	Tables.....	274
8.9	Figures	275

Section IV - Conclusions

9	Conclusions.....	291
9.1	Discussions	291
9.2	Further work	295

References

References.....	300
-----------------	-----

Appendices

Appendix – A-1 – Optimisation of sphere centre coordinates	A-1
Appendix – A-2 – Contact details of suppliers for plane artefact components.....	A-5
Appendix – A-3 – XML interface for calibration software -showing bundle adjustment section	A-7
Appendix – A-4 – Description of folders for calibration software MATLAB [®] functions and C mex files	A-11
Appendix – A-5 – C++ interface for implementation of adaptive look-up table (LUT).....	A-15
Appendix – A-6 – Journal and conference papers.....	A-23

Section I –
Introduction and
literature survey

1 Introduction

1.1 Background

Object shape can be described in terms of geometric information which includes features and attributes for arrangement in space – whether 1-D, 2-D or 3-D features. Such features include a variety of different types of curves and surfaces of varying levels of complexity, which when combined together, can be used to describe a variety of objects. Thus, the shape of an object provides a unique way for describing and identifying an object. In real life, objects have complex shapes and with developments in computer technology, computer-aided design (CAD) and computer-aided manufacture (CAM), such objects can be designed on computer and then manufactured. Dimensional measurement of sculptured surfaces is becoming more and more important in manufacturing, since many products are designed and manufactured as sculptured surfaces with a requirement for high precision [1]. Therefore, many industries require rapid and precise measurement of the surface profile of manufactured components.

In product development and manufacturing, Designers and Engineers depend on computer-aided design (CAD) software for designing their products. On manufacturing such products, to ensure conformance to intended design, it is common practice that the shape of the manufactured product be measured and checked against an idealised model, e.g. a CAD model. As part of the quality assessment of certain products, structural integrity testing may be required, where a test specimen is intentionally deformed and such deformation would need to be measured. Also, as in-service parts age, they wear, and in order to avoid accidents and to increase working efficiency, part dimensions need to be periodically checked to identify if they need to be repaired or replaced [2]. It is in situations such as these that the technology of 3-D shape measurement systems has become relevant and important, providing the means for accurately measuring the shape of 3-D objects. A variety of applications of 3-D shape measurement include control for intelligent robots, obstacle detection for vehicle guidance, dimension measurement for die development, stamping, panel geometry checking, and accurate stress, strain and vibration measurement [3].

The typical instruments for measuring surface shape are mechanical coordinate measuring machines (CMMs), which measure surface shape by point-wise probing on as many positions on the object as possible. In recent years, optical shape measurement systems (SMS), which require no contact with the object surface, have also been developed, with 360 degree measurement of an object made possible by combining multiple sensors. However, in comparison with CMMs, though optical SMSs provide significantly more 3-D coordinates in much shorter time scales, a combination of conditions have restricted their use in industry. For example, environmental conditions which are typically found at industrial sites, such as low-frequency vibration, uncontrolled temperature and background illumination, would adversely affect measurement accuracy and reliability [4]. On the other hand are issues such as the complexity of calibration and measurement procedure, which would also affect measurement accuracy and instrument set up time, and could lead to the requirement for an expert user in order to achieve acceptable measurement accuracy. Thus the running cost of using an optical SMS for regular measurement tasks at an industrial site becomes potentially prohibitive. Therefore, the key objective of this work is the development of an automated calibration method that would be suitable for an industrial environment and easily adaptable for a wide range of measurement volumes.

This chapter aims to provide an introduction to this thesis by highlighting the project's aims and objectives. Also, some concepts of measurement, namely accuracy, precision and traceability are introduced, including a description of Mechanical CMMs and optical SMSs. However, more attention is focussed on the measurement principle of an optical SMS. Finally, the chapter concludes with an outline of the thesis, indicating the key discussions points in subsequent chapters.

1.2 Project aims and objective

Optical methods provide a non-contact means of evaluating the physical properties of objects. A variety of optical methods have been developed for measuring 3-D shape to high precision. These include: time or light in flight, laser scanning, moiré method, laser speckle pattern sectioning, interferometry, photogrammetry, laser tracking system, and projected structured light [3]. Detailed discussions on these methods have been made by Chen et al [3] and Coggrave [4]. However, the method of focus here

will be projected structured light systems, and more specifically, fringe projection systems. Therefore, any reference hereafter to an optical SMS will refer to a fringe projection system. It should be noted that in fringe projection systems, the projected fringe would normally be a binary (e.g. Gray code systems) or sinusoidal intensity profile [5]. However, discussions will focus only on systems that project sinusoidal intensity profiles.

Research has been going on at Loughborough University since 1996 on an optical SMS based on projection of sinusoidal fringe patterns using low cost data projector and digital camera to obtain 3-D coordinates [4, 6-11]. Developments led to a patented technique for combining different fringe patterns. The technique involves projecting a sequence of computer-generated patterns of light and dark fringes onto the object using a spatial light modulator. Analysis of the data from an off-axis digital camera allows a depth value to be calculated at each camera pixel. A spin out company, Phase Vision Ltd, was created to assist in the transfer of this technology to industry. Applications where Loughborough's optical SMS (Figure 1-1) has been applied include:

1. Aircraft spar testing at Airbus (Figure 1-2): the optical SMS was used for measuring the out-of-plane displacement field over a millimetre length scale. A 32 tonne load was applied to the specimen and the resulting displacement due to the load was measured.
2. Measuring the cylinder head volume of a Jaguar V8 engine.(Figure 1-3 (a) and (b)) [12]: in the conventional method, the volume of oil required to fill the chamber was measured and used for estimating the cylinder head volume. This took a considerable amount of time, requiring typically 2 hours for each engine block. However, using an optical SMS, the volume was measured within a few seconds, with agreement to within 0.5%.

The steps in the data acquisition and processing chain of an optical SMS include [13]: phase measurement, calibration, data processing, and data presentation, which will be introduced in Section 1.5. Problems associated with an optical SMS usually occur at each of these steps. Past research at Loughborough [4, 6-11], has focussed on phase measurement, and in this work attention is now focussed on the subsequent steps, namely calibration, data processing and data presentation of an optical SMS.

In terms of application, this project aims to extend the use of optical SMSs to both structural testing and manufacturing in the Aerospace industry. Sponsored by Airbus UK, the project is focused on measuring components of aircraft wings used in civil aircraft such as the new Airbus A380, where the complex surface profile of large aluminium skin panels are up to 35 m in length. The optical SMS is expected to be used for measuring out-of-plane displacement during structural failure tests, where load is dynamically applied to a test specimen. The large length scales of the test specimen and consequent difficulty in making whole body (i.e. 360°) measurement as a result of their sheer size, pose a number of challenges that this thesis seeks to address. It is obvious that in such an application, point-wise measurements using, say a mechanical CMM, is not feasible. Therefore, the primary objective of this work is the development of an automated calibration method suitable for a multi-sensor optical SMS that will be used in an industrial environment (i.e. where absolute control over environmental conditions, such as temperature, vibration and illumination, is not practicable) and easily adaptable for a wide range of measurement volumes. The importance of automating the procedures and methods for multi-sensor systems has been highlighted by Clarke et al. [14]. It is expected that the incorporation of such a novel calibration method would ensure that the optical SMS is easily deployable and usable for a variety of applications, thereby enhancing the adoption of the optical SMS in industry.

The automated calibration process developed in this work is model based rather than empirical, thus providing a faster, automated, user-friendly, and traceable calibration. The model developed is based on a photogrammetric approach, thus combining the advantages of a fringe projection system that has high coordinate throughput, and photogrammetry that provides for a robust and accurate calibration [15]. In establishing the mapping from the measured phase values to real world Cartesian coordinates, parameters are described for the constituent sensors in the optical SMS (such as translation parameters, rotation parameters (Euler angles), lens centre, and lens distortion parameters). In addition, the calibration technique is expected to be useable over large measurement volumes (greater than the order of $1 \times 1 \times 1 \text{ m}^3$), thus providing high precision measurements over large areas of object surface at any one

time, and thereby overcoming the limitations of point-wise measurement. It also incorporates a means of linking multiple SMSs in a modular fashion, in order to provide dimensional measurements with 360° surface coverage. The technique is therefore expected to lead to rapid deployment of the system at new measurement sites, have immunity to environmental disturbances, and offer an improvement in the current calibration accuracy [4, 15].

The optical SMS developed at Loughborough has high scan rates (of order 10^6 s^{-1}), and the 3-D coordinates calculated from it is used to generate a 'point cloud' containing geometric information on the shape of the object under inspection. It is desirable that the measured shape data be presented in a form that can be exported to other software systems for visualisation purposes. With such a vast quantity of shape data acquired by the SMS, another key aspect of this work was the development of efficient and robust techniques for tessellating the measured cloud of points in order to export to industry accepted file formats for visualisation on a computer. In addition, efficient techniques that use the measured data in conjunction with its idealised model (e.g. CAD model) for automated quality control, are discussed.

1.3 Fundamental concepts of measurement

1.3.1 Accuracy and precision

In the modern world, metrology plays a vital role to protect the consumer and to ensure that manufactured goods conform to prescribed dimensional and quality standards [16]. Quite often, the terms 'accuracy' and 'precision', are incorrectly used interchangeably. The accuracy of a measurement is the degree of closeness to the *true value* (a quantity whose value has in principle been measured without error) [16, 17]. Since the *true value* of a measurement cannot be obtained experimentally, the only way to obtain an estimate of accuracy is to use a higher level measurement standard in place of the measuring instrument to perform the measurement and to then use the resulting mean value as the *true value* [16]. Accuracy is thus commonly expressed as the difference between the measurement result of the instrument and the result of that of a higher level measurement standard [16].

On the other hand, the precision of a measurement is the degree of scatter of the measurement result, when the measurement is repeated a number of times under

specified conditions. Although the term precision is used only in the general sense, it can also be quoted as a numerical characteristic of a measurement system. Precision can also be explained in terms of performance parameters of a measurement system – discrimination, repeatability and reproducibility. Discrimination is the quality that characterizes the ability of the measuring instrument to react to small changes of the quantity being measured [16, 17]. If a quantity is gradually changing, the ability of the measuring instrument to detect such a change is its discrimination. Repeatability refers to the closeness of the agreement between results of successive measurements carried out under the same conditions of measurement within a relatively short interval of time [16, 17]. Reproducibility is the closeness of agreement between results of successive measurements carried out under changed conditions of measurement i.e after a sufficiently long period of time [16, 17]. The conditions of measurement to be considered in terms of repeatability and reproducibility include [16]:

1. Measurement procedure
2. The observer
3. Environmental conditions
4. Location

Repeatability is often expressed as the standard deviation of the measurement result. Reproducibility is rarely computed in metrology, though widely used and useful in certain cases [16].

1.3.2 Measurement standards

The concept of a *true value* for every measurement provides a means for specifying the accuracy of a measurement system. However, since it is impossible to get a completely error-free measurement, the accuracy of a measurement is described in terms of a *conventional true value*. The *conventional true value* is obtained by comparing the test item with a higher-level measurement standard under defined conditions [16]. There are therefore different levels of measurement standards.

Measurement standards can be categorized into levels based on metrological quality or geographical location [16]. With regard to metrological quality, the different levels of measurement standards include:

1. Primary
2. Secondary

3. Working

Primary standards have the highest metrological quality and their values are not referenced to other standards of the same quantity. They are maintained by an international network of national physical standards laboratories, such as the National Physical Laboratory (NPL) of the UK, and National Institute of Science and Technology (NIST) of the USA. The International Bureau of Weights and Measures (BIPM) maintain the primary standards to which national physical standards laboratories of different countries reference their standards. The base units of the SI metric metrology system are metre, kilogram, second, ampere, Kelvin, candela. For example, the base unit of length, the metre, is defined by the BIPM as “the length of the path travelled by light in vacuum during a time interval of $1/299\,792\,458$ of a second ” [18]. From these base units, other units are derived for quantities such as area, volume, speed, etc.

Secondary standards are standards whose values are assigned by comparison with primary standards of the same quantity. A working standard, which is at the bottom of the hierarchy of measurement standards, is a standard used to calibrate or check measurement accuracy of instruments. Industrial and day-to-day measuring instruments therefore have a working standard.

Based on geographical location, measurement standards can be categorised into the following levels:

1. National primary standard
2. Secondary standard
3. Tertiary standard

The above refers to the hierarchy within a given country. Usually, the national hierarchy scheme is incorporated in the metrology law of the country [16]. The maintenance of a hierarchy of measurement standards provides the key concept to *traceability* in measurement science. The traceability of a measuring instrument signifies that its value has been determined by an unbroken chain of comparisons with a series of higher level standards with stated uncertainties [16]. Such comparisons are done within time frames depending on the specific requirements for the measuring system and their acceptable uncertainty.

1.4 Coordinate measuring machines

Coordinate measuring machines (CMMs) are used to obtain 3-D point data from objects, from where the shape of such objects can be deduced. The power of the CMM is derived from its ability to compute, from measured points, any one of a whole family of types of dimensional quantities: position of features relative to part coordinates, distances between features, sizes of features, forms of features, such as flatness, circularity, and cylindricity, and angular relationships between features, such as perpendicularity [19]. Conventional mechanical CMMs measure an object's shape by probing its surface at discrete measuring points. However, as object parameters such as diameter, distance or angle cannot be evaluated directly from the coordinates of the measured points, some best-fit algorithm is applied to the measured data set [20].

Typically, a CMM consists of a granite base, a main structure (bridge, horizontal, vertical, gantry, etc.), a probing system (consisting of a probe head and stylus) and software (see Figure 1-4 which shows a CMM with a bridge structure). Major brands of CMMs include products from Zeiss, Metris LK, Leitz, Mitutoyo, and Hexagon Metrology. The cost of CMMs is highly dependent on specifications expressed as a statement of accuracy based on ISO 10360, which includes the achievable uncertainty of measurement, probing error and scanning error. However, these quantities are consequences of a variety of factors such as mechanical and thermal stability of components, type of probe head and quality of software. For example, a CMM with a measurement uncertainty of $0.7 \mu\text{m} + L/600$, where L = length in mm in measurement volume, probing error of $1.5 \mu\text{m}$ and scanning error of $1.5 \mu\text{m}$ in 48 s, could cost as much as £140k.

However, although these point-wise methods can potentially produce high-precision measurements, their drawbacks include the requirement for contact with the surface under test and the localised measurement area [4]. Using a CMM could therefore prove to be quite challenging or basically unachievable in applications where the object cannot be easily measured by contact probing. An example could be making measurements for automated quality control on a production line, where products would need to be measured in very short time frames, typically in a few seconds.

Thus the main features of requiring no contact with the measured object, and ability to measure over large areas of the object surface, have led to much interest in developing whole-field (or full-field) optical metrology techniques [4].

1.5 Measurement principle of optical SMS

1.5.1 Phase measurement

Optical shape measurement systems provide a significantly faster alternative to CMMs since they require no contact with the sample and millions of coordinates can be measured in a few seconds. The benefits of using non-contact measurement systems include lower inspection costs, better quality control, faster production, smaller tolerances, fewer defects, and the ability to reverse engineer [21].

Phase measurement refers to the data acquisition process for the 3-D coordinates. Non-contact measurement of surface profile is usually dependent on techniques based on image cues, triangulation, various interferometric methods (including wavelength change, displacement of the test surface, and shifting the illumination beams) and projection of structured light patterns [4]. Passive profile sensors measure the test surface under natural illumination by examining image cues such as shading or texture, while active profile sensors typically require temporal control of the illumination, focus, or relative position of the test surface. The structured light technique benefits from easy implementation, ease of changing parameters by using a computer controlled liquid crystal display (LCD) or digital mirror device (DMD), and fast full field measurement, which have lead to commercialisation in the coordinate measuring and machine vision industries [3].

In an optical SMS, the determination of the three degrees of freedom which locate a point $P(x,y,z)$ on a test surface, involves contributions from a camera and projector. From a camera image plane, pixel coordinates (x,y) determine two degrees of freedom, and from a projector, structured light (fringe) patterns are projected onto the test surface, and on the basis of their intensity values, the third degree of freedom, surface height (z) is determined (see Figure 1-5). Each projection of a temporal sequence of intensities can be regarded as a code that uniquely identifies each section in the measurement volume. The two dimensional intensity pattern (or interferogram)

generated is phase modulated by the physical quantity being measured [4]. A sinusoidal interferogram may be represented by the following continuous intensity function [4]:

$$I(x, y) = I_0(x, y) + I_M(x, y) \cos \phi(x, y) \quad (1-1)$$

where $I_0(x, y)$ is the background illumination, $I_M(x, y)$ is the intensity modulation, $\phi(x, y)$ the phase term related to the physical quantity being measured, and (x, y) are the spatial coordinates in the reference frame of the image.

Interest is therefore in the change of the phase term from a reference (initial phase), which would represent a change in a physical quantity. As the initial phase is random, phase values and phase change values are usually wrapped back in the principal range $[-\phi, \phi]$. The relationship between a wrapped phase and unwrapped phase may be stated as [4]:

$$\phi(m, n, t) = \phi_w(m, n, t) + 2\pi v(m, n, t) \quad (1-2)$$

where $\phi(m, n, t)$ is the unwrapped phase, $\phi_w(m, n, t)$ is the wrapped phase, and $v(m, n, t)$ is an integer valued correcting field.

The *wrapped* phase map then has to be *unwrapped* using an appropriate phase unwrapping algorithm. The majority of algorithms can be classed as either a temporal phase unwrapping or spatial phase unwrapping algorithm according to whether the unwrapping procedure is performed along the time axis or along one or more of the spatial axes, respectively [4].

The current optical SMS at Loughborough has been developed using an optimised real-time temporal phase unwrapping algorithm leading to a demonstrated measurement precision, under good conditions, of 1 part in 20,000 of the measurement volume dimension [4, 9, 10]. Sequences of phase-shifted fringe patterns, of varying spatial frequency, are projected onto the object (typically 32, 31, 30, 28, 24, and 16 fringes across the field of view), leading to 250,000 coordinates being generated in less than a second.

1.5.2 Calibration

Calibration is the process of providing a mapping from phase map to real world coordinates. It could be considered as the process of determining the external and internal parameters of camera and projector. These parameters include translational parameters, rotational parameters, lens centre, and lens distortion model parameters. By identifying the position and orientation of the sensor components in 3-D space, the coordinates of an object can be measured relative to the identified coordinate system. Calibration establishes the coordinate system in which coordinates would be measured relative to, and therefore, it is expected that the process needs to be as accurate as possible. However, calibration could be a rather long and pains-taking process involving a significant amount of user input. Therefore, the calibration method of an optical SMS significantly affects the ease of instrument set up and consequently, set up time.

1.5.3 Data processing and presentation

When the measured point cloud data has been obtained using the optical SMS, it becomes necessary to efficiently process this data so that it can be presented in a meaningful way. For instance for complete 360 degree coverage of an object, different views of the object (with data relative to different coordinate systems) have to be patched together to form a single model by merging the data from these different views into a single coordinate system. The accuracy of a measurement system is also determined by the matching accuracy [3]. The combined shape data can for example, be used to compare with an idealised model of the object (e.g. CAD model) to quantify the difference in shape. In addition, to be able to visualise the measured point cloud, the data set would need to be exported to visualisation or CAD software. In order to achieve this, the measured point cloud would need to be presented in a file format that is acceptable for importing into the visualisation or CAD software. This presents a challenge in that a significantly large amount of data would need to be efficiently manipulated to reduce processing time.

1.5.4 Sensor design

In a situation where a large object is to be measured, a number of cameras and projectors may have to be used. Options include either a stationary single-camera single-projector system, or a multiple-camera multiple-projector system, or a single-camera multiple-projector system, or a multiple-camera single-projector system. The

planning of adequate inspection and measurement strategies which regulate the manufacturing process, is the key to maintaining product quality [22]. However, the specific details of each application would dictate which of these possible alternatives should be adopted in any given situation. On deciding on the number of cameras and projectors, another important issue that needs to be considered is their optimal positions within the measurement volume in order to achieve maximum coverage of the object, without missing out essential details during the measurement process. The process of identifying the most appropriate position of the camera(s) and projector(s) in a given measurement task is referred to as sensor design, which aims to reduce instrument set up and measurement time, and enhance accuracy. There are therefore a number of techniques for sensor design or planning, which could be classified into four categories [3]:

1. Generate and test: where sensor configurations are initially generated and then evaluated using performance functions and mission constraints (e.g. object detection, recognition and manipulation, and scene reconstruction)
2. Synthesis approach: where analytical relations between mission constraints and sensor parameters are built.
3. Sensor simulation system: where the objects, sensors and light sources are unified into a virtual environment and then used in a generate-and-test approach to find sensor configurations.
4. Expert system approach: where a rule-based expert system to determine sensor configurations.

Further details on these techniques can be found in the reference [3].

1.6 Thesis outline

In Chapter 2, major sensor calibration methods in the close range photogrammetry, computer vision, and optical engineering (fringe projection) research communities are discussed. The different calibration method classifications are also highlighted. A mathematical model for sensors is described in terms of quantitative parameters that define the sensor imaging geometry. Calibration methods suitable for multi-sensor arrangements are also described. Issues relating to calibration artefact design, in terms of geometric and material considerations are discussed, including examples of calibration artefacts.

The novel calibration process developed in this work is described in Chapter 3. Discussions are focussed on the calibration parameters (which include sensor (i.e. camera and projector) and calibration artefact parameters), mathematical model and the procedure for computing a point cloud from each measured data set. Also, a novel method for bundle adjustment (a method for non-linear optimisation of calibration parameters) is introduced. The calibration process is described in terms of two phases; the initialisation, where the sensor parameters are initialised, and the refinement phases, where the sensor and calibration artefact parameters are refined in a bundle adjustment.

Several calibration artefacts to calibrate the SMS were designed as the project progressed. The first set of these, referred to as 'sphere artefacts', are described in Chapter 4. A novel 3-D Hough transform developed for detecting spheres within a measured point cloud is introduced. Discussions here are focussed on establishing parameterisation for sphere detection, development of an optimised sparse matrix model for the Hough accumulator space, accurate peak detection in Hough space, and other post-processing related issues. The performance of the method on simulated and real data is also discussed. Part of the contents of this chapter have formed the basis for a journal paper recently accepted for publication in *Optical Engineering* [23].

Chapter 5 describes the development of a second set of calibration artefacts, referred to as 'plane artefacts' and discusses the extension of the 3-D Hough transform (described in Chapter 4, for sphere detection) for plane detection. The method's performance is reviewed on simulated and real data. The adaptation of the 3-D Hough transform to a 1-D Hough transform for efficient detection of a single plane or a set of multiple nominally parallel planes is discussed. Part of the contents of this chapter have formed the basis for a journal paper being prepared for submission, which is a follow up to reference [23].

The implementation of the novel calibration process developed in this work is described in Chapter 6. Discussion is focussed on how to obtain initial estimates for camera and projector parameters, feature detection and automatic control point selection, and how to estimate the pose of calibration artefacts within the SMS's

world coordinate system. Issues relating to the application of constraints to the calibration parameters in order to achieve numerical stability of the bundle adjustment are discussed. The structure of the software implementation of the calibration process is described. Also discussed in this chapter are experimental results of calibrating the SMS using the respective calibration artefacts developed in this work. Part of the contents of this chapter have been included in a recently accepted SPIE conference paper [24].

In Chapter 8, a method developed for efficiently tessellating a point cloud for exporting to standard CAD file formats is described. Examples of objects measured with the SMS and exported to visualisation software are also shown. 3-D data acquired by an optical SMS is useful for making a comparison with an ideal model of the measured object. Thus, methods for registering the measured point cloud from the SMS with a CAD model are also described.

A discussion on the use of the new calibration process to calibrate the optical SMS at Airbus's site is contained in Chapter 8. Some of the challenges of calibrating and using the SMS at an industrial site are also highlighted. Results of using the optical SMS in a structural test are shown. Here, comparison is made with other measurement methods such as digital image correlation. Part of the contents of this chapter have formed the basis for a recently accepted SPIE conference paper [24].

Finally, in the last chapter, an overview of the main novel aspects of the work is made, highlighting conclusions and recommendations for further work.

1.7 Figures

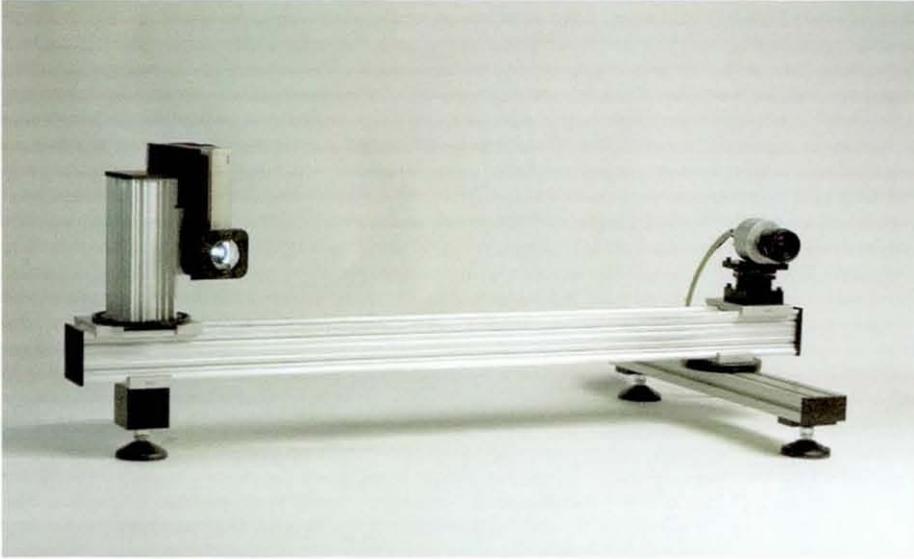


Figure 1-1: Single camera single projector SMS.

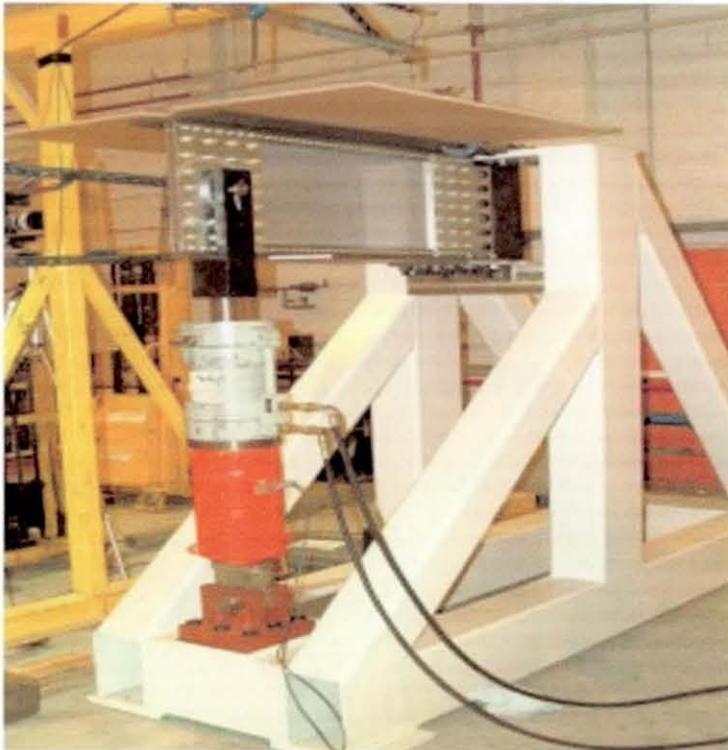
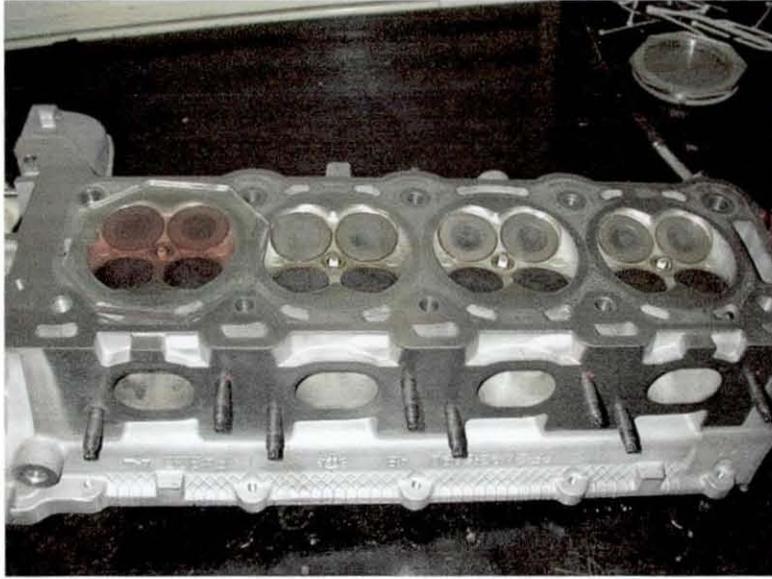


Figure 1-2: Aircraft Spar Testing at Airbus.

(a)



(b)

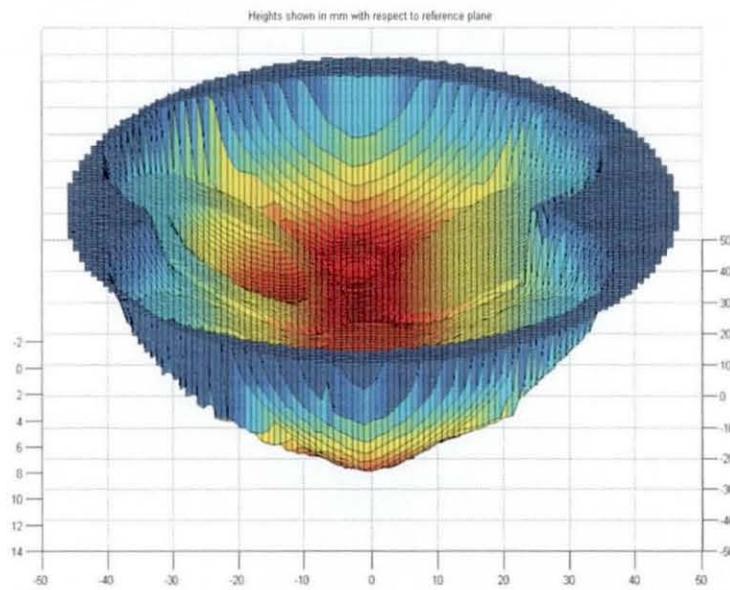


Figure 1-3: Cylinder Volume Measurement at Jaguar [12]. (a) Engine block. (b) Optically measured volume within agreement of 0.5% of conventional measurement method.



Figure 1-4: A mechanical CMM with a bridge structure.

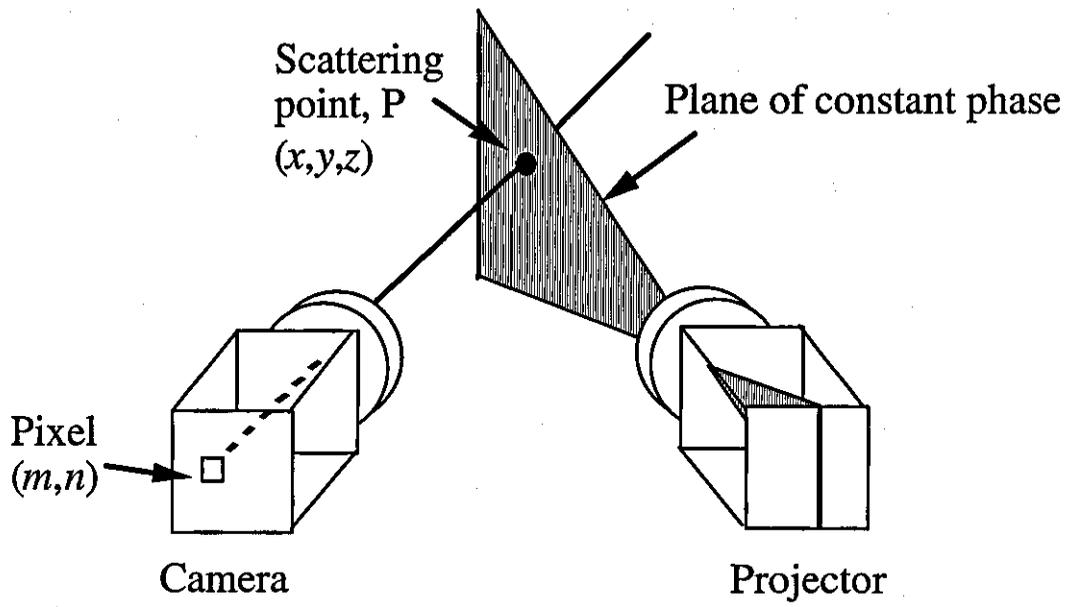


Figure 1-5: Measurement principle of optical SMS [5].

2 Sensor Calibration Methods

2.1 Introduction

In optical 3-D shape measurement systems (SMS), the light source and imaging sensor, are key factors for overall accuracy [3]. Important parameters for a light source include uniformity, weight, intensity profile and speckle or dot size, while for an imaging sensor (usually a charge coupled device (CCD) or charge injection device (CID)), they include speed, range, dynamic range and accuracy [3]. Depending on the optical method being used, some quantity encoding the shape data is subsequently extracted from the image. For example, in a structured light system, the measured values are the phase values of projected fringes, or of moiré patterns, and the pixel or image coordinates of a camera [15]. Calibration techniques have been investigated for optical 3-D measurement because most measurement methods usually acquire coordinates indirectly [15, 25]. The coordinate calculation from the measurement process includes the measured coordinates and the system parameters (geometric parameters) of the sensor, thus establishing a mapping from the quantity that encodes the shape information to 3-D coordinates [15]. Applications of sensor calibration include [26]:

1. dense reconstruction (mapping an image point to a 3-D point)
2. visual inspection (for example in quality control in manufacturing)
3. object localisation (for example in industrial part assembly and robot navigation)
4. sensor localisation (for example in robot control and path planning)

In this chapter, discussions are focussed on sensor calibration techniques and their classifications in the close range photogrammetry, computer vision, and optics (i.e. structured light method) fields. In addition calibration techniques for multiple-sensor measurement systems are discussed. The standard mathematical model used in the photogrammetry community for sensor imaging geometry is described. The different types of artefacts used in sensor calibration techniques are described, including discussions on calibration artefact design in terms of geometric and material considerations.

2.2 Sensor calibration techniques

2.2.1 Calibration in close range photogrammetry

The fundamental task of photogrammetry is to rigorously establish the geometric relationship between the image and the object as it existed at the time of the imaging event [27]. This relationship is expressed in the form of a mathematical model of the camera as a projective geometry (as shown in Figure 2-1), which describes an approximation of the internal geometry, position and orientation of the camera in the 3-D scene. The redundant value from four measurement values taken from at least two positions, can be used to determine object coordinates and model parameters [15]. The aim of the calibration process is therefore to efficiently and accurately estimate the parameters that describe the camera model. There are different types of calibration techniques, but their complexity can vary considerably, depending upon the complexity of the imaging system geometry [28]. Based on this variety, authors give different reasons for classifying calibration techniques.

Camera modelling could be broken down into four steps [26]:

1. relating a point in the world coordinate system to a point in camera coordinate system through a rotation matrix and translation vector.
2. applying a projective transformation to a point on the image plane
3. applying the deviation to the image point due to lens distortion.
4. transforming coordinates in the camera image plane from metric units (usually millimetres) to pixels (i.e. the computer image coordinate system).

The following mathematical model is based on that presented by Schreiber and Notni [15], which in turn follows the analysis in the classic photogrammetry text book of Kraus [29], where the lenses of the sensors are represented as pinholes (i.e. projection centre). From Figure 2-1 we can write,

$$\begin{bmatrix} \xi_M^{(c)} \\ \eta_M^{(c)} \end{bmatrix} = \frac{c}{z'} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} \xi_H \\ \eta_H \end{pmatrix} \quad (2-1)$$

where $\xi_M^{(c)}, \eta_M^{(c)}$ = image coordinates

x', y', z' = auxiliary coordinate system, with its origin in the projection centre and the $x'-y'$ plane parallel to the $\xi-\eta$ plane of the image coordinate system

c = camera constant, the distance between the projection centre and image plane along the optical axis, i.e. the perpendicular distance to the image plane

ξ_H, η_H = image coordinates of the principal point.

Deviations $d\xi$ and $d\eta$ from the origin of the image plane can be described with functions that contain additional parameters. For example, the correction for radial distortion can be written as

$$d\xi = (\xi_M^{(c)} - \xi_H)(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \quad (2-2)$$

$$d\eta = (\eta_M^{(c)} - \eta_H)(k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) \quad (2-3)$$

with $r = [(\xi_M^{(c)} - \xi_H)^2 + (\eta_M^{(c)} - \eta_H)^2]^{1/2}$

Eqn. (2-1) can then be written

$$\begin{bmatrix} \xi_M^{(c)} \\ \eta_M^{(c)} \end{bmatrix} = \frac{c}{z'} \begin{pmatrix} x' \\ y' \end{pmatrix} + \begin{pmatrix} \xi_H \\ \eta_H \end{pmatrix} + \begin{pmatrix} d\xi \\ d\eta \end{pmatrix} \quad (2-4)$$

The geometric relationship between the auxiliary (local) coordinate system and the world coordinate system is expressed in Eqn. (2-4) as

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \mathbf{R}(\omega, \phi, \kappa) \begin{bmatrix} x_M - x_O^{(c)} \\ y_M - y_O^{(c)} \\ z_M - z_O^{(c)} \end{bmatrix} \quad (2-5)$$

where x_M, y_M, z_M = world coordinates of the object

$x_O^{(c)}, y_O^{(c)}, z_O^{(c)}$ = world coordinates of the projection centre

$\mathbf{R}(\omega, \phi, \kappa)$ = orthonormal rotation matrix, which rotates the world coordinate system parallel to the auxiliary system

$$\mathbf{R} = \begin{pmatrix} \cos \phi \cos \kappa & \cos \omega \sin \kappa + \sin \omega \sin \phi \cos \kappa & \sin \omega \sin \kappa - \cos \omega \sin \phi \cos \kappa \\ -\cos \phi \sin \kappa & \cos \omega \cos \kappa - \sin \omega \sin \phi \sin \kappa & \sin \omega \cos \kappa + \cos \omega \sin \phi \sin \kappa \\ \sin \phi & -\sin \omega \cos \phi & \cos \omega \cos \phi \end{pmatrix} \quad (2-6)$$

where ω, ϕ, κ are Euler angles (see Figure 2-2).

The local image plane coordinates for the measured point are

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \xi_M^{(c)} - \xi_H \\ \eta_M^{(c)} - \eta_H \end{pmatrix} \quad (2-7)$$

and since this point occurs at $z' = -c$, we have

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} \xi_M^{(c)} - \xi_H \\ \eta_M^{(c)} - \eta_H \\ -c \end{pmatrix} \quad (2-8)$$

The set of parameters $x_0^{(c)}, y_0^{(c)}, z_0^{(c)}, \omega, \phi$, and κ are usually referred to as the camera's *extrinsic* or *external* or *exterior orientation* parameters. ξ_H, η_H, c are usually referred to as the camera's *intrinsic* or *internal* or *interior orientation* parameters, while the coefficients of the polynomial in the LHS of Eqns. (2-2) and (2-3), k_1, k_2, k_3 , are referred to as *lens distortion* parameters. Using Eqns. (2-4), (2-5) and (2-8), we can create the collinearity equations that establish the mapping between image space and object space:

$$\begin{aligned} [\xi_M^{(c)} - \xi_H](k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) &= -c \frac{r_{11}(x_M - x_0^{(c)}) + r_{12}(y_M - y_0^{(c)}) + r_{13}(z_M - z_0^{(c)})}{r_{31}(x_M - x_0^{(c)}) + r_{32}(y_M - y_0^{(c)}) + r_{33}(z_M - z_0^{(c)})} \\ [\eta_M^{(c)} - \eta_H](k_1 r^2 + k_2 r^4 + k_3 r^6 + \dots) &= -c \frac{r_{21}(x_M - x_0^{(c)}) + r_{22}(y_M - y_0^{(c)}) + r_{23}(z_M - z_0^{(c)})}{r_{31}(x_M - x_0^{(c)}) + r_{32}(y_M - y_0^{(c)}) + r_{33}(z_M - z_0^{(c)})} \end{aligned} \quad (2-9)$$

where r_{ij} are the elements of the rotation matrix, \mathbf{R} , given in Eqn. (2-6).

An image can be considered to be a bundle of rays, with each ray originating from the 3-D world and passing through the centre of the camera lens and terminating on the camera image plane. In close range photogrammetry, known 3-D coordinates (called control points) are measured as part of the calibration process and the bundle of rays is adjusted in a minimisation process called *bundle adjustment*. In photogrammetry, the term 'calibration' is normally used to refer to the retrieval of only the internal and lens distortion parameters, providing a quantitative description of the imaging geometry of the camera and its lens. A 'calibrated' camera is thus one in which the

internal and lens distortion parameters are available. On the other hand, the process of retrieving the external parameters of a camera, its position and orientation in 3-D space, is called *resection*. Assuming that the camera's internal parameters are known, at least three non-collinear control points are necessary for the resection of a camera [27, 30].

Intersection on the other hand refers to the process of computing real world Cartesian coordinates from image coordinates using camera parameters. Resection is therefore only an intermediate stage, often followed by intersection or by bundle adjustment [27, 30]. However, in the computer vision community, calibration refers to the process of retrieving any of the system parameters from images of a 3-D scene. Fraser [31] highlights some of the key differences in terms of the concept of calibration in the photogrammetry and computer vision communities, indicating that there are often practical distinctions between the way calibration parameters are applied.

Bundle adjustment is the problem of refining a visual reconstruction to produce jointly optimal 3-D structure (real world coordinates) and viewing parameter (external and/or internal) estimates [32]. This involves minimising some non-linear cost function that quantifies the model fitting error. Other example error models include robust least squares, intensity based methods (intensity based matching of image patches), and implicit models [32]. These cost functions can be put into two main categories: those based on minimising an algebraic error and those based on minimising a geometric or statistical image distance [33]. Algebraic error functions are typically based on the minimisation of a set of linear equations created from 3-D to 2-D point correspondences, while geometric cost functions are typically based on the geometric distance between measured and projected points in image space. Algebraic functions are usually used as a starting point for a non-linear optimisation of a geometric or statistical cost function. More details on the various issues to consider with regard to bundle adjustment can be found in Triggs et al [32].

The solutions proffered for increasing the probability of accurately retrieving calibration parameters tend to complicate the model and necessitate proper evaluation of results. However, proper design of the acquisition geometry to allow recovery of the calibration parameters is an art; the results must be carefully evaluated to ensure

that correlations between external and internal parameters or *weak* imaging geometries have not compromised the accuracy of the results [27]. Weak imaging geometry refers to situations where effects of small changes in the internal parameters of the camera cannot be distinguished from the effects of small changes in the external parameters. Weak geometric configurations are directly functions of the baseline to depth ratio, and the effect is more pronounced when this is less than 0.3 [34]. In addition, a photogrammetric network with weak geometric strength can make the calibration process more time consuming [35]. Issues relating to the effect of geometric strength on sensor parameters in multiple sensor networks will be discussed in Section 2.3.

The idea of *additional parameters* in bundle adjustment came about as a result of the attempt to determine to high accuracy the internal geometry of the camera. This involves determining parameters that provide a correction for the deviation from collinearity of rays expressed in Eqn. (2-9). These parameters include radial distortion, decentring or tangential distortion, out-of-plane and in-plane image distortion [31, 36]. Eqn. (2-2) can thus be expressed as :

$$d\xi = d\xi_r + d\xi_d + d\xi_u + d\xi_f \quad (2-10)$$

where $d\xi_r$ = radial distortion, $d\xi_d$ = decentring or tangential distortion, $d\xi_u$ = image plane unflatness, and $d\xi_f$ = in-plane distortion.

At high magnifications (within stand-off distances of less than about 15 times the camera focal length), the variation of distortion becomes most pronounced [31]. However, Fraser [37] discourages over-parameterisation (the use of too many additional parameters) by demonstrating that it weakens the solution for object space coordinates. It is also suggested that in some medium accuracy applications, four internal calibration parameters, namely k_1 coefficient for radial distortion $d\xi_r$, offset of principal point ξ_H and η_H , and the principal distance c , may be sufficient [31]. Further details on modelling for additional parameters and lens calibration methods can be found in some key photogrammetry papers and books [29, 31, 35, 37, 38].

The term "*on-the-job calibration*" which refers to the use of images of an object that contains control points to carry out a bundle adjustment with additional parameters, is

considered as the most common method of close-range camera calibration, [35]. However, “*self-calibration*” involves a bundle adjustment with additional parameters, independent of object space, without the requirement for control points [35, 36, 38].

From the above, it is obvious that for a successful calibration using a photogrammetric approach, a number of factors would have to be considered. Some of the factors to consider for improving the quality of calibration are [31, 39, 40]:

1. Distribution of points in the image: the number of points in the measurement volume should be well-distributed. It is suggested that there should be at least twelve on each image, and at least twenty for the entire measurement.
2. Photogrammetric network configuration:
 - Use of highly convergent imaging configuration
 - Roll diversity: incorporation of camera roll angles by acquiring images in both horizontal and vertical orientations, with at least one image rolled approximately 90 degrees differently to other images.
3. Number of images to acquire: a minimum of six images should be acquired if the calibration object is essentially flat (i.e. two-dimensional), or four images if the object is three-dimensional. In addition, these images should be taken from at least three different locations in the measurement volume, providing a well distributed 3-D object point field.

In summary, the most important factors that affect the accuracy of a photogrammetric measurement include: the resolution and quality of the camera, size of the measured calibration object, the type of target or surface feature, number of images acquired, geometry of images relative to each other and the object, and the accuracy of control information [14, 27, 40]. Mikhail et al [27] suggest that the initial measurement layout should be simulated to ensure that the required level of accuracy will be reached. The covariances from the simulated solution give an indication of the results to be expected from the measurement, barring any uncorrected systematic errors or other problems.

2.2.2 Calibration in computer vision

The concepts of calibration in computer vision have their foundation in close-range photogrammetry, and the sensor model is similar to the central perspective projection

described in 2.2.1. With time, these techniques have evolved and more striking differences in concepts, parameterisation and procedure are now evident. A simple example is, unlike in close-range photogrammetry, it is common practice that all z coordinates in the camera coordinate system are considered to lie on the positive Z -axis, i.e. having a value of c , where c is the principal distance, thus defining a left hand coordinate system [33]. However, the world coordinate system is a right-handed coordinate system. This is unlike the case in photogrammetry, where a right-handed coordinate system is used for both camera and world coordinate systems. Hartley and Mundy [41] give more details on the relationship between photogrammetry and computer vision, mentioning areas of similarity and highlighting the key differences in terminology and goals.

In computer vision, the mapping from 2-D image points to 3-D coordinates is usually expressed as a linear mapping of homogeneous coordinates (i.e. a set of $n+1$ coordinates is used to represent an n -dimensional space) thus:

$$\mathbf{U} = \mathbf{VM} \tag{2-11}$$

where \mathbf{U} = image point coordinates expressed as $\mathbf{U} = [\mathbf{U}_1 \ \mathbf{U}_2 \ \mathbf{U}_3]^T$, where \mathbf{U}_i $=_{1...2}$ are row vectors with \mathbf{U}_3 = a row vector of ones.

\mathbf{V} = a 3×4 matrix, called the *camera matrix*

\mathbf{M} = object space points expressed as $\mathbf{M} = [\mathbf{M}_1 \ \mathbf{M}_2 \ \mathbf{M}_3 \ \mathbf{M}_4]$, where \mathbf{M}_i $=_{1...4}$ are column vectors

The camera matrix can be further expressed as :

$$\mathbf{V} = \mathbf{KE} \tag{2-12}$$

where \mathbf{K} = an upper triangular 3×3 matrix, called the *camera calibration matrix*, which contains the camera's internal parameters thus:

$$\mathbf{K} = \begin{bmatrix} f & 0 & \alpha\xi_H \\ 0 & f & \beta\eta_H \\ 0 & 0 & 1 \end{bmatrix}$$

where α is a scaling factor for the ξ axis, β = scaling factor for the η axis

and where $E = a 3 \times 4$ matrix, which contains the camera's external parameters.

The *aspect ratio*, $S = \alpha/\beta$, usually has a value equal to or close to 1.

Further details on the various models and terms utilised in computer vision can be found in reference [33].

2.2.3 Classifications of sensor calibration techniques

Discussions in this section highlight the different classifications of sensor calibration techniques in the literature, with regard to the photogrammetry and computer vision fields. Classifications typically refer to the use of a calibration artefact, the number of images required and the implementation used to estimate the calibration parameters. Based on these classifications, in some instances, useful comparisons of the different techniques are also made, in terms of performance and accuracy.

Marshall et al [28] have classified calibration techniques as:

- Reference artefact method or arbitrary calibration function, in which the results of measuring a physical reference artefact are compared with the calibration data obtained from measuring the artefact by independent, traceable, means.
- System geometry method or model-based method, in which a system equation obtained from a photogrammetric model is applied to correct the geometry of the imaging system.

A combination of the two methods described above can be used to improve the accuracy of calibration in a self-calibration approach, as described in 2.2.1, and also simplify the calibration procedure [3, 35].

Agrawal and Davis [42] classify camera calibration methods into two broad categories:

1. Methods which use a calibration object with a fixed 3-D geometry and methods with 'generic' 3-D geometry e.g. coplanar points.
2. Methods which do not use a calibration object, also referred to as self-calibration

Gonzalez et al [43] did a comparative analysis of eight calibration methods for static cameras using a pattern as a reference. The methods investigated were classified

based on the number of views required and the planarity of the calibration artefact thus:

1. Methods which use a single view of a non-planar pattern: these include those by Faugeras [44], Tsai [45] (classic and optimised version), Lineal, Ahmed et al [46], and finally, Hekkila and Silven [47].
2. Method which uses a single view of a planar pattern: Batista's method [48]
3. Method which uses multiple view of a planar pattern: Zhang's method [49]

They conclude that there exists a strong coupling between camera internal and external parameters, and suggest that multiple images of a calibration object should be used when an accurate calibration is paramount.

Zollner and Sablatnig [50] compared three calibration techniques – the direct linear transformation (DLT) method proposed by Abdel-Aziz and Karara [51], Tsai's method [45] and Zhang's method [49]. The methods were compared in terms of performance and accuracy for a single view and multiple views of a calibration artefact consisting of circular control points. In the single view case, in the presence of severe radial distortion, Tsai's method gave the best performance. In the multiple view case, although Zhang's method performed well in terms of convergence features (e.g. convergence rate), execution time was very slow in comparison with the DLT method (implemented in conjunction with a bundle adjustment).

Salvi et al [26] also did a comparative review and accuracy analysis of some of the most frequently used camera calibration techniques in computer vision. The methods investigated were those of Hall et al [52], Faugeras-Toscani [53, 54], Tsai [45] (classic and optimised version), and Weng et al [55]. They classified these techniques based on the implementation used to estimate the parameters of the camera model:

1. Non-linear optimisation techniques: this usually includes lens distortion and requires the minimisation of a function which describes distances between points in image space. They are usually iterative and require a good initial guess in order to converge.
2. Linear techniques: which use least-squares method to compute a mapping from 3-D points to 2D points. However, because they do not model lens distortion, they provide a rough accuracy for the system. Examples include those of Hall et al [52], Toscani-Faugeras [53, 54].

The results from the comparative study were conditioned to the structure of the 3-D points, and image processing tools were used in image segmentation and further points extraction. An interesting observation made is that including a large quantity of parameters into the model of the camera does not imply that the accuracy obtained is necessarily better. The conclusion is made that non-linear methods are more accurate than linear methods and that the modelling of radial distortion alone is sufficient when high accuracy is required. They also suggest that in applications where a low accuracy calibration is acceptable, the fast and simple method of Hall et al [52] is sufficient.

The problems with linear calibration techniques is that they are not optimal estimators, and they ignore lens distortion [56]. In most sensors, distortion corrections are proportional to the distance from the image principal point raised to some power, and the errors associated with poor estimates for the principal point location can be significant even for a camera system with moderate distortion. However, despite these shortcomings, they provide a good starting point for iterative methods (usually non-linear methods), enhancing the latter's ability to converge to a global minimum. Chen [57] suggests that in a case where the distortion parameters are negligible, trying to recover them would lead to over-parameterisation that could affect the calibration process.

Apart from the above mentioned methods (Hall et al [52] and Toscani-Faugeras [53, 54]), examples of other linear methods include methods proposed by Agrawal and Davis [42], and the DLT method [51]. The DLT approach, being the popularly used method in most of the literature reviewed by the author, will be discussed in more detail. The DLT method has its origin in photogrammetry and is typically used in both the photogrammetry and computer vision fields for initialising external and internal sensor parameters, which are subsequently refined by some other computationally intensive, iterative method.

In the absence of distortion, the DLT between a point (x_M, y_M, z_M) in object space and its corresponding image space coordinates $(\xi_M^{(c)}, \eta_M^{(c)})$ can be expressed by the linear fractional equations:

$$\xi_M^{(c)} = \frac{L_1 x_M + L_2 y_M + L_3 z_M + L_4}{L_9 x_M + L_{10} y_M + L_{11} z_M + 1} \quad \text{and} \quad \eta_M^{(c)} = \frac{L_5 x_M + L_6 y_M + L_7 z_M + L_8}{L_9 x_M + L_{10} y_M + L_{11} z_M + 1} \quad (2-13)$$

where $L_{i=1, \dots, 11}$ are referred to as the DLT parameters.

These equations are based on the collinearity condition that the object point, lens centre, and ideal image point all lie on a straight line. The above equation can be rearranged thus:

$$\begin{bmatrix} \xi_M^{(c)} \\ \eta_M^{(c)} \end{bmatrix} = \begin{bmatrix} x_M & y_M & z_M & 1 & 0 & 0 & 0 & 0 & -\xi_M^{(c)} x_M & -\xi_M^{(c)} y_M & -\xi_M^{(c)} z_M \\ 0 & 0 & 0 & 0 & x_M & y_M & z_M & 1 & -\eta_M^{(c)} x_M & -\eta_M^{(c)} y_M & -\eta_M^{(c)} z_M \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} \quad (2-14)$$

The above can be written as

$$\begin{bmatrix} \xi_M^{(c)} \\ \eta_M^{(c)} \end{bmatrix} = \mathbf{M}\mathbf{L} \quad (2-15)$$

where $\mathbf{M} = \begin{bmatrix} x_M & y_M & z_M & 1 & 0 & 0 & 0 & 0 & -\xi_M^{(c)} x_M & -\xi_M^{(c)} y_M & -\xi_M^{(c)} z_M \\ 0 & 0 & 0 & 0 & x_M & y_M & z_M & 1 & -\eta_M^{(c)} x_M & -\eta_M^{(c)} y_M & -\eta_M^{(c)} z_M \end{bmatrix}$ is a $2n \times 11$

matrix (i.e. two rows for each point correspondence) and \mathbf{L} is a 12×1 vector, as shown above in Eqn. (2-13), whose elements could be rearranged as a 3×4 matrix

$$\mathbf{A} = \begin{bmatrix} L_1 & L_2 & L_3 & L_4 \\ L_5 & L_6 & L_7 & L_8 \\ L_9 & L_{10} & L_{11} & 1 \end{bmatrix}, \quad \text{which is known as the camera matrix.}$$

\mathbf{L} in Eqn. (2-15) can be solved directly using Gaussian elimination which gives a solution in a least squares sense. Thus, the error over a set of point correspondences in image space $\xi_M^{(c)}, \eta_M^{(c)}$ and object space x_M, y_M, z_M , is given as

$$\min_L \Sigma \left(\begin{bmatrix} \xi_M^{(c)} \\ \eta_M^{(c)} \end{bmatrix} - \mathbf{ML} \right)^2$$

The camera matrix, A , can then be decomposed to retrieve all the external parameters and internal parameters (without lens distortion parameters). The internal parameters can be retrieved thus:

$$\xi_M^{(c)} = \frac{L_1 L_9 + L_2 L_{10} + L_3 L_{11}}{L_9^2 + L_{10}^2 + L_{11}^2} \quad (2-16)$$

$$\eta_M^{(c)} = \frac{L_5 L_9 + L_6 L_{10} + L_7 L_{11}}{L_9^2 + L_{10}^2 + L_{11}^2} \quad (2-17)$$

$$c_x = \sqrt{\frac{L_1^2 + L_2^2 + L_3^2}{L_9^2 + L_{10}^2 + L_{11}^2} - (\xi_M^{(c)})^2} \quad (2-18)$$

$$c_y = \sqrt{\frac{L_5^2 + L_6^2 + L_7^2}{L_9^2 + L_{10}^2 + L_{11}^2} - (\eta_M^{(c)})^2} \quad (2-19)$$

$$c = \frac{c_x + c_y}{2} \quad (2-20)$$

Also, the sensor pinhole coordinates $(x_o^{(c)}, y_o^{(c)}, z_o^{(c)})$ with respect to the world coordinate system can be calculated thus:

$$\begin{bmatrix} x_o^{(c)} \\ y_o^{(c)} \\ z_o^{(c)} \end{bmatrix} = - \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix}^{-1} \begin{bmatrix} L_4 \\ L_8 \\ 1 \end{bmatrix} \quad (2-21)$$

If $A = L_9^2 + L_{10}^2 + L_{11}^2$, then the elements of the rotation matrix, \mathbf{R} , can be calculated thus:

$$R_{31} = \frac{L_9}{A} \quad R_{32} = \frac{L_{10}}{A} \quad R_{33} = \frac{L_{11}}{A} \quad (2-22)$$

$$R_{11} = \frac{\xi_H^{(c)} R_{31} - \frac{L_1}{A}}{C_x} \quad R_{12} = \frac{\xi_H^{(c)} R_{32} - \frac{L_2}{A}}{C_x} \quad R_{13} = \frac{\xi_H^{(c)} R_{33} - \frac{L_3}{A}}{C_x} \quad (2-23)$$

$$R_{21} = \frac{\eta_H^{(c)} R_{31} - \frac{L_5}{A}}{C_y} \quad R_{22} = \frac{\eta_H^{(c)} R_{32} - \frac{L_6}{A}}{C_y} \quad R_{23} = \frac{\eta_H^{(c)} R_{33} - \frac{L_7}{A}}{C_y} \quad (2-24)$$

The Euler angles ω, ϕ, κ can be extracted from \mathbf{R} thus:

$$\phi = \sin^{-1} R_{13} \quad \omega = \cos^{-1} \left(\frac{R_{33}}{\cos \phi} \right) \quad \kappa = \cos^{-1} \left(\frac{R_{11}}{\cos \phi} \right) \quad (2-25)$$

However, one needs to be careful in retrieving Euler angles from a rotation matrix, as a result of the singularities that could occur if there is 0 or 180 degree rotation about the y axis (referred to as *gimbal lock*). Other methods, such as Ganapathy's method [58], could also be used in retrieving the parameters.

As mentioned earlier, the DLT method uses known coordinates x_M, y_M, z_M , with their corresponding image points $\xi_M^{(c)}, \eta_M^{(c)}$, to determine camera parameters. Therefore, a minimum of 6 non-coplanar points is required in order to get a solution. Using the DLT parameters, $L_1 - L_{11}$, 3-D coordinates of image points can be calculated with at least 2 images from the following,

$$\begin{pmatrix} \xi - \xi_H \\ \eta - \eta_H \\ -c \end{pmatrix} = S[\mathbf{R}(\omega, \phi, \kappa)] \begin{bmatrix} x_M - x_O^{(c)} \\ y_M - y_O^{(c)} \\ z_M - z_O^{(c)} \end{bmatrix} \quad (2-26)$$

where S is the aspect ratio mentioned in Eqn. (2-12), and typically set to 1.

Agrawal and Davis [42] developed an algorithm that uses three or more images of a sphere to simultaneously calibrate the internal and external camera parameters by using a semi-definite programming approach (i.e. optimisation of symmetric positive

semi-definite matrix variables using a linear cost function and linear constraints). Four internal parameters are retrieved using a multi-step approach – the aspect ratio is first estimated, followed by the principal point, and finally, the focal length.

The most significant sources of error in a calibration include: insufficiency in modelling lens distortion, changes in illumination conditions between camera exposures, camera electronic noise, and uncertainty in the measurements of the 3-D coordinates of the control points [56]. Chen [57] concludes that k_1 is still the most significant distortion parameter and the precision of mapping deteriorate when more unknowns are involved. Also, there exists a strong correlation between many of the parameters, making it very difficult to solve for the full set of distortion correction terms.

2.2.4 Calibration methods for structured light measurement systems

The previous calibration procedure for the optical SMS developed at Loughborough and described by Saldner and Huntley [7], required the accurate translation of a reference glass flat on a translation stage. A polynomial was then fitted to the set of data at each pixel for each position of the reference flat using least-squares minimization. The glass flat was translated through the measurement volume in the z-direction and the phase map recorded at approximately 20 sample positions along the axis. This procedure was rather time consuming (up to an hour) and only provides an accuracy in the region of 1 part in 1,000 of the measurement volume dimension [4]. Furthermore, it requires a glass flat of dimensions exceeding the field of view of the camera, and is not easily extendable to multi-camera systems.

Hu et al [25] proposed a two-step approach; the first step involves measuring the parameters to determine their approximate values, while the second step involves measuring a calibration plate at different positions and then iteratively estimating more accurate parameters. Without measuring for these initial estimates, the calculation may become exhaustive and eventually converge to wrong values. The initialisation procedure requires accurate translation of a flat plate using a motorised linear stage, with measurements taken at different positions. The system parameters are then estimated by comparing the results with those obtained by a mechanical

CMM. A measurement error of 0.23 mm (standard deviation) was achieved, however, the system field of view was not mentioned. Non-linearity error and image distortion of projector and camera were not taken into consideration in this technique.

Guo et al [59] proposed a calibration method for a fringe projection system that involved translating a target plane to a sequence of given positions with known depths and using a least-squares estimation algorithm with linear computation to retrieve system parameters. The key issues tackled in their method include developing a mapping relationship between phase difference and depth map, the measurement procedure of phase maps during calibration, and the process of retrieving system parameters. Sitnik et al [60] developed a digital fringe projection system for large-volume 360-degree shape measurement. The calibration also involves translating a known object using a translation stage. In addition, their method requires that the size of the calibration object should be slightly larger than the measurement volume.

Legarda-Saenz et al [61] developed a method for calibrating a single-camera single-projector structured light system, in which a unique coordinate system is defined for both devices, thus introducing a rigidity constraint into the transformation process. The camera model used was similar to that used in photogrammetry, combining pinhole coordinates, Euler angles and lens distortions. In Zhang and Huang's method [62], by treating the projector as a camera, they were able to adapt a stereo vision calibration method to a single-camera single-projector structured light system. A red or blue checkerboard was used as a calibration object, with the camera and projector being simultaneously and independently calibrated. In this method, multiple measurements of the calibration object are made in order to retrieve sensor internal parameters (excluding lens distortion parameters). Subsequently, only one calibration measurement is required to obtain the sensor's external parameters. The calibration was evaluated by making multiple measurements of a planar white board in a $342 \times 376 \times 658 \text{ mm}^3$ volume, with measurements giving a maximum rms error of 0.22 mm.

Examples of other calibration techniques for a fringe projection system include polynomial coefficient estimation using least-squares fitting to the measurement data,

and artificial neural networks [59, 60, 63-66]. The main drawback of the polynomial estimation method is the lack of physical significance of the polynomial coefficients, and the fact that there is no guarantee of achieving a high accuracy even if a high degree polynomial is used. This is due to the possible occurrence of the Runge phenomenon (i.e. large oscillations that occur during polynomial interpolation). On the other hand, artificial neural network implementations involve a time-consuming training procedure.

Self-calibration using bundle adjustment, as described in section 2.2.1 could also be used for calibrating a fringe projection system [15, 61]. Schreiber and Notni [15] were able to successfully combine a fringe projection method with a photogrammetry approach to achieve a self-calibrating optical system. By rotating the grating by 90 degrees, for each camera pixel, two phase values are obtained, therefore four coordinates per camera pixel are retrieved. It is worth noting that the mathematical model for a stereo camera system is equivalent to a basic fringe projection system i.e. a single-camera single-projector system as shown in Figure 2-3). The aim of a self-calibrating optical system is to achieve a more efficient measurement procedure. By this combination, drawbacks inherent in one method have been overcome by the characteristic advantage in the other. For instance, 360 degree measurement using a fringe projection system is often an interactive, time consuming process, typically involving either moving the measurement system or object to multiple positions as a result of problems with shadowing and specular light, while photogrammetry offers a comfortable way to do this [15]. However, while photogrammetry involves time-consuming calculations which yield few measured points (e.g. 80 points per second achieved by Niini [67]), the fringe projection method is much faster and generates a much larger number of measured points. Therefore, by combining these two methods, a high number of object points can be acquired rapidly. Thus, the calibration techniques discussed earlier, which have been used for many years in the photogrammetry and computer vision communities, are being investigated to see how some of these ideas could be adopted in fringe projection systems.

The claimed benefits of self-calibration when used in a fringe projection system include [15]:

- insensitivity to environmental changes and vibrations due to simultaneous determination of 3-D coordinates and system parameters
- no calibration equipment is required
- data points would lie within a single coordinate system, meaning that for whole body (360 degree) measurements, subsequent matching/registration of single views would not be necessary

Discussions with one of the authors of this paper (G. Notni) [15], however, revealed that the method is not in fact completely self-calibrating: a ball bar or similar artefact needs to be incorporated at some stage of the calibration in order to provide an absolute length scale. Such an approach is also suggested in photogrammetry [29].

2.3 Multiple sensor calibration methods

The use of multiple sensors introduces observational redundancy and more bundles of imaging rays, which leads to improved triangulation precision and calibration accuracy [36]. Indeed, the study of multiple sensor calibration is mostly concerned with the extension of self-calibration of single sensor systems to multiple sensor systems. Typical studies have focussed on how the geometric strength of a multiple sensor network and calibration procedure affect the accuracy and precision of parameters estimated from a bundle adjustment. Clarke et al. [14] mention some of the advantages and disadvantages of using a multiple sensor measurement system. The advantages include redundancy of measurement, statistical based measurement (i.e. an rms error can be associated with each measured coordinate and consequently, a global standard deviation for the whole measurement), and flexibility of measurement system. However, on the other hand, the disadvantages are the requirement of expert knowledge and the complexity of the measurement process.

Fraser et al. [36] described the self-calibration of a multiple-sensor system made up of a network of six different CCD cameras, making observations on how different sensor parameters and calibration errors (both in image and object space) are affected by observational redundancy and different lenses (28 mm, 20 mm and 15 mm focal length). The multiple sensors were calibrated for measuring a $5 \times 2.5 \text{ m}^2$ object, and accuracies in excess of 1:100,000 (as a fraction of the largest dimension) and precisions in excess of 1:200,000 were achieved. The method of Agrawal and Davis [42], involves simultaneously calibrating multiple sensors using three images of

spheres, where system parameters are optimally recovered for each camera using semi-definite programming. The method is a non-iterative solution which can be used to obtain a good initial estimate for iterative methods. However, the method's accuracy is heavily dependent on the quality of ellipse detection and boundary fitting in image space. In addition, the method could perform poorly if the spheres are imaged near the image centres. Zhang et al [68, 69] have also used three images of spheres for calibrating multi-sensor systems, however, rather than a non-linear approach used by Agrawal and Davis [42], they used a linear approach which weakens the effect of ellipse detection on calibration accuracy. They also conclude that poor performance of the method is possible if the spheres are imaged near the image centres and that the method could be used to obtain a good initial estimate for iterative methods.

Pedersini et al. [70] calibrated a multiple sensor measurement system by positioning of a portable calibration object in various unknown positions in order to fill a measurement volume. This approach ensures that all parameters estimated during self-calibration are consistent across the measurement volume. By using image coordinates of natural image scene features during the image acquisition process, parameter drift of the acquisition system can be detected and tracked, and sensor parameters are corrected accordingly. Clarke et al. [14] describe a calibration procedure which involves the use of circular retro-reflective targets to be attached to the object to be measured. Through the use of image processing techniques, images of the target are identified and labelled, and then used in a bundle adjustment, thus achieving object space precision of 1 part in approximately 30,000. It is concluded that the accuracy of this method is dependent on the number of observations, type of target or surface feature, geometric strength of the sensor network, and operator experience.

Schreiber and Notni [15] describe the arrangement of multiple sensors for a fringe projection system. Unlike in the case of multiple sensor setups for photogrammetric systems, no markers are required on the object surface and no matching of the respective single views is required. In addition, a minimum of two projectors and one camera are required in the set up. Though the camera is not used in the coordinate calculation, it determines the measuring raster on the object surface, thus ensuring that

only phase values are required. Self-calibration is therefore achieved using just the projector parameters and object space coordinates in a bundle adjustment, leading to an uncertainty of $\sigma \approx D \times 10^{-5}$, where D = lateral extent of measurement field.

2.4 Calibration artefact design

2.4.1 Geometric considerations

In this section, discussions will focus on the calibration objects used in calibration methods described in Sections 2.2 and 2.3. The first decision with regard to the design of a calibration artefact has to do with its shape and size. One key conflict to be resolved in a practical design is those between a geometry that facilitates a measurement by the imaging system and one that is representative of real-world objects and can be measured by conventional measuring instruments [28]. In addition, one of the main sources of inaccuracy in calibration methods is the accuracy with which the 3-D coordinates of the calibration artefact are known and characteristics of the artefact [56, 71-73]. For system parameters estimated from a calibration to give accurate measurements reliably, the calibration artefact needs to be positioned at different locations within the 3-D measurement volume [62, 71]. Therefore, to ensure ease of detection and measurement, calibration objects need to have visually and geometrically significant features. Such features which include lines, centre of gravity of circles, and corners of squares, are treated as control points as part of the calibration process. The choice of feature depends on [74]:

1. size of calibration object,
2. focal length of camera,
3. stand-off distance between object and sensor,
4. light or illumination conditions.

These basic features can be identified in shape data of a 2-D planar object, virtual 3-D object (i.e., a composite of multiple poses of a 2-D planar object) or a true 3-D object having significant changes in surface shape in all 3 Cartesian dimensions. 2-D calibration objects suffer from the disadvantage that a complete set of sensor parameters cannot be retrieved without *a priori* knowledge, usually of internal parameters like focal length and displacement of principal point. In addition, specialist equipment such as a translation or rotation stages may need to be used where some accurate manipulation of the artefact is required within the measurement volume. Multiple views of a 2-D calibration object can be used to simulate a virtual 3-

D object. However, this would require accurate translation in one direction, usually along the Z-axis, without giving room for rotation. In addition, it may also require the use of specialist equipment e.g. a robotic manipulator or specialist holder [74]. 3-D calibration artefacts on the other hand, allow for a more accurate calibration, by providing more knowledge of the 3-D scene, which allows the robust retrieval of sensor parameters. However, they can be quite difficult and expensive to manufacture and the process of measurement can be quite complicated as a result of shadowing. Examples of 3-D calibration objects include regular shapes such as cylinders, cubes, spheres, etc. A common type of artefact is a planar surface or multiple planar surfaces having either a given colour, a checkerboard pattern, circles or a combination of these. For example, Figure 2-4 shows the calibration artefact used by Heikkila [56] which consists of two perpendicular planes with 256 circles on each plane.

Liu et al. [75] used a 3-D object as a calibration artefact, a three-step plane with circular retro-reflective targets. Hu et al. [25] used an aluminium plate sprayed with white paint to calibrate a fringe projection system. The 3-D coordinates of holes within the plate were measured by a coordinate measuring machine (CMM). Zhang and Huang [62] used a flat checkerboard to calibrate a structured light system. For calibrating the camera, a flat black-and-white coloured checkerboard was used, while a flat red-and-blue coloured checker board was used for calibrating the projector. Chen and Liao [76] used a calibration artefact made up of two objects for calibrating a fringe projection system. Firstly, there is a calibrating block made with laser lithography, having accurately positioned patterns. Secondly, a silicon substrate with a smooth surface coating synthesized by chemical vapour deposition (CVD). The calibration object is translated to preset locations along the Z-axis to form a non-planar measurement space. Sitnik et al. [60] used a calibration object made up of circles to calibrate a fringe projection system, by positioning on a translation stage and translating along the Z-axis in order to calibrate for the measurement volume. From a calibration matrix, the coefficients of a fifth order polynomial are determined and used for calculating real world Cartesian coordinates. Therefore, the method requires translation to a minimum of 5 positions.

Schonfeld et al [77] used an aluminium block incorporating three tilted planes, translating it several times in increments along the Y-axis to cover the measuring

volume of the sensor. Kuhmstedt et al [78] used a 'staircase' shaped reference artefact for the determination of the 3-D orientation of the rotational axis of a structured light rotary scanner. The three top surface planes are used to determine points of intersection for each of several rotational positions which eventually describe the rotational axis.

Cubes have been used in some systems as a calibration artefact. Valkenburg and McIvor [79] used a 150 mm cube with 72 circular fiducial marks (of 5 mm radius) arranged on three faces for the calibration of a structured light sensor. The 3-D location of each fiducial mark has been independently determined to an accuracy of 0.1 mm. System performance is evaluated by measuring the cube in several different positions. However, it was suggested that the performance of the system could be improved by using a better calibration artefact because more fiducial marks with better distribution would be measured more accurately.

Light-emitting diodes (LEDs) have also been incorporated with other objects as a calibration artefact, as calibration markers. For the calibration of a stereophotogrammetry-based sensor, Deacon et al [80] used a corner-shaped artefact consisting of three flat plates positioned at right angles, each containing 36 holes with LEDs inserted into them. The hole positions have been independently calibrated using a CMM. In use, the LEDs can be switched on or off to give subsets which are used to determine the orientation of each plane. A more traditional design of calibration artefact for a stereophotogrammetry system is given by van den Heuvel [81].

Machacek et al. [72] describe three different types of objects used for a two-step calibration of a stereo camera system for measurements in large volumes ($2 \times 2 \times 1 \text{ m}^3$). There are two planar calibration objects: (i) a large aluminium plate ($1.55 \times 1.1 \text{ m}^2$) with 280 calibration markers with a diameter of 10 mm, whose size is consistent with the measurement volume, and used as a reference for the two-step calibration process; and (ii) a smaller black anodised aluminium plate ($0.4 \times 0.3 \text{ m}^2$) with 680 calibration markers of 1 mm diameter, used to calibrate the internal parameters of the cameras. For a correct calibration, these plates have to be translated precisely within the measurement volume, and the translation direction must be perpendicular to the

plates. Finally, the third calibration object consists of a bar, with two pinhole LEDs which are at a known relative distance from each other (500.2 mm), used as calibration markers. This type of rigid bar calibration object was first used by Borghese and Cerveri [73] to calibrate a multiple-camera system.

Godhwani et al. [82], used reference spheres for calibration of a projector in a multiple-sensor structured light system. In this method, the cameras are calibrated using a cube with a set of rings, before the projector calibration is performed. Circular profiles are projected onto the sphere and the radius and centre coordinates of the sphere are determined using a least squares fit. The difference between the measured and actual sphere parameters (radius and centre coordinates) is used as the error in observation in a Kalman estimate of the projector parameters.

Penna [83] calibrated the scale factor of a single camera using a precisely fabricated sphere, such as a ball bearing, based on the observation that the scale factor of the camera is related to the distortion in an image of a circle. Other researches have been able to expand on the idea of using spheres as a calibration object, by developing methods for calibrating for more camera parameters. Xu et al. [84] used multiple images of three balls to calibrate the internal and external parameters of a single camera. Agrawal and Davies [42], and Zhang et al. [68, 69] also used three images of spheres to calibrate the internal and external parameters of multiple cameras.

The use of certain 3-D objects as calibration artefacts for multiple sensor systems may not be feasible as the objects may not be simultaneously visible in all sensors [42, 68, 69]. For example, a cube in a particular orientation would give two different 3-D views when observed from two different positions. This would mean that the 3-D data or surface contour acquired from the object in a particular orientation would differ from one camera to the other. A planar object could also suffer from this problem if multiple sensors are to be simultaneously calibrated for 360 degree measurement. A sphere has the unique property that from whichever position it is viewed, it reveals a curved surface. Hence, shape data of a sphere from two different camera views would give slightly different 3-D views of the contour of a sphere, with which one can estimate the sphere's centre using 3-D feature detection techniques. Therefore, spheres have been used for calibrating multiple sensor systems with a common field

of view[42, 68, 69]. The sphere artefacts developed in this work are described in Chapter 4.

A combination of two nominally parallel planes (in the form of two plates, with each plate made to high accuracy with respect to flatness and parallelism) separated by a fixed, known distance could also be quite useful. This constraint provides a length scale which can be introduced into a calibration process and potentially simplify the detection of the orientation of the artefact, because it gives rise to the unique property that any single point on one of the planes has the same perpendicular distance to any point on the other plane's surface. Details of the issues relating to the design of a plane artefact based on this concept, and developed in this work, are discussed in Chapter 5.

Multiple 3-D objects of different geometry have also been combined together as an artefact. However, they have usually been used for testing the performance of optical systems and not as calibration artefacts. Beraldin et al [85] used a set of five test objects, each possessing geometric features designed to test a particular aspect of an optical system's performance. For surface measurements, four objects having known surface parameters such as cylinders, spheres, and tilted planes, have been used. The fifth calibration object is a flat plate with circular holes of known sizes for the determination of edge measurement performance. Marshall et al [28] have used a combination of seven cylinders, six tetrahedra, and two planes in a single artefact (see Figure 2-5) for assessing the performance of commercial 3-D whole body imaging systems. The cylinders are used for assessing measurement integrity off-axis and at extreme regions of the measurement volume. The planes are used for assessing planarity and separation repeatability.

2.4.2 Material considerations

For a calibration artefact that could be used in different operating environments, it is necessary that it should be thermally stable and have low coefficient of thermal expansion i.e. its temperature change should not be significantly large over a short timescale. Coupled with this, it should have high strength, ensuring consistency in its dimensions over time. Although high strength and thermal stability are desirable, it is also important that the artefact should have low density, as this would ensure low

weight and allow for easier handling by a user. With regard to an optical system, it would be undesirable for a calibration artefact to be transparent or have a reflecting surface finish, so as to ensure that valid data can be obtained from its surface. This can be achieved through coating with a suitable matt finish, depending on the method for shape measurement. However, further surface treatment could lead to the introduction of small changes in the dimensions of the artefact. Therefore, the key characteristics of the material from which the calibration artefact should be made are:

1. low thermal conductivity
2. low coefficient of thermal expansion
3. high strength
4. low density
5. non-specular, opaque surface finish

2.5 Summary

Calibration is a key factor affecting the measurement accuracy of an optical SMS. To establish a mapping from image space to object space, models consisting of parameters that describe in quantitative terms the sensor imaging geometry have been developed. A variety of calibration techniques have therefore been developed within the photogrammetry and computer vision communities, which can be applied to the calibration of an optical SMS based on the fringe projection method. However, it is important to ensure consistency in terms of modelling, parameterisation and coordinate orientation to avoid recovering erroneous calibration parameters. A variety of classifications have been suggested for sensor calibration techniques. These classifications are based on the requirement for a reference artefact with known control points, the cost function used in the calibration model, the number of views required for the calibration process and the requirement for non-linear optimisation of calibration parameters.

The type of calibration technique to be used is dependent on the number of sensors in the measurement system and their arrangement, the level of accuracy to which 3-D coordinates should be measured, and other requirements specific to the measurement, such as the measurement environment, speed of calibration, and calibration artefacts available. Most of these methods require the use of calibration objects having features, such as circle centres or square vertices (in the form of a chequer board),

whose coordinates have been measured using an independent measuring instrument, such as a CMM. Typically, the choice of feature is dependent on the calibration object's size, focal length of camera, distance between object and scene, and illumination conditions. In addition, it is expected that calibration objects should be thermally stable, rigid, light weight for easy handling and its surface should be non-specular.

As part of the calibration process, images of the calibration object are acquired at different positions in the measurement volume. In some methods this positioning has to be done accurately, while in other methods, this may not be necessary. Appropriate feature detection algorithms are then used to detect specific features on images of the artefact, which are considered as control points. The control points are then used in either an iterative or non-iterative scheme to obtain the calibration model parameters. However, calibration methods referred to as self-calibration methods, do not require information on the 3-D object space (i.e. control points), and therefore, do not need a calibration object.

2.6 Figures

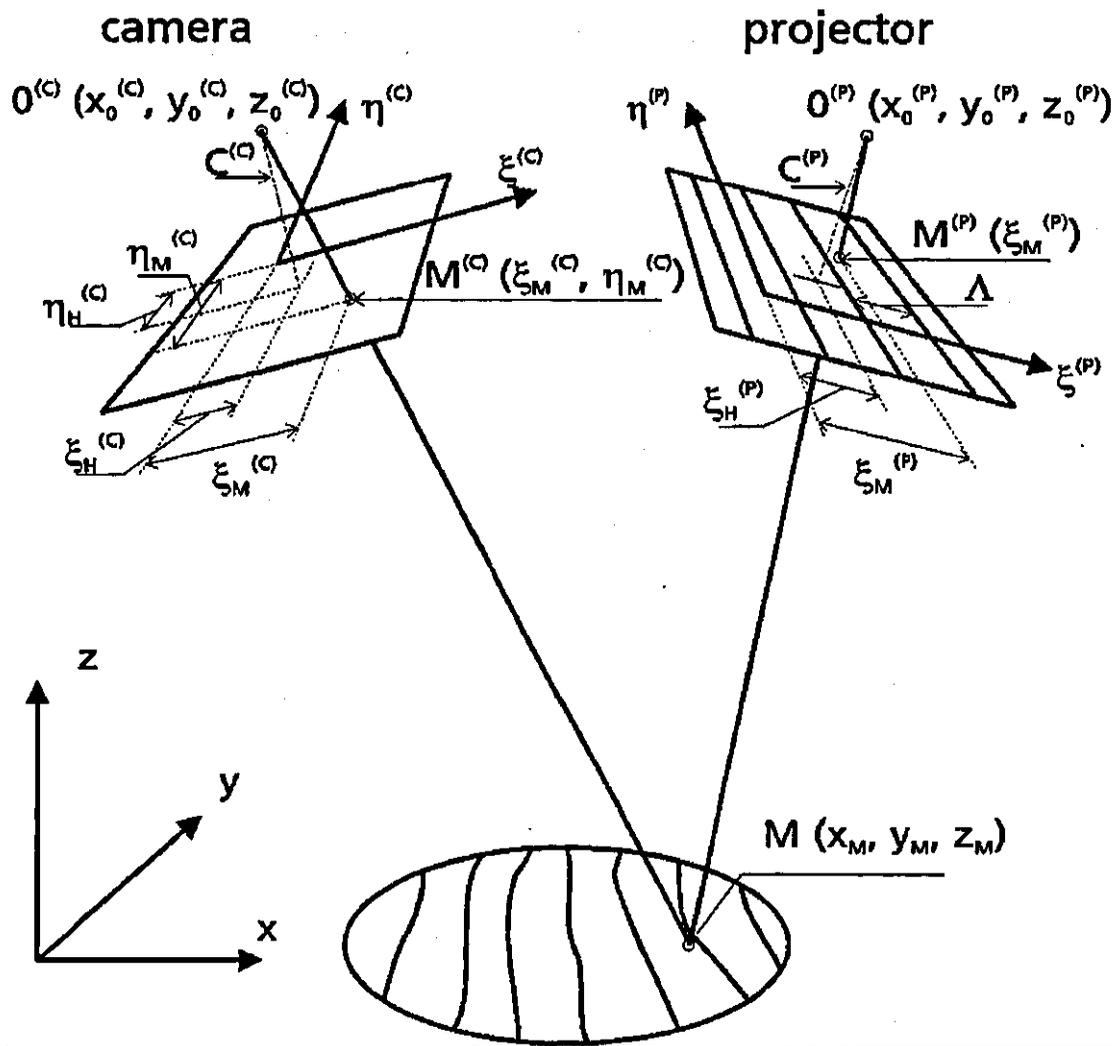


Figure 2-1: Relationship between measurement values and 3-D coordinates in a fringe projection system [15].

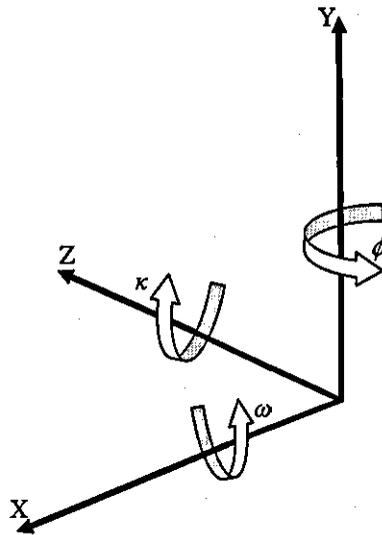


Figure 2-2: X, Y, Z axes and their corresponding Euler angles, ω , ϕ and κ .

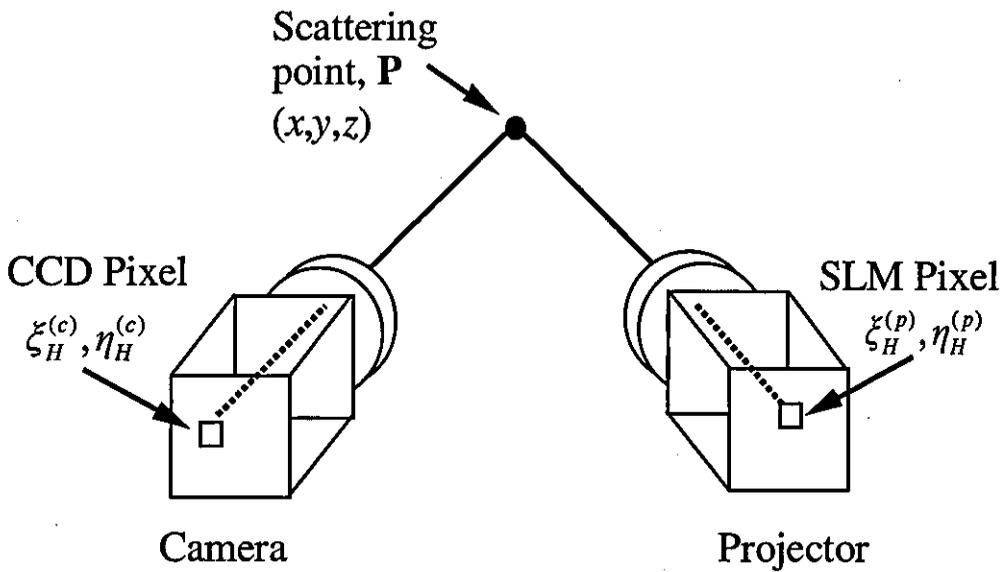


Figure 2-3: Relationship between camera and projector in a fringe projection system

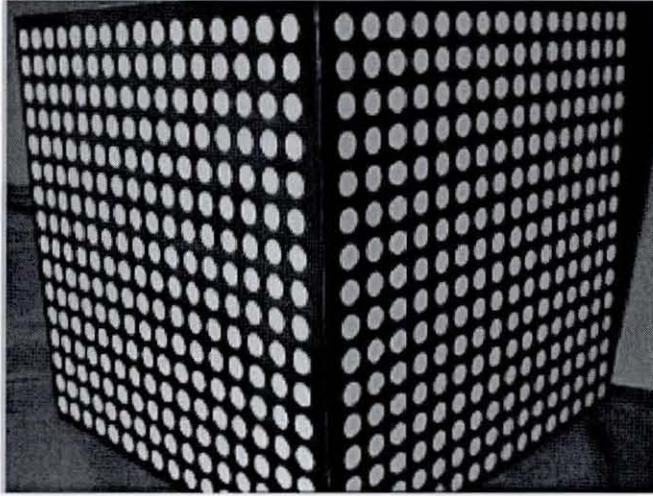


Figure 2-4: The Calibration used by Heikkila [47] consisting of two perpendicular planes with 256 circles on each plane.

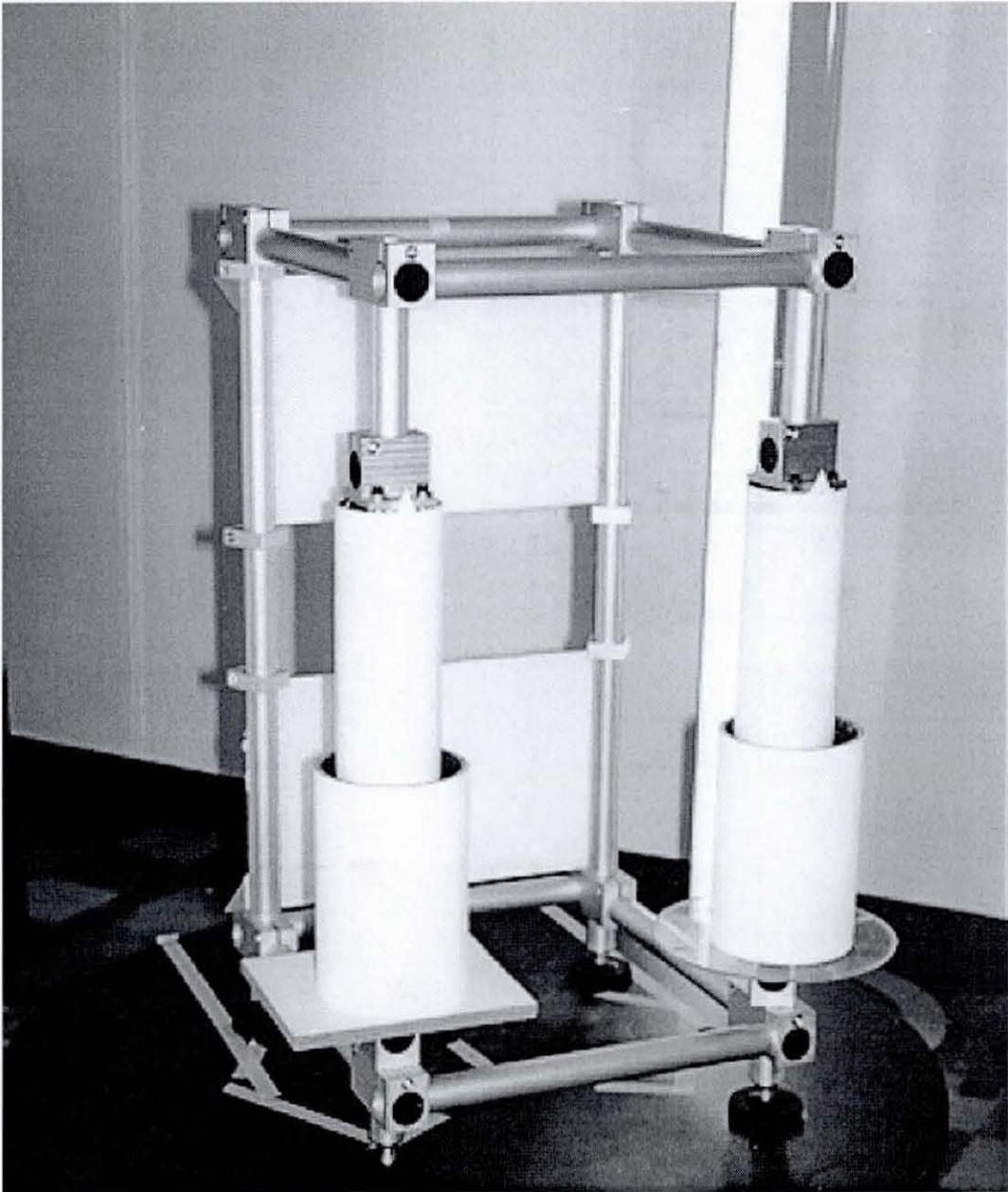


Figure 2-5: Reference artefact framework [28].

Section II –
Development of automated
calibration method

3 Description of Model for New Calibration Process

3.1 Introduction

In Chapter 2, the different sensor calibration methods for optical shape measurement systems (SMS) have been highlighted, including discussions on the relevance of the ease, reliability and accuracy of calibration methods for practical measurement systems. Calibration model parameters have been described in terms of sensor parameters (i.e. parameters of camera(s) and projector(s)) and other parameters which are defined by the calibration method's mathematical model. Bundle adjustment, one of the most commonly used methods for sensor calibration, involves the non-linear optimisation of these parameters that describe the calibration model. In many bundle adjustment methods, the use of a calibration artefact is required, where control points are extracted from the surface of the artefact and used in the optimisation of the model parameters.

A new calibration process has been developed to utilise a novel bundle adjustment model proposed by Huntley [86]. The new calibration process can be broken into two distinct phases: (1) initialisation and (2) refinement. In the initialisation phase, the sensor parameters are initialised using a linear calibration method such as the direct linear transformation (DLT) method (described in Chapter 2), while the refinement phase involves non-linear optimisation of the calibration model parameters in a bundle adjustment. In this chapter, the model for the novel bundle adjustment model developed by Huntley [86] is introduced. The calibration concept and the process of computing 3-D Cartesian coordinates from data from the optical SMS are described. The objective function to be minimised in the bundle adjustment method is discussed, including modifications incorporated to improve the rate of convergence. The chapter concludes with an overview of the new calibration process.

3.2 Calibration concept

3.2.1 Model for bundle adjustment

The model for bundle adjustment developed by Huntley [86] differs from conventional bundle adjustment, in which coordinates of points on the object surface are taken to be unknowns and are solved for as part of a large matrix inversion

process. For optical SMSs, one may have 10^6 or more unknown coordinates and even though the relevant matrices are sparse, dealing with such large numbers of unknowns becomes unwieldy. By contrast, this method involves projecting the rays through the sensor pinholes and minimising either the minimum distances between distances of closest approach, or the distances between the points of closest approach and known control point coordinates. The minimisation is therefore in object space and not in image space. Thus, the size of the matrices is fixed by the number of unknown calibration parameters, and not by the number of pixels or control points.

The sensor model is based on a photogrammetric approach (described in Chapter 2), made up of 12 parameters, namely:

- the external parameters: x_0, y_0, z_0 (pinhole coordinates), ω, ϕ, κ , (Euler angles describing the orientation of image coordinate system relative to the world coordinate system),
- the internal parameters: ξ_H, η_H, c (offset of principal point along image plane and focal length) and
- the lens distortion parameters: k_1, k_2, k_3 , which are coefficients for the polynomial describing radial distortion

In the model, three right-handed coordinate systems are introduced (see Figure 3-1), with sensor parameters providing the relationships between these coordinate systems:

1. the sensor coordinate system (SCS), with the sensor pinhole, say, O_e , as origin
2. the image coordinate system (ICS), defined on the image plane (which is assumed to be in front of the pinhole rather than behind) and parallel to the SCS, having the centre of the image as the origin
3. the world coordinate system (WCS)

The calibration process consists of holding the calibration artefact in multiple orientations and locations ('poses') within the measurement volume, whilst measurements are made using all possible combinations of cameras and projectors. Therefore, the bundle adjustment method requires not only estimates of the sensor parameters, but also estimates of each pose of the calibration artefact.

From each camera-projector pair we obtain the phase gradient information for horizontal and vertical fringe orientations, ω_x and ω_y , (in the range $-\pi$ to $+\pi$), which encode the shape information of the artefact in each pose. Further details on the shape data acquisition process are discussed in Chapter 6. It should be noted that the pixel dimensions of each phase gradient map is the same as the pixel resolution of the camera, with each valid pixel corresponding to a 3-D point on the object surface. Therefore, for a 1024×1024 pixel camera, each phase gradient map would be a 1024×1024 matrix, and consequently, its corresponding point cloud computed from a single projector fringe sequence could be expressed as three 1024×1024 matrices for the respective x, y, z coordinates. Estimating each pose of the calibration artefact from the point cloud can be achieved through identifying specific features on the artefact (e.g. spheres) using an appropriate feature detection technique. Thus, the 3-D Cartesian coordinates of the identified feature is used to calculate the transformation from the local coordinate system to the SMS' coordinate system.

3.2.2 Computing a point cloud

Consider first a single-camera single-projector SMS, for which the two sensor lenses can be represented by position vectors \mathbf{R}_1 and \mathbf{R}_2 as shown in Figure 3-2. Assuming the availability of initial estimates for the sensor parameters, in order to compute a point cloud, we project onto the object space the rays coming from the pixel coordinate (ξ_1, η_1) (camera) and (ξ_2, η_2) (projector). These rays can be represented by the vectors \mathbf{u}_1 and \mathbf{u}_2 , respectively, which start at the sensor pinholes. We calculate the scattering point to be at position vector \mathbf{s} , where \mathbf{s} lies at the midpoint of the shortest line joining the vectors \mathbf{u}_1 and \mathbf{u}_2 . The length of this shortest line is denoted ϵ_1 . Thus, given $\mathbf{R}_1, \mathbf{R}_2, \mathbf{u}_1$ and \mathbf{u}_2 , it is possible to evaluate the required position vector \mathbf{s} . This mapping from phase gradient maps to 3-D Cartesian coordinates can be summarised as some function:

$$x, y, z = f(\xi, \eta, \omega_x, \omega_y, C_1 \dots C_{12}, P_1 \dots P_{12}) \quad (3-1)$$

where x, y, z = world coordinates

ξ = image plane coordinates of camera along X axis

η = image plane coordinates of camera along Y axis

ω_x = image plane coordinates of projector (phase values) along X axis

ω_y = image plane coordinates of projector (phase values) along Y axis

$C_1 \dots C_{12}$ = twelve parameters of camera (i.e. photogrammetric model of external, internal and distortion parameters)

$P_1 \dots P_{12}$ = twelve parameters of projector (i.e. photogrammetric model of external, internal and distortion parameters)

To use Eqn. (3-1), the non-dimensional image plane coordinates of camera, $U = 1, 2, 3, \dots, M$ and $V = 1, 2, 3, \dots, N$, are scaled to some length (in this case millimetres) based on knowledge of the charged coupled device (CCD) physical dimensions, with the centre of the image plane set as the origin. Following from the convention used in image processing literature, it should be noted that the origin of the camera's $M \times N$ pixel coordinate system is at the top left corner of the image. For the projector, image plane coordinates, ω_x and ω_y , are scaled from radians to millimetres using the spatial light modulator (SLM) physical dimensions. For the camera and projector respectively, the image plane coordinates can thus be calculated in the appropriate units of length in the following manner

$$\begin{aligned} \xi_c &= \left(\frac{U-1}{M-1} - 0.5 \right) N_{x,c} \\ \eta_c &= \left(\frac{V-1}{N-1} - 0.5 \right) N_{y,c} \end{aligned} \tag{3-2}$$

$$\begin{aligned} \xi_p &= \left(\frac{\omega_x}{2\pi} \right) N_{x,p} \\ \eta_p &= \left(\frac{\omega_y}{2\pi} \right) N_{y,p} \end{aligned} \tag{3-3}$$

where M, N = number of pixels along ξ and η axes of camera

$N_{x,c}, N_{y,c}$ = length of camera CCD physical dimensions along ξ and η axes

$N_{x,p}, N_{y,p}$ = length of projector SLM physical dimensions along ξ and η axes

3.2.3 Description of the 'pose' of artefact

In the calibration of multi-sensor SMSs, where shape data of the calibration artefact is acquired using different camera-projector pairings, the respective computed point clouds give different 3-D views of the artefact. Once the artefact's features in these

respective point clouds are detected, the scattering points (and their corresponding pixel coordinates in the phase maps) belonging to each feature need to be labelled consistently for all point clouds of all poses of the artefact. With this approach, each feature parameter (for example, the coordinates of a sphere centre) is uniquely matched across poses and across camera-projector pairs. We would for example know which pixels in the phase maps of a given camera-projector pair refer to say, sphere 1, for all poses of the artefact.

The estimate of each pose is the transformation that will need to be applied to go from a local coordinate system defining the artefact (established by a mechanical CMM), to the world coordinate system (WCS) that characterises the measurement volume. This can be described in terms of position and orientation by six parameters, t_x , t_y , and t_z (components of a translation vector along X, Y, Z axes) and Euler angles, ω , ϕ and κ (rotations about X, Y, Z axes) respectively. The transformation can be expressed as:

$$\mathbf{W}_s = \mathbf{R}_s \mathbf{W}_c + \mathbf{T}_s \quad (3-4)$$

where \mathbf{W}_s = coordinates in SMS coordinate system i.e. world coordinate system,

\mathbf{W}_c = coordinates in local coordinate system,

\mathbf{R}_s = rotation matrix computed from Euler angles ω , ϕ and κ thus,

$$\begin{pmatrix} \cos \phi \cos \kappa & \cos \omega \sin \kappa + \sin \omega \sin \phi \cos \kappa & \sin \omega \sin \kappa - \cos \omega \sin \phi \cos \kappa \\ -\cos \phi \sin \kappa & \cos \omega \cos \kappa - \sin \omega \sin \phi \sin \kappa & \sin \omega \cos \kappa + \cos \omega \sin \phi \sin \kappa \\ \sin \phi & -\sin \omega \cos \phi & \cos \omega \cos \phi \end{pmatrix},$$

$$\mathbf{T}_s = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \text{translation vector.}$$

As the respective components of \mathbf{R}_s and \mathbf{T}_s are refined during the bundle adjustment process, they are applied to \mathbf{W}_c in Eqn. (3-4) to compute \mathbf{W}_s . However, the configuration of the calibration artefact may require application of constraints to some of its rigid body transformation during bundle adjustment. This is achieved through defining one or more of the pose parameters to be constant during the adjustment. The constraints applied to the calibration artefacts developed in this work are discussed in Chapter 6.

3.3 Bundle adjustment in object space

3.3.1 Description of expression for minimisation

The calibration concept for the bundle adjustment has been described in Section 3.2. The novel bundle adjustment method involves the minimisation in object space of two quantities: (1) the minimum distances between distances of closest approach, ε_1 and (2) the distances between the points of closest approach and *known* control point coordinates, ε_2 . This can be expressed as an objective function, F , with two terms

$$F = \gamma_1 \sum_i \varepsilon_{1,i}^2 + \gamma_2 \sum_i \varepsilon_{2,i}^2 \quad (3-5)$$

over all the i pixels for the camera-projector pair that contain valid data, where γ_1 and γ_2 are scalars that allow for different weighting of the terms. The first term on the right hand side is the sum of squares of ε_1 errors between rays projected from the camera and projector pinholes (i.e. the sum of squares of errors of the *calculated* control point coordinate). On the other hand, the second term, is the sum of squares of ε_2 errors between the known and calculated control point coordinates. The calculation of ε_2 is therefore dependent on the geometric features of the calibration artefact. For example, for a sphere, ε_2 is simply the distance between the sphere centre coordinate estimated from a measured point cloud of the sphere (computed using sensor parameters) and the known centre coordinate (computed using the artefact's estimated pose, \mathbf{R}_s and \mathbf{T}_s , in Eqn. 3-4). Further details of how ε_2 is calculated for specific artefacts will be shown in Chapter 6.

The minimisation process therefore involves differentiating F with respect to the model parameters thus

$$\frac{dF}{dt} = 2\gamma_1 \sum_i \varepsilon_{1,i} \frac{d\varepsilon_{1,i}}{dt} + 2\gamma_2 \sum_i \varepsilon_{2,i} \frac{d\varepsilon_{2,i}}{dt} \quad (3-6)$$

where t is a generic parameter referring to the parameters which are 'free' to change. The number of parameters, N_p , can be calculated as:

$$N_p = N_{fs} + N_{fa} \quad (3-7)$$

where N_{fs} = number of free sensor parameters, and

N_{fa} = number of artefact poses \times number of free artefact parameters.

The bundle adjustment refines not only the camera and projector parameters but also the estimate of orientation parameters for each pose of the calibration artefact. The calibration artefact is thus free to undergo arbitrary rigid body translation and rotation during the bundle adjustment. However, as stated earlier the use of some types of calibration artefact may require the introduction of some constraints through 'fixing' some parameters (i.e. selecting the parameters that should not change during bundle adjustment). On the other hand, this necessitates pinning down the position and orientation of one of the cameras or projectors (e.g. camera 1) by fixing its external parameters. For example, in calibrating, say, an optical SMS consisting of a single camera-projector pair by measuring a calibration artefact in a single pose, the maximum value of N_p should be 24 (6 free parameters for the camera including distortion parameters, 12 for the projector, and 6 for the artefact pose), since the external parameters of the camera would need to be fixed.

3.3.2 Optimisation of parameters

The set of equations in Eqn. (3-5) is nonlinear and therefore cannot be solved directly [86]. The iterative scheme used involves solving for the vector \mathbf{d} in

$$\mathbf{H}\mathbf{d} = -\mathbf{g} \quad (3-8)$$

where \mathbf{g} is a column vector of the derivative of F (see Eqn. (3-5)) with respect to each N_p parameter, and \mathbf{H} is the Hessian matrix (matrix of second derivatives such that $H_{ij} = \partial^2 F / \partial x_i \partial x_j$). Thus, the solution vector \mathbf{x} is then updated as follows at the k th step:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k \quad (3-9)$$

where from Eqn. (3-8), $\mathbf{d}_k = -\mathbf{H}_k^{-1}\mathbf{g}_k$. Evaluation of the Hessian matrix would be cumbersome and we therefore use the Gauss-Newton method in which an approximation to the Hessian matrix is given by $\mathbf{J}_k^T \mathbf{J}_k$, where \mathbf{J} is the Jacobian:

$$\mathbf{H}_k \approx 2 \begin{pmatrix} \sum_i \left(\frac{\partial \varepsilon_i}{\partial x_1} \right)^2 & \sum_i \left(\frac{\partial \varepsilon_i}{\partial x_1} \right) \left(\frac{\partial \varepsilon_i}{\partial x_2} \right) & \cdot & \cdot \\ \sum_i \left(\frac{\partial \varepsilon_i}{\partial x_2} \right) \left(\frac{\partial \varepsilon_i}{\partial x_1} \right) & \sum_i \left(\frac{\partial \varepsilon_i}{\partial x_2} \right)^2 & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \sum_i \left(\frac{\partial \varepsilon_i}{\partial x_1} \right)^2 \end{pmatrix} \quad (3-10)$$

Eqns. (3-8) and (3-9) are iterated from an initial estimate for \mathbf{x} until convergence is achieved. However, the Gauss-Newton method has problems if \mathbf{H}_k is close to singular, which consequently affects its performance and rate of convergence [87]. Eqn. (3-8) is thus modified to the Levenberg-Marquardt method, with the current implementation utilising Fletcher's strategy for modifying the scalar, μ_k [153] in

$$(\mathbf{H}_k + \mu_k \mathbf{I}) \mathbf{d}_k = -\mathbf{g}_k \quad (3-11)$$

where $\mu_k \geq 0$ and \mathbf{I} is an identity matrix with the same matrix size as \mathbf{H}_k .

In addition, Eqn. (3-9) is also modified by introducing a scalar, α , thus:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k \quad (3-12)$$

where α is normally set at 1 at the beginning of the bundle adjustment. However, if at the k^{th} iteration, the current value of α increases F , a new value for α which minimizes F is computed. This updated α is used for subsequent iterations in Eqn. (3-12).

3.4 New calibration process

As stated earlier, the calibration process can be broken into two major phases, but these can be further broken down into four main sub-processes: (A) shape data acquisition, (B) sensor parameter initialisation, (C) shape data post-processing, and (D) bundle adjustment.

Shape data acquisition is common to both phases of the calibration process and refers to the measurement process for acquiring shape information of the calibration artefact. The shape information for each camera-projector pair consists of a pair of unwrapped phase maps, ω_x and ω_y . A sequence of fringe patterns is projected from each projector, while the camera(s) acquire images of the projected patterns. In the

initialisation phase (Figure 3-3), using each camera-projector pair, a single pose of an artefact is measured, the required features on the artefact are detected from the acquired shape data and used by the DLT method to obtain initial estimates of the respective camera and projector parameters. In this case, the artefact features are detected in image space.

On the other hand, the refinement phase of the calibration process (Figure 3-4), involves shape data acquisition, shape data post-processing and bundle adjustment. A second calibration artefact is measured in different poses in the measurement volume using all camera-projector pairs. Subsequently, for each camera-projector pair, measured data (unwrapped phase maps, ω_x and ω_y) is converted to a point cloud, where the required number of artefact features is detected, artefact pose estimated, and the required number of control points is selected from the detected features. Once the artefact features have been detected, an estimate of its pose is calculated using the known control point coordinates. On processing the measured data sets for all camera-projector pairs for all artefact poses, finally, the extracted control point information, pose and sensor parameters are used as initial estimates in a bundle adjustment (i.e. a non-linear optimisation). The bundle adjustment refines the parameters for all cameras, projectors and artefact poses, thus minimising the objective function describing the calibration model.

In general, once the parameters for all cameras and projectors have been initialised, the refinement phase is carried out in order to process all the data which will be used for obtaining the optimal sensor parameters. Manually processing such a large quantity of data would prove to be a very difficult task for a user of the optical SMS. Automatic 3-D feature detection is thus a crucial factor for enabling the automation of the refinement phase of the calibration. The calibration procedure is therefore implemented with the refinement phase expected to run automatically. This has involved developing robust and efficient 3-D feature detection, pose estimation, and control points selection methods. In addition, it has also led to the development of an appropriate data structure to handle the variety of parameters required at each stage of the calibration. Separate calibration artefacts have also been designed for the two calibration phases. For the initialisation phase, a 2-D planar artefact is used, while a

3-D artefact is used in the refinement phase. Chapters 4, 5, and 6 describe these artefacts in more detail, including the methods developed for automatically detecting the 3-D artefact's features in a point cloud and the implementation of the calibration process.

3.5 Summary

The new calibration process based on a novel bundle adjustment model has been described. The novel bundle adjustment model involves the minimisation of error metrics in object space, which include either the minimum distances between distances of closest approach, or the distances between the points of closest approach and known control point coordinates. The control point coordinates are obtained from a calibration artefact made up of a set of 3-D features with known position and orientation established in a local coordinate system by a mechanical CMM. The bundle adjustment model is characterised by sensor parameters and artefact pose parameters. The sensor parameters provide a mapping from the sensor coordinate system to the world coordinate system, while the pose parameters provide the rigid body transformation for the control point coordinates from a local coordinate system to the world coordinate system, i.e. to object space. The calibration process therefore aims to provide these parameters in the most efficient way to the bundle adjustment model for refinement in a non-linear optimisation. The calibration process thus involves all the sub-processes for initialising these parameters and then refining in a bundle adjustment.

The calibration process developed for multiple sensors has been described in terms of two phases. In the initialisation phase, a 2-D artefact is placed at a single position in the measurement volume, and measurements are made using all camera-projector pairings. The parameters of each sensor are initialised by applying a linear calibration method (e.g. the DLT method) on measured data of respective camera-projector pairs. In the refinement phase, a 3-D artefact is placed at multiple positions in the measurement volume and measurements are made using all camera-projector pairings. A point cloud is computed from the respective measured data sets and estimates of the pose of the calibration artefact are calculated. These pose estimates and sensor estimates are then passed on to the bundle adjustment model for refinement.

A consequence of using multiple poses of the calibration artefact in the refinement phase is that the user of the optical SMS would end up having to manually process a large quantity of data. The calibration procedure has therefore been implemented with a view to automating the refinement phase and subsequent chapters describe the methods developed to achieve this objective.

3.6 Figures

Sensor coordinate system, *SCS*

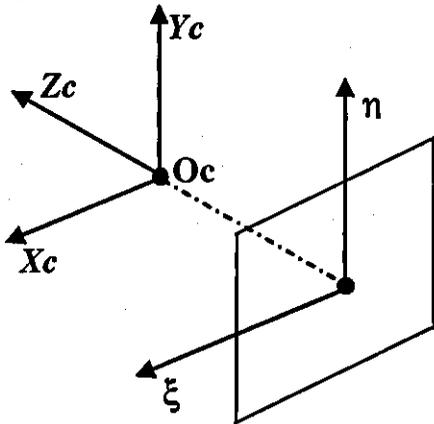
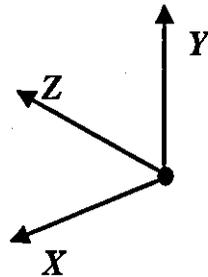


Image coordinate system, *ICS*



World coordinate system, *WCS*

Figure 3-1: Coordinate systems in the optical SMS., where O_c is the pinhole of the sensor .

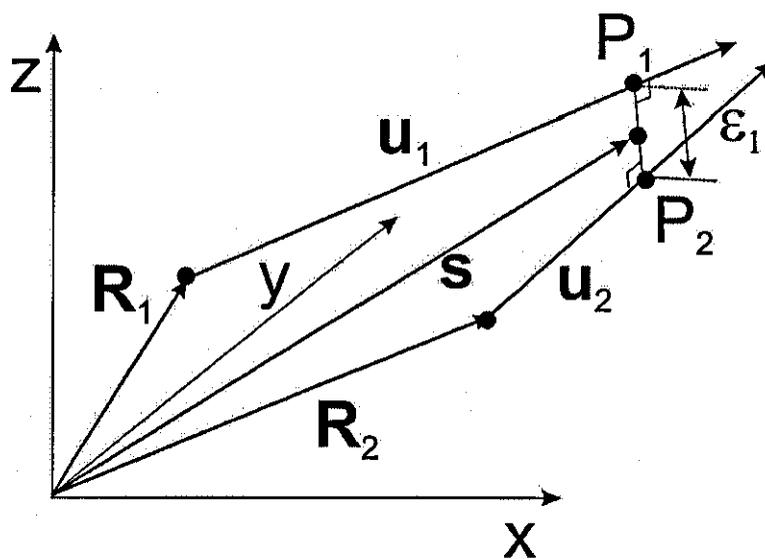


Figure 3-2: Vector picture in object space [86].

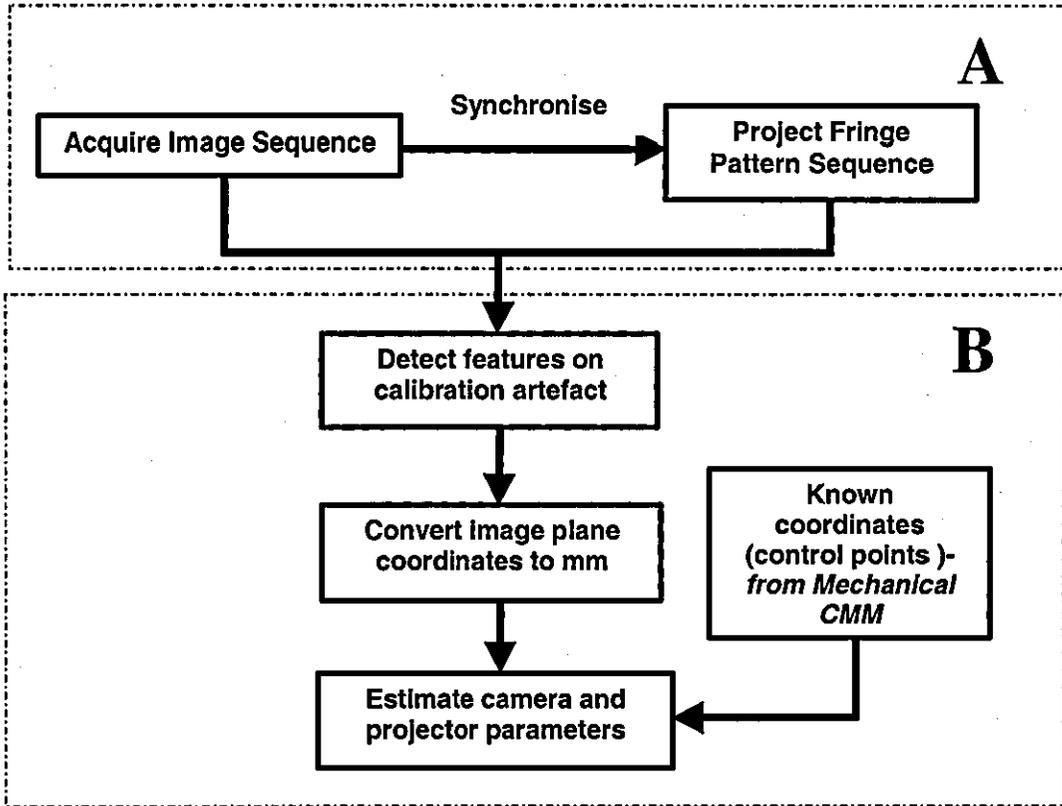


Figure 3-3: Initialisation phase of new calibration process – initialisation of camera and projector parameters. ‘A’ section is shape data acquisition, ‘B’ section is sensor parameter initialisation.

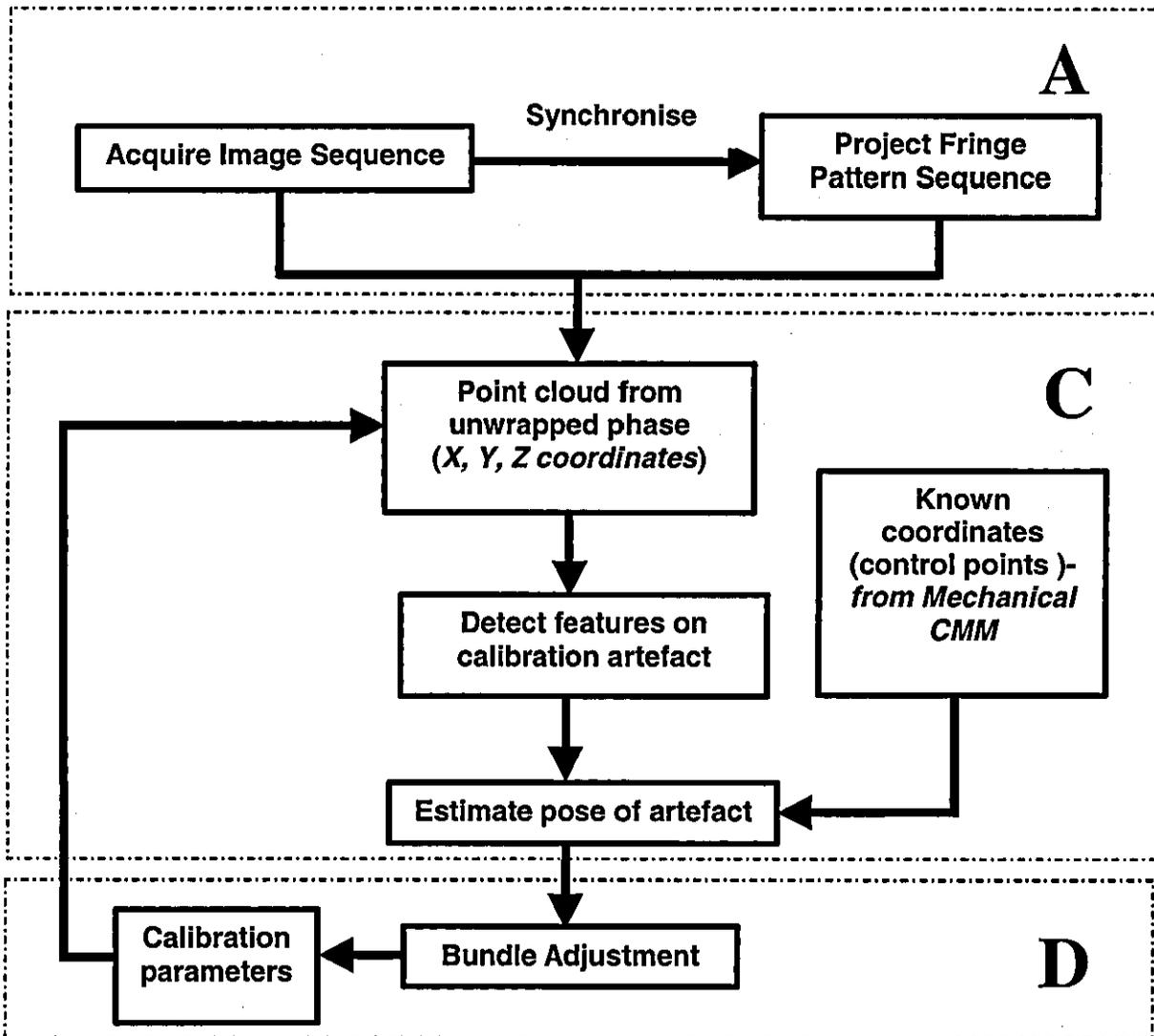


Figure 3-4: Refinement phase of calibration process – pose estimation and novel bundle adjustment. 'A' section is shape data acquisition, 'C' section is shape data post-processing and 'D' section is bundle adjustment.

4 Sphere Artefacts and Sphere Detection

4.1 Introduction

The importance of a calibration artefact in the calibration of an optical 3-D shape measurement system (SMS) has already been highlighted in chapter 2. Depending on the measurement system, calibration method, measurement volume and the required accuracy of the calibration, a variety of features of different sizes can be used to provide the control points which constitute calibration artefacts. However, the accuracy to which control points on the artefact have been measured and subsequently detected during calibration has a direct effect on the accuracy of calibration.

For multiple 3-D sensors, using parameters that define the sensor model, from each camera-projector pair we can obtain a distinct 'point cloud' of 3-D coordinates. The equations for calculating the coordinates of the points from the measured phase values are described in Chapter 3. In order to calibrate for the full measurement volume, measurements of the artefact are made at different locations within the volume. However, this leads to a large amount of data that has to be processed. Thus, the calibration process is in practice a non-trivial task normally requiring significant user input and processing time. Many optical SMS calibration methods rely on the ability to label regions within each point cloud as belonging to a known region on the surface of the artefact. Feature detection therefore plays a significant role in the ease, speed, and accuracy of the calibration of 3-D sensors.

The advantage of using spheres as calibration artefacts for multiple sensor systems has been highlighted in Chapter 2 [42, 68]. A sphere has the unique property that from whichever position it is viewed, it reveals a single curved surface with which one can estimate the location of the sphere's centre, and subsets of this single surface are in general visible to each camera-projector pair in a multiple sensor system. Therefore, calibration artefacts consisting of spheres were designed and manufactured. In this chapter, discussions will focus on the sphere artefact developed in this work, including the basis for its selection, and different types of arrangements and configurations. Also to be discussed is the method developed for automatically detecting spheres and identifying control points in point clouds. Part of the contents of this chapter have formed the basis for a journal paper recently accepted for

publication in Optical Engineering [23]. In addition, some parts of this chapter are included in a recently accepted SPIE conference paper [24].

4.2 Multiple-sphere artefacts

4.2.1 Arrangement of spheres

In the design of a multiple-sphere artefact, the main issues which were considered included the total number of spheres on the artefact and the arrangement of the spheres. A variety of possible arrangements were considered, but an important objective was to ensure that as many spheres as possible are visible from many viewing directions. This is a critical issue especially for multiple sensor SMSs. Also, it was necessary to ensure that there is a lack of symmetry, coordinate wise, across the artefact (i.e. no region is a mirror image of the other), to simplify the identification of the orientation of the artefact. The manufacturing process selected for making the artefact (which will be discussed in 4.2.2) also leads to a restriction in size of the base plate of the artefact to $270 \times 270 \text{ mm}^2$.

From the above considerations, three different calibration artefacts were designed and manufactured. All the artefacts consist of multiple spheres with a radius of 23 mm, supported above a $270 \times 270 \text{ mm}^2$ planar base. The first artefact consisted of 41 spheres stacked on three different base plates (Figures 4-1 (a) and (b)). On three sides of the artefact, spheres with unique stalk heights have been introduced to simplify the determination of its orientation. The second artefact as shown in Figures 4-2 (a) and (b), consisted of 33 spheres with nominally identical stalk heights of 5 mm, i.e. planar sphere centre coordinates. Finally, a third artefact (Figures 4-3 (a) and (b)) was made consisting of 33 spheres with stalk heights randomly distributed within a 5 mm to 45 mm range above the top surface of the base. In addition, half-sized versions of these three artefacts were also designed i.e. with spheres of radius 11.5 mm supported on a $135 \times 135 \text{ mm}^2$ planar base, for use in smaller fields of view. Therefore, a total of 6 artefacts were designed and manufactured.

4.2.2 Manufacturing Process

The different number of sphere artefacts designed could prove expensive to manufacture, especially to high precision. It was therefore decided that rapid manufacturing would provide a cost effective means of testing the viability of each

arrangement. Following the required material conditions for calibration artefacts set out in Chapter 2, with better material properties and stability of parts, a powder based process, selective laser sintering (SLS) was selected [88].

The SLS process involves sintering or melting a powdered raw material as a laser selectively scans the surface of a powder bed to create a two-dimensional solid shape [88]. A thin layer of powder is then added to the top of the bed so that a layer of powder can be traced by the laser, bonding it to the layer below. So, through this process, a full three-dimensional shape is created layer by layer. This implies that after manufacture, a solid object is fully embedded within a mass of powder. Figure 4-4 shows one of the artefacts after removing from the SLS machine. During the sintering process, heat is built up within the part, therefore on completion the part is allowed to gradually cool in the machine to avoid distortion. However, the size of each of the artefacts relative to the working volume of the machine could potentially lead to disproportionate heating and cooling during the manufacturing process, thus causing significant distortions in the part. Therefore, to ensure minimal material usage in rapid manufacturing, the design of the artefacts was altered by hollowing out the under side of the base plate to create a uniform distribution of holes (see Figure 4-5).

4.3 Ball bar

A ball bar simply consists of two spheres separated by a known, fixed distance. By using just two spheres there are some practical benefits over the multiple-sphere artefacts. Firstly, the time to detect two spheres is much shorter than for say 33 spheres. In addition, the matching of control points to the SMS's coordinate system is more robust. Due to the reduced number of spheres, the total cost of making the ball bar to high accuracy will be much lower than for the sphere artefacts. Finally, the ability to build the ball bar artefact using off-the-shelf precision components is possible, making traceability easily achievable and the ability to reliably increase or decrease the separation between the spheres thus providing easy extension to different scales of fields of view. On the other hand, the main drawback of the ball bar, compared to an artefact with multiple spheres, is the reduced coverage of the measurement volume in any given point cloud. Therefore, provision should be made for positioning at different locations and in different orientations within the volume, using say, a camera tripod and camera tripod arm.

Ball bars are popular calibration objects for mechanical coordinate measuring machines (CMMs). The typical radius of spheres for commercially available ball bars are usually less than or equal to 25.4 mm. Although some suppliers are able to manufacture spheres of larger radius, such an order is considered as a custom made part, thus making the cost prohibitive. However, there are precision ball manufacturers who manufacture spheres to high accuracy and to wide range of sizes. In addition, some manufacturers make ball related products such as CMM reference balls and tooling balls, which are spheres joined to a stem usually having a threaded end. Typical materials for manufacturing precision spheres include tungsten carbide, steel, and ceramics (e.g. zirconia, alumina, silicon nitride).

It was proposed that the ball bar artefact should consist of two CMM reference balls and a length bar. CMM reference balls are standard metrology products used for evaluating CMMs, while on the other hand, length bars are commonly used as a length standard for maintaining traceability in dimensional metrology. Length bars are commercially available in a variety of lengths, and can be purchased individually or as a set. Length bars are typically made of hardened high-quality steel and available in four grades of accuracy, reference, calibration, grade 1 and grade 2 [89]. In addition, they are usually manufactured with either flat faces (reference and calibration grades) or threaded holes (grades 1 and 2) at both ends, therefore various combinations can be joined together to form a variety of standard lengths. Therefore, the following components were purchased: (1) two calibrated tungsten carbide CMM reference balls with diameter 50.8 mm ($\pm 2.5 \mu\text{m}$ accuracy) on a stem of 100 mm and (2) a set of 10 grade 1 length bars (with threaded holes) of lengths ranging from 12.7 mm to 762 mm. For example, Figure 4-6 shows the two CMM reference balls attached to both ends of a 101.6 mm length bar to form a ball bar calibration artefact. Typically, CMM reference balls are made to a reflective surface finish which would lead to measurement problems for an optical sensor. Therefore, the author engaged in further discussions with the manufacturer (Spheric Trafalgar Ltd), who advised that further surface treatment could be applied to the spheres to give a dull-grey surface finish. However, when measuring the ball bar artefact with the optical sensor, it was observed that the dull-grey surface finish did not completely eliminate specular reflection on the two spheres. Therefore, to further reduce specular reflection, developer powder was applied to the spheres.

4.4 Automated Feature Detection

4.4.1 Objective

As highlighted in Chapter 2, most calibration methods require the use of a calibration artefact having known features which have been independently measured. Apart from the accuracy of the independently measured feature coordinates (i.e. control points), calibration accuracy is also significantly influenced by the accuracy to which these features are detected. Thus, a significant amount of time is typically dedicated to detecting and ordering these artefact features from measured data sets. Automated feature detection could therefore provide functionality for simplifying and speeding up the calibration process of the optical SMS, thereby enabling rapid set up and deployment for measurement tasks.

4.4.2 Finding Circles using the Hough transform

The Hough transform is recognised as a powerful tool in shape analysis which gives good results even in the presence of noise and occlusion [90]. Though Hough [91] proposed this method of identifying patterns in images, it was aimed initially at detecting straight lines. However, Duda and Hart [92] extended the method to find more general classes of curves in images and were the first to use the Hough transform to detect circles. Kimme et al [93] achieved a computationally efficient improvement on this circle finding concept, by using the direction of the gradient to vote in the Hough accumulator, and were able to achieve up to 82% reduction in processing time. Therefore, the Hough transform is often used for detecting simple 2-D features or shapes such as lines and circles, while techniques such as neural networks and genetic algorithms are used for the detection of more complex shapes.

For 2-D images, the coordinates of circles need to be quantized into a pixel coordinate system, in addition, some pre-processing is necessary before the Hough transform can be carried out. The pre-processing involves edge detection to locate the edge of the circle, and thresholding to remove background noise. An edge in an image refers to a pixel where a significant change in intensity occurs over a short distance. This change or discontinuity can be calculated from the first derivatives of the image with respect to its two axes. For an image $I(x,y)$, a finite differencing scheme is often used to calculate the first derivatives :

$$\nabla I(x, y) = [I_x, I_y] \quad (4-1)$$

where

$$I_x = \frac{\partial I(x, y)}{\partial x_i} \approx I(x_{i+1}, y_i) - I(x_i, y_i) \quad (4-2)$$

$$I_y = \frac{\partial I(x, y)}{\partial y_i} \approx I(x_i, y_{i+1}) - I(x_i, y_i) \quad (4-3)$$

However, more accurate edge detection could be achieved through the use of gradient operators (e.g Robert's cross operator, 3x3 Prewitt operator, Sobel operator, and 4x4 Prewitt operator). On completing these operations, it is expected that only circle edge coordinates would be selected for use in the Hough transform.

In the parameterised form, the equation of a circle with centre coordinate (a, b) is:

$$(x_i - a)^2 + (y_i - b)^2 = r^2 \quad (4-4)$$

where (x_i, y_i) are the Cartesian coordinates of a point on the circle edge, (a, b) are the coordinates of the centre, and r is the radius.

The above equation has three unknown parameters, (a, b, r) , and every edge coordinate (x_i, y_i) in image space corresponds to a surface on a right circular cone in (a, b, r) parameter space (see Figure 4-7). The (a, b, r) parameter space is commonly referred to as the *accumulator array*, which from here on would be referred to as the *Hough space*. The Hough space can therefore be considered to be a histogram having multi-dimensional *bins*, where edge coordinates contribute votes, and the coordinates of bins having high votes should coincide with circle centre coordinates. The dimension of the Hough space corresponds to the number of unknown parameters in the equation of the family of curves being sought [94]. In a case where we have *a priori* knowledge of r , the Hough space becomes two-dimensional.

Considering a circle edge coordinate (x_i, y_i) as shown in Figure 4-8, we can write the equations

$$(x_i - a) = r \cdot \sin \theta_j, \quad (4-5)$$

$$(y_i - b) = r \cdot \cos \theta_j. \quad (4-6)$$

Therefore, for each edge point (x_i, y_i) , with θ_j in the range 0 to 2π , there would be a corresponding circular locus of possible (a, b) values in the Hough space. From Figure 4-9, this means that each edge coordinate of the circle in (x, y) space (the green dots) would contribute votes in multiple bins, and the bin coordinate corresponding to the circle centre would contain the maximum votes. This implies that most of the bins of the Hough space would contain redundant data, thus proving computationally expensive if a large number of circle edge coordinates is being considered. However, the orientation of the edge coordinates provides an additional constraint on the set of possible parameter values [95]. The maximum gradient of each edge coordinate on a circle should always point in a direction towards the circle's centre. Therefore, with *a-priori* knowledge of the circle radius, r , a coordinate one radius away in the maximum gradient direction would mark the expected position of the circle centre. This restricts each edge coordinate to just one vote for a single bin in Hough space. The maximum gradient direction is given by

$$\theta = \tan^{-1} \left(\frac{I_y}{I_x} \right) \quad (4-7)$$

The strength of the edge is

$$\|\nabla I(x, y)\| = \sqrt{I_x^2 + I_y^2} \quad (4-8)$$

$$\cos \theta = \frac{I_y}{\|\nabla I(x, y)\|} \quad (4-9)$$

$$\sin \theta = \frac{I_x}{\|\nabla I(x, y)\|} \quad (4-10)$$

Combining Eqns. (4-1) to (4-10), we obtain the circle's centre coordinates

$$a = x_i - r \cdot \frac{I_x}{\|\nabla I(x, y)\|} \quad (4-11)$$

$$b = y_i - r \cdot \frac{I_y}{\|\nabla I(x, y)\|} \quad (4-12)$$

For each edge coordinate (x_i, y_i) , we calculate the corresponding coordinates of the bin for which a vote is contributed, thus, large votes are built up at bins corresponding to circle centre coordinates. However, for a distorted circle e.g. an ellipse, rather than voting for the same bin in Hough space, edge coordinates not only vote for the bin at its true centre coordinates, but also in other bins around it. To compensate for this loss of votes, edge coordinates could be made to vote in bins that lie on a line segment of length $2dr$ (where dr is in pixel coordinates) along the gradient direction. This is achieved through a range of values centred around r , which are applied to Eqns. (4-11) and (4-12) thus:

$$r_j = r - dr : r + dr \quad (4-13)$$

The next stage of the Hough transform is to successfully identify these coordinate bins that correspond to the circle centres through peak detection. Further discussions on peak detection in Hough space are given in Section 4.5.4.

Yuen et al [96] did a comparative study of the Standard Hough Transform (SHT), Gerig and Klein method of Hough Transform (GHT), Gerig and Klein Method of Hough Transform using Edge Gradient Information (GHTG), Two-Stage Hough Transform (21HT), and Fast Hough Transform (FHT) methods in terms of accuracy, reliability, computational efficiency, and storage requirements, and concluded that the GHTG method was the best overall. Although the GHTG method requires modest storage in 2-D, and is reasonably fast, accurate and reliable, its only drawback is its inability to detect concentric circles (which will not be an issue when extended to sphere detection) [96]. A general conclusion by Yuen et al [96], is that more complicated variations of the Hough transform do not necessarily outperform straightforward approaches. Therefore, a good performance of the GHTG method would also be expected for sphere detection. The problem is simplified in this case

since we have *a priori* knowledge of the sphere radius, thus restricting the number of sought parameters to three.

4.4.3 Finding sphere centres using the Hough Transform

There is not much information in the literature regarding the use of the Hough transform on 3-D data. This is probably due to the problems of excessive memory storage requirements and computational complexity, and the bulk of most contributions to the literature aim to provide solutions to these problems [90]. van der Glas et al [97, 98] developed a Hough-based method which expands on the concept for detecting circles to detecting spheres. Developed primarily for medical applications, their method automatically detects the centre and size of a sphere in 3-D grey-value images generated by CT or MRI scanners. Rabbani and van den Heuvel [99] have also used the Hough transform for automatic cylinder detection. Instead of using a full 5-D Hough space for detecting all the cylinder parameters, two sequential steps of lower-dimensional Hough transforms were used.

However, in the current work, the motivation is to develop a robust, computationally efficient, and accurate 3-D Hough transform that can be used for detecting spheres in large 3-D data sets (presented in the form of point clouds), suitable for use in the calibration of an optical SMS. 'Large' in this case means typically of order 10^6 points. This novel implementation of the 3-D Hough transform, which is based on an extension of the 2-D method for detecting circles, exploits the sparse nature of the 3-D Hough space for spheres through the use of an optimised sparse 3-D matrix model in order to provide compact data storage and efficient data access. An efficient peak detection algorithm has also been developed to detect peaks in the Hough accumulator space to identify the sphere centre coordinates. Further post-processing of the estimated sphere centre coordinates through non-linear optimisation has also been investigated to improve on the accuracy of the method. Finally, a robust algorithm has been developed to determine the orientation of the sphere artefacts so as to identify and label all the spheres automatically.

Moving to three dimensions, the equation of a sphere is a straightforward extension of Eqn. (4-4):

$$(x_i - a)^2 + (y_i - b)^2 + (z_i - c)^2 = r^2 \quad (4-14)$$

where r is the sphere radius expressed in the same units as the 3-D Cartesian coordinates (x, y, z) . The fact that the points from a 3-D sensor already define the surface of the component means that all points in the 3-D point cloud data contribute votes in Hough space. The dependence on robust edge detection algorithms needed for 2-D implementations using gray-scale images is therefore removed. A further difference is that a 3-D sensor provides a direct measure of the Cartesian coordinates of the points in object space, whereas a 2-D Hough transform is normally applied to the projection of the scene as viewed in image space. The drawback of working in image space is that the apparent circle radius is normally dependent on the object range, which may be unknown. Returning to the 3-D case, the unit outward-pointing surface normal $\hat{\mathbf{n}} = [\hat{n}_x, \hat{n}_y, \hat{n}_z]^T$ (where superscript T denotes the matrix transpose operator) can be estimated at each measured point (x_i, y_i, z_i) using a least squares fit of a plane to the coordinates of the point and its nearest neighbours. The computed sphere centre coordinates from the i th point in the cloud are obtained as follows:

$$\begin{aligned} a_i &= x_i - r \hat{n}_x \\ b_i &= y_i - r \hat{n}_y \\ c_i &= z_i - r \hat{n}_z \end{aligned} \quad (4-15)$$

Once all points have voted, through application of Eqn. (4-15) followed by binning, the Hough space can be searched for peaks whose locations represent the coordinates of the candidate sphere centres.

Although simple in principle, a naive extension from 2-D to 3-D leads to much larger data storage requirements. Significant savings can be made, however, by recognising that only a small portion of the Hough space would normally contain useful data. For example, in a single-camera single-projector SMS, with the camera having 1000×1000 pixels, and Hough space discretised into 10^9 bins, a maximum of one bin in 1000 in the Hough space on average would contain any data. In practice the fraction would be even smaller since the votes tend to cluster into regions near the sphere centres (see Figure 4-10 and 4-11 for an example). As a result, the Hough space can be thought of as a sparse matrix. The data storage problem of the Hough

transform for sphere detection can therefore be considered as a data storage and efficient data access problem of a sparse matrix.

4.4.4 Sparse matrix representation

4.4.4.1 Methods

Sparse matrix representations are pervasive in scientific and engineering application codes and they often arise from finite difference, finite element, or finite volume discretizations of partial differential equations (PDEs) or from discrete, network-type problems [100]. The main purpose of a sparse matrix representation is to reduce data storage space by ensuring that only non-zeros are stored and easily accessed. Many different ways of storing sparse matrices have been devised to take advantage of the structure of the matrices or the specific nature of the problem for which they arise [101]. Dongarra et al [100] developed a sparse matrix library, SparseLib++, and concentrated on what they referred to as the most commonly used data structures. They include [100]:

- Sparse vector: a list of non-zero elements with their indices without ordering of the elements,
- Coordinate Storage: a list of non-zero elements and their respective row and column indices. This method is considered to be memory and computationally inefficient.
- Compressed Row Storage: all non-zero entries are stored row by row in a one-dimensional real array, **A**, together with an integer array, **IA**, containing their column indices, a pointer array, **PA**, which contains the addresses in **A** and an integer array, **JA**, that contains indices to the beginning of each row.
- Compressed Column Storage: similar to compressed row storage above, but instead of rows being stored in **A**, the non-zeros entries are stored column by column.
- Compressed Diagonal Storage: involves storing the sub- and super-diagonals contiguously.
- Jagged Diagonal Storage.
- Block Compressed Row Storage: takes advantage of square dense blocks of non-zeros in some regular pattern.

All of these methods require *a priori* knowledge of the region of the matrix that will contain the non-zeros. To apply these methods in this application would require allocation of memory for the full Hough space. Only after population of the space could a compressed form be created.

A more efficient alternative was found by implementing a *hash table* as an optimised 3-D sparse matrix representation. This approach avoids the need to allocate, even if only temporarily, a large amount of memory for the Hough space. Also, it minimises the time to store and retrieve data by providing a direct mapping between matrix indices and their corresponding values.

4.4.4.2 Hashing

A hash table data structure is an array of fixed size containing *keys* and data *values* [102]. A key may be any bit pattern (e.g. an ASCII text string or a structure containing integer values) to which some value is associated and can be considered to be part of a group of data by which the hash table is sorted, indexed, and cross referenced [103]. The process of hashing can be broken down into two aspects, namely, *hash table indexing* and *key search*.

Hash table indexing refers to the mapping of keys to a finite number of storage cells by a mapping transformation called a *hash function*. The hash function implements a dictionary in which keys are mapped to hash table addresses [104]. For a hash table with size M , the hashing function maps the key to a table address in the range $[0, M-1]$, however, this mapping is not guaranteed to be unique, with the number of possible unique keys much larger than M in many applications. A simple hash function for numerical integer key values, k , could take the form [105, 106]:

$$h(k) = k \text{ modulus } M \quad (4-16)$$

where $h(k)$ is the computed hash table address.

The hash table data structure provides an interesting model for implementing a sparse matrix since the number of unique combinations of row and column indices (i.e. the number of possible unique keys) may be much larger than the allocated table storage size.

Pseudo-random number generator algorithms with deterministic behaviour provide a popular basis for implementing hash functions since they are able to create a uniformly distributed sequence of integer values with little or no discernible pattern

other than broad statistical properties [104]. The hash function implemented in this work is derived from a pseudo-random number generator [107] as follows

$$h(k) = (k * f) \gg (d \& s) \quad (4-17)$$

where '*' is a fixed width binary multiplication operator, '>>' is a bitwise right shift operator, '&' is a bitwise AND operator, k is a hash table key, f is a large valued integer constant with respect to the range of values that can be represented by the arithmetic processor (e.g. using hexadecimal notation, $f=0x41C64E6D$ for a 32-bit processor), d and s are integer constants chosen to scale the result into the range $[0, M-1]$. A suitable choice for f will result in an arithmetic overflow for many key values and leads to an address distribution $h(k)$ that is approximately uniform across the range $[0, M-1]$.

Since the hash function does not provide a unique mapping from key to hash table address, collisions must be resolved using a process referred to as *key searching*. In general the cost of read and write operations on the hash table is reduced by minimising the frequency of these collisions for a given set of keys. Two common techniques for handling collisions are *open addressing* and *chaining*, both of which provide the following advantages [102]:

1. hash table addresses are calculated in constant time using a simple arithmetic formula
2. sufficient memory is available for storing key-value pairs
3. calculated hash table addresses distribute elements uniformly throughout the allocated memory

In the open addressing method, memory is allocated for the hash table based on *a-priori* knowledge of the number of elements to be stored. For example, the simplest open addressing method, *linear probing* [106], implements a scheme such that the next position in the table is searched when a collision is detected. This method relies on the availability of empty space in the table in order to resolve collisions.

In this work, the chaining method implemented using a *linked list* structure has been chosen, since this enables efficient dynamic memory allocation [103-106]. A linked list is a collection of nodes of non-contiguous memory locations, which are combined to form a linear ordering such that each node contains a reference to a data type and a link to the next node on the list [105]. Here, insertion of new nodes can occur at only one end of the list - at the 'head'. Therefore, key searching is uni-directional. Thus, in

this scheme, the hashing table can be visualised as a vector in which each element stores a linked list, with keys mapped to a linked list using the hash function given in Eqn. (4-17).

In comparison with open addressing, the chaining technique is insensitive to clustering, requires less memory, and provides better performance when the number of stored entries exceeds the hash table size. In applications where the number of keys is not known in advance, the chaining technique is preferred since it provides efficient dynamic memory allocation. Algorithms for hashing and linked lists are further described in [102, 105, 106].

4.4.5 Peak detection in Hough space

4.4.5.1 Methods for peak detection

Once all 3-D coordinates have been used for voting in Hough space, a threshold, t , can be applied to remove clutter, ensuring that only coordinate bins containing relatively large numbers of votes, with strong peaks, are left remaining. Although all the 3-D coordinates on a sphere surface should ideally contribute a vote for the same coordinate bin in Hough space, in practice, peaks build up in a cluster of coordinate bins around the true sphere centre coordinate. This is chiefly due to calibration errors of the measuring instrument and electrical noise from its constituent sensors. Increasing the dimensions of the bins can be used to reduce this effect, with the additional benefit of a reduction in memory requirements. However, if carried too far, this will be at the expense of a reduced accuracy in the estimate of the sphere centre coordinates.

One means of quantifying the bin size appropriate to a specific sensor system is through the root mean square deviation W of the (a_i, b_i, c_i) coordinates calculated from Eqn. (4-15). W is defined as

$$W = \sqrt{\frac{\sum_i (a_i - \bar{a})^2 + \sum_i (b_i - \bar{b})^2 + \sum_i (c_i - \bar{c})^2}{n}} \quad (4-18)$$

where n is the total number of contributing points on the sphere surface, and \bar{a} , \bar{b} and \bar{c} are respectively the means of the a_i , b_i and c_i . Figure 4-12 provides an example of Hough space, before and after binning, using simulated data from a hemi-sphere. The

point cloud consisting of 2724 points, from a hemi-sphere with radius = 20mm and centre coordinate $a = 100$ mm, $b = 100$ mm, and $c = 100$ mm, is shown in Figure 4-12 (a). Random numbers with a standard deviation of 200 μm have been added to each of the x , y and z coordinates. The resulting scatter in the (a_i, b_i, c_i) coordinates, computed from Eqn. (4-15) using a 3×3 kernel to estimate the surface normals, is apparent in Figure 4-12 (b). The distribution of votes on a slice through Hough space ($c = 100$ mm) is shown in Figure 4-12 (c). The bin size (1 mm along each axis) is 32% of the computed W value (3.09 mm). The effect of varying the noise level on the number of bins containing votes, the vote count of the bin with maximum votes and W , may be seen from Table 4-1, in which the results of simulations at different σ values (5 μm , 10 μm , 20 μm , 50 μm , 100 μm and 200 μm) are compared. The bin size was fixed at $1 \times 1 \times 1 \text{ mm}^3$ throughout.

It is clear from Figure 4-12 (c) and Table 4-1 that even when considering the results of idealised simulations, some method of handling coordinate clusters will be required. In this section, we discuss some of the different peak detection methods that are available from the literature.

Fisher and Naidu [108] compared five algorithms – Gaussian approximation, centre of mass, linear interpolation, parabolic estimator, and the Blais and Rioux detectors, for determining the peak image position of an image line or stripe to sub-pixel resolution. Furukawa and Shinagawa [109] focussed on detecting peaks robustly by avoiding spurious peaks, particularly in complicated scenes, by introducing a butterfly evaluator (the butterfly referring to the pattern of the Hough transform of a line segment). One of the well-known peak detection algorithms is to determine a global threshold and extract every local maximum whose vote is more than the threshold in Hough space [109]. This is commonly referred to as a local maximum method. O'rouke and Sloane [110] proposed two general data structures, Dynamically Quantized Space (DQS) and Dynamically Quantized Pyramids for peak detection in multi-dimensional histograms. O'Gorman and Sanderson [111] proposed a converging squares algorithm for efficiently detecting peaks in multi-dimensional data.

Based on the concept of clustering, a number of peak detection methods have been developed. Clustering yields a decomposition of the histogram into a few non-

overlapping intervals, and labelling of the clusters results in multi-thresholding of the image [112]. A clustering algorithm based on the minimum volume ellipsoid (MVE) robust estimator was introduced by Jolion *et al.* [112], while Cheng [113] used a mean-shift algorithm for cluster analysis. More sophisticated methods like neural networks and genetic algorithms have also been used for peak detection [113]).

A combination of the local maximum method, clustering and the centre of mass method has been used in this implementation of peak detection in the 3-D Hough space, similar to the approach used by Warr, et al [95]. In this work, the connectivity checking of peaks [95] has been replaced with a more computationally efficient method through connected component labelling of the Hough space, as discussed in the next section [94]. This groups clusters of candidate coordinates that are deemed likely to have originated from a single identifiable feature. The centre of mass of each labelled object is then calculated as described by Warr et al [95].

4.4.5.2 Connected component labelling

Connected component labelling refers to the process of finding sets of pixels in an image that are connected to one another, and assigning to all such pixels a unique label, which is usually an integer. Each group of connected pixels is therefore considered as a single object with all its pixels having the same label. In a 2-D image, a pixel P , not located on the image boundary, has eight immediate neighbours, to the North, North-East, East, South-East, South, South-West, West, and North-West. A 4 or 8 connectivity check can therefore be carried out (see Figures 4-13(a) and (b)). In a 3-D volume, a 6, 18, or 26 connectivity check can be carried out by simply expanding on the concept of 4 or 8 connectivity to 3-D space.

There are two types of algorithms for connected component labelling [114]:

1. Recursive algorithms: these involve recursively assigning a label to a pixel and all its neighbours. This is achieved through repeated forward and backward scans of the image (see Figures 4-14(a) and (b)) [94, 115].
2. Sequential algorithms: this usually requires two passes over the image, where labels which have already been used are assigned to a pixel and all its neighbours.

In this work, the implementation of connected component labelling for peak detection is based on the method of recursive algorithms. Initially, a unique label is assigned to each bin in the Hough space. A forward mask is then used to scan along each axis of

the Hough space, taking bin coordinates in ascending order. The smallest label within a component is then propagated within the group of objects. A backward mask is then used to scan along each axis of the Hough space taking bin coordinates in descending order, also propagating the smallest label within the group of objects. Of course the nature of the forward and backward masks is dependent on the type of connectivity checking that is required. The algorithm stops when there is no change in any of the labels in the volume. The algorithm can be summarised thus:

1. Label all non-zero values with a unique label, say from 1 to n , where n = number of non zeros after applying an appropriate threshold, t .
2. Select scan mode (after first scan, alternate backward and forward modes).
3. Get coordinate of the i^{th} bin, $Q_i(a_j, b_k, c_l)$, where j, k, l are the indices into the Hough space along X, Y and Z axis respectively.
4. Check neighbours of Q_i (depending on the type of connectivity specified), including Q_i , and identify the smallest label, L , in the group.
5. Assign L to Q_i and all its neighbours.
6. Go to next non-zero bin (step 3).
7. If no label has changed after going through the volume, then stop, else go to step 2.

A detailed description of the algorithm can be found in references [94, 114, 115].

From the connected component labelling algorithm, an $n \times 4$ table of labels is obtained, where n is the number of non-zero bins in the Hough space after applying an appropriate threshold, t . Therefore, the size of the label table is dependent on the value of t . For each row, the first three columns indicate the bin coordinates, while the fourth column contains the label assigned to the bin. Therefore, on the fourth column in the label table, all the connected bins of each cluster in Hough space will have the same value. However, a single coordinate needs to be calculated from the cluster of coordinates. This is achieved by calculating the centre of mass of each cluster through

weighted averaging of sphere centre coordinates. For each cluster, the mean x , y , z coordinates for the sphere centre, \bar{x} , \bar{y} and \bar{z} , are given by

$$\bar{x} = \frac{\sum_{i=1}^n a_i V_i}{N}, \quad \bar{y} = \frac{\sum_{i=1}^n b_i V_i}{N}, \quad \bar{z} = \frac{\sum_{i=1}^n c_i V_i}{N} \quad (4-19)$$

where the normalisation term, N , is given by

$$N = \sum_{i=1}^n V_i \quad (4-20)$$

and where V_i is the number of votes for voxel i , and n is the number of voxels in the cluster.

4.4.6 Identification of non-viable peaks

To avoid detecting false peaks that identify false sphere centre coordinates, all the average coordinates corresponding to each cluster are checked to ensure that non-viable centre coordinates are discarded. Since we have *a priori* knowledge of the radius of each sphere and the number of spheres in the point cloud, this information can be used to identify centre coordinates that are unrealistic. If the Euclidean distance between two candidate centre coordinates, \mathbf{r}_1 and \mathbf{r}_2 , is below a certain threshold (taken here as $|\mathbf{r}_1 - \mathbf{r}_2| < 1.9r$), the coordinate with the higher number of votes is retained, while the other is discarded as a false centre coordinate. In the case where two candidate centre coordinates are close together ($|\mathbf{r}_1 - \mathbf{r}_2| < 0.01r$), the coordinates are merged together, and a single coordinate which is a weighted average of the two is used as an improved estimate. If the required number of centre coordinates is not detected, this would mean that the global threshold set for removing clutter in the Hough space is not appropriate. This threshold is then increased or decreased automatically.

Having estimated all sphere centre coordinates in the point cloud, the known radius of the sphere can be used to identify all the corresponding points on the sphere surface. In this way, we are able to uniquely identify all the 3-D coordinates that lie on sphere surfaces in the point cloud and differentiate between them.

4.4.7 Post-processing of sphere centres

4.4.7.1 Nonlinear optimisation

In applications where the accuracy of the sphere centre coordinates is important, it is advantageous to introduce a method to improve on the accuracy to which they have been estimated with the 3-D Hough transform. One approach is to use all the 3-D coordinates \mathbf{r}_i ($i = 1, 2, \dots, n_s$) that have been identified by the 3-D Hough Transform as belonging to a given sphere to improve the estimate of its centre coordinates \mathbf{r}_0 through least squares analysis. A measure of the error between the optically measured radius of each sphere and its ideal radius may be written as

$$E_i = |\mathbf{r}_i - \mathbf{r}_0| - r \quad (4-21)$$

The objective function, S , to be minimised is

$$S = \sum_{i=1}^{n_s} |E_i|^2 \quad (4-22)$$

which is a nonlinear function of the three unknown components of \mathbf{r}_0 . Newton's method was selected to solve this minimisation problem, modified according to the Levenberg and Marquardt algorithm [87] to prevent the objective function from increasing if the initial estimate of \mathbf{r}_0 is too far from the true minimum. For the j^{th} iteration, \mathbf{r}_0 was calculated thus:

$$\mathbf{r}_{0,j+1} = \mathbf{r}_{0,j} - w_j \mathbf{d}_j \quad (4-23)$$

where $\mathbf{d}_j = \mathbf{H}_j^{-1} \mathbf{G}_j$, the modified search direction, \mathbf{G}_j and \mathbf{H}_j are respectively the gradient vector and Hessian matrix calculated at each iteration from the analytical derivatives of Eqns. (4-20) and (4-21) (details shown in Appendix A-1), w_j is the weighting applied to \mathbf{d}_j , which is set to 1 at the start of the optimisation process, however, if the current value of w_j increases S , a new value for w_j which minimizes S is computed, and used for subsequent iterations in Eqn. (4-23).

4.4.7.2 Ordering of coordinates

Although the 3-D Hough transform identifies the spheres in the point cloud, the resulting list of sphere centre coordinates is not in general ordered, and furthermore the coordinates are specified in the coordinate system of the optical SMS. In order to

use the estimated sphere centre locations as part of calculations in the calibration of a multiple-camera multiple-projector SMS, it is necessary to associate each identified sphere uniquely with a given sphere on the artefact whose coordinates are known (e.g., through separate measurement with a mechanical CMM). Also, the transformation between the mechanical CMM and optical SMS coordinate systems needs to be estimated.

This in general is a six-dimensional non-linear optimisation problem, and is similar to what is usually referred to as the Procrustes problem or the rigid body movement problem [116]. There are generally two categories of solutions for this relative pose problem: quaternion based and singular value decomposition - based (SVD-based) [117]. In quaternion based methods [118, 119], rotations are represented as quaternions in order to simplify the problem of enforcing the orthonormality constraint which arises when using matrices to represent rotations. However, SVD has been proven to be both reliable and numerically efficient to calculate the transformation matrices for this problem [116].

Arun et al [120] proposed a non-iterative algorithm for the estimation of rigid body motion and relative attitude of a rigid object with respect to a reference, involving the SVD of a 3x3 matrix. This algorithm has been used by Sjudahl and Oreb [116] as part of a stitching strategy for the inspection of large, predominantly flat high quality surfaces, where a set of individual phase measurements are stitched together into a unified larger phase map. The objective is to find the transformation that minimises the sum

$$\min_{\mathbf{R}, \mathbf{T}} \sum_{i=1}^n \|(\mathbf{R}\mathbf{A}_i + \mathbf{T}) - \mathbf{B}_i\|^2 \quad (4-24)$$

where \mathbf{A}_i = misaligned data set, \mathbf{B}_i = Reference data set,

\mathbf{R} = rotation matrix, \mathbf{T} = translation vector.

The transformation is obtained through calculating the zero mean matrices, $\bar{\mathbf{A}}_i$, $\bar{\mathbf{B}}_i$ i.e deviation from centroid, for each data set thus:

$$\bar{A}_i = A_i - \bar{A} \quad (4-25)$$

$$\bar{B}_i = B_i - \bar{B} \quad (4-26)$$

$$[u, w, v] = SVD(\bar{B}_i \bar{A}_i^T) \quad (4-27)$$

where $\bar{A} = \frac{\sum_{i=1}^n A_i}{n}$, the centroid of misaligned data set, $\bar{B} = \frac{\sum_{i=1}^n B_i}{n}$, the centroid of the reference data set, *SVD* = implementation of singular value decomposition, $u = 3 \times 3$ orthogonal matrix, $w = 3 \times 3$ diagonal matrix, $v = 3 \times 3$ orthogonal matrix

In the least squares sense, the optimal rotation matrix M , between A and B is

$$R = uv^T \quad (4-28)$$

and the optimal translation vector, T , is

$$T = \bar{B} - R\bar{A} \quad (4-29)$$

The process of labelling the spheres is straightforward in the case of the ball bar artefact which simply consists of two spheres of known separation –the sphere closer to the origin is assigned label ‘1’, while the other is assigned label ‘2’. For the more complex multiple spheres calibration artefacts, a two-step process was used. This consisted of first identifying and labelling the four spheres furthest from the geometric centre of both optically-measured and mechanically-measured coordinate lists. The SVD-based algorithm provided an initial estimate of the required coordinate transformations. The second step then involved finding the closest optically-measured sphere to each mechanically-measured sphere, thereby allowing the list of optically-measured coordinates to be placed in the same order as the list of mechanically-measured coordinates. A second application of the SVD algorithm provided an improved estimate of the required coordinate transformations to bring the two sets of coordinates into registration. The algorithm can be summarised as follows:

1. identify the four corner spheres of the artefact: this is achieved by determining the centroid of all the optically measured centres, O (an $n \times 3$ matrix of x, y, z coordinates) and mechanically measured centres M (an $n \times 3$ matrix of x, y, z coordinates) respectively. The Euclidean distance of each coordinate to the centroid is then calculated. The set of four coordinates, O_c and M_c , which have

the *furthest* distances to the respective centroid of **O** and **M**, should represent the centre coordinates of the spheres at the four extreme corners of the artefact.

2. From these four coordinates, **O_c** and **M_c** respectively, a reference coordinate, **S₁**, is chosen arbitrarily while the others are labelled, **S₂**, **S₃**, and **S₄**. The Euclidean distance of **S₂**, **S₃**, and **S₄** are calculated to **S₁**, respectively. The coordinate having the furthest distance to **S₁** should be a sphere coordinate diagonal to **S₁**, and will be labelled **S₃**. The remaining two coordinates will be labelled **S₂** and **S₄**, or **S₄** and **S₂** (see Figure 4-15). This will prove important because there are eight possible orientations in a clockwise sense as follows (with each row representing the order of selection of sphere centre coordinates):

$$\begin{bmatrix} S_1 S_2 S_3 S_4 \\ S_2 S_3 S_4 S_1 \\ S_3 S_4 S_1 S_2 \\ S_4 S_1 S_2 S_3 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} S_1 S_4 S_3 S_2 \\ S_4 S_3 S_2 S_1 \\ S_3 S_2 S_1 S_4 \\ S_2 S_1 S_4 S_3 \end{bmatrix}$$

3. For each of the eight orientations of **O_c**, the SVD algorithm based on Eqns.(4-23) to (4-28) is used to obtain the corresponding transformations, **R_k** and **T_k**, (where $k = 1,..8$), with respect to **M_c** (whose orientation is fixed as [**S₁ S₂ S₃ S₄**]). As **R_k** is orthogonal, therefore, orientations where the determinant of **R_k** gives a value of -1, i.e $det(\mathbf{R}_k) = -1$, means a reflection or roto-inversion. For proper rotation matrices, $det(\mathbf{R}_k) = 1$ [120]. Thus, **R_k**, is checked for roto-inversion.
4. For orientations that give $det(\mathbf{R}_k) = 1$, the transformation is applied to **M_c** coordinates, and a closest proximity approach is used in matching corresponding **O** coordinates, thereby re-ordering **O**.

$$\mathbf{M}_c^T = \mathbf{R}_k \mathbf{M}_c + \mathbf{T}_k \tag{4-30}$$

We would therefore expect that the transformation calculated from the k^{th} orientation which re-orders **O_c** to give the lowest least squares sum is the best match for **M_c**.

5. On identifying the ordering that provides a best match in a least squares sense, the SVD algorithm is then applied once again to M and the ordered data set, O , to get a better estimate of the rotation matrix and translation vector. This transformation is then applied to M :

$$M^T = RM + T \quad (4-31)$$

4.4.8 Experimental Results

The artefact having 33 spheres with stalk heights randomly distributed within a 5 mm to 45 mm range above the top surface of the base is used for discussions in this section. Although the manufacturing process selected meant that the spheres were not made to high accuracy, it provided a convenient method of rapidly prototyping different artefact designs. The artefact was measured with a mechanical CMM and used to calibrate a two-camera two-projector optical SMS (Figure 4-16). The artefact was measured in a variety of orientations at different locations in the measurement volume. The results of using the 3-D Hough transform to identify and label the spheres in experimental shape data are discussed in this section.

Figures 4-17(a) and (b) shows sample grey-level intensity images of the artefact as viewed by one camera, with uniform illumination from the two projectors. Using the reversed exponential fringe projection algorithm with each projector in turn [9], corresponding point clouds were computed for each camera-projector pair. The surface rendered point clouds resulting from camera 1 with the respective projectors (C1-P1 and C1-P2) are shown in Figure 4-18. The spheres identified by the Hough transform algorithm, operating on the data from Figure 4-18, are shown in Figure 4-19.

After ordering of the spheres as described in Section 4.4.7.2, a root mean square (RMS) error was calculated from the E_i values, defined by Eqn. (4-21), for each sphere in turn. Figures 4-20 and 4-21 show the results for all the detected spheres in point clouds from two camera-projector pairs, using the r_0 values calculated by the Hough transform. In the first pair there are an average of $n_s = 2280$ 3-D coordinates per sphere, while for the second pair, $n_s = 1910$. Also shown in the same figures are the results obtained by subsequent optimization of the sphere centre estimates as described in Section 4.4.7. It can be seen that carrying out a non-linear optimisation of sphere centre coordinates significantly improves on the accuracy of the sphere centre coordinates as demonstrated by the reduction in RMS errors. The residual

discrepancies reflect a combination of measurement error, and deviations from sphericity in the manufactured artefact.

The effect of Hough space resolution on the memory requirements, calculation speed, and accuracy of estimating sphere centre coordinates have been investigated experimentally. The improvement in performance of the sparse matrix implementation relative to that of the dense matrix implementation in terms of memory usage and speed have also been compared. The size of the 3-D Hough space used to store votes is dependent on the range of coordinates (with respect to each axis in the point cloud) and its resolution i.e. scale with respect to object space. Using the shape data from one of the camera-projector pairs as an example, Table 4-2 gives details of the performance of the sparse and dense implementations. For the data set used in the performance analysis, the Hough space was a $273 \times 281 \times 160 \text{ mm}^3$ volume, while a 9×9 kernel was used to estimate the surface normals. We observe that by using an optimised sparse matrix representation, a very significant reduction in memory storage was achieved equal to approximately 95% of that required by the equivalent dense matrix implementation. As the 3-D Hough space bin size is increased, the memory storage benefits of using the sparse matrix implementation are reduced, while on the other hand, the estimates of sphere centres become less accurate. However, there is no significant effect on feature detection time, which in each case was under 1 s for the full set of 33 spheres on a computer with a Pentium4 processor (2.80GHz, 1GB RAM). Although the primary motivation of the sparse matrix representation was reduced memory requirements, these voting times are also less than the equivalent dense implementation. To this figure should be added approximately 1 s for post-processing operations such as peak detection, and typically 6 s for pre-processing steps such as computation of the surface normals.

The influence of the value of the global threshold, t , in Hough space on the accuracy of the estimated sphere centre coordinates has also been evaluated, by computing the magnitude of the error between the optically-measured and mechanically-measured sphere centres. The choice of t has some effect on the amount of clutter removal and hence the ease with which local maxima can be identified. Table 4-3 shows the effect of changing t for each of three Hough space resolutions from Table 4-2. As t is reduced, more coordinates contribute to the estimate of the sphere centre coordinate, thus improving its accuracy. It is therefore recommended that the global threshold be

set to a value as low as possible, consistent with identifying the correct number of expected features, in order to minimize measurement errors.

During a full calibration of the measurement volume, datasets from typically several tens of artefact poses are acquired, requiring several spheres to be identified, labelled, and have their centre coordinates estimated. The Hough transform algorithm described here normally achieves this fully automatically, thereby contributing to a user-friendly calibration process.

4.5 Summary

The development of calibration artefacts consisting of spheres was investigated in terms of total number of spheres, stalk heights, arrangement and manufacturing process. Three different artefacts with a multiple number of spheres were designed and manufactured using selective laser sintering. Also, a ball bar artefact was designed as a low-cost, scalable and versatile alternative to the multiple-sphere artefacts. This artefact consists of off-the-shelf precision components which are easy to source and assemble together, namely two CMM reference balls and a length bar. The separation between the two spheres can thus be increased or decreased by simply using a length bar with an appropriate length or a combination of length bars. However, in comparison with the multiple-sphere artefacts, its main draw back is that a significantly smaller fraction of the measurement volume is covered in one measurement. Therefore, in practice, provision should be made for efficient positioning at different locations and in different orientations within the measurement volume.

A novel 3-D Hough transform has also been presented for detecting spheres in a point cloud, by extending the strategy for detecting circles in 2-D images to 3-D feature detection. To overcome problems associated with the large memory requirements of the typically sparse 3-D Hough accumulator space, an optimised sparse 3-D matrix model based on a hash table has been developed to provide compact data storage and efficient data access. In addition, efficient and accurate peak detection of Hough space votes was achieved through connected component labelling and weighted averaging of coordinates with votes. Improvement in the accuracy of estimates of sphere centre coordinates was achieved through non-linear optimisation of the centre coordinates. For ordering individual sphere coordinates with respect to independently measured

coordinates, an algorithm based on singular value decomposition (SVD) was found to be reliable.

The performance of the 3-D Hough transform was assessed experimentally in terms of accuracy and speed by applying the method at different resolution settings of the Hough space and at different threshold values. At a scale of 1:1 of Hough space to object space (i.e. a bin size of $1 \times 1 \times 1 \text{ mm}^3$), the sparse matrix implementation used only an equivalent of 5% of the memory storage requirement for the dense matrix implementation in the detection process and reduced the computation time by about 50%. On completion of pre-processing operations, computation times for the Hough transform algorithm are typically within 1 s for a 33-sphere artefact, with total detection times of under 2 s.

4.6 Tables

Table 4-1 : Effect of noise on placement of votes in Hough space (with bin size of $1 \times 1 \times 1 \text{ mm}^3$) for a hemi-sphere with radius of 20 mm

Noise level, σ (μm)	Number of bins containing one or more votes	Vote count of bin with the maximum votes	W – size of cluster in Hough space (mm)
5	27	2432	0.43
10	32	2176	0.45
20	40	1656	0.52
50	85	710	0.87
100	204	463	1.57
200	554	333	3.09

Table 4-2 : Performance of 3-D Hough transform

Hough space bin size (mm ³)	Number of Hough space bins	Memory (Mb)		% of dense matrix used by sparse matrix	Voting time (s)		Average error with CMM (mm)
		Dense	Sparse		Dense	Sparse	
1x1x1	273x281x160	49.8	2.29	4.6	1.08	0.92	0.99
2x2x2	136x140x80	6.09	0.67	11.0	0.87	0.77	1.17
4x4x4	68x70x40	0.76	0.22	28.9	0.97	0.75	1.77
8x8x8	34x35x20	0.10	0.06	60.0	0.76	0.69	4.29

Table 4-3 : Effect of value of global threshold, t , on accuracy of sphere detection

Number of Hough space bins = 273x281x160		Number of Hough space bins = 136x140x80		Number of Hough space bins = 68x70x40	
Threshold (Vote count)	Average error (mm)	Threshold (Vote count)	Average error (mm)	Threshold (Vote count)	Average error (mm)
60	0.99	120	1.09	270	1.61
110	0.99	215	1.16	368	1.70
160	1.01	310	1.19	465	1.76
210	1.01	405	1.21	563	1.80
260	1.03	500	1.23	660	1.80

4.7 Figures

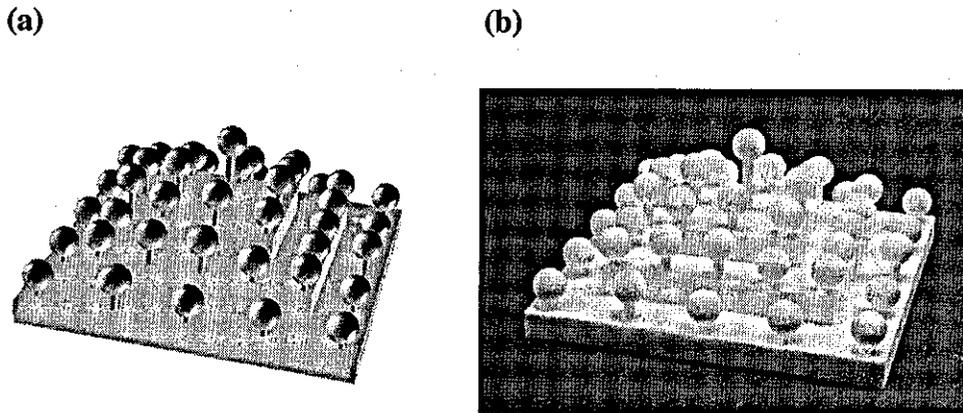


Figure 4-1: Artefact with spheres on three different planes. (a) Surface rendered image of CAD model; (b) The manufactured part.

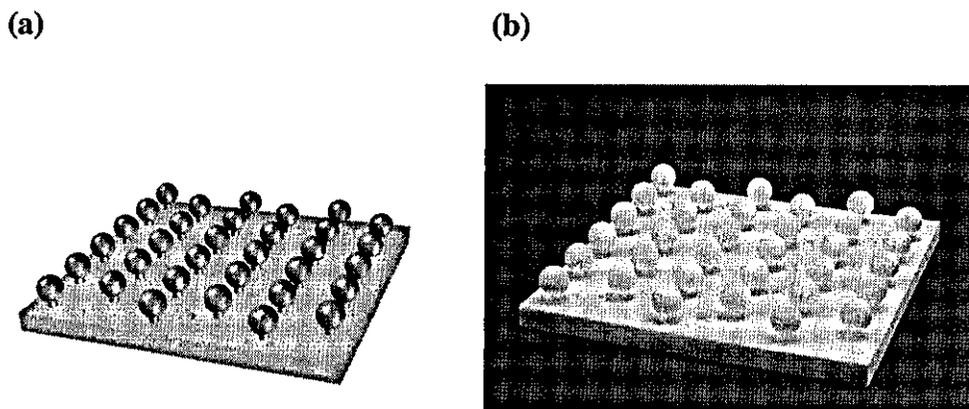
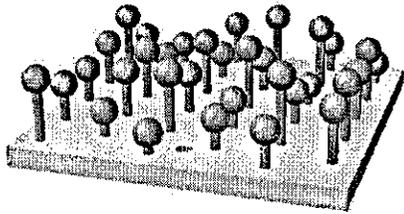


Figure 4-2: Artefact with spheres with similar stalk heights. (a) Surface rendered image of CAD model; (b) The manufactured part.

(a)



(b)

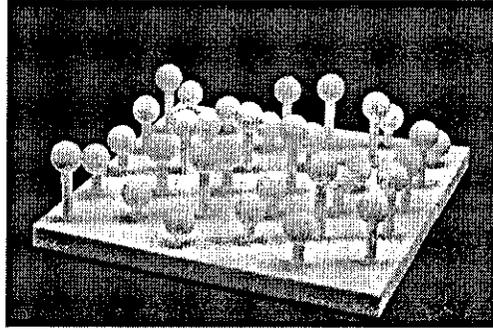


Figure 4-3: Artefact with spheres having randomly distributed heights. (a) Surface rendered image of CAD model; (b) The manufactured part.



Figure 4-4: Artefact covered with powder on leaving the SLS machine.

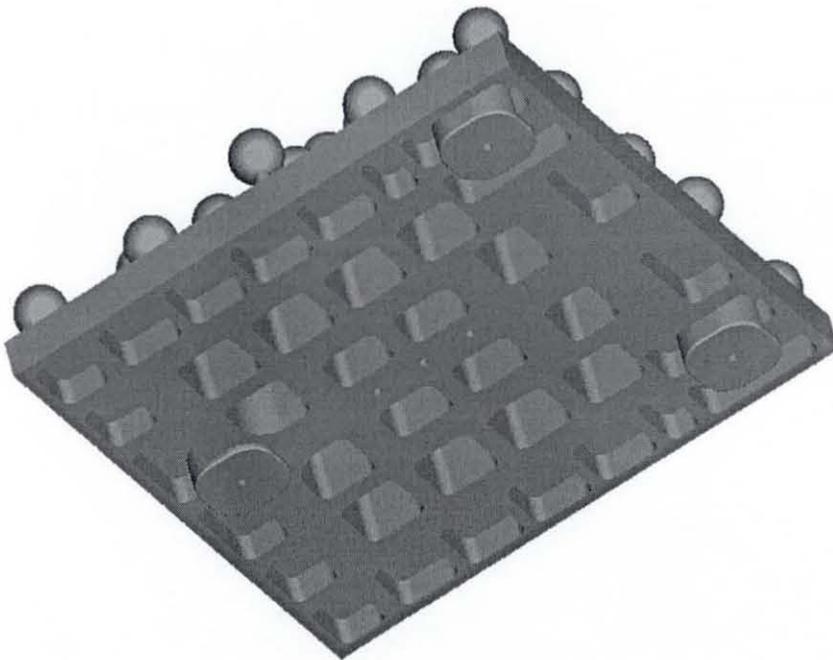


Figure 4-5: Surface rendered image of CAD model of a multiple-sphere artefact showing the base plate altered with pattern of holes.

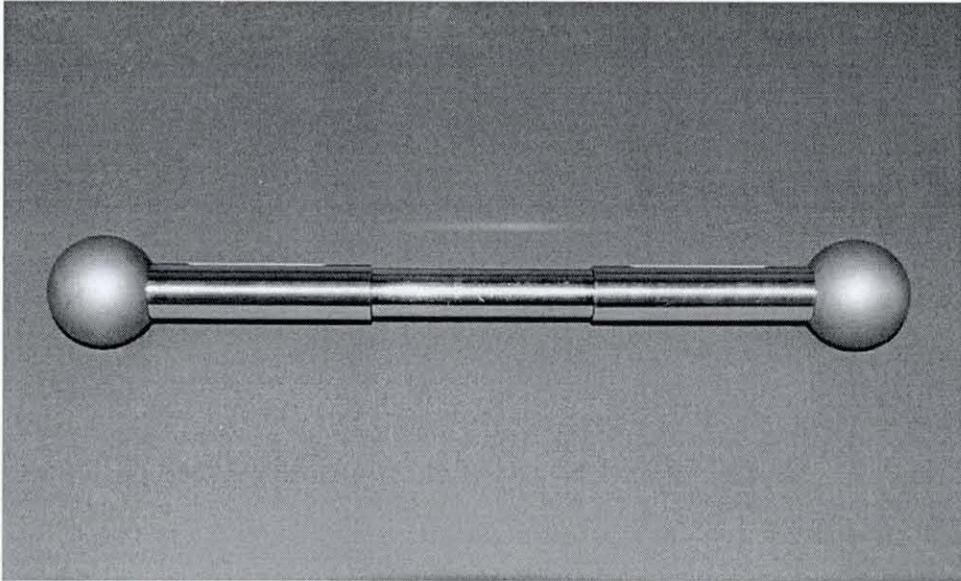


Figure 4-6 : Ball bar artefact made up of two CMM reference balls (with radius of 25.4 mm) on stalks and a 101.6 mm length bar.

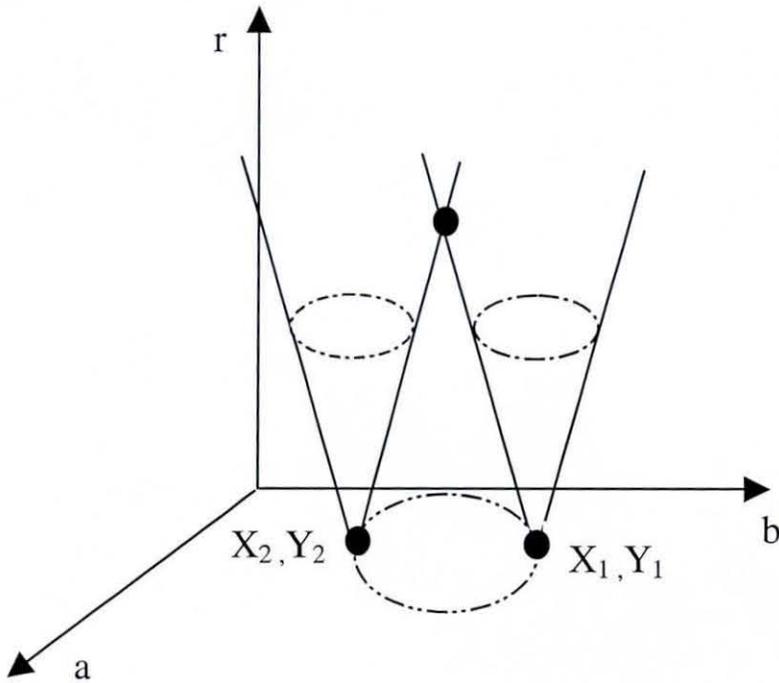


Figure 4-7 : Hough Space [96].

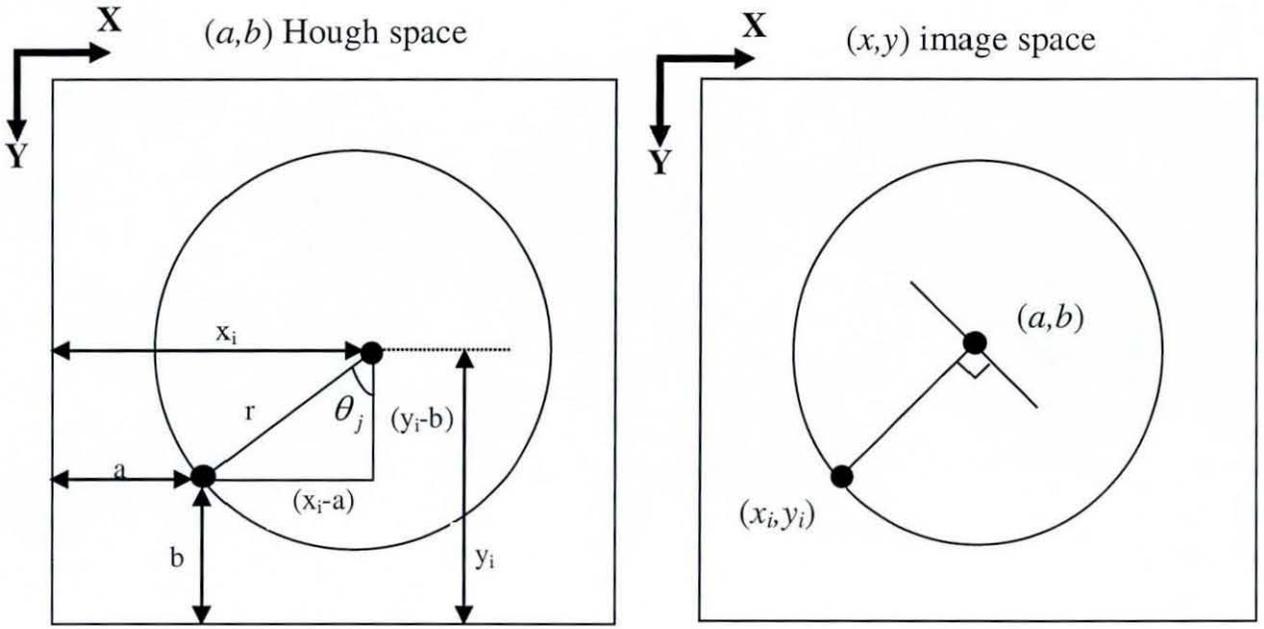


Figure 4-8 : Circle in (a,b) Hough space, and (x,y) image space.

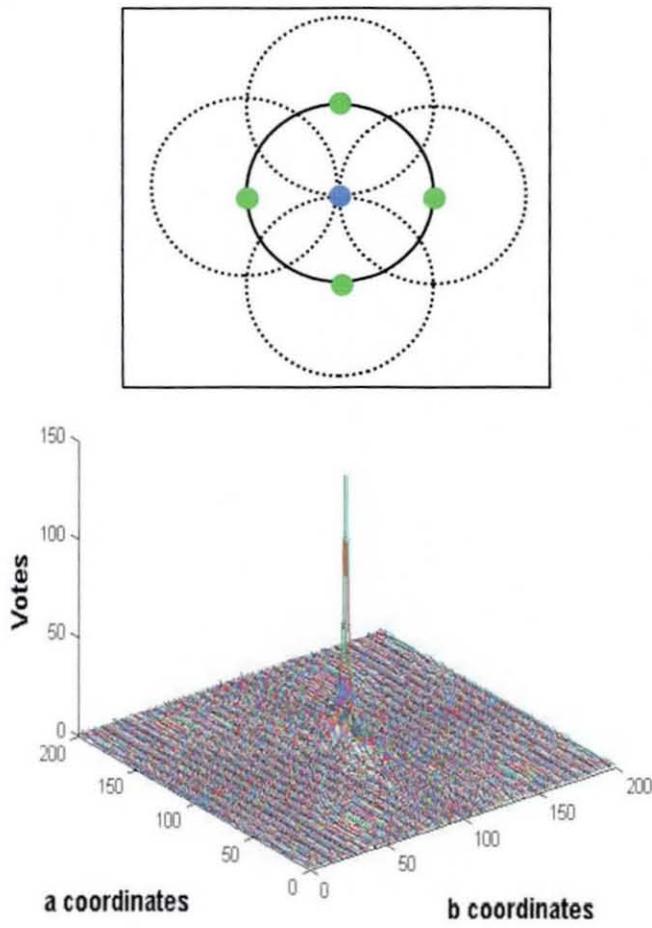


Figure 4-9 : Voting in the Hough Space.

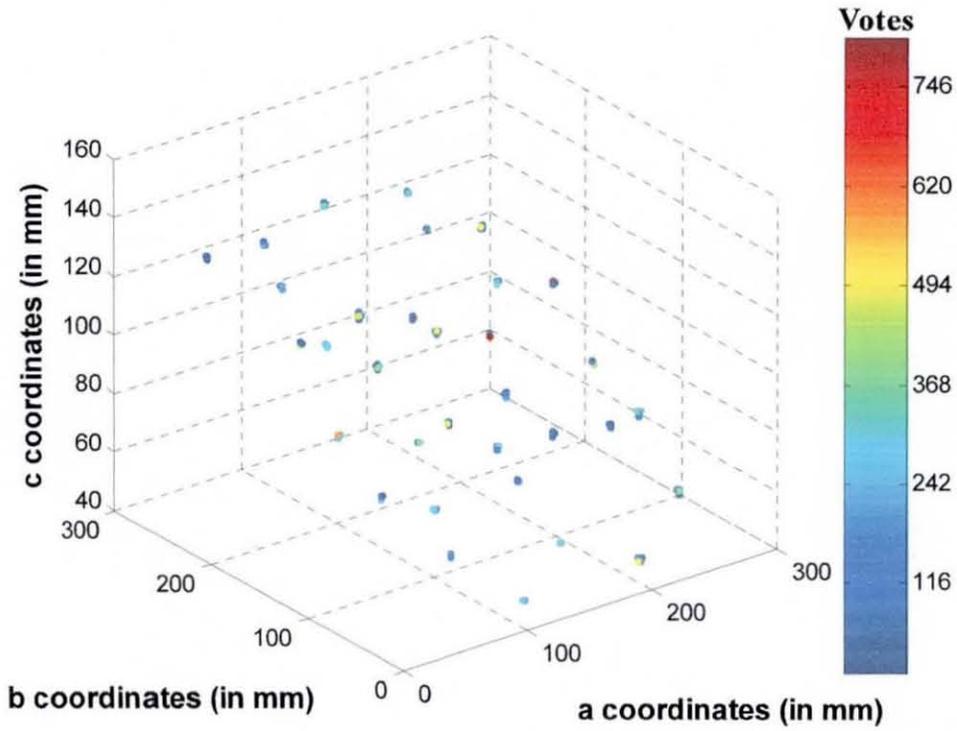


Figure 4-10 : 3-D Hough space ($273 \times 281 \times 160 \text{ mm}^3$) calculated from a calibration artefact comprising 33 spheres.

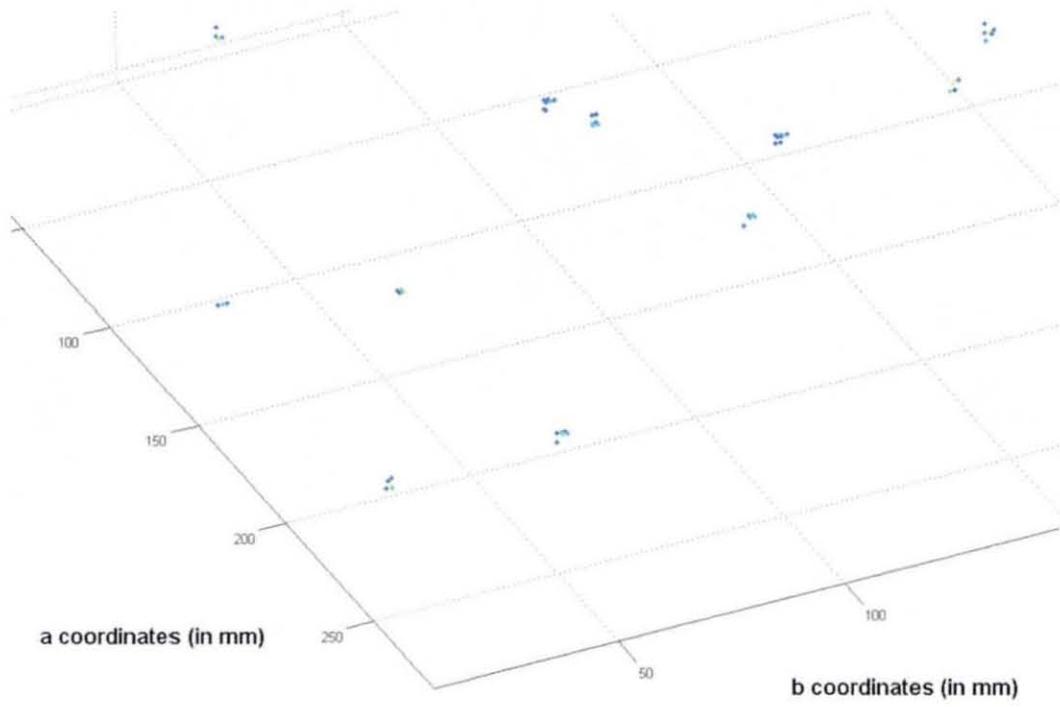
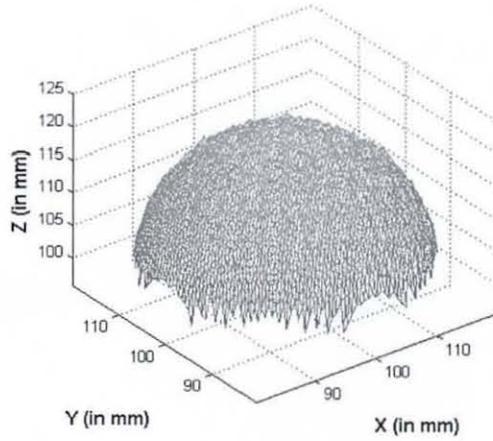
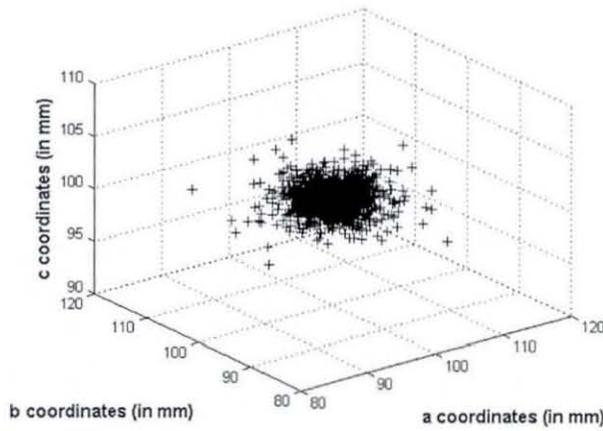


Figure 4-11 : Enlarged region of Hough space from Figure 4-10

(a)



(b)



(c)

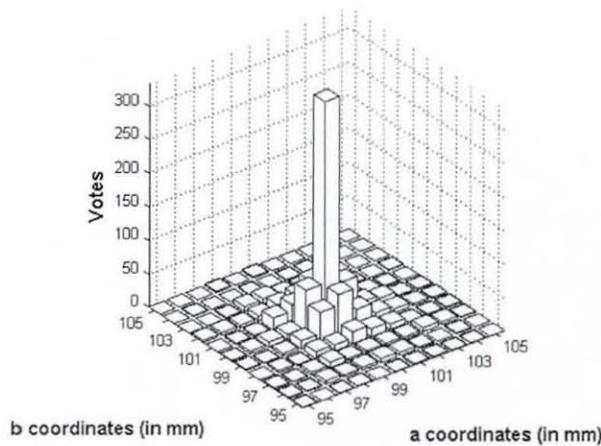
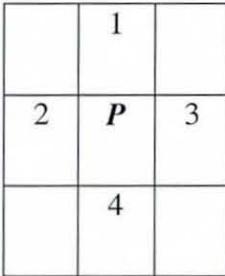


Figure 4-12 : Simulated hemi-sphere with radius of 20mm and the equivalent Hough space, with noise of standard deviations (σ) = 200 μ m introduced to respective x, y, and z coordinates. (a) 3-D plot of hemi-sphere; (b) 3-D plot of estimated sphere centre coordinates calculated using Eqn. (4-15); (c) Distribution of votes on a slice through Hough space (at $c = 100$ mm).

(a)



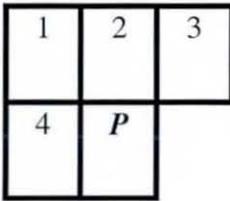
(b)



Figure 4-13 : Types of Connectivity for 2-D images.

(a) Four connectivity; (b) Eight connectivity

(a)



(b)

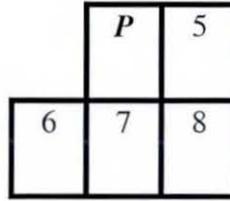


Figure4-14 : Types of mask for (a) Forward mask

(b) Backward mask

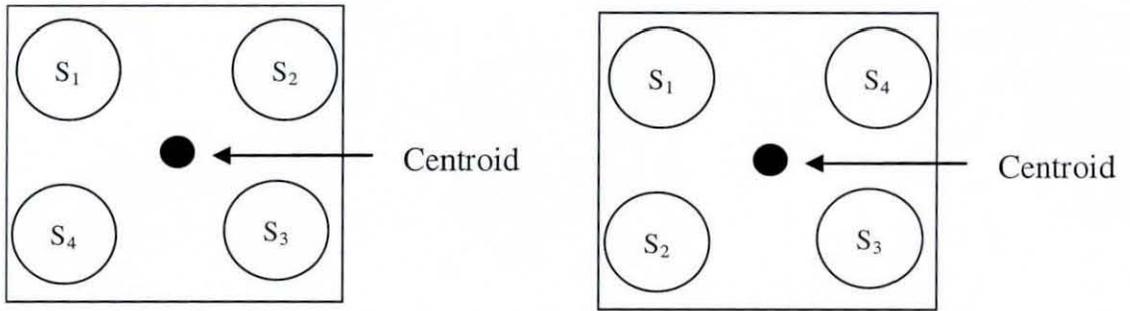


Figure 4-15: Identifying orientation of artefact through four furthest sphere centre coordinates which are located at the four corners of artefact.

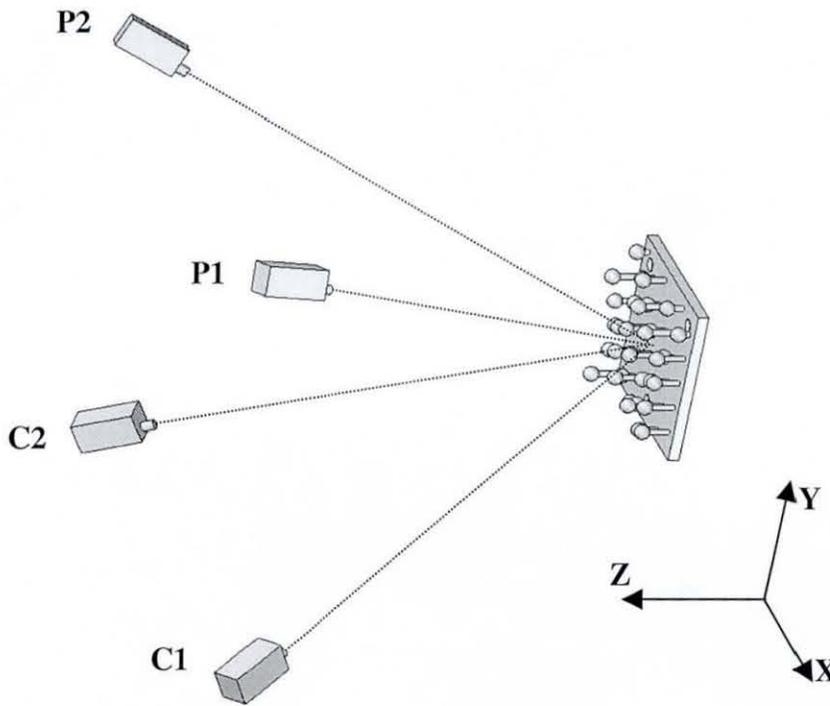
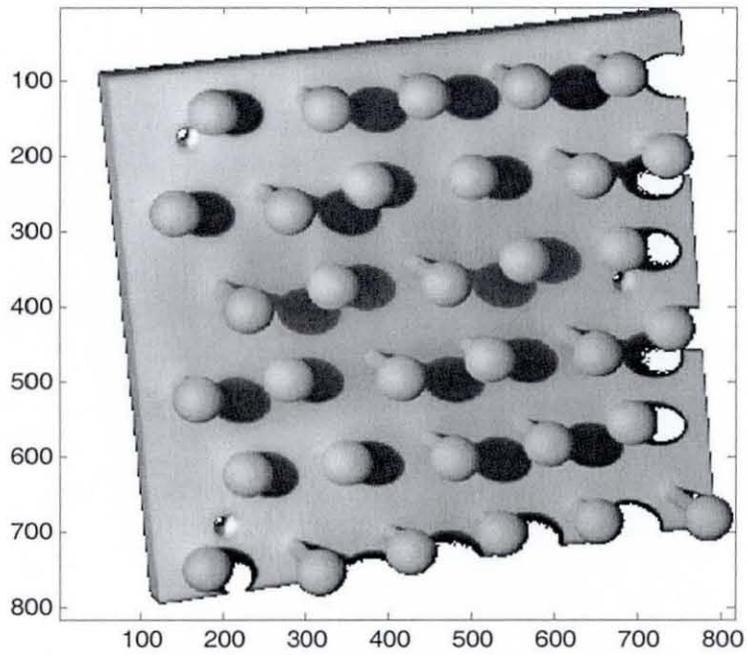


Figure 4-16 : Two-camera two-projector shape measurement system with calibration artefact.

(a)



(b)

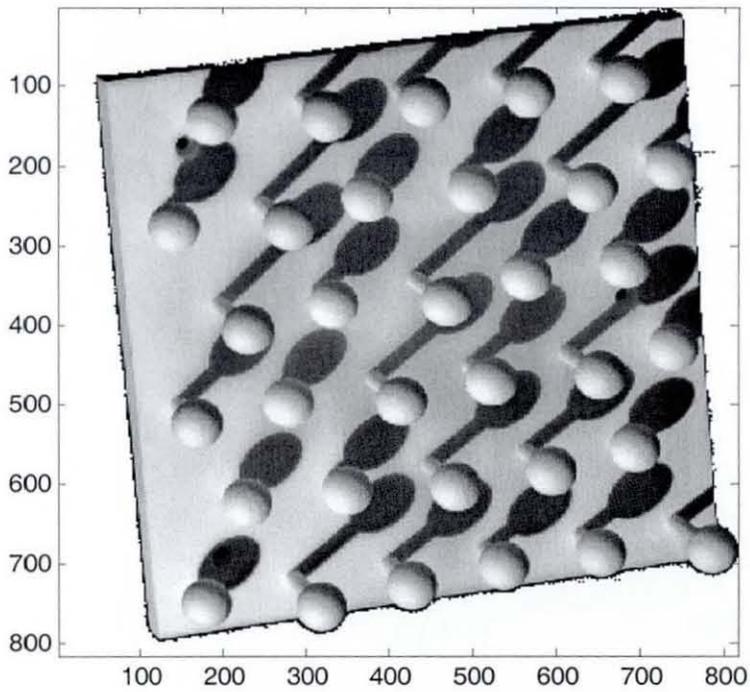
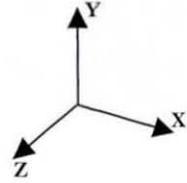
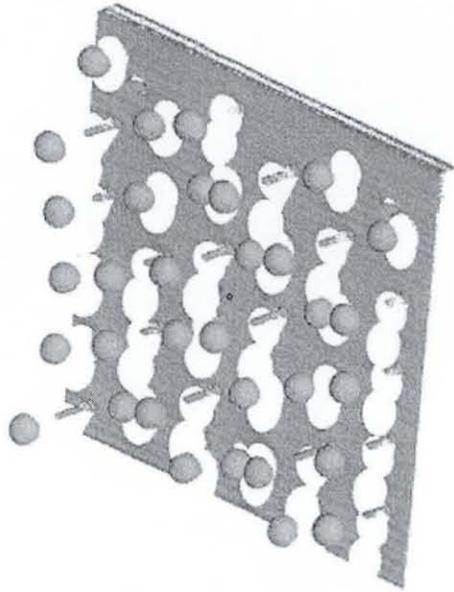


Figure 4-17 : Greyscale intensity images acquired using camera C1 : (a) Illuminated using projector P1; (b) Illuminated using projector P2. Scales on horizontal and vertical axes are in units of pixels.

(a)



(b)

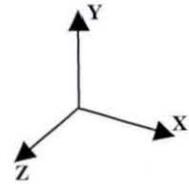
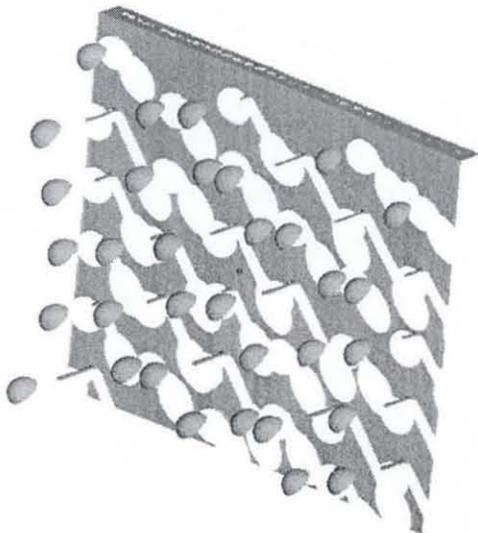


Figure 4-18 : Surface rendered point clouds from two camera-projector pairs with artificial lighting and smooth shading applied. (a) C1-P1; (b) C1-P2.

(a)



(b)

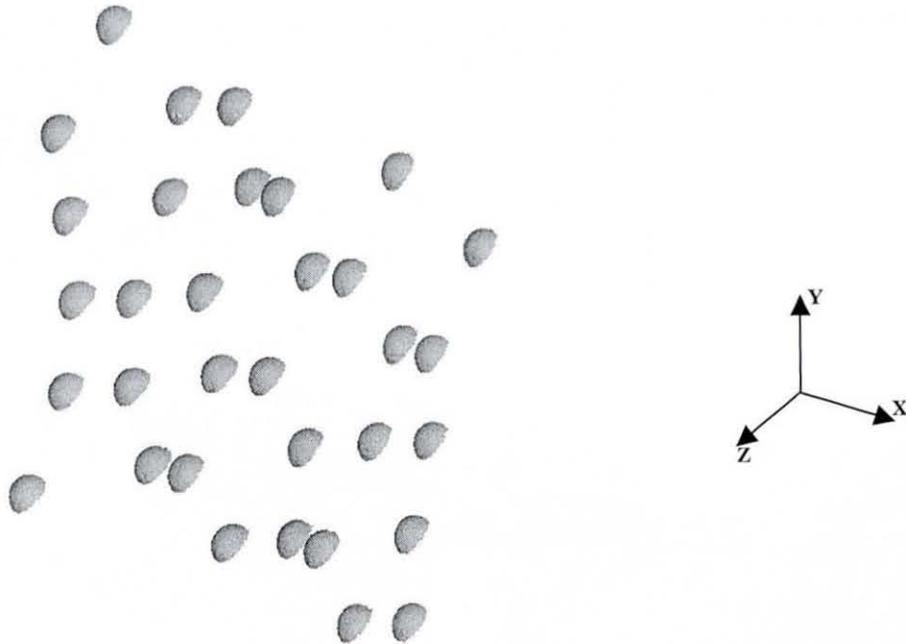


Figure 4-19 : Detected sphere coordinates from Figure 4-18. (a) C1-P1;
(b) C1-P2.

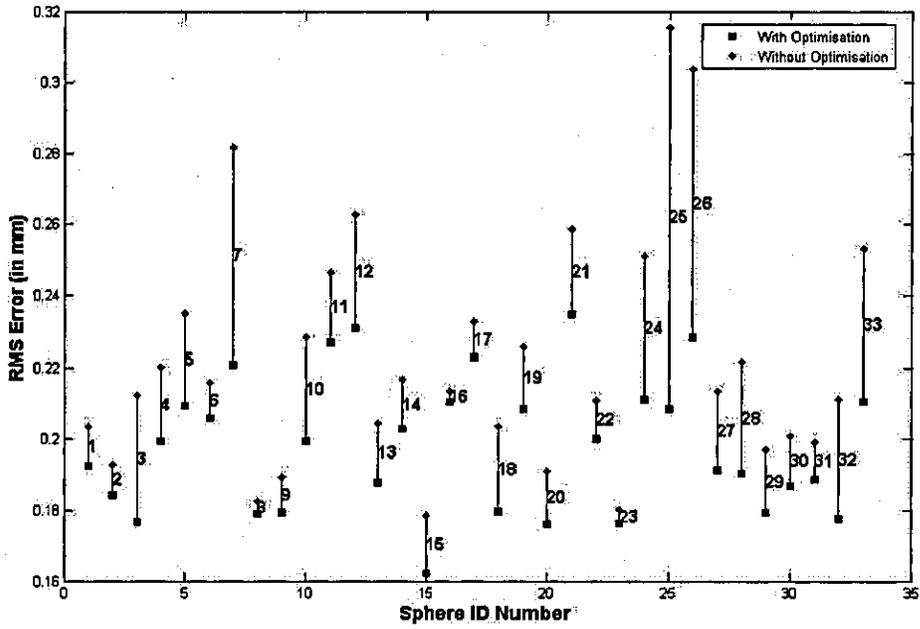


Figure 4-20 : Error before optimisation and after optimisation (C1-P1).

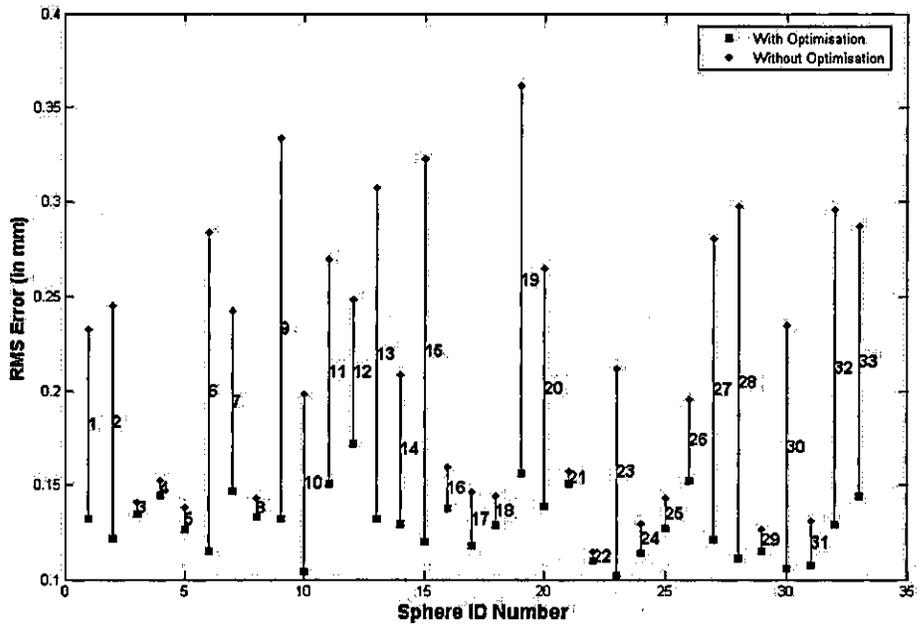


Figure 4-21 : Error before optimisation and after optimisation (C1-P2).

5 Plane Artefacts and Plane Detection

5.1 Introduction

The sphere calibration artefact and the method for automatically detecting sphere centres using a novel 3-D Hough transform (HT) were described in Chapter 4. Although spheres have the benefit of defining a point in space from almost all viewing directions, they have some drawbacks such as: (i) rapidly varying surface normal direction and hence back-scattered illumination intensity across the sphere's surface, which can introduce systematic measurement errors; (ii) cost of manufacture; (iii) cost of demonstrating traceable dimensions to high precision. Planes on the other hand have the benefit of a constant surface normal orientation across the artefact, and are easy to manufacture and validate to sub-wavelength accuracy using interferometric techniques. A length scale can in principle be introduced by making use of two parallel planes of known separation. Therefore, the use of planes as a calibration artefact for an optical shape measurement system (SMS) was investigated, and the results presented in this chapter.

A plane can be characterised by a 3-D point in space, \mathbf{P} , and a normal vector, \mathbf{n} , which defines its orientation in 3-D space. This implies ideally, that the perpendicular distance of \mathbf{P} to all 3-D points on the planar surface along the direction of \mathbf{n} , should be equal to zero. Although this would not hold true for a manufactured planar surface, this property provides a means of characterising its *flatness*, which is a measure of how planar the surface is. *Parallelism* on the other hand is a measure of how accurately two nominally-parallel planar surfaces are oriented.

In this chapter, discussions are focussed on the main issues considered in the process of selecting the constituent parts of the plane artefact in terms of cost, geometric properties and material properties. In addition, a detailed description of the adaptation of the 3-D HT (described in Chapter 4 for sphere detection) to plane detection is given. This includes how to establish an appropriate parameterisation for planes, and discussions on the performance of the method on simulated and experimental data. Part of the contents of this chapter forms the basis for a journal paper being prepared, which is a follow up to reference [23].

5.2 Artefact geometry

As in the case of the ball bar artefact described in Chapter 4, the main objective in the design of the plane artefact was to focus on the use of standard off-the-shelf precision made parts. It was therefore proposed that the artefact would be made of two parallel rectangular planes: a small upper one, and a second larger one to act as a base plate. To provide a calibrated length scale, the smaller plane would be offset by a fixed known distance away from the larger plane using a length bar. As highlighted in the case of the ball bar artefact, the use of a length bar allows different known distances between the two planar surfaces to be introduced, thus providing scalability of the plane artefact to different measurement volumes. By placing the smaller plane in front of the large plane, towards the extreme end of one of the shorter sides (see Figure 5-1), the occlusion of the larger plane is minimized. Such positioning should prove beneficial for calibrating multiple sensor SMS, where projector illumination is possible from multiple positions.

5.3 Artefact component parts

A search was carried out for off-the-shelf high precision planar parts from which the artefact could be constructed. A supplier may not have the required tooling to handle an order for a non-standard size and would need to invest in new tooling for a one-off part or a small order, thus increasing the cost of the part. Therefore, the use of off-the-shelf parts would reduce the unit cost of the artefact.

One popular planar reference artefact is the gauge block, which is normally rectangular and used for checking linear dimensional tolerances. The type of materials used for making gauge blocks give an indication of the type of materials that could be suitable for use as a reference artefact or working standard. The most common materials used for making gauge blocks are steel and ceramics such as zirconia and tungsten carbide. Zirconia interestingly has a white, diffuse finish, which seems suitable especially for an optical SMS. Other advantages include thermal and mechanical stability comparable to steel and resistance to corrosion. However, from discussions with potential suppliers ceramics are quite expensive to source, both in terms of raw material and manufacturing costs. One supplier, for example, Dynamic Ceramic, pointed out that a tonne of zirconia could cost as much as £189k. In addition, from the author's enquiries, the maximum commercially available off-the-

shelf plate was $100 \times 100 \times 5 \text{ mm}^3$, which was too small for the field of view being considered. On the other hand, although steel and aluminium plates are much cheaper to source, they often have a shiny surface finish. This can be controlled to some extent (either during measurement or through a further finishing process) to ensure that valid measurement points can be obtained off their surface. Therefore, as a consequence of potentially high cost of ceramics and commercial availability of only small sized plates, it was proposed that non-ceramic parts would be used.

A low velocity abrading finishing process, called *lapping* is often carried out on precision made parts in order to improve flatness and parallelism of the surfaces. Lapping, which is the process for machining surfaces to high dimensional accuracy, involves the removal of controlled, very small amounts of material [121]. It is accomplished with loose abrasive grains between a tolling plate or wheel (called the lap) and the work surface to be finished. Loose abrasives are used with a vehicle such as oil to remove material from the work piece at a very controlled rate. It is therefore a long pain-staking process that requires a considerable amount of time, and is consequently, an expensive process. Smooth lapped surfaces are generally non-reflective, whereas polished surfaces are highly reflective. Thus, lapping is normally carried out prior to polishing to achieve the following results [121]:

1. high dimensional accuracy
2. correction of minor imperfections of surface
3. refinement of surface finish
4. close fit between mating surfaces

Therefore, the key factors that affect the cost of planar parts are commercial availability, type of material and the required flatness and parallelism of surfaces.

5.4 Weight of artefact

As mentioned earlier in Chapter 2, it is desirable that a calibration artefact be thermally and mechanically stable. However, in order to ensure a versatile artefact, the planes should have a planar surface area consistent with the measurement volume. For example, for a $300 \times 200 \times 100 \text{ mm}^3$ measurement volume, the larger plane could be a $250 \times 160 \times 25 \text{ mm}^3$ plate. Though steel with relatively low thermal expansion coefficient ($11.3 \times 10^{-6} \text{ }^\circ\text{C}^{-1}$) and thermal conductivity ($0.048 \text{ W mm }^\circ\text{C}^{-1}$), and high modulus of elasticity (205 GPa) possesses the required thermal and mechanical

properties, it has a density of 7850 kg m^{-3} , and therefore, the plate would weigh 7.85 kg. However, this could prove quite challenging to handle manually. On the other hand, aluminium, which has higher thermal expansion coefficient ($22 \times 10^{-6} \text{ }^\circ\text{C}^{-1}$) and thermal conductivity ($0.16 \text{ W mm }^\circ\text{C}^{-1}$) and lower modulus of elasticity (70 GPa) in comparison with steel, has a lower density of 2700 kg m^{-3} . This implies that an aluminium plate would be lighter by almost a factor of 3 in comparison with a steel plate of the same size. A compromise then has to be sought with regard to the mechanical and thermal issues, as well as the usability of the artefact in service. It was therefore concluded that for small measurement volumes, steel plates would be utilised, while for larger measurement volumes, aluminium plates may be preferable.

5.5 High precision planar parts

5.5.1 Gauge blocks

As stated earlier, gauge blocks are rectangular shaped reference artefacts used for checking linear dimensional tolerances. The requirements of gauge blocks include [122]: the actual dimensions must be known, the faces must be parallel (to within a specified tolerance), the surface must have a smooth finish, and the surfaces must be flat (again, to within a specified tolerance). They are typically made to very high standards of accuracy in terms of surface finish, flatness and parallelism of faces e.g. $0.05 \text{ } \mu\text{m}$ (depending on length) and are available in five grades of accuracy, calibration, 00, 0, 1 and 2 [89]. The calibration, 00 and 1 grades are intended to be used for calibration at various levels, while Grade 1 and 2 are used for regular production and inspection measurements. However, the high accuracy is available on only two planar surfaces of the block, and the distance between these two surfaces defines the length which the block is required to gauge. In addition, from discussions with suppliers (see Appendix A-2), they are available in boxed sets containing a range of standard sizes, with the maximum standard size being $100 \times 35 \times 9 \text{ mm}^3$, and the largest possible surface area that can be measured from such a block is $100 \times 35 \text{ mm}^2$. This rather small surface area, therefore, makes them unsuitable for use as a plane artefact in this work.

5.5.2 Surface plates

Surface plates are commonly used as a support for measurement systems and provide a reference base from which parts are measured. They are typically made of cast iron

or granite with manufacturing quality usually described in terms of grade AA (high accuracy grade, for use in metrology rooms), grade A (lower accuracy grade, for use in quality control), and grade B (lowest accuracy grade, for use in the production shop floor). The top surface is made flat to high accuracy, with grade AA flat to $0.25\ \mu\text{m}$. From discussions with suppliers (see Appendix A-2), the surface colour range is restricted. Although some are available with a grey or pink colour, they are mostly only available in black. In addition, they are quite heavy (a $300\times 300\times 100\ \text{mm}^3$ granite surface plate weighs 23kg), which would be quite difficult to manipulate and handle. These issues make them unsuitable as a calibration artefact in this work.

5.5.3 Optical flats

Optical flats are finely polished flat surfaces used as a reference against which the flatness of a reflective test surface can be compared. They are usually circular and made from glasses or ceramics including fused quartz, silica, and zerodur (see Appendix A-2 for suppliers). Their flatness is described as a function of the wavelength of light (λ) with typical flatness values of $1/4\lambda$, $1/10\lambda$ and $1/20\lambda$. The fact that they are transparent implies that they may not be suitable as a calibration artefact for the optical SMS used in this work. However, some surface alteration operation, such as shot blasting and etching, or painting, could be carried out to make their surfaces opaque, although this would lead to a deterioration in their flatness. It was for this reason that the use of optical flats was not considered further, although they may be of interest for systems developed in the future.

5.5.4 Precision ground blocks

These are precision made parts for providing an accurate reference plane in machine tools. They have a rectangular shape, precision ground on all surfaces, with a number of holes drilled through. Some of these holes are also tapped. The standard sizes include – $25\times 50\times 75\ \text{mm}^3$, $1''\times 2''\times 3''$ and $2''\times 4''\times 6''$. Though made to high precision (from $0.15\ \mu\text{m}$ to within $2.5\ \mu\text{m}$ for a $25\times 50\times 75\ \text{mm}^3$ block), the largest size commercially available is approximately $50\times 100\times 150\ \text{mm}^3$. As a plane artefact, this would be quite restrictive in terms of lack of scalability to large measurement volumes. In addition, considering the number of holes that have already been drilled through the part, potential limitations of measurement and selection of control points off the surfaces are evident.

5.5.5 Precision ground plates

These are precision ground steel or aluminium plates, typically used in injection moulding machines. The standard sizes commercially available range from $160 \times 80 \times 25 \text{ mm}^3$ to $800 \times 630 \times 63 \text{ mm}^3$. In addition, they are available at relatively low cost. For these reasons, precision ground plates can be regarded as suitable components of the plane artefact. Therefore, $250 \times 160 \times 25 \text{ mm}^3$ and $160 \times 80 \times 25 \text{ mm}^3$ precision ground steel plates were purchased from Burrhart Machinery Ltd. The assembled artefact is shown in Figure 5-2. The supplier did not indicate a specific value for the flatness of the thickness surfaces of the plates, therefore, they were measured using a mechanical coordinate measuring machine (CMM). 32 positions were measured respectively on the two plates indicating flatness of $4.95 \text{ }\mu\text{m}$ for the smaller plate, while $6.41 \text{ }\mu\text{m}$ was measured as the flatness of the larger plate. In this case, the respective flatness value from the measured data sets is the maximum perpendicular distance from the best fit plane. With separation introduced using a 50.8 mm length bar, the measured top surfaces of the two plates were parallel to 0.014 degrees. It was envisaged that if an improvement is desired in the flatness and parallelism of the planar surfaces of the plates, they could be sent off to independent lapping service providers, which would consequently increase the cost of the artefact.

5.6 Selection of parts

Based on discussions with suppliers and comparison of different products, it was concluded that in this work, a cost effective, relatively accurate, stable, and user friendly plane artefact would consist of two precision ground plates made from steel and a length bar. With a combination of a set of length bars, a variety of separation distances between the two plates can be introduced to cater for different measurement volumes or calibration requirements.

5.7 Plane detection using the 3-D Hough transform

5.7.1 Finding Lines using the Hough Transform

The 2-D HT [91] for circle detection has been described in Chapter 4. The HT has historically been the main means of detecting lines and has been further developed and refined for this purpose [123]. In this section, the parameterisation for line detection will be introduced, and then extended for plane detection.

A point in space can be defined in one of two ways [123]: either a pair of coordinates or a set of lines that pass through that point. In addition, a set of collinear points, having a set of lines passing through each of them, has only one line which is common to all points. In the parameterised form, the equation of a line is:

$$y_i = mx_i + c \tag{5-1}$$

where x_i and y_i are the usual Cartesian coordinates of the edge points, m is the slope, and c is the intercept.

The above equation has two parameters, (m,c) , and ideally, every edge point $[x_i,y_i]$ in image space corresponds to a line in (m,c) Hough space, thus leading to multiple votes in Hough space. However, efficient methods have been introduced which make use of edge gradient information such that each edge point contributes only a single vote in Hough space. One of these such methods, the 'foot-of-normal' method [95, 123, 124], has been used in this work and will be discussed in further detail.

With available edge gradient information, an infinite line, L , with the same direction in Hough space, can be constructed for each edge. Starting from an arbitrary origin, a normal, \mathbf{n} , will intersect L at a point (x_0,y_0) , which is called the *foot-of-normal position* (FON). From the FON, two parameters, (θ, ρ) which represent the direction and magnitude of the normal can be defined. Therefore, in Hough space, votes are accumulated at bin coordinates that correspond to FON positions. For a given edge (x_i,y_i) with gradient (g_x, g_y) , and the FON, (x_0,y_0) , it is found that

$$\frac{g_y}{g_x} = \frac{y_0}{x_0} \tag{5-2}$$

$$(x - x_0)x_0 + (y - y_0)y_0 = 0 \tag{5-3}$$

Eqns. (5-2) and (5-3) can be used to solve for (x_0,y_0) thus

$$x_0 = vg_x \tag{5-4}$$

$$y_0 = vg_y \tag{5-5}$$

where

$$v = \frac{xg_x + yg_y}{g_x^2 + g_y^2} \tag{5-6}$$

Errors in the estimation of the FON are dependent on its distance from origin, therefore to minimise errors, the centre of the image is normally set as the arbitrary origin [123].

5.7.2 Parameterization for Plane Detection

Sarti and Tubaro [125] describe an extension of the HT for line detection to the 3-D domain in the development of an iterative technique for detecting and characterising planar rock fractures. The parameterisation is based on the equation of a plane

$$ax + by + cz = 1 \tag{5-7}$$

The above equation can also be expressed as

$$x \cos \alpha + y \cos \beta + z \cos \gamma = d \tag{5-8}$$

where α, β, γ are the angles between x -, y -, z - axis, respectively, and d is the distance from the plane to the origin.

The parameters of the plane can be evaluated thus

$$\begin{aligned} a &= -\frac{\cos \alpha}{d}, \\ b &= -\frac{\cos \beta}{d}, \\ c &= -\frac{\sqrt{1 - \cos^2 \alpha - \cos^2 \beta}}{d}. \end{aligned} \tag{5-9}$$

Considering all possible pairs of α, β values, corresponding values for d can therefore be computed and used for voting in the 3-parameter (α, β, d) Hough space.

This iterative technique involves the progressive removal of cumulative layers of the Hough space, whereby at every iteration, the leading planar fracture (i.e. the plane whose bins have the largest votes) is detected and subsequently removed from the Hough space, such that the next iteration is used to detect the next most dominant plane. However, the method was developed for binary 3-D datasets without gradient information, thus the process of voting in the 3-D Hough space is computationally intensive.

In this section, the extension of the FON parameterisation to plane detection is described. The FON parameterisation described by Warr et al [95] is easily extendable to planes and results in a 3-D voting space for the FON coordinates. For a plane, P , with a normal, \mathbf{n} (that starts from an arbitrary origin), the FON can be described as the 3-D Cartesian coordinate (x_0, y_0, z_0) of the intersection of \mathbf{n} and P (see Figure 5-3). Equations (5-2) to (5-6) can be extended for planes as follows:

$$(x_i - x_0)x_0 + (y_i - y_0)y_0 + (z_i - z_0)z_0 = 0, \quad (5-10)$$

where x_i, y_i, z_i are the 3-D coordinates on a planar surface.

We also have that

$$x_0 = \frac{n_{y,i}}{n_{x,i}} x_i, \quad z_0 = \frac{n_{z,i}}{n_{x,i}} x_i, \quad (5-11)$$

where $\mathbf{n}_i = [n_{x,i}, n_{y,i}, n_{z,i}]^T$ is the unit inward -pointing surface normal estimated at each measured point x_i, y_i, z_i . The normal can be estimated, for example, using a least squares fit of a plane to the coordinates of the point and its nearest neighbours.

Solving Eqns. (5-10) and (5-11), we therefore have the respective estimates for the FON from all i coordinates on the plane as follows:

$$\mathbf{E}_i = \begin{pmatrix} x_{0,i} \\ y_{0,i} \\ z_{0,i} \end{pmatrix} = v_i \begin{pmatrix} n_{x,i} \\ n_{y,i} \\ n_{z,i} \end{pmatrix} \quad (5-12)$$

where

$$v_i = \frac{x_i n_{x,i} + y_i n_{y,i} + z_i n_{z,i}}{\|\mathbf{n}_i\|} \quad (5-13)$$

However, since $\|\mathbf{n}_i\| = 1$, then

$$v_i = x_i n_{x,i} + y_i n_{y,i} + z_i n_{z,i} \quad (5-14)$$

Therefore, v_i is a measure of the perpendicular distance from the origin to the i^{th} point on the plane characterised by a FON (x_0, y_0, z_0) . Thus, for a perfectly flat plane, v_i should be the same for all points on the plane.

5.7.3 Voting in 3-D Hough space and parameter estimation

Voting in the 3-D Hough space has already been discussed in Chapter 4, therefore only issues specific to plane detection will be discussed in this section. Unlike sphere detection (as described in Chapter 4), where the radius is supplied to the detection algorithm, for plane detection, *a priori* knowledge of any of its parameters is not required. Using Eqn. (5-12), the estimated FON calculated for each 3-D coordinate, \mathbf{E}_i , is rounded to integers which define the Hough space coordinate bins in which the vote will be placed. The accuracy of estimates of the surface normals of surface facets of the plane directly affect the magnitude and direction of the position vector, \mathbf{P}_v , defining the location of coordinate bins in Hough space. In comparison with the case of spheres where the magnitude of \mathbf{P}_v is the known radius of the sphere, for planes, the magnitude of \mathbf{P}_v is affected by the accuracy of surface normals, and as a result of noise, there would typically be a wider spread of votes in Hough space. This implies that the accuracy of estimated surface normals is crucial to the accuracy of the FON, \mathbf{F}_k for each k^{th} plane in a point cloud. In Chapter 4, increasing the size of Hough space bins was shown to be a solution to problems associated with noisy 3-D datasets. As surface normals are estimated by least squares fitting of a plane to the coordinates of the point and its nearest neighbours, another solution could be to increase the kernel size used for this process to say, a 9×9 kernel. On calculating \mathbf{E}_i , followed by

binning and voting in Hough space, peak detection and weighted averaging (which have been described in Chapter 4), we can compute an estimate of F_k .

An optional post-processing step could be carried out as a means of improving F_k . The process involves identifying all the points that belong to the k^{th} plane characterised by F_k , and least squares fitting of the identified points to obtain the best fit plane coefficients. Using F_k , a normal for the k^{th} plane can be computed by normalising F_k :

$$\mathbf{n}_k = \frac{\mathbf{F}_k}{\|\mathbf{F}_k\|} \quad (5-15)$$

\mathbf{n}_k is therefore an improved estimate of the surface normal for all points that lie on the surface of the k^{th} plane.

The equation of a plane can be written thus

$$\mathbf{r}_i \mathbf{n}_k = d_i \quad (5-16)$$

where \mathbf{r}_i is the 3-D coordinate, and d_i is the perpendicular distance to the plane defined by F_k .

It is expected that all q points which lie on the k^{th} plane characterised by F_k should have values for d which would be very small in comparison with other points in the point cloud. Therefore, a threshold, t_p , is set such that points with absolute d_i values below t_p (i.e all the coordinates that meet the condition, $(d_i \leq t_p)$, where $i = [1 \dots q]$) are considered to be points that lie on the k^{th} plane. Once all the 3-D coordinates that lie on the plane characterised by F_k have been identified, through least squares fitting, the perpendicular distances of each of the individual points to the best fit plane can be minimised. This approach is referred to as *orthogonal distance regression*, where we seek to find the coefficients a_1, a_2, a_3, a_4 , that minimises the following equation

$$f(a_1, a_2, a_3, a_4) = \frac{\sum |a_1 x_i + a_2 y_i + a_3 z_i + a_4|^2}{a_1^2 + a_2^2 + a_3^2} \quad (5-17)$$

Finding the orthogonal distance regression plane is an eigenvector problem and the best solution utilises singular value decomposition (SVD) [126].

Applying SVD to the constraint matrix B , (where $[B] = \begin{bmatrix} x_1 & y_1 & z_1 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & z_n & 1 \end{bmatrix}$) gives

$$[U, D, W] = \text{SVD}(B) \tag{5-18}$$

The fourth column of W contains the best fit coefficients, a_1, a_2, a_3 , and a_4 . From the best fit coefficients, we can compute a plane's best fit normal vector,

$$\mathbf{n}_f = \frac{[abc]}{\|[abc]\|} \tag{5-19}$$

In this case, it is automatically assumed that the least squares plane contains the centroid, \mathbf{r}_c , of the data-set [126]. Having identified the points that lie on the k^{th} plane using Eqn. (5-16) and applying a threshold, t_p , Eqn. (5-12) (in this case, the i^{th} point would only refer to a point that lies on the plane) is used to estimate \mathbf{r}_c as an average of (x_i, y_i, z_i) thus,

$$\mathbf{r}_c = \begin{bmatrix} \frac{\sum_{j=1}^q x_j}{q} \\ \frac{\sum_{j=1}^q y_j}{q} \\ \frac{\sum_{j=1}^q z_j}{q} \end{bmatrix} \tag{5-20}$$

The equation of the plane is given by

$$\mathbf{r}_c \mathbf{n}_f = d \tag{5-21}$$

where d in this case is the perpendicular distance from the origin to the plane. However, since \mathbf{F}_k also lies on this plane, it also satisfies the above equation.

$$\mathbf{F}_k \mathbf{n}_f = d \quad (5-22)$$

With the improved estimate of the plane's surface normal, \mathbf{n}_f , and using Eqns. (5-21) and (5-22), we can show therefore that the optimal FON, \mathbf{F}_k' , which is an improved estimate for \mathbf{F}_k is

$$\mathbf{F}_k' = (\mathbf{r}_c \mathbf{n}_f) \mathbf{n}_f \quad (5-23)$$

5.7.4 Voting in 1-D Hough space

Consider a situation for the 3-D HT implementation, where there is a small difference in the size of the planes in the point cloud. In Hough space we would expect small differences in the largest vote in the respective clusters which belong to each plane. Therefore, if the initial value of t_v is too severe and leads to the detection of less than the required number of planes in the point cloud, reducing the value of t_v could end up being too conservative, leading to the acceptance of false planes as valid. Thus for the 3-D HT, it is a challenge to determine an appropriate initial value for t_v .

For the special case where all the planes in the measurement volume are parallel to one another, Eqn. (5-14) provides a means of using a 1-D HT as an alternative to the 3-D HT. In such a case, any two distinct planes can be guaranteed to have two distinct v_i values. However, where the planes can have arbitrary orientation, it is possible for two independent planes to have the same perpendicular distance from the origin. In such a case, the peaks in the 1-D HT belonging to the two planes would overlap and therefore this technique would not work. In this work, where the two plane calibration artefact has parallel planes, however, makes such a concept an interesting alternative to the full 3-D HT. Thus, the simplicity of the 1-D HT is expected to make it a more efficient alternative to the 3-D version.

A 1-D histogram of v_i is carried out from which the bin, v_m , containing the maximum vote is used to determine the predominant value of v . A threshold, t_v , can be set such that all the v_i values that are close to v_m (i.e. that meet the condition $(v_m - t_v) \leq v_i \leq (v_m + t_v)$), are considered to be points that belong to the same plane. This criterion can therefore be used to identify all the points that lie on respective planes. Consequently, the accuracy of the 1-D approach is dependent on the

successful identification of as many points as possible that belong to each plane in the point cloud. These q points are then used in Eqn (5-12) to obtain the first estimate for the FON of the k^{th} plane, $F_{k,1}$. Therefore, the parameters that affect detection capability include the number of bins for the 1-D Hough space, and the value for the threshold, t_v .

In practice, as a result of noise in the point cloud and the spatial constraint of a 1-D voting space, there is no guarantee that sufficient points would be identified to allow an accurate estimate of the plane parameters. Therefore, further applications of the 1-D HT are carried out as a means of enhancing the reliability of the plane detection process. Starting from $F_{k,1}$, another 1-D HT is carried out. However, rather than computing a surface normal for each i^{th} coordinate, a better estimate of the normal of the k^{th} plane, $n_{k,2}$, is computed using $F_{k,1}$ in Eqn. (5-15) and Eqn. (5-14), to obtain improved $v_{i,2}$ values. For a set of parallel planes, $n_{k,2}$ could be considered as an improved estimate of the normal of all the planes. This would therefore ensure that each coordinate in the point cloud votes in the most appropriate bin in the 1-D Hough space. On completion of this second pass of the 1-D HT, at this stage, it is expected that for $v_{i,2}$, when the same threshold, t_v , is applied, a very significant number of points on the k^{th} plane would be identified. Finally, using Eqns. (5-18) to (5-23), the least squares fitting process described earlier is used to determine the best fit plane coefficients, and subsequently, the FON, $F_{k,2}$. An alternative for computing $F_{k,2}$ using Eqn. (5-23) is

$$F_{k,2} = \sum_{i=1}^q (r_i n_{k,2}) n_{k,2} \quad (5-24)$$

where $r_i = (x_i \ y_i \ z_i)^T$ is the i^{th} point on the plane

It is necessary to highlight that the procedure described above is required for detecting each plane. Thus, once all the 3-D coordinates on a plane have been identified, they are masked out to ensure that only the 3-D coordinates of undetected planes are processed. The method is therefore iterative, whereby, at each iteration, the plane having the highest votes in the 1-D Hough space is identified. The total detection time of the 1-D HT is therefore directly proportional to the required number of planes to be detected in the point cloud.

5.8 Results of 3D Hough Transform for Planes

5.8.1 Introduction

It has been shown in Chapter 4 that the bin size and the global votes threshold affect the accuracy of parameter estimation in Hough space. Discussions in this section focus on the HT's plane detection capability when applied to simulated and experimental 3-D datasets of planes. The datasets were processed on a computer with a Pentium4 processor (2.80GHz, 1GB RAM).

5.8.2 Simulated data

Consider a planar surface of $200 \times 200 \text{ mm}^2$ (see Figure 5-4) consisting of 200 points, with the centroid of the plane assigned the coordinate, $r_{c,x} = 200 \text{ mm}$, $r_{c,y} = 200 \text{ mm}$, $r_{c,z} = 200 \text{ mm}$, and the FON $x_0 = 0 \text{ mm}$, $y_0 = 0 \text{ mm}$, $z_0 = 200 \text{ mm}$. Random numbers are introduced to the respective x , y , z coordinates with known standard deviations. For a dataset with $\sigma = 200 \text{ }\mu\text{m}$ and Hough space bin size of $1 \times 1 \times 1 \text{ mm}^3$, the resulting scatter in the (x_0, y_0, z_0) coordinates, computed from Eqn. (5-12) using a 3×3 kernel to estimate the surface normals, is apparent in Figure 5-5(a). The distribution of votes on a slice through Hough space ($z_0 = 199 \text{ mm}$) is shown in Figure 5-5 (b). Figures 5-5(c) and (d) show the histogram of v_i values during the 1st and 2nd iterations of the 1-D HT. The smaller range of v_i values in the 2nd iteration, indicate an improvement in the plane's surface normal as calculated from the FON estimated from the 1st iteration.

As mentioned in Section 5.7.3, the estimated surface normals calculated from least squares fitting of facets on the plane affect the position and magnitude of the estimated FON in Hough space. Therefore, the magnitude of vote spread in Hough space is strongly dependent on the accuracy of the estimated FON. In Table 5-1, the effect of varying noise levels at different σ values ($5 \text{ }\mu\text{m}$, $10 \text{ }\mu\text{m}$, $20 \text{ }\mu\text{m}$, $50 \text{ }\mu\text{m}$, $100 \text{ }\mu\text{m}$ and $200 \text{ }\mu\text{m}$) are compared, when a 3×3 kernel is used in estimating surface normals of the respective point clouds, with the Hough space bin size fixed at $1 \times 1 \times 1 \text{ mm}^3$. The quantity, W , which is a measure of spread of votes in Hough space has been introduced in Chapter 4. It can be observed that the value of W increases as the noise level is increased, and also, the W values are significantly large when compared with those obtained from spheres (as shown in Chapter 4). Therefore, an effective means of handling coordinate clusters will be required, and most especially, some practical means of achieving significant reductions in W for each plane in the point cloud. It is

worth mentioning, however, that the drawback of the FON parameterisation is that errors in the FON position depend on the distance from origin [123]. Thus, the value of W would increase as the plane is moved further from the origin. This is demonstrated in Table 5-2 which shows how W changes when the centroid of the plane is simply translated along the X axis.

On the other hand, for $\sigma = 200 \mu\text{m}$, and the bin size fixed at $1 \times 1 \times 1 \text{ mm}^3$, it can be seen from Table 5-3 that as the kernel size is increased, the number of bins along each axis of the Hough space decreases, and consequently leads to a decrease in the number of coordinate bins with at least one vote and smaller W values. This implies that by simply changing the size of the convolution kernel used for computing surface normals, we can compensate for noise in the point cloud and thus improve on the accuracy of the normals. Therefore, the size of the kernel used in estimating the surface normals influences the number of bins in Hough space i.e. its resolution, and the magnitude of vote spread. Also, the memory savings of using the optimised sparse matrix reduce as the resolution of the Hough space becomes more coarse. In Table 5-4 it is shown for $\sigma = 200 \mu\text{m}$ (with a 5×5 kernel for estimating surface normals) and varying the bin sizes ($1 \times 1 \times 1 \text{ mm}^3$, $2 \times 2 \times 2 \text{ mm}^3$, $4 \times 4 \times 4 \text{ mm}^3$, $8 \times 8 \times 8 \text{ mm}^3$, and $16 \times 16 \times 16 \text{ mm}^3$), that further post-processing of the detected FON through least squares fitting, leads to a significant improvement in detection accuracy (i.e. the Euclidean distance between the estimated FON and the true FON), with total detection time typically less than half a second. In this case, the detection accuracy of the 1-D HT is comparable with the 3-D HT, and time savings are not significantly different.

Applying a change to the resolution of the 3-D Hough space (either as a result of changing bin size or convolution kernel size) implies that the global votes threshold, t_v , would need to be reset in order to accurately estimate the FON. However, setting a generous value for the distance threshold, t_p , for example, 3 mm, (i.e. the allowable perpendicular distance from any point on the plane to the detected FON) to select points on the planar surface which would be used for least squares fitting, ensures that a significant number of points are identified, thus improving accuracy of the FON.

Therefore, once an initial value is set for t_v , if the required number of planes is not detected, t_v can be automatically increased or decreased by some preset factor.

The method was also tested on simulated shape data of multiple planes (as shown in Figure 5-6) for comparison of the performance of the 3-D HT and the 1-D HT in terms of detection capability and computational efficiency. It should be noted in order to improve on the accuracy of the FON, the detection process in these tests included least squares fitting of points on the plane in order to compute the best fit coefficients of the detected planes. Five planes were simulated and arranged in a step like fashion (with a step height of 10 mm), with each plane containing 200 coordinates and noise of $\sigma = 200 \mu\text{m}$. When a 3×3 kernel is used to compute surface normals for this point cloud of multiple planes, from the 3-D Hough space (with bin size of $1 \times 1 \times 1 \text{ mm}^3$) shown in Figure 5-7(a), it can be seen that it is quite difficult to distinguish between clusters of bins containing votes contributed from respective planes. Thus, for the 3-D HT, noisy multiple plane datasets potentially lead to peak detection and consequently, efficiency problems.

Indeed, in this case, identifying the most appropriate combination of parameters i.e. bin size, convolution kernel size, and global votes threshold, required a significant number of attempts. However, it was observed that the most influential parameter was the size of the convolution kernel for computing the surface normals, therefore, a rather large kernel (19×19) was used, leading to improved estimates for the FON (see Figure 5-7(b) for the 3-D Hough space) and a reduction in the number of 3-D Hough space bins to $20 \times 7 \times 60$ bins. Consequently, the bin size was set as $1 \times 1 \times 1 \text{ mm}^3$ and t_v set at 500. With this approach, the 3-D HT was able to successfully detect all the five planes. In practise, it is therefore suggested that for noisy datasets, a large convolution kernel (greater than 3×3) should be used in computing surface normals.

On the other hand, the 1-D HT performed satisfactorily even when a 3×3 kernel is used for computing surface normals, detecting all the planes with less effort in determining appropriate values for the number of 1-D Hough space bins, t_v or t_p . Figures 5-8 (a) and (b) show the 1-D Hough space during the 2 iterations of the 1-D HT, when a 3×3 kernel is used for computing surface normals, while 5-8 (c) and (d)

show the 1-D Hough space when a 19×19 kernel is used. It can be seen that for both kernel sizes, at the second iteration of the 1-D HT, we are able to distinguish between the peaks that belong to each plane. Although for this dataset the 3-D HT can also be utilised in the same mode as the 1-D HT (i.e. multiple runs of the HT whereby in each run, the bin coordinate with maximum votes is selected, thus detecting the nominally parallel planes one-by-one), however, in this mode, it is not as robust and computationally efficient as the 1-D HT.

Table 5-5 shows the detection accuracy of the 3-D HT and 1-D HT when used on the multiple planes point cloud dataset. As mentioned earlier, errors in the FON position increase with distance from the origin, thus it is observed that the detection accuracy decreases from the lowest step to the topmost step. However, the two methods demonstrate their ability to identify all the points belonging to respective planes in the point cloud.

5.8.3 Experimental data

The 3-D and 1-D HT methods were also tested on experimental shape data acquired with a two-camera two-projector SMS. It should be noted that for each camera-projector pair, a 3-D coordinate is computed for a camera pixel containing a valid pair of phase values, therefore, once all the 3-D coordinates have been computed and the coordinates that lie on a plane have been detected, their corresponding pixel locations can easily be identified. Thus the set of pixels belonging to the respective planes can be selected and labelled. Also, in these tests, in order to improve on the accuracy of the FON, the detection process included computing the best fit coefficients of the detected planes. Therefore, the detection process involves pre-processing of point cloud data (e.g. computing of surface normals), applying the 3-D HT or 1-D HT on the point cloud to obtain the FON for each plane (i.e. voting and peak detection) and post-processing of the FON to improve accuracy (i.e. using identified points on each detected plane to compute the plane's best fit coefficients in a non-linear optimisation). Results relating to one of the camera-projector pairs, camera-1 projector-1 (C_1P_1) will be discussed.

Firstly, the methods were tested on a point cloud of the two-plane calibration artefact. The 3-D HT and 1-D HT were able to successfully identify the two respective planes.

The detected planes from the point cloud from C_1P_1 are shown in Figure 5-9, with the first plane consisting of $n_p = 191,196$ 3-D coordinates and for the second plane, $n_p = 80,722$. For the 3-D HT, the Hough space bin size was set as $1 \times 1 \times 1 \text{ mm}^3$, convolution kernel as 19×19 , and votes threshold, t_v , set at 900. Figures 5-10 (a) and (b) show the 3-D Hough space for the data set before and after the application of t_v . The 3-D HT was completed in less than 2 s. However, to this figure should be added typically 6 s for pre-processing steps such as computation of the surface normals, and 4 s for post-processing steps such as non-linear optimisation of the estimated FON. The rms of the perpendicular distance, d , between points on the detected planes and their respective FONs is $133.3 \text{ }\mu\text{m}$ for plane 1 and $398.4 \text{ }\mu\text{m}$ for plane 2. Figures 5-11 (a) and (b) show histograms of d for the two planes. The residual discrepancies reflect a combination of measurement error, and deviations from flatness in the manufactured artefact.

For the 1-D HT, the number of bins was set at 1000, while the distance threshold, t_p , was set at 2 mm. Figures 5-12 (a) and (b) show the 1-D Hough space after the 1st and 2nd iterations of the 1-D HT. In this case, the 1-D HT was completed in approximately 2 s, with about 6 s for pre-processing and 4 s for post-processing. The rms of d is $133.3 \text{ }\mu\text{m}$ for plane 1 (which is the same as that of the 3-D HT) and for plane 2, $388.8 \text{ }\mu\text{m}$ (which is slightly different from that of the 3-D HT). In Figures 5-13 (a) and (b), histograms of d for the two planes are shown, which reveal a distribution similar to that obtained by the 3-D HT.

Secondly, the 3-D HT and 1-D HT methods were tested on a point cloud of a four-step block and were able to successfully identify the four respective planes for all camera-projector pairs. In the results of C_1P_1 (see Figure 5-14), the number of points, n_p , on each detected plane were 22430, 23085, 20881 and 19321 respectively. For the 3-D HT, bin size was $2 \times 2 \times 2 \text{ mm}^3$, convolution kernel set as 19×19 and t_v set at 450. The settings for the 1-D HT was similar to that used for detecting the two-plane artefact. Finally, the detection time of the 1-D HT was approximately 2 s, while that of the 3-D HT was approximately 2 s.

As mentioned earlier, for a noisy point cloud of planes, the HT suffers from peak detection problems which may require changing a combination of correlated

parameters in order to detect all the planes. Thus for a multi-sensor optical SMS where multiple point clouds would need to be processed, once an appropriate initial value has been set for the bin size, the kernel size for computing surface normals, and t_v , if the 3-D HT or 1-D HT fails to detect all the required number of planes for a particular dataset, we can simply automatically increase or decrease t_v while keeping other parameters fixed. However, although the 1-D HT is unreliable for detecting planes of arbitrary orientation, it has been shown to be computationally efficient for detecting nominally parallel planes. Therefore, as a result of its simplicity, robustness to noise, proven computational efficiency and detection capability for nominally parallel planes (e.g. the two-plane calibration artefact), it was concluded that the 1-D HT should be used for automated plane detection in the calibration process of the multi-sensor optical SMS. When the 1-D HT method is used in a calibration, another parameter that could be included to improve on detection reliability, which could be used as a basis for changing t_v , is a tolerance (based on the known distance between the planes) within which the perpendicular distance between the detected plane FONs should be.

5.9 Summary

A calibration artefact consisting of two planes was investigated as an alternative to the sphere calibration artefact (described in Chapter 3), to provide a low-cost solution for small measurement volumes through the use of standard off-the-shelf precision made parts. It was concluded that the artefact should be made of two parallel rectangular planes (a small one and larger one) and a length bar, which would provide a fixed, known calibrated length scale. A number of possible off-the-shelf precision made planar parts were investigated, namely gauge blocks, surface plates, optical flats, precision ground blocks and precision ground plates. Precision ground plates are commercially available as steel or aluminium in a variety of standard sizes and are relatively low cost. Therefore, the plane artefact developed consists of two steel precision ground plates ($250 \times 160 \times 25 \text{ mm}^3$ and $160 \times 80 \times 25 \text{ mm}^3$), and a length bar. Though improved accuracy of the flatness and parallelism of the planar surfaces of the plates through lapping is possible, however, it is a costly process. Therefore, it is suggested that lapping should be carried out on the plates when a high accuracy calibration is required

The methods which were developed for detecting planes in a point cloud are an extension of the 2-D Hough transform for line detection, based on the foot-of-normal parameterisation. A parameterisation for the foot-of-normal position of planes was established and applied using the optimised 3-D Hough transform (described in Chapter 4). A procedure was also developed for improving the estimated foot-of-normal position through least squares fitting to obtain the best fit coefficients of the plane. In addition, a 1-D version of the 3-D HT (i.e. voting in a 1-D Hough space) was developed specifically for detecting nominally parallel planes.

The 3-D HT and 1-D HT were respectively tested on simulated and experimental datasets on a computer with Pentium4 processor (2.80GHz, 1GB RAM). The simulated data consisted of single and multiple planes respectively, where the 3-D HT and 1-D HT successfully detected all planes in approximately 2 s. The experimental datasets consisted of multiple planes, and detection time was typically 2 s for the 3-D HT and 1-D HT respectively. However, pre-processing steps such as computation of the surface normals could take up 6 s, while post-processing steps such as non-linear optimisation of the estimated FON could take up to 4 s.

It was observed that in noisy datasets, the 3-D HT suffers from peak detection problems, which lead to significant amount of user input, through multiple attempts at respectively setting its bin size, size of convolution kernel (for computing surface normals) and global votes threshold, t_v . However, it was observed that the kernel size used for computing surface normals could be used to compensate for noisy datasets, thus ensuring a more coarse Hough space and consequently, stronger peaks for the true FON. On the other hand, the performance of the 1-D HT is mainly dependent on kernel size for computing surface normals, number of bins in the 1-D Hough space, t_v , and distance threshold, t_p (for identifying points belonging to respective planes). Thus, the simplicity of the method ensured that plane detection required significantly fewer attempts at refining these parameters. Therefore, it was concluded that in the calibration process of the multi-sensor optical SMS, the 1-D HT should be used for automated plane detection of the two-plane artefact.

5.10 Tables

Table 5-1: Effect of noise on placement of votes in Hough space (with bin size of $1 \times 1 \times 1 \text{ mm}^3$) for a $200 \times 200 \text{ mm}^2$ plane, when 3×3 kernel is used in estimating surface normals

Standard Deviation of noise	Number of bins in Hough space	Number of bins with at least one vote in Hough space	Vote count of bin with the maximum votes	W – size of cluster in Hough space (mm)
5	4×4×8	67	14302	0.84
10	9×9×15	211	5358	1.69
20	18×18×31	769	1561	3.38
50	48×48×75	4263	269	8.45
100	108×100×147	13599	69	16.88
200	274×233×254	28562	16	33.65

Table 5-2 : Effect of the distance of plane to origin of coordinate system on size of corresponding cluster in Hough space (W).

X coordinate of centroid of plane (mm)	W – size of cluster in Hough space (mm)
0	29.4
200	33.6
400	44.2
600	57.6
800	72.3

Table 5-3 : Effect of size of convolution kernel for computing surface normals of plane with noise (rms of 200 μm) on a Hough space with bin size of $1 \times 1 \times 1 \text{ mm}^3$

Kernel size	Number of bins in Hough space	Memory (MB)		% of dense matrix used by sparse matrix	Number of bins with at least one vote in Hough space	W – size of cluster in Hough space (mm)
		Dense	Sparse			
3x3	274x233x254	64.86	0.83	1.28	28562	33.65
5x5	73x62x119	2.15	0.25	11.63	7837	11.83
7x7	38x46x68	0.48	0.07	14.58	2428	6.13
9x9	22x23x37	0.07	0.03	42.86	1101	3.75
11x11	17x16x30	0.03	0.02	66.67	520	2.55

Table 5-4 : Comparison of detection accuracy of 3-D Hough transform before and after least squares fitting to improve FON position, at different bin sizes.

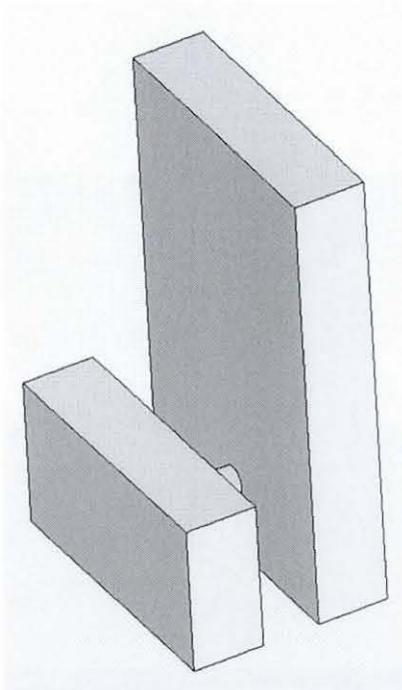
Bin size (mm^3)	Accuracy of plane detection – <u>without</u> least squares fitting to improve FON position (μm)	Accuracy of plane detection – <u>with</u> least squares fitting to improve FON position (μm)
1x1x1	59.3	10.6
2x2x2	366.2	16.2
4x4x4	380.6	16.2
8x8x8	366.2	16.2
16x16x16	520.9	16.2

Table 5-5 : Multiple planes : comparison of detection accuracy of the 3-D HT and 1-D HT after least squares fitting

Plane ID number	Detection accuracy of 1-D HT (μm)	Detection accuracy of 3-D HT (μm)
1	10.6	10.6
2	16.7	16.7
3	26.8	26.8
4	40.3	40.3
5	56.9	56.9

5.11 Figures

(a)



(b)

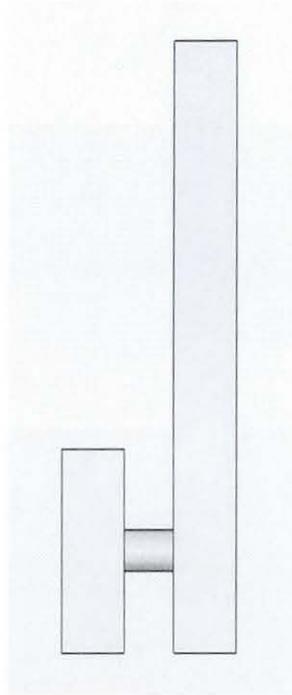


Figure 5-1 : Proposed arrangement for planar artefact consisting of two planes with a fixed known distance (a) projective view (b) side view.

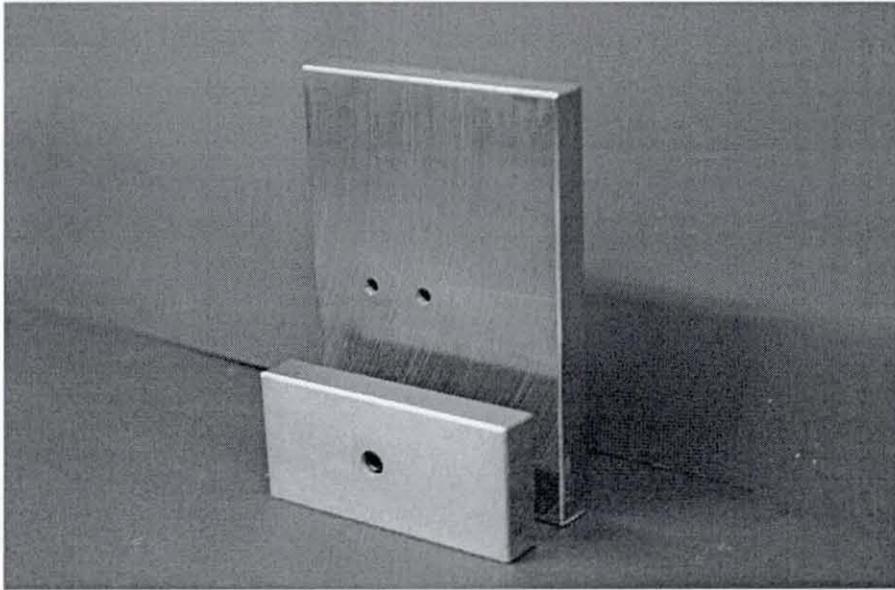


Figure 5-2 : Planar artefact consisting of two precision ground plates and a length bar.

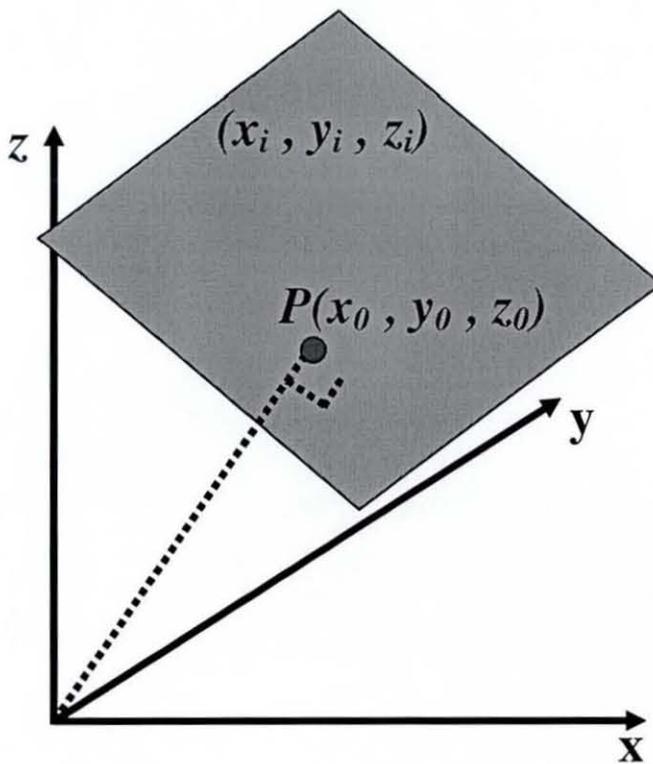


Figure 5-3 : Foot of normal (FON) position of plane.

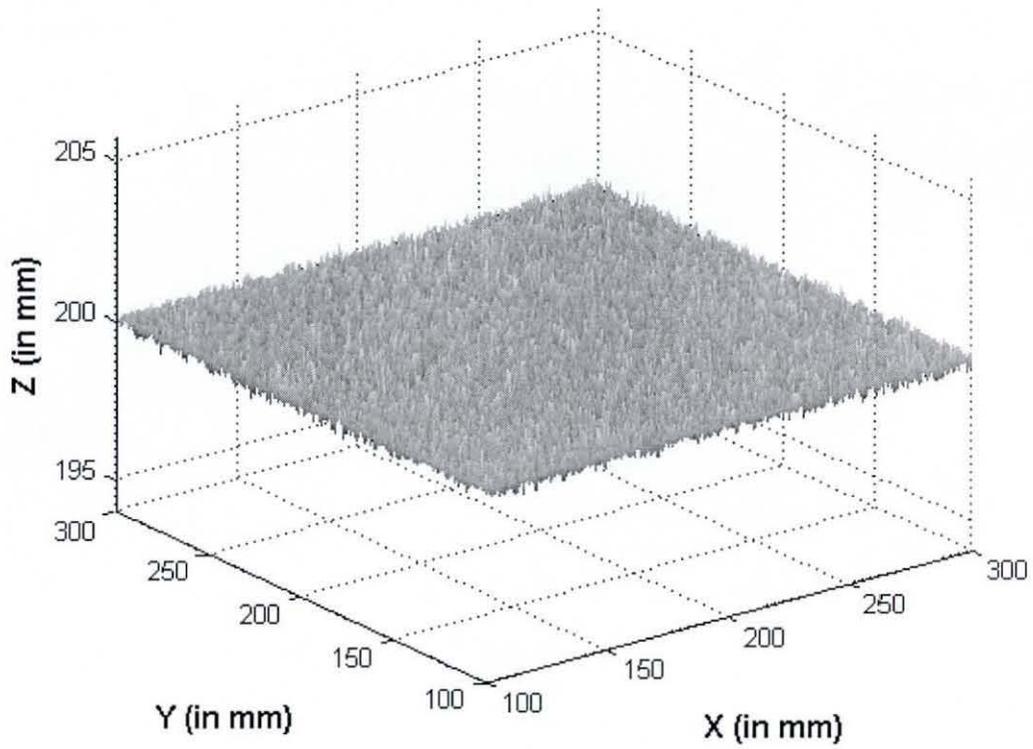
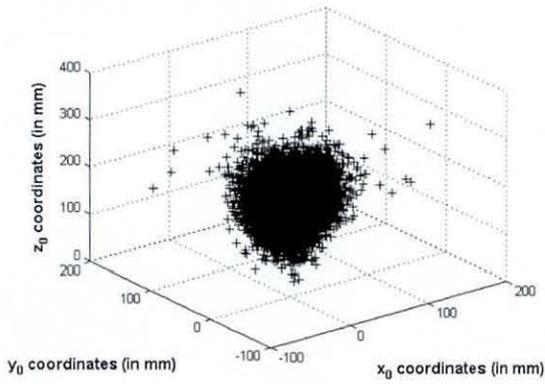
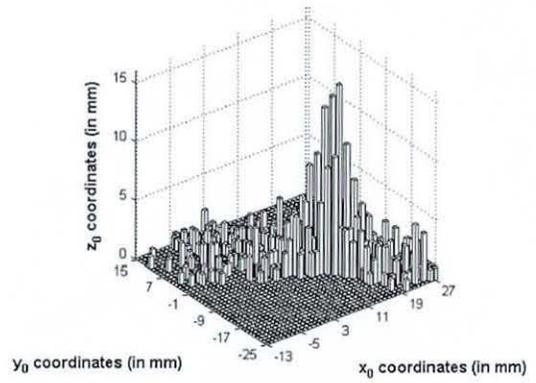


Figure 5-4: Point cloud of $200 \times 200 \text{ mm}^2$ simulated plane with noise ($\sigma = 200 \text{ }\mu\text{m}$).

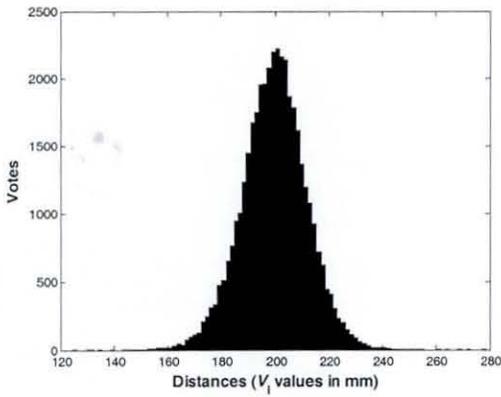
(a)



(b)



(c)



(d)

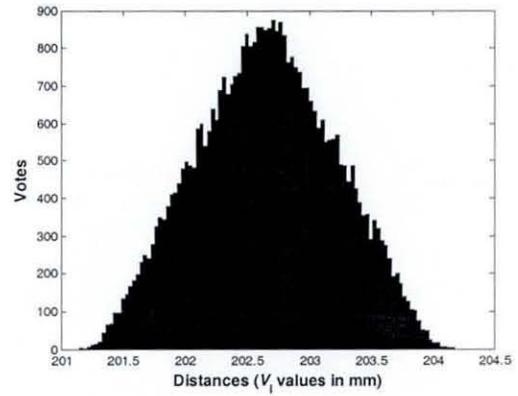


Figure 5-5: Hough space figures for $200 \times 200 \text{ mm}^2$ simulated plane with noise ($\sigma = 200 \mu\text{m}$); (a) The estimated FON coordinates before binning for 3-D HT; (b) Distribution of votes on a slice through Hough space (at $z_0 = 199 \text{ mm}$); (c) Histogram of v_i values for 1st iteration of 1-D HT; (d) Smaller range of v_i values in histogram for 2nd iteration of 1-D HT.

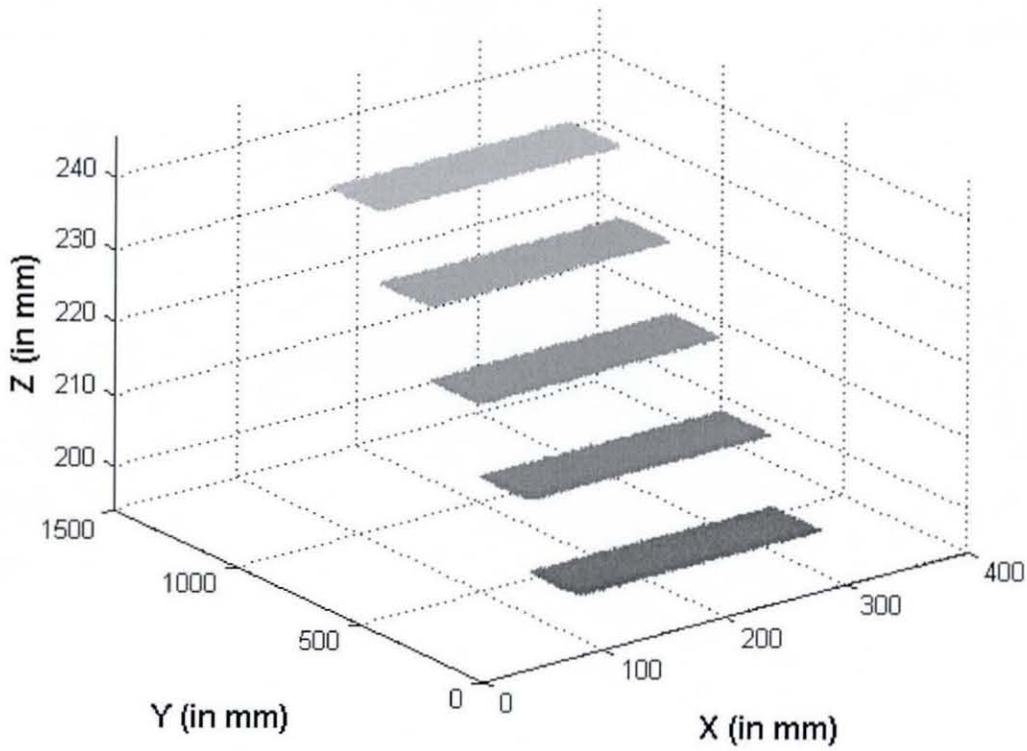


Figure 5-6: 3-D point cloud containing 5 simulated $200 \times 200 \text{ mm}^2$ planes with σ of $200 \text{ }\mu\text{m}$ on each plane.

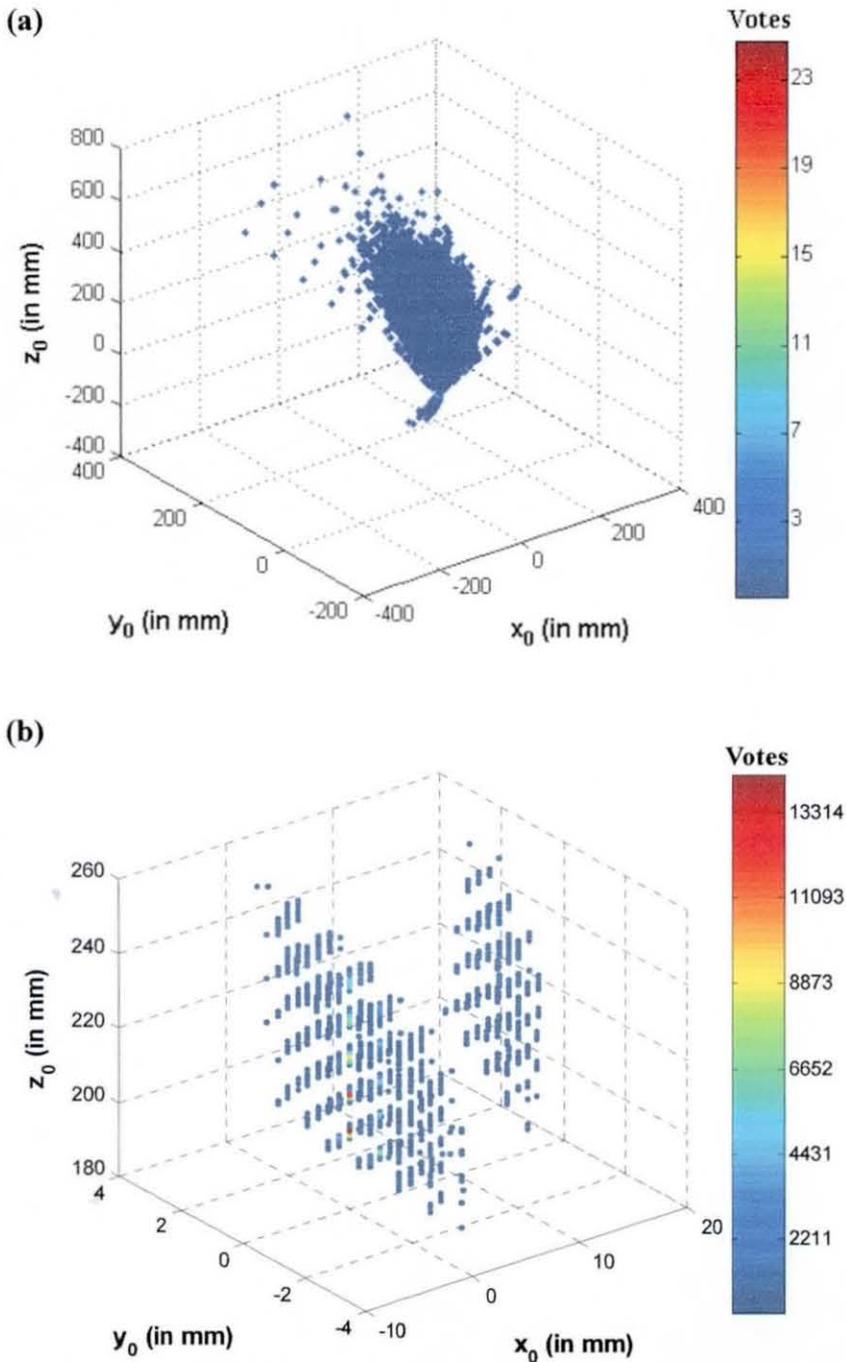


Figure 5-7: 3-D Hough space with bin size $1 \times 1 \times 1 \text{ mm}^3$ containing data from simulated multiple-planes of size $200 \times 200 \text{ mm}^2$ with noise σ of $200 \mu\text{m}$. (a) Hough space when 3×3 kernel is used in computing surface normals; (b) Hough space when 19×19 kernel is used in computing surface normals.

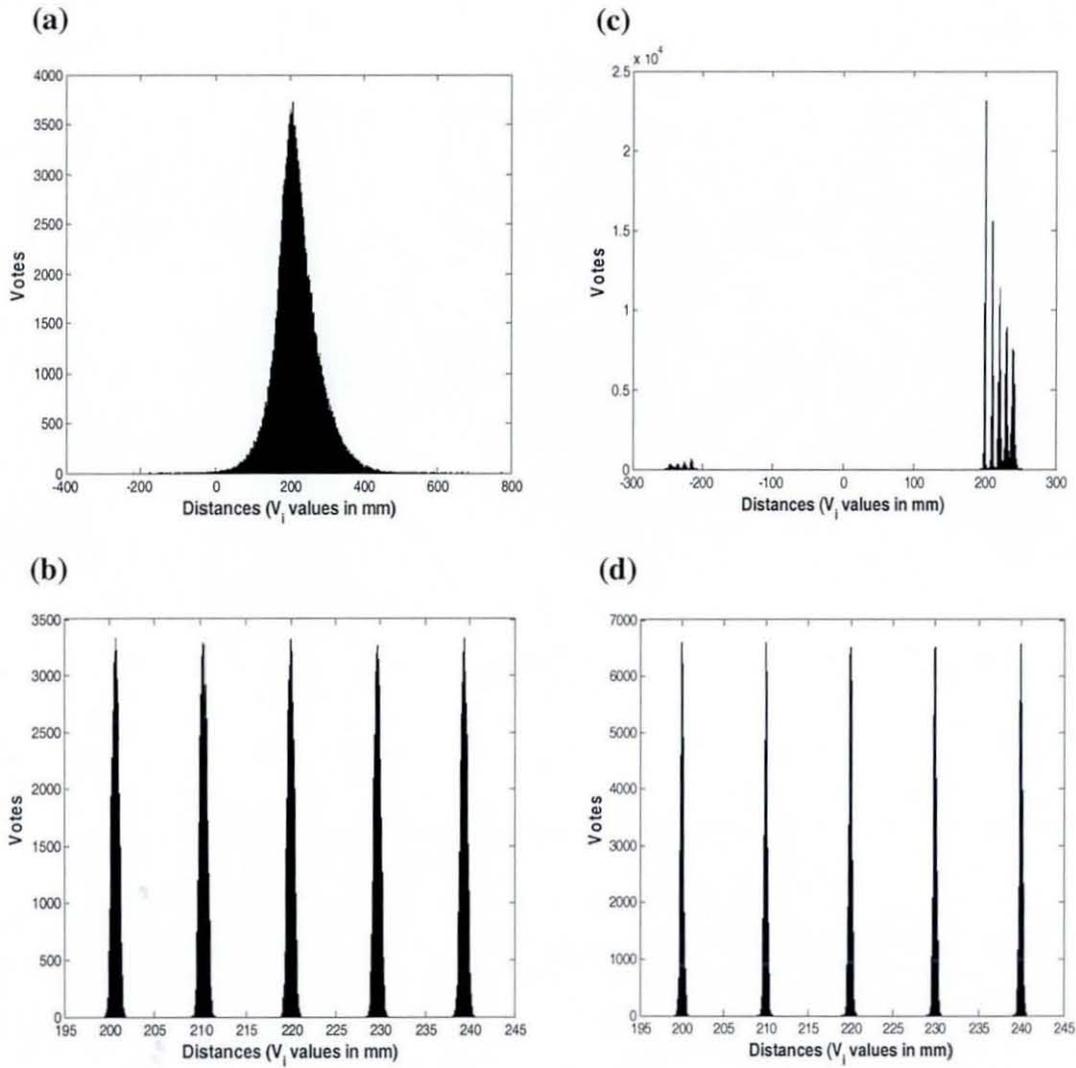


Figure 5-8: 1-D Hough space (consisting of 500 bins) containing data from simulated multiple-planes of size $200 \times 200 \text{ mm}^2$ with noise σ of $200 \mu\text{m}$. (a) Hough space after 1st iteration of 1-D HT when a 3×3 kernel is used for computing surface normals; (b) Hough space after 2nd iteration of 1-D HT when a 3×3 kernel is used for computing surface normals; (c) Hough space after 1st iteration of 1-D HT when a 19×19 kernel is used for computing surface normals; (d) Hough space after 2nd iteration of 1-D HT when a 19×19 kernel is used for computing surface normals.

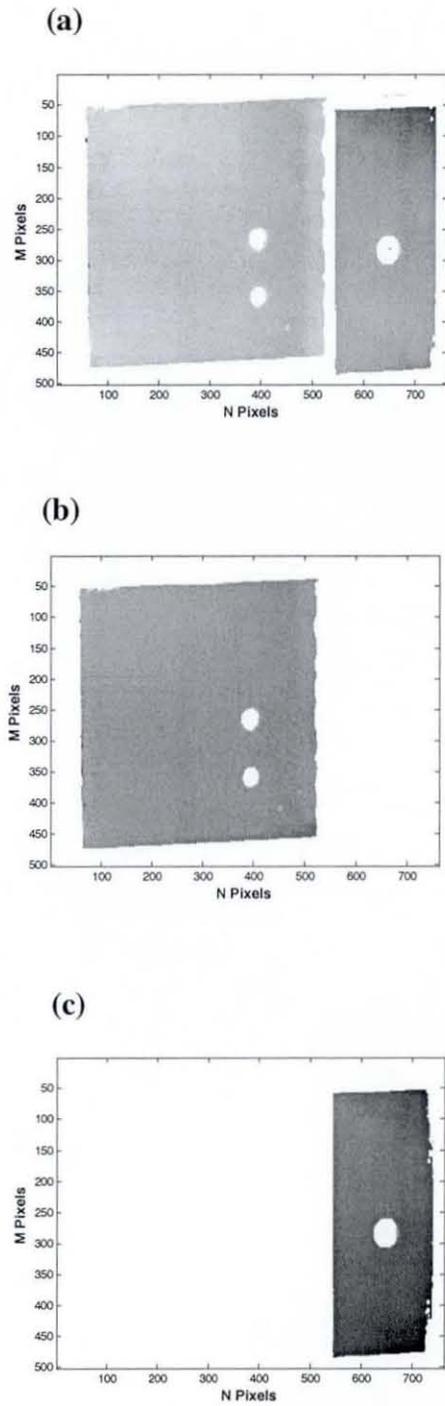


Figure 5-9: (a) Intensity image of two-plane artefact for C_1P_1 ; Intensity images of the detected planes; (b) Plane 1; (c) Plane 2.

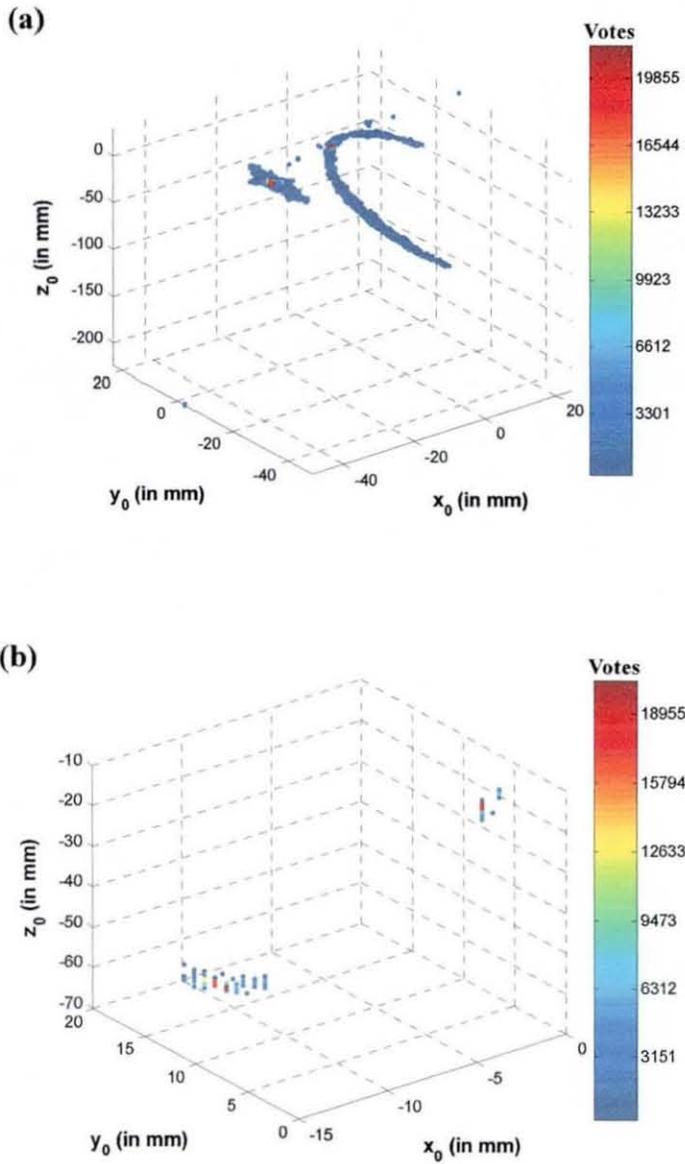


Figure 5-10: The 3-D Hough space populated with data from point cloud acquired by C_1P_1 . (a) The Hough space when no threshold is applied to votes (b) The Hough space after a threshold of 900 is applied to votes.

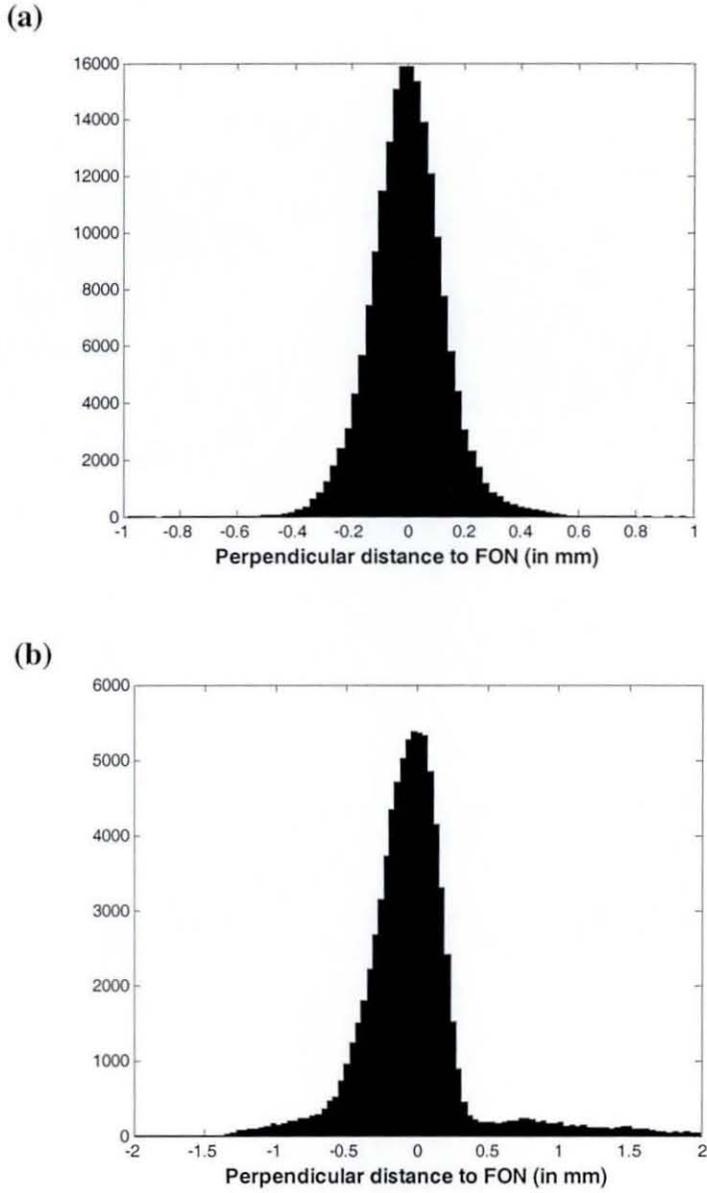


Figure 5-11: Histogram of the perpendicular distance, d , between points on the detected planes and their respective FONs estimated using the 3-D HT. (a) Histogram for plane 1; (b) Histogram for plane 2.

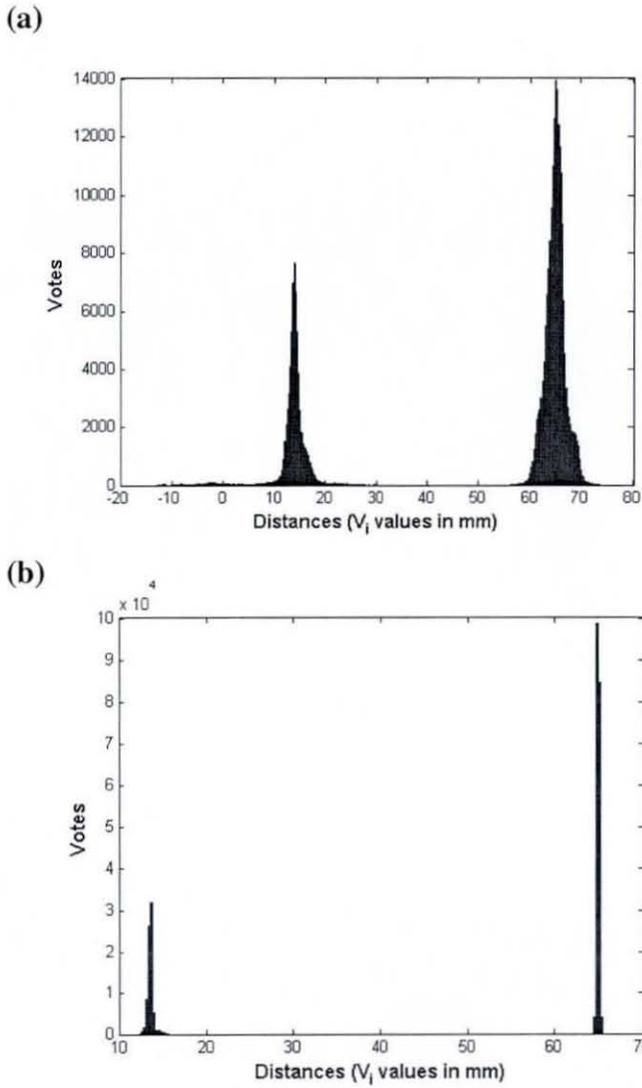


Figure 5-12: The 1-D Hough space (made up of 1000 bins) populated with data from point cloud acquired by C_1P_1 . The distinct peaks are representative of each plane in the point cloud. (a) The Hough space at the 1st iteration of the 1-D HT; (b) The Hough space at the 2nd iteration of the 1-D HT.

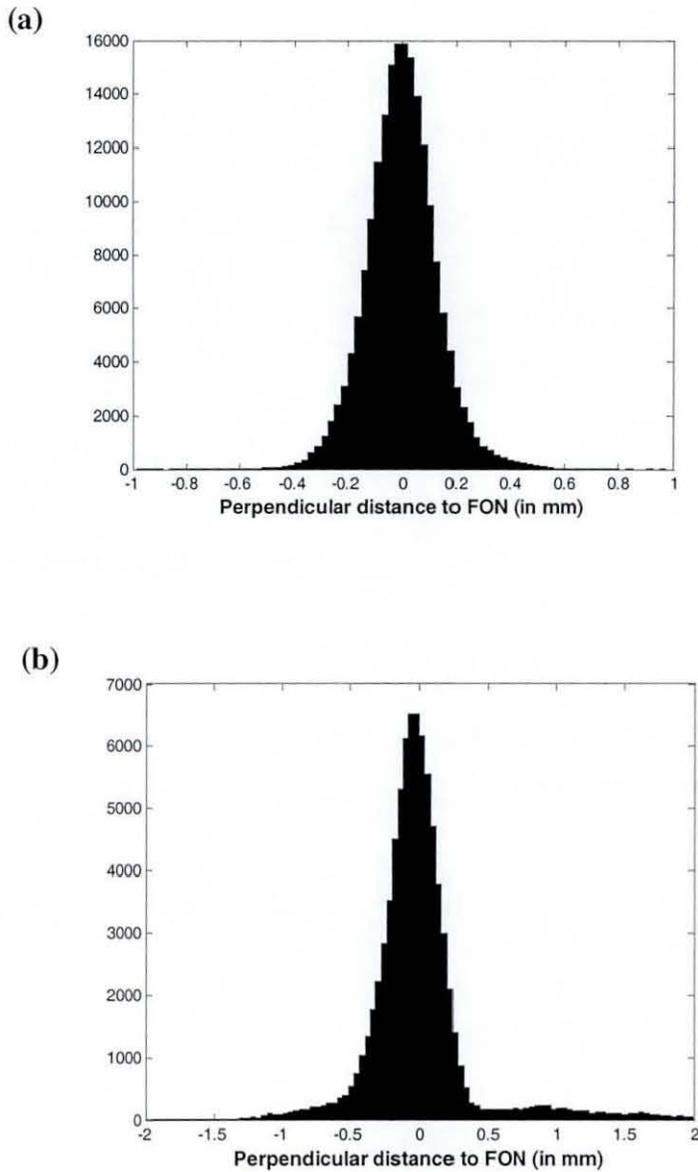


Figure 5-13: Histogram of the perpendicular distance, d , between points on the detected planes and their respective FONs estimated using the 1-D HT. (a) Histogram for plane 1; (b) Histogram for plane 2.

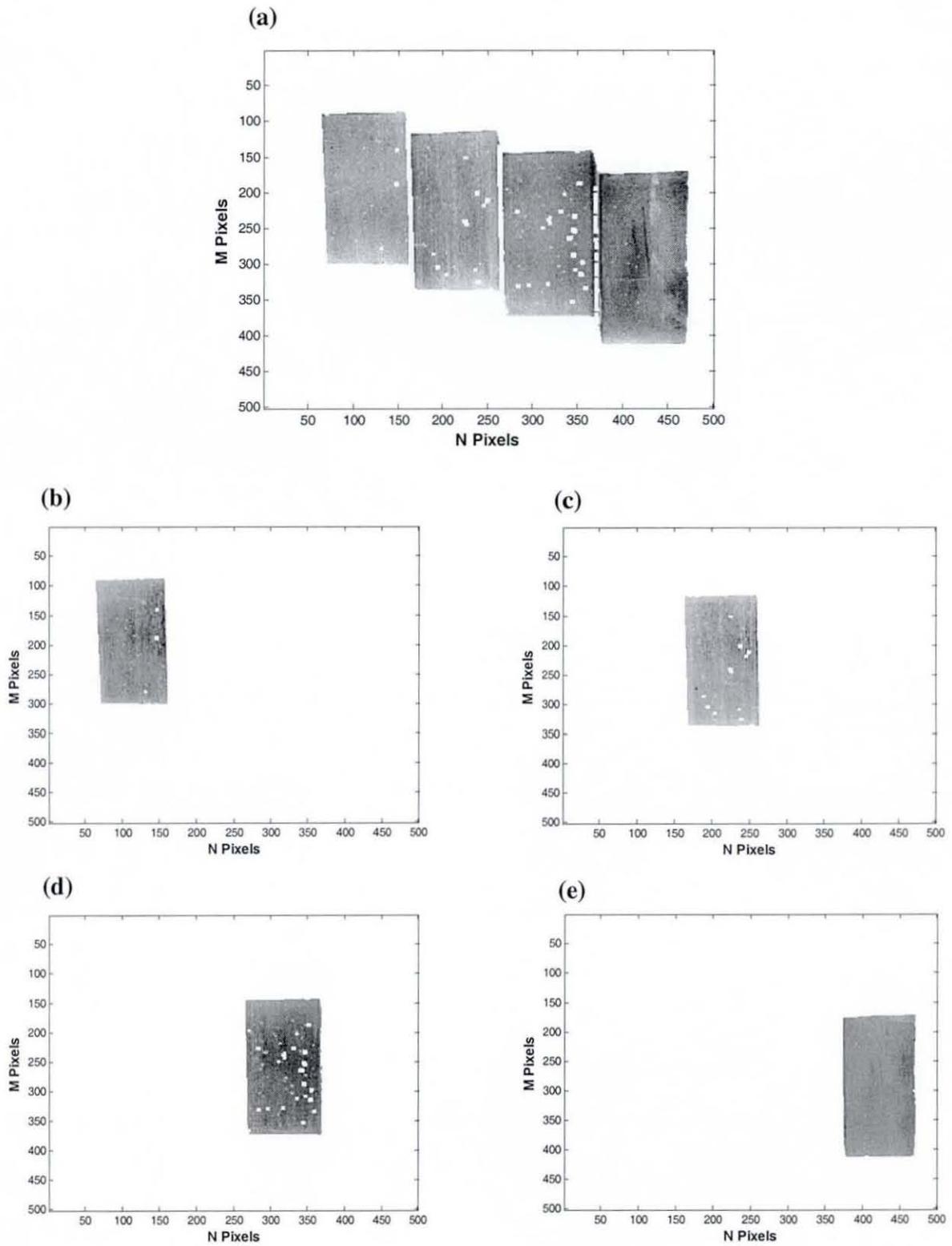


Figure 5-14: (a) Intensity image of stepped block for C_1P_1 ; Intensity images of the detected planes; (b) Plane 1; (c), Plane 2; (d) Plane 3; (e) Plane 4.

6 Implementation of Calibration Process

6.1 Introduction

In Chapter 3, the new calibration process based on a novel bundle adjustment method was introduced. The calibration process has been described in terms of two major phases: (1) initialisation and (2) refinement. Also, the sub-processes: (1) shape data acquisition, (2) sensor parameter initialisation, (3) shape data post-processing, and (4) bundle adjustment, have been introduced. Shape data acquisition involves the measurement of a calibration artefact, which is common to both the initialisation and refinement phases, where the measured shape data from each camera-projector pair is saved to disk. Subsequently, the measured data can be read from disk and used either to initialise or to refine the calibration parameters. This implies that the calibration process can be carried out *online* (i.e. immediate processing of data after each measurement) or *offline* (i.e. processing of data after all measurements have been made). However, the implementation discussed here refers to offline calibration.

The initialisation phase refers to the process for initialising estimates for the sensor parameters of the optical shape measurement system (SMS). This involves measuring a single pose of a 2-D calibration artefact, and on completion, the measured shape dataset for each camera-projector pair is read from disk, from where the direct linear transformation (DLT) method (discussed in Chapter 2) is used to retrieve initial estimates of the respective camera and projector parameters. It should be noted that the 2-D artefact used at this stage is different from the artefacts described in Chapters 4 and 5, and will be described in this Chapter.

The refinement phase of the calibration process involves shape data acquisition, shape data post-processing and bundle adjustment. In shape data acquisition, multiple poses of a second calibration artefact are measured and saved to disk. In this case, the calibration artefact consists of a set of spheres or planes (described in Chapters 4 and 5) of known position and orientation in a local frame of reference established by measurements on a mechanical coordinate measuring machine (CMM). Shape data post-processing involves

reading each set of measured shape data from disk and using the initial estimates of the sensor parameters to convert to a point cloud. From the point cloud, the required number of artefact features is detected (using methods described in Chapters 4 for spheres or Chapter 5 for planes), the artefact pose is estimated, and the required number of control points is selected from the detected features.

On processing all the measured data sets for all poses and all camera-projector pairs, finally, the initialised calibration parameters (sensor parameters and pose parameters) and control points are used in the bundle adjustment method to refine the calibration parameters in a non-linear optimisation. The refinement phase is thus computationally intensive, and to allow for automation of this step, various parameters are introduced to enable user control at the outset. These parameters include, for example, the number of artefact poses to be used for calibration, the number of control points per artefact, the thresholds for the phase error and modulation maps for selecting valid pixels in each measurement, and parameters for 3-D feature detection e.g. the global votes threshold for detecting spheres or planes. The objective is that once the sensor parameters have been initialised, the refinement phase is run automatically to obtain the optimal calibration parameters which consist of the sensor and artefact pose parameters. Discussions referring to 'automatic' calibration thus refer to the refinement phase of the calibration process. It is worth noting that the accuracy of the refined sensor parameters obtained after bundle adjustment is strongly dependent on the acquisition of accurate shape data and the use of only points that lie on the surface of respective artefact features as control points. If as a result of inaccurate shape data or the selection of *rogue* control points, the bundle adjustment fails to reach its potential minimum, the calibration parameters could be further improved by using them in a subsequent loop to process the measured data and carry out another bundle adjustment. Therefore, multiple sequences of the refinement phase are also possible.

In this chapter, the shape data acquisition process, which is common to the two phases of the calibration process, is described. The procedure for computing initial estimates for each sensor is explained, including a description of the 2-D calibration artefact designed

for this purpose. Although the calibration artefacts (multiple-sphere, ball bar, and two-plane) and their feature detection methods have been fully described in Chapters 3 and 4, how their geometric features are utilised in the calibration process is highlighted here. Methods for automatically selecting the pixels in image space corresponding to control points from detected 3-D features of the artefact are described. The bundle adjustment process is discussed in terms of the degrees of freedom of calibration parameters. The data structure for handling the variety of settings for the calibration process is described, including the software implementation of the calibration process. The chapter concludes with discussions on results obtained using the calibration process on datasets from the respective 3-D calibration artefacts, describing the procedure for calibration, performance of the calibration process, and measurement accuracy of the optimised sensor parameters. It is worth noting that part of the contents of this chapter have been included in a recently accepted SPIE conference paper [24].

6.2 Shape Data Acquisition

The shape data acquisition stage of the calibration process consists of positioning the calibration artefact within the measurement volume whilst measurements are made using all possible combinations of cameras and projectors. In the refinement phase of the calibration the calibration artefact is held in multiple orientations and locations ('poses'), while for the initialisation phase, it is held in a single orientation and position i.e. a single pose. For each camera-projector pair, the projector projects a sequence of fringe patterns, while the camera acquires the images of these patterns on the measured object. The deliverables from each camera-projector pair include an intensity map, and for horizontal and vertical fringes respectively (after combining data from the acquired images), phase gradient map, modulation map and phase error map (see Figure 6-1). Thus for measurements with each camera-projector pair (with camera resolution of 1024×1024 pixels), there are seven datasets, with total file size of 26Mb. From acquired images of each fringe orientation, the modulation and phase error maps provide information for identifying valid pixels in the phase gradient maps; thus, thresholds are set for these respectively. The phase gradient maps for respective fringe orientations, ω_x and ω_y , encode the shape information with values in the range $-\pi$ to $+\pi$, with each valid pixel

corresponding to a 3-D point on the object surface. It should be noted that the dimensions of the phase gradient maps is the same as the dimensions of the camera. Therefore, for a 1024×1024 pixel camera, each phase gradient map would be a 1024×1024 matrix, and consequently, its corresponding point cloud could be expressed as three 1024×1024 matrices for the respective x , y , z coordinates. On completion of each measurement sequence, the data is saved to disk.

6.3 Sensor parameter initialisation

6.3.1 2-D DLT method

As shown in Section 3.2.2 of Chapter 3, in order to use Eqns. (3-1) and (3-2) to compute a point cloud, initial estimates for the sensor parameters are required. The DLT method which involves the minimisation of an algebraic error function can be used to estimate these parameters. Issues relating to its adaptation as part of the calibration process for the optical SMS will be discussed here. The DLT method requires the use of control points – the 3-D DLT requires at least 6 non-coplanar points, while the 2-D DLT requires at least 4 coplanar points. Another key requirement is that these control points should be well distributed in the sensor's field of view.

The 3-D DLT described in Chapter 2, leads to the retrieval of the external and internal parameters of the sensor model. However, the 2-D DLT retrieves only the external parameters, thus requiring a priori knowledge of the internal parameters of the camera, ξ_H, η_H, c . This consequently leads to a reduction in the minimum number of control points required to 4 coplanar points. It is safe to set the principal point offset to a nominal setting of $\xi_H, \eta_H = 0$ for a camera. However, the shift in the projection of light in commercial video projectors implies that the value for one of the principal point offset parameters would always be non-zero. Although the nominal value of the sensor's focal length could be used as an initial estimate for the principal distance, c , it can also be estimated using the magnification of the image of the artefact, which is simply the ratio of a distance in image space to a distance in object space. Therefore, if we consider a distance between a pair of points in image space, Y_i , and the distance between their

corresponding points in object space, Y_o (see Figure 6-2), the magnification, m , can be expressed as

$$m = \frac{Y_i}{Y_o} \quad (6-1)$$

provided that the line joining the points is perpendicular to the optical axis of the camera. Similarly, m can be expressed as a ratio between Z_i (the distance from the CCD or SLM to the pinhole, O_c , which is equivalent to c), and Z_o (the distance from, say, the centre of the DLT artefact to the pinhole, O_c), thus

$$m = \frac{Y_i}{Y_o} = \frac{Z_i}{Z_o} \quad (6-2)$$

Therefore,

$$c = \frac{Z_o Y_i}{Y_o} \quad (6-3)$$

The 2-D DLT between a point in 2-D object space with coordinates (x_M, y_M) and its equivalent coordinates (ξ_M, η_M) in image space (see Figure 6-3) can be written as

$$\begin{bmatrix} \xi_M \\ \eta_M \end{bmatrix} = \begin{bmatrix} x_M & x_M & 1 & 0 & 0 & 0 & -\xi_M x_M & -\xi_M y_M \\ 0 & 0 & 0 & x_M & x_M & 1 & -\eta_M x_M & -\eta_M y_M \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \end{bmatrix} \quad (6-4)$$

$L_1 \dots L_8$ are referred to as the DLT parameters. The above can also be written as

$$\begin{bmatrix} \xi_M \\ \eta_M \end{bmatrix} = \mathbf{M}\mathbf{L} \quad (6-5)$$

where $\mathbf{M} = \begin{bmatrix} x_M & y_M & 1 & 0 & 0 & 0 & -\xi_M x_M & -\xi_M y_M \\ 0 & 0 & 0 & x_M & y_M & 1 & -\eta_M x_M & -\eta_M y_M \end{bmatrix}$ is a $2n \times 8$ matrix (i.e. two

rows for each point correspondence) and \mathbf{L} is the 9×1 vector as shown above in Eqn.

(6-4). \mathbf{L} can then be solved in the least squares sense thus:

$$\mathbf{L} = \mathbf{M}^{-1} \begin{bmatrix} \xi_M \\ \eta_M \end{bmatrix} \quad (6-6)$$

The residual for each point correspondence is calculated by applying the computed vector \mathbf{L} in Eqn. (6-6), and then deducting the LHS from the RHS of Eqn. (6-5) i.e.

$\left(\begin{bmatrix} \xi_M \\ \eta_M \end{bmatrix} - \mathbf{M}\mathbf{L} \right)$. This quantity is a measure of the accuracy of the 2-D DLT.

The elements of \mathbf{L} could be rearranged to give the camera matrix, $\mathbf{A} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_4 & L_5 & L_6 \\ L_7 & L_8 & 1 \end{bmatrix}$. The

external sensor parameters can then be extracted from the camera matrix, \mathbf{A} , in the following way [56]. Firstly, we create a matrix \mathbf{P} , using the known internal parameters

$$\mathbf{P} = \begin{bmatrix} -c & 0 & \xi_H & 0 \\ 0 & -c & \eta_H & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6-7)$$

An estimate for the rotation matrix, $\hat{\mathbf{R}}$, can be calculated using the first two columns of \mathbf{A} ,

$$\hat{\mathbf{R}} = \mathbf{P}^{-1} \begin{bmatrix} L_1 & L_2 \\ L_4 & L_5 \\ L_7 & L_8 \end{bmatrix} \quad (6-8)$$

while an estimate for the translation vector, $\hat{\mathbf{T}}$, can be calculated using the third column of \mathbf{A} thus,

$$\hat{\mathbf{T}} = \mathbf{P}^{-1} \begin{bmatrix} L_3 \\ L_6 \\ 1 \end{bmatrix} \quad (6-9)$$

We can then calculate a scaling factor, λ , to scale $\hat{\mathbf{R}}$ and $\hat{\mathbf{T}}$ thus:

$$\lambda = \sqrt{R_{11}^2 + R_{21}^2 + R_{31}^2}, \quad (6-10)$$

$$\hat{\mathbf{R}}' = \hat{\mathbf{R}}/\lambda \quad \text{and} \quad \hat{\mathbf{T}}' = \hat{\mathbf{T}}/\lambda \quad (6-11)$$

At this stage, $\hat{\mathbf{R}}'$ is presently a 3×2 matrix and a third column can be computed using the vector cross product of its 1st and 2nd columns ($\hat{\mathbf{R}}'_{i1}$ and $\hat{\mathbf{R}}'_{i2}$) which gives

$$\hat{\mathbf{R}}'_{i3} = \hat{\mathbf{R}}'_{i1} \times \hat{\mathbf{R}}'_{i2} \quad (6-12)$$

However, because $\hat{\mathbf{R}}'$ (which is now a 3×3 matrix) does not satisfy the orthonormality constraint of a standard rotation matrix, Singular Value Decomposition (SVD) can be used to normalise and decompose it thus:

$$[\mathbf{U}\mathbf{E}\mathbf{W}] = \text{SVD}(\hat{\mathbf{R}}') \quad (6-13)$$

Therefore, the orthonormal version of $\hat{\mathbf{R}}'$, which is the rotation matrix, \mathbf{R} , to rotate the world coordinates to the sensor's local coordinate system, can be calculated thus:

$$\mathbf{R} = \mathbf{U}\mathbf{E}'\mathbf{W}^T, \quad (6-14)$$

where $\mathbf{E}' = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & |\mathbf{UV}^T| \end{bmatrix}$. From Eqn. (6-14), the Euler angles can be retrieved as shown

below

$$\phi = \sin^{-1} R_{13}, \quad \omega = \cos^{-1} \left(\frac{R_{33}}{\cos \phi} \right), \quad \kappa = \cos^{-1} \left(\frac{R_{11}}{\cos \phi} \right) \quad (6-15)$$

Note that $\hat{\mathbf{T}}$ is an estimate of sensor pinhole coordinates with respect to the sensor's local coordinate system. Therefore, to calculate the pinhole coordinates $(x_o^{(c)}, y_o^{(c)}, z_o^{(c)})$ with respect to the world coordinate system, we would need to apply \mathbf{R}^{-1} (note that for a rotation matrix, $\mathbf{R}^{-1} = \mathbf{R}^T$) thus

$$\mathbf{T} = \begin{bmatrix} x_o^{(c)} \\ y_o^{(c)} \\ z_o^{(c)} \end{bmatrix} = -\mathbf{R}^T \hat{\mathbf{T}} \quad (6-16)$$

Using the initial estimates of the sensor's external and internal parameters for all cameras and projectors, we are then in a position to compute a point cloud for the shape data acquired by each camera-projector pair. Once the point cloud has been computed, ϵ_1 is also computed, providing a means of quantifying the accuracy of the estimated sensor parameters for the camera and projector respectively. If the average of the absolute values of ϵ_1 for valid pixels is quite large (say, greater than 10 mm), this indicates that some of the estimates for the sensor parameters of either the camera or the projector have significant errors.

6.3.2 Calibration artefact for sensor parameter initialisation

The use of control points as part of the DLT method led to the consideration of a new calibration artefact, apart from those discussed in Chapters 4 and 5. As the DLT method

would simply provide initial estimates for the sensor parameters, the control points are not required to be measured to high accuracy, and the calibration artefact could therefore be a very basic, low cost, light weight object. It was decided that the DLT artefact would consist of a printed pattern of circles on a flat surface (see Figure 6-4). As the artefact's orientation would initially determine the SMS' world coordinate system, ideally, it should be placed perpendicular to the camera's optical axis. However, in a multi-sensor system, where the sensors would be in different orientations, such positioning of the artefact may not be possible for all sensors. This implies that the acquired images of the circles may not necessarily be circular but elliptical. Therefore, the 2-D Hough transform (HT) (described in Chapter 4 for detecting circles) has been implemented to be robust enough to handle such elliptical shapes, ensuring that as many circles as possible can be detected.

Once the artefact has been measured with the optical SMS, the 2-D HT (described in Chapter 4) can be used to detect the circle centre coordinates from the respective intensity images. The detected circle centre coordinates, a_i and b_i , in this case would be in pixel coordinates. For the camera, we can convert from pixel coordinates to millimetres using Eqn. (3-2) in Section 3.2.2 of Chapter 3. However, for the projector, the detected circle centre coordinates, a_i and b_i , are used to extract the corresponding image plane coordinates from the phase gradient maps, and image plane coordinates are calculated using Eqn. (3-3) in Section 3.2.2 of Chapter 3. Finally, the centre coordinates of each detected circle in the camera image plane are matched with their corresponding known 2-D coordinates, and the 2-D DLT uses this information to estimate the external sensor parameters for the camera through application of Eqns. (6-5) to (6-17). The same process is used to estimate the projector external parameters by using the 2-D DLT method on the equivalent circle centre coordinates in the projector image plane. On the other hand, the distance between two circle centres in image and object space is used in Eqns. (6-1) to (6-3) to estimate the focal length of the respective cameras and projectors.

It should be noted that in a multi-sensor arrangement, as each camera-projector pair makes a measurement of the 2-D artefact, we would be presented with multiple data sets which could be used to get initial estimates for each camera and projector. For example,

in a two-camera two-projector SMS, for say C_1 , we can use the acquired shape data from either C_1P_1 or C_1P_2 to get estimates for the external parameters of C_1 . Like wise, for say P_1 , we can use the acquired shape data from C_1P_1 or C_2P_1 to get estimates for external parameters of P_1 . Although using one of such multiple datasets for respective sensors may be sufficient, however, it is suggested that all the datasets should be processed to calculate L using Eqn. (6-6), which could then be used to calculate the residual of Eqn. (6-5). From the datasets processed for each respective sensor, the retrieved L which leads to the lowest value of RMS of residuals would be selected, and the external sensor parameters retrieved using Eqns. (6-7) to (6-17).

The number of circles required on the 2-D artefact is dependent on the desired accuracy to which the 2-D DLT method is required to estimate sensor parameters. In general, the larger the number of control points, the higher the accuracy of the 2-D DLT method of sensor parameter estimation. However, this will of course lead to a greater processing time for estimating circle centre coordinates from intensity images of the artefact. The estimated parameters from the 2-D DLT method simply provides initial estimates for a non-linear optimisation (as part of the refinement phase of the calibration process), therefore, it is suggested that a conservative number of circles, say a 7×7 array of circles, should be used. Also, the diameter of the circles should be consistent with the measurement volume so as to ensure that when the artefact is imaged by the camera during shape data acquisition, a significantly large number of pixels are used to characterise each circle. This could help to simplify the detection process and lead to an improvement in detection accuracy.

6.4 Shape data post-processing

6.4.1 Initialising pose parameters for artefacts

As discussed in Chapter 3, the pose of an artefact is simply the transformation from the artefact's local coordinate system to the measurement volume's coordinate system. The artefact pose is thus described in terms of six parameters, three translational (i.e. t_x , t_y , and t_z) and three rotational (i.e. Euler angles, ω , ϕ and κ). For a multi-sensor SMS, errors in the initial estimates of sensor parameters (i.e. prior to bundle adjustment) lead to

misalignment in the point cloud computed from different camera-projector pairs. This implies that on the first sequence of calibration, the estimated pose of the artefact in respective point clouds as measured by all camera-projector pairings will normally be slightly different. However, in the bundle adjustment method, irrespective of the number of measuring camera-projector pairs, it is required that only a single set of parameters be estimated for each artefact pose. Thus, the initial estimate of each artefact's pose is calculated as the mean of pose parameters from respective point clouds of all camera-projector pairs.

For the multiple sphere and ball bar artefacts, the process of detecting sphere centre coordinates and ordering with control points has been described in Chapter 4. The parameters estimated from the ordering process are the rotation matrix, \mathbf{R} , and translation vector, \mathbf{T} , that describe the transformation from the control point's local coordinate system to the measurement volume's coordinate system. The Euler angles ω, ϕ, κ (extracted from \mathbf{R}) and \mathbf{T} , thus provide an initial estimate of a multiple sphere or ball bar artefact's pose.

In a calibration using the two-plane artefact, the 1-D HT described in Chapter 5 is used to estimate the foot-of-normal coordinates for the two planes. Although the key features of this artefact are simply two 3-D coordinates, the relationship between these 3-D coordinates differs from those of the ball bar artefact where the distance between the 3-D coordinates is known. In this case, the perpendicular distance between the two 3-D coordinates is known and fixed. Therefore, the problem of supplying an initial estimate of the pose of each plane needs to be solved algebraically. The expression derived by Huntley [86] for estimates of the pose of the two-plane artefact has been used in this work and will be described in Section 6.5.1.

6.4.2 Control point selection

In the implementation of the calibration process, two types of control point coordinates are required: (1) the known control point and, (2) the calculated control point. The *known* control point coordinates refers to the coordinates established by an independent measurement system such as a mechanical CMM. The *calculated* control point

coordinate refers to the coordinates of scattering points that lie on a 'control surface' (e.g. a sphere or plane's surface) which is established by using the sensor parameters to project rays from image space of camera and projector to object space (i.e. the point of closest approach of rays as described in Section 3.2.2 of Chapter 3). Indeed, knowledge that a measured scattering point lies on a control surface provides a valuable constraint in the bundle adjustment even though the precise coordinates of the point are unknown. The methods for detecting the calibration artefacts consisting of spheres and planes have been discussed in Chapters 4 and 5. Using these methods, we can identify the pixel coordinates in the phase maps that belong to each detected sphere or plane. The number of control points used in the bundle adjustment is a major factor influencing the method's computation time. Therefore, it is expected that only a subset of the identified pixel coordinates for each artefact feature would normally be used in the bundle adjustment. However, it is necessary that the subset should consist of points uniformly distributed across the surface of the artefact's features. An algorithm was thus developed and implemented to provide for automatic uniform selection of a given number of pixel coordinates for each detected artefact feature.

Consider a circular region of interest (ROI) in image space (see Figure 6-5), which could for example represent the pixels belonging to a sphere of a multiple-sphere or ball bar artefact, where we seek to select a total of $n_r \times n_c$ coordinates. Across row pixel coordinates of the ROI respectively, we find the minimum and maximum coordinates, M_{\max} and M_{\min} . Similarly, we find the equivalent for the column pixel coordinates, N_{\max} and N_{\min} . These coordinates are used to define a quadrilateral that encloses the ROI, with vertices having the following pixel coordinates $\mathbf{B}_1 = (M_{\min}, N_{\min})$, $\mathbf{B}_2 = (M_{\min}, N_{\max})$, $\mathbf{B}_3 = (M_{\max}, N_{\min})$, and $\mathbf{B}_4 = (M_{\max}, N_{\max})$. Each line segment of the quadrilateral thus has start and end points defined by these vertices.

The line segment with the longest length, say L_1 , is selected as the most dominant line and paired with its opposite facing line, L_2 (see Figure 6-5). We then calculate n_c coordinates uniformly along the vector defined by the coordinates of the start and end vertices of L_1 and L_2 respectively. Thus, a set of start and end coordinates of n_c lines are

defined between L_1 and L_2 . For each calculated coordinate $C_{1,i}$ along L_1 and the corresponding coordinate $C_{2,i}$ along L_2 , we obtain the coordinates for a line, D_i . On each line, D_i , we then calculate the coordinates of uniformly spaced points and identify which of these coordinates lie within the circular ROI. From the set of valid pixels which lie within the circular ROI, we select n_r coordinates, $E_{i,j}$ (where $i = 1, \dots, n_c$, and $j = 1, \dots, n_r$), (see Figure 6-6). It should be noted that each calculated coordinate is rounded-off to the nearest integer in order to convert to a pixel coordinate. To ensure even distribution across the ROI, it is important to set a threshold, n_t , for the minimum number of valid pixels per line. For example, if at the i^{th} position on L_1 and L_2 (i.e. on the line, D_i), $n_r > n_t$, then we would need to check the $(i+k)^{\text{th}}$ positions along L_1 and L_2 , (i.e. the line, D_{i+k} , where $k = \text{integer defining the offset, normally set to } 2$) until the criterion is met. However, if the criterion is not met, this indicates that sufficient pixels are not available in the data set probably as a result of severe thresholds applied in selecting valid pixels or problems in the feature detection process.

6.5 Bundle adjustment in object space

6.5.1 Relationship between pose parameters and control point coordinates

In Chapter 3, the novel bundle adjustment method that forms the basis of this new calibration process has been described. The bundle adjustment model is expressed as an objective function, F , with two terms, ε_1 , the sum of squares of errors of the mismatch offset distance of rays from each stereo pair of camera and projector (i.e. a measure of the error in the calculated control point coordinate), and ε_2 , the sum of squares of errors between the known and calculated control point coordinates (which lie on a control surface). For all calibration artefacts, the process for computing ε_1 is the same. Indeed, after calibration, for subsequent measurements using each camera-projector pair, ε_1 provides a measure of the quality of the calculated Cartesian coordinate at each pixel of the measured shape data i.e. the phase gradient maps. Thus, after using the modulation and phase error maps to select valid pixels in the phase gradient maps, the computed value of ε_1 at each valid pixel could be used as a criterion for further validation.

However, the process for calculating ε_2 is dependent on the characterisation of the 3-D features which constitute the calibration artefact. During bundle adjustment, the estimate of the pose of the artefact (i.e. the six degree of freedom transformation, comprising the Euler angles, ω, ϕ and κ (which are used to compute a rotation matrix, \mathbf{R}), and the three elements of a translation vector, \mathbf{T}) is applied to \mathbf{c}_{oi} , the known control point coordinates, to put these into registration with the measurement volume coordinate system thus :

$$\mathbf{c}_{ii} = \mathbf{R}\mathbf{c}_{oi} + \mathbf{T} \quad (6-17)$$

\mathbf{c}_{ii} is the position of the control point in global coordinates (i.e. with respect to measurement volume's coordinate system). In addition, the 3-D Cartesian coordinates, $\mathbf{s}_{i,j}$, of pixels belonging to each detected artefact feature (uniformly selected in image space as described in Section 6.4.2) are calculated by using the sensor parameters to project from image space into object space. ε_2 is thus a characterisation of the error between \mathbf{c}_{ii} and $\mathbf{s}_{i,j}$.

For an artefact consisting of spheres, e.g. the multiple sphere and ball bar artefacts, for each respective sphere, the relationship between the i^{th} point on the sphere surface and the corresponding sphere centre \mathbf{s}_{oi} is established by the known radius, r_i , of the sphere. \mathbf{s}_{ii} , the transformed known sphere centre coordinates is calculated as

$$\mathbf{s}_{ii} = \mathbf{R}\mathbf{s}_{oi} + \mathbf{T} \quad (6-18)$$

However, errors in the initial estimates of the sensor parameters imply that the distance between a calculated control point on the sphere's surface, $\mathbf{s}_{i,j}$, and \mathbf{s}_{ii} (i.e. the radius as determined from data from the optical SMS), will normally be different from r_i . The bundle adjustment thus aims to minimise this error which is given as:

$$\varepsilon_{2,i,j} = |\mathbf{s}_{i,j} - \mathbf{s}_{ii}| - r_i \quad (6-19)$$

On the other hand, for the two-plane artefact, ε_2 refers to the perpendicular distance from the transformed foot-of-normal position, \mathbf{p}_{ii} , of the i^{th} plane to a calculated control point

which lies on the plane's surface, $s_{i,j}$. Eqn. (6-17) does not apply to the foot-of-normal position as the foot-of-normal defined in local coordinates, \mathbf{p}_{oi} , is no longer in general the new foot of normal point \mathbf{p}_{fi} after transformation. Huntley [86] provides an algebraic solution to the pose parameters of the two-plane artefact which will be described here. The solution is based on the knowledge that there is no change in the perpendicular distance between points on the respective planes of the two-plane artefact when it is rotated about its local Z axis or translated along either its local X or Y axis.

Firstly, if we assume that $\mathbf{p}_{oi} = (0,0, g)$ (where g is a constant and the normal of the plane is along the Z axis), then we know that the new normal, \mathbf{n}_{oi} for the plane after rotation is along the direction

$$\mathbf{n}_{oi} = \mathbf{R}\mathbf{p}_{oi} = g \begin{pmatrix} \sin \phi \\ -\sin \omega \cos \phi \\ \cos \omega \cos \phi \end{pmatrix} \quad (6-20)$$

After translation by the displacement vector \mathbf{T} , the new foot-of-normal position can be calculated as

$$\mathbf{p}_{fi} = \alpha \mathbf{n}_{oi} = \alpha g \begin{pmatrix} \sin \phi \\ -\sin \omega \cos \phi \\ \cos \omega \cos \phi \end{pmatrix} \quad (6-21)$$

where $\alpha = \frac{|\mathbf{p}_{oi}|}{|\mathbf{p}_{fi}|}$. Provided that $\cos \phi \neq 0$, the Euler angles ω and ϕ can be calculated

thus:

$$\omega = \arctan \left(\frac{-\mathbf{p}_{oi}(2)}{\mathbf{p}_{oi}(3)} \right) \quad (6-22)$$

$$\phi = \arctan \left(\frac{\mathbf{p}_{oi}(1)}{-\mathbf{p}_{oi}(2) / \sin \omega} \right) \quad (6-23)$$

Similarly,

$$\phi = \arctan\left(\frac{p_{oi}(1)}{p_{oi}(3)/\cos\omega}\right) \quad (6-24)$$

Although Eqns. (6-23) and (6-24) are equivalent, in practice Eqn. (6-23) is used if $\sin\omega > \cos\omega$, otherwise, Eqn. (6-24) is used.

With regard to the calculation of the translation vector \mathbf{T} , there is an infinite set of feasible \mathbf{T} vectors consistent with the measured data, and only one needs to be chosen. If we take \mathbf{T} to be along the \mathbf{n}_{oi} vector, i.e. in a direction normal to the plane, then it can be calculated as

$$\mathbf{T} = \beta \mathbf{n}_{oi} \quad (6-25)$$

where $\beta = \alpha - 1$.

For the two plane artefact, the bundle adjustment thus aims to minimise the error given as

$$\varepsilon_{2,i,j} = \mathbf{n}_{ui} \bullet (\mathbf{s}_{i,j} - \mathbf{p}_{ui}) \quad (6-26)$$

where $\mathbf{n}_{ui} = \frac{\mathbf{p}_{ui}}{|\mathbf{p}_{ui}|}$ is the unit vector normal to the plane, which is calculated by normalising

\mathbf{p}_{ui} .

It is worth noting that, in the context of the new calibration process a 'control point' refers to a calculated coordinate on the surface of a 3-D feature (i.e. either a sphere or plane's surface) and not a specific coordinate (such as a sphere centre or foot-of-normal position).

As mentioned earlier, in the bundle adjustment process, the sensor and artefact pose parameters are refined in a non-linear optimisation. During bundle adjustment, the calibration artefact should ideally be allowed to undergo arbitrary rigid body translation and rotation during the bundle adjustment. This means that all the six parameters for each q^{th} pose of the artefact, $[t_x, t_y, t_z, \omega, \phi, \kappa]_q^T$ are free to change during the non-linear

optimisation. However, the physical set up of the sensor and the geometry of the calibration artefact implies that constraints need to be applied to specific parameters in order to ensure convergence and hence accurate parameters after bundle adjustment. This is achieved by forcing the selected parameters to be constant during the optimisation process. In the implementation of the calibration process, parameters that should be constant during bundle adjustment are said to be 'fixed', while those that should be free to change are said to be 'free'. As highlighted in Chapter 3, the introduction of rigid body transformation of the calibration artefact implies that the external parameters of one of the sensors i.e. a camera or projector, should be 'fixed' during bundle adjustment. On the other hand, the geometric setup of the artefact requires the user to specify which of its six pose parameters should be 'fixed' during bundle adjustment. This aspect is considered in the following sections.

6.5.2 Parameter constraints for multiple-sphere or ball bar artefact

For a calibration using the multiple sphere artefact, the 3-D coordinates for the known control points are set with the centroid of the dataset as the origin. The coordinates are written to a text file and referred to as an *artefact* file. The artefact file for either a multiple sphere or ball bar takes the following format consisting of blocks of 4 rows (first three rows for 3-D coordinate and fourth row for the radius), one per sphere:

X	0	0.01	0
Y	0	0.01	0
Z	0	0.01	0
R	0	5.00	0

The first column is the keyword, the second column a flag array indicating which parameters are to be fixed and which are to be free (a value of '0' means fixed, while a value of '1' means free), the third column is an array of the values associated with the keywords, and the fourth column is presently set up for future use to indicate the accuracy of the respective values in the third column. Thus the file for the multiple sphere artefact would be made up of a 132×4 array (33 spheres at 4 rows per sphere), while that of a ball bar artefact would be an 8×4 array (2 spheres also at 4 rows per sphere).

To estimate each artefact's pose during shape data post-processing, the known sphere coordinates are read from the artefact file and matched with the estimated sphere centre coordinates obtained by the 3-D HT from the measured data. During bundle adjustment, the multiple-sphere artefact requires no constraint because there are no symmetry axes and all estimated pose parameters are set as 'free'. For a calibration using the ball bar artefact, which consists of just two spheres separated from each other by a fixed distance, the known control points for each sphere in the artefact file are set such that the separation of the sphere centre coordinates is along the Z axis. For example, for a ball bar artefact with separation between the two sphere centres measured as 40 mm, in the ball bar artefact file, the coordinate for the known control point (i.e. sphere centre coordinates) for the first sphere, S_1 , could be written as [0, 0, 10]; while for sphere 2, S_2 , [0, 0, 50], with all units in mm. This configuration implies that there is rotational symmetry about one axis, and therefore during bundle adjustment, κ (i.e. the pose parameter for rotation about the Z axis), should be set as 'fixed'. All other parameters are 'free'.

6.5.3 Parameter constraints for two-plane artefact

Considering a single plane P, with foot-of-normal components x_f, y_f, z_f . If $x_f = y_f = 0$, P, can be translated along the x and y directions, and rotated about the κ Euler axis, with no visible effect. After transformation to global coordinates, x_f and y_f will no longer in general be non zero, but the same principle applies – only one displacement component should be free. It is expected that if the estimated x and y displacement components at the start of the bundle adjustment have errors, then these will therefore remain but be compensated for by an error in the z displacement component that puts the plane back into the correct position. The same principle applies to the two-plane artefact, and therefore, when used in a bundle adjustment, only z_f and the ω and ϕ Euler angles are allowed to be 'free', while the remaining three parameters are set as 'fixed'. In order to be set up correctly in the artefact file, it is essential that the foot-of-normal coordinates for the planes be defined with their normals aligned along the Z axis. For example, for a two-plane artefact with the planes separated by a distance of 40 mm, the known control points (i.e. foot-of-normal coordinates) could be set up as follows: plane 1, $P_1 = [0, 0, 10]$; plane 2, $P_2 = [0, 0, 50]$, with all units in mm.

6.6 Software implementation

6.6.1 Extensible markup language (XML) description

The completion time of the shape data post-processing stage is highly dependent on the number of camera-projector pairs for each measurement and the number of artefact poses. For example, if 20 poses of the ball bar artefact are used to calibrate a two-camera two-projector SMS, and the processing of data from each camera-projector pair for each pose takes, say, n seconds to complete, then, it will take $80n$ seconds (i.e., the number of camera-projector pairs per measurement \times number of artefact poses $\times n$) to complete each iteration. In a similar vein, the number of control points per artefact is a key factor affecting the completion time of the bundle adjustment. The repetitive nature of shape data post-processing therefore implies that n is bound to increase significantly if the processing involves significant user input.

The refinement phase of the calibration process requires the values of a significant number of parameters to be specified. Examples include the thresholds to be applied to the respective phase error and phase modulation maps to select valid pixels from the corresponding phase gradient map, and the votes threshold for 3-D feature detection using the 3-D HT. However, some of these parameters may vary from one camera-projector pair to the other, and also from one artefact pose to the other, thus affecting the reliability of the calibration process. For example, in detecting spheres, an appropriate votes threshold has to be set for the 3-D HT. However, this value may not be appropriate for detecting spheres in all datasets, and would need to be automatically reset. The sheer quantity of parameters required in controlling the data acquisition of the SMS, as well as the refinement phase of the calibration process indicate that a naïve composition of various input files would prove unwieldy. In the course of the project, it was decided that all the relevant parameters associated with the SMS should be saved in an XML (extensible mark-up language) document. The representation of these parameters in an XML document enhances the implementation of the automatic calibration process, especially in cases when parameters are updated. Also, this enhances the scalability of the current implementation, such that for example, a graphic user interface (GUI) can easily be integrated into the calibration software.

XML is used for digital representation of documents such that they can be put in some kind of computer-readable notation, in order for a computer to store, process, search, transmit, display and print them [127]. This is possible once the document is represented in some kind of structure. An XML document is simply a structured document that is both computer and human readable. The hierarchical structure of an XML document can be broken down into components which are called 'elements'. Each element represents a logical component of a document and not only contains strings, but could also contain other elements. The element that contains all of the other elements is known as the *root* element. The elements contained in the root are called sub-elements. Sub-elements which contain one or more sub-elements are called *branches*, while those without are called *leaves*. Elements can also contain extra information attached to them called *attributes* which describe properties of the elements. The hierarchy is as shown below:

```
<root>
  <branch element_id= "1">
    <leaf>
      </leaf>
    </branch>
  </root>
```

In order to access an XML document's structure and content, a computer program called an XML processor or parser is required. The parser developed in Java by jdom.org. [128] has been used in this work.

6.6.2 Description of calibration XML document

The root element for the SMS's XML document is 'pv_sms'. The branch element which contains all the sub-elements that relate to the calibration of the SMS is 'pv_calibration'. 'pv_calibration' contains the sub-element 'pv_bundle_adjustment' which contains all the sub-elements referring to the calibration. The sub-elements of 'pv_bundle_adjustment' give information on the calibration, such as the type of calibration artefact, settings for the number of iterations of the calibration, the thresholds for the phase error and modulations, the full file path where the acquired SMS data has been stored, etc. With

regard to the calibration parameters, there is a 'pv_iteration' sub-element, which contains an input section, 'pv_input', where the initial estimates of the sensor and pose parameters are set. The output section, 'pv_output', contains the optimal sensor and pose parameters obtained after bundle adjustment. Appendix A-3 shows an example of the calibration section of the XML document for a single-camera single-projector SMS before and after calibration with one artefact pose, highlighting the quantity of data written to file during calibration.

6.6.3 Structure of software

In order to achieve rapid implementation of the algorithms and methods developed in this work, MATLAB[®] was selected as the programming language to develop the calibration software. MATLAB[®] is a high level technical computing language that provides functionality that avoids low-level administrative tasks such as declaring variables, declaring data types and allocating memory [129]. Add-on toolboxes which are collections of MATLAB[®] functions are also available to solve problems in areas such as signal and image processing, statistics and optimisation. MATLAB[®] is an interpreter (i.e. each MATLAB[®] statement is processed individually) and could be slow when executing certain block of statements e.g. recursive computations implemented as a 'for' loop. MATLAB[®] thus provides functionality for integrating computer programs written with programming languages such as C, C++ and Fortran. The interface computer programs, called *mex* files, are callable in MATLAB[®] as DLLs (dynamically linked libraries). Indeed, the philosophy used in the software implementation of the calibration process was that once a block of statements in an implemented MATLAB[®] function file or the function itself was discovered to be slow, it was implemented in C. An example is the implementation of the voting process for the 3-D HT used for sphere and plane detection which has been implemented as a C mex file. The variety of MATLAB[®] functions and C mex files of the calibration software have been grouped into folders based on functionality. The contents of these folders are described in Appendix A-4.

In the initialisation phase of the calibration, the initial estimates of the respective camera and projector parameters are retrieved and written to a text file. At the beginning of the refinement phase, a MATLAB[®] function is called, which reads the contents of this text

file and writes it to an XML document. In addition, other parameters are initialised and written to the XML document. A user has the option of either editing the parameters of the function, or simply editing an already existing XML document. As mentioned earlier, the current implementation of the refinement phase runs only in offline calibration mode, i.e. all the measurements should have been made by the SMS and saved to disk before calibration. Although only small changes are needed to enable online calibration, it would be useful from a user's perspective if such an implementation were also to provide a GUI front end.

From the XML document, we retrieve the number of sequences of calibration, which determines how many times the calibration process will be carried out, the number of poses that will be used in each sequence and the number of camera-projector pairs per pose (see dotted black arrow in Figure 6-7). Thus for each pose, the parameters for the i^{th} camera and the j^{th} projector are read from the XML document including their respective specifications (i.e. pixel resolution and sensor physical dimensions), and the SMS data associated with the pair is also read from disk. Using this information, a point cloud is computed and the spheres or planes in the point cloud are automatically detected by using parameters in the XML document which relate to the required 3-D feature detection method.

The known control points are read from disk and combined with the detected artefact features to estimate the artefact's pose (i.e. the transformation (translation and rotation) that will take the known control points (sphere centre or plane foot-of-normal coordinates) from a local frame to the SMS' world coordinate system). Pixels referring to calculated control points on the surface of each detected artefact feature are selected for use in the bundle adjustment. The total number of selected pixels is dependent on the required number of control points per artefact feature as specified in the XML document. The estimated pose of the artefact using data from the i^{th} camera and the j^{th} projector is written to the XML document and then data for the next camera-projector pair is processed (see black feedback arrow in Figure 6-7). Once data from all camera-projector pairs for the current pose is processed, datasets for the next pose are processed in the

same manner. It should be noted that there could be problems with the dataset of an artefact pose acquired by a camera-projector pair, say as a result of occlusion of the artefact during measurement. In such an occurrence, all data associated with the current artefact pose (though valid data may have been acquired by other camera-projectors pairs) are not included in the bundle adjustment. This implies that the number of artefact poses used for the calibration is reduced by 1, and the relevant sub-elements in the XML document are updated.

On completion of shape data post-processing, where all the artefact poses have been estimated and control points selected, the processed data is used along with the estimated sensor parameters in a bundle adjustment for non-linear optimisation. On convergence, the optimised sensor and pose parameters are written to the XML document, including the number of iterations, the initial and final values of the square root of mean values of F , ε_1 and ε_2 respectively. If more than one calibration sequence has been requested by the user, the optimised sensor parameters are used as an initial guess in a subsequent calibration sequence (see grey feedback arrow in Figure 6-7). On completion of the calibration process, if more than one sequence was run, the ID number of the sequence with the best set of calibration parameters (i.e. the sequence with the lowest value of F) is identified and written to the XML document. After calibration, this sequence ID number can be read from the XML document and used to select the sensor parameters which will be used to compute the 3-D Cartesian coordinates of subsequent measurements made with the optical SMS.

6.7 Experimental results

6.7.1 Description of experiments

The experimental results that will be discussed in this Section were acquired using a two-camera two-projector optical SMS. In discussions, the two cameras will be referred to as C_1 and C_2 , while the two projectors will be referred to as P_1 and P_2 . Also, the four different camera-projector pairs will be referred to as C_1P_1 , C_1P_2 , C_2P_1 , C_2P_2 . The specification for the camera (Vosskuhler CCD1300QFB) is $M = 1024$ pixels, $N = 1024$ pixels, $N_x^c = 6.6\text{mm}$, $N_y^c = 6.6\text{mm}$. The specification for the projector (Vision Plus

V1100Z) is $M = 1024$ pixels, $N = 768$ pixels and $N_x^c = 10.506$ mm, $N_y^c = 14.008$ mm .

The cameras and projectors were appropriately aligned to ensure that a large portion of the proposed measurement volume (estimated as $400 \times 400 \times 500$ mm³) was within the respective fields of view of cameras and projectors. In addition, the focus of the projector had to be set such that projected images are defocused within the measurement volume [4, 11]. The optical SMS was placed approximately 1630 mm away from the centre of the proposed measurement volume. The specific relative distances of the pinholes of the cameras and projectors were obtained during the initialisation phase of the calibration process.

Initially, a 2-D artefact was placed near the centre of the volume and measured with all camera-projector pairings. Subsequently, measurements of the respective artefacts (multiple sphere, ball bar and two plane artefacts) were made at six positions within the volume, i.e. six poses for each artefact. The acquired shape data from all camera-projector pairs was saved to disk. Finally, offline calibration was carried out using the measured datasets on a computer with a Pentium4 processor (2.80GHz, 1GB RAM). The thresholds set for the modulation and phase error maps of each camera projector pair are given in Table 6-1. These were used in the post processing of the shape data of the respective 3-D artefacts for selecting valid pixels in each dataset. Subsequent subsections will discuss the results of calibrating the optical SMS using shape data from the different types of 3-D artefacts. Discussions will focus on assessing the performance of the calibration process by comparing the values of specific quantities before and after calibration. These quantities include the objective function, F , and its two contributing terms, ε_1 and ε_2 (with their corresponding weightings set as $\gamma_1 = \gamma_2 = 1$), and the point cloud mismatch error (which will be described in Section 6.7.3). For the bundle adjustment process, the criterion for convergence was set based on the magnitude of the change in F (in μm) between two consecutive iterations, and in this case, it was set at 10 μm . However, if this criterion was not met, then the bundle adjustment was allowed to carry on until the 150th iteration. As stated in Section 3.3.2 of Chapter 3, if F increases, an attempt is made to force a reduction. However if a reduction is not possible, this indicates that there are problems with the dataset being processed. In the discussions in subsequent

sections, it should be noted that the values of F , ε_1 and ε_2 which will be quoted are the square root of their mean values as calculated over all control points used in the bundle adjustment.

The implementation of the bundle adjustment provides for optimising 12 sensor parameters. As stated in Section 2.2.1 of Chapter 2, the k_1 distortion parameter is often regarded as sufficient for modelling lens distortion in close range photogrammetry. Therefore, for all the sensors, the k_2 and k_3 distortion coefficients were set as 'fixed'. The external parameters of C_1 were set as 'fixed' to pin down the measurement volume's coordinate system (as suggested in Section 6.5.1), while its internal and k_1 parameters were set as 'free'. For C_2 , P_1 and P_2 , their external, internal and k_1 parameters were all set as 'free'.

6.7.2 Initialisation of sensor parameters using the 2-D DLT method

In this phase of the calibration, the 2-D-artefact used was simply a piece of paper on which was printed an array of 7×7 circles. The printed pattern of circles was attached to a planar surface positioned on a frame placed close to the centre of the proposed measurement volume. The data sets for C_1P_1 and C_2P_2 were read from disk, from where circle centre pixel coordinates are estimated by applying the 2-D HT on the intensity image of the artefact (see Figure 6-8 which shows the intensity image from C_1P_1 and C_2P_2 with the estimated circle centre coordinates as crosses). As mentioned in Section 6.3.1, the 2-D DLT method requires estimates for the internal parameters of the sensor. For C_1 and C_2 , the principal point offset is set as $\xi_H, \eta_H = 0$. However, the shift in the illumination beam from commercial video projectors implies that the value for one of the principal point offset parameters would always be non-zero. A number of trials of different values were applied and used to compute a point cloud for each camera-projector pair, to visually check the ε_1 values across all valid pixels. Thus, the following the principal point offsets for P_1 and P_2 , as $\xi_H = 0, \eta_H = 6$ mm respectively, gave reasonable results. Indeed, this setting for the principal point offsets is consistent with the results obtained by Zhang and Huang [62], when calibrating a similar projector. In a

similar *generate and test* approach, the principal distance, c , for C_1 and C_2 was set as 35 mm, while for P_1 and P_2 , it was set as 30 mm.

Using the process described in Section 6.3 which involves the use of the 2-D DLT method, for each camera-projector pair, the estimated circle centre coordinates are used to estimate the external parameters of the camera and projector respectively. The accuracy of the estimated parameters can be calculated by computing the difference between the calculated coordinates of the circle centres (i.e. coordinates computed by using the estimated parameters to project world coordinates of control points to image space) and the estimated circle centre coordinates (i.e. coordinates estimated by the 2-D HT). This is calculated for each control point from Eqn. (6.5) by deducting the rhs of the equation from its lhs (since the elements of L are populated by the 2-D DLT method). The RMS of this residual value for all control points for C_1 , C_2 , P_1 , and P_2 was calculated as 2.43 μm , 3.09 μm , 2.91 μm , and 3.32 μm respectively. Tables 6.2 - 6.5 show the initial estimates of all the sensor parameters obtained by the 2-D DLT method.

6.7.3 Results for calibration using multiple-sphere artefact

Results from calibrating the optical SMS using the multiple-sphere artefact with randomly distributed stalk heights are discussed in this section. As mentioned in Section 6.5.2, the sphere centre coordinates were measured using a mechanical CMM and used to set up an artefact file. It is important to provide an easy and effective means of arbitrarily orientating and positioning the artefact in the measurement volume. Therefore, underneath the artefact's base plate, a hole was tapped at its centre to allow for attaching the artefact to a camera tripod. Once the artefact was securely attached to the camera tripod, after each measurement, the set up was moved to a new position in the measurement volume. Using all camera-projector pairings, six poses of the sphere artefact were measured in different orientations and positions in the measurement volume. The initial measured pose of the artefact was such that its normal was approximately in the same direction as the Z axis of the SMS's world coordinate system. The artefact was then translated along the Z axis to be measured at approximately 60 mm and 120 mm respectively, from its initial position. Subsequently, the artefact was returned to its initial position, rotated 90 degrees counter-clockwise about the Z axis, and

measured. Once again, in this same orientation, the artefact was translated along the Z axis to be measured at approximately 60 mm and 140 mm respectively, from its initial position. It should be noted that the requirement that all spheres on the artefact should be visible in each camera's field of view restricted achievable rotations and translations to mostly the Z axis.

With regard to shape data post processing, the point cloud was calculated using the initial sensor parameters and acquired shape data of each camera-projector pair. From the point cloud, the 33 spheres of the artefact were detected and the pose of the artefact was estimated. Figure 6-9 shows the locations of all the detected sphere centres in the measurement volume. 4 pixel coordinates corresponding to scattering points which lie on the surface of each detected sphere i.e. 132 calculated control points per artefact, were selected for use in the bundle adjustment process. Therefore, a total of 7128 calculated control points were used in the bundle adjustment process. The shape data post processing took 1037 seconds (17 minutes) to complete. However, it should be noted that while processing some point clouds, the global votes threshold of the 3-D HT had to be reset automatically. Problems such as this, demonstrated the robustness of the calibration process, ensuring that multiple attempts are made until the required number of spheres in the point cloud are detected.

The bundle adjustment for the datasets in this experiment did not converge based on the earlier stated criterion of 10 μm , by the time of the 150th iteration, which took 893 seconds to complete. The final values of the respective camera and projector parameters after bundle adjustment are as shown in Tables 6.2 - 6.5. The square root of the mean value of F reduced from 2.19 mm to 1.47 mm. The ε_1 term reduced from 1.48 mm to 0.80 mm, indicating a reduction of up to 46%. On the other hand, the ε_2 term reduced from 1.61 mm to 1.23 mm, showing a 24% reduction. A second iteration, with the optimised sensor parameters used as initial estimates in the calibration process, led to only a small further reduction in F , ε_1 and ε_2 , to the values 1.31 mm, 0.76 mm and 1.07 mm respectively. However, to ensure a fair comparison with results of calibration using other 3-D artefacts, results relating to the first iteration only will be discussed. Thus for

the estimated measurement volume of $400 \times 400 \times 500 \text{ mm}^3$, the figure of 1.23 mm for the ε_2 values represents an estimated measurement accuracy of around one part in 400.

Figures 6-10(a) and (b) show the square root of the mean of squares (RMS) of ε_1 values over all respective camera-projector pairs and all poses (i.e. Pose 1 - Pose 6), computed using the initial and optimised sensor parameters. Figures 6-11(a) and (b) show the RMS of ε_2 values over all respective camera-projector pairs and all poses. As mentioned in Section 6.4.1, errors in the initial estimates of the sensor parameters leads to misalignment in the combined point clouds computed from acquired data from the respective camera-projector pairs. For each camera pixel which contains valid data from the two projection directions, there will in general be a difference in the two calculated real world coordinates. Figure 6-12 shows this mismatch for the two cameras across all poses, indicating improvements across all acquired datasets. Detailed discussions will focus on acquired data for C_1P_1 and also acquired data of all camera-projector pairs for Pose 3.

The greyscale intensity image of Pose 3 of the multiple sphere artefact acquired by C_1P_1 is shown in Figure 6-13 (a). Figure 6-13 (b) shows the ε_1 errors for the point cloud computed using initial and optimised sensor parameters on acquired data from C_1P_1 . It can be seen that there is a reduction in the magnitude of the errors. However, most especially across the sphere surfaces, there is an evident occurrence of a significant gradient in the ε_1 error distribution. Further investigation into this problem suggested that this may be due to errors in the acquired shape data. The slightly translucent nature of the material used to manufacture the sphere artefact implies that when it is illuminated directly by a light source (in this case a projector), some of the light gets scattered internally and re-emitted. This effect is probably strongest around the edges of the sphere and will give a bias in the computed projector ray direction. Under ambient light this is not an obvious problem. During shape data acquisition, this effect led to errors in the computed phase gradient maps, which consequently led to significant errors in the computed point cloud and also the poor performance of the bundle adjustment process.

Further evidence for this effect being due to the material, rather than the artefact geometry, was that the corresponding plots from the ball bar experiments (where the sphere material was hardened steel covered in developer powder) showed negligible gradients in the ε_i errors. Details of these results will be shown in Section 6.7.4.

Errors in initial sensor parameter estimates give rise to differences in the estimated pose of the artefact from point clouds of respective camera-projector pairs. Tables 6-2 to 6-5 show the initial and optimised sensor parameters, while Table 6-6 shows the estimated pose parameters from the point cloud from all camera-projector pairs for Pose 3 at the first and second iterations of the calibration. The reduction in the standard deviation for most of the pose parameter demonstrates the reduction in the misalignment of the respective point clouds of the camera-projector pairs. However, errors in the acquired data prevent the achievement of significant reductions in the standard deviations.

Figures 6-14(a) and (b) show values for ε_1 , calculated using initial and optimised sensor parameters respectively. For the values computed using the initial sensor parameters, most lie in the range -3 mm to +3 mm of the histogram. Although the process of calibration reduces the degree of spread as demonstrated by the approximately Gaussian distribution, the peak is centred around 1 mm. For the ε_1 values, the standard deviation reduced from 0.93 mm to 0.38 mm. A similar occurrence is observed in Figures 6-15(a) and (b), which show results for the ε_2 values (calculated using the optimised sensor and optimised pose parameters), where the standard deviation reduced from 0.95 mm to 0.74 mm. A slight improvement in the point cloud mismatch errors can also be seen in Figure Figures 6-16(a) and (b).

6.7.4 Results of calibration using ball bar artefact

The results of calibrating using the ball bar artefact are discussed in this section. It is necessary to note that the ball bar artefact used in this case is slightly different from the artefact described in Chapter 5. Indeed the artefact used in this experiment was a prototype to test the concept of calibration using the ball bar artefact. In this case, the ball bar artefact consisted of two 38.1 mm diameter steel spheres and a rod. The steel spheres

have a reflective surface and these had to be spray coated with developer powder in order to prevent specular reflection. The distance between the centres of the two spheres was measured as 231.016 mm using a mechanical CMM (having a measurement accuracy of 27 μm) and used to set up an artefact file as specified in Section 6.5.2.

In this experiment, the artefact was set up on a frame that allowed for arbitrary translation and rotation. Thus, after each measurement, the orientation of the artefact was changed from vertical to horizontal or vice versa and the frame on which it was set up was moved to a new position in the measurement volume. Using all cameras and projectors, six poses of the ball bar artefact were measured in different orientations and positions in the measurement volume. Three poses of the artefact were measured in vertical orientations i.e. sphere separation along the Y axis of the SMS' world coordinate system, while the remaining three were measured in horizontal orientations i.e. sphere separation along the X axis of the SMS' world coordinate system. While in either orientation, the artefact was placed at three different positions along the Z axis of the measurement volume.

With regard to the shape data post processing of the measured data, in selecting valid pixels, the thresholds set for the modulation and phase error maps of each camera projector pair are given in Table 6-1. From the computed point cloud of each camera-projector pair, the two spheres of the artefact were detected and the pose of the artefact was estimated. The detected sphere centres in the measurement volume are shown in Figure 6-17. 64 pixel coordinates corresponding to scattering points which lie on the surface of each detected sphere i.e. 128 calculated control points per artefact, were selected for use in the bundle adjustment process. Therefore, a total of 3072 calculated control points were used in the bundle adjustment process. The shape data post processing was completed in 464 seconds. Figure 6-18 shows the greyscale intensity image of the ball bar artefact acquired during measurement by C_1P_1 . In the figure, only intensity values of valid pixels are displayed, while the red and green coloured pixels are the pixels automatically selected for use in the bundle adjustment.

For the bundle adjustment process, the criterion for convergence was similar to that set in Section 6.7.3 for data from the multiple sphere artefact. The bundle adjustment converged after 45 iterations in 255 seconds, with the objective function, F , reduced from 2.38 mm to 116 μm . The ε_1 term reduced from 1.58 mm to 63 μm , indicating a reduction of up to 96%, while the ε_2 term reduced from 1.78 mm to 97 μm , showing a 95% reduction. Thus for the estimated measurement volume of $400 \times 400 \times 500 \text{ mm}^3$, the figure of 97 μm for the ε_2 values represents an estimated measurement accuracy of around one part in 5,000. The final values of the respective camera and projector parameters after bundle adjustment are shown in Tables 6-2 to 6-5. Figures 6-19 (a) and (b) shows RMS values for ε_1 over all camera-projector pairs and all poses, while Figures 6-20 (a) and (b) show RMS values for ε_2 over all camera-projector pairs and all poses (i.e. Pose 1 to Pose 6). Figures 6-21 (a) and (b) show the mean point cloud mismatch errors for the two cameras across all poses. Further detailed discussions will focus on acquired data for C_1P_1 and also acquired data of all camera-projector pairs for Pose 3.

Table 6-7 shows the estimated pose parameters from the point cloud from all camera-projector pairs for Pose 3 at the first and second sequences of the calibration i.e. from point clouds computed using initial and optimised sensor parameters, respectively. The reduction in the standard deviation of all the pose parameters demonstrates the significant reduction in the misalignment of the respective point clouds from each camera-projector pair. Figures 6-22 (a) and (b) show values for ε_1 calculated using initial and optimised sensor parameters respectively. For the values computed using the initial sensor parameters, the two peaks in the histograms represent data sets referring to scattering points on the surface of the two respective spheres. However, this marked difference is completely eliminated through the calibration process as seen by the approximately Gaussian distribution, centred on the origin. For the ε_1 values, the standard deviation reduced from 1.01 mm to 0.09 mm. A similar improvement is observed in Figures 6-23(a) and (b), which show the ε_2 values calculated using the optimised sensor and optimised pose parameters, with the standard deviation reduced from 1.05 mm to 0.13 mm. In addition, a significant improvement in the point cloud mismatch errors can also

be seen in Figures 6-24 (a) and (b). This demonstrates that although only subsets of all the measured scattering points of the spheres were used in the bundle adjustment, these are sufficient to provide very significant improvements in the calculated sensor and pose parameters.

6.7.5 Results of calibration using two-plane artefact

The results of calibrating using the two-plane artefact are discussed in this section. The plane artefact used in this case is similar to the one described in Chapter 5, consisting of two steel plates and a 25.4 mm (1 inch) length bar. The distance between the two plates of the artefact was measured as 51.090 mm using a mechanical CMM (having a measurement accuracy of 27 μm and by point probing 32 points on each of the steel plates), was used to set up an artefact file as specified in Section 6.5.3. In addition, the parallelism of the plates of the artefact was measured as 0.012°, with the flatness of the larger plate measured as 8.4 μm , while that of the smaller plate measured as 8.2 μm . However, the ideal distance between the two plates could be estimated based on the CMM measured thickness of say the smaller plate (25.68 mm) and the nominal length of the length bar, which in this case gave a separation of 51.088 mm. The steel plates have a shiny reflective surface and these had to be spray coated with developer powder in order to prevent illumination beam reflections from entering the camera aperture. However, it was discovered this approach did not completely eliminate the occurrence of specular reflection. Therefore, the artefact was rotated 10 degrees about the X and Y axes. In this experiment, the artefact was set up on a frame that allowed for arbitrary translation and rotation. Thus, after each measurement, the frame (on which the artefact was set up) was moved to a new position in the measurement volume along the Z axis of the measurement volume. However, after measurement at each artefact pose, the acquired shape data had to be analysed to assess if the orientation of the artefact led to specular reflection.

With regard to the shape data post processing of the measured data, in selecting valid pixels, the thresholds set for the modulation and phase error maps of each camera projector pair are given in Table 6-1. From the computed point cloud of each camera-projector pair, the two planes of the artefact were detected using the 1-D HT and the pose of the artefact was estimated. Figure 6-25 shows the detected foot-of-normal positions in

the measurement volume. 144 pixel coordinates corresponding to scattering points which lie on the surface of each detected plane i.e. 288 control points per artefact, were selected for use in the bundle adjustment process. Therefore, a total of 6912 control points were used. The shape data post processing was completed in 590 seconds. Figure 6-26 shows the greyscale intensity image of the two plane artefact acquired during measurement by C₁P₁. In the figure, only intensity values of valid pixels are displayed, while the red and green coloured pixels are the pixels automatically selected for use in the bundle adjustment.

For the bundle adjustment process, the criterion for convergence was similar to that set in Section 6.7.3. However, it was observed that the bundle adjustment process was unstable when datasets of the artefact measured within the negative Z axis of the measurement volume are included in the non-linear optimisation. This implies that estimated foot-of-normal positions with negative global Z coordinates would lead to stability problems in the bundle adjustment. In order to compensate for this, a positive offset (in this case 800 mm) is introduced to translate the sensors in the positive Z axis of the global coordinate system i.e. translating the sensor pinhole coordinates in the [0,0,1] direction, leading to a change only in Z coordinates. After bundle adjustment, the offset is removed from the optimised sensor pinhole coordinates by applying a negative offset (in this case, -800 mm) in the [0,0,1] direction. With this modification, the bundle adjustment converged after 34 iterations and took 175 seconds to complete. The final values of the respective camera and projector parameters after bundle adjustment are shown in Tables 6-2 to 6-5. The objective function, F , reduced from 1.87 mm to 230 μm . The ε_1 term reduced from 1.23 mm to 127 μm , indicating reduction of up to 90%, while the ε_2 term reduced from 1.41 mm to 189 μm , showing 87% reduction. Thus for the estimated measurement volume of $400 \times 400 \times 500 \text{ mm}^3$, the figure of 189 μm for the ε_2 values represents an estimated measurement accuracy of around one part in 2,600. Figures 6-27 (a) and (b) shows RMS values for ε_1 over all camera-projector pairs and all poses, while Figures 6-28 (a) and (b) shows values for ε_2 over all camera-projector pairs and all poses (i.e. Pose 1 to Pose 6). Figures 6-29 (a) and (b) show values of the point cloud mismatch for the

two cameras across all poses. Further detailed discussions will focus on acquired data for C_1P_1 and also acquired data of all camera-projector pairs for Pose 3.

Table 6-8 shows the estimated pose parameters from the point cloud from all camera-projector pairs for Pose 3 at the first and second sequences of the calibration i.e. from point clouds computed using initial and optimised sensor parameters respectively. The reduction in the standard deviation of all the pose parameters demonstrates the significant reduction in the misalignment of the respective point clouds from each camera-projector pair. Figures 6-30 (a) and (b) show values for ε_1 , calculated using initial and optimised sensor parameters, respectively. The values computed using the initial sensor parameters, result in two peaks in the histograms which represent data from the scattering points on the surface of the two planes. These two peaks are completely eliminated through the calibration process (as seen by the approximately Gaussian distribution centred at 0). For the ε_1 values, the standard deviation reduced from 0.38 mm to 0.12 mm. A similar improvement is observed in Figures 6-31 (a) and (b), which show values for ε_2 calculated using the optimised sensor and optimised pose parameters, and the standard deviation reduced from 0.60 mm to 0.17 mm. In addition, a significant improvement in the point cloud mismatch errors can also be seen in Figures 6-32 (a) and (b).

6.7.6 Comparison of results from calibration using different types of artefacts

As stated earlier, problems with the acquired shape data of the multiple sphere artefact lead to significant differences in the optimised parameters in comparison with parameters obtained using data from the ball bar and two plane artefacts. Figure 6-33 shows how the square root of mean values of F changed during the bundle adjustment process using acquired data of the respective calibration artefacts. It can be seen for datasets from the ball bar that not only is there convergence to the lowest value of F , but also, rapid convergence is achieved. Indeed, by the tenth iteration, the bundle adjustment had already converged quite close to the minimum. In comparison, for datasets from the two-plane artefact, a similar pattern can be observed, although the value of F is slightly larger. However, for datasets from the multiple sphere artefact, only a small reduction in F is achieved, and the convergence criterion is not met.

The accuracy of the sensor parameters obtained from calibrating the optical SMS using the three respective artefacts can be calculated. The accuracy can be calculated as the difference between a known length and the equivalent length calculated from the point cloud acquired by the optical SMS. For example, the length between the spheres in the ball bar artefact has been measured by a mechanical CMM. This same length can be calculated from datasets of the ball bar artefact by using the 3-D HT to detect the sphere centres and using non-linear fitting to improve on the centre coordinates accuracy. The accuracy of the SMS would therefore be the absolute difference between the SMS measured and the CMM measured lengths. It is expected that the accuracy of the measured length between the spheres of the ball bar artefact would be highest for parameters obtained from calibration using the ball bar.

Using all acquired datasets of the ball bar artefact, Figure 6-34 (a) shows the accuracy of the sphere separation using the sensor parameters obtained from calibration using the multiple sphere artefact to compute the equivalent point cloud. The average across all datasets is calculated as 2.87 mm. Figure 6-34 (b) shows the accuracy using the sensor parameters obtained from calibration using the ball bar artefact, with the average calculated as 73.2 μm . Figure 6-34 (c) shows the accuracy using the sensor parameters obtained from calibration using the two plane artefact, with the average calculated as 3.21 mm.

Similarly, the distance between the two plane artefact has been established by a mechanical CMM. This same distance can be calculated from datasets of the artefact by using the 1-D HT to detect the two foot-of-normal positions of the planes in order to identify and distinguish between the points that lie on the respective planes. Through non-linear fitting of the identified points on the respective planes (described in Chapter 5), the best fit surface normal of each plane can be calculated. In practice, the normals of the two planes are slightly different, thus the average of the estimated normals is used as the effective surface normal of the artefact. In addition, the identified points on the respective planes are used to calculate the centre of mass of each plane. Thus, using the

effective surface normal of the planes, the separation between the two planes is calculated as the perpendicular distance between their centres of mass. Similarly, the accuracy of the measured distance between the planes of the two plane artefact would be highest for parameters obtained from calibration using the two plane artefact.

Using all acquired datasets of the two plane artefact, Figure 6-35 (a) shows the accuracy of the plane separation when the sensor parameters obtained from calibration using the multiple sphere artefact are used to compute the equivalent point cloud. The average across all datasets is calculated as 0.77 mm. Figure 6-35 (b) shows the accuracy using sensor parameters obtained from calibration using the ball bar artefact, with the average across all datasets calculated as 0.18 mm. Figure 6-35 (c) shows the accuracy using sensor parameters obtained from calibration using the two plane artefact, with the average across all datasets calculated as 0.14 mm.

From the above it can be seen that sensor parameters obtained using data from the ball bar provide an accuracy close to that achievable by the parameters obtained from the two plane artefact (which in this case is supposed to have the highest level of accuracy as we are simply processing the same dataset that was used in obtaining optimal sensor parameters). On the other hand, for the sensor parameters obtained from calibration with the two plane artefact, the accuracy of measuring the length of the ball bar is quite low. This rather low accuracy could be as a result of the lack of perfect parallelism of the planes.

From the above discussions, it can be seen that the sensor parameters obtained from calibration with the ball bar artefact provide an acceptable level of accuracy across datasets of artefacts consisting of spheres and planes. It can therefore be concluded that the ball bar artefact is the most suitable artefact for calibrating the optical SMS. Its suitability can be summarised thus

- low cost and easy assembly of its constituent parts
- viewing is possible from any direction by sensors

- scalability to different measurement volumes and better flexibility to enable arbitrary rotations and translations
- reduced processing time of acquired data
- rapid convergence during non-linear optimisation of calibration parameters in bundle adjustment process
- better accuracy of optimised sensor parameters after calibration for 3-D shape measurement

However, the ball bar artefact's main drawback is the reduced coverage of measurement volume in any given point cloud, which consequently leads to the contribution of only a small fraction of camera pixels to the calibration process for any pose. Thus, in comparison with the multiple sphere and two-plane artefacts, to calibrate a large volume using a ball bar artefact, more poses would be required.

6.8 Summary

The new calibration process has been described in terms of four sub-processes: shape data acquisition, sensor parameter initialisation, shape data post-processing and bundle adjustment. The shape data acquisition stage of the calibration process consists of positioning the calibration artefact within the measurement volume whilst measurements are made using all possible combinations of cameras and projectors. In the initialisation phase of the calibration process, data from a single pose of a 2-D calibration artefact is acquired, while in the refinement phase, data from multiple poses of a 3-D calibration artefact is acquired. The low cost 2-D calibration artefact developed specifically for sensor parameter initialisation has been described. It consists of a printed pattern of circles, with the possibility of scaling up to different measurement volumes. However, the three 3-D calibration artefacts, the multiple sphere, ball bar and two plane artefacts have been described in Chapters 4 and 5.

To obtain initial sensor parameters, the 2-D HT is used to estimate pixel coordinates of the circle centres from respective intensity images of the 2-D artefact. In conjunction with the physical dimensions of the CCD/SLM, these pixel coordinates are used to calculate respective sensor image plane coordinates from the acquired shape data. With knowledge of the respective sensor internal parameters, the calculated image plane coordinates are

used by the 2-D DLT method to obtain estimates of the external parameters. The initial sensor parameters are then used to compute point clouds from the acquired shape data of the 3-D artefact, from where the 3-D HT is used to detect either spheres or planes (and their corresponding pixel coordinates in the X, Y, Z matrices of the point cloud). The detected spheres or planes provide information for estimating the pose of the artefact, which consists of six parameters that provide the geometric transformation of the artefact's control points (i.e. the known control points of either a sphere centre or plane's foot-of-normal position, established in a local coordinate system by a mechanical CMM) to the global coordinate system established in the measurement volume. In addition, a robust algorithm is used for automatic and uniform selection of a subset of the pixel coordinates corresponding to scattering points on the surface of the spheres or planes. The estimated sensor and pose parameters are then refined in the bundle adjustment process through a non-linear optimisation.

The physical set up of the sensor and the geometry of the 3-D calibration artefact necessitates the introduction of constraints during bundle adjustment. It is required that the external parameters of one of the sensors should be set constant during bundle adjustment as a means of keeping the global coordinate system fixed. On the other hand, although the multiple sphere artefact does not require the application of constraints, for a ball bar artefact, the pose parameter referring to rotation about the Z axis, κ , should be constant. For the two plane artefact, translation along the X and Y axes (t_x and t_y), and rotation about the Z axis (κ), should be constant. Example artefact files have been shown, which describe the format for setting up a text file to which the control point coordinates of each artefact are written. During the calibration process, the coordinates of the known control points are read from the artefact files and used to estimate the pose of the artefact.

The software implementation of the calibration has been described in terms of the document for storing calibration settings and the software structure. XML has been selected as the means for storing the calibration settings. An XML document is a structured, human and computer readable document, providing a reliable and scalable means for initialising and updating calibration settings, without having to deal with

numerous configuration files. The structure of the calibration XML document has also been described, highlighting the main sub-elements. To provide operational efficiency, the calibration software has been implemented as a combination of MATLAB[®] function files and C mex files.

The new calibration process was used to calibrate a two-camera two-projector SMS for a 400×400×500 mm³ volume, using data from the 2-D artefact and subsequently, the three 3-D calibration artefacts, the multiple sphere, ball bar and two plane artefacts. The calibration was carried out in offline mode, whereby all respective measurements of the 2-D and 3-D artefacts were made and the acquired shape data saved to disk, with datasets processed afterwards to retrieve sensor parameters. Shape data for initialising estimates of the sensor parameters were obtained by measuring the 2-D artefact near the centre of the measurement volume. Data sets for refining sensor parameters were obtained by measuring a 3-D artefact in multiple poses in the measurement volume. The acquired shape data of the 2-D artefact was processed and used to successfully calculate initial estimates of the cameras and projector parameters. Also, the acquired shape data of the 3-D artefacts were automatically processed, with successful detection of the required number of artefact features, pose estimation and control point selection. The initial sensor parameters, pose parameters and control points were then used to carry out a bundle adjustment in object space.

With data from the multiple sphere artefact, results after bundle adjustment show a reduction in the value of the two contributing terms of the objective function. The ε_1 term reduced to 0.78 mm, while the ε_2 term reduced to 1.15 mm. However, the convergence criterion was not met and less than 40% reduction of the respective error terms was achieved after 150 iterations. With respect to the measurement volume, the final value of the ε_2 term represents an accuracy of just one part in 400. This rather poor performance was discovered to be a result of the translucent nature of the multiple sphere artefact. Calibration results using data from the ball bar artefact show reductions of up to 95% of the contributing terms of the objective function, with the convergence criterion met after 45 iterations. The calibration process was quicker as a result of processing just

two spheres per camera-projector. The ε_1 term reduced to 63 μm , while the ε_2 term reduced to 97 μm . With respect to the measurement volume, the final value of the ε_2 term represents an accuracy of around one part in 5,000. Finally, calibration results using data from the two plane artefact also show reductions of almost 90% of the contributing terms of the objective function, with the convergence criterion met after 34 iterations. The time to calibrate was comparable with the ball bar, as a result of processing just two planes per camera-projector pair and a quick convergence rate. The ε_1 term reduced to 127 μm , and the ε_2 term reduced to 228 μm . With respect to the measurement volume, the final value of the ε_2 term represents an accuracy of around one part in 2,600

The accuracy of the sensor parameters obtained using the different artefacts to calibrate the optical SMS was assessed by calculating the absolute difference between SMS measured and mechanical CMM lengths. These lengths include the distance between the spheres of the ball bar artefact and the perpendicular distance between the planes of the two plane artefact. In comparison with other 3-D artefacts, sensor parameters from using data from the ball bar artefact to calibrate the SMS show accuracies of 73.2 μm and 0.18 mm for these two lengths respectively. It is concluded that the ball bar artefact's low-cost, scalability, user friendliness, faster processing time and accuracy of sensor parameters after calibration justify its selection as the most suitable artefact for calibrating a multi-sensor optical SMS.

6.9 Tables

Table 6-1 : Thresholds for selecting valid pixels from acquired shape data of 3-D calibration artefacts

Camera-projector pair	Type of artefact					
	Multiple-sphere		Ball bar		Two-plane	
	Error threshold	Modulation threshold	Error threshold	Modulation threshold	Error threshold	Modulation threshold
C ₁ P ₁	0.1	0.1	0.2	0.2	0.5	0.05
C ₁ P ₂	0.1	0.1	0.2	0.2	0.5	0.05
C ₂ P ₁	0.1	0.1	0.2	0.1	0.5	0.05
C ₂ P ₂	0.1	0.1	0.2	0.1	0.5	0.05

Table 6-2 : Parameters for C₁ before and after calibration

Sensor Parameters	Initial estimates	Results from calibration – Type of Artefact		
		Multiple sphere	Ball bar	Two-plane
x (mm)	820.0390	820.0390	820.0390	820.0390
y (mm)	-83.0996	-83.0996	-83.0996	-83.0996
z (mm)	1936.3120	1936.3120	1936.3120	1936.3120
ω (radians)	-3.0583	-3.0583	-3.0583	-3.0583
ϕ (radians)	-0.3613	-0.3613	-0.3613	-0.3613
κ (radians)	-1.5421	-1.5421	-1.5421	-1.5421
ϵ_H (mm)	0.0	0.0426	0.4124	0.0298
η_H (mm)	0.0	-0.1302	-0.0616	0.0024
c (mm)	35.0	35.1148	37.1330	37.5713
k ₁ (dimensionless)	0.0	-27.80×10^{-6}	3.25×10^{-6}	1.02×10^{-6}
k ₂ (dimensionless)	0.0	0.0	0.0	0.0
k ₃ (dimensionless)	0.0	0.0	0.0	0.0

Table 6-3 : Parameters for C_2 before and after calibration

Sensor Parameters	Initial estimates	Results from calibration – Type of Artefact		
		Multiple sphere	Ball bar	Two-plane
x (mm)	-648.6690	-657.1725	-726.8967	-708.1289
y (mm)	-90.3212	-90.3179	-86.7674	-91.8896
z (mm)	1898.3767	1906.4520	1933.1206	1931.2081
ω (radians)	-3.0532	-3.0531	-3.0512	-3.0521
ϕ (radians)	0.3487	0.3447	0.3656	0.3726
κ (radians)	1.5313	1.5318	1.5368	1.5305
ϵ_H (mm)	0.0	-0.0226	-0.2506	-0.0042
η_H (mm)	0.0	-0.0286	0.8070	1.2935
c (mm)	35.0	35.2536	37.7885	38.1396
k_1 (dimensionless)	0.0	-22.01×10^{-6}	1.49×10^{-6}	1.08×10^{-6}
k_2 (dimensionless)	0.0	0.0	0.0	0.0
k_3 (dimensionless)	0.0	0.0	0.0	0.0

Table 6-4 : Parameters for P_1 before and after calibration

Sensor Parameters	Initial estimates	Results from calibration – Type of Artefact		
		Multiple sphere	Ball bar	Two-plane
x (mm)	779.1982	784.8274	847.9029	843.4840
y (mm)	677.7406	687.6412	778.4281	767.9064
z (mm)	1797.8695	1812.9493	1960.1406	1946.0030
ω (radians)	3.0402	3.0343	3.0271	3.0286
ϕ (radians)	-0.3562	-0.3576	-0.3562	-0.3651
κ (radians)	1.5349	1.5321	1.5289	1.5293
ϵ_H (mm)	6.0	5.9682	6.3575	6.7201
η_H (mm)	0.0	0.0926	0.1859	-0.1661
c (mm)	30.0	30.7163	35.2137	35.3900
k_1 (dimensionless)	0.0	11.96×10^{-6}	-5.37×10^{-6}	5.05×10^{-6}
k_2 (dimensionless)	0.0	0.0	0.0	0.0
k_3 (dimensionless)	0.0	0.0	0.0	0.0

Table 6-5 : Parameters for P₂ before and after calibration

Sensor Parameters	Initial estimates	Results from calibration – Type of Artefact		
		Multiple sphere	Ball bar	Two-plane
x (mm)	-615.3461	-629.6728	-743.4892	-727.8925
y (mm)	670.7147	681.4251	777.2442	761.1229
z (mm)	1799.5872	1820.7499	1961.4765	1949.8211
ω (radians)	3.0456	3.0427	3.0286	3.0306
ϕ (radians)	0.3497	0.3452	0.3586	0.3653
κ (radians)	1.5883	1.5894	1.5986	1.5908
ϵ_H (mm)	6.0	6.0692	6.3715	6.6476
η_H (mm)	0.0	-0.0535	0.4140	0.7967
c (mm)	30.0	30.7434	35.1998	35.3730
k_1 (dimensionless)	0.0	3.97×10^{-6}	2.75×10^{-6}	4.84×10^{-6}
k_2 (dimensionless)	0.0	0.0	0.0	0.0
k_3 (dimensionless)	0.0	0.0	0.0	0.0

Table 6-6 : Estimated pose parameters for artefact pose 3 of multiple-sphere artefact before and after calibration.

Camera- Projector Pair	Initial Sensor Parameters						Optimised Sensor Parameters					
	Euler Angles (radians)			Translation Vector (mm)			Euler Angles (radians)			Translation Vector (mm)		
	ω	ϕ	κ	t_x	t_y	t_z	ω	ϕ	κ	t_x	t_y	t_z
C₁P₁	-0.0468	0.0041	-0.0035	88.4428	74.3989	49.2120	-0.0287	0.0067	-0.0062	80.6286	76.6999	42.9517
C₁P₂	-0.0326	-0.0034	-0.0042	88.3458	73.4565	50.4448	-0.0292	0.0099	-0.0051	79.8464	75.4270	42.4575
C₂P₁	-0.0320	0.0174	-0.0075	89.9017	73.4925	49.7629	-0.0305	0.0132	-0.0032	81.2941	75.3767	41.9969
C₂P₂	-0.0441	0.0141	-0.0086	89.2041	74.7924	50.4148	-0.0242	0.0153	-0.0055	80.6368	76.5524	42.7044
Standard Deviation	7.7 $\times 10^{-3}$	9.5 $\times 10^{-3}$	2.5 $\times 10^{-3}$	0.728	0.667	0.588	2.7 $\times 10^{-3}$	3.8 $\times 10^{-3}$	1.3 $\times 10^{-3}$	0.592	0.709	0.407

Table 6-7 : Estimated pose parameters for artefact pose 3 of ball bar artefact before and after calibration.

Camera-Projector Pair	Initial Sensor Parameters						Optimised Sensor Parameters					
	Euler Angles (radians)			Translation Vector (mm)			Euler Angles (radians)			Translation Vector (mm)		
	ω	ϕ	κ	t_x	t_y	t_z	ω	ϕ	κ	t_x	t_y	t_z
C₁P₁	1.5190	0.0156	0.0000	98.2394	258.4274	18.0152	-1.5947	-0.0057	0.0000	61.1466	-77.9612	-65.5117
C₁P₂	1.5339	0.0084	0.0000	100.0830	258.7870	21.9696	-1.5955	-0.0059	0.0000	61.1692	-77.9867	-65.3478
C₂P₁	1.5343	0.0080	0.0000	99.4988	258.8816	20.9037	-1.5953	-0.0058	0.0000	61.2381	-77.9555	-65.3816
C₂P₂	1.5202	0.0012	0.0000	100.8054	258.7858	19.0834	-1.5949	-0.0063	0.0000	61.2951	-77.8856	-65.4837
Standard Deviation	8.4 $\times 10^{-3}$	5.9 $\times 10^{-3}$	-	1.085	0.200	1.777	0.37 $\times 10^{-3}$	0.26 $\times 10^{-3}$	-	0.068	0.043	0.079

Table 6-8 : Estimated pose parameters for artefact pose 3 of two-plane artefact before and after calibration.

Camera-Projector Pair	Initial Sensor Parameters						Optimised Sensor Parameters					
	Euler Angles (radians)			Translation Vector (mm)			Euler Angles (radians)			Translation Vector (mm)		
	ω	ϕ	κ	t_x	t_y	t_z	ω	ϕ	κ	t_x	t_y	t_z
C_1P_1	0.2180	-0.2488	0.0000	-174.8450	-148.7936	671.7979	0.2347	-0.2461	0.0000	-151.4883	-140.2448	586.4599
C_1P_2	0.2301	-0.2582	0.0000	-180.0715	-155.5102	663.8161	0.2358	-0.2457	0.0000	-151.2093	-140.8745	586.2461
C_2P_1	0.2294	-0.2426	0.0000	-170.3471	-156.5064	670.1742	0.2342	-0.2465	0.0000	-151.7001	-139.9351	586.4163
C_2P_2	0.2154	-0.2457	0.0000	-173.2906	-147.7116	675.0475	0.2333	-0.2463	0.0000	-151.5894	-139.4517	586.7245
Standard Deviation	7.6 $\times 10^{-3}$	6.7 $\times 10^{-3}$	-	4.074	4.518	4.719	1.0 $\times 10^{-3}$	0.34 $\times 10^{-3}$	-	0.210	0.596	0.198

6.10 Figures

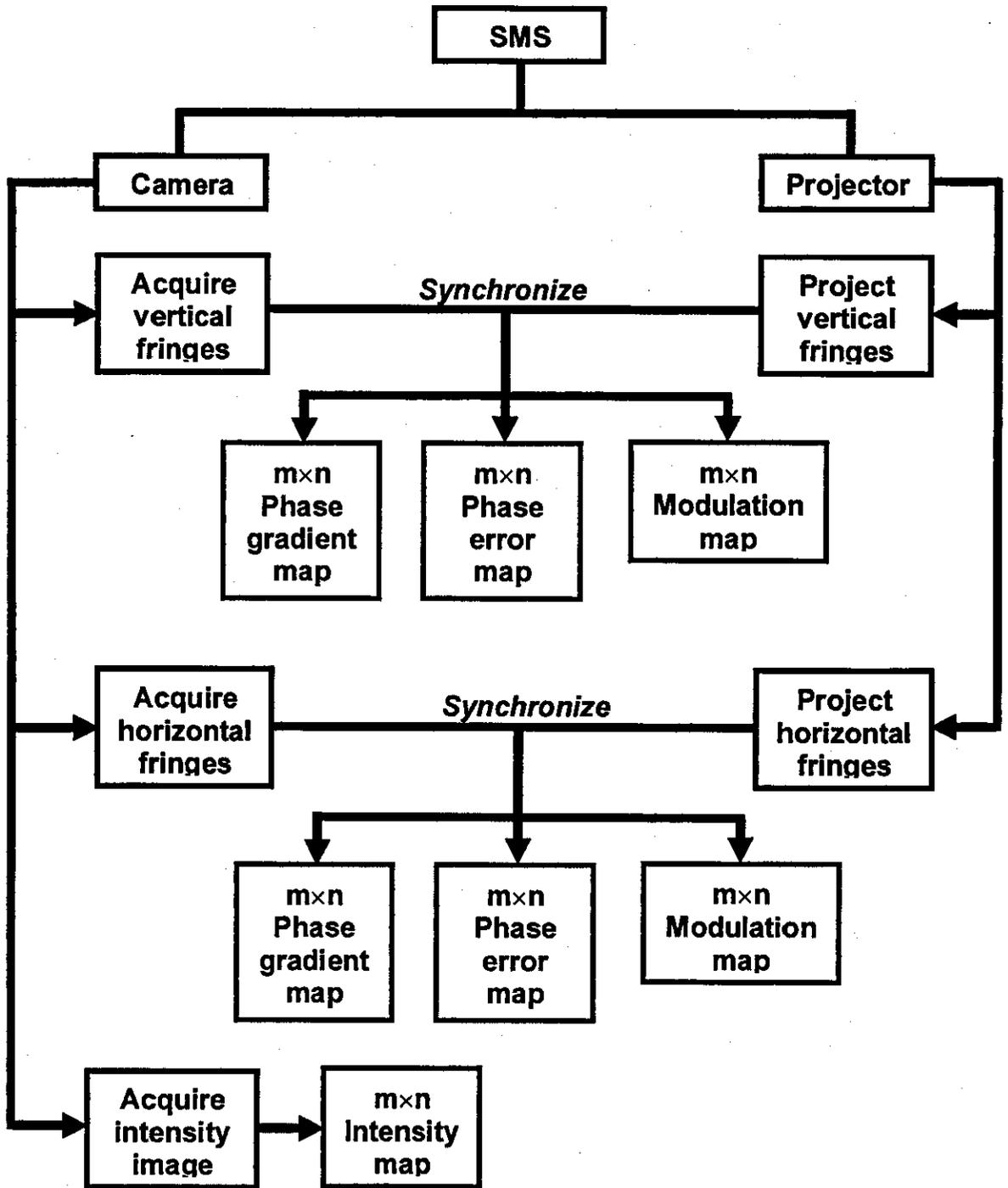


Figure 6-1: Process of shape data acquisition. Deliverables are indicated by boxes with blue outline.

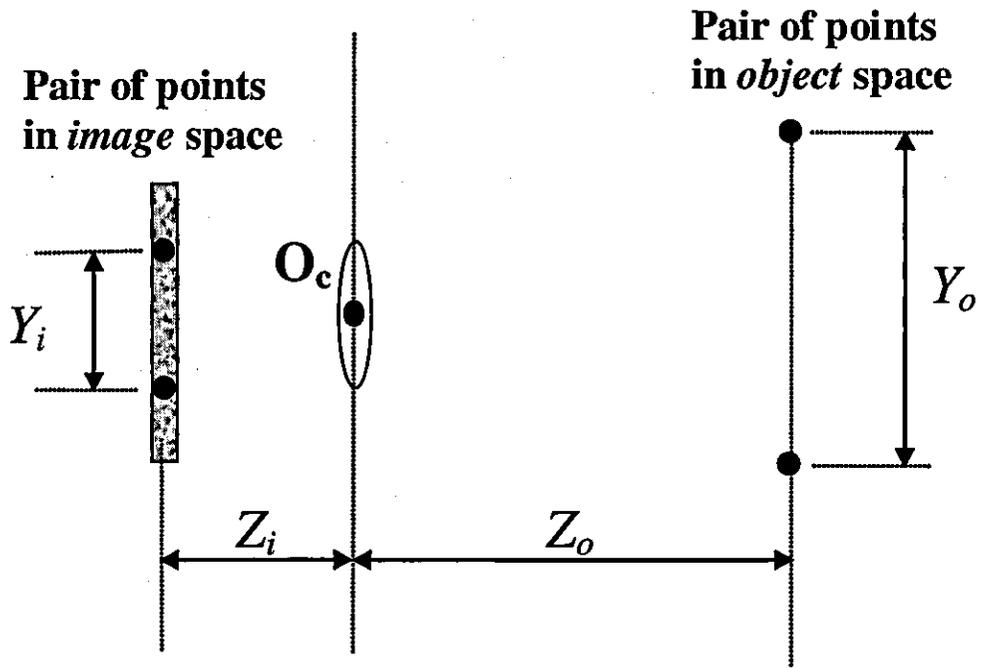


Figure 6-2: Magnification, m , of a pair of image points to estimate focal length, c .

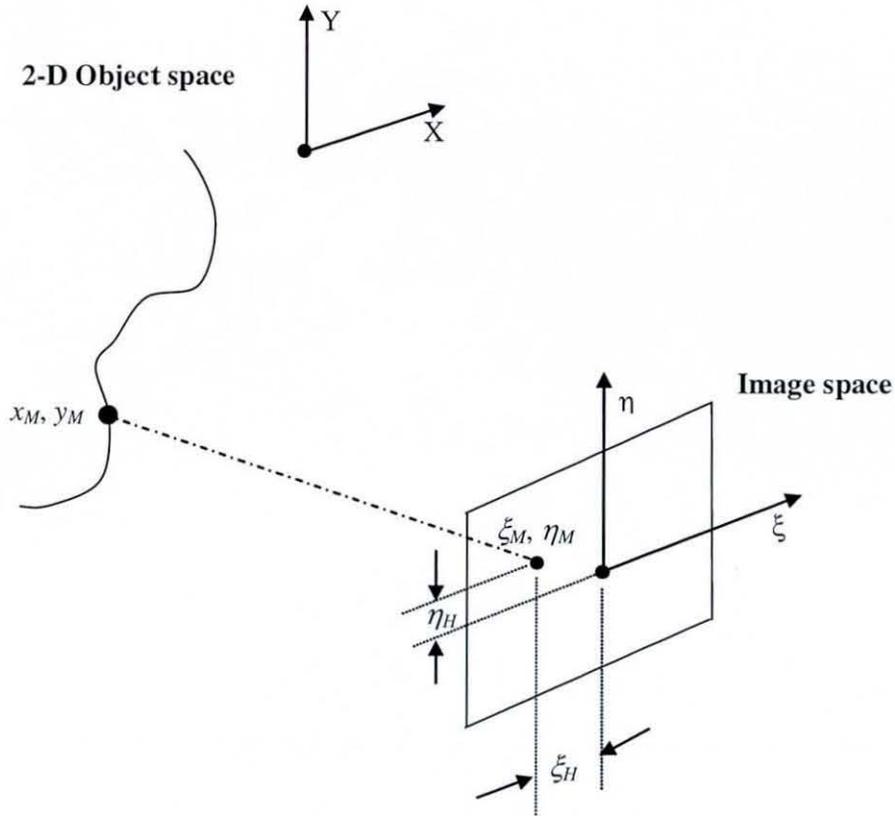


Figure 6-3: Relationship between a point in 2-D object space and image space.

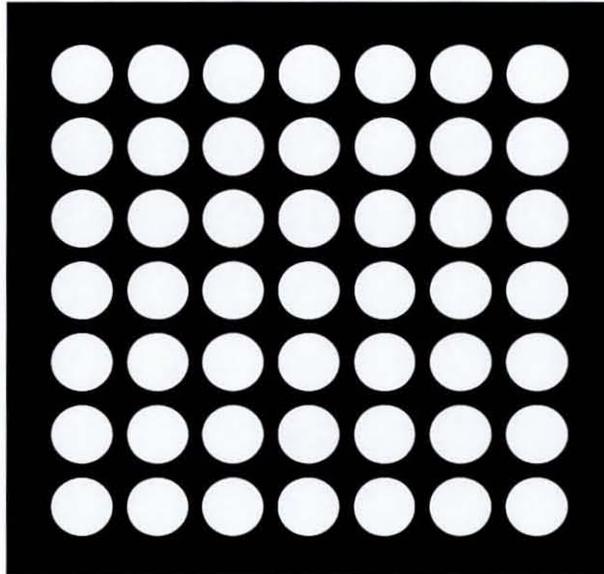


Figure 6-4: Image of 7x7 circles which could be printed and used as 2-D artefact for getting initial sensor parameters for the SMS.

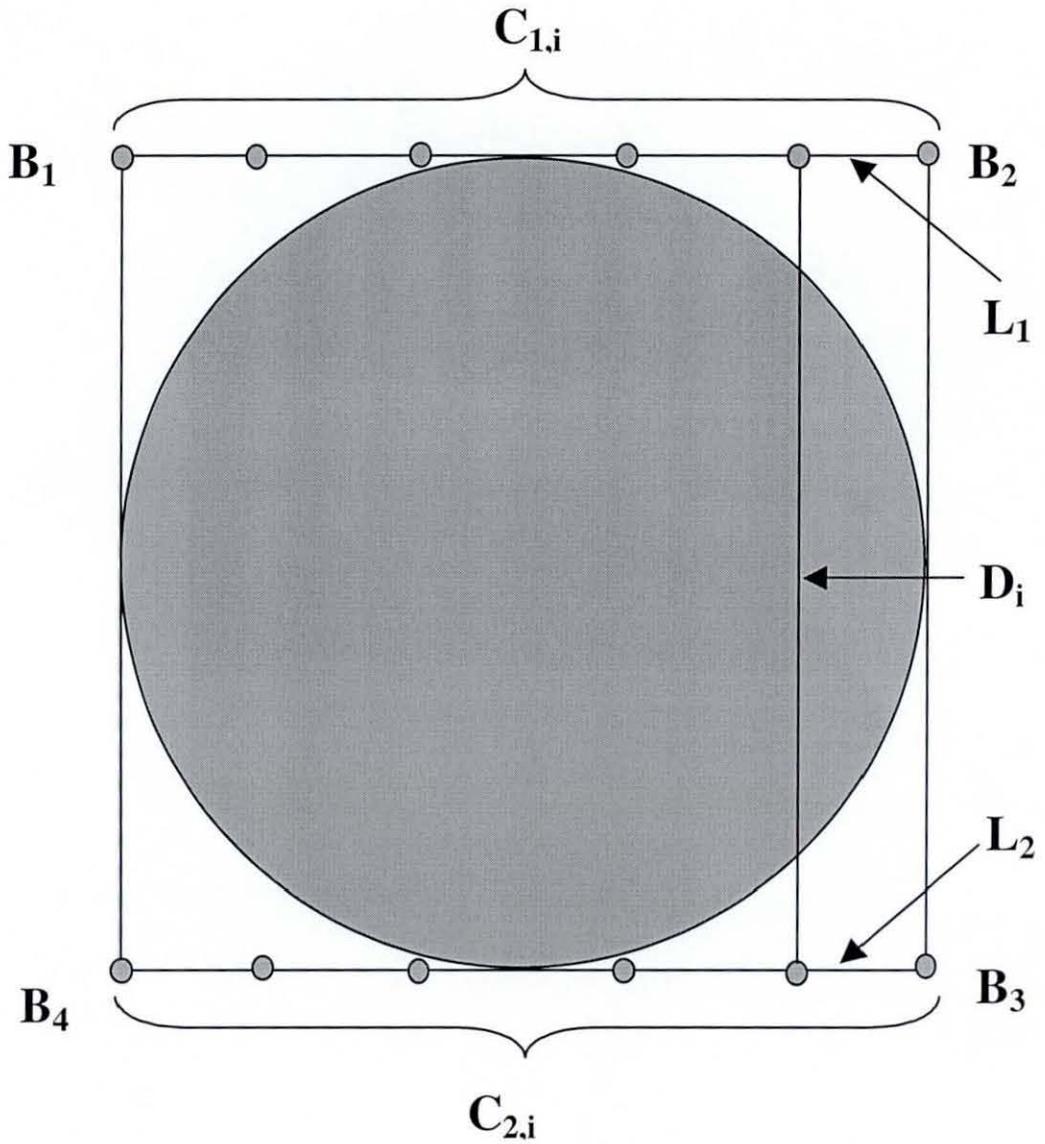


Figure 6-5: Process for automatic control point selection showing vertices of quadrilateral enclosing ROI and two sides L_1 and L_2 . $C_{1,i}$ and $C_{2,i}$ are the start and end points of respective lines where pixels would be selected.

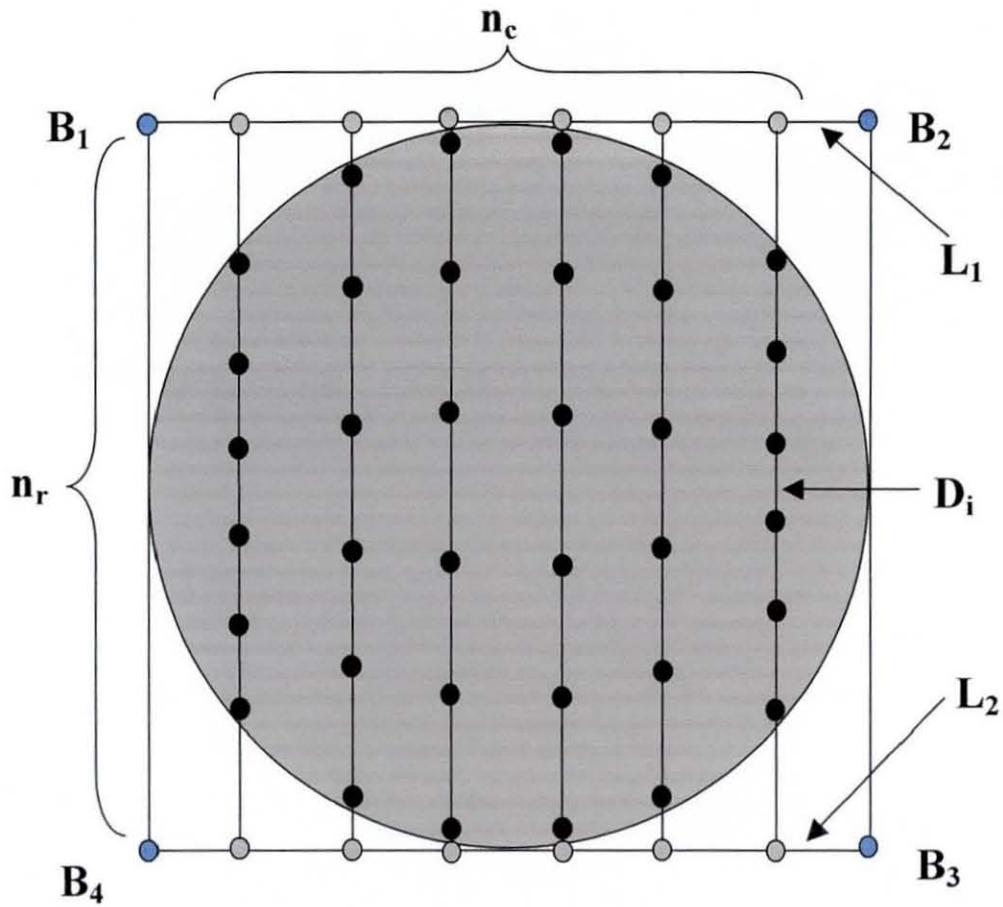


Figure 6-6: Process for automatic control point selection showing evenly selected lines, D_i , with $C_{1,i}$ and $C_{2,i}$ as coordinates of the start and end points. $n_r \times n_c$ pixels, E_{ij} (i.e. black circles) are evenly selected on D_i within circular ROI.

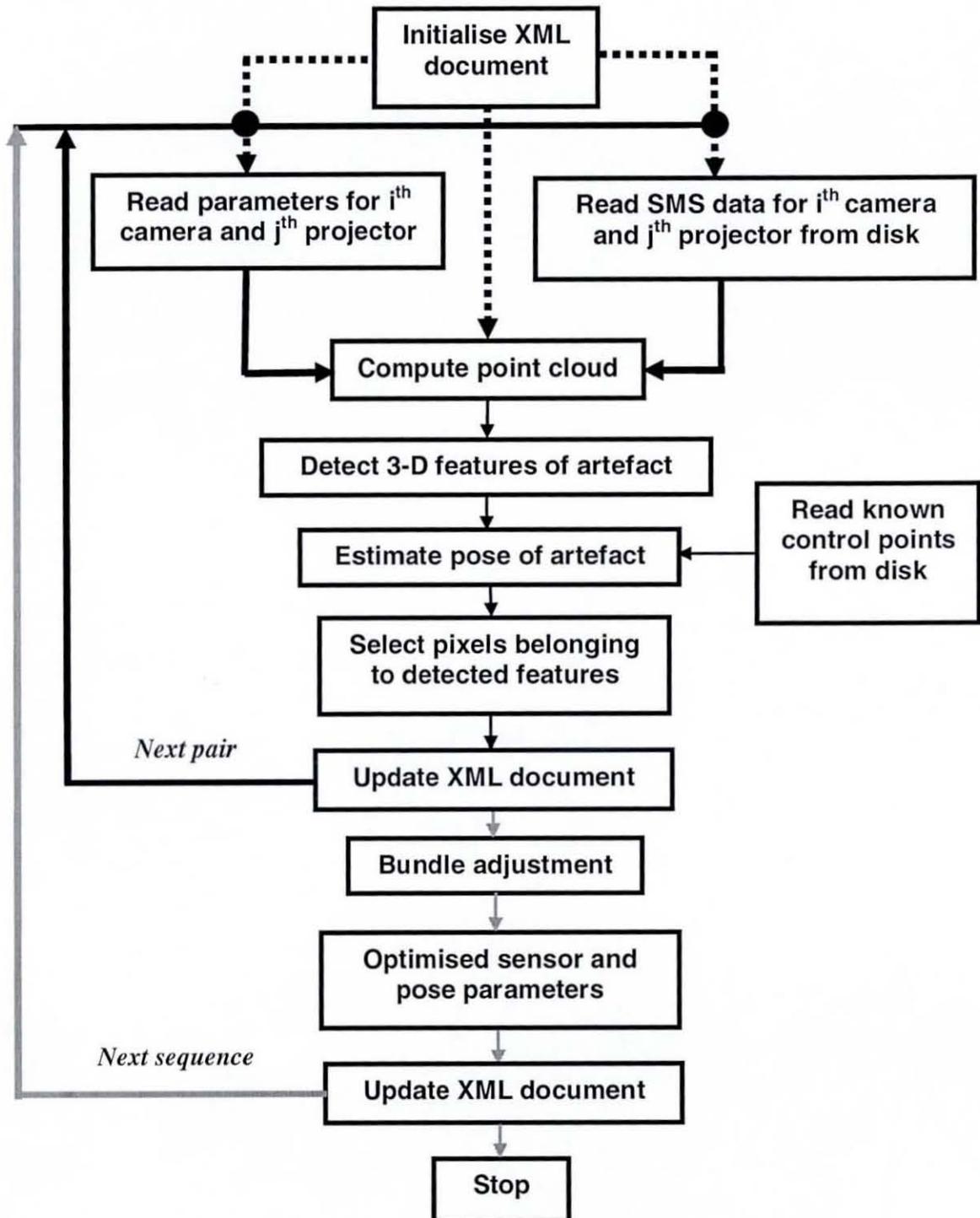


Figure 6-7: Software implementation of calibration process.

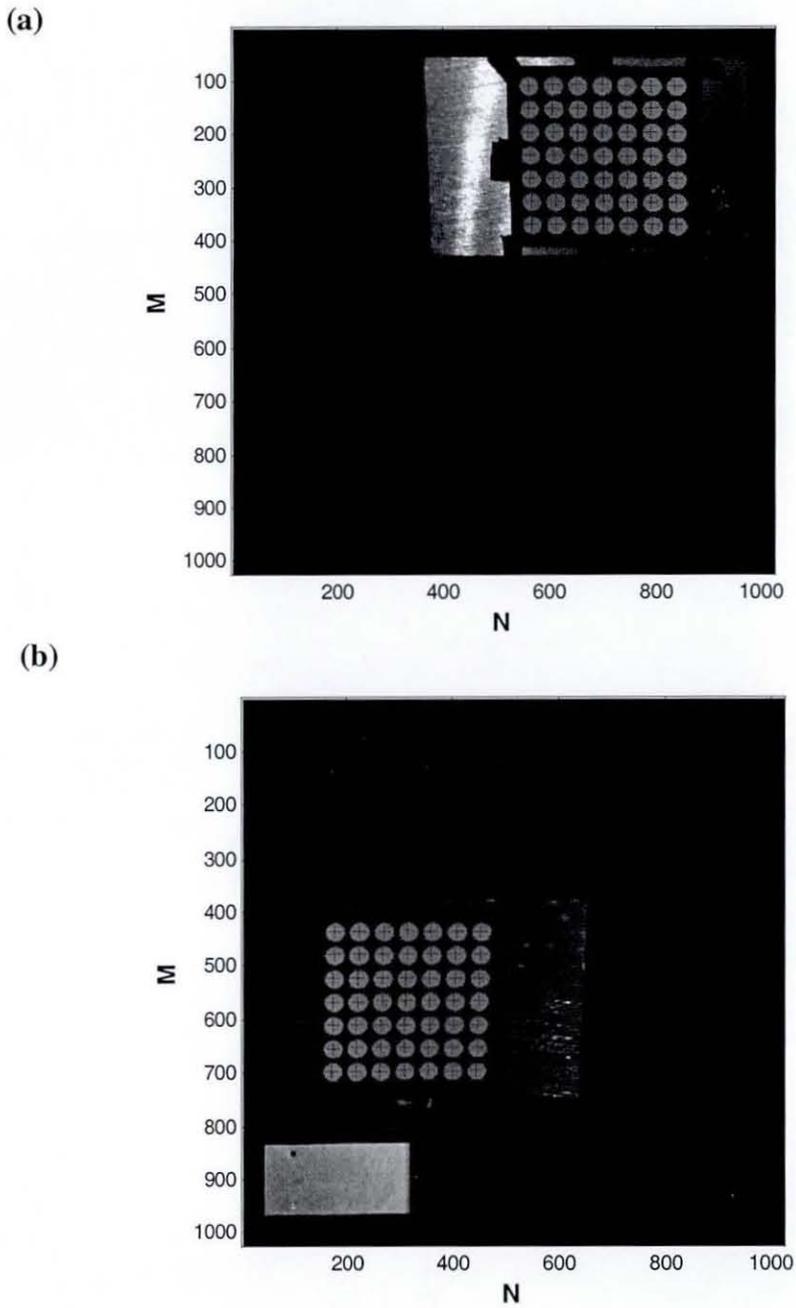


Figure 6-8: Greyscale intensity images of 2-D calibration artefact used for determining initial estimates of external sensor parameters using the 2-D DLT. Crosses indicate the circle centre coordinates detected by the 2-D HT. (a) Acquired image from C_1 ; (b) Acquired image from C_2 .

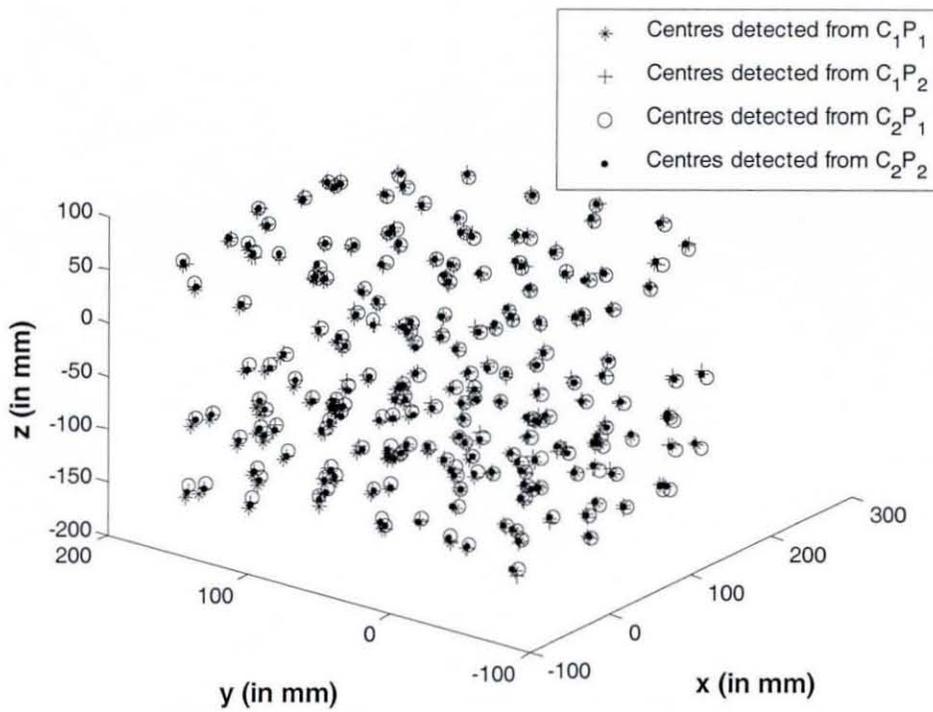
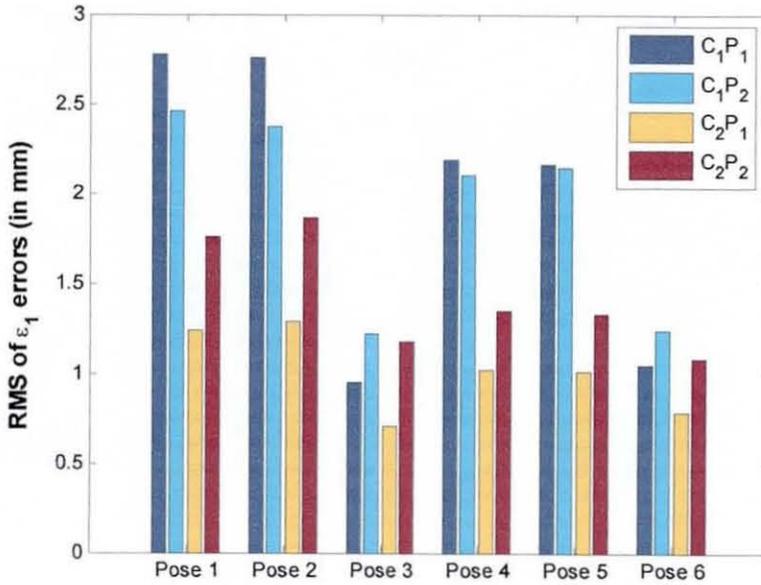


Figure 6-9: Location of the sphere centres detected from point cloud computed using initial estimates of sensor parameters.

(a)



(b)

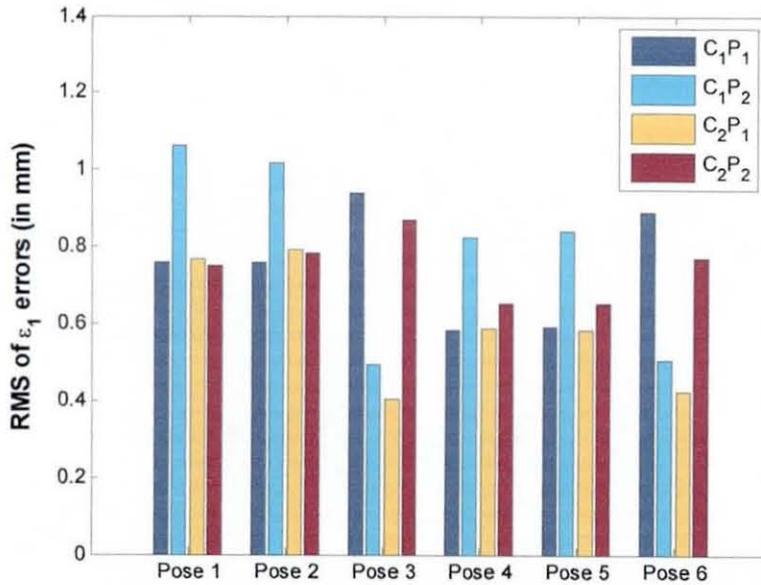
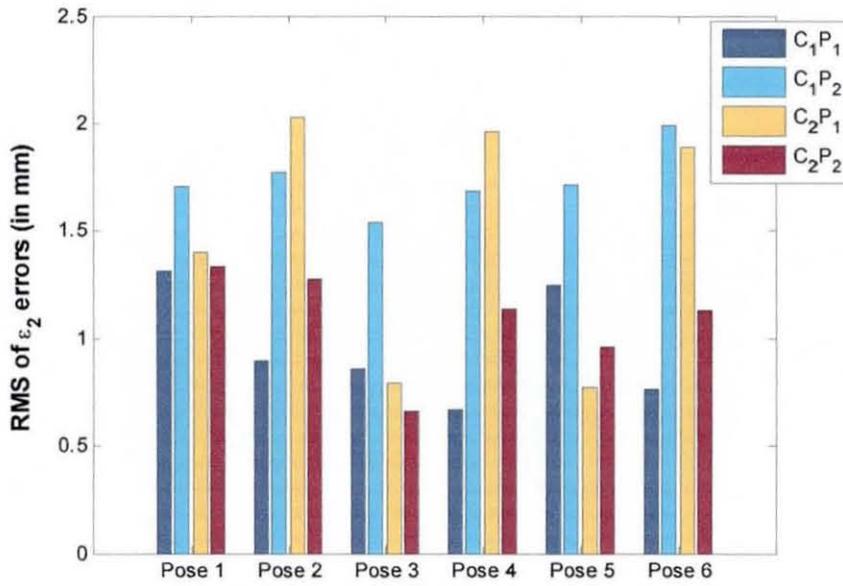


Figure 6-10: RMS of ε_1 values as calculated using acquired data of poses of the multiple-sphere artefact and sensor parameters. (a) Each column represents the RMS of ε_1 values using data from respective camera-projector pairs and initial sensor parameters. (b) Each column represents the RMS of ε_1 values using data from respective camera-projector pairs and optimised sensor parameters.

(a)



(b)

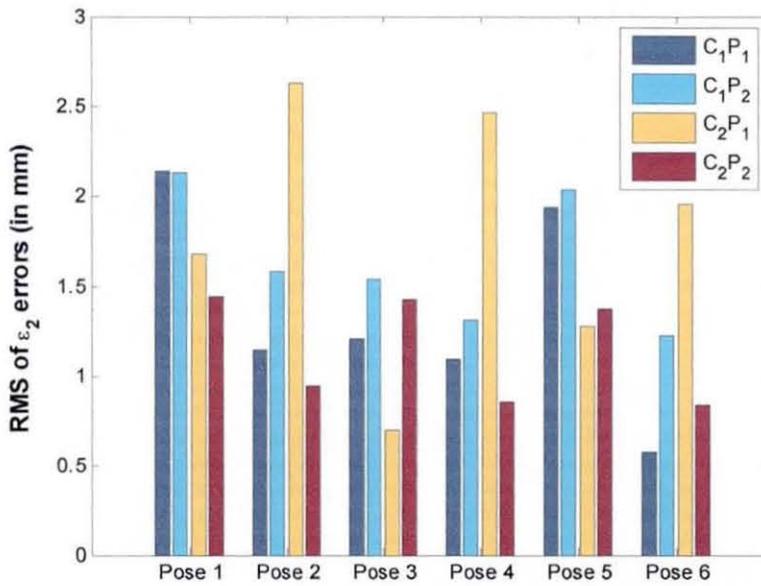
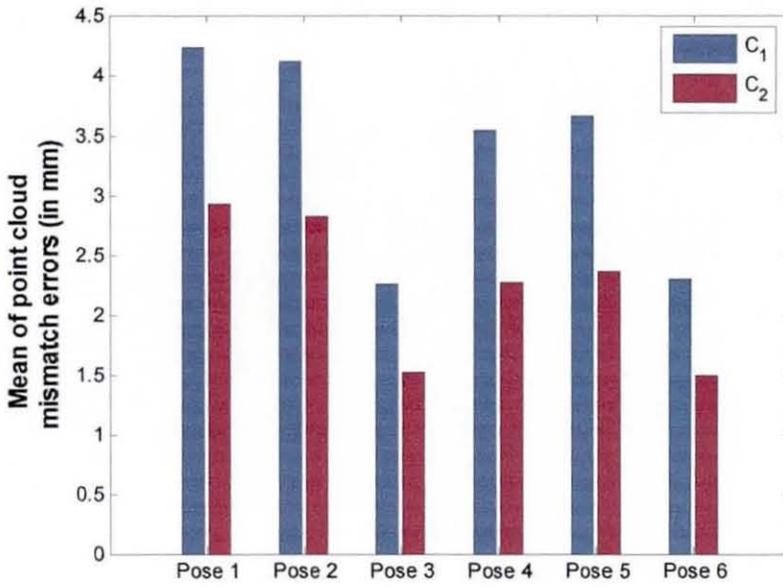


Figure 6-11: RMS of ε_2 values as calculated using acquired data of poses of the multiple-sphere artefact and sensor parameters. (a) Each column represents the RMS of ε_2 values using data from respective camera-projector pairs and initial sensor parameters. (b) Each column represents the RMS of ε_2 values using data from respective camera-projector pairs and optimised sensor parameters.

(a)



(b)

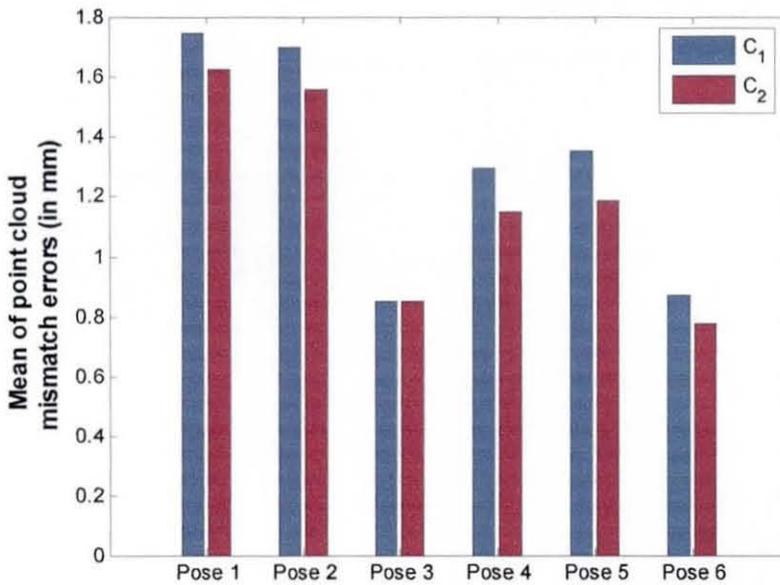


Figure 6-12: Mean of point cloud mismatch error values as calculated using acquired data of poses of the multiple-sphere artefact and sensor parameters. (a) Each column represents the mean of point cloud mismatch error values using data from respective cameras and initial sensor parameters. (b) Each column represents the mean of point cloud mismatch error values using data from respective cameras and optimised sensor parameters.

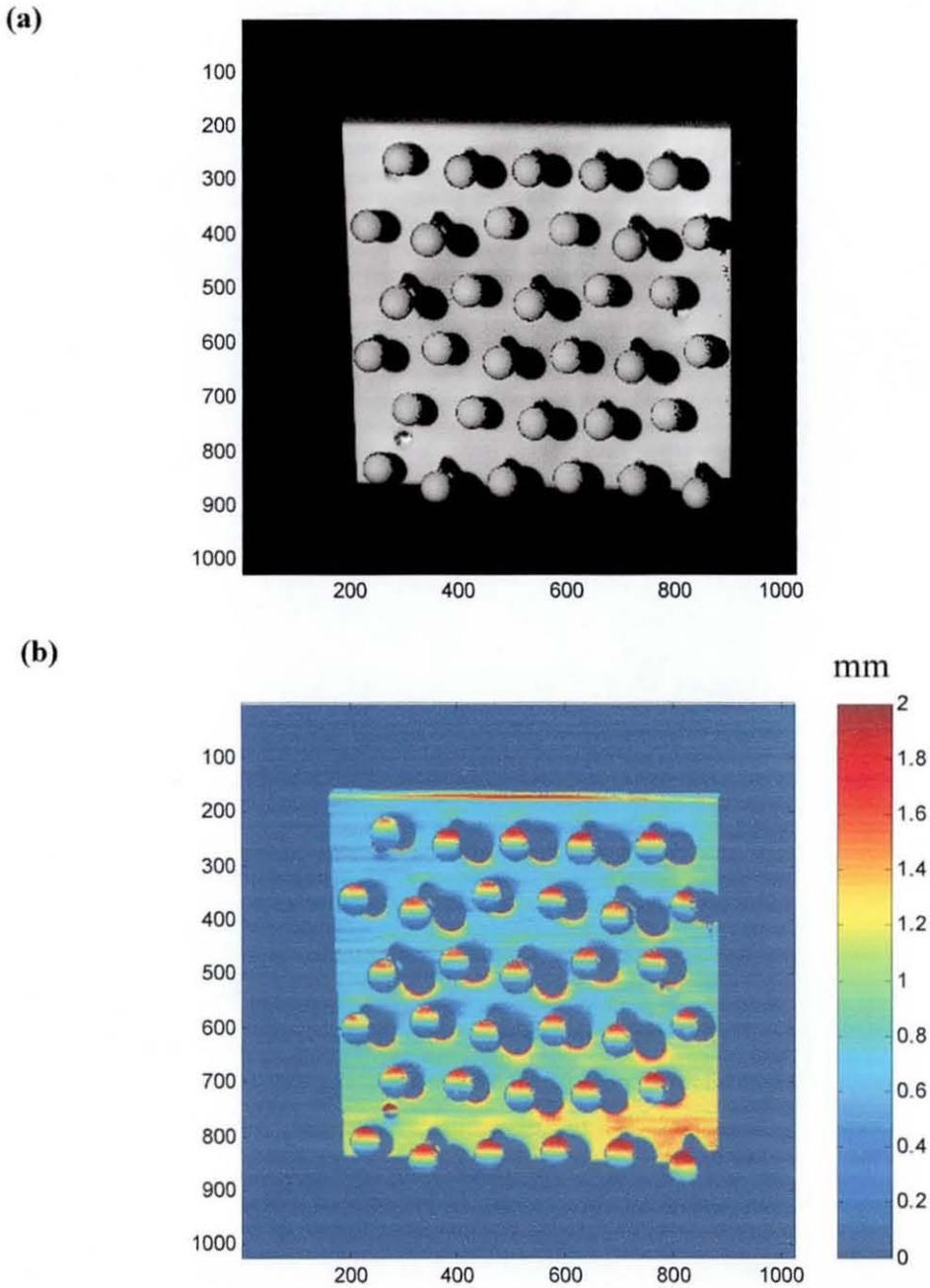


Figure 6-13: Multiple-sphere artefact consisting of 33 spheres on stalks set on a $270 \times 270 \text{ mm}^2$ base plate. (a) Greyscale intensity image of multiple-sphere artefact at pose 3 (acquired by C_1P_1), showing intensity values of valid pixels; (b) Image showing distribution of ε_1 values of valid pixels calculated using parameters of C_1 and P_1 on the acquired shape data. The gradient in ε_1 across pixels corresponding to scattering points on each sphere surface is indicative of systematic errors in acquired shape data.

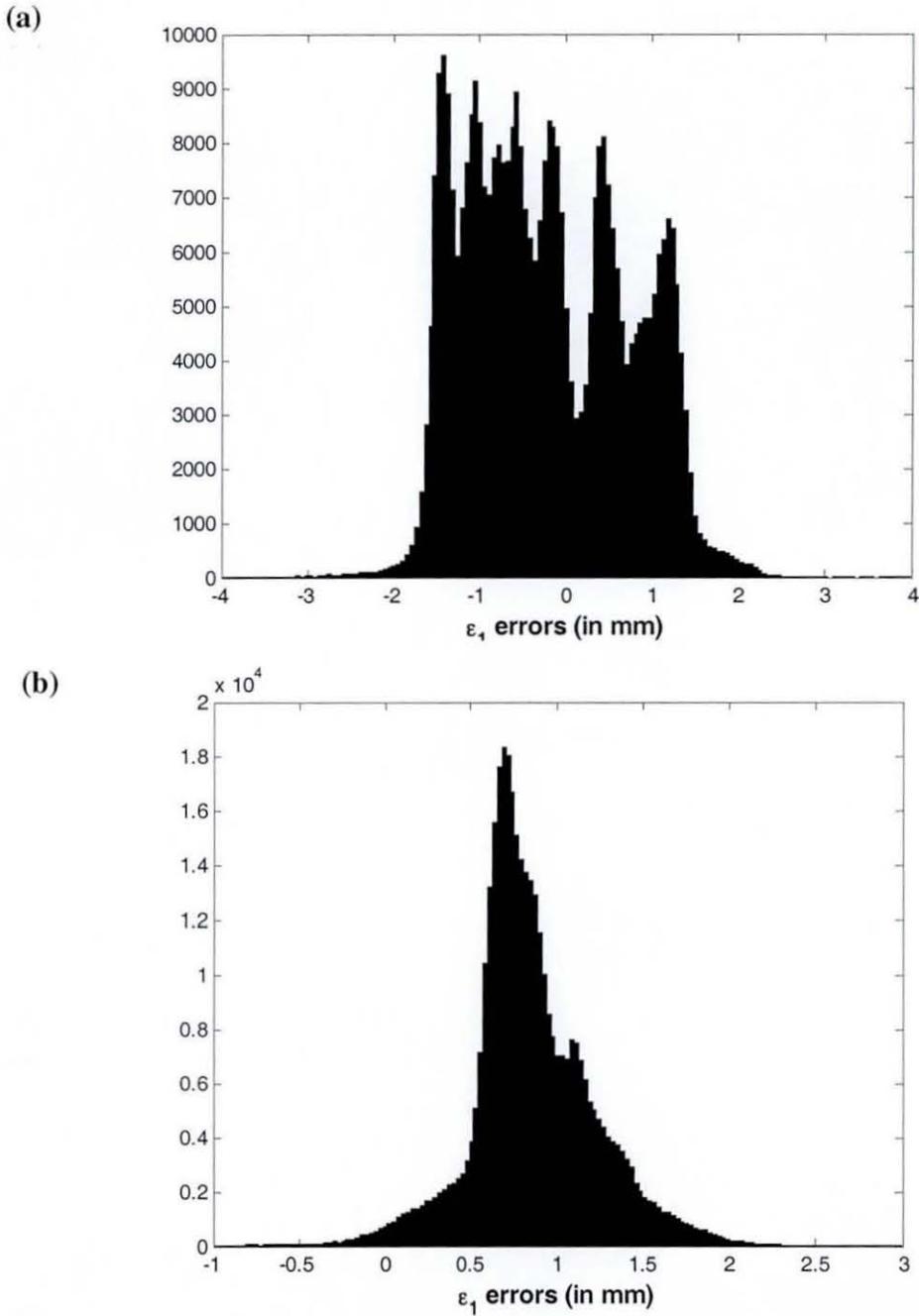


Figure 6-14: Histogram of 100 bins for ϵ_1 values of valid pixels calculated using parameters of C_1 and P_1 on acquired shape data of pose 3 of the multiple-sphere artefact. The vertical axis represents the number of votes in each bin. (a) Histogram with values calculated using initial sensor parameters; (b) Histogram with values calculated using optimised sensor parameters.

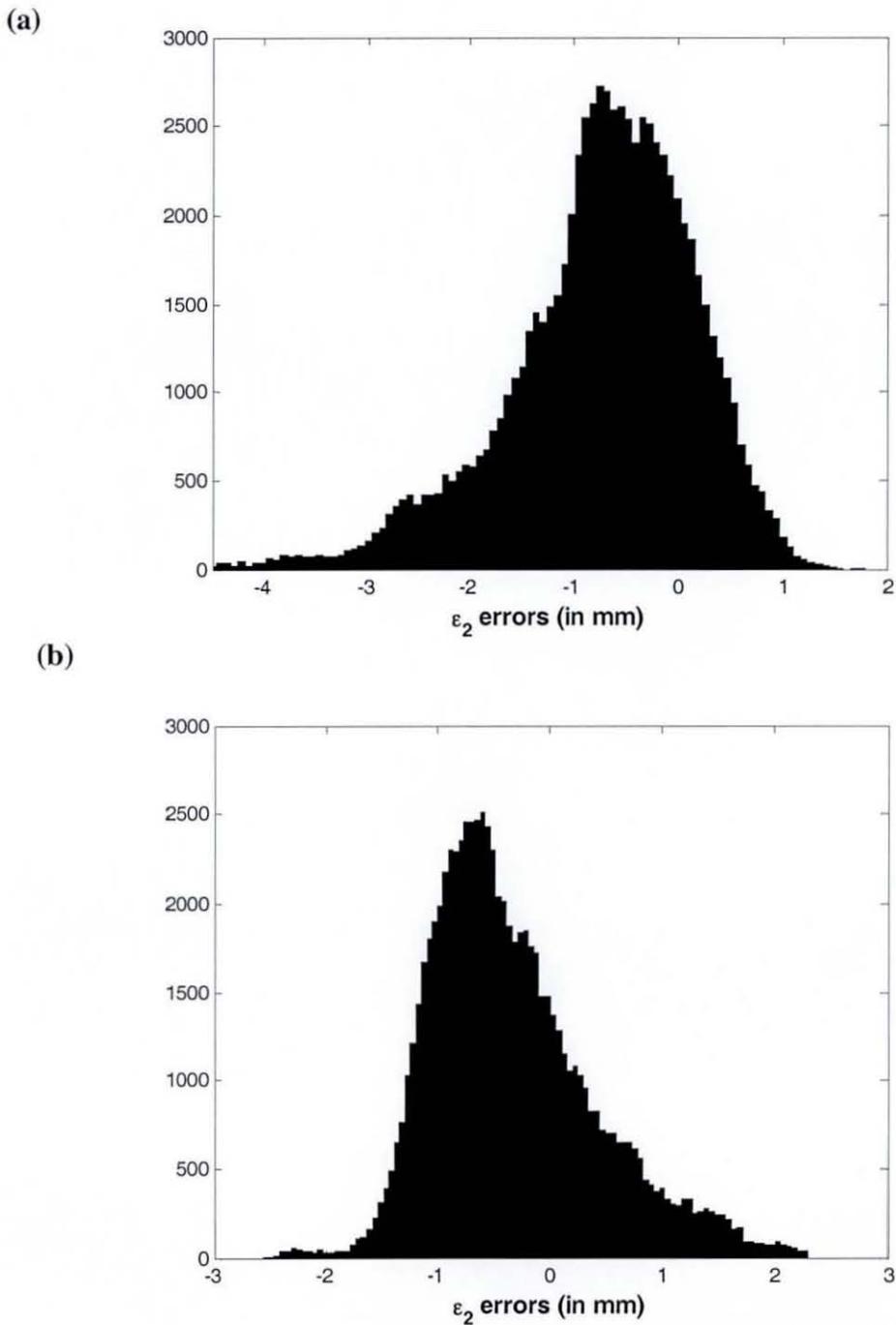


Figure 6-15: Histogram of 100 bins for ϵ_2 values of valid pixels calculated using parameters of C_1 and P_1 on acquired shape data of pose 3 of the multiple-sphere artefact. The vertical axis represents the number of votes in each bin. (a) Histogram with values calculated using initial sensor parameters; (b) Histogram with values calculated using optimised sensor parameters.

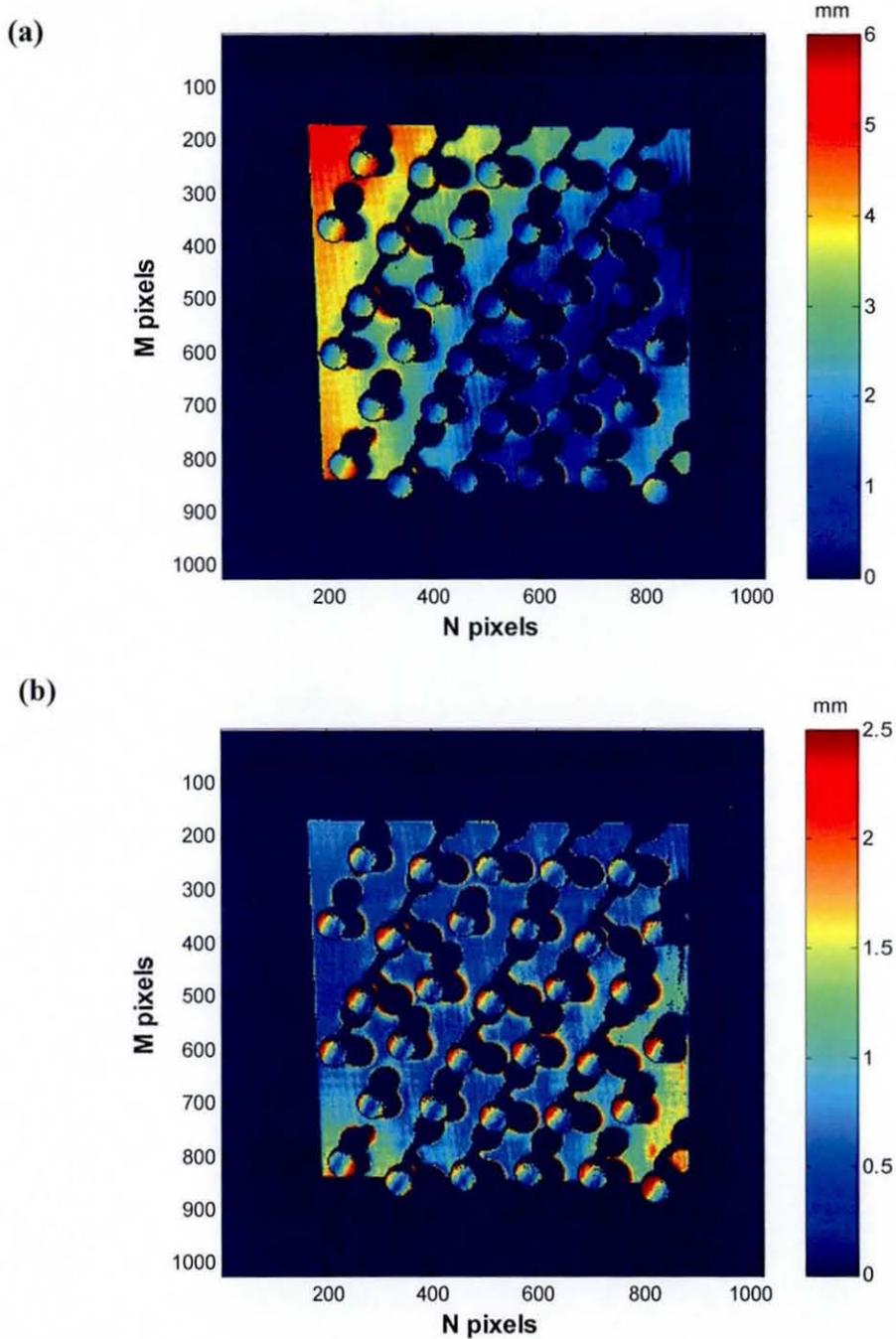


Figure 6-16: Point cloud mismatch error of pose 3 of the multiple-sphere artefact acquired by C_1 , i.e. difference in 3-D coordinates calculated for valid pixels common to acquired shape data of C_1P_1 and C_1P_2 . The respective point clouds were calculated using the acquired shape data and parameters of C_1 , P_1 , and P_2 . (a) Difference when initial sensor parameters are used; (b) Difference when optimised sensor parameters are used.

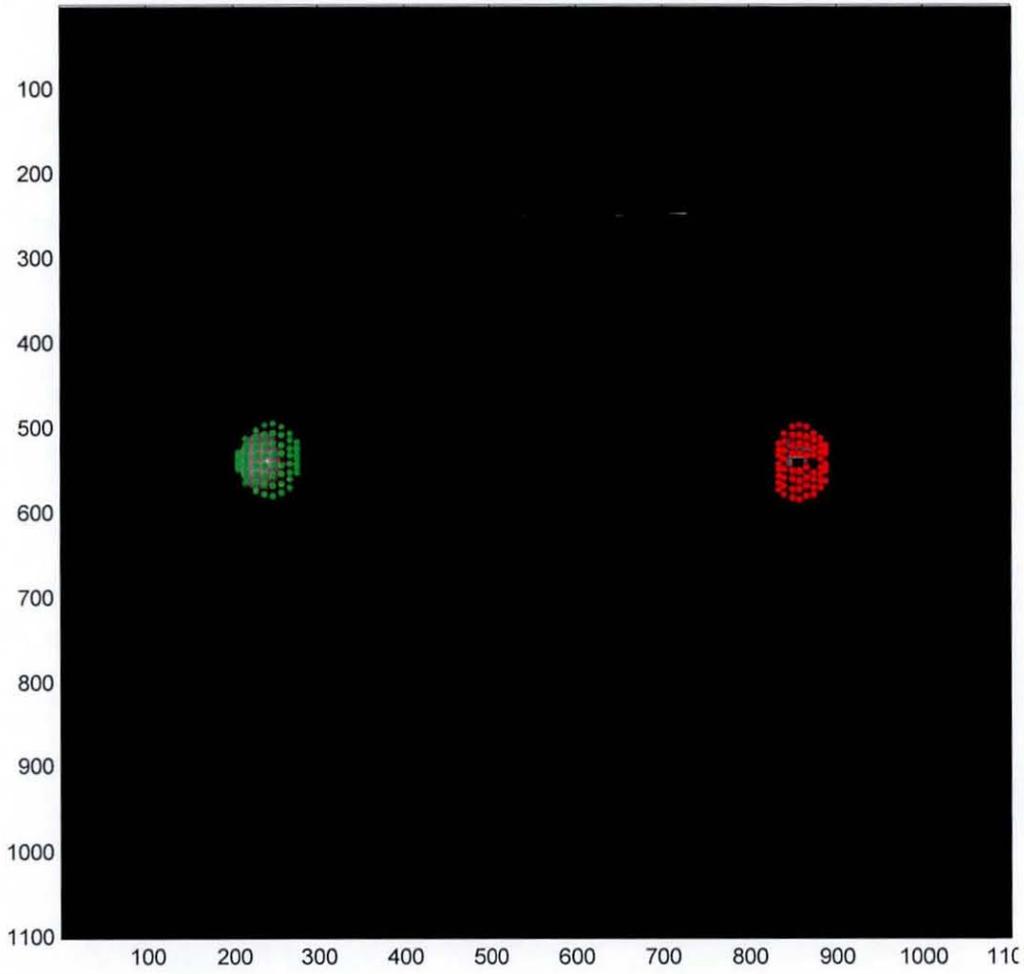
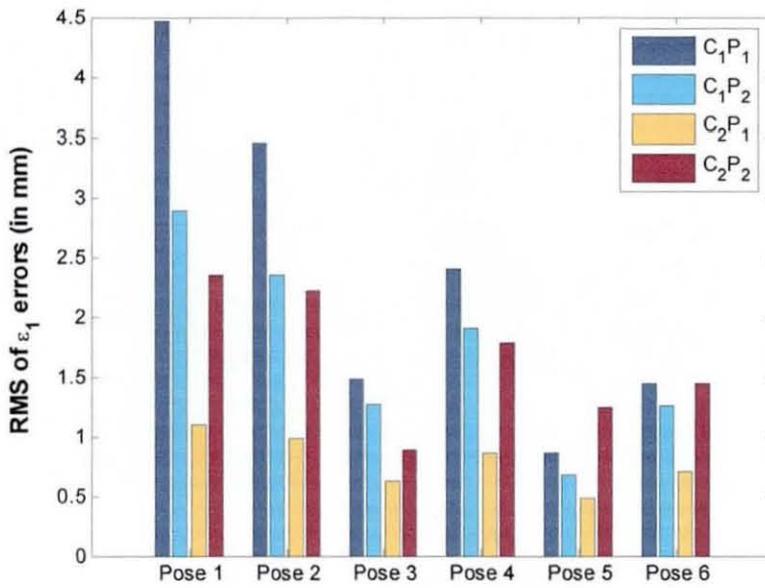


Figure 6-18: Greyscale intensity image of ball bar artefact at pose 3 (acquired by C_1P_1), showing intensity values of valid pixels. The pixels displayed in red and green correspond to pixels automatically selected for use in the bundle adjustment.

(a)



(b)

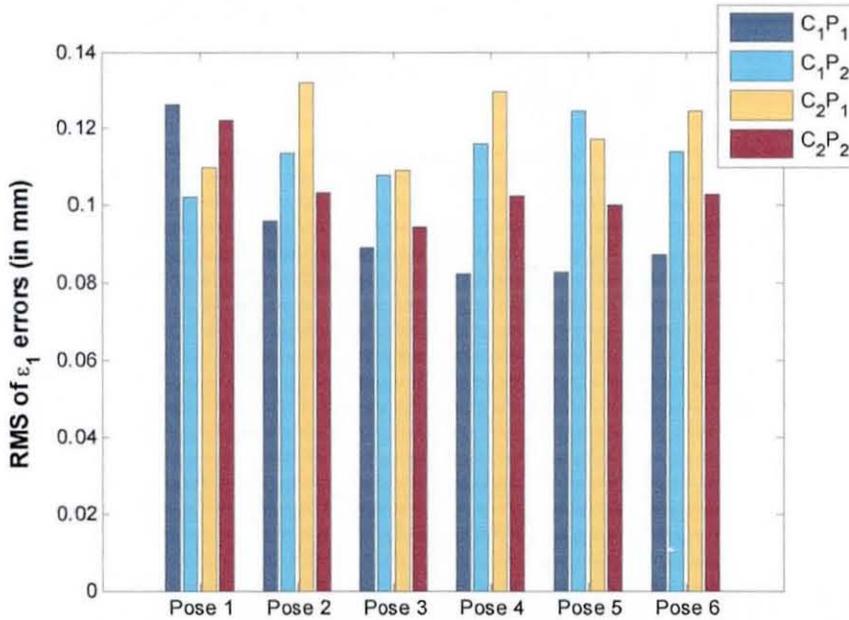


Figure 6-19: RMS of ε_1 values as calculated using acquired data of poses of the ball bar artefact and sensor parameters. (a) Each column represents the RMS of ε_1 values using data from respective camera-projector pairs and initial sensor parameters; (b) Each column represents the RMS of ε_1 values using data from respective camera-projector pairs and optimised sensor parameters.

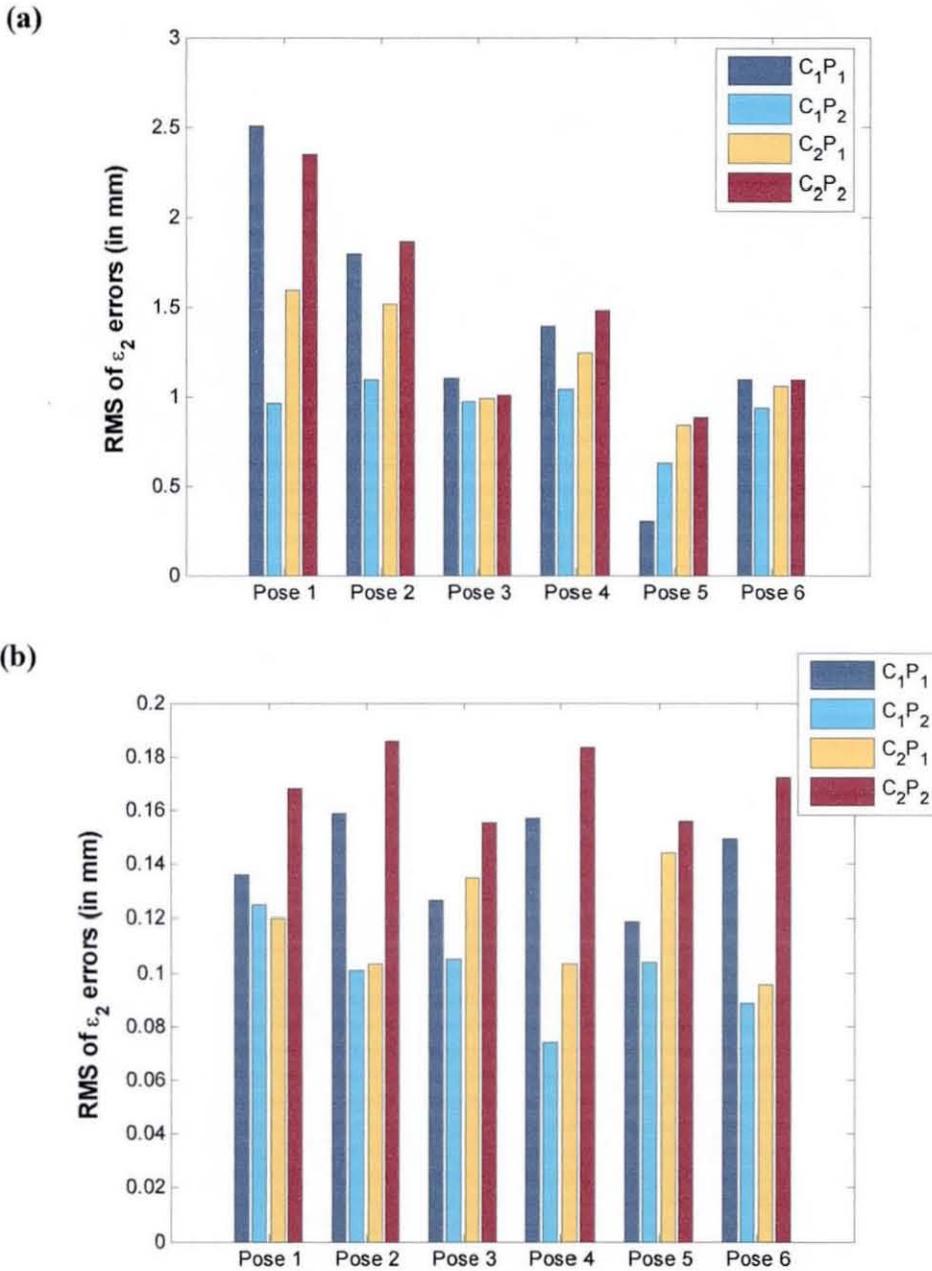


Figure 6-20: RMS of ε_2 values as calculated using acquired data of poses of the ball bar artefact and sensor parameters. (a) Each column represents the RMS of ε_2 values using data from respective camera-projector pairs and initial sensor parameters; (b) Each column represents the RMS of ε_2 values using data from respective camera-projector pairs and optimised sensor parameters.

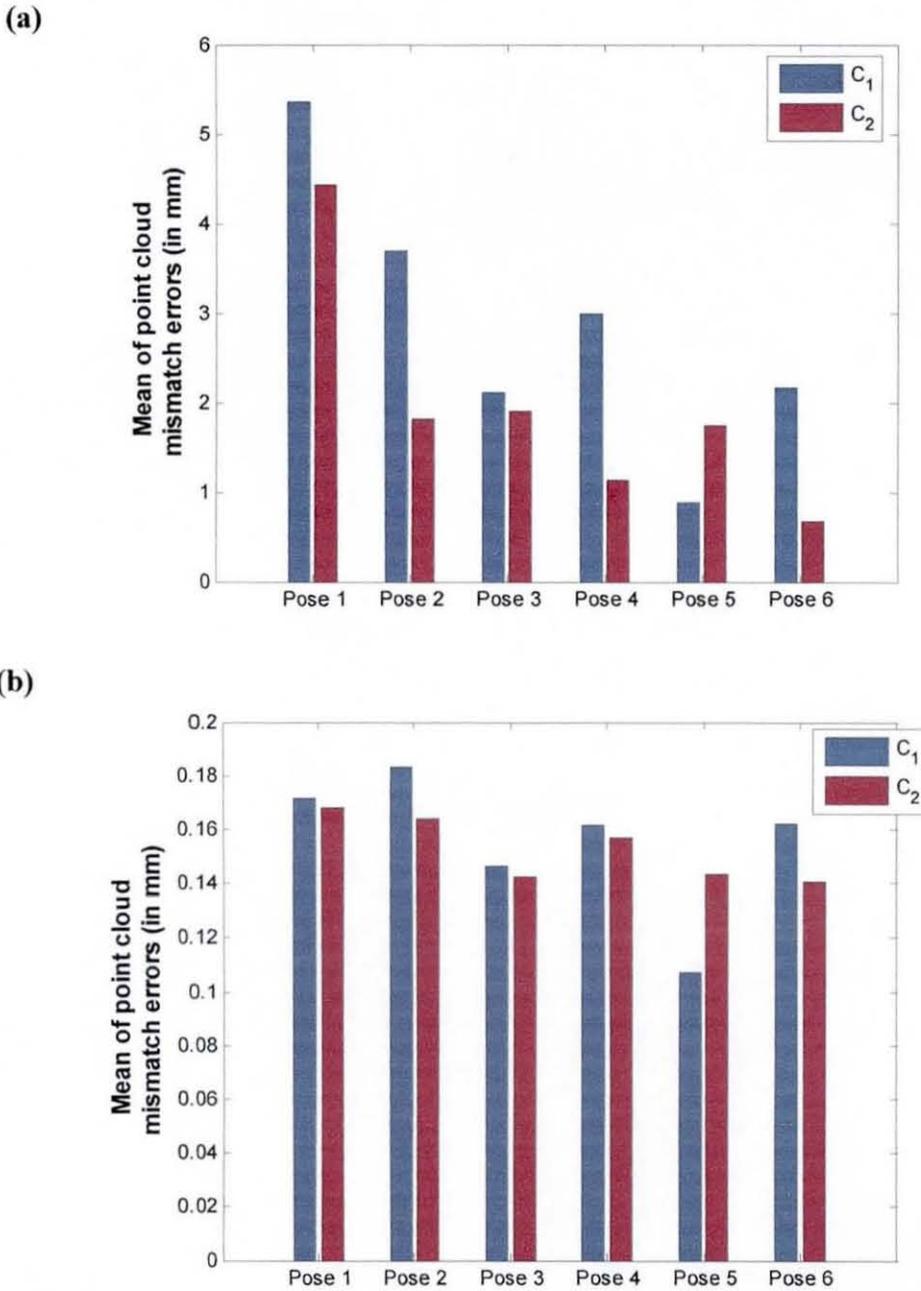


Figure 6-21: Mean of point cloud mismatch error values as calculated using acquired data of poses of the ball bar artefact and sensor parameters. (a) Each column represents the mean of point cloud mismatch error values using data from respective cameras and initial sensor parameters; (b) Each column represents the mean of point cloud mismatch error values using data from respective cameras and optimised sensor parameters.

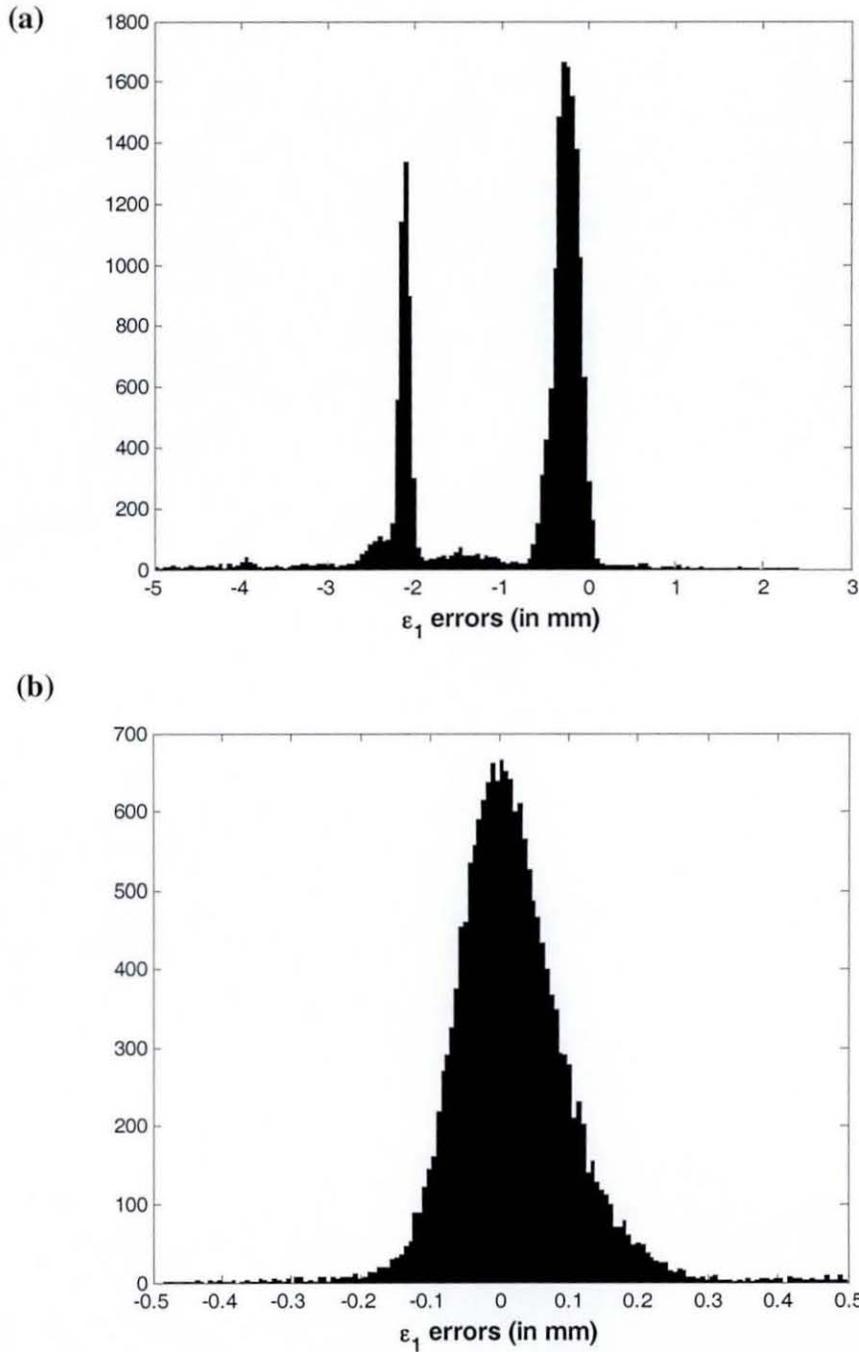


Figure 6-22: Histogram of 100 bins for ε_1 values of valid pixels calculated using parameters of C_1 and P_1 on acquired shape data of pose 3 of the ball bar artefact. The vertical axis represents the number of votes in each bin. (a) Histogram with values calculated using initial sensor parameters; (b) Histogram with values calculated using optimised sensor parameters.

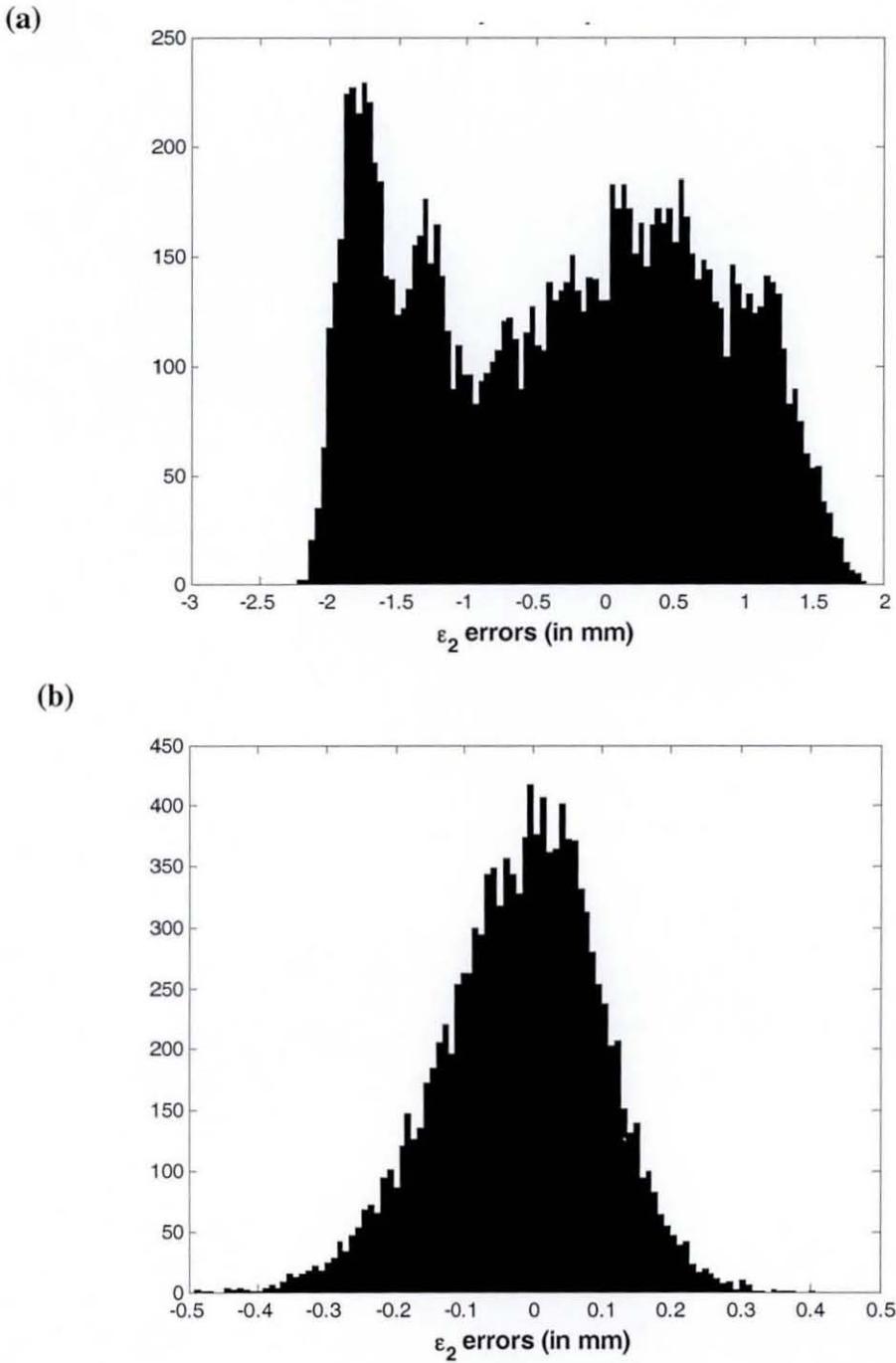


Figure 6-23: Histogram of 100 bins for ε_2 values of valid pixels calculated using parameters of C_1 and P_1 on acquired shape data of pose 3 of the ball bar artefact. The vertical axis represents the number of votes in each bin. (a) Histogram with values calculated using initial sensor parameters; (b) Histogram with values calculated using optimised sensor parameters.

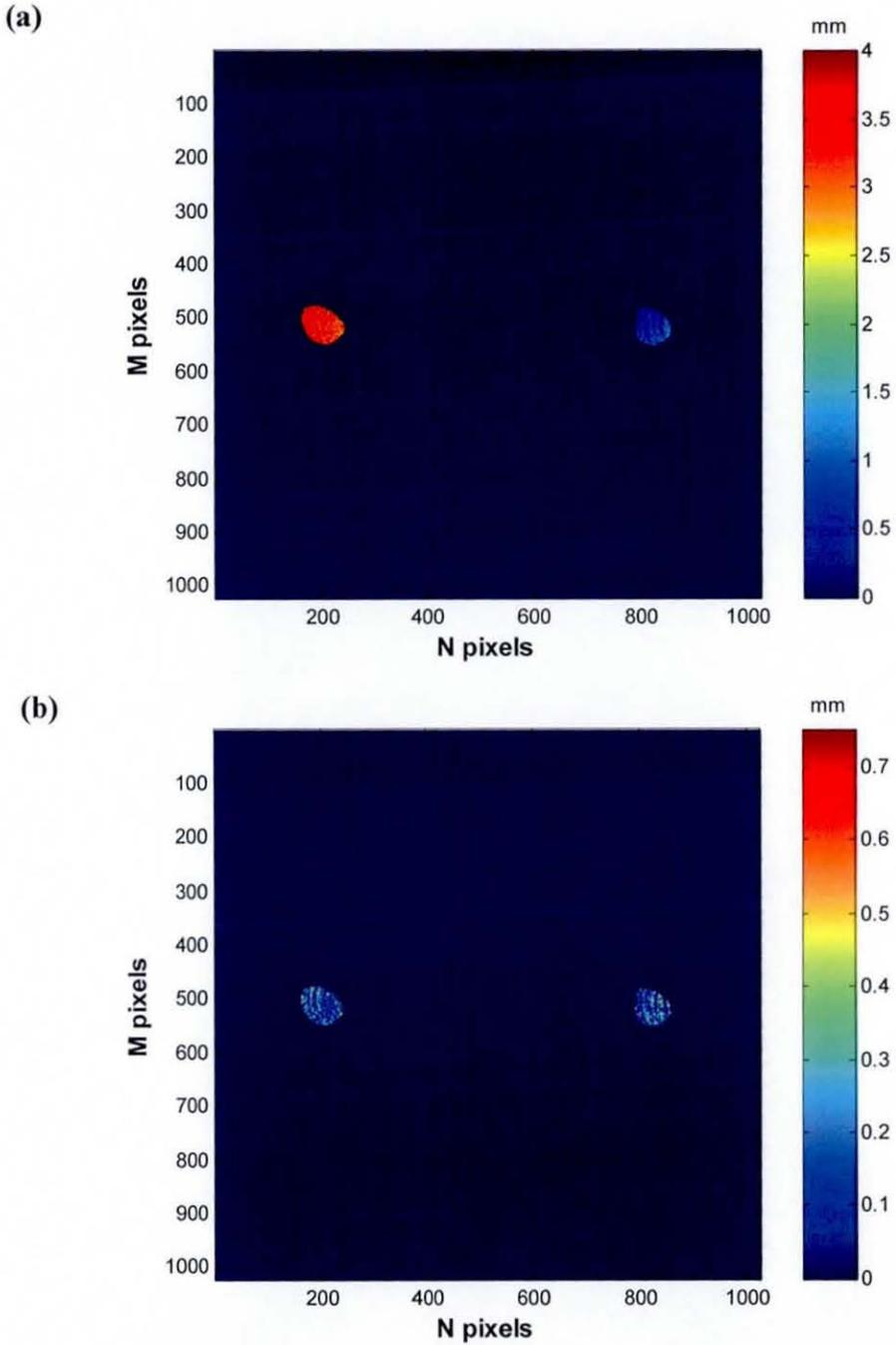


Figure 6-24: Point cloud mismatch error of pose 3 of the ball bar artefact acquired by C_1 , i.e. difference in 3-D coordinates calculated for valid pixels common to acquired shape data of C_1P_1 and C_1P_2 . The respective point clouds were calculated using the acquired shape data and parameters of C_1 , P_1 and P_2 . (a) Difference when initial sensor parameters are used; (b) Difference when optimised sensor parameters are used.

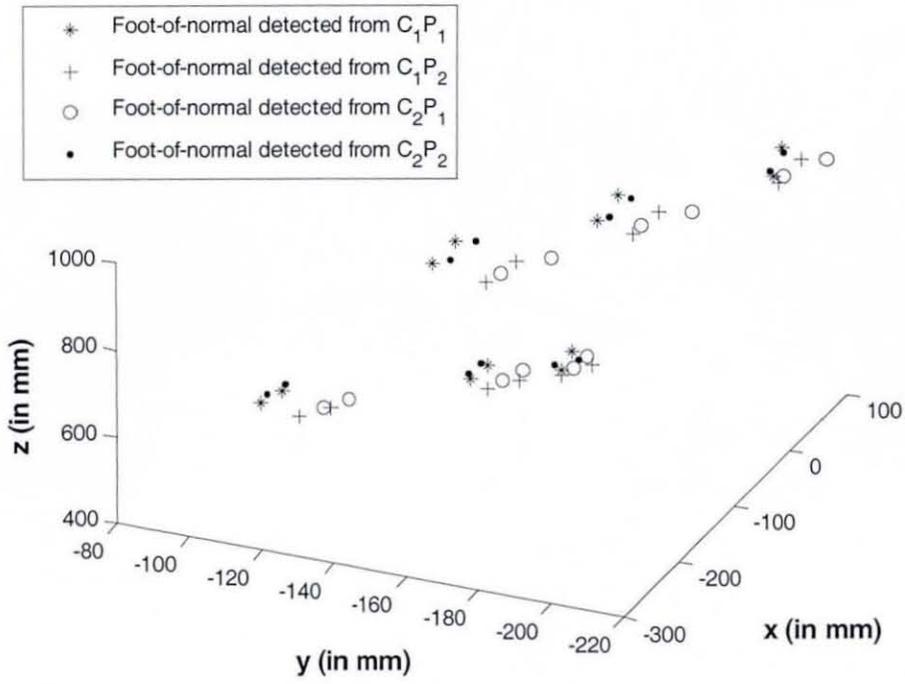


Figure 6-25: Location of the foot-of-normal positions detected from point cloud computed using initial estimates of sensor parameters.

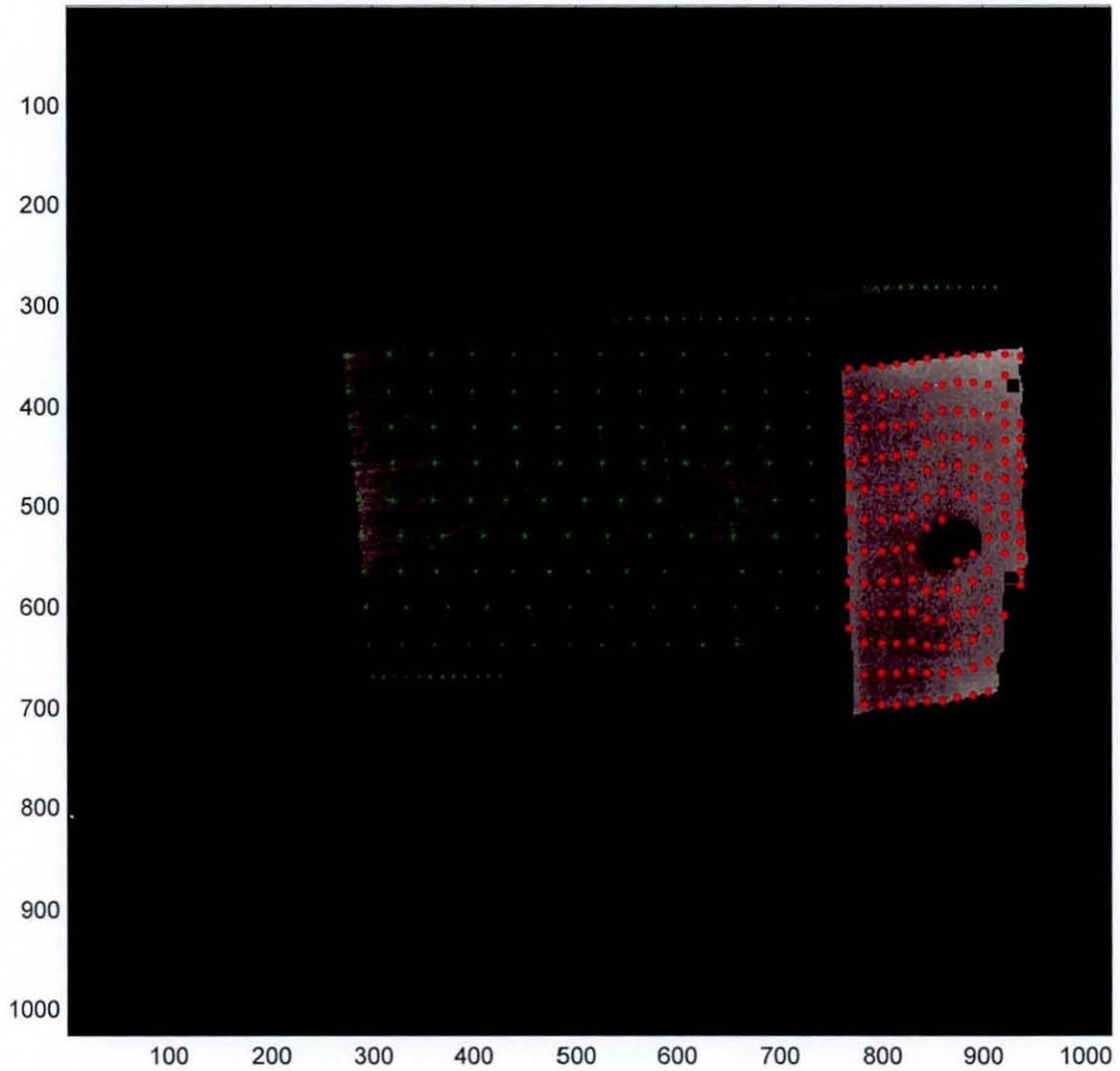


Figure 6-26: Greyscale intensity image of two-plane artefact at pose 3 (acquired by C_1P_1), showing intensity values of valid pixels. The pixels displayed in red and green correspond to pixels automatically selected for use in the bundle adjustment.

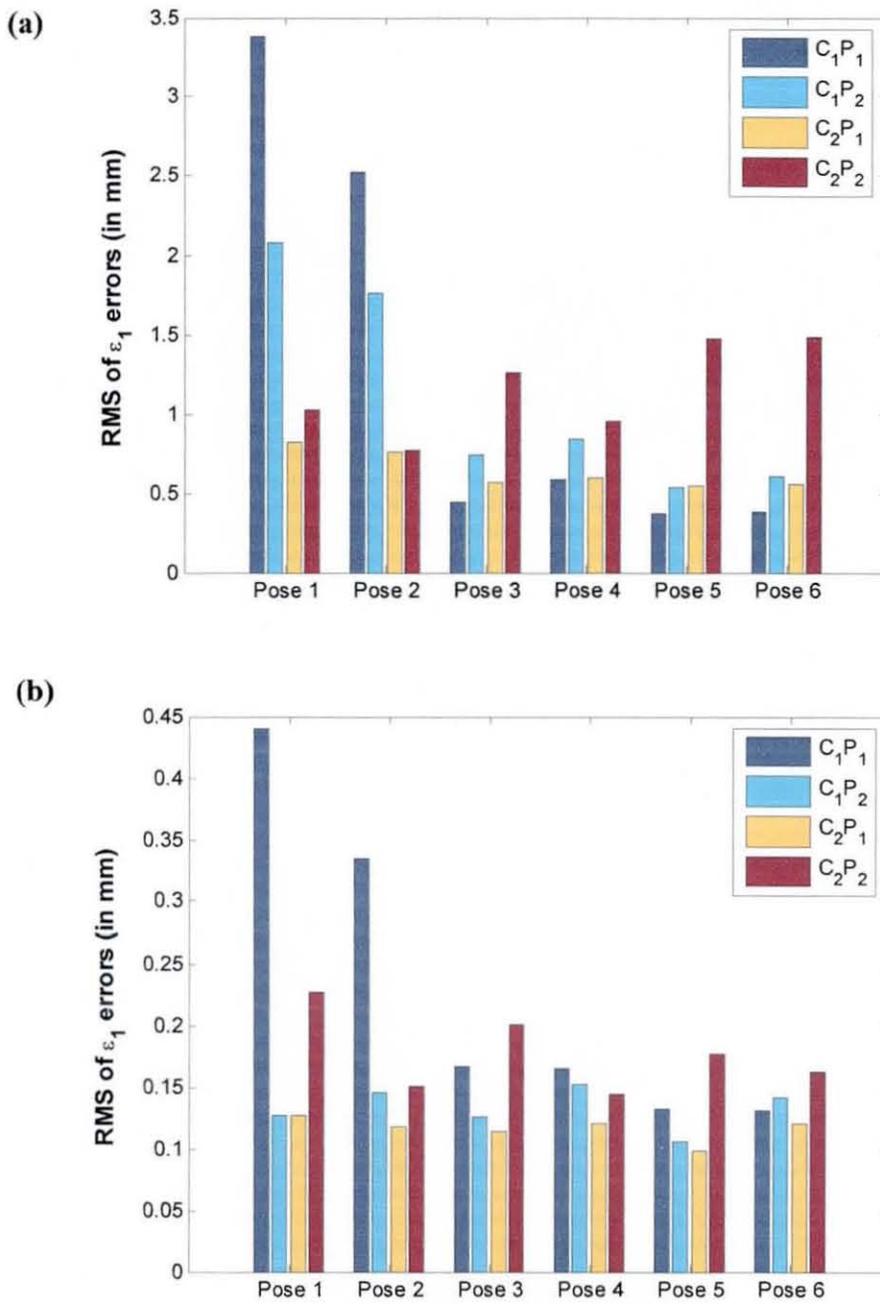


Figure 6-27: RMS of ε_1 values as calculated using acquired data of poses of the two-plane artefact and sensor parameters. (a) Each column represents the RMS of ε_1 values using data from respective camera-projector pairs and initial sensor parameters; (b) Each column represents the RMS of ε_1 values using data from respective camera-projector pairs and optimised sensor parameters.

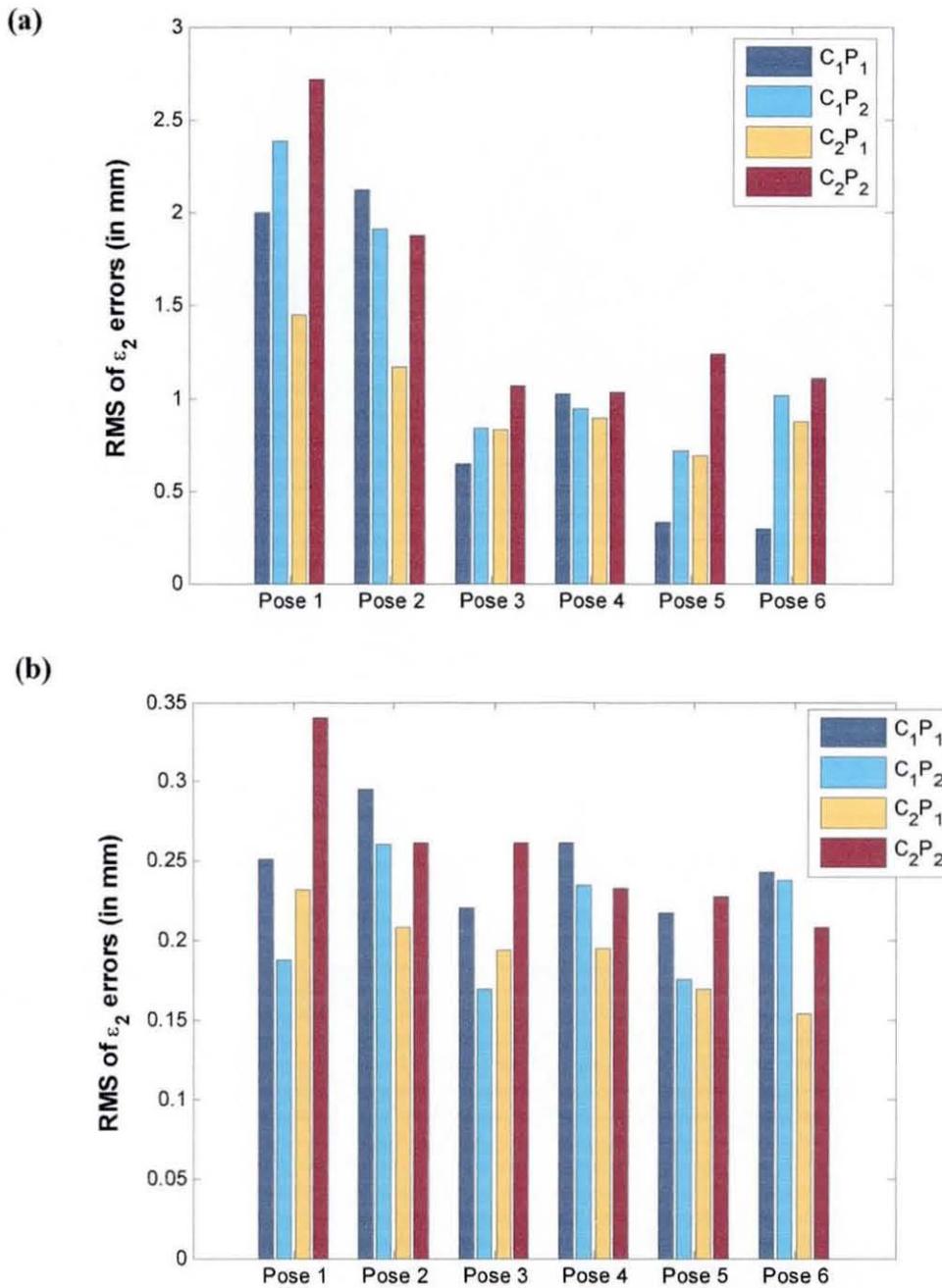


Figure 6-28: RMS of ε_2 values as calculated using acquired data of poses of the two-plane artefact and sensor parameters. (a) Each column represents the RMS of ε_2 values using data from respective camera-projector pairs and initial sensor parameters; (b) Each column represents the RMS of ε_2 values using data from respective camera-projector pairs and optimised sensor parameters.

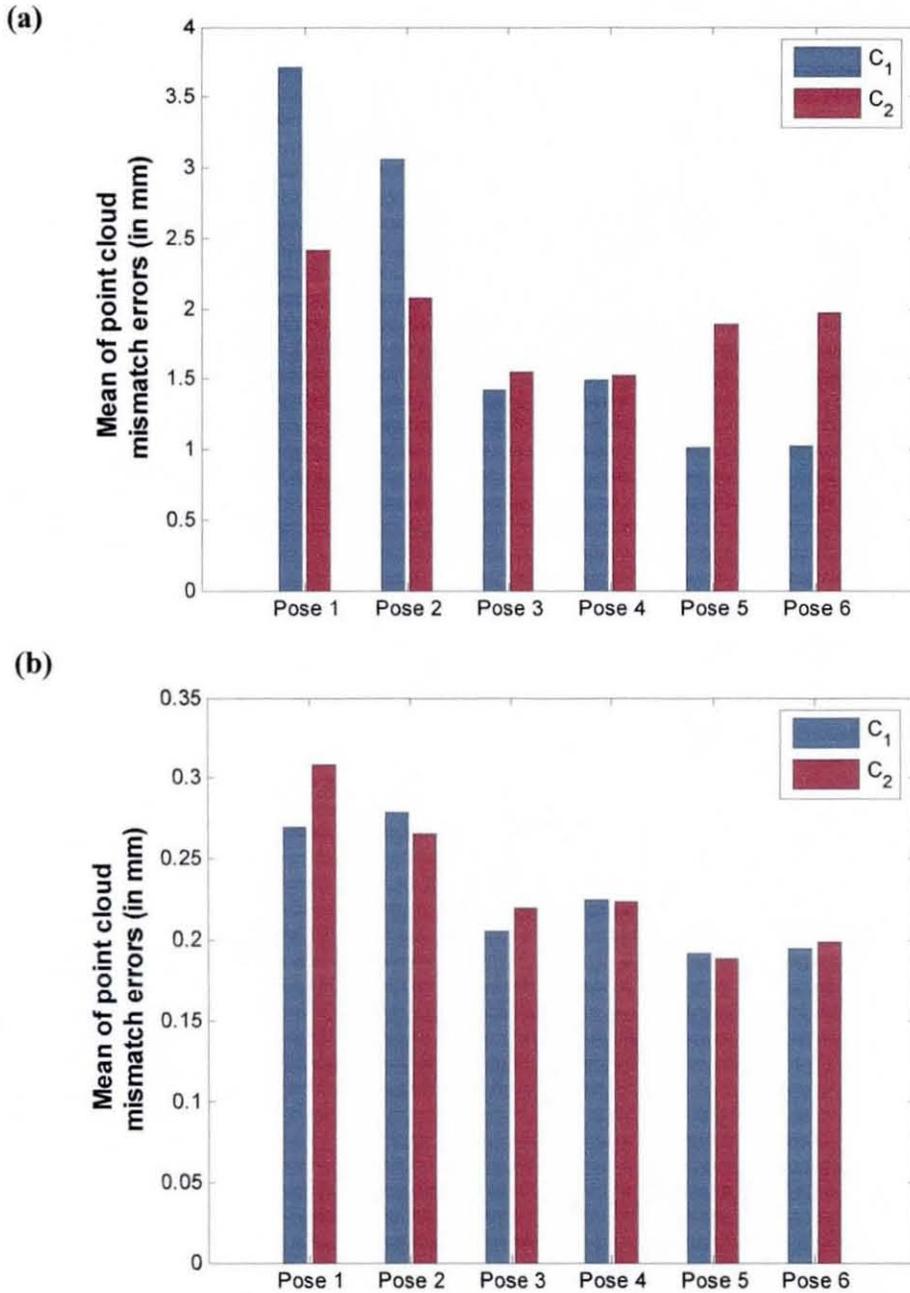


Figure 6-29: Mean of point cloud mismatch error values as calculated using acquired data of poses of two-plane artefact and sensor parameters. (a) Each column represents the mean of point cloud mismatch error values using data from respective cameras and initial sensor parameters; (b) Each column represents the mean of point cloud mismatch error values using data from respective cameras and optimised sensor parameters.

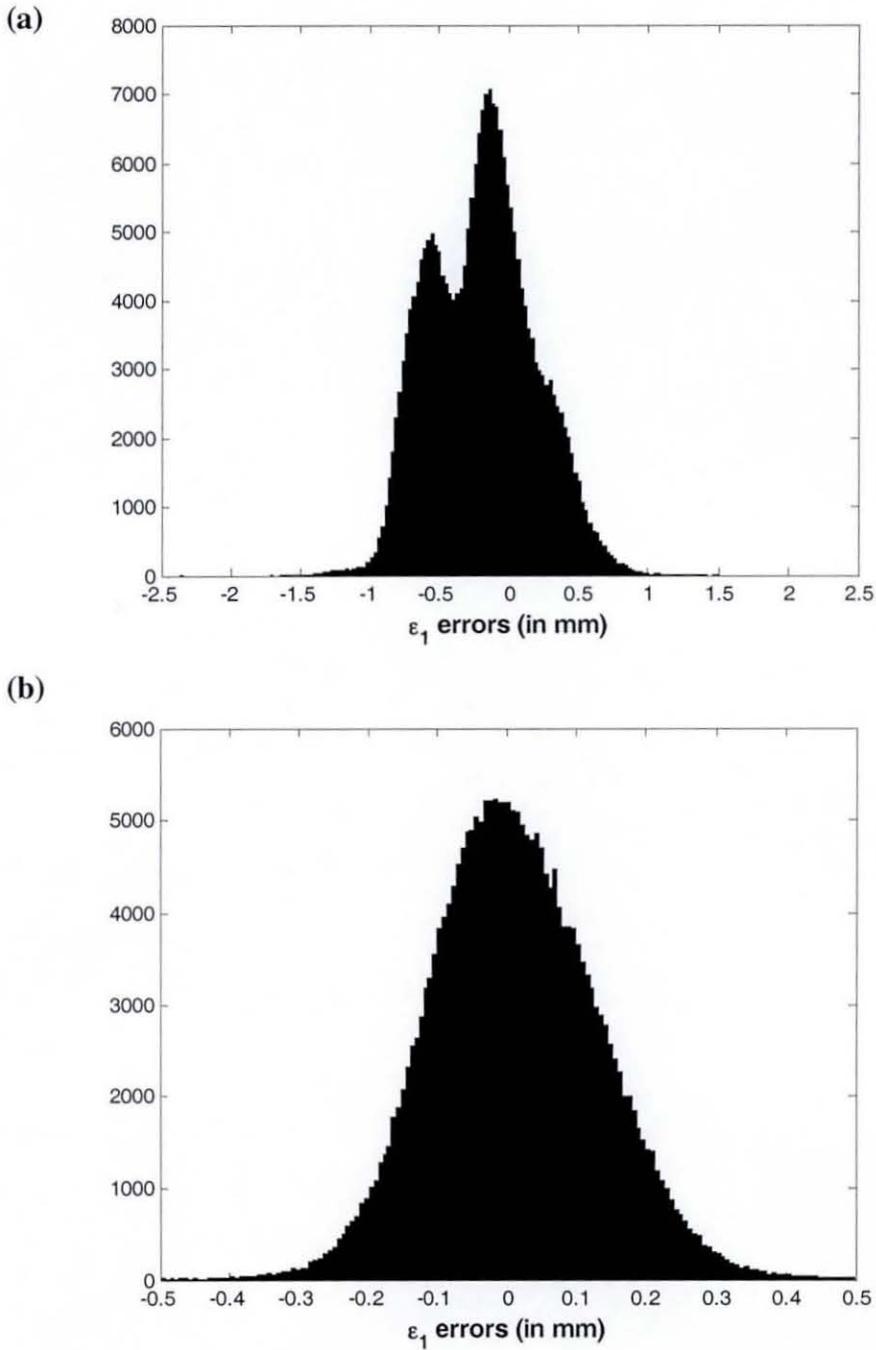


Figure 6-30: Histogram of 100 bins for ϵ_1 values of valid pixels calculated using parameters of C_1 and P_1 on acquired shape data of pose 3 of the two-plane artefact. The vertical axis represents the number of votes in each bin. (a) Histogram with values calculated using initial sensor parameters; (b) Histogram with values calculated using optimised sensor parameters.

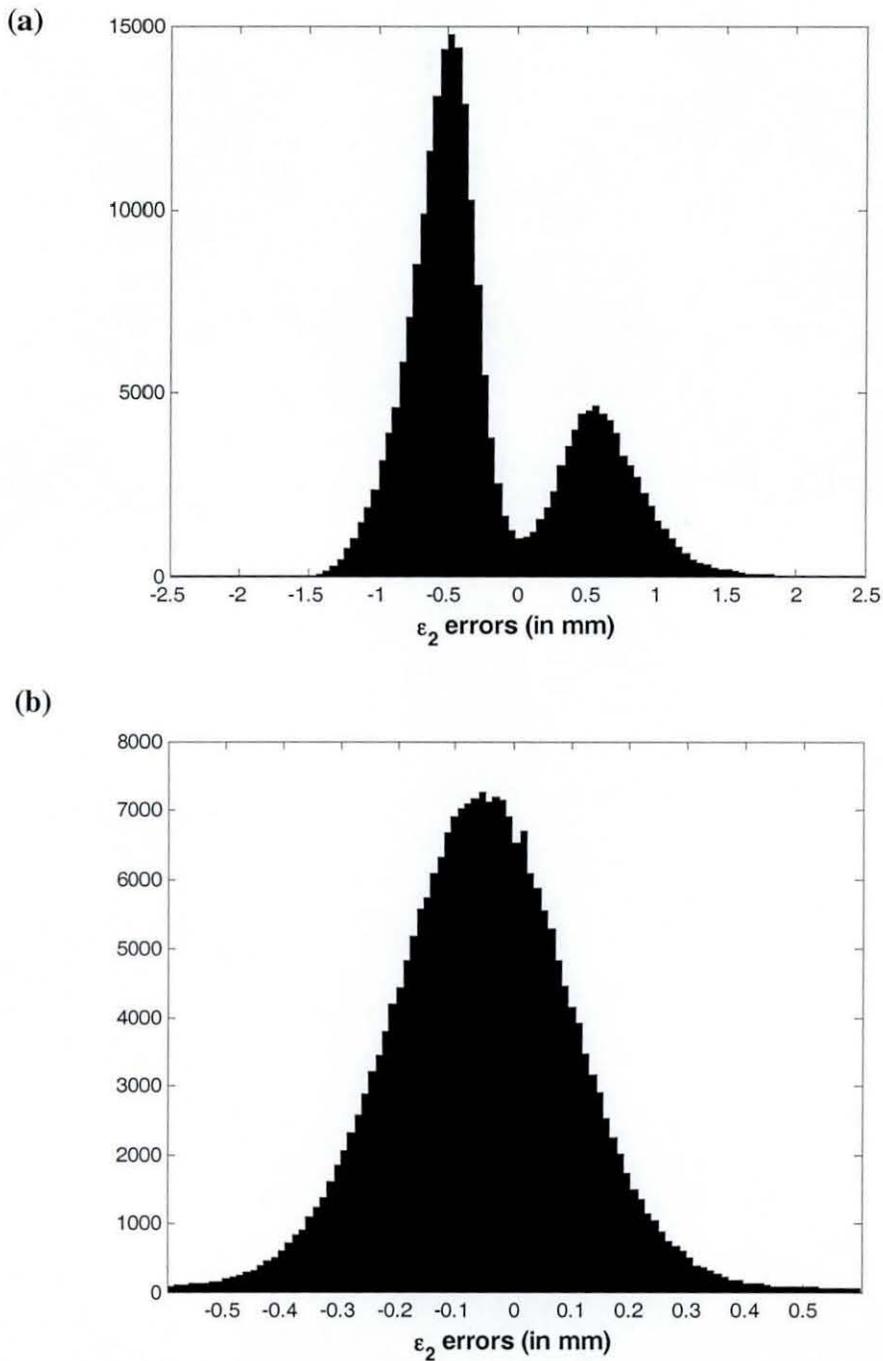


Figure 6-31: Histogram of 100 bins for ε_2 values of valid pixels calculated using parameters of C_1 and P_1 on acquired shape data of pose 3 of the two-plane artefact. The vertical axis represents the number of votes in each bin. (a) Histogram with values calculated using initial sensor parameters; (b) Histogram with values calculated using optimised sensor parameters.

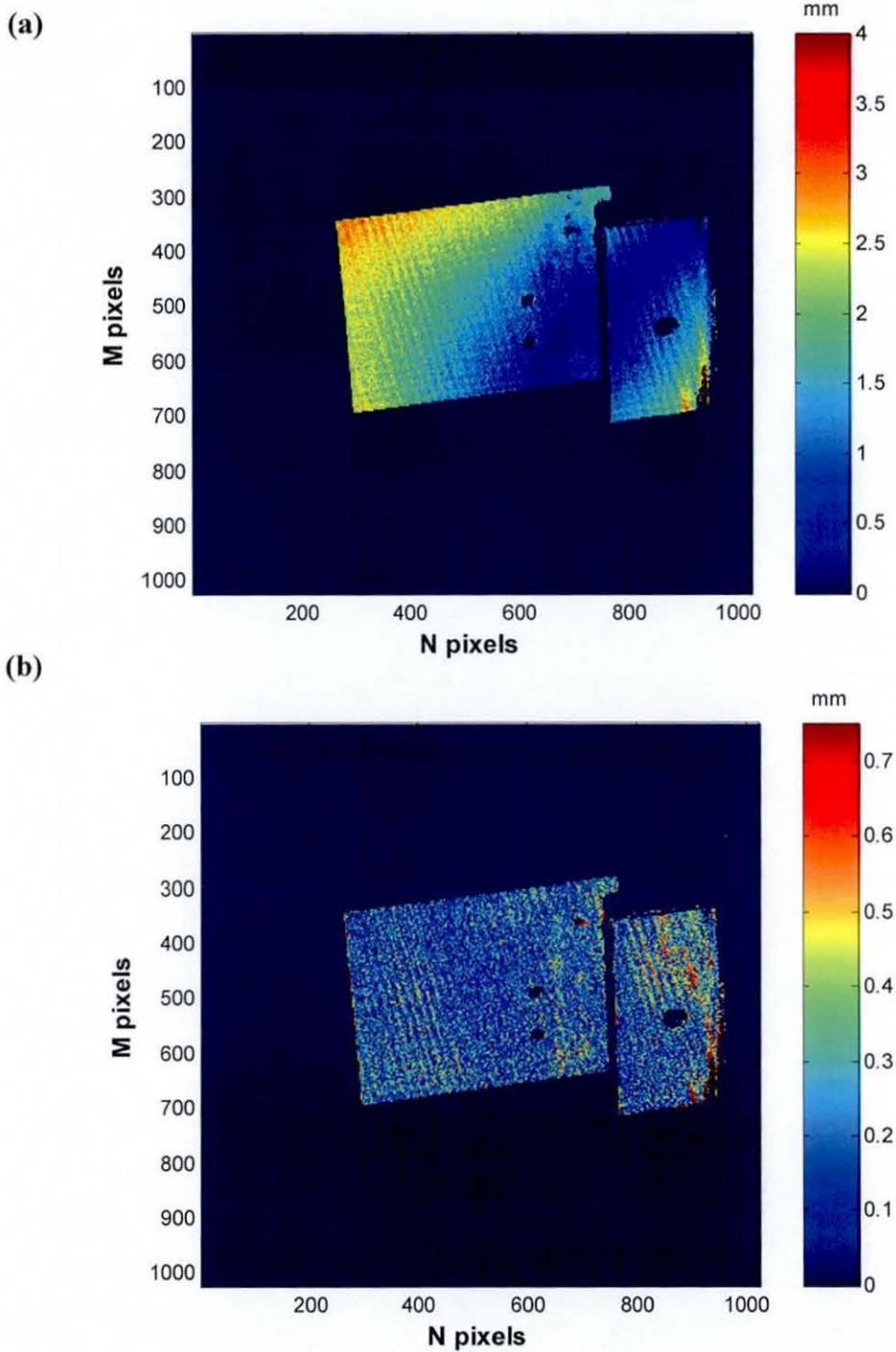


Figure 6-32: Point cloud mismatch error of pose 3 of the ball bar artefact acquired by C_1 , i.e. difference in 3-D coordinates calculated for valid pixels common to acquired shape data of C_1P_1 and C_1P_2 . The respective point clouds were calculated using the acquired shape data and parameters of C_1 , P_1 and P_2 . (a) Difference when initial sensor parameters are used; (b) Difference when optimised sensor parameters are used.

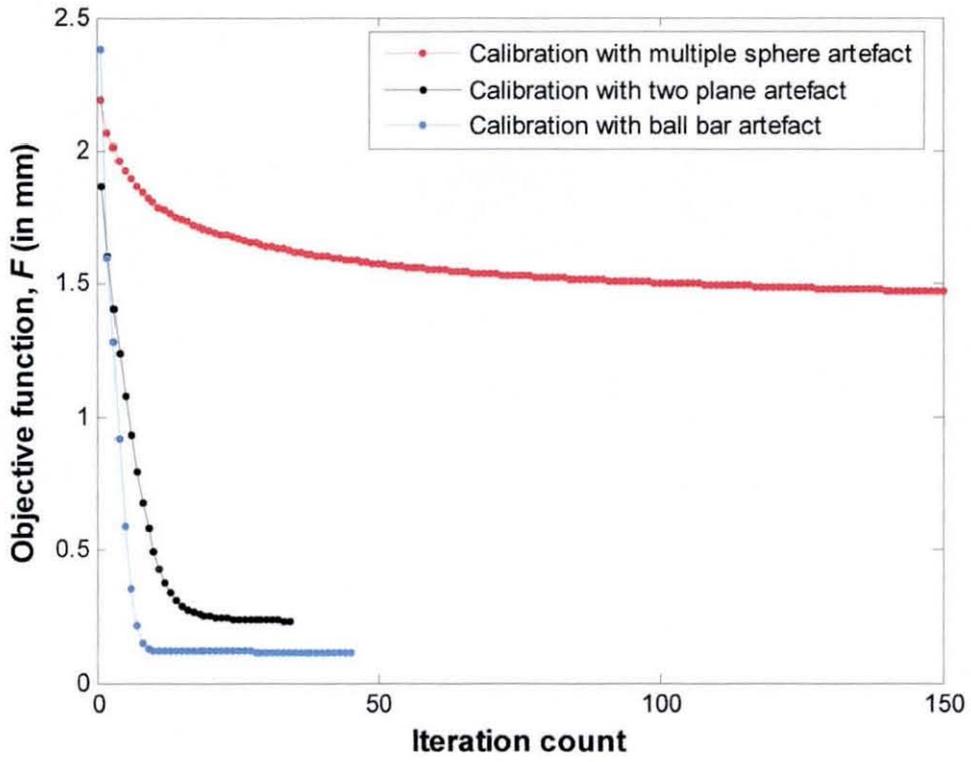
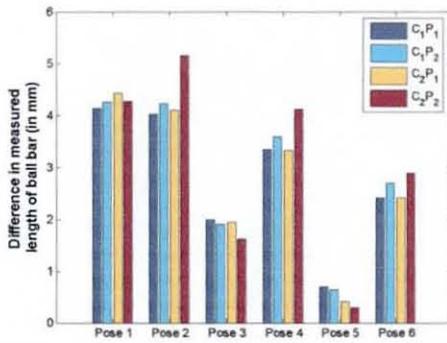
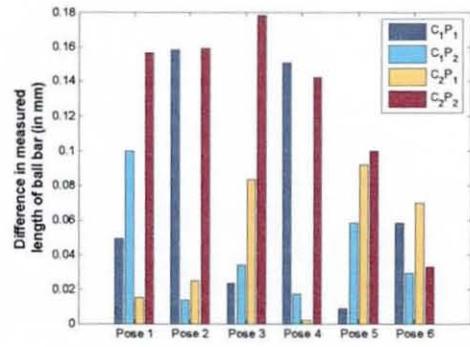


Figure 6-33: Convergence rate of the objective function F , using data from the multiple-sphere, ball bar and two-plane artefacts respectively.

(a)



(b)



(c)

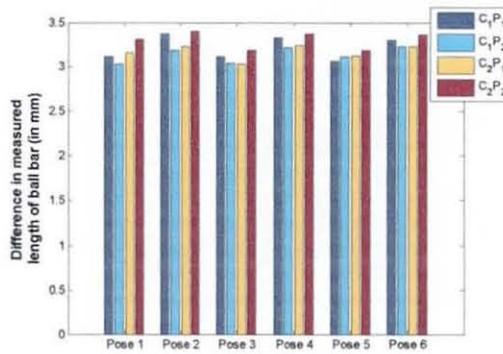
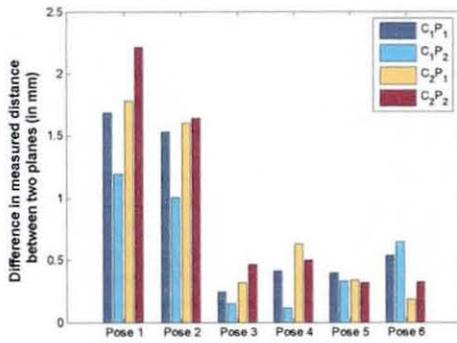
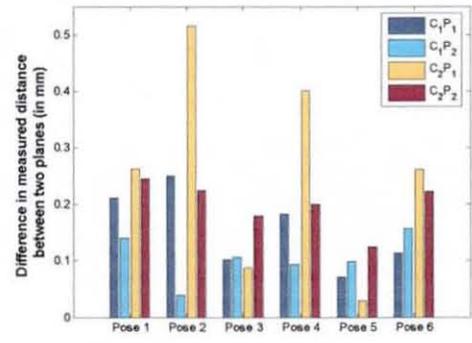


Figure 6-34: The absolute difference between the optical SMS measured and the CMM measured distance between the spheres of the ball bar artefact. (a) The calculated difference when sensor parameters obtained from calibration using multiple-sphere artefact are used to compute point cloud; (b) The calculated difference when sensor parameters obtained from calibration using ball bar artefact are used to compute point cloud; (c) The calculated difference when sensor parameters obtained from calibration using two-plane artefact are used to compute point cloud.

(a)



(b)



(c)

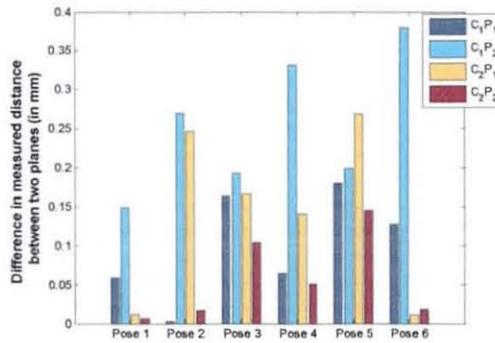


Figure 6-35: The absolute difference between the optical SMS measured and the CMM measured distance between the two planes of the two-plane artefact. (a) The calculated difference when sensor parameters obtained from calibration using multiple-sphere artefact are used to compute point cloud; (b) The calculated difference when sensor parameters obtained from calibration using ball bar artefact are used to compute point cloud; (c) The calculated difference when sensor parameters obtained from calibration using two-plane artefact are used to compute point cloud.

Section III –
Post – processing of
measured coordinates

7 Matching of Measured Coordinate Data to CAD Models

7.1 Introduction

In Chapter 6, an automated calibration process for a multi-sensor optical shape measurement system (SMS) has been described. The outcome of the calibration process is a set of sensor parameters which are optimised for a given measurement volume. For each measurement made by each camera-projector pair in the SMS, a point cloud consisting of a set of 3-D coordinates is computed using these optimised parameters, where each 3-D coordinate corresponds to a valid pixel in the acquired shape data. After calibration, the SMS can be used to measure objects for inspection and quality assessment purposes. In product design and manufacture, it is desirable to have such data compared against an idealized geometric model of the part under inspection. The idealized geometric data is typically in the form of a computer-aided design (CAD) model developed using CAD or solid modelling software. It is advantageous to use CAD models in inspection because the models contain an exact specification of an object [130]. However, in order to make a comparison of these two datasets, they have to be appropriately 'matched'. Matching refers to the process of determining the optimal translation vector and rotation matrix required to align two 3-D point data sets represented in different coordinate systems. Therefore, the added functionality of exporting the measured data from the SMS for visualisation and matching against a CAD model would enhance the prospects of the SMS as a practical inspection system.

In CAD software, the 3-D coordinates of a CAD model are usually saved in a predefined file format. The file format is usually specific to the software and if the CAD model is to be exported for further processing or visualisation on different CAD software, it would need to be converted to a file format that is supported by the recipient CAD software. There is thus a variety of such file formats. Clearly, to extract the 3-D coordinates of a CAD model for matching with point cloud data, it is necessary to have knowledge of the file format in which the model has been saved. In addition, for visualisation of the point cloud data from the SMS, the 3-D coordinates need to be written in a file format which can be imported into either CAD or point cloud visualisation software.

The matching problem arises because the coordinate system of the optical SMS (where the point cloud is measured relative to) is different from the coordinate system of the CAD model. The point cloud can be considered to be a rigid body relative to the SMS's coordinate system, while the CAD model can also be considered as a rigid body relative to some local coordinate system. The problem could therefore be considered as a misalignment between two point cloud data sets (the measured data and model data), and the process of matching these two data sets is often referred to as registration. Accurate registration of free-form surfaces is an essential requirement for dimensional inspection and, as such it is relevant to many branches of manufacturing industry [131].

Registration of two 3-D datasets is a classical problem in technical computing, often referred to as the orthogonal Procrustes problem, or the rigid body, movement problem [116]. This problem can be posed mathematically as a non-linear optimisation problem over six variables (three translational and three rotational). The solution to this problem is to find the transformation (translation and rotation) that would correct the misalignment and minimise an appropriate objective function describing the difference between the two data sets [116]. With a significant number of coordinates obtained from the SMS, it therefore becomes a challenge to efficiently and accurately bring the measured point cloud into registration with the CAD model. However, it is worth noting here that only the background work relating to the registration problem will be discussed in this chapter

In this Chapter, a method is described for efficiently creating a 3-D mesh from the point cloud of the optical SMS which is exportable to known CAD file formats. Examples of the different CAD file formats will be mentioned, with emphasis on the two file formats implemented in this work. Some observations from applying the meshing algorithm on measured data of the multiple-sphere artefact (i.e. part of the dataset discussed in Chapter 6), where the tessellated data set is exported to CAD file formats, will be discussed. The key methods of point cloud registration will be highlighted, with discussions focussing on the dominant method, the iterative closest point algorithm (ICP). In addition, some identified efficient variants of the ICP will be described. A new method proposed for point cloud registration using a 3-D look-up

table will be introduced, including a description of the C++ implementation of the method.

7.2 Meshing of point cloud from optical SMS

7.2.1 Description of algorithm for tessellating point cloud

Typically, CAD file formats require 3-D data to be presented in the form of triangular facets, quadrilateral facets, curved lines or surfaces (e.g. NURBS). Thus, visualisation of the point cloud data from the optical SMS in CAD or visualisation software is possible once the data set is tessellated to form a mesh consisting of triangular or quadrilateral facets. The key advantage of shape data acquisition of the optical SMS is that a 3-D coordinate is calculated for every valid pixel, thus neighbourhood information is available after point cloud generation.

Consider 4 neighbouring pixels shown in Figure 7-1 (a), with labelling in clockwise order, the four possible triangles that can be created using any 3 pixels as vertices are shown in Figures 7-1 (b), 7-1 (c), 7-1 (d), 7-1 (e). Triangles 1 and 2 can be created when all 4 pixels are valid, however, when 3 pixels are valid, either Triangle 1, 2, 3 or 4 can be created. The matrices for storing the coordinates of the point cloud could be quite large. For example, if a 1024×1024 pixel camera is used in the shape data acquisition, the point cloud would consist of three 1024×1024 matrices, one each for the *x*, *y*, and *z* coordinates. Thus, it is necessary to have an efficient way of identifying the vertices that belong to each triangle.

The 4 neighbouring pixels can be considered to be a bit map consisting of ones and zeros which make up a 1×4 bit array, **B**. If a pixel is valid, it is assigned a value of 1; otherwise, it is assigned a value of 0. Therefore, with only 3 valid pixels required to create at least one triangle, we can create 4 distinct binary numbers which represent the different triangles. Also, when all the pixels are valid, a unique binary number is created. Thus with knowledge of which pixels are valid, we can identify the triangles that can be created. As stated earlier, the pixels are labelled from 0 to 3 in clockwise order with the pixel at the bottom right hand corner assigned the value of 0. Similarly, the bit ordering goes from 0 to 3. Therefore, for the bit array, **B**, the equivalent decimal values can be calculated thus:

$$V = B(0) \times 2^0 + B(1) \times 2^1 + B(2) \times 2^2 + B(3) \times 2^3 \quad (7-1)$$

For example, for Triangle 1, the values of the bit array would be $B(0) = 1$, $B(1) = 1$, $B(2) = 1$ and $B(3) = 0$. The decimal value from the contents of **B** would be calculated as $1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 = 7$. Table 7-1 shows decimal values calculated for all the possible sets of valid pixels.

From the SMS we can obtain a Mask, **M**, which is a matrix with the same size as the camera's pixel resolution, where each element contains a value (1 or 0) to indicate if the acquired shape data at that pixel is valid. If $M(i,j)=1$, then the measured value at pixel (i,j) is valid, while $M(i,j)=0$ indicates that it is invalid. We then create a matrix **R**, with the same size as **M**. If we consider a 2×2 sub-mask consisting of the rows $R(i-1, j-1)$ $R(i-1, j)$ and $R(i, j-1)$, $R(i, j)$ (for example, created from the first set of 4 neighbouring pixels, where $i=2, j=2$) we can extract the state of these pixels from **M**, and use a clockwise ordering as stated earlier to create the bit array, **B**. On calculating the equivalent decimal value of **B**, the value is simply assigned to $R(i,j)$. Using the 2×2 sub-mask, the process for populating **R** and identifying the pixel coordinates for populating the list of facets can be summarised by the following MATLAB code:

```

for i = 2:NRows
    for j = 2:NCols
        % Create bit array from 2x2 sub-mask
        B = [ M(i,j) M(i,j-1) M(i-1,j-1) M(i-1,j) ];
        R(i,j) = B(1)*2^0 + B(2)*2^1 + B(3)*2^2 + B(4)*2^3;
        if R(i,j) == 15 % then all 4 pixels are valid
            % create Triangles 1 and 2 or create a
            % quadrilateral
            AddTriangle(B,1,2) . . . . .
            . . . . .
        elseif R(i,j) == 7 % create Triangle 1
            AddTriangle(B,1)
            . . . . .
        elseif R(i,j) == 13 % create Triangle 2
            AddTriangle(B,2)
            . . . . .
        elseif R(i,j) == 11 % create Triangle 3
            AddTriangle(B,3)
            . . . . .
        elseif R(i,j) == 14 % create Triangle 4
            AddTriangle(B,4)
            . . . . .
        else
            % we have less than three valid pixels
        end
    end
end
end

```

Based on R , which therefore contains the connectivity information required for building up a list of facets, *AddTriangle* can be considered as a function for assigning the (i,j) indices for the vertices of each facet. Using the calculated value of x, y, z coordinates of two neighbouring pixels which are possible vertices of an edge in a facet, we can calculate the physical length of all the facet's edges. Although two pixels which form a facet edge may be neighbours in image space, however in object space, they could be far from each other as the edge length could be quite significant. Therefore, a threshold value, t_f , has been introduced to ensure that a facet is created only when all the physical lengths of each of its edges are within tolerance (i.e. facet edge lengths $l_i < t_f$, where i is the number of facet edges).

It is worth noting that the objective in this work is the ability to efficiently create a mesh from the measured point cloud data from the SMS using pixel neighbourhood information. However, as a result combining small facets to obtain a complete mesh of the measured data, we are left with significantly large data sets which could lead to data analysis, manipulation and storage problems. For example, for a relatively flat surface, large facets could be used rather than sets of smaller facets created from the 3-D coordinates of neighbouring pixels. Therefore, methods such as that of Lee et al [132] have been proposed for 3-D point cloud data reduction.

7.2.2 Data representation in CAD Software (CAD file formats)

With developments in computer graphics, computational geometry, and computer technology, there is a variety of CAD software being used for designing CAD models in industry. Leading CAD software includes AutoCAD[®], Unigraphics[®], SolidEdge[®], I-DEAS[®], CATIA[®], ProEngineer[®], SolidWorks[®], etc, where CAD models are represented in a variety of file formats or data structures. An important issue to be considered was how to acquire data of a CAD model in a usable form from CAD software, since they could be represented in different software-platform-dependent (native) file formats. Examples include the DXF format for AutoCAD[®], the OBJ format for Autodesk[®] AliasStudio[™], and SolidEdge[®] PAR file format.

There are presently a number of 'neutral' CAD file formats available for representing CAD data, which could be used for transferring CAD data from one CAD package to another, with each package having a translator for pre-processing (for converting

native file formats to neutral file formats) and post-processing (for reading neutral file formats and converting to native file formats). The main examples include IGES (Initial Graphics Exchange Specification), STEP (Standard for the Exchange of Product Data – which is an ISO standard ISO 10303), VDA-FS (German national standard based on IGES), STL (Stereo Lithography files), and VRML (Virtual Reality Modelling Language). Generally, most leading CAD software have translators for pre-processing and post-processing the above mentioned neutral file formats. Discussions however, will focus on the file formats that were implemented in this work.

7.2.3 STL file format

The Stereo Lithography (STL) file format is the simplest of the above mentioned neutral file formats. STL was developed for Rapid Prototyping, where a CAD model is tessellated into triangular facets using a triangulation convention, where the data for each facet is represented with a normal and three vertices, and stored in an ordered list. Indeed, STL is the standard format for storing 3-D data in the Rapid Prototyping industry. The vertices are listed in counter-clockwise order such that they follow the *right-hand-rule*, ensuring that the direction of the facet's normal points outward (Figure 7-2 (a)). In addition, the vertices must adhere to the *vertex-to-vertex* rule, where a facet's vertex must not lie on an edge i.e. between two vertices (see Figures 7-2 (b) and 7-2 (c)). There are two formats for STL files - the ASCII format (which is both human and computer readable) and the binary format (which is computer readable only). The format for an ASCII STL file is [133]

```
solid name
  (
    facet normal nx ny nz
      outer loop
        vertex v1x v1y v1z
        vertex v2x v2y v2z
        vertex v3x v3y v3z
      endloop
    endfacet
  )
endsolid name
```

The strings in bold refer to STL keywords, while the other characters refer to the values that need to be written. The '**facet normal - endfacet**' block represents each facet, thus a typical STL file contains multiples of this block. The 'name' parameter is the name associated with the model and is usually a string stating the name of the file

and other relevant information. However, the other values are associated with the facet and should be written as floating point numbers. Excerpts from an example ASCII STL file for a model with a single facet is given below:

```
solid example.stl
  facet normal 1.0 1.0 1.0
    outer loop
      vertex 0.0 0.0 0.0
      vertex 1.0 0.0 0.0
      vertex 1.0 1.0 0.0
    endloop
  endfacet
endsolid example.stl
```

From the above, it is obvious that the list of facets from an STL file could be very long even for a relatively simple CAD model. However, significant savings can be achieved by saving the file in binary format rather than ASCII format. The format for a binary STL file is given in Table 7-2, with each facet represented by 50 bytes. Thus a binary STL file size is $84+50n$ bytes, where n is the number of facets in the model.

Although simplistic in structure, the STL file format is considered to be quite inefficient in that [133]:

- Facet normals need not be stored since they could be calculated from the three vertices of the triangular facet.
- Every vertex is stored for each facet it lies on, thus, vertices are stored a multiple number of times.

7.2.4 OBJ file format

The *obj* file format (also sometimes referred to as the Wavefront *obj* format) is the format for storing 3-D models in Autodesk®AliasStudio™. Autodesk®AliasStudio™ is CAD software used for the creative design for surface modelling, rendering and animation [134]. The file format provides support for both polygonal and free-form objects, and is a popular file format for visualising point cloud data. Polygonal objects consist of points, lines and faces, while free-form objects consist of curves and surfaces. The keywords of the *obj* file format can be classified by data type as vertex data, free-form curve or surface attributes, geometric elements, connectivity of free-form surfaces, object groupings and display or render attributes [135]. The data types

referring to the polygonal objects used in this work will be described, while further details on specifications for other data types can be found in reference [135, 136].

The obj file format supports the representation of 3-D coordinates as vertices and triangular or quadrilateral facets. In addition, it provides a means for storing vertex texture coordinates which specify the texture mapping of each vertex. The 'v' keyword is used to specify the coordinates for a vertex, while 'vt' is used to specify the texture of the vertex with a number between 0 and 1. After the section associated with vertex data, the vertices of a geometric element (e.g. a face) are set based on the list of vertex data types. Indeed, the ordering of the vertex and texture vertex list is important in order to appropriately assign the vertices to a face. The keyword 'f' is used to set vertices for a face, with each vertex written in the format - 1st vertex/1st vertex texture/1st vertex normal e.g. a set of vertices could be written as f 1/1/1 2/2/2 3/3/3. An example for a single facet is shown below, where it can be seen that after the list of coordinates for the three vertices and three texture vertices are specified, they are assigned to a facet, 'f'.

```
v v1x v1y v1z
v v2x v2y v2z
v v3x v3y v3z
vt vt1x vt1y vt1z
vt vt2x vt2y vt2z
vt vt3x vt3y vt3z
f 1/1 2/2 3/3
```

The keywords 'usemtl' and 'mtllib' are combined to set the display attributes of the CAD model. mtllib refers to a separate material library file which contains colour definitions for facets in terms of diffuse, ambient and specular colours, and other features such as specularity, refraction and transparency [135]. With the 'usemtl' keyword, reference is made to a material already defined in the material library file. After specifying the material of the facet with 'usemtl', all the faces defined afterwards are assigned this attribute until the next 'usemtl' keyword. An example is shown below:

```
mtllib material.mtl
v 0.0 1.0 0.0
v 0.0 0.0 0.0
v 1.0 0.0 0.0
v 1.0 1.0 0.0
v 0.0 2.0 0.0
v 1.0 2.0 0.0
vt 0.0 0.0 0.0
vt 0.0 1.0 0.0
vt 0.0 0.0 1.0
usemtl blue
f 1/1 2/2 3/3
usemtl red
f 4/1 5/2 6/3
```

It should be noted that obj files can also be written in binary format with '.mod' as the file name extension rather than '.obj'. However, the binary format does not seem to be popular and the author was unable to come across detailed specifications.

7.2.5 Results for exporting measured data for visualisation

In this Section, an example will be shown of a 3-D mesh created from one of the acquired datasets of the multiple-sphere artefact for calibrating the optical SMS (discussed in Section 6.7.3 of Chapter 6). In addition, a comparison will be made of the STL and OBJ file formats in terms of the time taken to write to disk and file size on a computer with a Pentium4 processor (2.80GHz, 1GB RAM).

The point cloud was computed by using the optimised sensor parameters on the acquired shape data of the camera-projector pair C_1P_1 . For the acquired shape data, there are 386,157 valid pixels. Figures 7-3 (a) 7-3 (b) and 7-3 (c) show the greyscale intensity image acquired by C_1 and the calculated unwrapped phase maps, ω_x and ω_y respectively. In these figures, only the values for the valid pixels are displayed. The calculated x, y, z coordinates for each valid pixel are shown in Figures 7-4 (a), 7-4 (b) and 7-4 (c).

The point cloud was tessellated using the algorithm described in Section 7.2.1, with the threshold t_f set at 1 mm. The process was completed in approximately 2 seconds, leading to a total of 373,005 vertices and 718,133 triangular facets. The tessellated data was exported to the STL (binary and ASCII versions respectively) and the obj file formats. The binary version of the STL file was written in 107 seconds, while the

ASCII version was written in 175 seconds. However, the obj file was written in 65 seconds. As stated earlier, the representation of the obj file is more efficient than STL, therefore, it is expected that the obj file would be quicker to write in comparison with STL files. With regard to computer memory size, a binary file format is a more efficient representation in comparison with ASCII, thus it was expected that the binary version of the STL file would be smaller in size. The binary version of the STL file was smallest at 35 Mb, followed by the obj file at 54 Mb and then 177 Mb for the ASCII version of the STL file. It is necessary to note that the memory savings of the binary version of the STL file over the obj file is less than a factor of 2. It would therefore be expected that significant memory savings will be achieved by writing the point cloud as a binary version of the obj file format. However, as stated earlier, the binary format has not been implemented in this work.

Magics[®] a piece of software used in the Rapid Prototyping industry and available in the Wolfson School was used for viewing the point cloud exported as an STL file. Open source software, such as MeshLab [137] could also be used to view the point cloud data when exported as STL and obj files. Figure 7-5 (a) shows the image of the rendered surface of one of the spheres in the point cloud. Figure 7-5(b) shows the edges of each of the triangular facets used to create the sphere's surface. It should be noted that it is common for visualisation software to provide a means of smoothing the imported data set to improve surface smoothness and thus improve the aesthetics of the surface rendered point cloud. Figure 7-6 (a) shows an image of the surface rendered point cloud of the multiple-sphere artefact without the application of smoothing surface (with artificial lighting), while Figure 7-6 (b) shows an image of the surface rendered point cloud with smoothing applied.

7.3 Methods for registration of point clouds

For the registration of free-form surfaces, two registration methods are considered to be of special interest [138]: the moment of inertia method (MOI) and least squares fitting. The MOI method does not rely on the correspondence between the measured and nominal points [138]. It involves calculating for each data set, the first two moments of the distribution geometry, the translation that aligns their centre of mass, and the rotation that aligns their principal axes. The least squares fitting can be described as the process of estimating the optimal transformation (rotation and

translation) that aligns the model shape and the measured shape, minimising the distance between the shapes and thereby allowing determination of the equivalence of the shape via a mean-square distance metric [139]. The principal method of calculating the transformation between two misaligned data sets is least squares fitting [138].

Besl and McKay [139] proposed the iterative closest point algorithm (ICP) as a solution to a free-form surface matching problem, which could be considered as a problem of registration of 3-D shapes. The main application of the ICP is to register digitised (measured) data from unfixtured rigid objects with an idealized geometric model prior to shape inspection [139]. The ICP is well-known and considered as a standard algorithm for solving registration problems [140]. The advantage of the ICP is the ability to register measured data independent of the representation of the geometric model, i.e. point sets, curves, and surfaces. The ICP is attractive because of its simplicity and its performance [141]. However, other methods have also been proposed solving the registration problem, but their performance is usually compared to the ICP for justification [140, 141].

Surface registration methods can be broken down into two main categories [142]:

1. primitive-based – requiring feature extraction of e.g. special points, contours and surface patches
2. surface-based – which involves minimising a function that describes the distance between two surfaces.

The surface-based approach uses all available information (usually *a priori*) to produce redundancy to allow for precise calculation of the transformation between the two surfaces.

7.4 Iterative Closest Point Algorithm (ICP)

7.4.1 Description of the Conventional ICP

The ICP proposed by Besl and McKay [139] can be used on the following representations of geometric data: point sets, line segment sets, implicit curves or surfaces (i.e. algebraic or non-polynomial curves or surfaces), parametric curves or surfaces and triangle sets i.e. faceted surfaces. The ICP can be described in terms of minimising the cost function, F :

$$F = \min_{\mathbf{R}, \mathbf{T}} \sum_{i=1}^N \|(\mathbf{R}\mathbf{M}_i + \mathbf{T}) - \mathbf{C}_i\|^2 \quad (7-2)$$

where N is the number of points in the measured data, $\mathbf{M}_{i=1\dots N}$ are the points that constitute the measured data from sensor, $\mathbf{C}_{i=1\dots N}$ are the corresponding points which lie on the surface \mathbf{H} on the model data (i.e. CAD model), $\mathbf{R} = 3 \times 3$ rotation matrix, $\mathbf{T} = 1 \times 3$ translation vector.

The aim of the method is to find the optimal transformation (\mathbf{R} and \mathbf{T}) which minimises the misalignment (i.e., the least squared distance) between two independently measured data sets $\mathbf{M}_{i=1\dots N}$ and $\mathbf{C}_{i=1\dots M}$. The rhs of equation 7-2 above is referred to as a *distance metric*. In the method, \mathbf{R} is initialised as a diagonal 3×3 matrix with ones at the diagonal, while \mathbf{T} is initialised as 3×1 vector of zeros. At the k^{th} iteration of the ICP, for each i^{th} point in the measured data, \mathbf{M}_i , we find the closest point \mathbf{C}_i which lies on the surface \mathbf{H} of the model data. This i^{th} point correspondence could be obtained by tessellating \mathbf{H} using say Delaunay triangulation and then finding the closest facet to \mathbf{M}_i . The closest point in the model data, \mathbf{C}_i (which lies on the closest facet), is therefore the projection of \mathbf{M}_i to the closest facet along its surface normal. It should be noted that other methods for obtaining point correspondences are available based on the underlying surface representation of the measured data, \mathbf{H} [139].

After establishing for all N measured points the point correspondences in the CAD model data, the problem reduces to a Procrustes problem, where a solution based on Singular Value Decomposition (SVD) has been described in Chapter 4. Eqns. (4-25) to (4-30) can then be used to obtain the incremental transformation \mathbf{R}_k and \mathbf{T}_k for the point correspondences, $\mathbf{C}_{i=1\dots N}$ and $\mathbf{M}_{i=1\dots N}$. \mathbf{R}_k and \mathbf{T}_k are then used to cumulatively update \mathbf{R} and \mathbf{T} thus:

$$\bar{\mathbf{R}} = \mathbf{R}_k \mathbf{R} \quad (7-3)$$

$$\bar{\mathbf{T}} = \bar{\mathbf{R}} \mathbf{T}_k + \mathbf{T} \quad (7-4)$$

The ICP can be summarised thus [117]:

1. Initialise the cumulative transformation parameters \mathbf{R} and \mathbf{T} to the identity transformation. Reset the iteration counter, k , to zero.
2. For each point \mathbf{M}_i in the measured data, compute the closest point (in terms of Euclidean distance) \mathbf{C}_i which lies on the surface of the model.
3. Using the correspondences from step 2, find the optimal rotation \mathbf{R}_k and translation \mathbf{T}_k , based on the SVD approach described in Chapter 4 i.e. solving for rotation and translation using Eqns. (4-25) to (4-30).
4. Apply the transformation from step 3 to all data points \mathbf{M}_i . Update the cumulative parameters $\bar{\mathbf{R}}$ and $\bar{\mathbf{T}}$ using Eqns. (7-3) to (7-4) based upon the incremental transformations \mathbf{R}_k and \mathbf{T}_k .
5. If a stopping criterion is satisfied, terminate, else, go to step 2.

There are several stopping criteria that could be used with the ICP [117]. Appropriate criteria should be selected based on the peculiarities of the geometry to be registered. Also, different types of distance metrics have been introduced as a means of improving stability and convergence rate.

The most computationally expensive step in the ICP algorithm is the computation and selection of closest points (step 2 above) [117, 131, 142]. Step 3 could be considered as a problem of estimating the relative pose of two point sets in which point correspondences are known [117]. The singular value decomposition (SVD) method discussed in Section 4.4.6.2 of Chapter 4 provides a numerically efficient method for solving such a problem [116].

7.4.2 Efficient Variants of the ICP

The ICP has become the dominant method for aligning 3-D models based purely on the geometry, and sometimes colour of the meshes [143]. Although the ICP is well known, it suffers from some fundamental drawbacks. These problems generally relate to identifying the closest points and optimisation [2, 143, 144]. In particular, the problems relate to the computationally intensive process of identifying the closest point on the geometric model for each point at each iteration, slow convergence and convergence to local minima. Therefore, most solutions have generally focussed on the following [131, 140, 143]:

1. computational efficiency in the closest point selection process.
2. avoiding local minima by using other metrics for the objective function.

3. using additional information besides geometry (e.g. colour, curvature).
4. setting appropriate termination criteria.

Based on these solutions, numerous variants of the ICP currently abound [143].

Rusinkiewicz and Levoy [143] studied some of the variants of the ICP, focussing on speed of convergence, accuracy, and ability to handle "difficult" geometry. Classification of variants was based on how they affected the algorithm, i.e. selection of points, matching of points, weighting of point pairs, rejection of some point pairs, assigning an error metric based on point pairs, and minimising the error metric. Simon [117] worked on the design, implementation and validation of fast and accurate methods for performing 3-D shape-based registration. By carrying out speed enhancements to the ICP, fast registration was achieved with a computation time reduced by a factor of nearly two orders of magnitude. The high accuracy was achieved through intelligent data selection and online accuracy estimation. Fitzgibbon [141] used a standard non-iterative non-linear optimiser, the Levenberg-Marquardt algorithm for minimizing the cost function. Ristic and Brujic [131] achieved computational efficiency by first fitting to an approximate polyhedral model before switching to NURBS. Further speed improvement was achieved by using a subset of the measured data at all but the final stages of the iteration. It was also suggested that the standard deviation of the measurement noise, if known, should be the basis of the ICP termination criteria.

The distance metrics commonly used in the ICP include [145]:

1. point-to-point distance [139], which uses the Euclidean distance between the corresponding points. However, for certain types of data and initial positions, the ICP algorithm based on this metric converges slowly.
2. point-to-plane [146], which uses the distance between a point and a planar approximation of the surface at the corresponding points. However, for noisy data sets or data which are a significant distance apart, the ICP fails to converge.

As an improvement on the above error metrics, Mitra et al [145] developed a point-to-surface metric which uses the distance between a point in the measured data and a surface of the model data for the ICP. This approach enables an improvement in the approximation of the distance between the measured data and model data, leading to

stable behaviour when the datasets are far apart and faster convergence rate. Consequently, an optimisation framework was proposed based on an objective function consisting of a second order approximant to the squared distance between the measured and model data. The local approximants are not only valid at the query point, but also in the neighbourhood around the query point. The error metric is given as

$$\varepsilon(\mathbf{R}, \mathbf{T}) = \sum_{i=1}^N d^2(\mathbf{R}\mathbf{q}_i + \mathbf{T}, \Phi_p) \quad (7-5)$$

where \mathbf{q}_i is a point on the surface of the measured data, Φ_p is the model surface, $d^2(\mathbf{R}\mathbf{q}_i + \mathbf{T}, \Phi_p)$ is the squared distance function to the surface Φ_p and N is the number of corresponding points. Two methods used to compute a quadratic local approximant to the squared distance function include:

1. use of local curvature of the surface to incorporate second order information into the distance function on-demand.
2. pre-computing the approximation of the global error by locally fitting quadric patches to the squared distance function to surface. In order to efficiently utilise this method, a special octree-like data structure, the d^2 tree proposed by Leopoldseder et al [147] was used to store the pre-computed quadric information.

The d^2 tree data structure can be used for both 2-D and 3-D data. The size of the cells is determined by the parameters used in initialising the data structure. These parameters include the maximum number of levels allowed and the error threshold. A higher error threshold enables more cells to estimate the squared distance function, thus leading to faster convergence of the algorithm.

Yamany et al [142] proposed a method for reducing the computational complexity of the ICP by applying a grid closest point (GCP) technique and a genetic algorithm to minimise the cost function describing the mismatch between two datasets. A 3-D space, \mathbf{G} , is described, which encloses the two datasets, where each of its cells stores displacement vectors representing the displacement from the closest point in the model data to the measured data. The GCP transform thus uses the information stored in \mathbf{G} to calculate the displacement between the two data sets. The GCP transform is

spatially quantized and its accuracy is largely dependent on the selection of the quantisation step. Therefore, in regions close to the model data, a fine mesh is created, while a coarse mesh is created in other regions.

7.5 Registration using a 3-D look-up table

7.5.1 Description of method

In the literature, a significant amount of work has been put into the process of improving the computational efficiency of estimating the closest point during the ICP through pre-computing approximants in a grid. For example, Mitra et al [145], Yamany et al [142] and Leopoldseeder et al [147] have all used this approach successfully. Although Pirouet [148] does not use the ICP, in a similar vein, Pirouet [148] investigated a novel approach to the registration problem using pre-calculation of look-up tables (LUTs) on a regular grid. Each grid point stores in the LUT Taylor series expansions of the vector displacement to the nearest surface. This has been found to provide a robust and efficient method in two dimensions. Pirouet [148] further extended the 2-D method to 3-D, however, only the 0th and 1st order terms of the Taylor series coefficients were used. As highlighted by Mitra et al [145], high accuracy in interpolation is expected when higher order terms of the Taylor series approximation are used. In addition, the method makes use of a regular grid, which leads to large memory requirements and computational inefficiency, making its application to typical industrial inspection tasks impracticable. Therefore, the intent of this work was to extend the data structure of Pirouet [148] to use higher order approximants and reduce memory requirements by using an adaptive grid. The adaptive grid would have multiple resolutions in the LUT, thus a fine grid would be created in areas around the model data and a coarse grid in other areas. It was proposed that the ICP would be implemented with the use of a pre-calculated adaptive LUT that would store up to 2nd order Taylor series coefficients to improve computational efficiency in calculating the distance to the closest points.

Consider a cell in a 2-D LUT consisting of four neighbouring grid points as shown in Figure 7-7, with the centre of the cell set as the origin of a local coordinate system. The point **P**, is part of the measured data $M_{i=1...N}$, where we would like to calculate the distance to its nearest corresponding point **Q**, in the model data, $C_{i=1...M}$. If we assume that the 2-D LUT has been initialised, then we would know the distances of

each of its grid points to the model data. These distances would be based on the distance metrics discussed earlier in Section 7.4.1. The components of the vector displacement between measured points $M_{i=1...N}$ and their corresponding closest points $C_{i=1...M}$, $[D_x, D_y]$, can be combined to give an estimate of the distance between each pair of corresponding points by applying the Taylor series approximation to each vector component separately. Therefore, both absolute distance and directional information should be stored when initialising the LUT data structure.

By a simple extension to 3-D space, we can simply approximate the distance of **P** to **Q** by using the 3-D vector form of the Taylor series approximation. The 3-D form of the Taylor series approximation is created by applying the scalar form, which is a function of three variables ($f(x + \delta x, y + \delta y, z + \delta z)$), to the x , y and z coordinates separately. Therefore, we obtain $D_x(x + \delta x, y + \delta y, z + \delta z)$, $D_y(x + \delta x, y + \delta y, z + \delta z)$, and $D_z(x + \delta x, y + \delta y, z + \delta z)$ respectively. The following equation retains the 0th, 1st and 2nd terms in the respective scalar form approximations as:

$$f(x + \delta x, y + \delta y, z + \delta z) \approx f(x, y, z) + [\delta x f_x + \delta y f_y + \delta z f_z] + \frac{1}{2!} [\delta x^2 f_{xx} + \delta y^2 f_{yy} + \delta z^2 f_{zz} + 2\delta x \delta y f_{xy} + 2\delta x \delta z f_{xz} + 2\delta y \delta z f_{yz}] \quad (7-6)$$

where Δx , Δy and Δz are the size of the cell along the X, Y and Z axes of the global coordinate system of the LUT, and δx , δy and δz are the distances to the origin of the cell along the X, Y and Z axes of the local coordinate system. The centre difference approximations to the first derivatives, second derivatives and cross derivatives are given in Eqns. (7-7) to (7-9) below:

$$\begin{aligned}
 f_x &\approx \frac{f(x + \Delta x, y, z) - f(x - \Delta x, y, z)}{2\Delta x}, \\
 f_y &\approx \frac{f(x, y + \Delta y, z) - f(x, y - \Delta y, z)}{2\Delta y}, \\
 f_z &\approx \frac{f(x, y, z + \Delta z) - f(x, y, z - \Delta z)}{2\Delta z}
 \end{aligned}
 \tag{7-7}$$

$$\begin{aligned}
 f_{xx} &\approx \frac{f(x + \Delta x, y, z) - 2f(x, y, z) + f(x - \Delta x, y, z)}{\Delta x^2}, \\
 f_{yy} &\approx \frac{f(x, y + \Delta y, z) - 2f(x, y, z) + f(x, y - \Delta y, z)}{\Delta y^2}, \\
 f_{zz} &\approx \frac{f(x, y, z + \Delta z) - 2f(x, y, z) + f(x, y, z - \Delta z)}{\Delta z^2}
 \end{aligned}
 \tag{7-8}$$

$$\begin{aligned}
 f_{xy} &\approx \frac{[f(x + \Delta x, y + \Delta y, z) - f(x + \Delta x, y - \Delta y, z)] - [f(x - \Delta x, y + \Delta y, z) - f(x - \Delta x, y - \Delta y, z)]}{4\Delta x \Delta y}, \\
 f_{xz} &\approx \frac{[f(x + \Delta x, y, z + \Delta z) - f(x + \Delta x, y, z - \Delta z)] - [f(x - \Delta x, y, z + \Delta z) - f(x - \Delta x, y, z - \Delta z)]}{4\Delta x \Delta z}, \\
 f_{yz} &\approx \frac{[f(x, y + \Delta y, z + \Delta z) - f(x, y + \Delta y, z - \Delta z)] - [f(x, y - \Delta y, z + \Delta z) - f(x, y - \Delta y, z - \Delta z)]}{4\Delta y \Delta z}
 \end{aligned}
 \tag{7-9}$$

In Pirouet's method [148], a 3-D array, \mathbf{G} , is created, consisting of regularly spaced grid points that completely enclose the model and measured point cloud datasets. \mathbf{G} is populated with the distance of each grid point to the nearest triangle facet on the model data, while three 3-D matrices \mathbf{G}_x , \mathbf{G}_y and \mathbf{G}_z , are populated with each component of the vector displacement from each grid point to the closest point on the nearest triangle facet on the model data (this implies that $\mathbf{G} = |\mathbf{G}_x^2 + \mathbf{G}_y^2 + \mathbf{G}_z^2|$). \mathbf{G}_x , \mathbf{G}_y and \mathbf{G}_z are therefore the combined LUT for approximating the vector displacement of each grid point in enclosing volume to the model data, $\mathbf{C}_{i=1\dots M}$.

Pirouet [148] describes methods for calculating the distance of a 3-D point to triangle facets and line segments. From say \mathbf{G}_x , the derivatives with respect to the X, Y and Z axes are calculated and used to populate the arrays $\mathbf{G}_{x_{fx}}$, $\mathbf{G}_{x_{fy}}$ and $\mathbf{G}_{x_{fz}}$. These arrays store the derivatives which are the Taylor series coefficients at each grid point. In using the LUT, if we consider a measured point, \mathbf{P} , we first find the closest grid point, \mathbf{G}_i in the LUT with indices (j, k, l) . The closest grid point indices (j, k, l) of \mathbf{G}_i can easily be calculated based on the known size of each cell Δx , Δy and Δz , along each axis. Thus, the indices (j, k, l) are used to extract from $\mathbf{G}_{x_{fx}}$, $\mathbf{G}_{x_{fy}}$ and $\mathbf{G}_{x_{fz}}$

the Taylor series coefficients at G_i , which are required in Eqns. (7-6) and (7-9) to calculate $D_x(j, k, l)$, which is the approximation of the X component of the vector displacement of P to the model data, $C_{i=1...M}$. Using a similar approach, we can separately approximate the Y and Z components of the vector displacement of P to Q. Thus all the elements of D_x , D_y and D_z are respectively populated. Therefore, the values stored in these three matrices can be used to approximate the distance of each i^{th} point in the measured data, $M_{i=1...N}$ to its corresponding closest point in the model data, $C_{i=1...M}$. Thus, during optimisation to obtain the optimal transformation between $M_{i=1...N}$ and $C_{i=1...M}$, for each new estimate of the transformation R and T we apply to $M_{i=1...N}$, the cost function, F, is given as

$$F = \sum_{i=1}^N \sqrt{D_x(w(i))^2 + D_y(w(i))^2 + D_z(w(i))^2} \quad (7-10)$$

where w is an $N \times 3$ matrix with each row containing the three indices (j, k, l) of each closest grid point to M_i .

Progress was made by the author in developing a data structure to efficiently handle the above, until the work done by Mitra et al [145] was published. They utilised the d^2 tree developed by Leopoldseder et al [147], which is similar in principle to the approach that had been proposed at the outset of this work. Therefore, the d^2 tree will be described in more detail here.

The d^2 tree is an octree data structure, where rather than having cells with the same size i.e. a regular grid of points, larger cells are created in the far field of the model data, while smaller cells are created in the far field and where the squared distance function, d^2 , of the model data is not differentiable i.e. near the medial axis of the model data (see Figure 7-8). Leopoldseder et al [147] highlight that the use of smaller cells near the medial axis of the model data is an important property for registration and active surface approximation problems. This strategy ensures higher approximation accuracy in the regions of interest and low approximation accuracy in other areas. The data structure stores values of d^2 and coefficients of quadratic approximating functions. The size of each cell is thus guided by its distance to the model data, the local geometry, the approximation error between the approximating function in a cell, and d^2 .

An octree cell structure, *AuxTree*, is used as a container for the d^2 data structure. The creation of Auxtree starts with a cube (called Level 0), which encloses the model data and regions of interest in 3-D space, which is further subdivided into eight sub-cubes. It is expected that the model data is available in the form of a triangulated surface, thus the list of triangle facets is processed to identify which of these sub-cubes contains any part of the model data and to further sub-divide. Each of these sub-cubes is then iterated until the cell size at the finest level is as small as the pre-defined allowable size of the structure. Therefore, these two conditions, contact with a triangle facet and minimum size of cell, determine the number of cells to be created and the resolution of Auxtree.

7.5.2 Description of C++ implementation

The adaptive LUT was implemented in C++ as a set of objects defined by C++ classes. At the top of the class hierarchy is the object, *PV_C_MeshLUT3D*, a container serving as the LUT's management structure. It provides an interface to the LUT, which is defined as a *PV_C_LookUpTable3D* object, thus shielding the underlying representation from the user. *PV_C_LookUpTable3D* contains the topmost level, which is a *PV_C_GridLevel3D* object that represents a cell which consists of 8 *PV_C_GridData* objects that represent each of the grid points of the cell. Each of the *PV_C_GridData* objects contain a pointer to two *PV_C_GridLevel3D* objects – the current level and the next level lower down the data structure's levels. In addition, each *PV_C_GridData* contains a *PV_C_ClientData* object which stores the Taylor series coefficients. Thus, the relationship between a cell and its sub-cells is basically through a linked list of *PV_C_GridData* objects (see Figure 7-9), where we can move between different levels of the data structure. The class definition for the C++ implementation is contained in Appendix A-5, however, Figure 7-10 shows a stripped version of the class hierarchy in UML class diagrams, indicating how the above classes relate to each other.

Once the data structure has been initialised, it is necessary to have an efficient means of navigating through the various cells. For every query point, we need to find the nearest grid point in the data structure to 'snap' to. On identifying the nearest grid point, we can then extract the Taylor series coefficients for estimating the distance of

the query point to the model data. Therefore, included in the implementation are definitions for iterator objects *PV_C_MeshLUT3DIter* and *PV_C_LUT3DIter*, which are similar in concept to the way iterators are used in linked-lists. *PV_C_MeshLUT3DIter* provides a simple interface to the user, thus hiding the implementation of *PV_C_LUT3DIter*, which actually provides the functionality for navigating through the data structure.

Although the basic framework of the adaptive 3-D LUT has been implemented, at the time the author was working on this aspect of the research, calibration of the SMS was identified as having a higher priority. This part of the project took longer than originally envisaged and therefore it was not possible to rigorously test the implementation of the adaptive 3-D LUT with simulated and experimental data. It is therefore suggested that future work could be dedicated towards this aspect.

7.6 Summary

The multi-sensor optical SMS developed at Loughborough has very high scan rates (of order 10^6 s^{-1}), and the calculated 3-D coordinates generate a point cloud which contains shape information of the object under inspection. This measured data can then be tessellated and exported to known CAD file formats for either visualisation in CAD software or for registration with a geometric dataset such as a CAD model. An algorithm has been described and implemented for efficiently tessellating the measured data of the optical SMS and exporting as an STL or obj file. The method has proven to be efficient with a point cloud of over 385,000 coordinates being tessellated in less than 2 seconds, with the tessellated point cloud data set exported as STL and obj files, and subsequently visualised in CAD software.

A CAD model contains exact specification of an object and is therefore desirable as a basis for comparison in inspection and product quality assessment. Accurate registration of free-form surfaces is thus an essential requirement for dimensional inspection and relevant to many branches of manufacturing industry. With such a large point cloud data set obtained from the SMS, it is therefore a challenge to efficiently and accurately match the point cloud with the CAD model. The proven method for matching point clouds to CAD models is the Iterative Closest Point (ICP) algorithm. However, its main drawback is the computationally intensive closest point

selection process, which involves the selection of corresponding points between two point cloud data sets – the measured data and model data. Thus, in the literature, many variants of the ICP have been developed to tackle this aspect of the method.

From previous work described in the literature, the use of pre-calculated look up tables of vector displacements to a surface is expected to be robust and efficient for 2-D and 3-D point cloud data. Rather than using a regular grid, adaptive grids have been developed, which would improve computational efficiency by ensuring that smaller cells are created in the near field of the model data and larger cells at the far field. Although an adaptive spatial decomposition algorithm for 3-D registration using the ICP has been implemented in C++, it has not been fully tested, and further work in this area is suggested. However, it is thought that this implementation has the potential for meeting the requirements of a practical inspection system.

7.7 Tables

Table 7-1: Decimal values for identifying triangle facets that can be created from 4 neighbouring pixels

Id of valid pixels	Id of triangles created	Decimal value
0, 1, 2	Triangle 1	7
0, 2, 3	Triangle 2	13
1, 2, 3	Triangle 3	14
0, 1, 3	Triangle 4	11
0, 1, 2, 3	Triangle 1 and Triangle 2	15

Table 7-2: Specification for binary format of an STL file [133]

Number of Bytes	Type of data	Description of parameter
80	Unsigned char	Header of file, usually describing contents and history of file
4	Unsigned long int	Number of facets in file
4	Floating point	X component of facet normal
4	Floating point	Y component of facet normal
4	Floating point	Z component of facet normal
4	Floating point	X coordinate of vertex 1
4	Floating point	Y component of vertex 1
4	Floating point	Z component of vertex 1
4	Floating point	X coordinate of vertex 2
4	Floating point	Y component of vertex 2
4	Floating point	Z component of vertex 2
4	Floating point	X coordinate of vertex 3
4	Floating point	Y component of vertex 3
4	Floating point	Z component of vertex 3
2	Unsigned int	Attribute byte count, normally set to 0 and used as a spacer between each facet

7.8 Figures

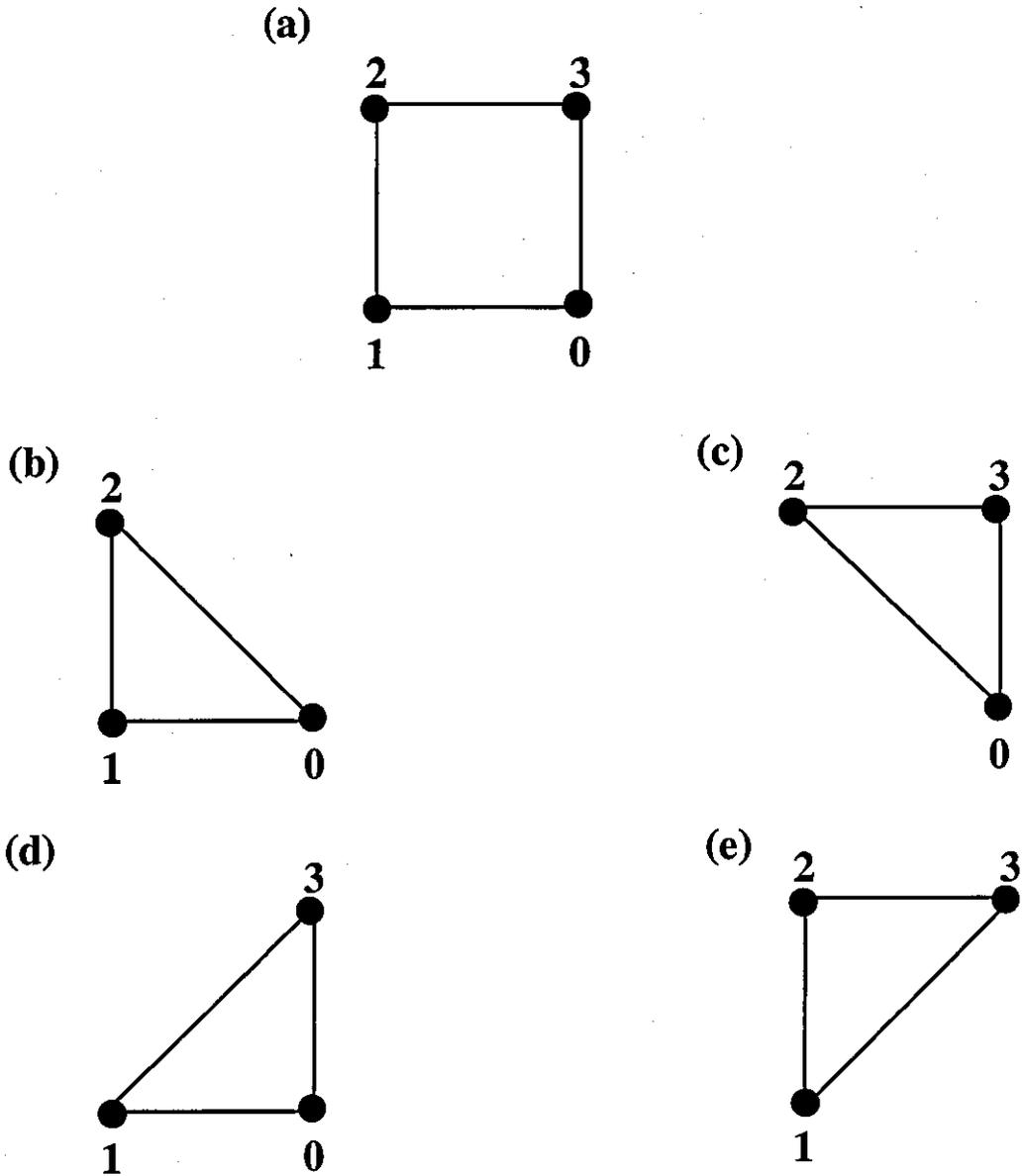


Figure 7-1: Four neighbouring pixels showing all the possible triangles that can be created. (a) the 4 grid points. (b) *Triangle1* created when vertex-3 is not valid. (c) *Triangle2* created when vertex-1 is not valid. (d) *Triangle3* created when vertex-2 is not valid. (e) *Triangle4* created when vertex-0 is not valid. When there are no invalid pixels, either *Triangle1* and *Triangle2* or a quadrilateral are created.

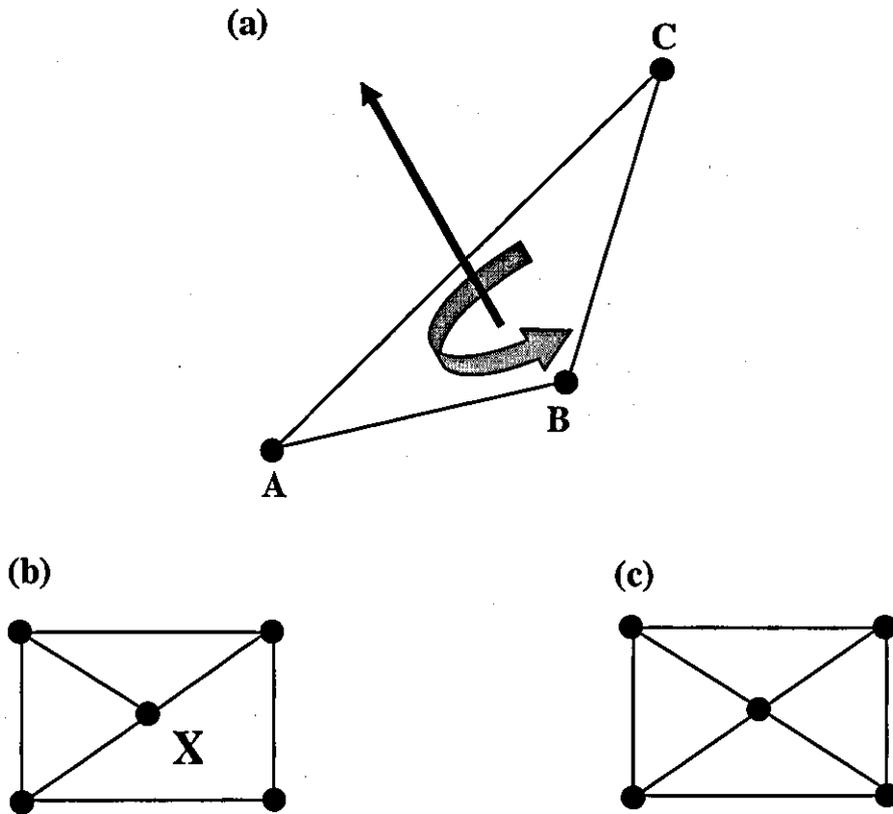


Figure 7-2: Convention for creating triangular facets in STL files. (a) Counterclockwise selection of vertices to create triangle facet and set the direction of the facet's normal. (b) Triangle with 'X' does not obey the vertex-to-vertex rule. (c) Triangle with 'X' now split into two in order to adhere to vertex-to-vertex rule.

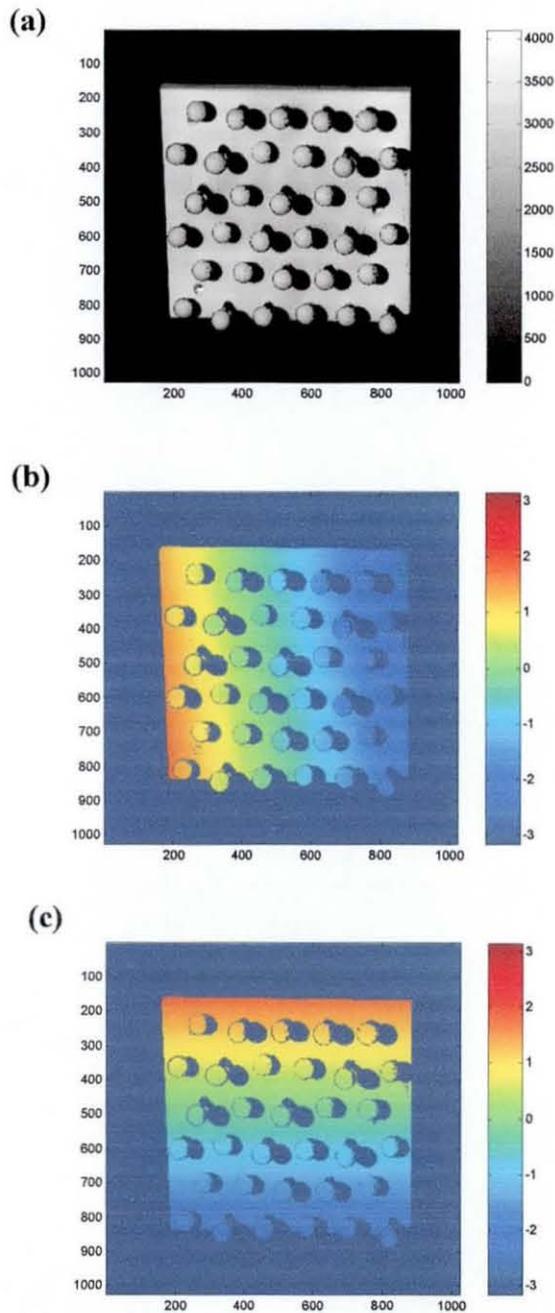


Figure 7-3: Images of multiple-sphere artefact acquired by C_1P_1 . Images show values only at valid pixels. (a) Greyscale intensity image (b) Unwrapped phase map with horizontal fringe orientation, ω_x (c) Unwrapped phase map with vertical fringe orientation, ω_y .

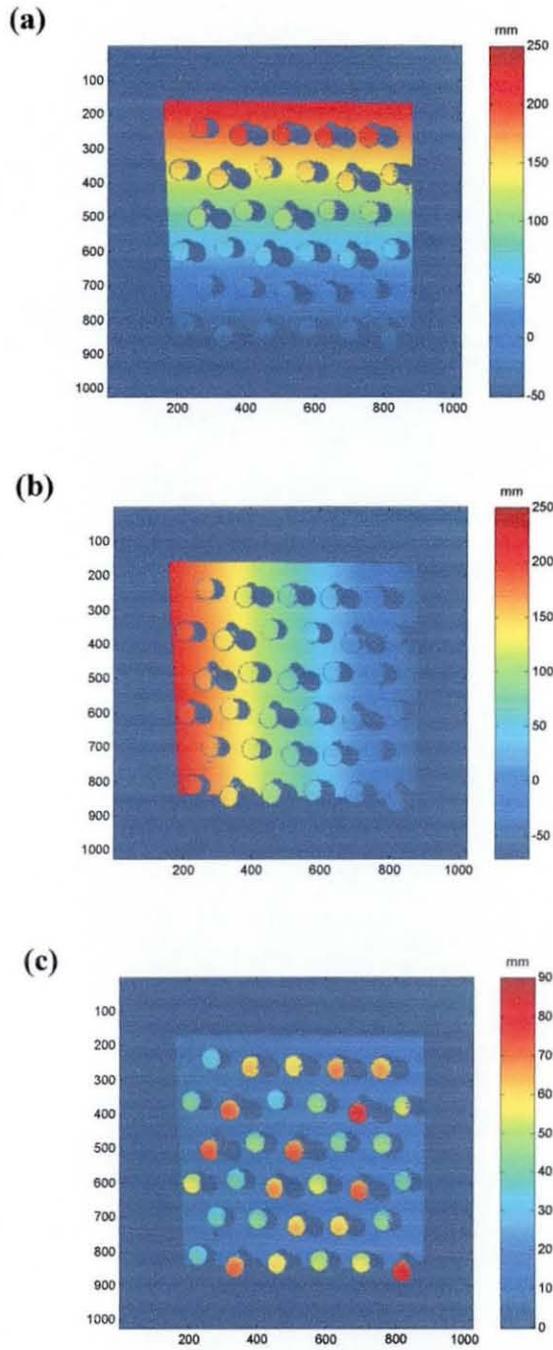
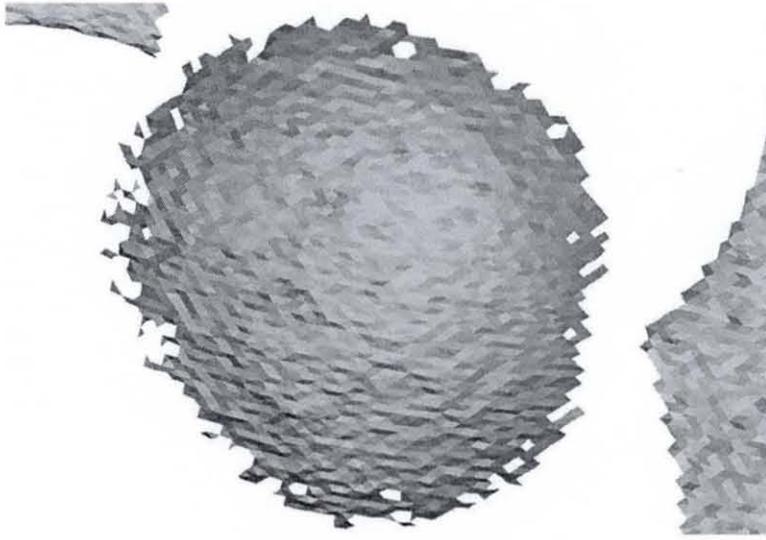


Figure 7-4: Images of x , y , z coordinates calculated using optimised sensor parameters and phase maps acquired by C_1P_1 . Images show values only at valid pixels. (a) x coordinates. (b) y coordinates. (c) z coordinates.

(a)



(b)

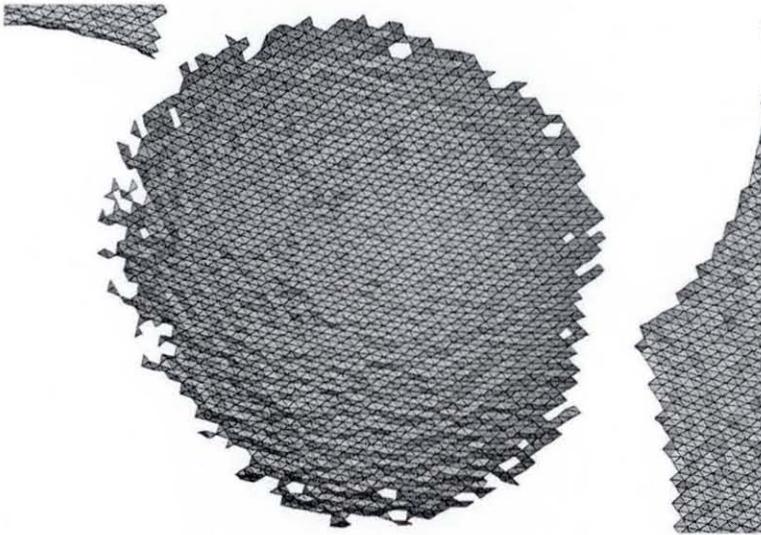


Figure 7-5: Close-up of one sphere to highlight the large number of triangular facets that make up the tessellated point cloud. (a) Image showing the rendered triangular facets with artificial lighting. (b) Image showing the rendered triangular facets and their edges.

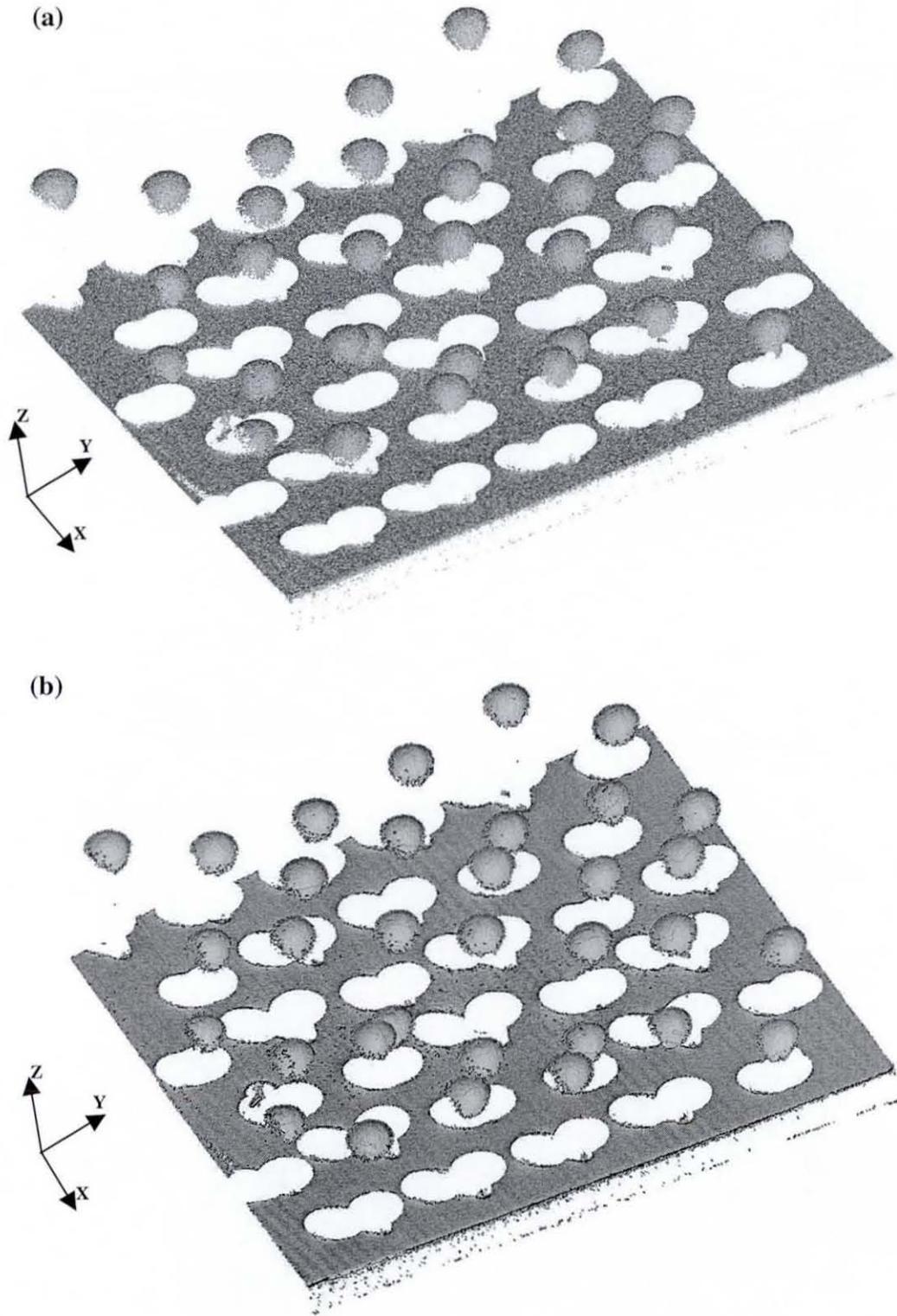


Figure 7-6: Tessellated point cloud showing the rendered triangular facets with artificial lighting. (a) The facets exported from the SMS without smoothing. (b) The facets with smoothing applied.

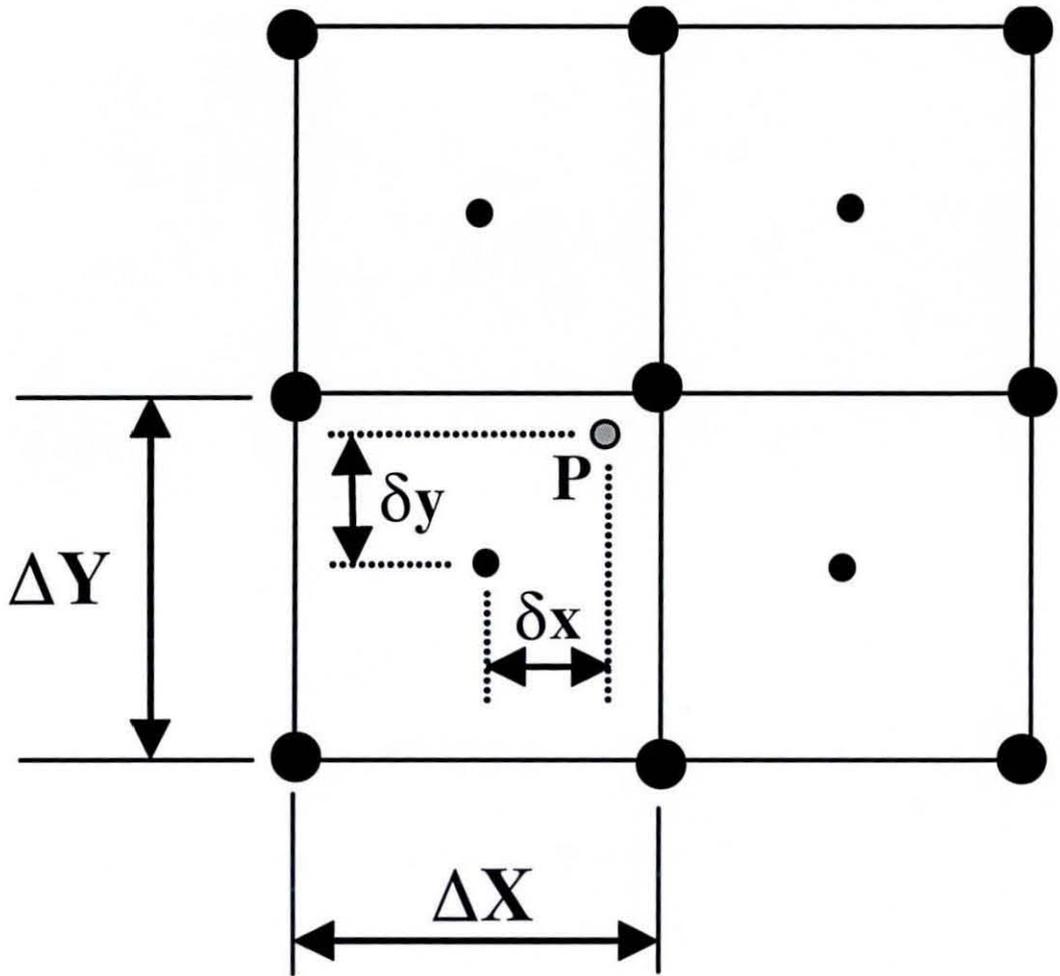
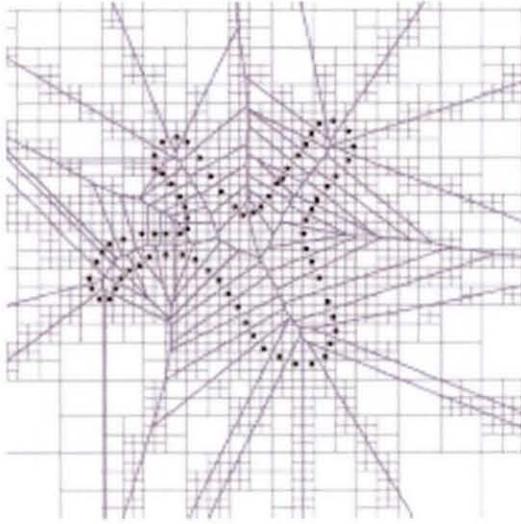


Figure 7-7: 2-D LUT showing the relationship between a measured point, P , and the grid points of a cell.

(a)



(b)

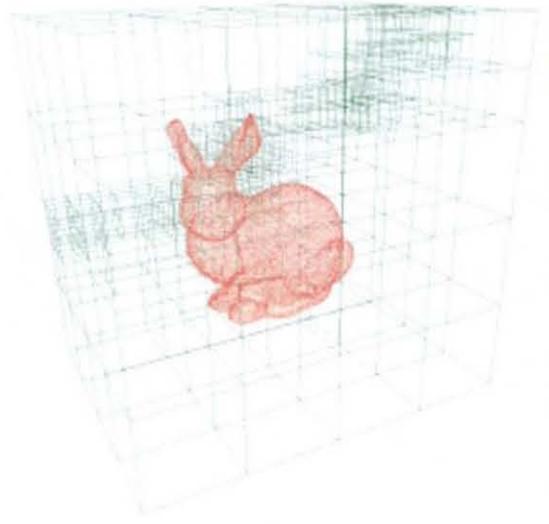


Figure 7-8: Example of an adaptive mesh for 2-D and 3-D point cloud data with smaller sized cells in the regions of interest and where the squared distance function, d^2 , is not differentiable (i.e. the medial axis) [145].

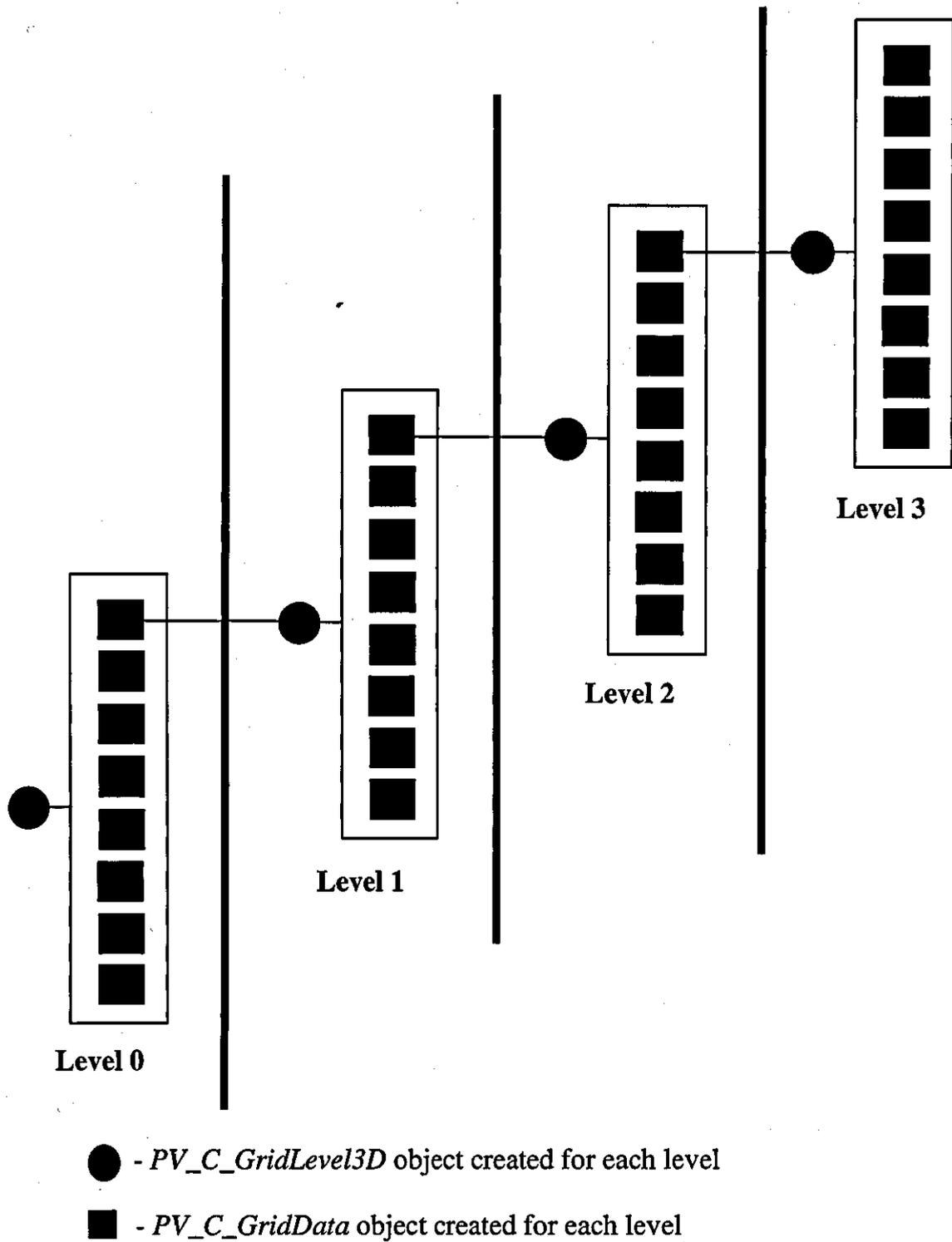


Figure 7-9: Example of how *PV_C_GridLevel3D* and *PV_C_GridData* objects are used for creating a multi-resolution grid. In this case, at each level, it is assumed that only the first *PV_C_GridData* object contains part of the model data.

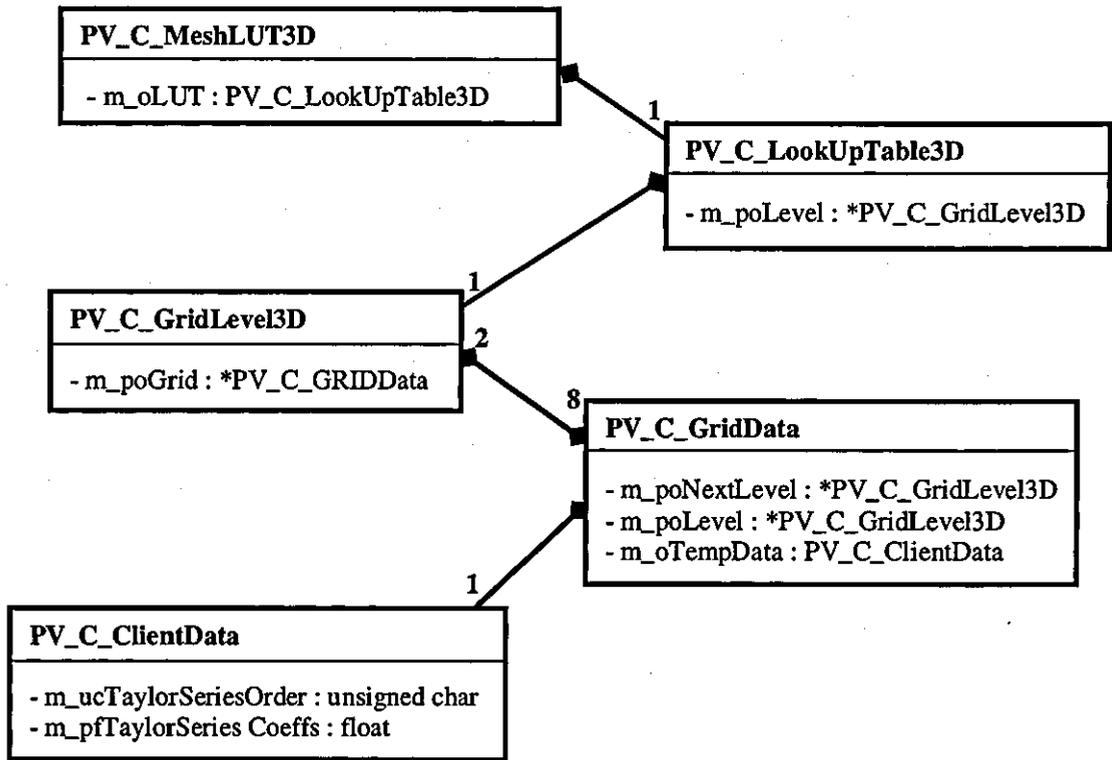


Figure 7-10: Stripped version of the class hierarchy of C++ implementation showing the relationship between the implemented objects.

8 Application of Multi-Sensor Optical Shape Measurement System

8.1 Introduction

The automated method for calibrating the multi-sensor optical shape measurement system (SMS) has been described in Chapter 6. In Chapter 7, the functionality for exporting the measured data of the SMS (i.e. the point cloud) for visualisation in CAD software or matching to an ideal geometric data set has also been described. After calibration, the SMS is able to measure the surface shape of an object and the point cloud can be viewed in CAD software or post processed in other ways deemed suitable for specific applications. An example application is in manufacturing quality control, where the point cloud is matched and compared against a CAD model of the measured object, to quantify product manufacture error. With this approach, the SMS could act as a filter in the product manufacturing process, accepting or rejecting each product or batch of products. In another approach, the point cloud could be processed to identify specific features which would be used as acceptance criteria. The SMS could also be used as an instrument to measure changes in shape due to mechanical or thermal loads, potentially providing all three displacement components.

The automated calibration process for the multi-sensor SMS using three different 3-D artefacts has been described in Chapters 3 and 6, with calibration using the ball bar artefact proven to be robust, reliable and efficient for different measurement volumes in the laboratory environment. Airbus UK, the sponsor of the research project, was keen to use the SMS to measure out-of-plane displacement fields during structural tests, where a test specimen is compressed in a test rig. Discussions in this chapter will therefore focus on the use of the SMS in an industrial environment. The importance of structural testing in the Aerospace industry will be highlighted, including how optical methods are now applied for measuring displacement. The author was involved with Dr. Coggrave of Phase Vision Ltd in using the optical SMS at the Airbus Filton site for the A380 and A300 test programs. In these tests, while the author's primary role was calibrating the SMS and generating 3-D data of the test specimen, Dr. Coggrave's role was technical support for the shape data acquisition system of the SMS (i.e. acquisition of absolute phase maps), post processing and detailed analysis of the SMS measured displacement fields. A description of how the

SMS is set up, calibrated and adapted for measuring out-of-plane displacement will be made. The implementation of the new calibration procedure at the Airbus site will also be discussed, where the challenges of setting up and calibrating the SMS at an industrial site will be highlighted. Finally, the results of one of the tests, where the SMS was used for measuring out-of-plane displacement fields, will be discussed. It is worth noting that part of the contents of this chapter have been included in a recently accepted SPIE conference paper [24].

8.2 Structural testing in the Aerospace industry

8.2.1 Types of structural tests

The aim of structural tests in the Aerospace industry is the validation of numerical or mathematical models and providing information for the development and certification of aircraft. The main challenges of a structural test include determining as accurately as possible the load distribution that will be imposed on the aircraft while in flight, accurately replicating and applying this load distribution to a test specimen, and accurately measuring the structural behaviour of the test specimen (e.g. displacement, stress and strain) as a result of the application of such loads. The test specimen could be the whole airframe structure, a combination of components, or individual components. Generally, there are two major types of tests, namely static and fatigue tests. In static tests, load is applied to the test specimen over a short time frame, while in fatigue tests, load (usually smaller than that applied in static tests) is dynamically applied to the test specimen over a long period of time ranging from days to months.

In terms of test programs for aircraft development, structural tests can be broken down into two major types: (i) certification tests and (ii) development tests. Certification tests are structural tests which must be carried out before an aircraft can be allowed in service in a particular geographical location or airspace. As the key objective of certification tests is aircraft safety, they are prescribed and supervised by a certifying body which could be national or regional in terms of authorisation. Examples include Civil Aviation Authority (CAA) for the UK and the European Aviation Safety Agency (EASA) which is an arm of the European Union responsible for aviation safety. One of the key functions of the EASA is the development of Joint Aviation Requirements (JAR) which are used in the aircraft certification process. Certification tests are classified in decreasing order of complexity and importance as

Category 1, Category 2 and Category 3. For example, in a Category 1 test such as the major fatigue test of the complete airframe structure, the certifying body is informed about the date of the test, briefed about details of the test, and their representatives must be in attendance. In Category 2 and 3 tests, the presence of representatives of the certifying authority is not required. The deliverables from a certification test include test results and relevant documents which should demonstrate compliance with airworthiness requirements (e.g. JAR).

On the other hand, development tests are tests which are business driven and not safety driven. Therefore, certifying authorities do not need to be informed of the event of development tests nor of their results. However, development test results could be sent as supporting documents when certification test documentation is submitted to a certifying authority. Development tests are typically incremental in nature and provide a means of developing new design, analysis and measurement methods. The new calibration method developed in this work was used to calibrate the optical SMS used in a series of four development tests at the Airbus site. Results from one such test will be discussed in the next section.

From discussions with professionals in the Aerospace industry, there are no particular standards in the industry that target optical sensors. However, there is presently an EU-funded initiative called SPOTS [149], which is aimed at producing a standard for optical sensors used for strain measurement.

8.2.2 Compression panel tests

In modern civilian aircraft, the wing structure is typically a box beam structure with a central skeleton consisting of stringers (longitudinal stiffeners), longitudinal spars and transverse ribs [150](See Figure 8-1). Attached to the central fixed structure are movable surfaces such as the flaps, and everything is covered with a 'skin' (consisting of skin panels which cover the top and bottom surfaces of the wing), to create the wing's overall dynamic shape [150]. A compression panel is a test specimen that is representative of the top section of the wing which experiences compressive stresses during flight as the wing responds to 'lift' forces, and typically consists of a top skin panel and stringers.

A compression panel test involves the application of load on a compression panel to measure strain, in-plane and out-of-plane displacement, loading rate and load at failure. The test specification and other detailed parameters relating to the structural tests in which the author was involved are confidential to Airbus and will not be discussed here. However, a general description of the test procedure will be given. At the outset, the design of the compression panel is given in the test specification prepared by the Stress Office at Airbus. Based on initial Finite Element (FE) analysis, critical loads and the regions of high stress or strain would have been identified. The identified critical loads provide information on how load would be applied during the tests. Also, strain gauges are positioned at the identified regions of high stress or strain on the compression panel in order to make measurements during the test. Strain gauges are devices that experience a change in resistance when they are stretched or strained, and thus able to detect very small displacements, usually in the range 0-50 μm , but also extendable to measure displacements of up to 50mm [89]. In previous structural tests at Airbus, linear variable displacement transducers (LVDTs) had been used for measuring out-of-plane and in-plane displacement. However, in recent times, full-field optical techniques have now replaced LVDTs for measuring displacement. Optical methods are used in structural testing for [151]:

- Validation of numerical or experimental models
- Measurement of the shape of components for quality assurance
- Measurement and monitoring of the shape or strain distribution during testing for the determination of buckling, deformation modes and stress concentrations
- Damage detection by identifying dents and scratches barely visible to the eye

The test rig used for these tests (supplied by J.R. Dare Ltd) has a capacity of 2000 Tonnes, but is calibrated to 1700 Tonnes. The rig is run in a closed loop service system where a piston applies axial load from the bottom platen, while four load cells (transducers that convert force to electric signals) equally measure the applied load and provide feedback to a computer. There are typically three sets of runs, firstly two *settling* runs and a *failure* run. However, for some tests, a *limit* run could be required. In each of the two settling runs, the objective is to get the compression panel properly positioned in the test rig by applying load in increments of 87.5 kN to 700 kN, and

then down back to 0 kN. For the limit run, load is applied until the predicted limit load (obtained from FE analysis) for the compression panel, and then down back to 0 kN. However, in the failure run, load is continuously applied until buckling failure of the compression panel. Figure 8-2 shows a one-camera two-projector optical SMS facing a compression panel positioned in a test rig.

Apart from the SMS, high speed and low speed 2-D digital image correlation (DIC) is also used during structural tests at Airbus to measure in-plane displacement. It should be noted that 3-D DIC could be used to measure out-of-plane as well as in-plane displacement. In DIC, a random 'speckle' pattern is applied to the test specimen surface and as load is applied during the structural test, a series of images of the pattern are acquired and digitally stored. As the test specimen experiences deformation, there is a commensurate change in the state of the image. Thus, for each image, by comparing the undistorted or reference image of the pattern of dots with the equivalent deformed image, the deformation of the specimen can be calculated. In 2-D DIC, in-plane displacement is measured by applying the speckle pattern either through spray painting, or through printing on paper. The shift in the speckle pattern is determined by an iterative procedure which fits the deformed reference image to the actual deformed image. The iterative procedure involves many subsets of the image, where a displacement vector is calculated for each subset, thus an array of displacement vectors can be obtained.

Unlike the DIC technique where a region of interest (say 16×16 pixel sub-images) is used to obtain each displacement vector, for the SMS, measurements at each pixel is independent. Therefore, the SMS has better spatial resolution than the DIC technique. However, as a result of direct viewing of the specimen, the DIC technique does not suffer from the shadowing problems of the SMS. Although the use of large sub-images could potentially increase the accuracy of the DIC technique, it leads to more computation time and an averaging effect on the resulting displacement field [152]. Other factors affecting the accuracy of the DIC technique include the speckle size and the correlation algorithm used [152].

Generally, the advantage of using optical methods for structural testing include [151]:

- Low recurring cost and low/medium capital cost
- Ease of application
- Ease of interpretation of results and direct relevance (2-D or 3-D deformation or strain) to FEA data
- Accuracy, reliability and repeatability
- Robustness and speed of application
- Size of systems and portability

8.3 Set up of SMS for measuring out-of-plane displacement

A one-camera two-projector SMS was set up for compression panel testing at Airbus' site in Filton, Bristol (see Figure 8-2). As the compression panel is primarily flat, there is no problem with shadowing of the part, and therefore, one camera is sufficient for shape data acquisition. The camera and projectors were mounted on an aluminium rail which is in turn mounted on two camera tripods. The camera's field of view was over $1.5 \times 1.5 \text{ m}^2$ with the stand-off distance of the centre of the calibrated volume, V , approximately 4.92 m. SMS set up involves relative positioning of cameras and projectors with respect to V , to ensure proper imaging and projection of fringe patterns. Firstly, the camera and projectors had to be aligned to ensure that the measurement volume filled a significant portion of the respective fields of view of the sensors. To achieve this, a projector screen was placed at the centre of V , (i.e. centralised within the camera's field of view), with a cross hair pattern (a rectangle with a line going through the centre both from top to bottom and left to right) projected from the two projectors respectively. The two images of the cross hair pattern on the projector screen provide information on alignment of the projectors with respect to the camera and the measurement volume. The objective is to ensure that the images of the cross hair are close to the centre of the cameras field of view. In addition, the centre lines of the respective cross hair patterns projected from the projectors should be approximately coincident on the projector screen. Once the cross hair patterns were projected, the projectors were then adjusted accordingly to ensure proper alignment of camera and projectors with respect to the measurement volume. The final position and orientation of the camera and projectors are shown in Figure 8-2. On completion of the alignment, an XML file was created and in the acquisition section, the parameters for shape data acquisition by the SMS were initialised. The

exposure time of the camera in particular had to be reset a number of times to ensure that minimal pixels were saturated during shape data acquisition.

The illumination lamp in the projectors generates a significant amount of heat which necessitates the inclusion of a built-in fan to extract the heat generated from within the projector's interior. However, it was observed during set up that the air currents (as a result of the heat extraction from the projector) flowed towards the projector lens and led to waviness in certain regions of the projected images. This occurrence had not been previously observed when the SMS was used in the laboratory. Clearly, measurement accuracy could be affected as a result of the projection of unsteady fringe patterns onto the measured object during the shape data acquisition process. To overcome this problem, air ducts were designed by Dr. Coggrave of Phase Vision Ltd., and manufactured using Rapid Manufacturing technology, to clip onto the exterior of the projector so as to divert the hot air being extracted from the projector away from its lens.

For the structural test, the compression panel is positioned laterally within the test cell and load is gradually applied to initiate buckling. To calibrate the SMS for this application, an area within the test rig should constitute the measurement volume where ideally the calibration artefact should be positioned and measured. However, the origin and orientation of the SMS' world coordinate system is the same as that in which the control points of the 2-D artefact (that is used to obtain initial estimates of sensor external parameters) are relative to. Therefore, the SMS was positioned at a convenient location away from the test rig and shape data for calibration was acquired. On completion of the shape data acquisition for calibration, the SMS was then moved into an appropriate position 4.92 m from the test rig (ensuring that the calibrated volume was set well within the test rig) with the panel approximately centred in the camera's field of view.

8.4 Description of calibration process at Airbus' site and results

8.4.1 Initialisation phase – shape data acquisition and sensor parameter initialisation

In calibrating the SMS for a large volume, the 2-D DLT method was used for the sensor initialisation process, with a 2-D calibration artefact providing control point coordinates. The process for sensor parameter initialisation using a 2-D calibration artefact has been described in Section 6.3 of Chapter 6. The 2-D calibration artefact is designed to be a basic, low cost, light weight object, consisting of a printed pattern of circles on a flat surface. When making measurements of the artefact, ideally, it should be positioned parallel to the frame of the SMS, near the centre of the proposed measurement volume. However, during measurements in large fields of view, it could be quite a challenge to find an appropriate way to keep the surface on which the pattern of circles is attached relatively flat, as well as parallel to the frame of the SMS. An artefact was therefore designed to consist of an aluminium snap-frame (used for framing A0 (841×1189 mm²) posters), and to improve rigidity, it was screwed on to a 12 mm thick medium density fibre (MDF) board. The pattern of circles was printed on A0 paper with matt laminate finish in order to protect its surface and reduce specular reflection. It should be noted that the image of the pattern of circles could also be printed on a flat printable surface such as a *foamex* board, however, this option more than doubled the cost. The total cost of the planar artefact, including the cost of the aluminium snap frame, MDF board and printing is approximately £86. Figure 8-3 shows the planar artefact, consisting of an array of 13×9 circles of radius 35 mm with circle centre pitch at 81 mm.

In creating the image of the pattern of circles for large volumes, the key issues considered related to easy identification of its orientation and easy estimation of the circle centre coordinates (i.e. estimation of control points) after printing on paper. For easy identification of the artefact's orientation, the circle at the centre of the artefact was set to a blue colour; circles on a diagonal were set to green and red, while all other circles were set to white, with the background in black. In addition, a line was created to connect the centre circle with the four closest circles to its north, west, east, and south respectively. To facilitate ease in measuring the pitch of the circle centres

once the image has been printed, tick lines were created at the edge of the image to coincide with the position of each row and column of circle centres.

Figure 8-4 shows the acquired images of the 2-D artefact including the detected circle centres. The residual of the DLT method using all circle centres as control points for C_1 , P_1 , P_2 was calculated as 6.64 μm , 18.92 μm and 15.32 μm respectively. Table 8-1 shows the initial estimates of all the sensor external parameters obtained by the 2-D DLT method. The acquired data of the 2-D artefact was also used to calculate initial estimates for the focal length of the camera and projectors. It should be noted that for the phase data acquisition in this case, the vertical fringe orientations were used to compute the image plane coordinates of the projector along the X axis, while the horizontal fringe orientations were used to compute the image plane coordinates along the Y axis. As discussed in Section 6.7.2 of Chapter 6, consequent on the manner in which the illumination beam is projected from each projector, the initial estimate for the principal point offset along the X axis of its image plane coordinate system should be non-zero. Therefore, the setting of the principal point offsets for both P_1 and P_2 , was set as $\xi_H = 0, \eta_H = 6\text{mm}$. For C_1 , the principal point offset was set as $\xi_H, \eta_H = 0$. The radial distortion coefficients for the camera and projectors were set as $k_1, k_2, k_3 = 0$. Figure 8-5 shows the relative distances between the sensor external parameters obtained using the 2 -D DLT method.

8.4.2 Refinement phase – shape data acquisition, shape data processing and bundle adjustment

The refinement phase of the calibration process, involving shape data acquisition, shape data processing and bundle adjustment has been described in Chapters 3 and 6. In the results to be discussed here, the ball bar artefact used for calibrating the SMS for a volume of approximately $1.5 \times 1.5 \times 0.9 \text{ m}^3$ was assembled using two 50.8 mm diameter coordinate measuring machine (CMM) reference balls and a combination of length bars joined together to give total length of 482.6 mm (i.e. 19"). Thus, the distance between the two sphere centres of the assembled ball bar (i.e. the sphere separation) was measured independently by a mechanical CMM as $727.2 \text{ mm} \pm 1.9 \mu\text{m}$. The surfaces of the respective spheres were spray coated with developer powder to give a matt white finish. However, the other parts of the ball bar, namely the two respective stalks of the CMM reference balls and the length bars, had to be covered

with black cardboard paper to prevent specular reflections and consequently, bad data being acquired.

For shape data acquisition, the ball bar artefact was positioned in the measurement volume by attaching to a tripod arm which was also attached to a camera tripod support (see Figure 8-6). However, the field of view of the camera implied that in order to calibrate for the upper regions of the measurement volume, the ball bar had to be positioned at a significant height. This proved to be clumsy at times, with the tripod arm and ball bar usually wobbling, and one had to wait for a few seconds for the oscillations to die away before making a measurement.

Prior to each measurement of the ball bar, the assembly (camera tripod, tripod arm and ball bar) was placed at some position in the measurement volume, with the tripod arm adjusted in order to set the orientation of the artefact. Measurements were made with the ball bar filling as much as possible the measurement volume by translating the assembly and adjusting the tripod arm to orientate the artefact vertically and horizontally, and also with rotations about the X, Y, and Z axes. These translations and rotations were estimated during the shape data processing stage of the calibration process. The calibration method therefore ensures easier calibration in an industrial environment as mechanical actuators are not required for accurately translating and rotating the artefact. In the data sets to be discussed in this Chapter, 32 poses of the artefact were measured in approximately 72 minutes. The acquired shape data was then used to carry out an off-line calibration in order to determine the optimal sensor parameters for the SMS.

The calibration section of the XML file for the measured data was initialised to set the parameters for the shape data processing and bundle adjustment stages. The key settings refer to parameters that specify directories where the shape data files are located and the parameters for sphere detection using the 3-D Hough transform (HT). Once the XML file had been set up, the off-line calibration was carried out automatically without further user intervention.

For each pose of the ball bar artefact, the two spheres were detected and 36 control points were selected automatically in point clouds computed from the two

camera-projector pairs (i.e., 144 control points per pose). Figure 8-7 (a) shows the centre coordinates of the detected sphere centres. During shape data processing, only 2 poses were automatically rejected (the spheres of the ball bar artefact were not detected in the point cloud of a camera-projector pair), thus data sets from 30 poses were used for the calibration and therefore, a total of 4320 control points were used in the bundle adjustment. From Figure 8-7 (b) which shows average error (computed using all camera-projector pairs from all good poses) of the SMS measurement of the ball bar (i.e. distance between the two sphere centres of the ball bar) and the CMM measurement, it can be seen that in up to 25 poses, the SMS measured the ball bar was shorter than the expected value. The combined rms is calculated as 9.74 mm and mean (based on absolute values) as 7.08 mm. This is as a result of errors in the initial sensor parameters used to compute the point cloud.

For the bundle adjustment process, the external parameters of the camera (C_1), and the k_2 and k_3 distortion parameters for all sensors were set as constant. As stated in Section 6.7.1 of Chapter 6, the difference between successive rms values of the objective function, F , was used as the criterion for convergence and set at 10 μm . Although this criterion was not met after 150 iterations, the ε_1 errors reduced from 2.23 mm to 0.63 mm, representing a 72% reduction, while the ε_2 errors reduced by 75% from 5.78 mm to 1.47 mm. Figures 8-8 (a), (b) and (c) shows the rate of convergence of F , ε_1 and ε_2 respectively. It can be seen that by the third iteration, F is close to the minimum and reductions are very small afterwards. Thus if the convergence criterion is increased by say a factor of 100, we would expect that convergence would have been achieved much earlier. It is therefore suggested that for the calibration of large volumes, a larger value for the convergence criterion could be used for the bundle adjustment process, as the inherent rapid convergence rate of the method ensures that the time cost of F being reduced further would only lead to very small changes in calibration parameters. Table 8-1 shows the value of the respective sensor parameters on completion of the bundle adjustment. Figure 8-9 shows the relative distances between the optimised sensor external parameters after bundle adjustment. Using the optimised sensor parameters, Figure 8-10 shows average error (computed using all camera-projector pairs from all good poses) of the SMS measured length of the ball bar and the CMM measured length, with combined rms calculated

as 0.94 mm (compared with 9.79 mm obtained using initial sensor parameters), and mean (based on absolute values) as 0.76 mm (compared with 7.08 mm obtained using initial sensor parameters).

Once calibration data (i.e. poses of the ball bar artefact) has been acquired with the SMS, it is assumed that the camera and projector's positions and orientation on the SMS rail do not change. In addition, focus or focal settings of the camera and projectors must remain fixed. Thus, in subsequent measurements with the SMS, the optimised sensor parameters were used for calculating respective point clouds from phase data acquired with each camera-projector pair.

8.5 Procedure for making measurements during test

After calibration, it was necessary to get the SMS appropriately set up and positioned for full-field measurement of the compression panel. This involved making test measurements to assess the gradient of the x , y , z coordinates, which indicate the level of tilt of the panel with respect to the axes of the SMS world coordinate system. Also the exposure time of the camera was set in the XML file to 19 milliseconds in order to be consistent with the illumination from the projectors relative to the compression panel, ensuring that minimal pixels are saturated during shape data acquisition.

In the structural test, prior to the application of load, the compression panel is measured in its initial state and subsequently, as compressive load is applied to the panel, measurements are made in quick succession with the SMS in order to measure out-of-plane displacement. The displacement is measured as the change in shape, which could be calculated from the change in the z coordinates at each valid pixel in the camera. For example, if the matrix of z coordinates at the initial state is Z_0 , and the matrix of z coordinates n seconds after the application of load is Z_n , the displacement at each pixel after n seconds, D_n , is calculated as:

$$D_n(i, j) = Z_n(i, j) - Z_0(i, j) \quad (8-1)$$

where i and j are the respective row and column indices of pixels valid in both Z_0 and Z_n .

It is therefore crucial that data acquisition is completed within a very short time frame. With shape data acquisition for each camera-projector pair typically completed

in less than 20 s, thus a time overhead would be incurred if the two camera-projector pairs acquire shape data during the structural test. Indeed, the objective of using a multi-sensor SMS is to avoid shadowing problems during data acquisition; however, the set up of the SMS relative to the compression panel did not reveal shadowing problems. Therefore, either of the two camera-projector pairs (C_1P_1 , C_1P_2) would acquire shape data during the structural test. Although C_1P_2 was selected for this test, the accuracy of the measured point cloud from both pairs is comparable.

8.6 Results from structural test

The SMS was used as part of the A380 test program and the results from test of 16th August 2006 will be discussed in this section. Figure 8-11 shows the compression panel that was used in this test, where during the failure run, buckling occurred at 3380 kN. A total of 40 measurements were made for the duration of the failure run and Figure 8-12 shows the load cycle with SMS sample times. Figure 8-13 (a) shows the displacement across the compression panel just before buckling failure, where 3344 kN load had been applied after 669 seconds. Across the panel, cross-sections (shown in Figure 8-13 (b)) have been taken along areas where more detailed information is required. Figure 8-13 (c) shows the distribution of displacement values across the panel. From Figure 8-13 (b), the black, blue, cyan and green line sections represent the path which lies approximately at the centre of each set of vertically tiled buckling mode shapes, and indicate displacement values ranging between approximately -11 mm and 8 mm. On the other hand, the red, yellow and green line sections in Figure 8-13 (b) represent the path which lies approximately at the location where stringers have been positioned (at the back of the panel), and show displacement values ranging between approximately -3.5 mm and 5 mm.

A similar pattern of displacement can be observed from the results of the Digital Image Correlation (DIC) system which measures out-of-plane displacement on the opposite side of the panel (as shown in Figure 8-14). The DIC measured displacement field is shown in Figure 8-15. It should be noted that, because the instruments were positioned on opposite sides of the specimen, the measurements were therefore made relative to two different coordinate systems. Thus, the buckling mode shapes are in opposite directions. Therefore in order to assess the level of agreement between the DIC and SMS measurements, the displacement values of the DIC were negated, the

column pixel coordinates of the SMS had to be flipped around (to extract x , y , z coordinates in same order with DIC), and an offset of -120 mm was introduced to the y coordinates of the SMS measurements. Thus the peaks and troughs of the displacement fields could be brought roughly into alignment. Comparison of DIC and SMS measurements of the two regions of interest (ROI) 'A' and 'B' shown in Figure 8-14 on the panel are discussed. For the SMS, the ROI was extracted at the approximate location of the speckle pattern on the opposite side of the specimen. The level of agreement between the displacement measured by the DIC and SMS for the ROIs is shown in Figures 8-16 and 8-17 respectively. From the figures, it can be seen that the maximum peak to trough displacement as measured by both the DIC and SMS is about 13 mm and 15 mm respectively. In addition, Airbus' Engineers confirmed that the buckling mode shape of the panel as measured by the SMS during the test were consistent with FE results. However, the author is not in a position to publish the details of this comparison as such information is confidential to Airbus.

8.7 Summary

Structural testing is typically carried out in the Aerospace industry for validation of numerical or mathematical models and providing information for the development and certification of aircraft. The major technical issues considered when planning for a structural test include accurately determining the load distribution that will be imposed on the aircraft while in flight, accurately applying the load distribution to a test specimen (which could be the whole airframe structure, a combination of components, or individual components), and accurately measuring the structural behaviour of the test specimen.

In recent times, optical techniques have been used in measurement of the shape of components for quality assurance and structural integrity assessment, and damage detection. In this chapter, an application of the optical SMS for structural testing in the Aerospace industry has been described. The optical SMS is used in tests for measuring out-of-plane displacement of a compression panel. In this case, displacement is calculated as the change in measured z coordinates from an initial or reference state. The automated calibration process has been successfully carried out at the Airbus site in Filton, Bristol, to calibrate a one-camera two-projector optical SMS for a $1.5 \times 1.5 \times 0.9 \text{ m}^3$ volume. The calibration was done in off-line mode and some of

the challenges associated with shape data acquisition on site were identified especially with regard to positioning of the artefact and avoiding specular reflection.

A low cost 2-D artefact suitable for large volumes has been described for use in the initialisation phase of the calibration process. Using acquired shape data of the 2-D artefact, its known coordinates, and the 2-D DLT was used to calculate initial estimates of the sensor external parameters. The acquired data was also used to estimate the focal length of the sensors. The ball bar artefact consisting of two CMM reference balls and length bar was used in the refinement phase of the calibration process. In order to reliably position the artefact in multiple poses in the measurement volume, the ball bar was attached to a tripod arm which was screwed on to a camera tripod stand. Thus, shape data of multiple poses of the ball bar were acquired from different positions in the measurement volume. In shape data processing, all acquired shape data for each camera-projector pair were automatically processed to select a total of 4320 control points which were then used to refine the sensor parameters in a bundle adjustment.

During the structural test, a compression panel is positioned in a test rig and load is applied until buckling failure. It is therefore necessary for the optical SMS to be properly positioned and also to make successive measurements as quickly as possible in order to accurately capture data in order to measure the change in shape. Although a one-camera two-projector SMS had been calibrated, the absence of shadowing problems in the measurement area implied that acquired data from just one camera-projector would be sufficient for measuring the displacement of the test panel. Thus, to reduce measurement time and consequently increase the number of measurements during the test, only one camera-projector pair (C_1P_2) acquired shape data. Displacement results as measured by the optical SMS have been presented, which show that the mode shapes and magnitude are comparable with results from the DIC system and from FE analysis (done prior to the test).

8.8 Tables

Table 8-1 : Parameters for C_1 , P_1 and P_2 , before and after calibration

Sensor Parameters	C_1		P_1		P_2	
	Initial estimates	Optimal estimates	Initial estimates	Optimal estimates	Initial estimates	Optimal estimates
x (mm)	591.4718	591.4718	1848.8133	1834.5249	-746.6633	-648.5009
y (mm)	133.2760	133.2760	185.2221	187.1440	172.3290	210.7506
z (mm)	4839.9202	4839.9202	4622.4015	4736.7368	4940.5135	4952.7189
ω (radians)	3.1369	3.1369	3.1289	3.1256	3.1300	3.1336
ϕ (radians)	-0.1077	-0.1077	-0.1487	-0.1461	-0.0360	-0.0505
κ (radians)	-3.1409	-3.1409	1.5606	1.5587	-1.5807	-1.5792
ξ_H (mm)	0.0	-0.1247	0.0	-0.6154	0.0	0.2584
η_H (mm)	0.0	0.3951	6.0	5.6990	6.0	6.0369
c (mm)	20.7002	20.3842	27.0646	27.1275	32.8257	32.4221
k_1	0.0	0.0	0.0	0.0	0.0	0.0
k_2	0.0	0.0	0.0	0.0	0.0	0.0
k_3	0.0	0.0	0.0	0.0	0.0	0.0

8.9 Figures

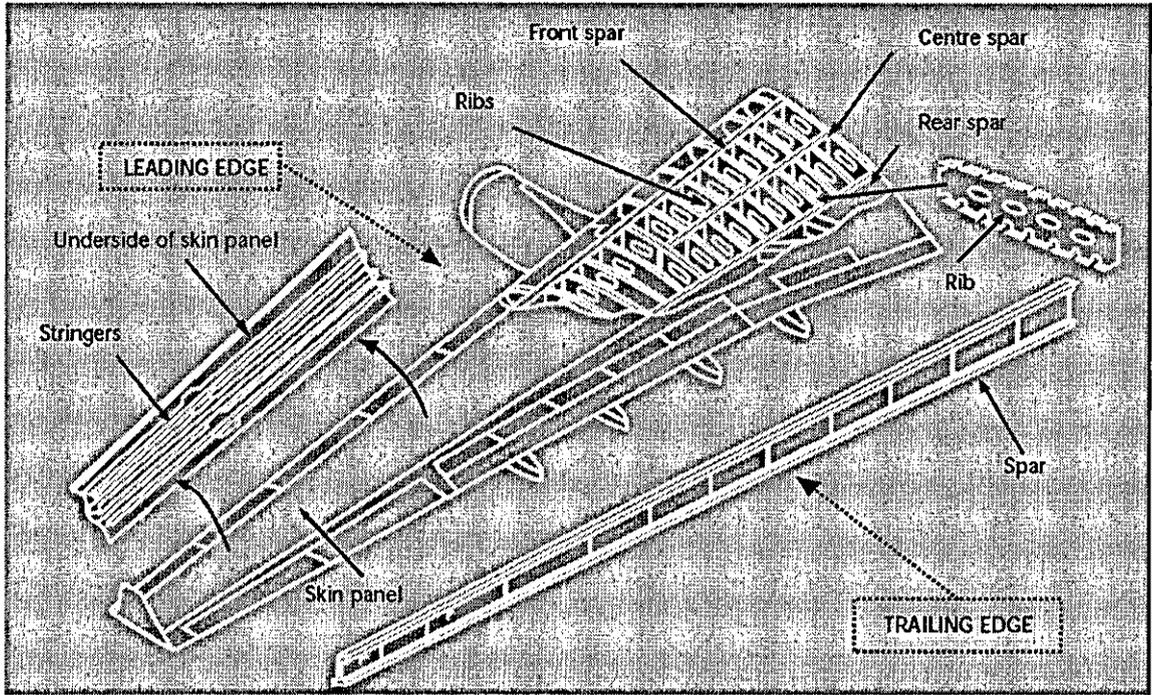


Figure 8-1: Main parts of an aircraft wing [150].



Figure 8-2: One-camera two-projector optical SMS facing a compression panel positioned in a test rig at Airbus' Bristol site.

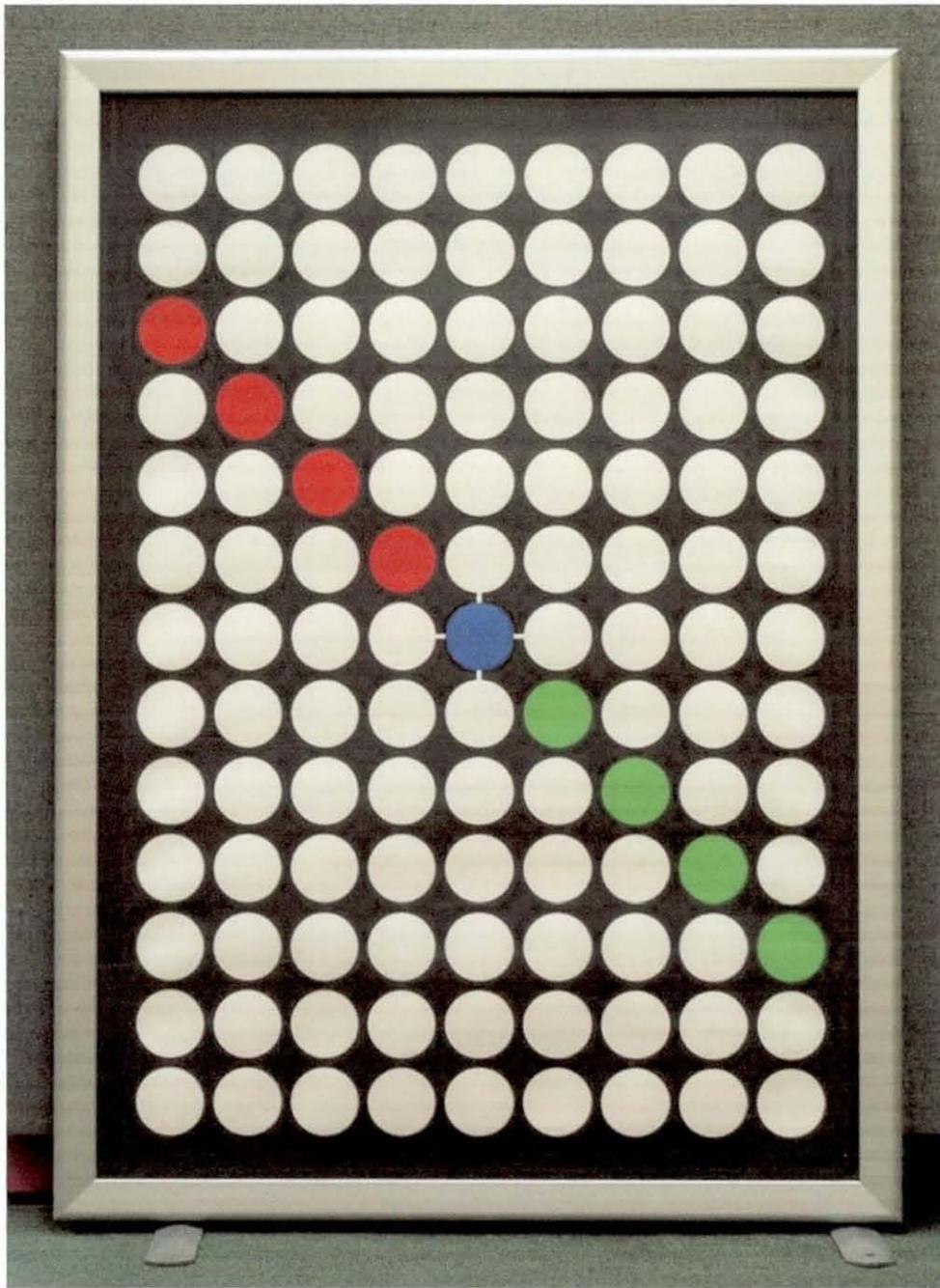


Figure 8-3: 2-D DLT artefact consisting of matt laminated A0 paper (on which is printed an array of 13×9 circles) and an A0 aluminium snap frame attached to a 12 mm thick MDF board.

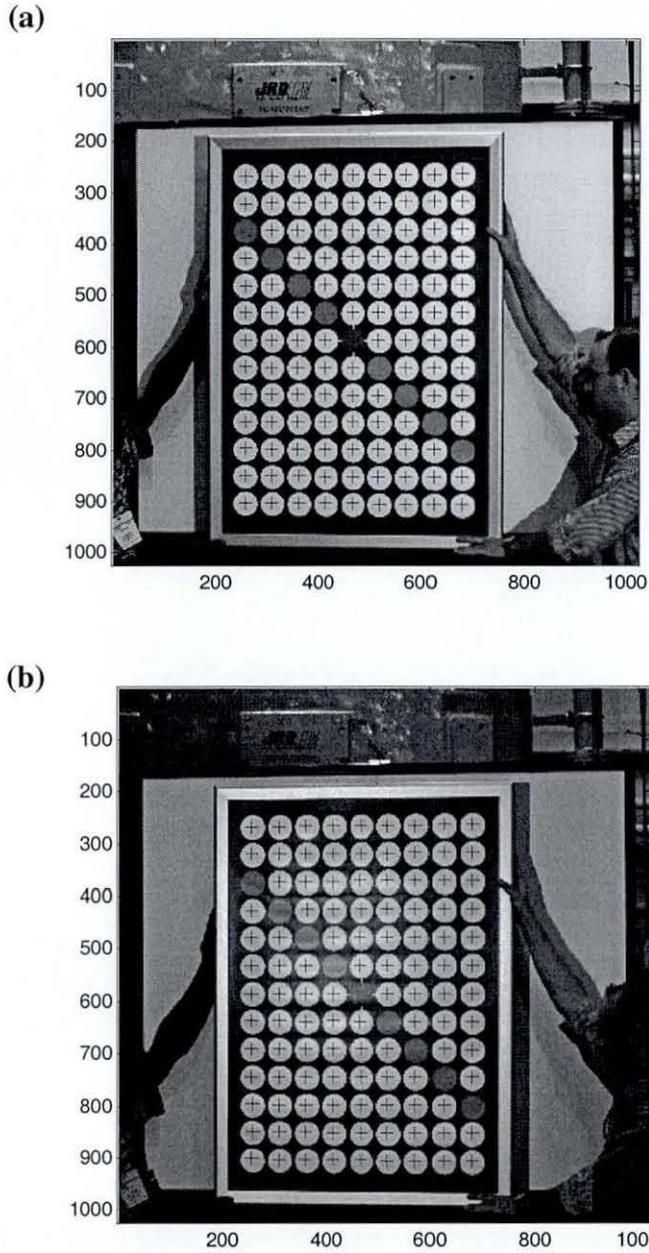


Figure 8-4: The acquired greyscale intensity image of 2-D artefact showing the circle centres detected by the 2-D Hough transform. (a) The detected circle centres for C_1P_1 . (b) The detected circle centres for C_1P_2

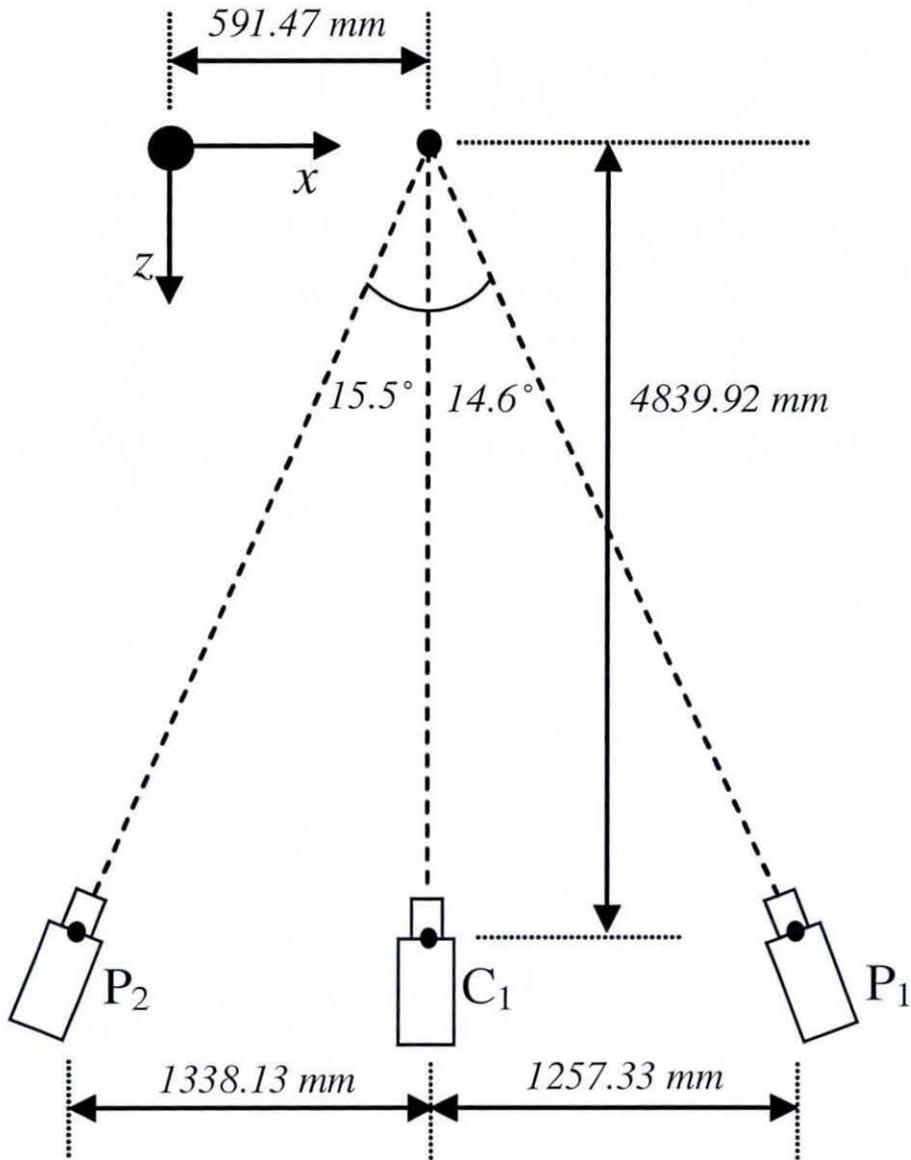


Figure 8-5: Relative distances between sensors based on external parameters obtained using the 2-D DLT method.

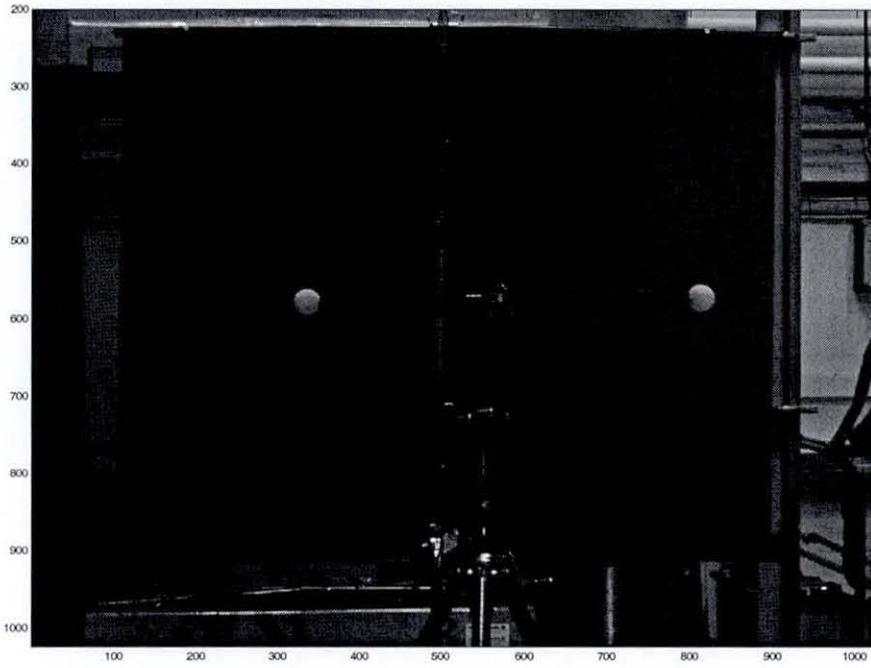


Figure 8-6: Ball bar artefact positioned in measurement volume using a camera tripod and tripod arm.

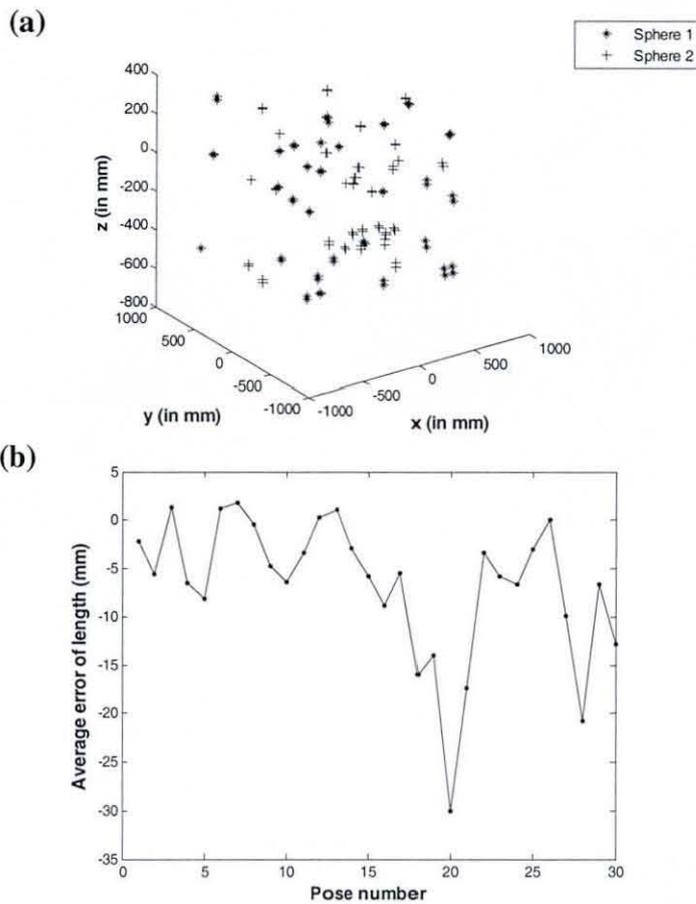


Figure 8-7: Sphere centres detected from point cloud computed using initial sensor parameters and acquired phase data of ball bar artefact. (a) All detected sphere centres of ball bar artefact in measurement volume; (b) Average of difference in the length of ball bar as measured by SMS and CMM.

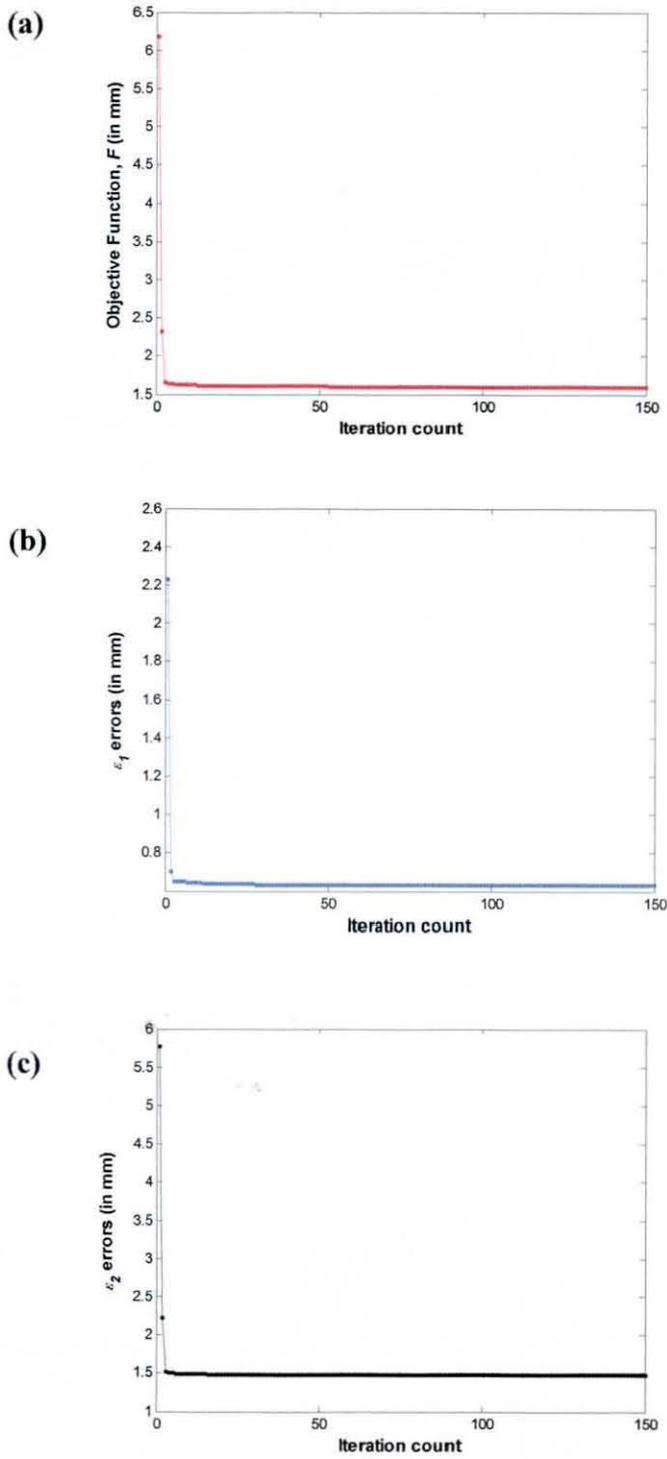


Figure 8-8: Convergence rate of rms values of objective function and contributing terms during bundle adjustment. (a) Objective function, F ; (b) ε_1 errors; (c) ε_2 errors.

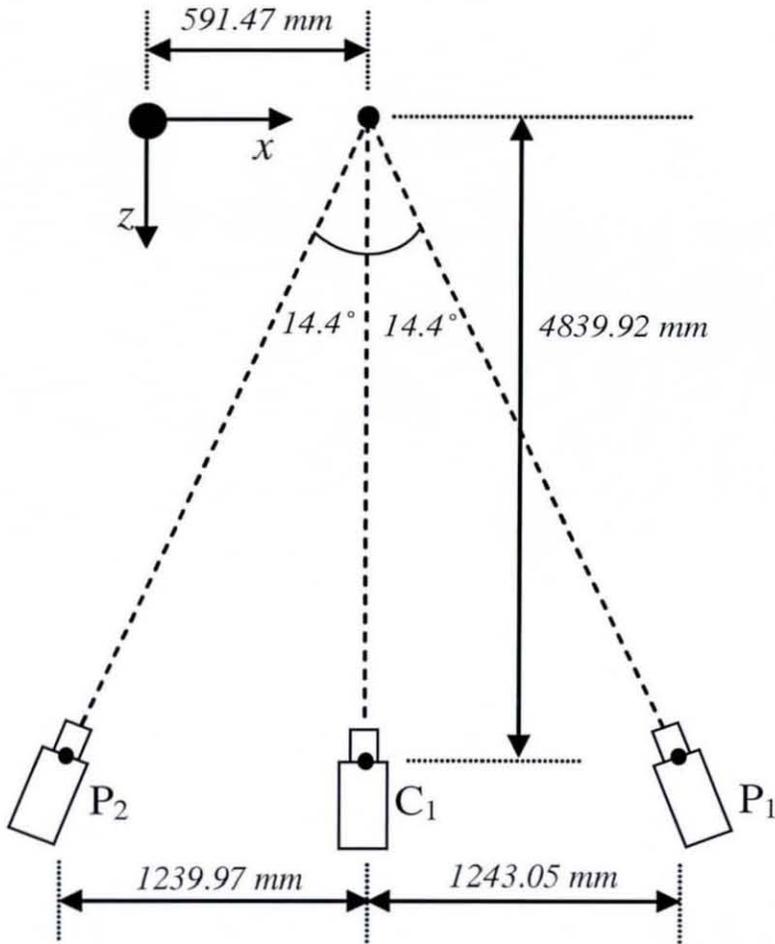


Figure 8-9: Relative distances between sensors based on external parameters obtained after bundle adjustment.

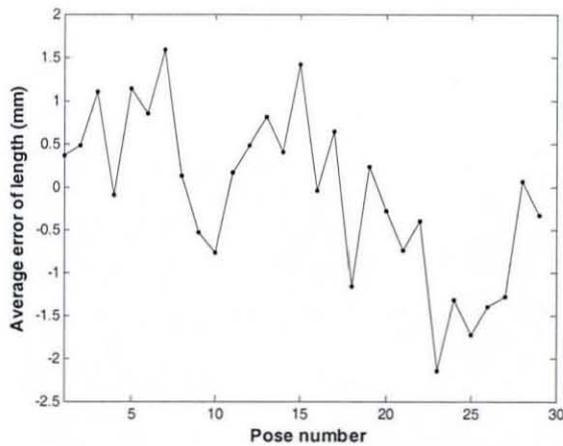


Figure 8-10: Average of difference in the length of ball bar as measured by SMS (with sphere centres detected from point cloud computed using optimised sensor parameters) and CMM.

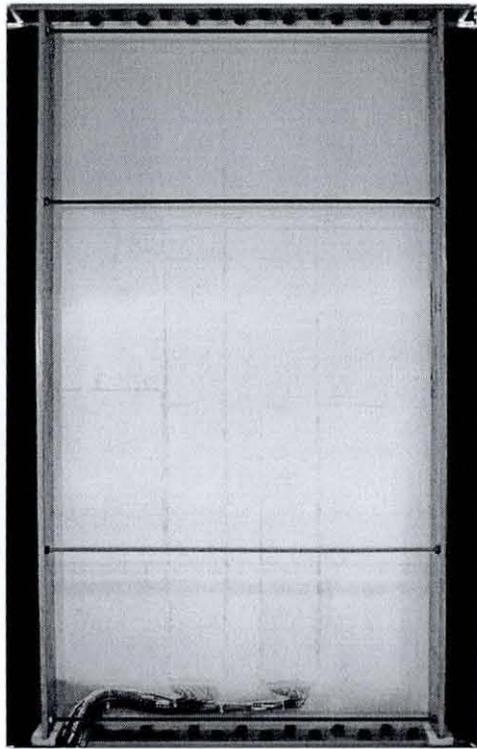


Figure 8-11: Compression panel used in structural test.

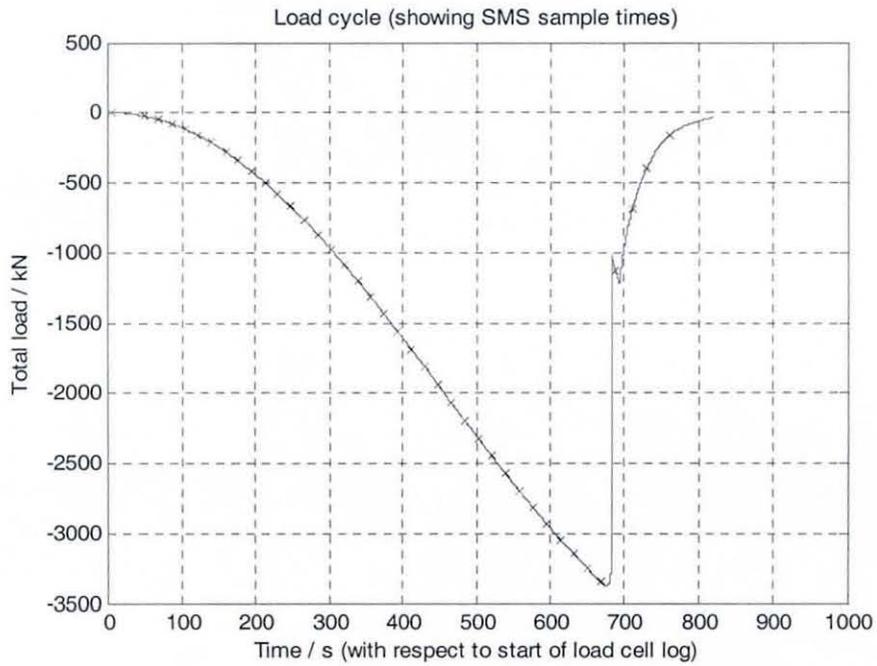


Figure 8-12: Load cycle showing SMS sample times with respect to load cell.

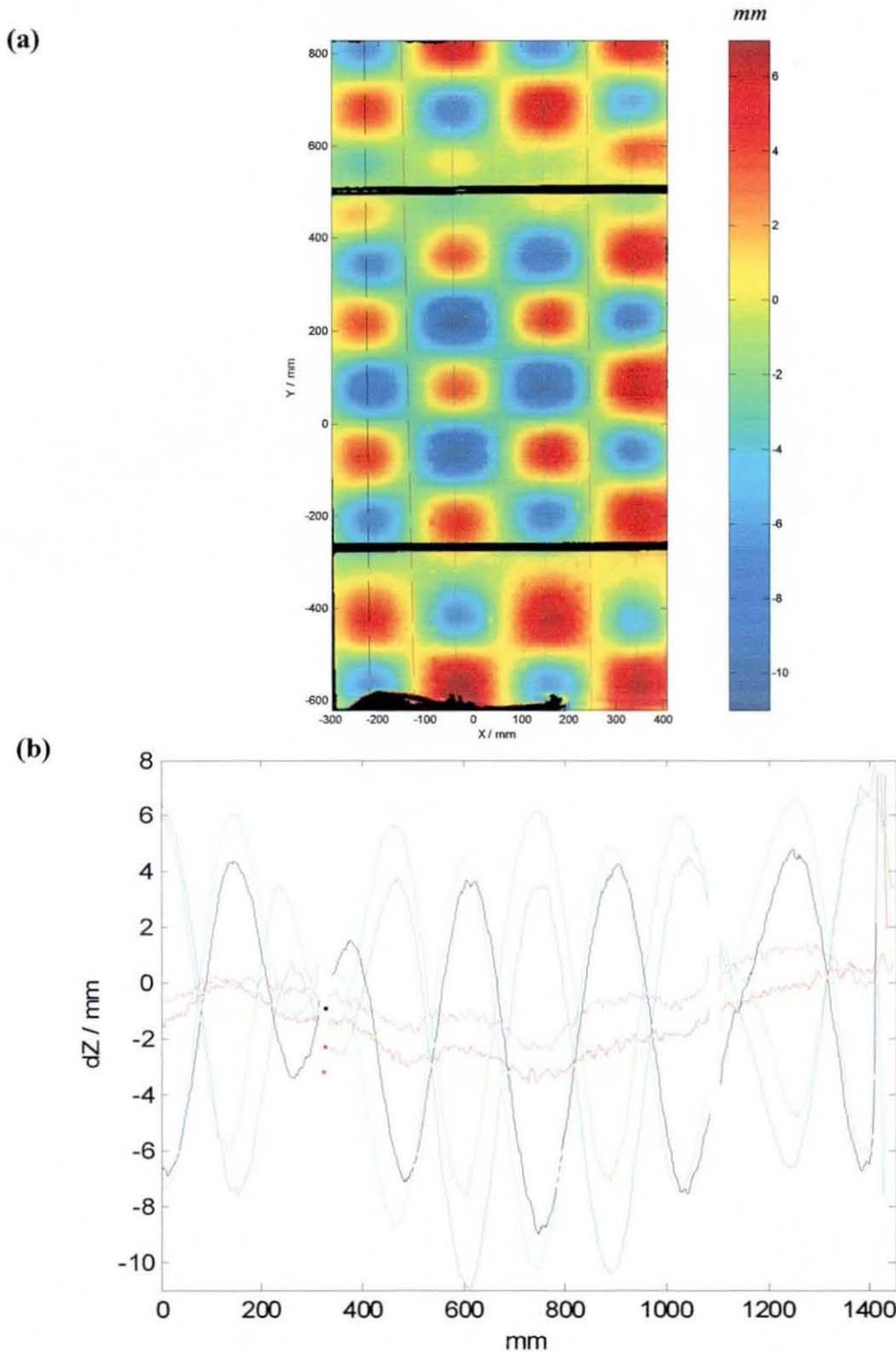


Figure 8-13: SMS measured displacement just before failure of compression panel after the application of 3344 kN, 669 s into the test. (a) Displacement map across the whole panel; (b) plot of displacement values across cross sections highlighted in (a). Cross sections are taken through the buckling mode shapes and at the position of stringers.

(c)

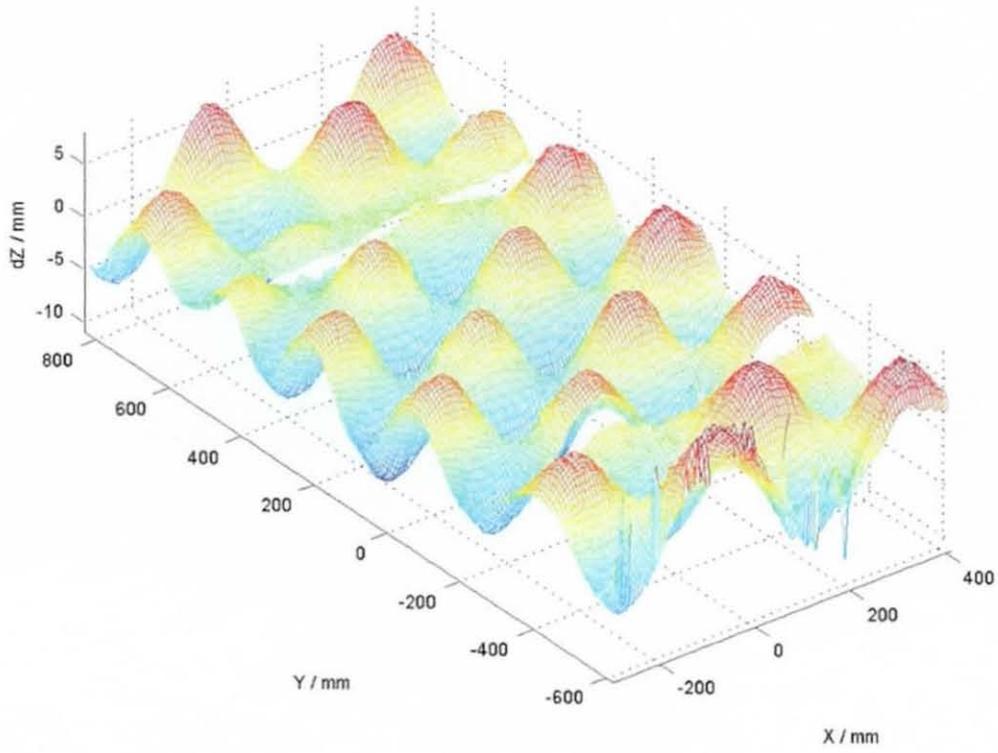


Figure 8-13 (c): Plot of displacement map across panel shown in (a).

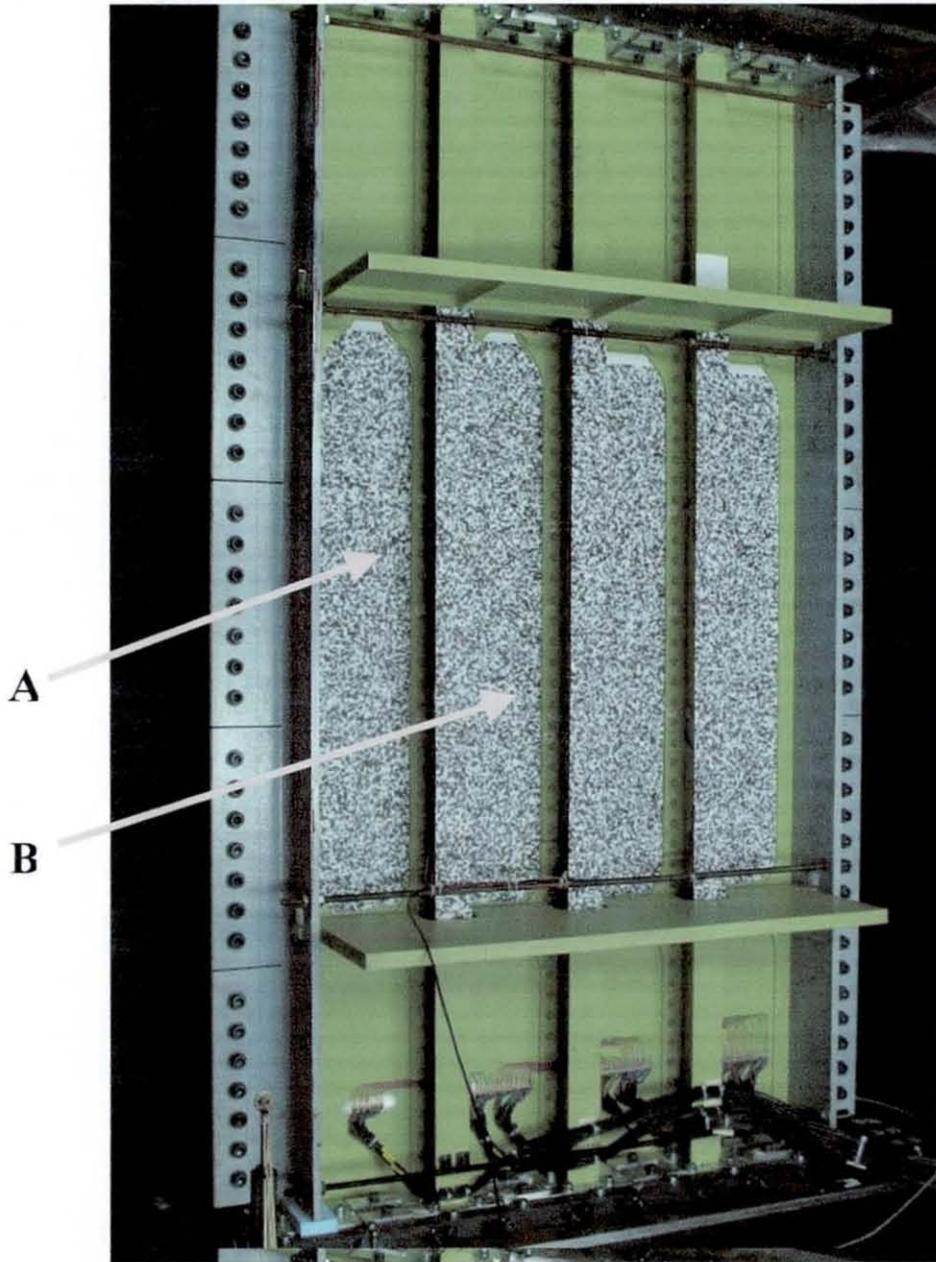


Figure 8-14: The opposite side of the compression panel showing the regions where paper with a speckle pattern was applied. Note that measurements were made on the surface of the panel AND on the surface of the three stringers. Regions 'A' and 'B' are the regions of interest for comparison of DIC measurements with SMS measurements (on opposite side of panel).

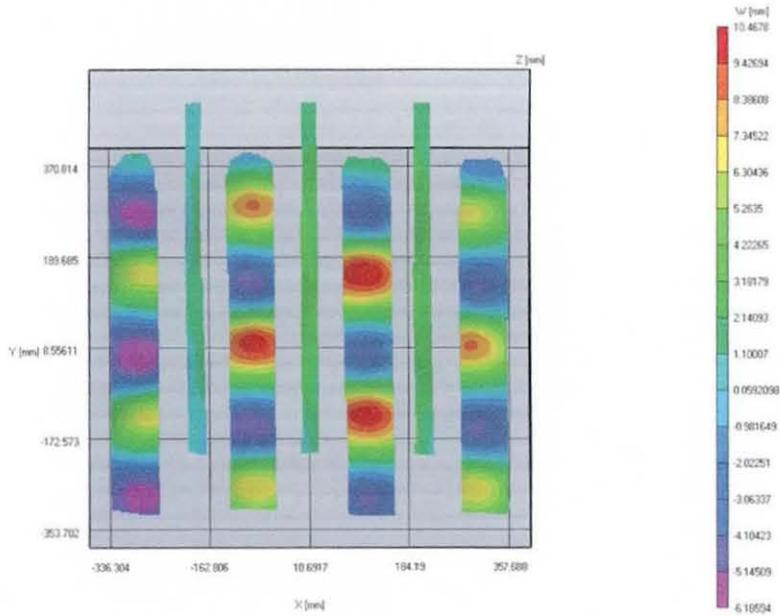


Figure 8-15: Measured displacement at failure of panel using high speed DIC at 3000 frames per second (fps).

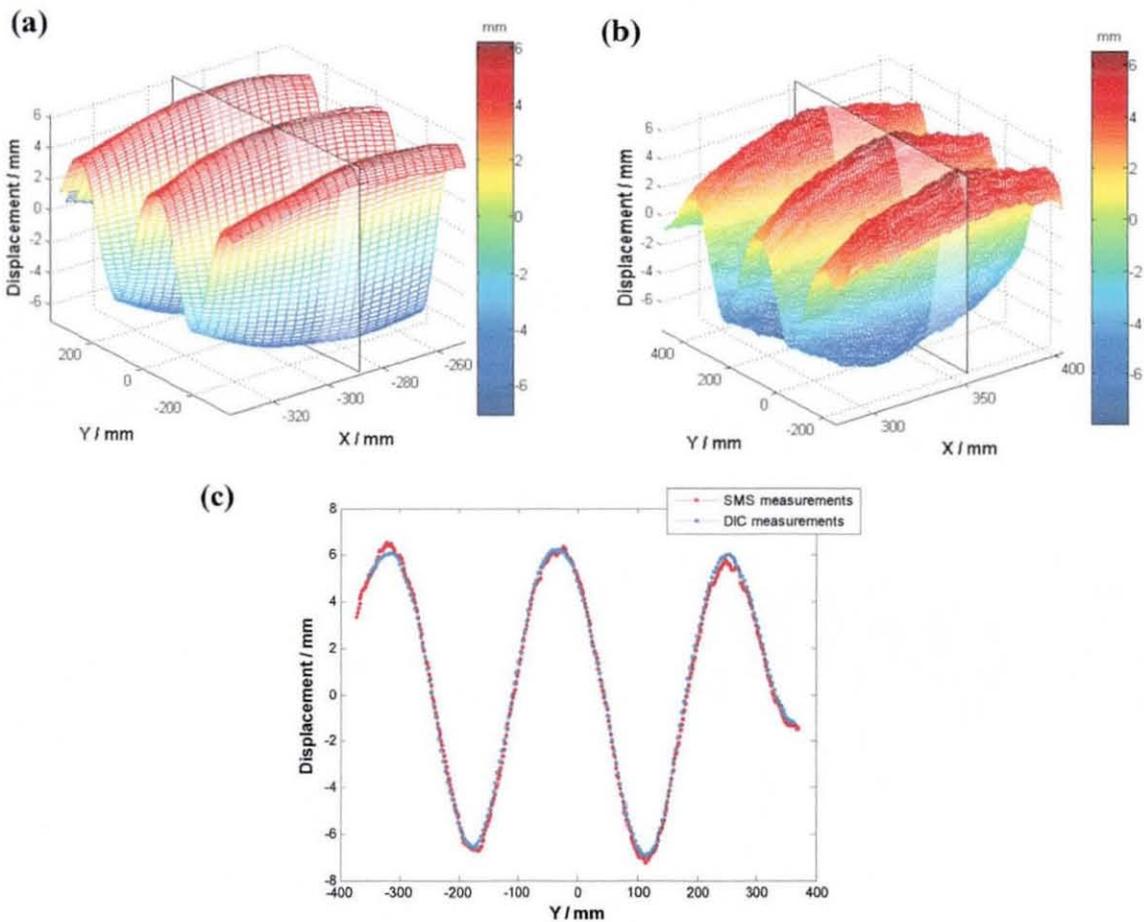


Figure 8-16: Comparison of compression panel displacement within ROI 'A' (shown in Figure 8-14) as measured by the 3-D DIC system and SMS. (a) Displacement measured by 3-D DIC; (b) Displacement measured by the SMS; (c) Comparison of displacement values of DIC and SMS along cross sectional planes in (a) and (b). Note that to allow for alignment of peaks and troughs, displacement values as measured by the DIC have been negated, with SMS Y coordinates offset by -120 mm.

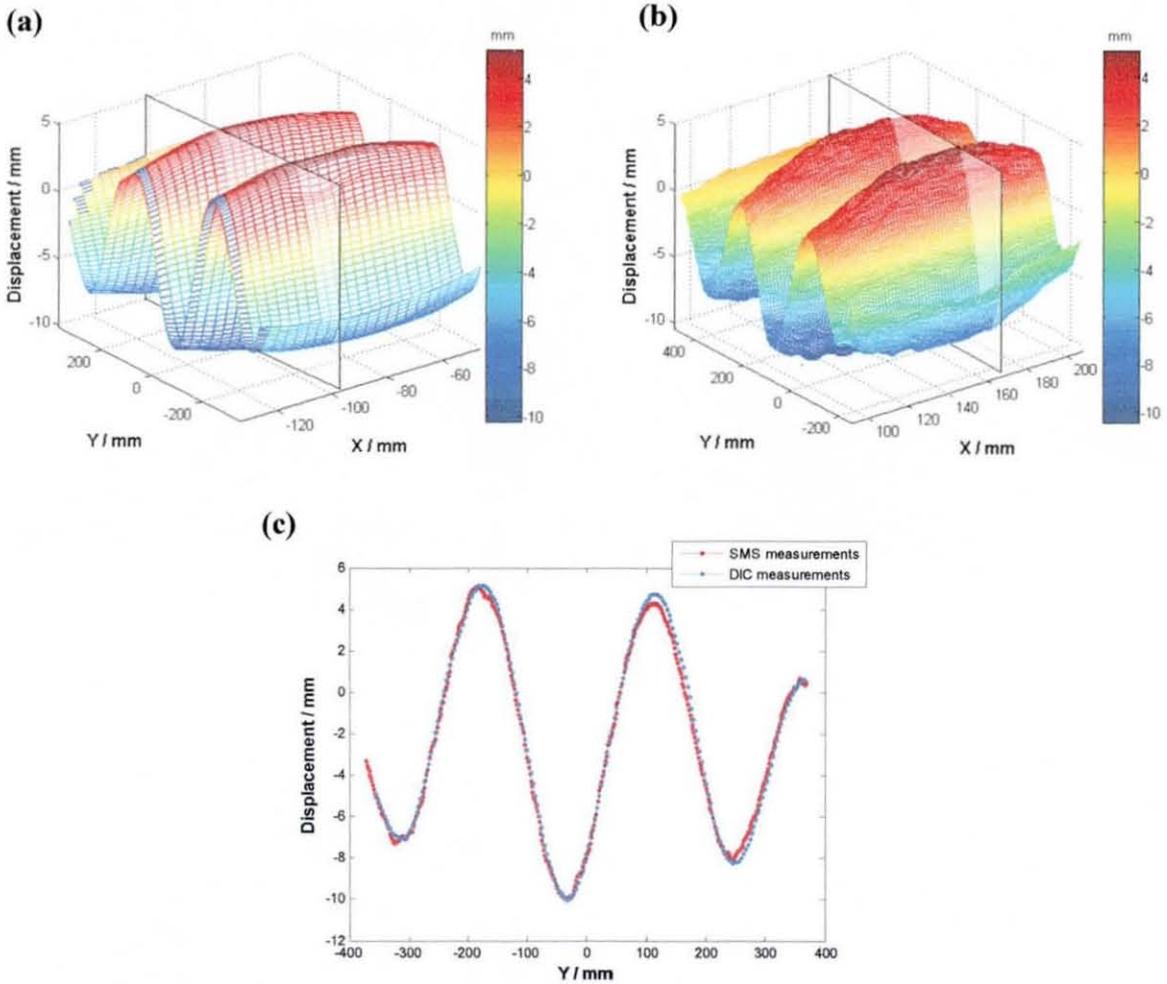


Figure 8-17: Comparison of compression panel displacement within ROI 'B' (shown in Figure 8-14) as measured by the 3-D DIC system and SMS. (a) Displacement measured by 3-D DIC; (b) Displacement measured by the SMS; (c) Comparison of displacement values of DIC and SMS along cross sectional planes in (a) and (b). Note that to allow for alignment of peaks and troughs, displacement values as measured by the DIC have been negated, with SMS Y coordinates offset by -120 mm.

**Section IV –
Conclusions**

9 Conclusions

9.1 Discussions

In recent times, a variety of optical methods have been developed for optical shape measurement systems (SMS), to measure the 3-D shape of component surfaces to high precision, for quality assurance, structural integrity assessment and damage detection. Optical SMS based on the fringe projection method have recently been commercialised as a result of its easy implementation, computer control using a spatial light modulator, and fast full-field measurement. An example is the multi-sensor optical SMS developed at Loughborough University based on the projected fringe technique and temporal phase unwrapping, which has very high scan rates (of order 10^6 s^{-1}) [4, 6-11]. However to enhance the wide acceptance in industry of sensors such as this, a number of issues need to be addressed. This thesis has therefore focussed on addressing issues relating to the calibration of multi-sensor optical SMS and the post processing of the measured 3-D data after calibration.

In practice, the calibration process of an optical SMS is a non-trivial task normally requiring significant user input. Therefore, a new calibration technique for the multi-sensor optical SMS has been investigated in order to address the following challenges [24]: the requirement for robust numerical techniques with the capability to integrate the multiple measurements to high accuracy within a single global coordinate system; the use of calibration artefacts which allow for unobstructed 360° viewing and scalable to different measurement volumes; automated detection and consistent labelling of calibration artefact features across multiple cameras; and finally, ability to make measurements for 360° coverage so as to avoid mounting the component on translation and/or rotation stages.

The new calibration method has been developed based on a photogrammetric approach where quantitative parameters are used to describe the sensor imaging geometry. Control points on the surface of a calibration artefact which have been independently measured using a mechanical coordinate measuring machine (CMM), are used to obtain the calibration parameters. The calibration parameters include sensor (i.e. for both camera and projector) internal, external and lens distortion

parameters, in addition to rigid body translation and rotation parameters that define the poses of the calibration artefact. This new calibration method thus combines the advantages of a fringe projection system that has high coordinate throughput, and photogrammetry that provides for a robust and accurate calibration.

The new calibration process can be broken into two major phases: (1) initialisation, where acquired shape data of a 2-D calibration artefact is used to initialise calibration parameters, and (2) refinement, where acquired shape data of multiple poses of a 3-D artefact are used in bundle adjustment i.e. the refinement of the calibration parameters in a non-linear optimisation. For the initialisation phase, the 2-D calibration artefact was designed to consist of an array of circles, while for the refinement phase, 3-D calibration artefacts were designed based on spherical and planar features. For spherical features, two types of artefacts were developed, namely, *multiple-sphere* and *ball bar* artefacts. Multiple-sphere artefacts which consist of multiple spheres with a radius of 23 mm, supported above a 270×270 mm² planar base were designed and manufactured using selective laser sintering. For example, one of the artefacts consists of 33 spheres with stalk heights randomly distributed within a 5 mm to 45 mm range above the top surface of the base. On the other hand, the ball bar artefact simply consists of two CMM reference balls separated by a bar of known length. Based on planar features, the *two-plane* artefact was developed, and it consists of two nominally parallel precision ground plates separated by a bar of known length.

The calibration process strongly relies on the ability to efficiently and reliably label regions within each point cloud as belonging to a known region on the surface of the calibration artefact. Automated feature detection techniques based on a novel 3-D Hough transform have been developed to efficiently and accurately detect spheres and planes from a point cloud. This approach has played a significant role in the ease, speed, and accuracy of the calibration of the multi-sensor SMS. The novel 3-D Hough transform extends the well-known strategy for detecting features such as lines and circles in 2-D images to 3-D space. The Hough accumulator implementation uses an optimised sparse 3-D matrix model based on a hash table, which provides compact data storage and efficient data access. Also, connected component labelling and weighted averaging have been introduced to enable efficient peak detection of votes in the Hough accumulator space. Application of these methods to simulated and

experimental shape data of spheres and planes have demonstrated the 3-D Hough transform's memory-saving benefits, computational efficiency and feature detection capability. The novel 3-D Hough transform developed for sphere detection is the subject of a journal paper recently accepted for publication in *Optical Engineering* [23], while the method as adapted to plane detection is the subject of a paper that is currently in preparation.

The automated calibration process has been tested on a two-camera two-projector SMS in the laboratory. To provide overall operational efficiency, the calibration software has been implemented as a combination of MATLAB[®] function files and C mex files, where XML was selected for storing the variety of calibration settings. Shape data for initialising estimates of the sensor parameters were obtained by measuring the 2-D artefact near the centre of the measurement volume. On the other hand, shape data used for refining sensor parameters were obtained by measuring a 3-D artefact in multiple poses in the measurement volume. The calibration was carried out in offline mode, whereby all respective measurements of the 2-D and 3-D artefacts were made and the acquired shape data saved to disk, with datasets processed afterwards to retrieve sensor parameters.

The calibration results using the respective 3-D artefacts have been compared, where the shape data from the ball bar artefact gave the most accurate sensor parameters, with a measurement accuracy of around one part in 5,000. It was concluded that in comparison with the other two artefacts, the ball bar artefact is the most suitable artefact for calibrating the multi-sensor optical SMS. This is as a result of its low cost, unobstructed 360° viewing by sensors, scalability to different measurement volumes, reduced shape data processing time, rapid convergence rate during bundle adjustment and better accuracy of sensor parameters. The new calibration technique using a ball bar artefact therefore enables rapid deployment of the SMS at new measurement sites, strengthens immunity to typical environmental disturbances at industrial sites, and offers an improvement in the calibration accuracy. Calibration of the SMS with the ball bar artefact is discussed in a recently accepted SPIE conference paper [24].

In many applications such as manufacturing quality control or structural testing, the measured point cloud from an optical SMS would need to be rapidly post-processed and exported to a standard file format in order to visualise measurement results on computer. The key advantage of shape data acquisition of the optical SMS is that for each camera-projector pair, a 3-D coordinate is calculated for every valid pixel, thus neighbourhood information is available after point cloud generation. An algorithm has been developed to efficiently combine the 3-D coordinates with pixel neighbourhood information in order to tessellate the measured coordinate data of the optical SMS. The method has proven to be efficient, with a point cloud having coordinates of the order of 10^6 being tessellated in less than 2 s. In addition, provision has been made for exporting the tessellated data to either an STL file format or Wavefront OBJ file format for visualisation in CAD or scientific visualisation software.

The automated calibration process using the ball bar artefact has been successfully used to calibrate a one-camera two-projector SMS at Airbus's Filton site, for measuring out-of-plane displacement during compression panel tests for the A380 and A300 test programmes. Structural testing is typically carried out in the aerospace industry for validation of numerical or mathematical models and providing information for the development and certification of aircraft. For measuring the shape of large, relatively immobile components that are commonplace in the aerospace industry, the calibration technique has the attractive feature that cameras and projectors can be added in a modular fashion to achieve 360° measurement capability. Initial estimates of the sensor parameters are obtained by measuring a low cost 2-D artefact consisting of an array of 13×9 control points, and using the 2-D direct linear transformation (DLT) method. For the refinement phase of the calibration, the ball bar artefact consists of 50.8 mm diameter spheres with sphere separation characterised by a mechanical CMM. In one of the on-site calibration results discussed in this work, a fractional error of around one part in 1,600 of the 1.5×1.5 m² field of view was achieved. The use of the SMS for structural testing at Airbus is the subject of a recently accepted SPIE conference paper [24].

9.2 Further work

The automated calibration process for a multi-sensor optical SMS has demonstrated that the optical SMS can be easily and accurately calibrated at an industrial site. However, for the ball bar artefact, the reduced coverage of the measurement volume in the respective point cloud of each camera-projector pair leads to the contribution of only a small fraction of control points to the bundle adjustment. Thus a number of measurements of the artefact at different orientations and positions would need to be made in order to achieve full coverage of the measurement volume. Therefore, a significant amount of time is required for the shape data acquisition of the ball bar artefact when calibrating volumes $> 1 \text{ m}^3$. However, this acquisition time could be reduced by mounting the ball bar artefact on a motorised device that allows for rotations about the X, Y, and Z axes. The rotation device could be mounted on a sturdy tripod (e.g. telescope tripod), such that at each position of the tripod in the measurement volume, the device would automatically control the orientation of the ball bar artefact for a set of measurements. This would ensure that only the tripod would need to be manually handled, and thus provide for a more rigid and repeatable positioning of the artefact in the measurement volume during measurements.

Recently, a carbon fibre bar has been constructed to replace the steel length bars of the ball bar artefact [24]. The carbon fibre bar has a length of 800 mm, diameter 50 mm, and wall thickness of 4 mm. The advantages of using the carbon fibre bar over length bars include [24]: lower weight which leads to reduced gravitational deflection; and significantly reduced coefficient of thermal expansion. However, the potential problem with the use of the carbon fibre bar is the dimensional changes that could occur when used under different humidity conditions. In order to demonstrate traceability of calibration, some work has been done by Dr. Coggrave of Phase Vision Ltd to characterise the sphere separation of the ball bar artefact assembly (i.e. carbon fibre bar and CMM reference balls). However, to further strengthen industry acceptance, it is suggested that the process of traceability would need to be validated in collaboration with the National Physical Laboratory (NPL). This may require development of a standard procedure for the calibration of the SMS, a procedure for characterisation of calibration artefacts and the development of other calibration and validation artefacts.

Presently, the bundle adjustment model is expressed as an objective function, F , with the minimisation of two terms, ϵ_1 , the sum of squares of errors of the mismatch offset distance of rays from each stereo pair of camera and projector (i.e. a measure of the error in the calculated control point coordinate), and ϵ_2 , the sum of squares of errors between the known control point (e.g. sphere centre or plane foot-of-normal position) and calculated control point coordinates which lie on a control surface such as a sphere or plane. However, further work could be done to introduce more terms required for computing F . One such constraint could be that the 3-D coordinates calculated at each camera pixel should be the same for all projectors. Presently, there are no constraints on the value of the 3-D coordinates which are common to all camera projector pairs since the optimisation is done on a pairwise basis. The only constraint at present is that the artefact pose is fixed across all cameras and projectors. It is possible that an additional constraint, e.g. a third term in the cost function which minimizes the distance between points computed for different projectors on the same camera may improve performance.

In the current implementation of the bundle adjustment, there is presently no restriction on the magnitude to which a calibration parameter can be changed (i.e. no allowance for specifying uncertainty of parameters). It is suggested that the current implementation be extended to allow for specifying the uncertainty of each parameter, in order to allow for more user control over the range of values of calibration parameters. It is also suggested that an investigation should be carried out into how the uncertainty of calibration parameters affect uncertainty in measured coordinates.

A CAD model contains an exact specification of an object and is considered to be a reliable template for comparison with measured data in inspection and product quality assessment. However, before the comparison can be made, we need to determine the optimal translation vector and rotation matrix required to register these two 3-D data sets which are represented in different coordinate systems. The proven method to date for point cloud registration is the iterative closest point (ICP) algorithm. However, its main drawback is the computationally intensive closest point selection process, which involves the selection of corresponding points between the measured data and model

data. A method has been proposed for the closest point selection process based on pre-calculated look up tables of vector displacements to a surface, where an adaptive grid would provide computational efficiency by ensuring that smaller cells are created in the near field of the model data and larger cells at the far field. Although this adaptive spatial decomposition algorithm for 3-D registration using the ICP has been implemented in C++, it has not been fully tested with data. It is therefore hoped that further work will include continued development of this algorithm, including rigorous testing with simulated and experimental data.

The calibration algorithms have been developed primarily in the MATLAB[®] programming language, which provides functionality for rapid prototyping of software. However, its main drawback is that it is inherently inefficient for recursive operations which are common in the calibration algorithms discussed in this work. To improve efficiency, some functionality has therefore been implemented as C mex files that interface with MATLAB[®]. However, the bundle adjustment aspect of the calibration has been completely implemented in MATLAB[®], and remains quite slow and therefore limits the number of control points that is typically used for calibration. Thus, in order to further improve operational efficiency, it is suggested that all the key bundle adjustment functions should be implemented as C mex files.

The calibration software has been designed primarily for calibrating in off-line mode. Allowing the software to run in on-line mode, where shape data would be processed immediately after acquisition would require a higher level of user interaction, with the user able to accept or reject each measured data set as part of the calibration procedure. In addition, most of the variables for user control of the calibration software are stored in an ASCII XML document, and would need to be manually edited using a text editor when carrying out an off-line calibration. It is anticipated that in future software implementations, editing the XML document will be completely hidden from the end-user and interaction with the calibration software will be through a graphical user interface (GUI). This will enable users to control input values to the calibration algorithms and visualise results in a more flexible way.

In the application of the multi-sensor optical SMS in industry, many metallic component surfaces have a shiny surface texture. This presents a unique challenge for

optical sensors since the scattered light from the component surface often contains a relatively strong specular component which would saturate the detecting camera pixel. During shape data acquisition, such specular reflection often leads to systematic errors in some regions of the recorded data set, which would translate to errors in the absolute phase maps and consequently, errors in the 3-D coordinates. Further work to reduce the occurrence and magnitude of these errors when measuring relatively shiny surfaces of as-manufactured components will extend the range of industrial manufacturing applications available to optical SMSs based on the fringe projection method. A number of methods have been suggested, including active modulation of intensity of the projected fringe patterns and use of linearly polarised light to differentiate between diffuse and specular reflection components. On the other hand, the intelligent use of multiple viewing and illumination directions in multi-sensor systems could also be investigated, whereby various illumination schemes could be tested on a CAD model of the part to be measured, in order to obtain optimal sensor positions and orientations that minimise the occurrence of specular reflection. In a similar vein, the SMS calibration software could include functionality for using simulated data that is representative of the optical and measurement setup to do a trial calibration. Thus, prior to the actual sensor calibration, the effect of sensor and artefact positions and orientations on numerical stability of the bundle adjustment can be investigated.

During structural tests in the Aerospace industry, a variety of instruments make measurements such as stress, displacement and strain. However, with such a variety of usually large measured data sets, the post-processing of the measured data is usually a time consuming process requiring personnel to work independently. It is suggested that some work be done on integrating the variety of measured data into a single database, where standard formats and procedures can be developed for data representation, data storage, and post processing. This could potentially reduce timescales for providing quantitative information relevant for aircraft development and certification.

References

References

- [1] C. Menq, H. Yau and G. Lai. "Automated Precision Measurement of Surface Profile in CAD-Directed Inspection", *IEEE Transactions on Robotics and Automation* 8(2) pp.268-278 (1992).
- [2] Z. Li, M. Graham and D. Robinson. "Point Cloud to Surface Best-Fitting", *Technical Report 2002GRC175*, Technical Information Series GE Global Research, (2002).
- [3] F. Chen, G. M. Brown and M. Song. "Overview of three-dimensional shape measurement using optical methods", *Optical Engineering* 39(1) pp.10-22 (2000).
- [4] C. R. Coggrave. "Temporal Phase Unwrapping: Development and Application of Real - Time Systems for Surface Profile and Surface Displacement Measurement", *PhD Thesis, Loughborough University* (2001).
- [5] J. M. Huntley. "Optical shape measurement technology: past, present, and future", *Proc SPIE 4076, Glasgow, UK, 22-24 May*, pp.162-173 (2000).
- [6] H. O. Saldner and J. M. Huntley. "Temporal phase unwrapping: application to surface profiling of discontinuous objects", *Appl. Opt.* 36(13) pp.2770-2775 (1997).
- [7] H. O. Saldner and J. M. Huntley. "Profilometry by temporal phase unwrapping and spatial light modulator-based fringe projector", *Opt. Eng.* 36(2) pp.610-615 (1997).
- [8] J. M. Huntley and H. O. Saldner. "Shape measurement by temporal phase unwrapping: comparison of unwrapping algorithms", *Meas. Sci. Technol.* 8(9) pp.986-992 (1997).
- [9] J. M. Huntley and H. O. Saldner. "Error reduction methods for shape measurement by temporal phase unwrapping", *J. Opt. Soc. Am. A* 14(12) pp.3188-3196 (1997).
- [10] C. R. Coggrave and J. M. Huntley. "High speed surface profilometer based on spatial light modulators", *Opt. Eng.* 38 pp.1573-1581 (1999).
- [11] C. R. Coggrave and J. M. Huntley. "Optimisation of a shape-measurement system based on spatial light modulators", *Opt. Eng.* 39 pp.91-98 (2000).
- [12] M. Jones. , "Calculation of combustion chamber volume using optical technology", *Mechanical Engineering 02MMC500 Individual Final Year Report, Loughborough University* (2003).
- [13] W. Osten. "Application of optical shape measurement for the nondestructive evaluation of complex objects", *Opt. Eng.* 39(1) pp.232-243 (2000).
- [14] T. A. Clarke, T. J. Ellis and S. Robson. "High accuracy 3-D measurement using multiple camera views", *IEE Colloquium Digest No. 1994/054*, (1994).

- [15] W. Schreiber and G. Notni. "Theory and arrangements of self-calibrating whole-body three-dimensional measurement systems using fringe projection technique", *Opt. Eng.* **39**(1) pp.159-169 (2000).
- [16] G. M. S. De Silva. *Basic Metrology for ISO 9000 certification*, Butterworth-Heinemann, Oxford, UK (2002).
- [17] P. H. Sydenham, N. H. Hancock and R. Thorn. *Introduction to measurement science and engineering*, Wiley, Chichester, UK (1989).
- [18] Bureau International des Poids et Mesures (BIPM). Bureau International des Poids et Mesures (BIPM), http://www.bipm.fr/en/si/base_units/, [Accessed – June 2005].
- [19] H. Kunzmann, J. Ni and F. Waldele. "Accuracy Enhancement", in *Coordinate measuring machines and systems*, J. A. Bosch, Ed., Marcel Dekker Inc, USA (1995).
- [20] J. Ni and F. Waldele. "Coordinate measuring machines", in *Coordinate measuring machines and systems*, J. A. Bosch, Ed., Marcel Dekker Inc, USA (1995).
- [21] T. A. Clarke. "Non-contact measurement provides six of the best", *Quality Today*, pp.s46-s48 (July 1998).
- [22] W. Winchell. *Inspection and Measurement in Manufacturing – keys to process planning and improvement*, Society of Manufacturing Engineers, Dearborn, MI, USA (1996).
- [23] O. O. Ogundana, C. R. Coggrave, R. L. Burguete and J. M. Huntley. "Fast Hough Transform for automated detection of spheres in three-dimensional point clouds", *Opt. Eng.* **46**, 051002 (2007).
- [24] J. M. Huntley, T. Ogundana, R. L. Burguete and C. R. Coggrave. "Large-scale full-field metrology using projected fringes: some challenges and solutions", *Proc. SPIE* 6616, 66162C (2007).
- [25] Q. Hu, P. S. Huang, Q. Fu and F. Chiang. "Calibration of a three-dimensional shape measurement system", *Opt. Eng.* **42**(2) pp.487-493 (2003).
- [26] J. Salvi, X. Armangue and J. Batlle. "A comparative review of camera calibrating methods with accuracy evaluation", *Pattern Recognition* **35** pp.1617-1635 (2002).
- [27] E. M. Mikhail, J. S. Bethel and J. C. McGlone. *Introduction to Modern Photogrammetry*, John Wiley & Sons, USA (2001).
- [28] S. J. Marshall, D. N. Whiteford and R. C. Rixon. "Assessing the performance of 3D whole body imaging systems", *Proc. 6th Numerisation 3D/scanning 2001 congress, Paris, France*, (2001).
- [29] K. Kraus. *Photogrammetrie Vol 1 and 2*, Ferd, Dummler Verlag, Bonn (1996).

- [30] Cooper, M. A. R. and S. Robson. "Theory of close range photogrammetry", in *Close range photogrammetry and machine vision*, K. B. Atkinson, Ed., Whittles Publishing, Scotland pp. 9-51 (1996).
- [31] C. S. Fraser. "Digital camera self-calibration", *ISPRS Journal of Photogrammetry & Remote Sensing* 52 pp.149-159 (1997).
- [32] B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon. "Bundle Adjustment – A Modern Synthesis", In *Proc. of the International Workshop on Vision Algorithms: Theory and Practice*, pp.298-372 (2000).
- [33] R. Hartley and A. Zisserman. *Multiple View Geometry in computer vision*, Cambridge university press, The Edinburgh Building, Cambridge, CB2 2RU, UK (2003).
- [34] J. A. Beraldin. "Integration of laser scanning and close-range photogrammetry - The last decade and beyond", *XXth International Society for photogrammetry and remote sensing (ISPRS) congress. Commission VII, Istanbul, Turkey, July 12-23*, pp.972-983 (2004).
- [35] T. A. Clarke and J. G. Fryer. "The development of camera calibration calibration methods and models", *Photogrammetric Record* 16(91) pp.51-66 (1998).
- [36] C. S. Fraser, M. R. Shortis and G. Ganci. "Multi-sensor system self-calibration", *Philadelphia, USA*, October 25-26, (1995).
- [37] C. S. Fraser. "On the use of non-metric cameras in analytical non-metric photogrammetry", *International Archives of Photogrammetry and Remote Sensing* 24(5) pp.156-166 (1982).
- [38] J. G. Fryer. "Camera calibration", in *Close range photogrammetry and machine vision*, K. B. Atkinson, Ed., Whittles Publishing, Scotland pp. 156-179 (1996).
- [39] D. Brown. "A strategy for multi-camera on-the-job self-calibration", *Institut Fur Photogrammetrie Stuttgart*
- [40] Geodetic Services Inc. "The Basics of Photogrammetry", Geodetic Services Inc, 1511 Riverview Drive, Melbourne, FL, USA 32901, http://www.geodetic.com/Downloadfiles/Basics_of_Photogrammetry.pdf, [Accessed-2005].
- [41] R. Hartley and J. L. Mundy. "The relationship between photogrammetry and computer vision", *Integrating photogrammetric techniques with scene analysis and machine vision*, pp.92-105 (1993).
- [42] M. Agrawal and L. S. Davis. "Camera calibration using spheres: A semi-definite programming approach", *Proceedings IEEE International Conference on Computer Vision, ICCV, Nice, France*, October 13-16, (2003).
- [43] J. I. Gonzalez, Hernandez Sosa, J. D., Dominguez Brito, A. C. and Naranjo Cabrera, A. M. "Comparative Analysis of Calibration Methods for a Static Camera", *Las Palmas de Gran Canaria, Spain*, February 7-11, (2005).

- [44] O. Faugeras. *Three-dimensional computer vision: a geometric viewpoint*, MIT Press, (1993).
- [45] R. Y. Tsai. "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Of-the-Shelf TV Cameras and Lenses", *IEEE Journal of Robotics and Automation* RA-3(4) pp.323-344 (1987).
- [46] M. Ahmed, E. Hemayed and A. Farag. "Neurocalibration: a neural network that can tell camera calibration parameters", September 1999, pp.463-468 (1999).
- [47] J. Heikkila and O. Silven. "A four-step camera calibration procedure with implicit image correction", *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp.1106-1112 (1997).
- [48] J. Batista, H. Araujo and T. A. de Almeida. "Iterative multi-step explicit camera calibration", pp.709-714 (1998).
- [49] Z. Zhang. "A flexible technique for camera calibration", *IEEE Trans. PAMI* 22(11) pp.1330-1334 (2000).
- [50] H. Zollner and R. Sablatnig. "Comparison of methods for geometric camera calibration using planar calibration targets", *Digital Imaging in Media and Education*, pp.237-244 (2004).
- [51] Y. I. Abdel-Aziz and H. M. Karara. "Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close Range Photogrammetry", *Proceedings of the ASP/UI Symposium on Close Range Photogrammetry*, 1-18, January 1971, (1971).
- [52] E. L. Hall, J. B. K. Tio, C. A. McPherson and F. A. Sadjadi. "Measuring curved surfaces for robot vision", *Comput. J.* 15 pp.42-54 (1982).
- [53] O. D. Faugeras and G. Toscani. "The calibration problem for stereo", *CVPR, Miami Beach, Florida*, 15-20 June, 1986, pp.15-20 (1986).
- [54] G. Toscani. "Systemes de calibration et perception du mouvement en vision artificielle", *Universite Paris Sud, Ph. D. Thesis* (1987).
- [55] J. Weng, P. Cohen and M. Herniou. "Camera calibration with distortion models and accuracy evaluation", *IEEE Trans. PAMI* 14 pp.965-980 (1992).
- [56] J. Heikkila. "Geometric Camera Calibration Using Circular Control Points", *IEEE Trans. PAMI* 22(10) pp.1066-1077 (2000).
- [57] F. Chen. "Application of Least-Squares Technique to Geometric Camera Calibration and Photogrammetric Flow Visualization", *ISA 43rd International Instrumentation Symposium, Orlando, Florida*, May 4-8, (1997).
- [58] S. Ganapathy. "Camera location determination problem", *Technical Report 11358-841102-20-TM*, AT&T Bell Laboratories, (Nov. 1984).

- [59] H. Guo, H. He, Y. Yu and M. Chen. "Least-squares calibration method for fringe projection profilometry", *Opt. Eng.* **44**(3) pp.0033603 (2005).
- [60] R. Sitnik, M. Kujawska and J. Woznicki. "Digital fringe projection system for large-volume 360-degree shape measurement", *Opt. Eng.* **41**(2) pp.443-449 (2002).
- [61] R. Legarda-Saenz, T. Bothe and W. Juptner. "Accurate procedure for the calibration of a structured light system", *Opt. Eng.* **43**(2) pp.494-471 (2004).
- [62] S. Zhang and P. S. Huang. "Novel method for structured light system calibration", *Opt. Eng.* **45**(8) pp.083601-1-083601-8 (2006).
- [63] W. Zhou and X. Y. Su. "A direct mapping algorithm for phase-measuring profilometry", *J. Mod. Opt.* **41** pp.89-94 (1994).
- [64] A. Asundi and Z. Wensen. "Unified calibration technique and its applications in optical triangular profilometry", *Appl. Opt.* **38**(16) pp.3556-3561 (1999).
- [65] H. Liu, B. Bard, G. Lu and S. Wu. "Absolute measurement of surface profiles in phase-shifting projected fringe profilometry", *Optical Manufacturing and Testing III*, pp.283-289 (1999).
- [66] Y. Y. Hung, L. Lin, H. M. Shang and B. G. Park. "Practical three-dimensional computer vision techniques for full-field surface measurement", *Opt. Eng.* **39**(1) pp.143-149 (2000).
- [67] I. Niini. "On the calibration of mapvision 4D system", *Proceedings of the ISPRS Commission III* (Sept. 2002).
- [68] H. Zhang, G. Zhang and K. -Y K. Wong. "Camera calibration with spheres: linear approaches", *International Conference on Image Processing (ICIP05) volume II, Genova, September*, pp.1150-1153 (2005).
- [69] H. Zhang, G. Zhang and K. -Y K. Wong. "Camera calibration from images of spheres", *IEEE Trans. PAMI* **29**(3) pp.499-503 (2007).
- [70] F. Pedersini, A. Sarti and S. Tubaro. "Multi-Camera systems", *IEEE Signal Processing Magazine*, (May 1999).
- [71] F. Pedersini, A. Sarti and S. Tubaro. "Accurate and simple geometric calibration of multi-camera systems", *Signal Processing* **77**(3) pp.309-304 (1999).
- [72] M. Machacek, M. Sauter and T. Rosgen. "Two-step calibration of a stereo camera system for measurements in large volumes", *Measurement Science and Technology* **14**(9) pp.1631-1639 (2003).
- [73] N. A. Borghese and P. Cerveri. "Calibrating a video camera pair with a rigid bar", *Pattern Recognition* **33**(1) pp.81-95 (2000).
- [74] H. Bakstein. "A Complete DLT-based Camera Calibration with a Virtual 3D Calibration Object", *Faculty of Mathematics and Physics, Charles University, Prague* (1999).

- [75] T. Liu, Cattafesta, L. N., III and R. H. Radeztky. "Photogrammetry applied to wind tunnel testing", *AIAA Journal* 38(6) pp.964-971 (2000).
- [76] L. Chen and C. Liao. "Calibration of 3D surface profilometry using digital fringe projection", *Meas. Sci. Technol.* 15(2005) pp.1554-1566 (2005).
- [77] H. Schonfeld, G. Hausler and S. Karbacher. "Reverse engineering using optical 3D sensors", *Proceedings SPIE Sensors, Sensor Systems and Sensor Data Processing*, 1998, pp.115-125 (1998).
- [78] P. Kuhmstedt, G. Notni, W. Schreiber and J. Gerber. "Full Hemisphere automatic optical 3D measurement system", *Proc. SPIE Sensors, Sensor Systems and Sensor Data Processing*, pp.261-265 (1997).
- [79] R. J. Valkenburg and A. M. McIvor. "Accurate 3D measurement using a Structured Light System", *Image and Vision Computing* 16 pp.99-110 (1997).
- [80] A. T. Deacon, A. G. Anthony, S. N. Bhatia and J. Muller. "Evaluation of a CCD-based facial measurement system", *Med. Inform.* 16(2) pp.213-228 (1991).
- [81] van den Heuvel, F. A. "Automated 3-D measurement with the DCS200 digital camera", *Proc. SPIE Optical 3D Measurement Techniques II* 2252 pp.63-71 (1994).
- [82] A. Godhwani, G. Bhatia and M. W. Vannier. "Calibration of a multisensor structured light range scanner", *Opt. Eng.* 33(4) pp.1359-1367 (1994).
- [83] M. A. Penna. "Camera calibration: a quick and easy way to determine the scale factor", *IEEE Trans. PAMI* 13(12) pp.1240-1245 (1991).
- [84] G. Xu, Y. Inoue and H. Teramoto. "Determining camera intrinsic and extrinsic parameters from multiple images of multiple balls", *Workshop on science of computer vision* (<http://www.suri.it.okayama-u.ac.jp/sov2002/xu.pdf>), 19-20 September, (2002).
- [85] J. A. Beraldin, S. F. El-Hakim and F. Blais. "Performance evaluation of three active vision systems built at the national research council of Canada", *Proc. Optical 3-D Measurement Techniques III, Vienna, Austria*, Oct. 2-4, 1995, pp.352-361 (1995).
- [86] J. M. Huntley. "Bundle adjustment method based on object-side ray tracing", *Technical Report 1*, Phase Vision Ltd, Loughborough, UK (2005).
- [87] J. J. McKeown, D. Meegan and D. Sprevak. *An Introduction to unconstrained optimisation - A computer illustrated text*, Hilger, Bristol, New York (1990).
- [88] H. Hopkinson and P. M. Dickens. "Emerging Rapid Manufacturing Processes", in *Rapid manufacturing: an industrial revolution for the digital age*, H. Hopkinson, R. J. M. Hague, and P. M. Dickens, Ed., John Wiley & Sons Ltd., West Sussex, England pp. 55-80 (2006).
- [89] A. S. Morris. *Measurement and Instrumentation Principles*, Butterworth-Heinemann, Oxford (2001).

- [90] V. F. Leavers. "SURVEY Which Hough Transform?", *CVGIP: Image Understanding* 58(2) pp.250-264 (September 1993).
- [91] Hough, P. V. C. "Method and means for recognising complex patterns", *US Patent Number* 3069654
- [92] Richard O. Duda and Peter E. Hart. "Use of the Hough Transformation To Detect Lines and Curves in Pictures", *Communications of the ACM* 15(1) pp.11-15 (January 1972).
- [93] C. Kimme, D. Ballard and J. Sklansky. "Finding Circles by an Array of Accumulators", *Communications of the ACM* 18(2) pp.120-122 (February 1975).
- [94] R. M. Haralick and L. G. Shapiro. *Computer and Robot Vision - Volume 1*, Addison Wesley Publishing Company, Inc., USA (1992).
- [95] S. Warr, Jacques, G. T. H. and J. M. Huntley. "Tracking the translational and rotational motion of granular particles: use of high speed photography and image processing", *Powder Technology* 81 pp.41-56 (1994).
- [96] H. K. Yuen, J. Princen, J. Illingworth and J. Kittler. "Comparative Study of Hough Transform methods for circle finding", *Image and Vision Computing* 8(1) pp.71-77 (1990).
- [97] M. van der Glas, F. M. Vos, C. P. Botha and A. M. Vossepoel. "Determination of position and radius of ball joints", *Proc. SPIE* 4684 pp.1571-1577 (2002).
- [98] M. van der Glas, F. M. Vos and A. M. Vossepoel. "Automatic determination of the center of rotation of the glenohumeral joint by a modified Hough transform", *"Proc. ASCI 2001, 7th Annual Conf. of the Advanced School for Computing and Imaging"*, Heijen, NL, May 30-June 1, pp.92-98 (2001).
- [99] T. Rabbani and van den Heuvel, F. "Efficient Hough Transform for automatic detection of cylinders in point clouds", *ISPRS WG III/3, III/4, V/3 Workshop "Laser scanning 2005"*, Enschede, the Netherlands, September 12-14, 2005, pp.60-65 (2005).
- [100] Jack Dongarra, Andrew Lumsdaine, Xomhui Niu, Roldan Pozo and Karin Remington. "Sparse Matrix Libraries in C++ for High Performance Architectures", *Second Object Oriented Numerics Conference*, pp.214-218 (1994).
- [101] Yosef Saad. "SPARSKIT: A basic tool kit for sparse matrix computations", *Technical Report* Computer Science Department, University of Minnesota, Minneapolis, MN 55455 (June 1994).
- [102] M. A. Weiss. *Data Structures and Algorithm Analysis*, The Benjamin/Cummings Publishing Company, Inc., California, USA (1994).
- [103] P. E. Black. "Dictionary of Algorithms and Data Structures", US National Institute of Standards - Information Technology Laboratory, <http://www.nist.gov/dads/>, [Accessed - 27 February 2006].

- [104] Wikipedia. "Hash Table", http://en.wikipedia.org/wiki/Hash_table, [Accessed – 7 September 2006].
- [105] M. T. Goodrich and R. Tamassia. *Data structures and algorithms in Java*, Wiley, Chichester, UK (2006).
- [106] R. Sedgewick. *Algorithms*, Addison-Wesley Publishing Inc, (1983).
- [107] B. W. Kernighan and D. M. Ritchie. *The C Programming Language*, Prentice Hall International, New Jersey, USA (1988).
- [108] R. B. Fisher and D. K. Naidu. "A Comparison of Algorithms for Subpixel Peak Detection", *Proceedings 1991 British Machine Vision Association Conference*, pp.217-225 (1991).
- [109] Y. Furukawa and Y. Shinagawa. "Accurate and robust line segment extraction by analyzing distribution around peaks in Hough space", *Computer Vision and Image Understanding* 92 pp.1-25 (2003).
- [110] J. O'Rourke and K. R. Sloan Jr. "Dynamic quantization: two adaptive data structures for multidimensional spaces", *IEEE Trans. PAMI* 6(3) pp.266-280 (1984).
- [111] L. O'Gorman and A. C. Sanderson. "The converging squares algorithm: an efficient method for locating peaks in multidimensions", *IEEE Trans. PAMI* 6(3) pp.280-288 (1984).
- [112] J. M. Jolion, P. Meer and S. Bataouche. "Robust clustering with applications in computer vision", *IEEE Trans. PAMI* 13(8) pp.791-802 (1991).
- [113] Y. Cheng. "Mean shift, mode seeking, and clustering", *IEEE Trans. PAMI* 17(8) pp.790-799 (1995).
- [114] R. Jain, R. Kasturi and B. G. Schunk. *Machine Vision*, McGraw-Hill, New York, London (1995).
- [115] K. Suzuki, I. Horiba and N. Sugie. "Linear-time connected-component labelling based on sequential local operations", *Computer Vision and Image Understanding* 89 pp.1-23 (2003).
- [116] M. Sjodahl and B. F. Oreb. "Stitching interferometric measurement data for inspection of large optical components", *Opt. Eng.* 41(2) pp.403 – 408 (2002).
- [117] D. A. Simon. "Fast and Accurate Shape-Based Registration", *PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA* (1996).
- [118] O. D. Faugeras and M. Hebert. "The representation, recognition, and locating of 3-d objects", *The International Journal of Robotics Research* 5(3) pp.27–52 (1986).
- [119] Horn, B. K. P. "Closed-form solution of absolute orientation using unit quaternions", *Journal of the Optical Society of America A* 4(4) pp.629–642 (1987).

- [120] K. S. Arun, T. S. Huang and S. D. Blostein. "Least-Squares Fitting of Two 3-D Point Sets", *IEEE Trans. PAMI* 9(5) pp.698-700 (1987).
- [121] www.jobshop.com. "Lapping", <http://www.jobshop.com/techinfo/lappingdef.shtml>, [Accessed -2 November 2006].
- [122] J. Hugh. "Manufacturing Engineer on a disk : Gauge Blocks", <http://claymore.engineer.gvsu.edu/~jackh/eod/manufact/manufact-116.html#pgfld-141526>, 31 August, 2001 [Accessed - 23 October 2006].
- [123] E. R. Davies. *Machine Vision: theory, algorithms, practicalities*, Academic Press, London UK (1990).
- [124] E. R. Davies. "Image space transforms for detecting straight edges in industrial images", *Pattern Recognition Letters* 4 pp.185-192 (1986).
- [125] A. Sarti and S. Tubaro. "Detection and characterisation of planar fractures using a 3D Hough transform", *Signal Processing* 82 pp.1269-1282 (2002).
- [126] Ask Dr. Math™ - The Math Forum. "Orthogonal Distance Regression Planes", <http://mathforum.org/library/drmath/view/63765.html>, [Accessed - 19 March 2006].
- [127] C. F. Goldfarb and P. Prescod. *The XML handbook*, Prentice Hall Inc, NJ 07458 USA (2001).
- [128] jdom.org. "jdom.org ", <http://www.jdom.org/docs/apidocs/org/jdom/package-summary.html> [Accessed - 11 December 2006].
- [129] The Mathworks. "MATLAB[®] ", <http://www.mathworks.com/>, [Accessed - 11 December 2006].
- [130] T. S. Newman and A. K. Jain. "A Survey of Automated Visual Inspection", *CVGIP: Image Understanding* 61 pp.231-262 (1995).
- [131] M. Ristic and D. Brujic. "Efficient registration of NURBS geometry", *Image and Vision Computing* 15 pp.925-935 (1997).
- [132] K. H. Lee, H. Woo and T. Suk. "Point data reduction using 3D grids", *Int J Adv. Manuf. Technol* 18 pp.201-210 (2001).
- [133] A. Joneja. "Rapid Prototyping: 3D Printing, Fused Deposition Modelling", http://iesu5.ieem.ust.hk/dfaculty/ajay/courses/ieem317/lab_toc.html, [Accessed- January 2007].
- [134] Autodesk AliasStudio. <http://usa.autodesk.com/adsk/servlet/index?id=6871131&siteID=123112>, [Accessed - 28 April 2007].
- [135] FileFormat.info. "Wavefront OBJ File Format Summary", <http://www.fileformat.info/format/wavefrontobj/egff.htm>, [Accessed - 15 January 2006].

- [136] M. Reddy. "Object files", <http://www.martinreddy.net/gfx/3d/OBJ.spec>, [Accessed - 15 January 2007].
- [137] Visual Computing Lab, ISTI, CNR. "MeshLab", v0.9 <http://meshlab.sourceforge.net>, (2006).
- [138] D. Brujic and M. Ristic. "Monte Carlo simulation and analysis of free-form surface registration", *Proc Instn Mech Engrs* 211(Part B) pp.605-617 (1997).
- [139] P. J. Besl and N. D. McKay. "A Method for Registration of 3-D Shapes", *IEEE Trans. PAMI* 14(2) pp.239-256 (1992).
- [140] H. Pottmann, S. Leopoldseder and M. Hofer. "Registration without ICP", *Technical Report 91*, Institute of Geometry, Vienna University of Technology, (2002).
- [141] A. W. Fitzgibbon. "Robust Registration of 2D and 3D Point sets", In *British Machine Vision Conference 2001* (2001).
- [142] S. M. Yamany, M. N. Ahmed and A. A. Farag. "Novel Surface Registration Using the Grid Closest Point (GCP) Transform", *Chicago, USA*, pp.809-813 (1998).
- [143] S. Rusinkiewicz and M. Levoy. "Efficient Variants of the ICP Algorithm", *Proceedings Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, pp.145-152 (2001).
- [144] Y. O. Halchenko. "Iterative Closest Point Algorithm. L1 Approach via Linear Programming", <http://www.onerussian.com/classes/cis780/icp-report.pdf>, [Accessed - April 2007].
- [145] N. J. Mitra, N. Gelfand, H. Pottmann and L. Guibas. "Registration of Point Cloud Data from a Geometric Optimization Perspective", *Eurographics Symposium on Geometry Processing*, (2004).
- [146] Y. Chen and G. Medioni. "Object modelling by registration of multiple range images", *Image and Vision Computing* 10(3) pp.145-155 (1992).
- [147] S. Leopoldseder, H. Pottmann and H. Zhao. "The d^2 -Tree: A Hierarchical Representation of the Squared Distance Function", *Technical Report 101*, Institute of Geometry, Vienna University of Technology, Vienna, Austria (2003).
- [148] J. Pirouet. , "Matching Point Clouds to CAD Models", *Loughborough University, Mechanical Engineering 02MMC500 Individual Final Year Report* (2003).
- [149] SPOTS project. "Standardisation project for optical techniques of strain measurement (SPOTS)", <http://www.opticalstrain.org/>, [Accessed- 25 March 2007].
- [150] R. Dettmer. "Big bird gets its wings", *IEE Review*, April pp.34-38 (2004).
- [151] R. L. Burguete. "Optical methods for deformation and strain measurement", *Personal Communication* (2006).

[152] Y. H. Huang, C. Quan, C. J. Tay and L. J. Chen. "Shape measurement by the use of digital image correlation", *Opt. Eng.* 44(8) pp.087011-1-087011-7 (2005).

[153] L. E. Scales. *Introduction to non-linear optimization*, Macmillan Publishers Ltd, London (1985).

Appendices

Appendix – A-1 – Optimisation of sphere centre coordinates

For each i^{th} point on the sphere surface, the error can be expressed as

$$\delta E_i = |\mathbf{r}_i - \mathbf{r}_0| - R \quad (1)$$

where $\mathbf{r}_i(x_i, y_i, z_i)$ is the i^{th} sphere coordinate, $\mathbf{r}_0(x_0, y_0, z_0)$ is the sphere centre coordinate estimated by the 3-D Hough transform (which we are going to optimise) and R is the *known* radius of the sphere. Thus, $R_i = |\mathbf{r}_i - \mathbf{r}_0|$ is the i^{th} *measured* radius of the sphere.

Therefore, the objective function we need to minimise is given as

$$S = \sum_i |\delta E_i|^2 = \sum_i (R_i - R)^2 \quad (2)$$

S can be expressed in more details as

$$S = \sum_i \left[(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 - 2R \left[(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 \right]^{1/2} + R^2 \right] \quad (3)$$

To apply Newton's method to the above equation, we need to populate a vector of first derivatives of S (the gradient vector, \mathbf{g}) and a matrix of second derivatives of S (the Hessian matrix, \mathbf{H}).

$$\mathbf{g} = \begin{bmatrix} \frac{\partial S}{\partial x_0} \\ \frac{\partial S}{\partial y_0} \\ \frac{\partial S}{\partial z_0} \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} \frac{\partial^2 S}{\partial x_0^2} & \frac{\partial^2 S}{\partial x_0 \partial y_0} & \frac{\partial^2 S}{\partial x_0 \partial z_0} \\ \frac{\partial^2 S}{\partial y_0 \partial x_0} & \frac{\partial^2 S}{\partial y_0^2} & \frac{\partial^2 S}{\partial y_0 \partial z_0} \\ \frac{\partial^2 S}{\partial z_0 \partial x_0} & \frac{\partial^2 S}{\partial z_0 \partial y_0} & \frac{\partial^2 S}{\partial z_0^2} \end{bmatrix} \quad (4)$$

Using Newton's method, the iterative scheme for improving the sphere centre coordinate, $\mathbf{r}_0(x_0, y_0, z_0)$, is therefore

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}_{j+1} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}_j - \mathbf{H}^{-1} \mathbf{g} \quad (5)$$

The first derivatives with respect to x_0 is expressed as,

$$\frac{\partial S}{\partial x_0} = -2 \sum_i \left[(x_i - x_0) - \frac{(x_i - x_0)R}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2}} \right] \quad (6)$$

The above equation can be rewritten as

$$\frac{\partial S}{\partial x_0} = -2 \sum_i (x_i - x_0) \left(1 - \frac{R}{R_i} \right) \quad (7)$$

Similarly, for $\frac{\partial S}{\partial y_0}$ and $\frac{\partial S}{\partial z_0}$,

$$\frac{\partial S}{\partial y_0} = -2 \sum_i (y_i - y_0) \left(1 - \frac{R}{R_i} \right) \quad (8)$$

$$\frac{\partial S}{\partial z_0} = -2 \sum_i (z_i - z_0) \left(1 - \frac{R}{R_i} \right) \quad (9)$$

The second derivatives with respect to x_0 can be expressed thus

$$\frac{\partial^2 S}{\partial x_0^2} = -2 \sum_i (-1) + 2R \sum_i \frac{\partial}{\partial x_0} \left(\frac{x_i - x_0}{R_i} \right) \quad (10)$$

$$\frac{\partial}{\partial x} \left(\frac{x_i - x_0}{R_i} \right) = \frac{R_i(-1) - (x_i - x_0) \left(\frac{\partial R_i}{\partial x_0} \right)}{R_i^2} \quad (11)$$

$$\frac{\partial^2 S}{\partial x_0^2} = \frac{-R_i^2 + (x_i - x_0)^2}{R_i^3} \quad (12)$$

$$\frac{\partial^2 S}{\partial x_0^2} = 2 \left(1 + R \sum_i \frac{(x_i - x_0)^2 - R_i^2}{R_i^3} \right) \quad (13)$$

Similarly, for $\frac{\partial^2 S}{\partial y_0^2}$ and $\frac{\partial^2 S}{\partial z_0^2}$,

$$\frac{\partial^2 S}{\partial y_0^2} = 2 \left(1 + R \sum_i \frac{(y_i - y_0)^2 - R_i^2}{R_i^3} \right) \quad (14)$$

$$\frac{\partial^2 S}{\partial z_0^2} = 2 \left(1 + R \sum_i \frac{(z_i - z_0)^2 - R_i^2}{R_i^3} \right) \quad (15)$$

The cross derivatives with respect to x_0 and y_0 is given as

$$\frac{\partial^2 S}{\partial x_0 \partial y_0} = \frac{\partial}{\partial y_0} \left[-2 \sum_i (x_i - x_0) \left(1 - \frac{R}{R_i} \right) \right] = 2R \sum_i (x_i - x_0) \left(\frac{1}{R_i^2} \right) \frac{\partial R_i}{\partial y_0} \quad (16)$$

where $\frac{\partial R_i}{\partial y_0} = \frac{1}{R_i} (y_i - y_0)$. Thus,

$$\frac{\partial^2 S}{\partial x_0 \partial y_0} = -2R \sum_i \frac{(x_i - x_0)(y_i - y_0)}{R_i^3}. \quad (17)$$

Note that $\frac{\partial^2 S}{\partial x_0 \partial y_0} = \frac{\partial^2 S}{\partial y_0 \partial x_0}$, $\frac{\partial^2 S}{\partial x_0 \partial z_0} = \frac{\partial^2 S}{\partial z_0 \partial x_0}$ and $\frac{\partial^2 S}{\partial y_0 \partial z_0} = \frac{\partial^2 S}{\partial z_0 \partial y_0}$

Similarly, for $\frac{\partial^2 S}{\partial x_0 \partial z_0}$ and $\frac{\partial^2 S}{\partial y_0 \partial z_0}$,

$$\frac{\partial^2 S}{\partial x_0 \partial z_0} = -2R \sum_i \frac{(x_i - x_0)(z_i - z_0)}{R_i^3} \quad (18)$$

$$\frac{\partial^2 S}{\partial y_0 \partial z_0} = -2R \sum_i \frac{(y_i - y_0)(z_i - z_0)}{R_i^3} \quad (19)$$

Appendix – A-2 – Contact details of suppliers for plane artefact components

Gauge blocks, surface plates, and length bars

Mahr Plc UK

Contact details: 01908 563700, www.mahr.com

Products: gauge blocks.

Broomfield Gauges

Contact details: 01484 664982, www.broomfieldgauges.com

Products: tungsten carbide gauge blocks.

Ely Metrology (Sales & Services) Ltd

Contact details: 01332 367475, www.eleyservice.com

Products: gauge blocks and granite surface plates.

Starrett

Contact details: 01835 866333, www.starrett.co.uk

Products: gauge blocks and granite surface plates.

Gandm Tools

Contact details: 01903892510, www.gandmtools.co.uk

Products: supplier of used machinery and metrology parts e.g. used length bar sets.

Optical Flats

Comar

Contact details: David Marsh - 01223 245470

Products: optical flats.

Melles Griot

Contact details: 01480 420800, www.mellesgriot.com

Products: optical flats optical flats.

Knight Optical

Contact details: 01634 296662, www.knightoptical.co.uk

Products: optical flats.

Gooch and Housego

Contact details: Michelle - 01460 52271, www.goochandhousego.com.

Products: ceramic and optical glass materials.

Oldham Optical

Contact details: 01723 506050

Products: optical flats.

CalPhotonics

Contact details: 0131 555 4848

Comments: optical flats. Have

Ceramics (e.g. Zirconia Tiles/Bars) and contract lapping services

Opus Metrology

Contact details: 01536204681, www.opus.co.uk

Products: gauge blocks and contract lapping.

Good Fellow

Contact details: 0800 7314653, www.goodfellow.com

Products: ceramic sheets and spheres e.g. zirconia and tungsten carbide.

Dynamic ceramic

Contact details: 01270 501000, www.dynacer.com

Products: ceramic tiles, bars and spheres.

Lapmaster International

Contact details : Kevin Hook (khook@lapmaster.co.uk). 01752 893 191,

www.lapmaster.co.uk

Products: contract lapping services for a variety of materials, such as hardened steel , sapphire, nickel, aluminium, glass up to 1.2 m diameter. Large articles can be lapped to a flatness of about 10 μm and parallelism of 5 μm to 10 μm .

Precision Lapping

Contact details: 0118 9735989, www.precision-lapping.co.uk

Products: contract lapping services for steel components only to flatness of 0.25 μm over 150 mm diameter and parallelism to 2.5microns.

Agate Products

Contact details: 0208 3978397, www.agateproducts.co.uk

Products: ceramic parts e.g. zirconia and contract lapping services on ceramic components only to flatness and parallelism of 1 μm .

Precision ground blocks

Cromwell Tools

Contact details: Leicester office - 0116 288 8444, <http://www.cromwell.co.uk/>

Products: variety of equipment and tools, including measuring equipment such as gauge blocks, precision ground blocks, etc.

Precision ground plates and flat bars

Burrhart Machinery Ltd

Contact details : 01582563400, www.burrhart.co.uk

Products: aluminium and steel precision ground plates.

Appendix – A-3 – XML interface for calibration software - showing bundle adjustment section

```

<?xml version="1.0" encoding="UTF-8"?>
<pv_sms>
  <pv_calibration>
    <pv_var name="Status">Valid</pv_var>
    <pv_var name="UseBundleAdjustmentIndex" class="i32">0</pv_var>
    <pv_var name="UseSequenceIndex" class="i32">1</pv_var>
    <!--Bundle adjustment settings-->
    <pv_bundle_adjustment index="0">
      <pv_var name="DateStamp">22-Apr-2007 19:57:01</pv_var>
      <pv_var name="CalibrationType">primary</pv_var>
      <pv_var name="NumOfIterations" class="i32">1</pv_var>
      <pv_var name="StartingIterationIndex" class="i32">0</pv_var>
      <pv_var name="NumOfPoses" class="i32">1</pv_var>
      <pv_var name="FilePathVector_ArtefactPose" class="cell">
        <pv_var>070419-150836</pv_var>
      </pv_var>
      <pv_var name="FilePath_SMS">G:/Data-19-04-07/Poses/</pv_var>
      <pv_var name="FilePath_CalibData">E:/tk/params</pv_var>
      <pv_var name="PlaneArtefact_Baseline"
class="f64">0.00000000e+000</pv_var>
      <pv_var name="DilateSize" class="f64">0.00000000e+000</pv_var>
      <pv_var name="PlaneArtefact_DistanceThreshold"
class="f64">1.00000000e+001</pv_var>
      <pv_var name="PhaseErrorMaxThresold" class="f64" size="1
4">5.00000000e-001 5.00000000e-001 5.00000000e-001 5.00000000e-
001</pv_var>
      <pv_var name="ModulationMinThreshold" class="f64" size="1
4">1.00000000e-001 1.00000000e-001 1.00000000e-001 1.00000000e-
001</pv_var>
      <!--Flags for debugging-->
      <pv_debug>
        <pv_var name="DrawFiguresFlag" class="i32">0</pv_var>
        <pv_var name="WriteToFileFlag" class="i32">1</pv_var>
        <pv_var name="DebugFlag" class="i32">0</pv_var>
        <pv_var name="PlotEpsilon1Flag" class="i32">0</pv_var>
        <pv_var name="PlotEpsilon2Flag" class="i32">0</pv_var>
        <pv_var name="CheckPlaneFlag" class="i32">0</pv_var>
      </pv_debug>
      <!--Hough transform variables-->
      <pv_hough>
        <pv_var name="VotesThreshold" class="i32" size="1 4">4000 6000
6000 6000</pv_var>
        <pv_var name="Radius" class="f64">2.54000000e+001</pv_var>
        <pv_var name="Limits" class="f64" size="1 5">8.00000000e-002
3.00000000e+000 0.00000000e+000 5.08000000e+001
0.00000000e+000</pv_var>
        <pv_var name="Connectivity"
class="f64">2.60000000e+001</pv_var>
        <pv_var name="CellsPerRadius"
class="f64">2.54000000e+000</pv_var>

```

```

    <pv_var name="RadLimitFactor"
class="f64">1.10000000e+000</pv_var>
  </pv_hough>
  <!--Calibration Artefacts-->
  <pv_artefact_type name="Sphere">
    <pv_var name="NumberOfPointsPerArt" class="i32">36</pv_var>
    <pv_var name="ArtSubtype">L200</pv_var>
  </pv_artefact_type>
  <pv_artefact_type name="Plane">
    <pv_var name="NumberOfPointsPerArt" class="i32">64</pv_var>
    <pv_var name="ArtSubtype">none</pv_var>
  </pv_artefact_type>
  <pv_var name="Calib_CPPair_DevIndices" class="i32" size="1 2">0
0</pv_var>
  <pv_scaling_params>
    <pv_var name="OffsetFactor" class="f64">5.00000000e-
002</pv_var>
    <pv_var name="ScaleFactor" class="i32">10</pv_var>
  </pv_scaling_params>
  <pv_iteration index="0">
  <pv_input>
    <pv_sensor_parameters>
      <pv_var name="c0_value" class="f64" size="12 1">-
4.09177042e+002 -5.86427447e+000 1.82700461e+003 -3.12382961e+000
2.23746340e-001 -3.13282525e+000 0.00000000e+000 0.00000000e+000
4.01217414e+001 0.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
      <pv_var name="c0_status" class="f64" size="12
1">0.00000000e+000 0.00000000e+000 0.00000000e+000 0.00000000e+000
0.00000000e+000 0.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
      <pv_var name="p0_value" class="f64" size="12
1">2.02951151e+002 2.41645172e+001 1.82889571e+003 -3.12215571e+000 -
2.88470780e-001 -1.56277842e+000 0.00000000e+000 6.00000000e+000
3.37351530e+001 0.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
      <pv_var name="p0_status" class="f64" size="12
1">1.00000000e+000 1.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
    </pv_sensor_parameters>
  </pv_input>
  <pv_output>
    <pv_sensor_parameters>
      <pv_var name="c0_value" class="f64" size="12 1">-
4.09177042e+002 -5.86427447e+000 1.82700461e+003 -3.12382961e+000
2.23746340e-001 -3.13282525e+000 0.00000000e+000 0.00000000e+000
4.01217414e+001 0.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
      <pv_var name="c0_status" class="f64" size="12
1">0.00000000e+000 0.00000000e+000 0.00000000e+000 0.00000000e+000
0.00000000e+000 0.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
    </pv_sensor_parameters>
  </pv_output>

```

```

    <pv_var name="p0_value" class="f64" size="12
1">2.02951151e+002 2.41645172e+001 1.82889571e+003 -3.12215571e+000 -
2.88470780e-001 -1.56277842e+000 0.00000000e+000 6.00000000e+000
3.37351530e+001 0.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
    <pv_var name="p0_status" class="f64" size="12
1">1.00000000e+000 1.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
    </pv_sensor_parameters>
    <pv_art_pose name="070419-150836">
    <pv_cam_proj_pair name="c0p0">
    <pv_var name="CmmMismatchError"
class="f64">2.08452152e+001</pv_var>
    <pv_var name="EulerAngles" class="f64" size="3
1">3.13909058e+000 -5.17374945e-001 0.00000000e+000</pv_var>
    <pv_var name="TranslationVector" class="f64" size="3
1">1.62726841e+002 2.13151494e+001 1.66982343e+002</pv_var>
    </pv_cam_proj_pair>
    <pv_optimised_pose>
    <pv_var name="EulerAngles" class="f64" size="3
1">3.14757974e+000 -3.30214851e-001 0.00000000e+000</pv_var>
    <pv_var name="TranslationVector" class="f64" size="3
1">4.77178292e+002 3.54885963e+001 2.59514743e+002</pv_var>
    </pv_optimised_pose>
    </pv_art_pose>
    <pv_scaling_params>
    <pv_var name="OffsetFactor" class="f64">5.00000000e-
002</pv_var>
    <pv_var name="ScaleFactor" class="i16">10</pv_var>
    </pv_scaling_params>
    <pv_var name="NumOfIterations"
class="f64">1.50000000e+002</pv_var>
    <pv_result name="ObjectiveFunction">
    <pv_var name="Start" class="f64">1.26032568e+001</pv_var>
    <pv_var name="End" class="f64">7.50996232e+000</pv_var>
    </pv_result>
    <pv_result name="Epsilon1">
    <pv_var name="Start" class="f64">3.44126700e-001</pv_var>
    <pv_var name="End" class="f64">2.29997469e-001</pv_var>
    </pv_result>
    <pv_result name="Epsilon2">
    <pv_var name="Start" class="f64">1.25985578e+001</pv_var>
    <pv_var name="End" class="f64">7.50643959e+000</pv_var>
    </pv_result>
    <pv_sensor_parameters>
    <pv_var name="c0_value" class="f64" size="12 1">-
4.09177042e+002 -5.86427447e+000 1.82700461e+003 -3.12382961e+000
2.23746340e-001 -3.13282525e+000 -7.07850714e+000 4.10534680e-001
3.74906373e+001 -7.27508858e-006 0.00000000e+000
0.00000000e+000</pv_var>
    <pv_var name="c0_status" class="f64" size="12
1">0.00000000e+000 0.00000000e+000 0.00000000e+000 0.00000000e+000
0.00000000e+000 0.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>

```

```
<pv_var name="p0_value" class="f64" size="12"
1">2.08441835e+002 2.62459410e+001 1.75364308e+003 -3.12015699e+000 -
1.24367008e-001 -1.56569619e+000 2.63817236e-001 5.90166350e+000
2.85855984e+001 -6.39132864e-007 0.00000000e+000
0.00000000e+000</pv_var>
  <pv_var name="p0_status" class="f64" size="12"
1">1.00000000e+000 1.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 1.00000000e+000 1.00000000e+000
1.00000000e+000 1.00000000e+000 0.00000000e+000
0.00000000e+000</pv_var>
  </pv_sensor_parameters>
  </pv_output>
  </pv_iteration>
  </pv_bundle_adjustment>
  </pv_calibration>
</pv_sms>
```

Appendix – A-4 – Description of folders for calibration software MATLAB[®] functions and C mex files

Calibration

Contains all the gateway scripts for doing the calibration of the SMS. This folder contains scripts that act as a gateway for doing a calibration using either a sphere artefact or a plane artefact. 'xml_calib_planes' is a script that calibrates the SMS using a two-plane artefact (see Figure A-1), where all the settings are read from an XML document 'xml_calib_spheres' is a script that calibrates the SMS using a multiple-sphere artefact (see Figure A-2), while 'xml_calib_ballbar' is a script that calibrates the SMS using a ball bar artefact (see Figure A-2).

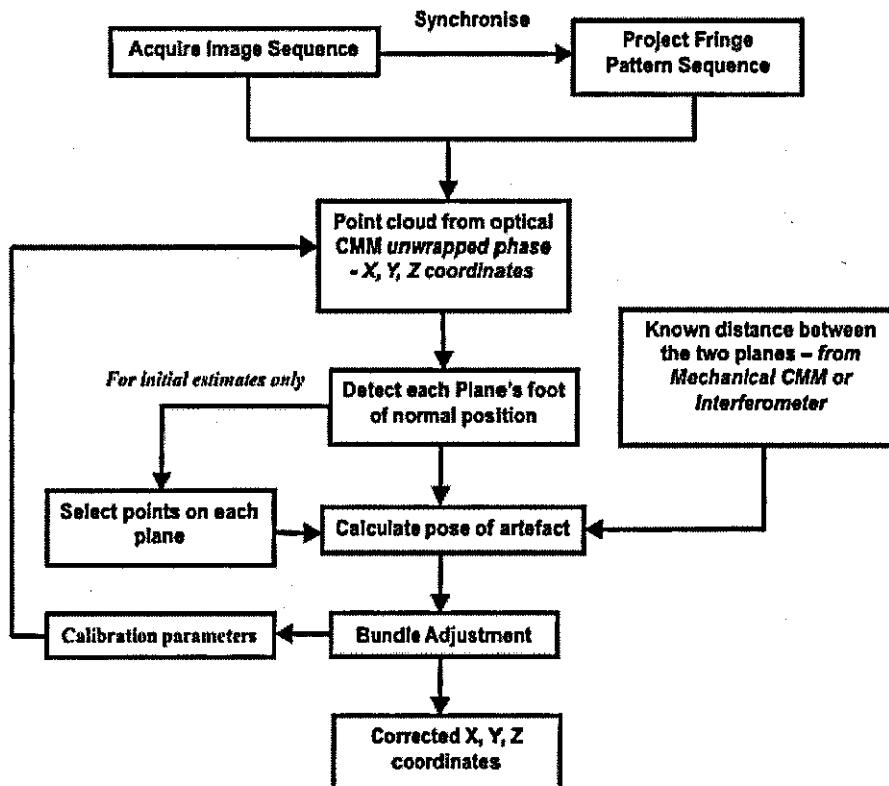


Figure A-4-1 : Calibration process using a plane calibration artefact

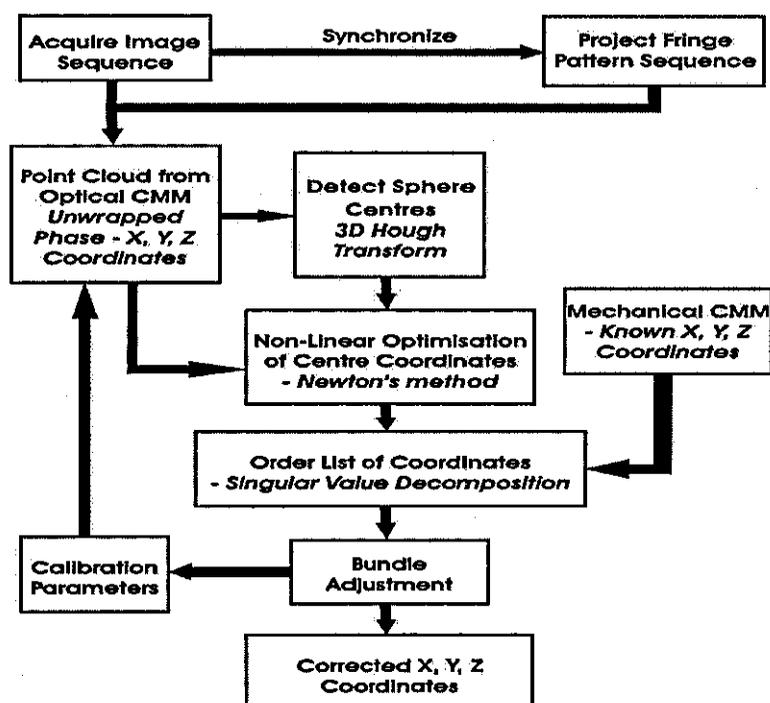


Figure A-4-2 : Calibration process using a sphere calibration artefact

2DHoughTransform

Contains all the function m-files for the 2D Hough transform which is used for detecting circles in images. The gateway function is 'pv_hough_transfrom2D'. This function can be used within other function m-files and GUI m-files. The main difference is that when used within a function m-file, the image being analysed and the detected circle centres are drawn in a new figure. For a GUI, it may be required that these are displayed in a specific axes object with a given handle. When 'pv_hough_transfrom2D' has an axes object handle passed as one of the input arguments, the function draws the results on such axes.

3DHoughTransform

Contains all the function m-files for the 3D Hough transform which is used for detecting spheres in a point cloud. The gateway function is 'pv_hough_transfrom3D'. The following C mex files are required with this function : 'pv_hmap_mex', 'pv_label_mex' and 'pv_hough_votes_mex'. Also included in this folder are functions for matching all the detected sphere centres with some known set of coordinates (e.g. those obtained from a CMM). These function m-files are used only when calibrating using a multiple-sphere or ball bar artefact.

PlaneDetection

Contains all the function m-files for detecting the foot of normal position of planes and also to estimate the pose of two planes relative one another. The functions in this folder are used when the SMS is calibrated using a plane artefact. The gateway function for detecting the foot of normal position of planes is 'pv_detect_plane'.

BundleAdjustment

Contains all the function m-files for carrying out the bundle adjustment algorithm. The gateway function is 'pv_bundleadj_lm-nonlinear_optV4_3'. The 'V4_3' string after some function m-files indicate the version numbers based on updates to the bundle adjustment algorithm. The function 'pv_bundleadj_set_system_parametersV4_3' creates the XI1 and XI2 tables in the format specified by Huntley [86].

PointCloudMeshing

Contains all the function m-files for creating a mesh from a point cloud and writing to either an 'obj' file format or 'stl' format. The gateway function is 'pv_mesh_point_cloud'. A C mex file 'pv_mesh_pt_cloud_mex' is required with this function.

SphereOptimisation

Contains all the function m-files for non-linear optimisation of sphere centres as described in Appendix A-1. This is an optional operation after sphere centres have been detected using the 3D Hough transform. Also included is a function m-file for labelling all the pixels in a phase map that belong to each respective sphere centre. The gateway function for non-linear optimisation is 'pv_sphereopt_newton'. The function requires the C mex file 'pv_hessianguardient_mex'.

Utilities

Contains general purpose function m-files which could be used as part of the calibration process or in the GUI. E.g. function for drawing the circle pattern for the 2-D DLT artefact, functions for initialising camera/projector parameters using the direct linear transformation method (DLT) with either coplanar or non-coplanar control points, function for calculating surface normals, etc.

XML

Contains function m-files for reading calibration variables from an XML file and also writing calibration variables to an XML file. The m-files are for processing certain variables relevant to an aspect of the calibration process. For example, settings for the cameras and projectors, Hough transform variables, bundle adjustment variables, and information on calibration artefacts. It also contains a general purpose function m-file

'pv_xml_create_element' which creates an XML element object as a child element of a user given parent element.

Appendix – A-5 – C++ interface for implementation of adaptive look-up table (LUT)

```

#define PV_MAX_LEVELS 8 // number of maximum levels to which
                        // lookup table iterator can reach

// Enumerator for selecting iterator's mode
enum PV_E_ITERMODE {PV_E_ONE, PV_E_MULT};

// move one level, or move multiple levels
typedef PV_E_ITERMODE PV_T_ITERMODE;

#define PV_T_GRID_FP float// To be able to change precision easily

typedef struct PV_T_FixedGrid
(
    PV_T_GRID_FP *pfTableLimits;
    struct GridParameters
    (
        PV_T_I16      SamplingFactor;
        PV_T_I16      MaxLevel;
    )GridParameters;
    struct AdaptiveGrid
    (
        PV_T_I16      SamplingFactor;// for dividing each cell
        PV_T_I16      MaxLevel;
        PV_T_GRID_FP MaxDistance;// maximum distance from object within which
                                //adaptive should be done.
    )AdaptiveGrid;
}PV_T_FixedGrid;

// ***** Class declarations *****

//*****
// A 3d Point object, with member functions for manipulating
// points in space
class _PV_LOOKUPTABLE_API PV_C_Point3D
{
public:
    // Default Constructor
    PV_C_Point3D();
    // Alternate Constructor
    PV_C_Point3D( PV_T_GRID_FP fX, PV_T_GRID_FP fY, PV_T_GRID_FP fZ );
    // Copy constructor
    PV_C_Point3D( const PV_C_Point3D& oPt );
    // Destructor
    virtual ~PV_C_Point3D() {};
    // Access functions to return x,y,z coordinates
    // Return x coordinate
    inline const PV_T_GRID_FP GetX() const { return m_fCoords[0];}
    // Return y coordinate
    inline const PV_T_GRID_FP GetY() const { return m_fCoords[1]; };
    // Return z coordinate
    inline const PV_T_GRID_FP GetZ() const { return m_fCoords[2]; };
    // Return as an array
    inline const PV_T_GRID_FP* const GetCoords() const { return &m_fCoords[0]; }
    // Access functions to set values to x,y,z coordinates
    PV_C_Point3D& SetX( PV_T_GRID_FP fX );
    PV_C_Point3D& SetY( PV_T_GRID_FP fY );
    PV_C_Point3D& SetZ( PV_T_GRID_FP fZ );

    // Operators

```

```

// Assignment
PV_C_Point3D& operator=( const PV_C_Point3D& oPt );
// Addition
friend PV_C_Point3D operator+( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2
);
// Subtration
friend PV_C_Point3D operator-( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2
);
// Multiplication
friend PV_C_Point3D operator*( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2
);
// Division by another PV_C_Point3D object
friend PV_C_Point3D operator/( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2
);
// Division by scalar integer
friend PV_C_Point3D operator%(const PV_C_Point3D& oPt1, const PV_T_U8 oInt);
// Division by scalar float
friend PV_C_Point3D operator/(const PV_C_Point3D& oPt1, const PV_T_GRID_FP
fFloat);
// Equal to
friend bool operator==( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2 );
// Not Equal to
friend bool operator!=( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2 );
// Less than
friend bool operator<( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2 );
// Less than or equal to
friend bool operator<=( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2 );
// Greater than
friend bool operator>( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2 );
// Greater than or equal to
friend bool operator>=( const PV_C_Point3D& oPt1, const PV_C_Point3D& oPt2 );
// Return the norm of point
friend PV_T_GRID_FP Norm( const PV_C_Point3D& oPt );
// Return the cross product of two PV_C_Point3D objects
friend PV_C_Point3D CrossProduct3D(PV_C_Point3D oPt1,PV_C_Point3D oPt2);
// Return the dot product of two PV_C_Point3D objects
friend PV_T_GRID_FP DotProduct3D(PV_C_Point3D ptPt1,PV_C_Point3D ptPt2);
private:
// voxel coordinates
PV_T_GRID_FP m_fCoords[3];
};

//*****
// The opaque data object being stored in the lookup table
class _PV_LOOKUPTABLE_API PV_C_ClientData
{
public:
// Default Constructor
PV_C_ClientData();
// Alternate Constructor
PV_C_ClientData( PV_T_U8 ucTaylorSeriesOrder );
// Copy Constructor
PV_C_ClientData( const PV_C_ClientData& Src );
// Destructor
virtual ~PV_C_ClientData() { }
// Free dynamic memory
void DisposeCoeffs(){ delete[] m_pfTaylorSeriesCoeffs; }

// Assignment operator
const PV_C_ClientData& operator=( const PV_C_ClientData& Src );
// Set the taylor series order
void SetTaylorSeriesOrder( PV_T_U8 ucOrder );
// Set the taylor series coefficients
void SetTaylorSeriesCoeffs( PV_T_GRID_FP *Coeffs );
// Access functions
inline PV_T_U8 GetTaylorSeriesOrder() const { return m_ucTaylorSeriesOrder; }
inline PV_T_GRID_FP* GetTaylorSeriesCoeffs() const { return
m_pfTaylorSeriesCoeffs; }
// --- TODO ---
// Read()

```

```

        // Write(PV_T_Grid_FP *pfTaylorSeriesCooeffs)
private:
    PV_T_U8      m_ucTaylorSeriesOrder;//order of taylor series polynomial
    PV_T_GRID_FP *m_pfTaylorSeriesCooeffs; // Taylor series coefficients for grid
point (voxel)
//which are dependent on the order of taylor series
//expansion. Apart from the grid point itself, 3 extra coords
//for 1st order, 9 extra coords for 2nd order(though only 6
//would need to be stored)
);

// Pre-declaration
class PV_C_GridLevel3D;

//*****
// Base point of lookup table management structure.
// Data for the lookup table is stored here.
class _PV_LOOKUPTABLE_API PV_C_GridData
{
public:
    // Default constructor
    PV_C_GridData();
    // Alternate Constructor
    // pass preformed structure pointer to data object
    PV_C_GridData( PV_T_U8 ucTaylorSeriesOrder );
    // Copy constructor
    PV_C_GridData( const PV_C_GridData& Src );
    // Made public so that access can be made to initialise Next level pointer
    //PV_C_GridLevel3D *m_poNextLevel; // pointer to the next (lower) grid level
    // Destructor
    virtual ~PV_C_GridData() { }
    // Free dynamic memory
    void DisposeGridData();
    // Assignment operator
    const PV_C_GridData& operator=( const PV_C_GridData& Src );
    // Access functions
    // Return Client Data
    inline PV_C_ClientData GetClientData() const { return m_oTempData; }
    // Return the order of client data
    inline PV_T_I16 GetTaylorSeriesOrder() const { return
m_oTempData.GetTaylorSeriesOrder(); }
    // Return the taylor series coefficients of client data
    inline PV_T_GRID_FP* GetTaylorSeriesCooeffs() const { return
m_oTempData.GetTaylorSeriesCooeffs(); }
    // Return the next level
    inline PV_C_GridLevel3D* GetNextLevel() const { return m_poNextLevel; }
    // Set the Taylor series order
    void SetTaylorSeriesOrder( PV_T_U8 ucTaylorSeriesOrder );
    // Set the client data
    void SetClientData( PV_C_ClientData& oClientData );
    // Set the next level
    void SetNextLevel( PV_C_GridLevel3D* poNextLevel );
    // Return the index of the level
    const PV_T_I16 GetLevelIndex() { return m_iLevelIndex; }

    // --- TODO ---
    // Calculate the coordinate of the Grid point
    //PV_C_Point3D CalculateCoord();
    // Calculate the taylor series coefficients of the Grid point
    // process: get the sampling factor from level where grid lies
    // use this along with oPoint to calculate coords around the grid pt
    // calculate each coefficient, and then call the setcoeffs fn of client data
    // to store the values
    void CalculateCooeffs( PV_C_Point3D oTriangleCoord[], PV_C_Point3D oPoint,
PV_C_Point3D oCellLimits );
protected:
    void Copy( const PV_C_GridData& Src );
private:
    PV_C_ClientData      m_oTempData;//instance of data object
    PV_T_I16             m_iLevelIndex;//identifier of the level where the grid point lies

```

```

        PV_C_GridLevel3D      *m_poNextLevel;//points to next level, if a condition for
subsampling

        //is fulfilled
};

//*****
// Level on lookup table having a number of grid points
class _PV_LOOKUPTABLE_API PV_C_GridLevel3D
{
public:
    // Default Constructor - set ucSamplingFactor = 2
    PV_C_GridLevel3D();
    // Alternate Constructor
    PV_C_GridLevel3D( PV_T_U8 ucSamplingFactor );
    // Copy constructor
    PV_C_GridLevel3D( const PV_C_GridLevel3D& Src );
    // Destructor
    virtual ~PV_C_GridLevel3D() { }
    // Free dynamic memory
    void DisposeLevel();
    // Assignment operator
    const PV_C_GridLevel3D& operator=( const PV_C_GridLevel3D& Src );
    // Access functions
    // Return sampling factor
    inline PV_T_U8 GetSamplingFactor() const { return m_ucSamplingFactor; }
    // Return number of data
    inline PV_T_I32 GetNumOfData() const { return m_iNumOfData; }

    // Return the GridData objects
    inline PV_C_GridData* GetGridData() const { return m_poGrid; } // vector (of
length m_iNumOfData)

//representing the grids on this level

protected:
    void Copy( const PV_C_GridLevel3D& Src );

private:
    PV_T_U8          m_ucSamplingFactor; // sampling for discretising
PV_C_GridData object
    PV_T_I32         m_iNumOfData; //number of grid data objects in the level
    PV_C_GridData   *m_poGrid; // vector (of length m_iNumOfData)
                    //representing the grids at this level
                    //PV_T_I32         m_iLevelId;// give the level an id number
};

//*****
// Management structure for the lookup table
//class _PV_LOOKUPTABLE_API PV_C_LookUpTable3D
class _PV_LOOKUPTABLE_API PV_C_LookUpTable3D
{
public:
    // Default Constructor - creates top level automatically
    PV_C_LookUpTable3D();
    // Alternate Constructor - creates top level automatically
    PV_C_LookUpTable3D( PV_T_GRID_FP *pfTableLimits, PV_T_I16* piThresh, PV_T_GRID_FP
fMaxDistance );
    // Copy constructor
    PV_C_LookUpTable3D( const PV_C_LookUpTable3D& Src );
    // Destructor
    virtual ~PV_C_LookUpTable3D() { }
    // Assignment operator
    const PV_C_LookUpTable3D& operator=( const PV_C_LookUpTable3D& Src );
    // Collapse all levels
    void DisposeAllLevels();
    // Collapse current level
    void DisposeThisLevel( PV_C_GridData *poGridData );
    // Create the lookup table data structure.
    void CreateDataStructure( PV_C_Point3D **ppfTriangleCoords, PV_T_I32 iNumOfFacets,

```

```

                PV_T_U8 ucTaylorSeriesOrder, PV_T_U8 ucSamplingFactor, PV_T_U8
StorageMode );
    // Create sub-levels (would be called recursively depending on if sub-division
    // criterion is met. In this case, if a triangle segment lies within a cell)
    void CreateLevel( PV_C_Point3D poTriangleCoord[], PV_C_Point3D oPoint,
                    PV_C_GridLevel3D *poLevel, PV_T_U8 ucTaylorSeriesOrder,
                    PV_T_U8 ucSubSamplingFactor, PV_C_Point3D oLevelLimits )

    // To detect if a triangle segment of model data lies within/intersects a cell,
    // which will be a condition for subdividing the cell
    bool Intersect( PV_C_Point3D oTriangleCoord[], PV_C_Point3D oPoint, PV_C_Point3D
oCellLimits);

    // Return value at a grid point in lookup table
    //PV_C_Point3D ReadFromGrid( PV_C_Point3D oPoint );
// PV_T_GRID_FP* ReadFromGrid( PV_C_Point3D oPoint );
    // Access function
    // Return the top level
    inline PV_C_GridLevel3D* GetTopLevel() const { return m_poLevel; }
    // Return the threshold for creating new levels
    inline const PV_T_I16* GetThreshold() const { return &m_piThresh[0]; }

    // Return the limits of lookup table
    inline const PV_T_GRID_FP* GetLimits() const { return &m_pfLimits[0]; }
protected:
    void Copy( const PV_C_LookUpTable3D& Src );

private:
    PV_C_GridLevel3D      *m_poLevel;      //the top level grid
    PV_T_GRID_FP          m_pfLimits[6]; // [xmin xmax ymin ymax zmin zmax] maximum
size of lookup table along x,y, and z axes
    PV_T_I16              m_piThresh[6]; // [Xthresh Ythresh Zthresh]
    PV_T_GRID_FP          m_fMaxDistance;
    //PV_T_GlobalList*    m_ptAllLevels; // linked list of all levels in lookup table
    //PV_T_LocalList*    m_ptLevels; // linked list of all levels on a layer
};

//*****
// Lookup table iterator
class _PV_LOOKUPTABLE_API PV_C_LUT3DIter
{
public:
    // Default Constructor
    PV_C_LUT3DIter();
    // Alternate Constructor
    PV_C_LUT3DIter( PV_C_LookUpTable3D oLUT, PV_C_Point3D oPoint );
    // Copy Constructor
    PV_C_LUT3DIter( const PV_C_LUT3DIter& Src);
    // Assignment Operator
    const PV_C_LUT3DIter& operator=(const PV_C_LUT3DIter& Src); // Assignment operator
    // Set a new point for iterating through LUT
    void SetNewPoint( PV_C_Point3D oPoint );
    // Get first Level in LUT - to be used for quick retrieval
    // for reading the distance from the LUT
    PV_C_GridLevel3D* GetFirstLevel2();

    // Get next Level in LUT - to be used for quick retrieval
    // for reading the distance from the LUT
    PV_C_GridLevel3D* GetNextLevel2();

    // Get last Level in LUT - to be used for quick retrieval
    // for reading the distance from the LUT
    PV_C_GridLevel3D* GetLastLevel2();
    // Get previous Level in LUT
    PV_C_GridLevel3D* GetPrevLevel();
    // Get level index at current iterator position in LUT
    inline PV_T_I32 GetLevelID() const { return m_iLevelIndex; }
    // Get level at a position in LUT
    PV_C_GridLevel3D* GetAtLevel();
    // Return grid data corresponding to an exp. point

```

```

PV_C_GridData* GetAtNearestGridData();
// Get last grid coordinate on a level
PV_C_Point3D GetAtNearestPointCoord();
// Get the point coordinate at current iterator position
PV_C_Point3D GetAtPointCoord();
// Get the point used to initialise iterator
inline PV_C_Point3D GetInterestPointCoord() const { return m_oPoint;}
#if PV_DETAILED_MOVEMENT// For a more detailed iteration within lookup table
// Get first Level in LUT
PV_C_GridLevel3D* GetFirstLevel();
// Get next Level in LUT
PV_C_GridLevel3D* GetNextLevel();
// Get last Level in LUT
PV_C_GridLevel3D* GetLastLevel();
// Get level at a position in LUT
PV_C_GridLevel3D* GetAtLevel( PV_T_I32 iLevel );
// Get grid index at current iterator position in LUT
inline PV_T_I32 GetGridID() const { return m_iGridIndex;}
// Get first grid point on a level
PV_C_GridData* GetFirstGridData();
// Get next grid point on a level
PV_C_GridData* GetNextGridData();
// Get previous grid point on a level
PV_C_GridData* GetPrevGridData();
// Get last grid point on a level
PV_C_GridData* GetLastGridData();
// Get last grid point on a level
const PV_C_GridData* GetAtGridData( PV_T_I32 iIndex );
// Return current grid data position of iterator
const PV_C_GridData* GetAtGridData();
// Get first grid coordinate on a level
PV_C_Point3D GetFirstPointCoord();
// Get next grid coordinate on a level
PV_C_Point3D GetNextPointCoord();
// Get previous grid coordinate on a level
PV_C_Point3D GetPrevPointCoord();
// Get last grid coordinate on a level
PV_C_Point3D GetLastPointCoord();
#endif // PV_DETAILED_MOVE
private:
    PV_C_LookUpTable3D    m_oLUT;
    PV_T_I32              m_iLevelIndex;
    PV_T_I32              m_iGridIndex;
    PV_T_I32              m_iIterMode;
    PV_C_GridData        m_oGridData;
    PV_C_Point3D         m_oGridCoord;
    PV_C_Point3D         m_oPoint;
    PV_C_Point3D         m_oLookUpPoint;
    PV_C_Point3D         m_oBasePoint;
    PV_T_I32              m_iNearestPointIndex;
    PV_C_Point3D         m_oLevelLimits;
    PV_T_GRID_FP         m_fDx;
    PV_T_GRID_FP         m_fDy;
    PV_T_GRID_FP         m_fDz;
    PV_T_GRID_FP         m_fMinDistance;
    PV_C_GridLevel3D     *m_poLevels[PV_MAX_LEVELS+1]; // store each level
    PV_T_I32              m_piIndex[PV_MAX_LEVELS+1]; // store the index of
    // 'snapped' grid data object
    PV_C_Point3D         m_poGridCoords[PV_MAX_LEVELS+1]; // store each grid coordinate
};

// Pre-declaration
class PV_C_MeshLUT3DIter;

//*****
// Wrapper class for access to the lookup able
//class _PV_LOOKUPTABLE_API PV_C_MeshLUT3D
class _PV_LOOKUPTABLE_API PV_C_MeshLUT3D
{
public:
    // Default Constructor

```

```

PV_C_MeshLUT3D();
// Alternate Constructor
PV_C_MeshLUT3D(PV_T_GRID_FP *pfTableLimits, PV_T_I16* piThresh, PV_T_GRID_FP
fMaxDistance);
// Alternate Constructor
PV_C_MeshLUT3D(PV_T_FixedGrid *ptGridParams);
// Copy constructor
PV_C_MeshLUT3D(const PV_C_MeshLUT3D& Src);
// Destructor
virtual ~PV_C_MeshLUT3D() {}
// Assignment Operator
const PV_C_MeshLUT3D& operator=(const PV_C_MeshLUT3D& Src); // Assignment operator
// Return distance to nearest surface from lookup table
PV_T_GRID_FP ReadFromGrid( PV_C_Point3D oPoint );
// Initialise iterator for MeshLUT
PV_C_MeshLUT3DIter InitialiseIterator( PV_C_Point3D oPoint );
// Populate the lookup table: initialise it and set values
void PopulateLUT( PV_C_Point3D **ppfTriangleCoords, PV_T_I32 iNumOfFacets,
PV_T_U8 ucTaylorSeriesOrder, PV_T_U8 ucSamplingFactor, PV_T_U8
StorageMode );
// Return the coordinates of the nearest lookup point to a point
PV_C_Point3D GetNearestGridPoint( PV_C_Point3D oPoint );
// Close up the lookup table
void DisposeLUT(){ m_oLUT.DisposeAllLevels(); }
// Get level index at current iterator position in LUT
inline const PV_C_LookUpTable3D GetLookupTable() const { return m_oLUT;}
protected:
void Copy( const PV_C_MeshLUT3D& Src );
private:
PV_C_LookUpTable3D m_oLUT;// the lookup table
};

//*****
// Lookup table iterator
class _PV_LOOKUPTABLE_API PV_C_MeshLUT3DIter
{
public:
// Default Constructor
PV_C_MeshLUT3DIter(); // Useful for initialising member variables to zero or NULL
// Alternate Constructor
PV_C_MeshLUT3DIter( PV_C_MeshLUT3D oMeshLUT, PV_C_Point3D oPoint );
// Copy Constructor
PV_C_MeshLUT3DIter( const PV_C_MeshLUT3DIter& Src);
// Assignment Operator
const PV_C_MeshLUT3DIter& operator=(const PV_C_MeshLUT3DIter& Src); // Assignment
//operator
// Set a new point for iterating through LUT
void SetNewPoint( PV_C_Point3D oPoint );
// Get first Level in LUT
PV_C_GridLevel3D* GetFirstLevel();
// Get next Level in LUT
PV_C_GridLevel3D* GetNextLevel();
// Get previous Level in LUT
PV_C_GridLevel3D* GetPrevLevel();
// Get last Level in LUT
const PV_C_GridLevel3D* GetLastLevel();
// Return the current level
const PV_C_GridLevel3D* Level();
// Return the current grid data object
const PV_C_GridData* GridData();
// Return the real world coordinate of the 'snapped'
// grid data object
PV_C_Point3D GridCoord();
// Return the distance to the nearest surface
PV_T_GRID_FP Distance();
// Move iterator to the next level and snap to a grid data object
PV_C_MeshLUT3DIter& operator++;
// Move iterator one level up and snap to a grid data object
PV_C_MeshLUT3DIter& operator--;
#if PV_DETAILED_MOVEMENT// For a more detailed movement within lookup table
// Get level at a position in LUT

```

```

PV_C_GridLevel3D* GetAtLevel( PV_T_I32 iLevel );
// Get level at current iterator position in LUT
PV_C_GridLevel3D* GetAtLevel();
// Get level index at current iterator position in LUT
//inline PV_T_I32 GetLevelID() const { return m_iLevelIndex;}
// Get first grid point on a level
PV_C_GridData* GetFirstGridData();
// Get next grid point on a level
PV_C_GridData* GetNextGridData();
// Get previous grid point on a level
PV_C_GridData* GetPrevGridData();
// Get last grid point on a level
PV_C_GridData* GetLastGridData();
// Get grid point for a particular point
const PV_C_GridData* GetAtGridData( PV_T_I32 iIndex );
// Return current grid data position of iterator
const PV_C_GridData* GetAtGridData();
// Return grid data corresponding to an exp. point
const PV_C_GridData* GetAtNearestGridData();
// Get first grid coordinate on a level
PV_C_Point3D GetFirstPointCoord();
// Get next grid coordinate on a level
PV_C_Point3D GetNextPointCoord();
// Get previous grid coordinate on a level
PV_C_Point3D GetPrevPointCoord();
// Get last grid coordinate on a level
PV_C_Point3D GetLastPointCoord();
// Get the point coordinate at current iterator position
PV_C_Point3D GetAtPointCoord();
// Get point coordinate of a grid point corresponding to an exp. point
PV_C_Point3D GetAtNearestPointCoord();
#endif // PV_DETAILED_MOVEMENT
private:
    PV_C_MeshLUT3D    m_oMeshLUT;
    PV_C_LUT3DIter   m_oLUTIter;
};

// ***** Global variable declarations *****
// Return the cross product of two PV_C_Point3D objects
PV_C_Point3D CrossProduct3D(PV_C_Point3D oPt1,PV_C_Point3D oPt2);

// Return the dot product of two PV_C_Point3D objects
PV_T_GRID_FP DotProduct3D(PV_C_Point3D ptPt1,PV_C_Point3D ptPt2);

// Function to calculate the distance of a pt to the plane
// represented by a triangle
PV_T_GRID_FP DistToTriangle( PV_C_Point3D *oTriangleCoord, PV_C_Point3D oPoint );

// Function to return nearest point on a triangle corresponding to a point in space
PV_C_Point3D NearestPtToTriangle( PV_C_Point3D *oTriangleCoord, PV_C_Point3D oPoint );

// Function to find the closest point on a set of line segments to a point of
// interest
PV_C_Point3D PointToLineSegments(PV_C_Point3D* poStartPoints,PV_C_Point3D* poEndPoints,
                                int iSegmentNum,PV_C_Point3D
poInterestPoint);

// Find minimum value in a vector
int IndexOfMinimumValue(double* pdArray,int iNum);

```

Appendix – A-6 – Journal and conference papers

Journal Papers

1. O. O. Ogundana, C. R. Coggrave, R. L. Burguete and J. M. Huntley. "Fast Hough Transform for automated detection of spheres in three-dimensional point clouds", *Optical Engineering* **46**, 051002 (2007).
2. O. O. Ogundana, C. R. Coggrave, R. L. Burguete and J. M. Huntley. "Automated detection of planes using a fast Hough Transform " (manuscript being prepared).

Conference papers

1. O. O. Ogundana, C. R. Coggrave, R. L. Burguete and J. M. Huntley. "Fast three-dimensional Hough transform for automated calibration of multiple 3-D sensors", *FASIG, Photon 06*, University of Manchester, September 2006.
2. J. M. Huntley, T. Ogundana, R. L. Burguete and C. R. Coggrave. "Large-scale full-field metrology using projected fringes: some challenges and solutions", *Proc. SPIE* **6616**, 66162C (2007).

