INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

Evaluation of Camera Calibration Performance With a Computer Based Camera Model

by

Andrew M. Wykurz

A thesis submitted to

The Faculty of Graduate Studies and Research

in partial fulfillment for the degree of

Master of Science in Information and System Science

Department of Systems and Computer Engineering

Carleton University

Ottawa, Ontario

January 10th, 2002

Copyright 2002.

Andrew M. Wykurz

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file *Votre référence*

Our file *Notre référence*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-68330-3

Canada

The undersigned recommends to
The Faculty of Graduate Studies and Research
acceptance of the thesis

**Evaluation of Camera Calibration Performance
with a Computer Based Camera Model**

submitted by

Andrew M. Wykurz, B Eng.

in partial fulfillment of

the requirements for the degree of

Master of Science in Information and System Science

Thesis Supervisor

Chairman, Department of Systems and Computer Engineering

Carleton University

January 10th, 2002

Abstract

Cameras are employed over an increasing range of fields as an instrument of measure. Camera calibration methods are the processes that provide the accurate mathematical relationship between the image and the world by estimating camera parameters. The measure of accuracy for this process is the difference between world points and back projected points.

MatCam is developed as a computer based mathematical abstraction of a broad range of camera models to accurately generate synthetic data as input to camera calibration methods. MatCam is employed to evaluate calibration methods by comparing estimated to known camera parameters and the mean error between world points and back projected points.

Tsai's non-coplanar calibration method is studied as an example of the value provided by MatCam in assessing the accuracy of parameter estimation and parameter reaction to effects of error sources. MatCam accurately generated synthetic input data and provided an error tolerance for the measurement of world points.

Acknowledgements

First, I would like to thank my wife, Cindy, and our three children, Mark, Laura and Carly, for their patience, understanding and support. Without it, my dream of a higher education at this point in our lives would not have been achievable. Also, without them, my desire to take this path would never have existed.

I would like to thank my supervisor, Dr. V. C. Aitken, who has been both encouraging and patient, and whose suggestions and demands kept me focused and challenged throughout my research and writing of this thesis. His passing of knowledge and true caring for my successful completion of the program is graciously appreciated. I would also like to thank Dr. H. M. Schwartz who provided academic counseling and guidance throughout my academic program. I would also like to thank the numerous students with whom I interacted throughout this program who were both informative and assisting in the completion of many courses.

I gratefully acknowledge financial assistance provided by the Department of National Defence in support of my academic program.

Contents

1	Introduction.....	1
1.1	Development of a Calibration Method.....	4
1.2	Thesis Statement	5
1.3	Thesis Overview	6
2	Camera Model and Calibration Methods	8
2.1	Introduction.....	8
2.2	Review of Camera Calibration Methods.....	9
2.3	Terminology and Notation	11
2.4	The Camera Model.....	13
2.5	What Does Camera Calibration Involve	15
2.5.1	Extrinsic Parameters.....	16
2.5.2	Intrinsic Parameters.....	18
2.5.3	Effective Focal Length (f).....	21
2.5.4	Horizontal Scale Factor (s_x).....	21
2.6	Distortion Components	22

2.6.1	Radial Distortion	24
2.6.2	De-Centering/Tangential Distortion.....	27
2.7	Sources of Error in Calibration	30
2.7.1	Image Resolution Noise	31
2.7.2	Digitization Error	34
2.7.3	Measurement Error.....	34
3	Tsai's Method	35
3.1	Introduction.....	35
3.2	Geometry.....	35
3.3	Tsai's Camera Model	39
3.4	Back Projection	43
3.5	Parameter Estimation	46
3.6	Evaluation	53
4	MatCam	55
4.1	Purpose of MatCam	55
4.2	Overview of MatCam.....	56
4.2.1	Input Data Generator.....	58
4.2.2	Details of Implementing MatCam.....	59
4.2.3	Inverse Perspective Projection Unit	73
4.3	Synthetic Data Examples	76

5	Evaluation of Tsai’s Calibration Method With MatCam.....	86
5.1	Introduction.....	86
5.2	Radial Distortion Tests.....	89
5.3	De-centering Distortion.....	95
5.4	Measurement Error in World Points and the Effect on Camera Parameters	114
5.5	Resolution Error in Image Point Positions and the Effect on Camera Parameters	133
5.6	Actual Camera Calibration Using Tsai’s Method.....	144
5.7	Chapter Summary.....	149
6	Conclusions and Proposals For Future Work	151
	References	155
	Appendices	
A	Improvement Methods.....	160
B	Software Code of MatCam	163

List of Tables

Table 2.1 Terminology.....	11
Table 2.2 Notation definitions.....	12
Table 4.1 Summary of point set errors.....	75
Table 4.2 Example of the error data available from MatCam perspective projection unit.	75
Table 5.1 Camera parameter used to generate input data for each study.....	88
Table 5.2 Summary of de-centering distortion study results.	97
Table 5.3 Summary of measurement error results.	117
Table 5.4 Legend for Figure 5.48.....	129
Table 5.5 Summary of resolution error results.....	135
Table 5.6 Legend for Figure 5.63.....	141
Table 5.7 Summary of actual calibration trials.	148

List of Figures

Figure 1.1 Camera calibration process.....	3
Figure 2.1 Pinhole camera model.....	13
Figure 2.2 Perspective projection.....	14
Figure 2.3 Reference frames.	17
Figure 2.4 Radial distortion about the optical axis.....	24
Figure 2.5 Components of radial distortion.	25
Figure 2.6 Pin-cushion and barrel distortion.....	26
Figure 2.7 De-centering distortion.	28
Figure 2.8 Tangential distortion geometry.....	29
Figure 2.9 Image of a calibration grid used in a camera calibration experiment.	32
Figure 2.10 Bottom left corner of the image.....	33
Figure 2.11 Enlarged image of a single calibration point.	33
Figure 2.12 Enlarged image point on back edge of calibration grid.	33
Figure 3.1 Camera geometry.....	36
Figure 3.2 Coplanar vs. non-coplanar.	38
Figure 3.3 Non-coplanar calibration jigs.....	39
Figure 3.4 Radial alignment constraint.	40
Figure 3.5 Four steps of Tsai's method.	41
Figure 3.6 Inverse perspective projection.	44

Figure 4.1 MatCam model.	57
Figure 4.2 Square calibration jig.	60
Figure 4.3 Spoke calibration jig.	62
Figure 4.4 World points transformed to F_c	63
Figure 4.5 Image points in camera reference frame F_c	64
Figure 4.6 Magnitude of error between the known undistorted points and calculated undistorted points using the synthesized distorted points.	66
Figure 4.7 Position of point set undistorted.	67
Figure 4.8 Radial distortion only.....	68
Figure 4.9 De-centering distortion only.	68
Figure 4.10 Radial and de-centering distortion.....	69
Figure 4.11 CCD array pattern of distorted image points.	70
Figure 4.12 CCD array field of view boundary applied to distorted point set.....	71
Figure 4.13 Distorted image point set limited to field of view only.	71
Figure 4.14 Modified image points.	72
Figure 4.15 Example of inverse perspective projection display.	74
Figure 4.16 Example of positive first order radial distortion.....	78
Figure 4.17 Example of positive first and second order radial distortion.	79
Figure 4.18 Example of negative first and second order radial distortion.	79
Figure 4.19 Example of de-centering distortion.....	80
Figure 4.20 Example of first and second order radial and de-centering distortion.....	81
Figure 4.21 Example of severe first and second order radial distortion.	81
Figure 4.22 Example of severe first and second order radial and de-centering distortion.	82
Figure 4.23 Example of offsetting the principal point.	83

Figure 4.24 Example of measurement error.....	84
Figure 4.25 Example of resolution error.	85
Figure 4.26 Example of digitization error.....	85
Figure 5.1 Difference between actual and estimated κ_1	90
Figure 5.2 Difference between actual and estimated κ_2	91
Figure 5.3 Difference between actual and estimated f	92
Figure 5.4 Difference between actual and estimated s_r	92
Figure 5.5 Difference between actual and estimated T	93
Figure 5.6 Rotation matrix convergence.	94
Figure 5.7 Magnitude of actual and estimated mean error.....	95
Figure 5.8 Effect of de-centering distortion on κ_1	99
Figure 5.9 Effect of de-centering distortion on κ_1	99
Figure 5.10 Effect of de-centering distortion on κ_2	100
Figure 5.11 Effect of de-centering distortion on κ_2	101
Figure 5.12 Effect of de-centering distortion on f	102
Figure 5.13 Effect of de-centering distortion on f	103
Figure 5.14 Effect of de-centering distortion on s_r	104
Figure 5.15 Effect of de-centering distortion on s_r	104
Figure 5.16 Effect of de-centering distortion on T_x	105
Figure 5.17 Effect of de-centering distortion on T_x	106
Figure 5.18 Effect of de-centering distortion on T_y	106
Figure 5.19 Effect of de-centering distortion on T_y	107
Figure 5.20 Effect of de-centering distortion on T_z	107

Figure 5.21 Effect of de-centering distortion on T_z	108
Figure 5.22 Effect of de-centering distortion on the magnitude of offset of T	108
Figure 5.23 Effect of de-centering distortion on the magnitude of offset of T	109
Figure 5.24 Components of T and magnitude of camera position offset due to de-centering distortion.	109
Figure 5.25 Effect of de-centering distortion on R	111
Figure 5.26 Effect of de-centering distortion on the mean error using calibration method.....	111
Figure 5.27 Effect of de-centering distortion on the mean error using calibration method.....	112
Figure 5.28 Effect of de-centering distortion on the mean error using inverse perspective projection and input data.....	112
Figure 5.29 Effect of de-centering distortion on the mean error using inverse perspective projection and input data.....	113
Figure 5.30 Comparison of mean error for calibration method and inverse perspective projection of input data when applying de-centering distortion.	113
Figure 5.31 Measurement error in world points and effect on κ_1	118
Figure 5.32 Measurement error in world points and effect on κ_1	118
Figure 5.33 Measurement error in world points and effect on κ_2	119
Figure 5.34 Measurement error in world points and effect on κ_2	120
Figure 5.35 Measurement error in world points and effect on f	121
Figure 5.36 Measurement error in world points and effect on f	121
Figure 5.37 Measurement error in world points and effect on s_x	122

Figure 5.38 Measurement error in world points and effect on s_x .	123
Figure 5.39 Measurement error in world points and effect on T_x .	124
Figure 5.40 Measurement error in world points and effect on T_x .	125
Figure 5.41 Measurement error in world points and effect on T_y .	125
Figure 5.42 Measurement error in world points and effect on T_y .	126
Figure 5.43 Measurement error in world points and effect on T_z .	126
Figure 5.44 Measurement error in world points and effect on T_z .	127
Figure 5.45 Measurement error in world points and effect on the magnitude of error in T .	127
Figure 5.46 Measurement error in world points and effect on the magnitude of error in T .	128
Figure 5.47 Comparison of error in components of T and offset from actual T .	128
Figure 5.48 Tracking camera position movements due to measurement error in the world points.....	129
Figure 5.49 Measurement error in world points and effect on R .	130
Figure 5.50 Effect of measurement error in the world points on the mean error using calibration method.....	131
Figure 5.51 Effect of measurement error in the world points on the mean error using inverse perspective projection.....	131
Figure 5.52 Comparison of mean error between calibration method and IPP of input data.	132
Figure 5.53 Image of a calibration grid used in a camera calibration experiment.....	134
Figure 5.54 Resolution error in image points and effect on κ_1 .	136
Figure 5.55 Resolution error in image points and effect on κ_2 .	137

Figure 5.56 Resolution error in image points and effect on f .	137
Figure 5.57 Resolution error in image points and effect on s_x .	138
Figure 5.58 Resolution error in image points and effect on T_x .	139
Figure 5.59 Resolution error in image points and effect on T_y .	139
Figure 5.60 Resolution error in image points and effect on T_z .	140
Figure 5.61 Resolution error in image points and effect on T .	140
Figure 5.62 Resolution error in image points and effect on the magnitude of error in T .	141
Figure 5.63 Tracking camera position movements.	141
Figure 5.64 Resolution error in image points and effect on the magnitude of error in R .	142
Figure 5.65 Effect of resolution error in the image points on the mean error.	143
Figure 5.66 View of camera and calibration jig.	145
Figure 5.67 Calibration jig with elevation blocks.	146
Figure 5.68 Box camera on a tripod.	147

Chapter

1 Introduction

Cameras are employed over a vast range of fields: aerial survey for map production; observation/navigation systems on remotely piloted vehicles; biomedical imaging systems; and industrial applications such as automated tolerance testing. These applications are but a few of a growing field. The use of a camera as an instrument of measure is securely grounded in the science of photogrammetry, defined [1] as:

Photogrammetry is the art, science, and technology of obtaining reliable information about physical objects and the environment through the processes of recording, measuring, and interpreting photographic images and patterns of electromagnetic radiant energy and other phenomena.

Physical objects and the environment are three-dimensional (3D) and images are two-dimensional (2D). In order to retrieve reliable information of the world from the image, the relationship between the two must be established and it is the camera model that provides this connection.

The camera model is a mathematical abstraction of a camera that provides the relationship between the world and the image formation process. The model is based on a camera and the position from which the camera is located in the world. In its fundamental

form, the camera is a lens and something to capture an image on, either film, digital array or other image display device. The image is the result of two factors, how light rays pass through the interior of the camera to get to the image capturing device, and where the camera is positioned in the world and what it can see. The camera model mathematically provides the relationship to retrieve reliable information between world and image.

Parameters are used to define the camera model: *intrinsic parameters* assist in defining the internal workings of the camera and *extrinsic parameters* define the camera position and orientation in the world. The intrinsic parameters deal with how light rays are transferred through the lens and where the rays are positioned on the image plane. The extrinsic parameters deal with the position and orientation of the camera in relation to the world. Both intrinsic and extrinsic parameters are important when using a camera as an instrument of measure and have a direct effect on the image. The values of the parameters are a combination of known and estimated. Known parameters are values provided through manufacturer's specifications and estimated parameters are those which can not be directly measured, but require an estimation of their values based on the current disposition of the camera. The estimation of camera parameters is accomplished by "camera calibration".

Camera calibration encompasses the estimation of parameters which allows one to relate information to the 3D world. In general terms, the camera calibration process involves, (see Figure 1.1):

- Accurately measuring calibration points in the world (world points).
- Capturing an image of the world points (image points).

- Determining the relationship between world points and image points (estimation of parameters).
- Using the estimated parameters to back project the image points into the world (back projection).
- Determining the difference between back projected points and world points (error analysis).

Figure 1.1 Camera calibration process.

The aim of calibration methods is to minimize the difference between world point coordinates and back projected point coordinates. In the camera calibration methods reviewed [2] [3] [10] [12] [16] [17] [21] the use of the comparison between world points and back projected points was the standard for error assessment. In [2] it was stated that it was difficult to obtain an accurate assessment of the calibration parameters to use as a

reference for error analysis. There has been a lack of studies that assess how well the calibration method estimates the actual parameters of the camera.

Research into the comparison of actual parameter values of a camera system with the estimated parameter values is lacking in that no published documentation was discovered. It is difficult to gauge the accuracy of estimated parameter values as there does not exist a tool to measure the actual value of each parameter. This leaves the assessment of accuracy of a calibration method to the comparison between the world points and back projected points. Although this standard of assessment supports the aim of camera calibration and has been accepted, the ability to assess how accurately the calibration method estimates the actual parameter values would provide insight to the strengths and weaknesses of the calibration method. If there was a means to precisely know the parameter values and have the ability to control these values, parametric studies could be conducted to better understand the calibration method, design, and performance.

1.1 Development of a Calibration Method

Photogrammetric cameras can be divided into two broad levels, highly accurate and off-the-shelf. The differences are in relative accuracy. "Highly accurate" refers to a camera built to exacting standards with a lower tolerance of error. Highly accurate cameras are synonymous with aerial survey and usually imply expensive equipment and extensive calibration in a laboratory environment. The "off-the-shelf" camera refers to lower cost equipment and calibration can be conducted on site by an operator of the equipment. Unless otherwise specified, from this point forward, any reference to a camera or camera calibration method is inherent to "off-the-shelf" cameras.

A major researcher and developer of camera calibration for off-the-shelf cameras is Roger Y. Tsai. In Tsai's work [2] five criteria for camera calibration were presented, four of these have remained relatively consistent throughout the majority of methods proposed to date: Accuracy, Efficiency, Versatility, and off-the-shelf cameras. The other criterion is autonomous where "the calibration procedure should not require operator intervention such as giving initial guesses for certain parameters, or choosing certain system parameters manually." This is a rather limiting criteria, especially when considering that some methods, with a good initial guess can be the difference between closing on a very accurate solution or no solution at all. Tsai's work [2] has been an unofficially accepted benchmark for the camera calibration community and has been a catalyst for future research.

Much research has been conducted for both single and stereo camera calibration and for charge coupled device (CCD) array cameras. In many cases, techniques for stereo calibration are an extension of single camera techniques.

1.2 Thesis Statement

Camera calibration methods involve the estimation of parameter values that provide the ability to accurately infer information of the world from an image. The actual parameter values and their effect on the calibration method remains uncertain. The development of a flexible computer based camera model, with known parameter values, that can interface with a calibration method to study the effects of distortion, noise, measurement error, resolution error and digitization error on the estimated parameter values can assist in the development of calibration methods. This flexible computer based camera model is MatCam developed by the author under the work of this thesis. With the use of MatCam,

parametric studies of calibration performance can be conducted to examine how the method reacts to a wider range of circumstances and situations than would normally arise in practice. MatCam generates highly controllable data for input into a calibration method and hence is a valuable tool in the assessment of calibration methods.

This thesis focuses on MatCam, developed to integrate with Tsai's non-coplanar calibration method, and used to study parameter estimation in controlled situations.

1.3 Thesis Overview

It is acknowledged that the difference between world points and back projected points, within the scope of "off-the-shelf" camera systems, is normally the overriding criteria for accuracy assessment. However, there exists a need to accurately assess the validity of calibration methods by standards other than this difference. The evaluation of a camera calibration method in a highly controlled simulation provides valuable insight into the estimated parameter values when distortion and sources of error are applied to the image and world points. This thesis focuses on the development of MatCam, an analysis tool to analyze the accuracy of estimated parameter values when controlled input data is injected into a calibration method.

The remainder of this thesis is organized as follows:

Chapter 2 provides the background to a camera model and the parameters that are estimated. A review of the camera model, including the perspective projection or pinhole camera, is presented as the basis for calibration methods. The geometry of perspective projection remains as a foundation in practically all calibration methods. A brief review of some of the more common calibration methods is presented and emphasis will be on what most methods require as data to conduct a calibration of a camera.

Chapter 3 discusses Tsai's non-coplanar radial alignment constraint method in detail. The method is well understood by the camera calibration community and provides a common basis with many other methods. It is then possible to draw inferences with other calibration methods that could interface with MatCam.

Chapter 4 explains the development and capabilities of MatCam and how it accurately represents a camera model with exactly known parameters. The camera model used to develop MatCam is an integration of a broad range of camera models as discovered in the literature. The capabilities of MatCam are presented and confirmed with simulated trials.

Chapter 5 consists of parametric studies conducted using MatCam to generate input data to Tsai's non-coplanar calibration method. The studies emphasize how the estimated parameter values vary from the actual values based upon controlled distortion and error inputs. The error between world points and back projected points is also analyzed. Results of estimated parameters from an actual calibration experiment using Tsai's method are included as well.

Chapter 6 provides conclusions, general discussion and proposals for future research.

Chapter

2 Camera Model and Calibration Methods

2.1 Introduction

The camera model is the basis of each calibration method. The model is the mathematical abstraction of a camera and provides the connection between the 2D image and the 3D world. The camera model is comprised of parameters: intrinsic parameters refer to a mathematical abstraction of the internal components of a camera, and extrinsic parameters refer to the mathematical abstraction of components external to the camera. Each calibration method must define a model, hold the model fixed and then estimate the parameters. How a calibration method defines the camera model will effect how parameters are estimated. A brief review of camera calibration methods is included in this chapter.

The basis of the camera model is the ideal camera, the pinhole camera. The pinhole camera is a perspective projection of world points to image points. A review of perspective projection is included in this chapter.

Also included in this chapter is a section that discusses sources of error in calibration. Errors can enter the calibration process due to human error and limitations in equipment, or a combination of both.

2.2 Review of Camera Calibration Methods

The calibration of cameras has been a topic of research prior to the development of digital cameras [20]. The development of digital video equipment and incorporating the use of computer systems, has broadened the number of methods used for calibrating a camera.

In 1987, Tsai [2] published a paper that re-energized the camera calibration community with a calibration method that used a radial alignment constraint as the cornerstone element of how he defined the camera model. At that point, research into camera calibration methods intensified and it continues to be a topic of significant research today.

A camera calibration method is the means by which reliable information of the world may be retrieved from an image. Camera calibration involves the determination of internal (intrinsic) and external (extrinsic) parameters which when combined can relate image points to world points. The definition of each parameter and where it fits into a camera model is discussed in Sections 2.5-2.6. Prior to reading the sections mentioned, the concept of distortion must be introduced as it is distortion and how it is dealt with that affects many of the camera models. A problem with optical lenses is that they all have distortion and it is the degree of distortion that must be estimated. Distortion is the changing of shape, which in camera models, is an intrinsic parameter. To account for distortion in a camera model normally requires a nonlinear function.

Perspective projection is at the base of all camera models and is easily understood as the pinhole camera. Without distortion it would be possible to draw a straight line from the world point through the pinhole to the image point. However, a camera does have distortion and therefore that is not what actually happens, but the use of the pinhole camera is a way of defining a simple camera model.

The Direct Linear Transformation (DLT) and modified DLT calibration methods define the camera model as a perspective projection where the “modified” method accounts for distortion “image refinement parameters” [3] [24]. The error between world and back projected points is too great to make the DLT a viable calibration method, except in a camera that exhibits very little distortion. As is expected, the modified DLT provides better accuracy in back projected points than the DLT. The strength of each method is the use of a linear model for the majority of the camera model which makes the mathematical abstraction easier to implement. When adding a nonlinear function to a camera model, an optimization of an objective function is required [29]. A drawback of nonlinear functions is that the determination of the global optimum cannot be guaranteed.

Full scale non-linear methods [9] estimate the parameter values such that the search pattern for the optimum parameter value is not limited to linear functions and therefore the estimated values can usually be finer “tuned”. They are usually more accurate than linear methods, however, they require good initial “guesses” to initiate the search and these methods are computer intensive. Non-linear search methods cannot guarantee a global optimum solution, and the selection of a good initial guess will greatly enhance the possibility of determining the optimum solution. Non-linear optimizations are usually quite computer intensive. The processing time to accurately determine the

calibration parameters is not normally an issue and any technique that accurately estimates the calibration parameters is useful.

Single stage and multi stage methods refer to how the parameters are estimated, either all at once or in successive stages. Several common calibration methods are multi-stage [2] [10] [12] [17]. In the majority of these methods, the initial stage is a linear estimation of the parameters, then hold the initial set of estimated parameters fixed, and estimate the second set of parameters. If there are more than two stages, then the process continues, holding the previously estimated parameters fixed and estimating the next set of parameters [10] [12]. Single stage methods, although mentioned, are not widespread. Single stage methods attempt to estimate all the parameters at once [4].

There exists a group of methods for stereo calibration (multiple cameras viewing the same scene from varying positions) that are similar to single camera calibration methods and are mentioned, but these are not the focus of this thesis and will not be pursued further.

2.3 Terminology and Notation

A clear understanding of the terminology and notation used throughout this thesis is essential. A brief summary of terms is provided at Table 2.1.

Table 2.1 Terminology.

World point	3D point in the world with accurately known coordinates
Image point	2D point in an image plane
Parameter	An estimated value
Back projected	3D point in the world that was calculated by parameters
Error analysis	Comparison of world point to back projected point
Input data	Image points that have been modified by distortion
Operator	A person that is required to use the calibration method and perform specific actions in the calibration process

A review of the literature uncovered a substantial variety of notation and therefore necessitated Table 2.2. The table also introduces parameter names which will be explained in the following sections of this chapter.

Table 2.2 Notation definitions.

Extrinsic	
F_{α}	Reference frame α
X, Y, Z	Elements of points in 3D space. If used singularly or in pairs, then represent an axis or plane
$[X_{\alpha}, Y_{\alpha}, Z_{\alpha}]^T$	3D point in a reference frame α , represented as a column vector
T	Generic translation vector
T_{β}^{α}	Translation vector from reference frame α to β expressed in F_{β}
R	Generic rotation matrix
R_{β}^{α}	Rotation matrix from reference frame α to β
Intrinsic	
f	Effective focal length
x, y	Elements of points in 2D space
$[x, y]^T$	Image point in generic 2D coordinates
$[x_u, y_u]^T$	x and y coordinates of an undistorted image point
$[x_d, y_d]^T$	x and y coordinates of an distorted image point
$[x_f, y_f]^T$	x and y coordinates of an image point in computer frame memory
C_x, C_y	Row and column number of the centre of computer frame memory
κ_1	First order radial distortion coefficient
κ_2	Second order radial distortion coefficient
κ_3, κ_5	De-centering distortion coefficients
d_x	Centre to centre distance between adjacent sensor elements in the CCD array in the X direction
d_y	Centre to centre distance between adjacent sensor elements in the CCD array in the Y direction
N_{cx}	Number of sensor elements of the CCD array in the X direction
N_{fx}	Number of pixels in an image line sampled by the computer
s_x	Horizontal image scale factor

2.4 The Camera Model

The camera model is a mathematical abstraction of a camera. In its most basic form, the camera model is a perspective projection with no lens distortion and this remains as the basis of the majority of calibration methods [1] [2] [4] [9] [16] [24] [28]. Perspective projection uses mathematical relations as illustrated in Figure 2.1 and from the ratio of dimensions:

$$\frac{a}{h} = \frac{A}{H} \quad (2.1)$$

Camera calibration methods have drawn on perspective projection and linear algebra [29] to create undistorted linear transformations from different planes as illustrated in Figure 2.2. Perspective projection and linear transformations are the basis of the Direct Linear Transformation (DLT) calibration method [3].

Figure 2.1 Pinhole camera model.

From Figure 2.2 with X as the axis providing depth, image point p_i can be determined by:

$$\frac{Y}{X} = \frac{y}{f} \quad (2.2)$$

$$\frac{Z}{X} = \frac{y}{f} \quad (2.3)$$

Figure 2.2 Perspective projection.

In Figure 2.2, the image plane is a distance f away from the origin and normal to the X axis. The object plane is defined in a 3D world coordinate system F_w . Solutions to the equations above are possible because the points in the object plane are known and the values of x and y can be identified by an operator. Using the equations above, the perspective projection of X, Y, Z onto the image plane defines p_i when f is known. The inverse perspective projection (back projection) is not as straight forward. To determine X, Y, Z when x, y and f are known, the equation of the object plane must be known relative to the camera and a ray from the origin of the camera reference frame, O_c .

through p_i can be projected. The intersection of the ray and the object plane is determined. This leads to a critical assumption that the equation of the object plane can be determined. This describes a monocular camera where there is a loss of depth and therefore an assumption with regards to depth and the object plane is required. More on this in Section 3.2.

2.5 What Does Camera Calibration Involve

In general, the camera calibration process involves:

- A camera model definition (see Section 2.4).
- Selection of world points.
- Capturing an image of the world points.
- Estimating the calibration parameters.
- Back projection.
- Error analysis.

The selection of world points involves accurately measuring the 3D coordinates of an object that may be viewed in a 2D image plane. A clarification with regards to image points is required at this time. Image points can be defined as undistorted and distorted. Undistorted image points will be explained shortly. Distorted image points are the image points that are captured by the camera. The estimation of the parameters requires the definition and selection of a camera model. A mathematical representation of the camera model is created. Once the parameters are estimated, then undistorted image points are calculated. Undistorted image points are 2D points that would exist if there was no

distortion in the camera. Undistorted image points are used with inverse perspective projection to produce back projected points.

In order for the back projection to occur, the position and orientation of the camera, relative to the world reference frame, is required. The position and orientation, of the camera reference frame, are estimated as part of the parameter estimation. In the back projection, the undistorted image points cannot be directly related to the world points. The undistorted image points can be related to the camera reference frame, see Figure 2.3. The undistorted image points are back projected in the camera reference frame and then a linear transformation is executed to get the back projected point in the world reference frame.

This is accomplished by estimating the intrinsic and extrinsic parameters. Intrinsic parameters involve the distorted to undistorted image point mapping and extrinsic parameters involve estimating the rotation and translation that relates the camera reference frame to the world reference frame [16]. Details of the extrinsic and intrinsic parameters will be expanded upon in the following sections.

2.5.1 Extrinsic Parameters

Within the camera calibration community, the extrinsic parameters are generally agreed upon to consist of six degrees of freedom: three parameters represent the rotation/orientation (or referred to as the pose) of the camera and the remaining three parameters comprise the translation vector, the vector that relates the origins of coordinate reference frames. Procedures for the estimation of extrinsic parameters may vary with each calibration method and depend upon the definition of the camera model. The knowledge of extrinsic parameters is necessary to linearly transform information from one reference

frame to another. Linear transformations are discussed in this thesis with detailed information available in [29].

In general, there are three common methods to deal with rotation: Euler angles, quaternions [26], and the roll, pitch and yaw angles. Each has its merits and weaknesses [13], depending upon the intended method of use. Details of use will not be expanded upon in this work.

Figure 2.3 Reference frames.

The translation vector is a column vector consisting of a translation in the X , Y and Z axes respectively from one reference frame F_a to the origin of a second reference frame F_b . The origin of each reference frame can be expressed in at least two if not more coordinate systems as per Figure 2.3, for example O_j in F_j , O_j in F_c and O_j in F_w is the same point expressed in three different reference frames. It is essential that standardized conventions be employed throughout any technique. This implies that all reference frames be designated either right-handed or left-handed systems and that the convention be followed throughout. In many of the systems, the right-hand axis for the world reference frame, F_w , is defined with positive directions as X axis right, Y axis forward and Z axis up. The camera reference frame, F_c , is usually defined with the Z axis coincident with the optical axis as in [1] [9] [12] [16] [21] but can also be defined with the X axis coincident with the optical axis as in [13] [31] [40] [41]. The first method tends to utilize the strength that the image plane coordinates can continue to be referred to as x and y without confusing the reader. If the Z axis, in F_c is positive down, X and Y will be positive forward and right respectively. Figure 2.3 has the X axis of F_c as positive forward, the Y axis positive left and Z axis positive up which is maintained throughout this thesis.

2.5.2 Intrinsic Parameters

The selection and definition of the intrinsic parameters are varied. Paramount to the use of the intrinsic parameters is the decision as to which parameters are estimated and those that are not. The manufacturer of the camera and/or ancillary equipment (i.e. CCD array and computer frame memory) can define a non-estimated parameter and its value can be

accepted as correct for the calibration method. An example of this would be the number of sensor elements in the x scan direction of the CCD array. Another non-estimated parameter is one that is selected based upon a judgement such as the centre of the array, the principal point. Of the various methods reviewed, the following parameters were generally estimated components:

- Effective focal length. See Section 2.5.3.
- Horizontal scale factor. See Section 2.5.4.
- Distortion coefficient(s). See Section 2.6.

Non-estimated or user defined intrinsic components have particular characteristics as well. This category of parameter, although normally included as known, can have an effect on the accuracy of the calibration method. This group of parameters includes, but should not be limited to, the:

- Principal point selection for the array;
- Pixel spacing distance in the CCD array;
- Number of elements in the x direction scan line for the CCD array; and
the
- Number of pixel elements in the computer image frame memory.

The centre or principal point of memory for the array (C_x, C_y) is a parameter of interest [37]. The principal point, or image centre, is considered to be the point of intersection of the optical axis with the image/sensor plane. There are no less than 15 methods for determining the centre of the image [7] ranging from non-imaging, single image and multi-image techniques. Each method for determining the principal point is

explained in [7] and a summary of principal point selection methods is provided below. The simplest method of the non-imaging methods is the numerical centre of the image. Single image techniques are subdivided into feature and non-feature based. One single image feature based technique utilizes the centre of vanishing points to determine the optical centre [8]. A non-feature based single image technique uses an autocollimated laser [30]. Multi-image techniques are feature based and divided into centre of expansion and focus of expansion methods. Overall, each method provided a different value for the principal point location, therefore the selection of which centre to use is related to the error tolerances of the application algorithm. Of the calibration methods reviewed in the literature, the centre of the array coordinates was the preferred choice for principal point location.

Manufactures of digital equipment will provide technical data as to the size of the sensor array and number of pixel elements in both the horizontal (dx) and vertical (dy) direction. With these values known, the pixel spacing in each direction for the CCD array is calculated. This value is normally accepted as accurate and was not the focus of any literature reviewed.

The number of elements in the x direction scan line for the CCD array N_{cx} and the number of pixel elements in the computer image frame memory N_{fx} have a specific relationship [34]. A camera array will have a specific number of pixels that will normally not match the resolution of the computer imaging frame memory (buffer) and that difference must be accounted for. If the resolution of the CCD array does not match the resolution of the computer frame memory, then the ratio of the two is determined. This provides the relational scaling between resolutions that is required. The selection of the

value for N_{fx} can also be modified if black lines appear on the edge of the image indicating a discrepancy in how the analog to digital conversion was accomplished for the CCD array. The number of black lines observed can reduce the value of N_{fx} . An example of this can be observed along the bottom edge of Figure 2.9 on page 32. In the calibration methods reviewed, N_{fx} and N_{cx} were accepted as accurate. Both N_{fx} and N_{cx} are discussed further in section 2.5.4.

2.5.3 Effective Focal Length (f)

The effective focal length (f) of a camera is the distance from the principal point of the lens to the focal point [35]. In terms of perspective projection, f is the distance along the optical axis from the point of focus to the image plane such that the optical axis is normal to the image plane. Knowledge of f and $[x, y]^T$, combined with an equation for the object plane to provide depth, is sufficient to determine the 3D point in the camera reference frame. The transformation from camera to world reference frame was discussed in Section 2.5.1.

2.5.4 Horizontal Scale Factor (s_x)

The requirement to include a horizontal scale factor as an estimated intrinsic parameter resides with the difference in scanning frequencies for the camera sensor and that of the computer frame memory [36]. The approximate scale factor is the ratio between the number of sensor elements in a row of the sensor array and the number of picture elements in a row of the computer image frame memory. The approximate scale factor is determined by

$$\tilde{s}_x = \frac{N_{cx}}{N_{fx}} \quad (2.4)$$

The correct value of s_x is related to the frequency of sampling of the CCD array in the camera and the frequency of sampling of the analog to digital (A/D) convertor

$$s_x = \frac{f_c}{f_f} \quad (2.5)$$

where

f_c is the clock frequency of the camera

f_f is the sampling frequency of the A/D convertor

As these frequencies are difficult to establish with accuracy, initially $\tilde{s}_x$ is used as an approximate value of s_x and is estimated as a parameter in the calibration process [30].

2.6 Distortion Components

Every off-the-shelf camera system is subject to measurable distortion and aberrations. Aberrations are optical defects that degrade the quality of the image and are categorized as spherical, chromatic, coma, astigmatism and curvature of field [35]. Distortion is the displacement of an image point from the position predicted by perspective projection [7]. Every lens has an element of distortion [1] [5] [14] [35] and with high-end photogrammetric equipment, the acceptable error tolerances applied are exacting. With mass production lenses (off-the-shelf), the effort and resources used to minimize errors are obviously not the same. All lenses exhibit distortion, regardless of the production process.

There are several types of distortion in a lens and varying options as to which should or should not be considered in the calibration process. Four types that cover the spectrum [32] are: radial, de-centering, out-of-plane unflatness influences and in-plane image distortion. The Manual of Photogrammetry [1] only considers radial and tangential distortion. Tsai's camera model [2] considers only radial distortion. Radial distortion is observed as either barrel or pin-cushion and is further discussed in Section 2.6.1. (Figure 2.6). De-centering distortion occurs when the lens components are mis-aligned which causes both radial and tangential distortion. De-centering distortion, when included in the camera model, includes both radial and tangential distortion components [6][32]. The requirement for the inclusion of tangential distortion in camera models varies. Tsai has stated [2] that tangential distortion need not be considered based upon the relative magnitude of error compared to radial distortion and the accuracy requirements for off-the-shelf camera systems. High accuracy requirements [1] would demand the inclusion of tangential distortion, and this is discussed further in Section 2.6.2. Out-of-plane unflatness effects are associated with CCD array cameras where the thin "wafers" would bow or crinkle and in-plane distortion is due to electronic influence. Both in and out-of plane distortion elements have not been considered further as the order of magnitude of these effects are insignificant [32].

Another form of distortion is called "interior orientation instability" that can attributed to the movement of the CCD array sensor with respect to the camera body. Unfortunately, there is no analytical solution to an unstable array sensor [32].

The modelling of distortion adds a non-linear component to the calibration process which will be discussed in the following sections on radial and de-centering distortion.

2.6.1 Radial Distortion

Radial distortion is symmetric about the principal point and occurs due to the refraction of light rays. Light rays traveling along the optical axis and passing through the optical centre of the lens are not distorted as the angle of incidence with the lens is zero. All other rays strike the lens at an angle and are refracted, see Figure 2.4.

Figure 2.4 Radial distortion about the optical axis.

The point P , distorted position, is the actual intersection of the light ray and the image plane and P' is the point of intersection if the light ray would have intersected the image plane with no distortion present. The angle α is the angle the distorted ray makes with the optical axis. The radius r is the radial distance from the principal point to the

undistorted image point and $r - \Delta r$ is the distorted distance. The relationship can be expressed as

$$\Delta r = r - f \tan \alpha \quad (2.6)$$

In an image plane, Δr can be represented by Δx and Δy as in Figure 2.5.

Figure 2.5 Components of radial distortion.

When the distortion coefficients, defined later in this thesis, are positive, barrel distortion results. Negative coefficients give pin-cushion distortion, as shown in Figure 2.6.

An expression for the radial distortion is approximated [32] by an odd ordered polynomial symmetric about the principal point.

$$\Delta r_r = \kappa_1 r^3 + \kappa_2 r^5 + \kappa_3 r^7 + \dots \quad (2.7)$$

Figure 2.6 Pin-cushion and barrel distortion.

The values of the κ_i coefficients in equation (2.7) define the shape of the non-linear function and must be estimated. The coefficients can be obtained through a least squares curve fitting computation and would be considered in terms of Δx_r (the change in x due to radial distortion) and Δy_r (the change in y due to radial distortion) where

$$\frac{\Delta x_r}{r} = \frac{\Delta x_r}{x_d} = \frac{\Delta y_r}{y_d} \quad (2.8)$$

and

$$\Delta x_r = x_d (1 + \kappa_1 r^2 + \kappa_2 r^4 + \dots) \quad (2.9)$$

$$\Delta y_r = y_d (1 + \kappa_1 r^2 + \kappa_2 r^4 + \dots) \quad (2.10)$$

The value of r , the radial distance from the principal point is determined by

$$r^2 = x_d^2 + y_d^2 \quad (2.11)$$

where $[x_d, y_d]^T$ is the distorted image point (the image point that is entered by an operator). In both [2] and [32] the estimation of κ_1 is considered sufficient to meet the error tolerance of the majority of off-the-shelf camera calibration methods. The inclusion of κ_2 and higher terms is only considered warranted when working with wide-angle lenses or for higher accuracy applications [1] [32].

2.6.2 De-Centering/Tangential Distortion

De-centering distortion is intertwined with both radial and tangential distortion and is symmetrical about an axis of zero distortion where all distortion is mirrored through that axis as indicated in Figure 2.7. De-centering distortion displaces an image along and perpendicular to the radial lines due to the imperfect centering of the lens and can be equated to adding a small prism to the lens system. De-centering distortion is normally caused by the improper installation of a lens during the assembly of a camera. De-centering distortion can be expressed as an equation in accordance with [6] with Δx_d (the change in x due to de-centering distortion) and Δy_d (the change in y due to de-centering distortion) as

$$\begin{aligned}\Delta x_d &= \left[P_1 (r^2 + 2x_d^2) + 2P_2 x_d y_d \right] \left[1 + P_3 r^2 + P_4 r^4 + \dots \right] \\ \Delta y_d &= \left[2P_1 x_d y_d + P_2 (r^2 + 2y_d^2) \right] \left[1 + P_3 r^2 + P_4 r^4 + \dots \right]\end{aligned}\quad (2.12)$$

where P_1 and P_2 are coefficients of the tangential profile function $P(r)$. The magnitude of this function can be expressed as:

$$P(r) = \left(\sqrt{P_1^2 + P_2^2} \right) r^2 \quad (2.13)$$

and:

$$P_1 = -J_1 \sin \varphi_o \quad (2.14)$$

$$P_2 = J_1 \cos \varphi_o \quad (2.15)$$

$$P_3 = \frac{J_2}{J_1}, \quad P_4 = \frac{J_3}{J_4} \quad (2.16)$$

$J_1, J_2, \dots$ are the coefficients of the profile function of the de-centering distortion and φ_o is the angle subtended by the axis of the maximum tangential distortion with the photo axis [33]. The photo axis is the axis perpendicular to the axis of zero distortion as depicted in Figure 2.7 and Figure 2.8 and is the axis of maximum tangential distortion. Figure 2.7 also illustrates the distortion of image points asymmetrically and perpendicular to radial lines from the centre. The tangential distortion profile is illustrated in Figure 2.8 where the P_i values define the profile function.

Figure 2.7 De-centering distortion.

When including de-centering distortion in a camera model, the maximum offset of an image point in a radial direction is $3P(r)$ and the maximum offset of an image point in a tangential direction is $P(r)$ which, when related to common digital camera array dimensions, leads to an error generally less than $10\text{ }\mu\text{m}$ at the extremities of the array. Tsai [2] and Fraser [32] discount the requirement to include de-centering distortion in their camera models due to the limited mathematical increase in accuracy on the overall error. Moreover, they state that current manufacturing methods do not normally allow the lens to be installed where ϕ_0 is considered significant [14] [15]. Also considered in the decision to not include higher order coefficients is the mathematical stability of solution techniques when including second or third order radial distortion coefficients.

De-centering distortion involves a tangential and radial distortion component as the two are linked through equation (2.12), and tangential distortion alone is not normally modeled.

Figure 2.8 Tangential distortion geometry.

2.7 Sources of Error in Calibration

In the calibration of camera parameters there remains a human factor, based on technological limitations, which can inject errors. These errors are introduced as image resolution noise, digitization of image coordinates error and measurement error.

Choices must be made by the operator calibrating a camera as to the selection of image points, $[x, y]^T$, that will be used in the calibration process. In some calibration methods, the selection of image points is done algorithmically [12] after an initial input is provided, but in most cases the operator must select the image points. The selection of image points is linked to the resolution of the image and the lens aberrations that could be present which will affect the clarity of the image.

The process of capturing an image of an analog world with a digital system can also introduce error into the calibration process. The digitization of an image divides the image in discrete pixels that is the best interpretation of the analog world. Actual detail of the image is lost in this adjustment into pixels. This single pixel adjustment is considered digitization error.

Measurement error deals with exactly how accurately the world points are known in the world coordinate system. An assumption in most calibration methods is that the world points are known to a high degree of accuracy and this is important. What happens if these points are slightly off and how does the calibration method react?

In the following sections, each error source is briefly discussed along with what can be done to limit the influence of the error and how to model each error source to allow the study of these errors on calibration methods.

2.7.1 Image Resolution Noise

Lens aberrations include spherical, coma, astigmatism and field curvature, distortion and chromatic. The details and effects of each of these are found in [35]. Each of these aberrations will have an effect on the position and clarity of the image. With the exception of distortion, discussed in section 2.6, the other aberrations are not normally modeled in calibration methods. The effect they have is to make the selection of image points difficult by adding “noise” to the area immediately surrounding the image point.

The image produced by a digital camera with a 640x480 CCD array and captured by a 1024x768 computer image frame buffer is similar to Figure 2.9. The camera used was a small “box” camera with a estimated focal length of approximately 6mm. The world points are on a planar surface with each point 10 cm away in the X and Y coordinate directions (x positive right and y positive up in a 2D plane). The selection of the centre of an image point for input to the calibration method is difficult. An enlarged figure of the bottom left portion of Figure 2.9 is included as Figure 2.10. As the image was captured by a 640x480 resolution CCD array, increasing the image size will not improve clarity for image coordinate selection. A choice must be made and therefore noise (resolution limitation) error is introduced. In Figure 2.10, a portion of Figure 2.9 is further enlarged to observe the discrete pixels and how a circle is represented when digitized. In Figure 2.11 where one image point from Figure 2.10 has been further enlarged, the digitization becomes more evident. The selection of the centre of the image point is difficult when image point is further away from the camera (Figure 2.12), and the resolution and clarity degrade. The accurate selection of the centre of these points becomes less likely.

Figure 2.9 Image of a calibration grid used in a camera calibration experiment.

Using as a basis, a 1024×768 image frame memory resolution, with a CCD array of 640×480 and pixel spacing of $1.325 \times 10^{-5} \text{m}$ in the x direction and a focal length of 0.012m . A 20 pixel error for a world point on the back edge of the calibration grid, approximately 2m away from the camera, equates to an error of 2.76mm between the back projected point and world point. Similarly, for a 10 pixel error for a world point 1metre from the camera, the difference between the back projected point and world point is 0.7mm . Regardless of the assumption as to how accurately the centre of the image point is selected, random error is introduced.

Figure 2.10 Bottom left corner of the image.

Figure 2.11 Enlarged image of a single calibration point.

Figure 2.12 Enlarged image point on back edge of calibration grid.

2.7.2 Digitization Error

Digitization error is the difference between the selected point centre and the actual centre. During the digitization of the image, the image point is divided into discrete pixels. The true centre of the image is altered due to the digitization process. This way, the actual centre is not normally a single pixel and this is digitization error. Digitization error is not considered significant for off-the-shelf camera models, but it is present.

2.7.3 Measurement Error

Measurement error is an assessment as to how accurately positions of world points are measured in F_w . A rational assumption is that the world point coordinates are known to a level of accuracy consistent with the future intended use of the camera. Regardless of the measurement method, there will be error introduced. The true effect this type of error has on the estimation of parameters is unknown.

Chapter

3 Tsai's Method

3.1 Introduction

Tsai's camera calibration method consists of a mathematical model of the image formation process and an algorithm for estimating model parameters from experimental data. The difference between world points and back projected points is the approach used to assess accuracy of the overall calibration procedure. The study of Tsai's method is considered essential whenever reviewing or developing any new or modified methods of camera calibration. Using this method as a comparative benchmark is considered necessary for the purposes of this thesis as current calibration methods continue to use Tsai's non-coplanar method in a like manner. Details of the MatLab [39] implementation of Tsai's method are contained in Appendix B as Tsai_cal.m [41].

3.2 Geometry

The development of a calibration method requires a camera model and a clear understanding of the components of the model. This section describes the geometry of the

camera model, in greater detail than provided in Section 2.4, prior to getting into the specifics of Tsai's camera model in Section 3.3.

Figure 3.1 Camera geometry.

Figure 3.1 illustrates a camera positioned on a platform. The world is represented as a flat plane within the field of view (FOV) of the camera. The flat earth assumption is reasonable and necessary, as will be discussed in Section 3.4.

There are two reference frames for the representation of 3D coordinates, the world reference frame (F_w) and the camera reference frame (F_c). The same 3D point can be represented in either reference frame. For notation purposes $[X, Y, Z]^T$ is a point in the world reference frame and $[X_c, Y_c, Z_c]^T$ is a point in the camera reference frame.

Tsai [2] and others [4], [10], [12] to reference a few, use the Z_c axis as the optical axis of the camera reference frame, the X_c axis is positive right and Y_c axis positive down. All notation in this chapter will be in accordance with this frame convention. The

reference frame for all other work in this thesis is consistent with the camera reference frame as illustrated in Figure 3.1.

“Camera point” implies a 3D point in the camera reference frame. The transformation between world to camera reference frames requires a translation and rotation. Reference [29] provides necessary and sufficient background on transformations. To transform 3D points between frames, the point must be translated to the intended reference frame origin and then rotated. A general transformation equation is

$$\begin{bmatrix} X_\beta \\ Y_\beta \\ Z_\beta \end{bmatrix} = R_\beta^\alpha \begin{bmatrix} X_\alpha \\ Y_\alpha \\ Z_\alpha \end{bmatrix} + [T_\alpha^\beta]_\beta \quad (3.1)$$

The translation is a vector from the origin of one reference frame to the origin of the other reference frame. For example, the translation from world to camera origin will require the translation vector to be expressed in camera frame coordinates. In general form, the translation vector is denoted as T . To identify the translation vector from one frame to another, $[T_\beta^\alpha]_\beta$ is the translation from α to β in the β reference frame. T may also be represented by its components T_x , T_y and T_z as in

$$T \equiv \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (3.2)$$

The rotation is necessary when reference frame axes are not parallel. The rotation is accomplished by applying a 3x3 rotation matrix R . The direction of rotation, either between camera to world or world to camera must be consistent. A general transformation equation is

$$\mathbf{R} \equiv \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (3.3)$$

where $r_{1...9}$ are computed from the Euler angles yaw, pitch and roll [13]. More on the specifics of $\mathbf{R}$ in Section 3.3.

Part of the geometry of a camera model is the calibration object defined by world points. The calibration object is any object where all the world points do not lie in a single plane, non-coplanar, see Figure 3.2. Coplanar points are inadequate for camera calibration [9] [16] [18] [19] [22] and therefore, non-coplanar points are necessary. Determining the world points of an object, such as these shown in Figure 3.2 may be difficult. Therefore world points in Tsai's method are arranged on a planar surface, referred to as a calibration grid or jig. The jig is elevated through successive elevation stages to create a 3D calibration object, Figure 3.3. Images of the calibration jig are captured at each successive stage.

Figure 3.2 Coplanar vs. non-coplanar.

Figure 3.3 Non-coplanar calibration jigs.

The world point coordinates of all points on the jig are necessary as the base data requirements for calibration. These coordinates are usually measured individually, with a tape measure, in the X and Y directions. The Z value of each world point of a stage would be the same as they are in an XY plane. The position of the world points are known in relation to an arbitrarily selected world origin (O_w) that is outside the field of view of the camera. The requirement to select the world origin outside the field of view relates to a specific requirement of Tsai's method, selecting the sign of T_y , which is discussed in Section 3.5.

3.3 Tsai's Camera Model

Tsai's camera model [2] is based on a radial alignment constraint. The radial alignment constraint is the assumption that only radial distortion is present in the camera model and therefore the undistorted image point P_u and the associated distorted image point P_d will be along the same radial line extending from the origin O , as illustrated in Figure 3.4 and Figure 2.5. The undistorted image point is defined as the image point that would be

present if there was no distortion in the perspective projection of world points. The radial alignment constraint restricts the location of the undistorted image point $[x_u, y_u]^T$ to lie along the radial line from O through $[x_d, y_d]^T$.

Figure 3.4 Radial alignment constraint.

The model is based upon a four step transformation from world points to image points. The model is how Tsai defines the camera to function. In Section 3.5, the calibration method is related to the estimation of the parameters based on the definition of Tsai's camera model. An overview of the steps of Tsai's camera model is illustrated by Figure 3.5 and is explained below.

Figure 3.5 Four steps of Tsai's method.

Step 1. Rigid body transformation from the world frame to camera frame. This is accomplished with the use of equation (3.1) where

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R_c^* \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_w^c \end{bmatrix}_c \quad (3.4)$$

The rotation matrix R_c^* is defined by the Euler angles yaw θ , pitch ϕ and roll ψ .

$$R_c^* = \begin{bmatrix} \cos \psi \cos \theta & \sin \psi \cos \theta & -\sin \theta \\ -\sin \psi \cos \phi + \cos \psi \sin \theta \cos \phi & \cos \psi \cos \phi + \sin \psi \sin \theta \sin \phi & \cos \theta \sin \phi \\ \sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi & -\cos \psi \sin \phi + \sin \psi \sin \theta \cos \phi & \cos \theta \cos \phi \end{bmatrix} \quad (3.5)$$

Step 2. Transformation from camera points to undistorted image points. This is a perspective projection using pinhole camera geometry

$$x_u = f \frac{X}{Z} \quad (3.6)$$

$$y_u = f \frac{Y}{Z} \quad (3.7)$$

Tsai uses the Z axis as the optical axis of his camera model. This step uses the perspective projection for 3D to 2D as illustrated in Figure 2.2. The dimensions of $[x_u, y_u]^T$ is metres.

Step 3. Radial lens distortion. Tsai's camera model includes only radial distortion. This is key to the definition of Tsai's camera model, otherwise the radial alignment constraint would not apply. Radial lens distortion is mathematically modelled as

$$x_d + \Delta x_r = x_u \quad (3.8)$$

$$y_d + \Delta y_r = y_u \quad (3.9)$$

where Δx_r and Δy_r are radial distortion components and defined as

$$\Delta x_r = x_d (\kappa_1 r^2 + \kappa_2 r^4 + \dots) \quad (3.10)$$

$$\Delta y_r = y_d (\kappa_1 r^2 + \kappa_2 r^4 + \dots) \quad (3.11)$$

and

$$r = \sqrt{x_d^2 + y_d^2} \quad (3.12)$$

The dimensions of $[x_d, y_d]^T$ is metres.

Step 4. Real image coordinate to computer image coordinate. In Tsai's camera model, the image is captured by a camera with distorted image points $[x_d, y_d]^T$ and the image is

displayed in computer frame memory as $[x_f, y_f]^T$. The resolution between camera and computer may not be equal which creates a scaling problem which is accounted for by d'_x where

$$d'_x = d_x \frac{N_{cx}}{N_{fx}} \quad (3.13)$$

and d_x is the centre to centre distance between adjacent sensor elements in the CCD array in the X direction. N_{cx} is the number of sensor elements of the CCD array in the X direction and N_{fx} is the number of pixels in an image line of the computer frame memory. The origin of the computer frame memory is at the top left and because the distorted image points are centered around the optical axis (Z axis for Tsai's model), the distorted image point coordinates must be renumbered by adding the value of the principal point $[C_x, C_y]^T$ to adjust the distorted point to the computer frame memory coordinates $[x_f, y_f]^T$. The dimensions of $[x_f, y_f]^T$ is pixels. This is accomplished by

$$x_f = s_x d_x'^{-1} x_d + C_x \quad (3.14)$$

$$y_f = d_y'^{-1} y_d + C_y \quad (3.15)$$

The horizontal scale factor, s_x , is included in (3.14) to account for the changing from camera array to computer frame memory as discussed in Section 2.5.4.

3.4 Back Projection

Back projection is the inverse perspective projection from 2D to 3D. Back projection requires the undistorted image points, the effective focal length and a flat earth plane assumption, as illustrated in Figure 3.1. The flat earth plane is a necessary assumption to

determine the 3D coordinates of the back projected points from the 2D undistorted image points, otherwise there is no means to provide a depth limit. This can be illustrated by equations (3.6) and (3.7), which are reproduced here

$$x_u = f \frac{X}{Z}$$

$$y_u = f \frac{Y}{Z}$$

and Figure 3.6.

Figure 3.6 Inverse perspective projection.

The back projected point can be anywhere along the ray extending from O_c through p_i unless an object plane is defined to provide an intersection point. The object plane is the flat earth assumption which provides the Z component to world point.

Once the parameters from the calibration method have been estimated, they are used in the back projection to calculate the back projected points. The first step is to get

the image frame points $[x_f, y_f]^T$ from the operator. The distorted image points are required and are calculated using

$$x_d = s_x^{-1} d_x (x_f - C_x) \quad (3.16)$$

$$y_d = d_y (y_f - C_y) \quad (3.17)$$

Once the distorted image points are calculated, then the undistorted image points can be determined

$$x_u = x_d (1 + \kappa_1 r^2 + \kappa_2 r^4) \quad (3.18)$$

$$y_u = y_d (1 + \kappa_1 r^2 + \kappa_2 r^4) \quad (3.19)$$

where $r = \sqrt{x_d^2 + y_d^2}$. The flat earth assumption is required to back project into the camera reference frame. Prior to this, a unit vector for each undistorted image point is calculated using the effective focal length f and $[x_u, y_u]^T$. The purpose of the unit vector is to calculate a scalar value such that the scalar times the unit vector will intersect the flat earth plane as represented in the camera reference frame [13] [41]. The equation for the unit vector ($\bar{r}$) is

$$\bar{r} = \frac{[x_u, y_u, f]^T}{|[x_u, y_u, f]^T|} \quad (3.20)$$

To determine the scalar value, λ , so $\bar{r}$ intercepts the flat earth plane in the camera reference frame

$$\lambda = \frac{[0, 0, -1]^T [T_c^*]_c}{[0, 0, 1]^T R_c^c \bar{r}} \quad (3.21)$$

The value of the back projected point in the world reference frame is

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}^T = R_c^* \begin{bmatrix} T_x^* \\ T_y^* \\ T_z^* \end{bmatrix} + \lambda \bar{F} \quad (3.22)$$

3.5 Parameter Estimation

To determine accurate back projection points, the parameters of the camera must be estimated. Stage one of Tsai's method estimates the 3D rotation matrix R_c^* , and T_x and T_y of the translation vector $\begin{bmatrix} T_x^* \\ T_y^* \end{bmatrix}$, and the horizontal scale factor s_x . Stage two estimates the effective focal length f , the distortion coefficients κ_1, κ_2 and the remaining element of the translation vector, T_z . Recall that

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = R_c^* \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} + \begin{bmatrix} T_x^* \\ T_y^* \\ T_z^* \end{bmatrix} \quad (3.23)$$

where

$$R_c^* = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix} \quad (3.24)$$

and

$$\begin{bmatrix} T_x^* \\ T_y^* \\ T_z^* \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (3.25)$$

The basis of Tsai's method is the radial alignment constraint where the undistorted and distorted image points must reside on the same radial line from the origin. If you take Z out of the world point coordinates, $\begin{bmatrix} X_c \\ Y_c \end{bmatrix}^T$ remains and it is possible for $\begin{bmatrix} X_c \\ Y_c \end{bmatrix}^T$ to

be parallel to $[x_d, y_d]^T$. If these two radial lines are parallel, then their cross product equals zero

$$\begin{bmatrix} x_d \\ y_d \end{bmatrix} \times \begin{bmatrix} X_c \\ Y_c \end{bmatrix} = \mathbf{0} \quad (3.26)$$

Therefore

$$x_d Y_c - y_d X_c = 0 \quad (3.27)$$

From (3.23) and (3.24)

$$X_c = (r_1 X_w + r_2 Y_w + r_3 Z_w + T_x) \quad (3.28)$$

$$Y_c = (r_4 X_w + r_5 Y_w + r_6 Z_w + T_y) \quad (3.29)$$

Now

$$x_d (r_4 X_w + r_5 Y_w + r_6 Z_w + T_y) = y_d (r_1 X_w + r_2 Y_w + r_3 Z_w + T_x) \quad (3.30)$$

A system of linear equations is now possible to estimate the values of r_{1-6} , using the distorted image points. The unit of measure of (3.30) is metres and the computer frame memory image points are expressed in pixels, therefore the conversion from $[x_f, y_f]^T$ to $[x_d, y_d]^T$ is determined using:

$$x_d = s_x^{-1} d_x (x_f - C_x) \quad (3.31)$$

$$y_d = d_y (y_f - C_y) \quad (3.32)$$

Now that $[x_d, y_d]^T$ points are known and equation (3.30) is established, a system of linear equations is now possible by rearranging (3.30) as

$$\begin{bmatrix} y_{di}X_{wi} & y_{di}Y_{wi} & y_{di}Z_{wi} & y_{di} & -x_{di}X_{wi} & -x_{di}Y_{wi} & x_{di}Z_{wi} \end{bmatrix} \begin{bmatrix} T_y^{-1}r_1 \\ T_y^{-1}r_2 \\ T_y^{-1}r_3 \\ T_y^{-1}T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \\ T_y^{-1}r_6 \end{bmatrix} = [x_{di}] \quad (3.33)$$

$i = 1 \dots N$ where N is the total number of world points used in the calibration.

To simplify equation (3.33), use

$$A = \begin{bmatrix} y_{di}X_{wi} & y_{di}Y_{wi} & y_{di}Z_{wi} & y_{di} & -x_{di}X_{wi} & -x_{di}Y_{wi} & x_{di}Z_{wi} \end{bmatrix} \quad (3.34)$$

where i now represents a row index, so A is an $N \times 7$ matrix.

Let

$$\begin{bmatrix} T_y^{-1}r_1 \\ T_y^{-1}r_2 \\ T_y^{-1}r_3 \\ T_y^{-1}T_x \\ T_y^{-1}r_4 \\ T_y^{-1}r_5 \\ T_y^{-1}r_6 \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} = a \quad (3.35)$$

and

$$b = [x_{di}] \quad (3.36)$$

where b is an $N \times 1$ matrix. Therefore, we have a familiar notation of

$$Aa = b \quad (3.37)$$

As the system of linear equations has 7 unknowns, there must be a minimum of 7 world points used in Tsai's calibration process.

Initially, the horizontal scale factor s_x is set to 1. With the number of calibration points much larger than seven, this over-determined system of linear equations is solved for seven unknowns, $T_y^{-1}s_x r_1, T_y^{-1}s_x r_2, T_y^{-1}s_x r_3, T_y^{-1}s_x T_x, T_y^{-1}s_x r_4, T_y^{-1}s_x r_5$ and $T_y^{-1}s_x r_6$ as a least squares estimate. Minimize

$$E_1(a) = \frac{1}{2} (Aa - b)^T (Aa - b) \quad (3.38)$$

Take the first derivative and set it equal to zero.

$$\frac{\partial E_1}{\partial a} = [Aa - b]^T A = 0 \quad (3.39)$$

Now rearranging

$$\begin{aligned} -b^T A &= -\hat{a}^T A^T A \\ A^T A \hat{a} &= A^T b \end{aligned}$$

and getting the first estimate of a , which is $\hat{a}$

$$\hat{a} = (A^T A)^{-1} A^T b \quad (3.40)$$

In Tsai's paper [2], there is a discussion that the third row of the rotation matrix must be orthogonal to the first two rows but no mention that the first two rows must be orthogonal.

Nothing in the above formulation constrains $[r_1 \ r_2 \ r_3]^T$ to be orthogonal to

$[r_4 \ r_5 \ r_6]^T$. This constraint is a modification added to Tsai's method by Aitken [13]

[41] as

$$[a_1 \ a_2 \ a_3] \begin{bmatrix} a_5 \\ a_6 \\ a_7 \end{bmatrix} = 0 \quad (3.41)$$

Define the matrix Q such that

$$Q = \begin{bmatrix} 0_{3 \times 4} & I_3 \\ 0_{4 \times 4} & 0_{4 \times 3} \end{bmatrix} \quad (3.42)$$

where I_3 is the 3×3 identity matrix. Now to solve the least squares problem for a and maintain the orthogonality constraint, the new minimization is

$$E(a, \lambda) = \frac{1}{2} (Aa - b)^T (Aa - b) + \frac{1}{2} \lambda a^T Qa \quad (3.43)$$

where λ is a Lagrange multiplier. Now

$$\begin{aligned} \frac{\partial E}{\partial x} &= (Aa - b)^T A + \lambda a^T Q \\ &= a^T (A^T A + \lambda Q) - b^T A = 0 \end{aligned} \quad (3.44)$$

$$\frac{\partial E}{\partial \lambda} = a^T Qa = 0 \quad (3.45)$$

To solve equations (3.44) and (3.45), a Gauss-Newton iterative optimization method [40] is employed. Specifics with regards to the Gauss-Newton method are at Appendix A [13] [41]. The cost function is defined as

$$F(x_c) = \begin{bmatrix} (A^T A + \lambda Q^T)a - A^T b \\ a^T Qa \end{bmatrix}, \quad x_c = \begin{bmatrix} a \\ \lambda \end{bmatrix} \quad (3.46)$$

where $F(x_c)$ is a 8×1 vector. Establish the Jacobian matrix

$$J(x_c) = \frac{\partial F}{\partial x_c} = \begin{bmatrix} A^T A + \lambda Q^T & Q^T a \\ 2a^T Q & 0 \end{bmatrix} \quad (3.47)$$

The initial value of x_0 is

$$x_0 = \begin{bmatrix} \hat{a} \\ 0 \end{bmatrix} \quad (3.48)$$

where $\hat{a}$ is the first estimate of a from equation (3.40) and $\lambda = 0$ as the initial estimate.

The Gauss-Newton iterations are

$$x_{n+1} = x_n - S [J(x_n)]^{-1} F(x_n) \quad n = 0, 1, 2, \dots \quad (3.49)$$

where $S \in (0, 1]$ is the step size to prevent “over-shooting” the solution.

To determine the final estimated values of $r_1 \dots r_6, T_x$ and T_y , the magnitude and sign of T_y is required. The magnitude of T_y is determined by:

$$|T_y| = \frac{1}{\sqrt{a_5^2 + a_6^2 + a_7^2}} \quad (3.50)$$

To determine the sign of T_y , select the object point i with image coordinates

$[x_f, y_f]^T$ furthest from the image centre, and world coordinates $[X_i, Y_i, Z_i]^T$, and select

T_y to be $+1$. From (3.35) and (3.50), compute initial values of $r_1 \dots r_6$ and T_x using

$$\begin{aligned} r_1 &= a_1 \frac{T_y}{s_x}; & r_2 &= a_2 \frac{T_y}{s_x}; & r_3 &= a_3 \frac{T_y}{s_x}; \\ r_4 &= a_5 \frac{T_y}{s_x}; & r_5 &= a_6 \frac{T_y}{s_x}; & r_6 &= a_7 \frac{T_y}{s_x}; & T_x &= a_4 \frac{T_y}{s_x} \end{aligned} \quad (3.51)$$

with $s_x = 1$ initially and

$$\begin{aligned} x &= r_1 X + r_2 Y + r_3 Z + T_x \\ y &= r_4 X + r_5 Y + r_6 Z + T_y \end{aligned} \quad (3.52)$$

If x and X have the same sign and y and Y have the same sign, then $T_y = +1$, else

$T_y = -1$. The reason equation (3.52) works in determining the sign of T_y is because of

the radial alignment constraint. The value for f and Z must both be positive based on

the way the camera reference frame is defined, and if x and X do not have the same sign

or y and Y do not have the same sign, then f and Z could be negative which is not

possible. The sign and magnitude of T_y is important in determining the values of $r_1 \dots r_6$ and T_x . With the sign of T_y known, recalculate $r_1 \dots r_6$ and T_x using (3.51).

To determine the value of s_x

$$s_x = \frac{|T_y|}{\sqrt{a_1^2 + a_2^2 + a_3^2}} \quad (3.53)$$

To calculate the values of $r_1 \dots r_6$, equation (3.51) is used and the remaining elements of R , r_7, r_8, r_9 , are determined using the cross product of the first two rows of R and the property that R is orthogonal.

The next stage is to estimate the values of f , κ_1 , κ_2 and T_z . At first, ignore lens distortion and establish an over-determined linear system to get an initial estimate for f and T_z

$$\begin{bmatrix} Y_i & -d_y y_i \end{bmatrix} \begin{bmatrix} f \\ T_z \end{bmatrix} = \begin{bmatrix} w_i d_y y_i \end{bmatrix} \quad (3.54)$$

where

$$y_i = r_4 X_i + r_5 Y_i + r_6 Z_i + T_y \quad (3.55)$$

$$w_i = r_7 X_i + r_8 Y_i + r_9 Z_i \quad (3.56)$$

A least squares solution of (3.54) is used as initial estimates for f and T_z , then a non-linear optimization is used to improve on the estimated values of f , κ_1 , κ_2 and T_z . In Tsai's original conference paper, he included κ_2 but dropped it from [2], however it is maintained in this thesis as a parameter to be estimated. Tsai suggests a steepest descent optimization scheme using either equation (3.57) or (3.58). In a modification to Tsai's

method by Aitken. as proposed by [13] [41], a Gauss-Newton optimization (see Appendix A) is implemented using two cost functions to be minimized

$$f \frac{r_1 X + r_2 Y + r_3 Z + T_x}{r_7 X + r_8 Y + r_9 Z + T_z} - \frac{d'_x}{s_x} x_d (1 + \kappa_1 r^2 + \kappa_2 r^4) = 0 \quad (3.57)$$

$$f \frac{r_4 X + r_5 Y + r_6 Z + T_y}{r_7 X + r_8 Y + r_9 Z + T_z} - d_y y_d (1 + \kappa_1 r^2 + \kappa_2 r^4) = 0 \quad (3.58)$$

where the partial derivatives are taken with respect to $T_z, f, \kappa_1, \kappa_2$ and $r = \sqrt{x_d^2 + y_d^2}$.

3.6 Evaluation

Tsai acknowledges that it is difficult to assess the accuracy of camera calibration parameters as the actual parameter values of real camera systems are not known. “It is difficult to obtain high accuracy ground truth for camera calibration parameters that can serve as absolute reference. Therefore, we will assess the accuracy of the two-stage camera calibration by how well it can sense or measure the 3D world” [2].

The measure of accuracy to be used in this thesis is the mean error (ME). The mean error is the mean difference between all world points and corresponding back projected points and does not include the Z component due to the flat earth assumption. defined as

$$ME = \frac{1}{N} \sum_{i=1}^N \sqrt{(X_{wi} - X_{Bi})^2 + (Y_{wi} - Y_{Bi})^2} \quad (3.59)$$

where $[X_B, Y_B]^T$ are the coordinates of the back projected points.

The accuracy assessment can be applied in either the world or camera reference frame. Tsai chooses to conduct the analysis in F_c but either method is acceptable. It is the mean error in F_w which is used in this thesis.

Chapter

4 MatCam

4.1 Purpose of MatCam

MatCam is a computer based mathematical abstraction of a camera model designed to interface with a camera calibration method to study how accurately the calibration method estimates the intrinsic and extrinsic parameters and to compare the back projected points with the world points. Here, we define “input data” as the image point data in the correct format to interface with the calibration method. With MatCam, the input data is generated with precision and the intrinsic and extrinsic parameters used to generate the data are known exactly. MatCam provides a valuable tool for the study of a calibration method’s estimated parameter values when various controllable conditions are applied to input data provided to a calibration method. MatCam provides an error assessment capability to determine the mean error (ME), equation (3.59), under controllable conditions. Input data is synthetically generated to replicate physical conditions such as distortion, and also to allow for the study of error sources which enter into the calibration process.

During the literature review, synthetic data of known parameter values was used to compare the output from the calibration method to precisely known parameters [21]. This control provided experimental confirmation that the proposed estimation method did exactly as intended. The details as to the confirmation of the accuracy of the synthetic data and model design in that work were not discussed and this provided the motivation for the development of MatCam.

MatCam provides the ability to generate synthetic input data that includes:

- Radial distortion, first and second order coefficients κ_1 and κ_2 .
- De-centering distortion with coefficients κ_4 and κ_5 .
- Measurement error of the calibration points.
- Resolution noise in the selection of calibration points in the image frame.
- Principal point offset within the CCD array.
- Image coverage reduction to study calibration methods when full image data is not available.
- Digitization error when converting analog images to digital.

4.2 Overview of MatCam

MatCam is divided into two parts: an input data generator and an inverse perspective projection control unit. The input data generator uses a perspective projection model with add on features to carefully adjust the data to meet the needs of the study to be conducted. A synthetic calibration grid, known to exact tolerance, and known camera parameters are used in the production of input data. The inverse perspective projection control unit uses

inverse perspective projection to back project to a flat earth plane. The calibration method is sandwiched between the two parts of MatCam. see Figure 4.1.

Figure 4.1 MatCam model.

The design of MatCam is based on a broad class of camera models proposed in the literature, [2], [3], [4], [7], [10], [11], [16], [17], [23], [25], [28], [32], [38], and [13] [41] which provide code segments (MatLab).

MatCam utilizes perspective projection of world points as in Figure 4.1 as the basis for input data generation. MatCam captures the world points as undistorted image points, modifies the points in a controlled manner as will be discussed in later sections, and provides this data as input to the calibration method to estimate parameters. Then the

inverse perspective projection unit determines back projected points from which error analysis is conducted. MatCam also provides a control feature of simultaneous display of world points, corresponding to the back projected points, to visualize the errors. Details as to how this works is presented below.

4.2.1 Input Data Generator

An overview of the input data generator is provided with the details of each step explained in the following sections. The initial requirement is to determine the format for the input data that the camera model and calibration method under study require. As Tsai's non-coplanar calibration method is used in Chapter 5 as a comparison tool for MatCam, the explanation of the input data will be consistent with the data requirements of Tsai's method. For example, three separate planes of world points and corresponding image points, in computer frame coordinates, are provided to the calibration method. Additionally, the technical data of pixel spacing and frame resolutions are provided. The input generator uses this same technical data in the production of the synthetic data as is used by the calibration method.

With a known camera position, known world points, and known intrinsic parameters for the camera, the input generator calculates the rotation matrix and determines the world point coordinates in F_w, F_c and the computer frame image point, $[x_f, y_f]^T$, for each world point. At this juncture, options are available to modify the data to meet the purpose of numerous studies. The options are presented at Section 4.1. The distortion options and error source options can be applied individually or in combination. Once the input data is produced, these can be sent to either the calibration

method or directly to the inverse perspective projection unit to display the back projected points and world points to allow a comparison of points and analyze error.

The program used to implement this mathematical algorithm is MatLab™ version 5.3 [39] and the code is contained in Appendix B. As MatCam is implemented in MatLab™, the numerical accuracy of the data is not being placed in question. This input data is accurately controlled (double precision floating point) to provide flexibility in the conditions as to how it is used and also to maintain the numerical integrity of the data.

4.2.2 Details of Implementing MatCam

Details for the generation of input data are provided in this section. At the end of each step, a summary of parameters introduced and used to that point in the process and any additional information that would be known is provided. As a note, a clear plan for the study to be conducted is required prior to initiating MatCam as parameter values and experimental set up are not options available once MatCam is initiated.

Step 1. Develop a virtual calibration grid/jig and determine the world points in the camera reference frame. In this step, depending upon the camera model of study, a calibration grid of world points is created. Figure 4.2 illustrates the calibration jig and the world points, in the world reference frame. In Figure 4.2, the world points are positioned in three coplanar stages. The three stages are the base, first and second stage. The base stage is set to zero elevation and the first and second stage are preset by the operator. In Figure 4.2, the base stage is represented by an “*”, the first stage up from the base is represented by a “+”, and the second stage is represented by a “◇”. The position of the camera in Figure 4.2 is shown as a single “◊” above the world points. This symbology of world point stages and camera position is maintained throughout this thesis. The position

of the camera is preset by the operator. The camera is shown with the optical axis vector which is designed to “look” directly at the centre of the base stage calibration plane. The centre of the calibration jig is preset by the operator. Additional features of the calibration jig are:

- shape of the calibration jig can be square Figure 4.2 or spoke Figure 4.3.
- outer dimensions.
- spacing of world points.
- elevation heights of stage one and stage two, and
- for the spoke pattern, the number of spokes.

Figure 4.2 Square calibration jig.

The features of Figure 4.3 are the same as Figure 4.2 except for the different shape of the calibration jig. As stated, the camera position and centre of the calibration jig are linked such that the optical axis is from the camera to the centre position of the base stage

calibration jig. These two points are used to determine the rotation matrix from the world to camera reference frames. The vector from the camera position to the jig centre, in the camera reference frame, is calculated and using the X axis as the optical axis, Y axis positive left and Z axis positive up. This transforms the world points from world reference frame F_w to camera reference frame F_c as in Figure 4.4. The feature and symbols of Figure 4.4 are the same as Figure 4.2, except the world points and camera position are now in the camera reference frame. The camera is positioned at the origin, $[0,0,0]^T$ of F_c . Details of the linear transformation are in accordance with Section 3.2.

The vector from the camera origin to the jig centre in F_c is $[X_c, Y_c, Z_c]^T$ and a constraint of zero roll is applied and $R_w'^c$ is calculated as

$$R_w'^c = \begin{bmatrix} X_c & -Y_c & -Z_c \cos\left(\tan^{-1}\left(\frac{Y_c}{X_c}\right)\right) \\ Y_c & X_c & -Z_c \sin\left(\tan^{-1}\left(\frac{Y_c}{X_c}\right)\right) \\ Z_c & 0 & \sqrt{X_c^2 + Y_c^2} \end{bmatrix} \quad (4.1)$$

Once $R_w'^c$ is calculated, $R_w'^c$ is normalized by dividing each column vector by its length to give R_w^c which is now the rotation matrix. R_w^c is confirmed to be orthogonal by the MatLab code.

Depending upon the camera model and calibration method under study, the jig can be modified from being multi-planar, as in this example, to single plane. Other options such as a 3D calibration object could be developed. As an option, preset by the operator prior to commencing the execution of MatCam, a measurement error may be applied to the world points to simulate errors in the accurate measurement of the world

points and analyze the effect on the estimated parameters. The measurement error is applied as a truncated normal distribution. The maximum measurement error per world point is truncated with a variance value to prevent severe outliers in the measurements. The maximum measurement error and variance are preset by the operator.

Figure 4.3 Spoke calibration jig.

At this point in the process, information that is known is:

- Effective focal length f .
- Translation vector T_c^w from the origin of F_w to the origin of F_c .
- Centre point of the calibration jig.
- Optical axis vector.
- Position of all world points in F_w , $[X, Y, Z]^T$ with or without measurement error applied.

- Rotation matrix R_w^c .

Figure 4.4 World points transformed to F_c .

Step 2. Determine undistorted image points in F_c . Using the resolution of the CCD array and the array dimensions, calculate the centre-to-centre pixel spacing distances d_x and d_y . With the pixel spacings and perspective projection, convert the world points in F_c to image points with dimensions in metres. As the optical axis is known (the X axis in F_c), and the world points are known in F_c , calculate the coordinates of the undistorted image points $[x_u, y_u]^T$. Figure 4.5 is an example of the world points represented as image points with the three stages of world points represented in a 2D image plane. The symbols for each stage are the same as in previous figures.

As the resolution of an image will likely have an effect on the ability to accurately select the precise centre of a world point in the image plane, an optional resolution noise factor modeled similar to the measurement error is available for application to the input data. A maximum resolution error, in pixels, and variance are preset by the operator. At this point in the process, additional information that is known is:

- CCD array resolution, to include N_{cx} the number of columns in the CCD array.
- Pixel-to-pixel spacing in x (d_x) and y (d_y) direction for CCD array.
- Position of undistorted image points $[x_u, y_u]^T$ in F_c .

Figure 4.5 Image points in camera reference frame F_c .

Step 3. Apply a distortion model. Distortion models, such as the one described in Section 2.6, require the centre of the array to be known. This is exactly how the

undistorted point set is positioned at this point in MatCam, as in Figure 4.5, and therefore it is the correct point in the process to apply distortion. The distortion model combines radial and de-centering distortion to adjust the undistorted point $[x_u, y_u]^T$. Either radial or de-centering distortion can be applied individually or combined. Radial distortion can also be sub-divided into first order or first and second order. The change in x component due to distortion is $\Delta x = \Delta x_r + \Delta x_d$ and similarly the change to y component due to distortion is $\Delta y = \Delta y_r + \Delta y_d$ where the subscripts r and d are for radial and de-centering distortion components. Therefore, using equations (2.9), (2.10), and (2.12)

$$\Delta x = x_d \left(1 + \kappa_1 r^2 + \kappa_2 r^4 \right) + \left(\kappa_4 \left(r^2 + 2x_d^2 \right) + 2\kappa_5 x_d y_d \right) \quad (4.2)$$

$$\Delta y = y_d \left(1 + \kappa_1 r^2 + \kappa_2 r^4 \right) + \left(2\kappa_4 x_d y_d + \kappa_5 \left(r^2 + 2y_d^2 \right) \right) \quad (4.3)$$

where κ_4 and κ_5 are coefficients of de-centering distortion and $r = \sqrt{x_d^2 + y_d^2}$. Note that

κ_3 has been reserved for potential use of third order radial distortion term. As $[x_d, y_d]^T$ are required and $[x_u, y_u]^T$ are known, an algorithm to accurately determine the distorted point set is required. The algorithm to determine the distorted point set is a key component of the MatCam. The algorithm uses Newton's method (see Appendix A) where the objective function

$$F1 = x_d \left(1 + \kappa_1 r^2 + \kappa_2 r^4 \right) + \left(\kappa_4 \left(r^2 + 2x_d^2 \right) + 2\kappa_5 x_d y_d \right) - x_u = 0 \quad (4.4)$$

$$F2 = y_d \left(1 + \kappa_1 r^2 + \kappa_2 r^4 \right) + \left(2\kappa_4 x_d y_d + \kappa_5 \left(r^2 + 2y_d^2 \right) \right) - y_u = 0 \quad (4.5)$$

is minimized. This is accomplished by taking the first partial derivative of each function using

$$J(F1, F2) = \begin{bmatrix} \frac{\partial F1}{\partial x_d} & \frac{\partial F1}{\partial y_d} \\ \frac{\partial F2}{\partial x_d} & \frac{\partial F2}{\partial y_d} \end{bmatrix} \quad (4.6)$$

and then apply Newton's method to determine each $[x_d, y_d]^T$. A check of the accuracy of how the distorted image points are calculated was conducted. The calculated distorted image points were used as variables in equations (4.4) and (4.5), with the known parameter values and the relative error for each point was determined and is shown in Figure 4.6. The maximum error was less than 2.2×10^{-16} metres.

Figure 4.6 Magnitude of error between the known undistorted points and calculated undistorted points using the synthesized distorted points.

Figure 4.7 illustrates the base level image points when no distortion is applied. All levels of image points are distorted, however for a clear understanding of how the image points are distorted, only the base level is shown. The undistorted image point is represented by

the “+” in Figure 4.7 through Figure 4.10. In Figure 4.8 through Figure 4.10, varying types of distortion are applied. If an image point has been distorted, it is represented in the figure as a straight vector with the tail of the vector at the base of the undistorted image point and the tip of the vector represents the distorted image point coordinates. Figure 4.8 illustrates radial distortion only. The values of $\kappa_1=500$, $\kappa_2=1.e+07$ are used and the figure indicates how the points are drawn inward indicating barrel distortion.

Figure 4.7 Position of point set undistorted.

Figure 4.8 Radial distortion only.

Figure 4.9 illustrates de-centering distortion only. The values of $\kappa_4=2$, $\kappa_5=0.33$ distorts the image points both radially and tangentially.

Figure 4.9 De-centering distortion only.

Figure 4.10 illustrates both radial and de-centering distortion combined. The values of $\kappa_1=500$, $\kappa_2=1.e+07$, $\kappa_4=2$, $\kappa_5=0.33$ are used.

Figure 4.10 Radial and de-centering distortion.

From this point forward, “distorted point set” implies the image points after distortion has been applied.

At this point in the process, additional information that is known is:

- Distorted image points $[x_d, y_d]^T$.
- Values of the distortion coefficient(s) used, $\kappa_1, \kappa_2, \kappa_4, \kappa_5$.

Step 4. Convert the distorted point set to pixel coordinates and apply a field of view (FOV) boundary. The distorted point set is converted from dimensions of metres to dimensions of pixels in the CCD array. Figure 4.11 is an example of a distorted point set of three levels of world points in $[x_f, y_f]^T$ coordinates. Note that all world points are

currently visible as image points. Based on the resolution of the CCD array, determine the image points that lie within the defined field of view of the camera. Only these image points will be used henceforth. Figure 4.12 is an example of applying the rectangular FOV boundary to the complete image point set. Horizontal and vertical “crosshairs” are provided as an aid to observe symmetry of image points.

Figure 4.11 CCD array pattern of distorted image points.

Figure 4.13 is an example of the same point set of Figure 4.12 with only the distorted image points that will be used for the calibration method. The same FOV boundary and crosshairs are visible with the symbols for each stage as in previous figures.

Figure 4.12 CCD array field of view boundary applied to distorted point set.

Figure 4.13 Distorted image point set limited to field of view only.

As the resolution of the CCD array and computer frame memory resolution may not match, a scaling may be conducted to convert to computer frame coordinates.

At this point in the process, additional information that is known is:

- Distorted image points in $[x_f, y_f]^T$ coordinates.
- Computer frame memory resolution, including N_{fx} , the number of columns.

Step 5. Modify the point coverage distribution in the image plane. Depending upon the calibration method being studied, the majority of methods require the image points to be in computer memory frame coordinates which is the current state for this step and MatCam could interface with the calibration method at this point. An additional option has been developed in MatCam to remove image points from the image plane to artificially create “holes” to force these unbalanced regions. This option is valuable to investigate how the calibration method reacts to unbalanced images. Figure 4.14 is an example of Figure 4.13 with a “corner” of image points removed from the input data.

Figure 4.14 Modified image points.

Step 6. Execute the camera calibration method or conduct inverse perspective projection of input data. Inverse perspective projection is an ideal case in that distortion is neglected. At this point, all parameters that created the input data are known and the data is in a form to interface with the calibration method. Option one is to interface the input data with the calibration method and conduct an estimation of parameters. Option two is to skip the calibration method and send the input data to the inverse perspective projection unit to display the image points as they would appear if looking through the camera lens with no distortion. Option two provides a visual confirmation of how the world points have been distorted. The corresponding world points are displayed and error analysis is conducted with both options.

4.2.3 Inverse Perspective Projection Unit

The inverse perspective projection unit calculates the back projected points. As MatCam uses world points as the initial step in the generation of input data, the same world points are used in the comparison to back projected points. To clarify, as the back projected points use a flat earth assumption, only the world points that are on the same flat earth plane are used for comparison purposes. The inverse perspective projection unit also conducts the error analysis of the mean error in both numerical and graphical form. It should be noted that only the back projected and world points within the FOV of the camera are included in the graphical display and numerical analysis. Figure 4.15 illustrates the display of inverse perspective projection unit where the back projected points are represented by the "*" and the world points are the "o". Also in the image is the camera position in the XY plane, represented as a "x", shown in the figure at $[3,2]^T$.

The back projected points in Figure 4.15 represent the distorted image points produced with the same parameters as Figure 4.10. Figure 4.15 shows the importance of considering distortion in the back projection process.

Figure 4.15 Example of inverse perspective projection display.

Numerical data provided by MatCam:

- Maximum single point error.
- Mean point error.
- Median point error.
- Error in X direction for each back projected point.
- Error in Y direction for each back projected point.
- Magnitude of error for each back projected point.

- Point label and world point coordinates.

Table 4.1 provides a summary of the numerical data discussed above.

Table 4.1 Summary of point set errors.

Maximum single point error	166.0 mm
Mean point error	33.4 mm
Median point error	24.3 mm

An example of the detailed numerical data used to generate Table 4.1 is at Table 4.2. Table 4.2 contains: the label number of each base level world point that was used in the calibration method with the X and Y coordinates of the world point, the error of the back projected point in both X and Y directions, and the magnitude of the error between the back projected point and the world point. Dimensions of each value is as per the table. The data contained in Table 4.2 can be referenced to the graphical information of Figure 4.15. Note that there is no error in point label 85 of Table 4.2 as this is the centre point of the calibration grid and is intersected by the optical axis.

Table 4.2 Example of the error data available from MatCam perspective projection unit.

Point Label	X (m)	Y (m)	X error (m)	Y error (m)	$ Error $ (mm)
46	4.0	1.5	-0.0013	-0.0194	19.4
57	3.0	2.0	0.0111	-0.0046	12.0
58	3.5	2.0	0.0030	-0.0068	7.4
59	4.0	2.0	-0.0006	-0.0128	12.8
60	4.5	2.0	-0.0050	-0.0231	23.6
61	5.0	2.0	-0.0151	-0.0366	39.6
69	2.5	2.5	0.0229	-0.0181	29.2
70	3.0	2.5	0.0062	-0.0078	10.0
71	3.5	2.5	0.0010	-0.0028	2.9

Point Label	X (m)	Y (m)	X error (m)	Y error (m)	$ Error $ (mm)
72	4.0	2.5	-0.0001	-0.0039	3.9
73	4.5	2.5	-0.0018	-0.0111	11.3
74	5.0	2.5	-0.0081	-0.0238	25.1
75	5.5	2.5	-0.0240	-0.0405	47.0
82	2.5	3.0	0.0205	-0.0338	39.6
83	3.0	3.0	0.0050	-0.0151	15.9
84	3.5	3.0	0.0005	-0.0037	3.8
85	4.0	3.0	0	0	0
86	4.5	3.0	-0.0007	-0.0036	3.7
87	5.0	3.0	-0.0052	-0.0140	15.0
88	5.5	3.0	-0.0182	-0.0304	35.4
89	6.0	3.0	-0.0454	-0.0517	68.8
95	2.5	3.5	0.0234	-0.0573	61.9
96	3.0	3.5	0.0066	-0.0308	31.5
97	3.5	3.5	0.0010	-0.0134	13.4
98	4.0	3.5	-0.0001	-0.0047	4.7
99	4.5	3.5	-0.0011	-0.0044	4.5
100	5.0	3.5	-0.0055	-0.0120	13.2
101	5.5	3.5	-0.0177	-0.0272	32.5
102	6.0	3.5	-0.0432	-0.0492	65.5
109	3.0	4.0	0.0109	-0.0579	58.9
110	3.5	4.0	0.0026	-0.0342	34.3
111	4.0	4.0	-0.0005	-0.0205	20.5
112	4.5	4.0	-0.0029	-0.0162	16.5
113	5.0	4.0	-0.0087	-0.0210	22.7
114	5.5	4.0	-0.0219	-0.0347	41.0
123	3.5	4.5	0.0055	-0.0692	69.4
124	4.0	4.5	-0.0010	-0.0502	50.2
125	4.5	4.5	-0.0064	-0.0419	42.4
126	5.0	4.5	-0.0149	-0.0440	46.4
137	4.0	5.0	-0.0016	-0.0974	97.5
138	4.5	5.0	-0.0116	-0.0851	85.9
150	4.0	5.5	-0.0024	0.1661	166.1

4.3 Synthetic Data Examples

In this section, examples of the distorted image points will be illustrated. The purpose of this section is to demonstrate the effectiveness of MatCam in the generation of distorted image point data for input to calibration methods. In each of the figures, the distorted

input data is represented by an “ $\circ$ ” and the world points are represented by an “ $*$ ”. The camera position is maintained consistently at coordinate $[3, 2, 4]^T$ and represented by the “ $\circ$ ” in the figures. The calibration jig centre is also maintained consistently throughout at $[4, 3, 0]^T$. As it may be easier to visualize the input data for radial and de-centering distortion set in a different pattern than a square calibration grid, the spoke calibration pattern with multiple spokes and a single world point at the end of each spoke is used. The length of each spoke leg is set to 1 metre. For the examples of the effects of the other errors as listed below, the square calibration grid is used with a 2m outer dimension around the centre point of the grid and 0.5m spacing of world points. The input data set is a representation of how the image would appear in the camera display. World points are always displayed as a visual reference base for comparison.

The following figures illustrate distortion and error artifacts in back projection performance:

- Radial distortion, first order term only, see Figure 4.16.
- Radial distortion, first and second order terms, see Figure 4.17.
- Radial distortion, first order and second order terms negative, see Figure 4.18.
- De-centering distortion, see Figure 4.19.
- Radial and de-centering distortion, see Figure 4.20.
- Severe radial distortion, first and second terms, see Figure 4.21.
- Severe radial and de-centering distortion, see Figure 4.22.
- Principal point offset, see Figure 4.23.
- Measurement error, see Figure 4.24.

- Resolution error, see Figure 4.25.
- Digitization error, see Figure 4.26.

Figure 4.16 illustrates first order radial distortion as compared to the world points with $\kappa_1=8000$. Note that the world points are outside the inverse perspective projection of the input data points indicating that positive radial distortion will be required to “correct” the image points. For a review of positive and negative radial distortion, refer to Section 2.6.1.

Figure 4.16 Example of positive first order radial distortion.

Figure 4.17 illustrates first and second order radial distortion as compared to the world points with $\kappa_1=8000$ and $\kappa_2=1.0e+07$. There is little difference to the previous example. The scale of κ_2 is important as values greater than $1.0e+07$ can cause severe radial distortion as will be illustrated in Figure 4.21.

Figure 4.17 Example of positive first and second order radial distortion.

Figure 4.18 illustrates negative first and second order radial distortion as compared to the world points with $\kappa_1 = -8000$ and $\kappa_2 = -1.0e+07$. The input data points are now shifted outside of the world points indicating that negative radial distortion will be required to “correct” the image points.

Figure 4.18 Example of negative first and second order radial distortion.

Figure 4.19 illustrates de-centering distortion as compared to the world points with $\kappa_4=4.5$ and $\kappa_5=0.5$. Note the “tilting-table” effect of de-centering distortion. The de-centering distortion adjusts the image points as though the points are on a flat table and the table is “tilted” about the centre point.

Figure 4.19 Example of de-centering distortion.

Figure 4.20 illustrates radial and de-centering distortion as compared to the world points with $\kappa_1=8000$, $\kappa_2=1.0e+07$, $\kappa_4=4.5$, and $\kappa_5=0.5$. This example illustrates how de-centering distortion has both a radial and tangential component. The first and second order radial distortion coefficients “pull” the input data in symmetrically, the de-centering “adds” radial distortion in one direction and subtracts in the other direction which explains why the image points are not symmetrically adjusted around the centre world point.

Figure 4.20 Example of first and second order radial and de-centering distortion.

Figure 4.21 illustrates severe first and second order radial distortion as compared to the world points with $\kappa_1=8000$ and $\kappa_2=1.0e+11$. This is an example of how MatCam can synthetically create a severe distortion case.

Figure 4.21 Example of severe first and second order radial distortion.

Figure 4.22 illustrates severe first and second order radial and de-centering distortion as compared to the world points with $\kappa_1=8000$, $\kappa_2=1.0e+11$, $\kappa_4=4.5$, and $\kappa_5=0.5$. There is no readily apparent change to Figure 4.21 with the application of de-centering distortion to radial distortion. This illustrates how MatCam can be combined to adjust severe distortion studies.

Figure 4.22 Example of severe first and second order radial and de-centering distortion.

The next series of figures deals with an error in the selection of the principal point and the errors that may be applied to the world points.

Figure 4.23 illustrates the effect of an offset of the principal point of the CCD array by 35 pixels in the X direction, compared to the world points. The offset is in the X direction, relative to the camera array with the camera positioned at $[3, 2]^T$, and no radial or de-centering distortion is applied. The effect of principal point offset is similar to de-centering distortion except that principal point offset is an offset of the input data.

Figure 4.23 Example of offsetting the principal point.

Figure 4.24 illustrates the application of measurement error to the world points. An unrealistic maximum error in measurement of 40mm was applied in each of the X and Y directions and 2mm in the Z direction for illustrative purposes. The error is applied randomly to each world point. The maximum mean error for the input data was 32.0mm. In standard applications, a maximum error of less than 5mm in the X , Y and Z directions is recommended. Random error is applied to the world points but a modification to the software could be made to use a standard "offset" to all world points if a calibration jig is used and only one measurement to an edge of the calibration jig is used.

Figure 4.24 Example of measurement error.

Figure 4.25 illustrates the application of resolution error to the world points. A maximum error in resolution equal to 5 pixels was applied in each of the X and Y directions for illustrative purposes. The error is applied randomly to each image point and maximum mean error for the input data was 26.1mm. In standard applications, a maximum error of less than 5 pixels in the X and Y directions is recommended. The 5 pixel error would be at the discretion of the operator and the quality of the image would be a factor in selecting the resolution error maximum value.

Figure 4.26 illustrates the application of digitization error to the world points. This is the adjustment of image points to the closest integer coordinates in the computer image frame. The image point is "moved" to the centre of the pixel and the mean error, in this example, to the input data was 4.2mm with an average distance from camera to world point of 4.2426m. The magnitude of error is not significant but quite likely present in the camera model and calibration process.

Figure 4.25 Example of resolution error.

Figure 4.26 Example of digitization error.

Chapter

5 Evaluation of Tsai's Calibration Method With MatCam

5.1 Introduction

The primary goal of developing MatCam was to create an accurate computer based mathematical model of a camera system from which camera calibration methods could be evaluated. MatCam provides the ability to conduct parametric studies of calibration methods without the requirement to consider the physical set up as a limitation. The flexibility of the position of the camera, the position of the calibration grid, the shape of the grid (square or circular), the spacing of world points, varying array resolutions and most important, knowing the exact values of the intrinsic and extrinsic parameter values that created the input data, provides a valuable tool to study camera calibration methods.

The purpose of the simulations conducted in Chapter 4 was to validate the model as operating correctly based upon the numerical algorithms employed to generate the input data, and to analyze the accuracy of the input data. The focus of this thesis is to develop a computer based camera model, MatCam, to evaluate the performance of camera

calibration methods. The purpose of this chapter is to use MatCam in the evaluation of a camera calibration method.

Tsai's non-coplanar calibration method was selected as the method to evaluate and therefore MatCam is configured for this task. A review of Tsai's method is discussed in Chapter 3. Note that Tsai's calibration method initially included a second order coefficient for radial distortion that was subsequently dismissed [2], [32]. The implementation [41] of Tsai's method for this thesis includes the second order coefficient and therefore it will be used as a parameter of study.

Calibration methods are defined by the camera models from which they were developed. MatCam is based on a broad range of camera models with options that provide the ability to study a calibration method's reaction to distortion and sources of error. As the options of study with MatCam are flexible, there are numerous studies that may be conducted. The emphasis of this chapter will be limited to four related studies:

- Increased range of radial distortion up to first and second order coefficients and the effect on estimated parameters and mean error;
- De-centering distortion and the effect on estimated parameters and mean error;
- Measurement error in world points and the effect on estimated parameters and mean error; and
- Resolution error in image points and the effect on estimated parameters and mean error.

Each study describes the purpose, the variables, what effects are to be monitored, and the results. Results are normally presented in graphical form with accompanying comments.

Each study is designed to compare the estimated parameters with the known parameter values and to study the mean error as the known parameter values are accurately controlled over a specific range. For the conduct of the parametric studies in this chapter, the mean error will be used as the assessment of accuracy in back projection. In each of the studies, the application of distortion or error sources is applied over a single or dual range of values. Each change of variable is referred to as a trial. For explanation purposes, the estimated parameter values is plotted against single or dual trial values. Variables were modified in a single or double loop and in each case, an explanation as to how the variables were changed and in what order is provided in the explanation of the study. If the trial number is referred to, it implies one iteration of a loop in the evaluation code.

To generate consistent input data, the parameter values are required to be known. Table 5.1 lists the camera parameters that are held constant for the generation of the input data throughout each study.

Table 5.1 Camera parameter used to generate input data for each study.

Parameter	Value
f	0.012m
s_r	1.0
R	$\begin{bmatrix} 0.2182 & -0.8944 & 0.3904 \\ 0.4364 & 0.4472 & 0.7807 \\ -0.8729 & 0 & 0.4480 \end{bmatrix}$
T	$[3 \ 2 \ 4]^T$
Jig Centre	$[4 \ 4 \ 0]^T$

In addition to the four studies with MatCam, a section is included of an actual camera calibration of six trials using Tsai's method. Each trial used the same camera and

world points with the camera position being the variable. Estimated parameter values are included for evaluation in Section 5.6.

5.2 Radial Distortion Tests

Tsai's non-coplanar calibration method is based upon a radial alignment constraint that considers radial distortion only. The method was published [2] considering only the first order coefficient of radial distortion (κ_1) but the implemented version used in this study has included the second order term (κ_2) [41]. The purpose of this radial distortion test is to study how accurately the calibration method estimates the parameters that were used to generate the input data when only radial distortion of the first and second order are applied and to compare the mean error between world points and back projected points.

Test Name: Parameter estimation reaction to varying κ_1 and κ_2 .

Objective: To confirm that the camera model correctly determines the intrinsic and extrinsic parameters of the camera system when κ_1 and κ_2 values are varied for the generation of input data, with no additional error sources, and all other parameters used to generate the input data are held constant. The mean error of each trial will also be evaluated.

Variables: κ_1 - Range from -400 to 3000 in step sizes of 200.

κ_2 - Range from 0 to 1.0e+07 in step sizes of 1.0e+06.

κ_1 was the variable in the inner loop and κ_2 was varied in the outer loop.

Test results: The camera parameters and back projection points were correctly estimated in every arrangement of κ_1 and κ_2 . Test results are presented in graphical form with the trial number as the horizontal axis and the value of

the difference between the actual and estimated parameter values as the vertical axis. The mean error values correspond to the mean error of the back projected points when using the calibration method and the world points. A mean error is also calculated for the IPP (inverse perspective projection) of the input data compared to the world points and used as a comparative tool.

First Order Radial Distortion (κ_1) – The maximum error in the difference between the estimated value and input value of κ_1 is $3.39\text{e-}07 \text{ m}^{-2} \equiv 0$, see Figure 5.1. Therefore, this parameter is assessed as having been properly estimated.

Figure 5.1 Difference between actual and estimated κ_1 .

Second Order Radial Distortion (κ_2) – The maximum error in the difference between the estimated value and input value of κ_2 is $0.0082 \text{ m}^{-4} \cong 0$, see Figure 5.2. Therefore, this parameter is assessed as having been properly estimated.

Figure 5.2 Difference between actual and estimated κ_2 .

Effective Focal Length (f) – Figure 5.3 illustrates the accuracy by which the calibration method estimated the effective focal length of the camera system. The maximum error in the difference between the estimated value and actual value of f is $6.71\text{e-}012\text{m} \cong 0$. Therefore, this parameter is assessed as having been properly estimated.

Horizontal Scale Factor (s_x) – The maximum error in the difference between the estimated value and actual value of s_x is $1.06\text{e-}014 \cong 0$, see Figure 5.4. Therefore, this parameter is assessed as having been properly estimated.

Figure 5.3 Difference between actual and estimated f .

Figure 5.4 Difference between actual and estimated s_x .

Elements of the Translation Vector – As can be seen in Figure 5.5, the translation vector was accurately estimated when radial distortion was applied. Note that the value of T_z appears to be separate from T_x and T_y which appear to “stick” together. As T_x and T_y are estimated in stage one of Tsai’s method and T_z is estimated in stage two, there appears to be a compensation for this. This observation is premature at this time, however it should be noted for subsequent studies.

Rotation Matrix Convergence – As a measure of how accurately the rotation matrix is estimated, the scaled Frobenious norm is applied to the difference between the actual and estimated R . This difference is used in each study and Figure 5.6 illustrates the error between actual and estimated R . The scale of measure is between 0 to 1 with 0 being the most accurate match. Figure 5.6 illustrates that R was accurately estimated by the calibration method.

Figure 5.5 Difference between actual and estimated T .

Figure 5.6 Rotation matrix convergence.

Mean Error - Figure 5.7 illustrates the mean error for each estimation of parameters within the study. Figure 5.7 illustrates that the largest mean error for each calibration run. When using the inverse perspective projection of the input data, the mean error was 67.3mm. The mean error for each calibration run was insignificant when back projecting using the estimated parameters. Also observed in Figure 5.7 is that the values of κ_1 and κ_2 were varied in a double loop with the value of κ_1 being the inside loop and appears to have a greater effect on the mean error than κ_2 . The "saw-tooth" indicates that mean error rises as κ_1 increases and κ_2 has little effect in comparison. This suggests that Tsai's assessment as to the relative significance of κ_2 is correct in that it may be excluded for the estimation of parameters, however it will continue to be a parameter of study in this chapter.

Figure 5.7 Magnitude of actual and estimated mean error.

Summary of Effects: The maximum mean error for all trials was $\cong 0$. Based on the error values, the calibration method is considered to have interpreted the input data and back projected accurately. The camera parameter values, when first and second order radial distortion are present, are also accurately estimated by the method. It was evident that the calibration method accurately assimilated the data and estimated the camera parameters correctly.

5.3 De-centering Distortion

The decision to exclude de-centering distortion in some camera models has been based on the relative effect de-centering distortion has on the overall accuracy of the model [2], [32]. De-centering distortion is similar to adding a prism to the lens and creating a “table-tilt” effect. What has not been evaluated is the effect de-centering distortion has on the accuracy of parameter estimation in Tsai’s model.

This de-centering distortion study was conducted over several ranges of κ_4 and κ_5 , with κ_1 and κ_2 set to constant values. Values for κ_1 and κ_2 were specifically chosen to be non-zero, as based on literature reviewed, the exclusion of radial distortion is not probable [1]. Parameter values for the generation of input data are in accordance with Table 5.1 and were held constant.

Test Name: Effect of de-centering distortion of estimated parameter accuracy and mean error accuracy.

Objective: Using a range of coefficients of de-centering distortion, analyze the estimated parameter values of Tsai's method and compare to actual values and conduct a comparison of mean error between calibration method and inverse perspective projection of input data.

Variables: De-centering distortion coefficient κ_4 with a range from 0 to 2.0 in steps of 0.2.

De-centering distortion coefficient κ_5 with a range from 0 to 1.0 in steps of 0.2.

Values for κ_1 and κ_2 were used and held constant as 1000 and 1.0e+07 respectively. The study was conducted as a double loop with κ_4 in the inner and κ_5 in the outer loop.

Test Results: For this study, the results of the effect on each of the parameters are best displayed graphically. Surface plots are shown with κ_4 plotted on the x-axis and κ_5 on the y-axis, and the magnitude of the parameter under investigation is on the z-axis. Two-dimensional plots of each estimated

parameter are also provided with the estimated value on the vertical axis and the trial number of the κ_4 and κ_5 pairing on the horizontal axis.

Actual parameter values used to generate the input data are in the actual input column of Table 5.2. Two main columns of the table are sub-divided as min/max. The minimum value of each parameter from all trials is in the first main column. The maximum value of each parameter from all trials is in the second main column. Tsai's method does not estimate κ_4 and κ_5 , therefore these parameters are excluded from the table. The maximum values for changes in the position (offset) of the camera from original position are provided as $\max|T_{Error}|$. The maximum error in the rotation matrix convergence is $\max \Delta R$ with a value of 0 indicating no error between the estimated and actual R . The mean error values correspond to the mean error of the back projected points when using the calibration method and the world points. A mean error is also calculated for the IPP of the input data compared to the world points and used as a comparative tool.

Table 5.2 Summary of de-centering distortion study results.

	Input Value	Estimated Parameter Minimum Value	Estimated Parameter Maximum Value
κ_1	1000	-137.1460	2143
κ_2	1.0e+07	-2.485e+07	5.851e+07
f	0.0120m	0.0117m	0.0124m
s_r	1.0	0.9987	1.0027
T_r	3.0m	2.9531m	3.0282m

	Input Value	Estimated Parameter Minimum Value	Estimated Parameter Maximum Value
T_y	2.0m	1.9610m	2.0384m
T_z	4.0m	3.9093m	4.1267m
$ T_{Error} $	0	8.3e-14m	0.1390m
ΔR		6.8e-15	0.0018
Maximum Mean Error Calibration Method		8.8e-11mm	19.4mm
Maximum Mean Error IPP		23.9mm	38.8mm

First Order Radial Distortion (κ_1) – The representation of the effects on κ_1 are

displayed in Figure 5.8 and Figure 5.9 with the actual input value of $\kappa_1 = 1000$. The estimated value of κ_1 ranged from -137.1460 to 2143. Figure 5.8 illustrates a relatively smooth transition in the estimated value of κ_1 when the variables of κ_4 and κ_5 are increasing. Figure 5.9 illustrates a “saw-tooth” pattern as κ_5 is increased in the outer loop and κ_4 increases in the inner loop. Figure 5.9 is a better representation of the estimated value of κ_1 when κ_4 and κ_5 are used as variables. The saw-tooth is consistent until trial 33 when $\kappa_4 = 2.0$ and $\kappa_5 = 0.4$, then the value of κ_1 drops sharply and then rebounds in trial 34. This drop and rebound occurs again in trial 44. In trail 55 and 56, there is a double drop and rebound. In trial 55, $\kappa_4 = 2.0$ and $\kappa_5 = 0.6$ and in trail 56, $\kappa_4 = 0.0$ and $\kappa_5 = 1.0$.

Figure 5.8 Effect of de-centering distortion on κ_1 .

Figure 5.9 Effect of de-centering distortion on κ_1 .

Second Order Radial Distortion (κ_2) – The value of κ_2 exhibited a near linear positive trend from $-2.5e+07$ to $5.9e+07$, as shown in Figure 5.10 and Figure 5.11. Figure 5.10

suggests that the estimation of κ_2 is less sensitive to de-centering distortion where one or either κ_4 or κ_5 values are less than 0.4. Conclusive observations would require an extensive study of the interaction of κ_1 , κ_2 , κ_4 and κ_5 as these parameters are inter-related. This observation is supported by Figure 5.11 as the saw-tooth pattern is the inverse of the κ_1 pattern, as shown in Figure 5.9. An explanation of the mirrored patterns for κ_1 and κ_2 is that they are estimated in stage two of Tsai's method, and optimized to reduce the mean error.

Figure 5.10 Effect of de-centering distortion on κ_2 .

Figure 5.11 Effect of de-centering distortion on κ_2 .

Effective Focal Length (f) – The estimated value of f had a range of from 0.0117m to 0.0124m and exhibited a near linear trend as shown in Figure 5.12 and Figure 5.13. The overall change in value of f was less than 0.4mm. In Tsai's calibration method, the rotation matrix and T_x and T_y are estimated in stage one. As will be seen shortly, the estimated position of the camera is offset from the actual camera position defined by T . The focal length f is estimated in stage two of Tsai's method, and is dependant on the camera position which is partially defined in stage one of the method by estimating T_x and T_y . The estimated value of f must compensate for the change in distance of the camera to the world points. To test this observation, trial numbers 11 and 12, as shown in Figure 5.13 correspond to opposite points of a saw-tooth. The distance from the actual camera position to the centre of the calibration grid resulted in a vector length of 4.5825m. Using the value of T for trial number 11, the distance from the estimated

camera position to the centre of the calibration grid resulted in a vector length of 4.7177m and an f of 0.0124m. Using the value of T for trial number 12, the distance from the estimated camera position to the centre of the calibration grid resulted in a vector length of 4.5616m and an f of 0.0119m. The longer the distance from the estimated camera to the world points, the longer the estimated focal length. The cause of longer/shorter focal lengths is attributed to the change in camera position defined by T with T_x and T_y estimated in stage one and then held fixed and stage two parameters are optimized to reduce the mean error.

Figure 5.12 Effect of de-centering distortion on f .

Figure 5.13 Effect of de-centering distortion on f .

Horizontal Scale Factor (s_x) – The range for s_x was from 0.9987 to 1.0027 and also exhibited a near linear trend similar to f , as shown in Figure 5.14 and Figure 5.15. The horizontal scale factor is a scaling parameter and is estimated in stage one of Tsai's method. The shape of the estimated parameter value in Figure 5.15 is consistent with the camera position "moving" away from the calibration grid centre. The parameters T_x and T_y define the camera position in stage one of the calibration when s_x is estimated. Figure 5.17 and Figure 5.19 confirm that the estimated camera position is moving away from the calibration grid as values for T_x and T_y both decrease and the calibration grid position in the XY plane is $[-4 \ -4]^T$. This explains why s_x is closely related to T_x and T_y .

Figure 5.14 Effect of de-centering distortion on s_x .

Figure 5.15 Effect of de-centering distortion on s_x .

Translation Vector (T) – The translation vector T is used to track the “movement” of the camera and subsequent effect on other parameter values. The estimated components

of T are shown in Figure 5.16 to Figure 5.21. Figure 5.16, Figure 5.18, and Figure 5.20 illustrate near linear trends in the changes in T when applying de-centering distortion. In Figure 5.17 and Figure 5.19 that illustrate the values of T_x and T_y respectively, the camera “moves” away from the centre of the calibration grid centre when κ_4 increases and the camera moves closer to the calibration grid centre when κ_5 increases. There is also a relationship between T and f , as discussed above, as when T moves further from the centre of the calibration grid, f increases. Similarly, when T moves closer to the centre of the calibration grid, f decreases. Figure 5.24 illustrates the values of each component of T and the magnitude of offset, from the actual camera position, for each trial. The maximum offsets occur at trials, 11, 22, 33, 44, 55 and 66 and correspond to when κ_4 is at the maximum value of the variable range. From Figure 5.24, the offset in T increases as de-centering distortion coefficients increase, which is to be expected.

Figure 5.16 Effect of de-centering distortion on T_x .

Figure 5.17 Effect of de-centering distortion on T_x .

Figure 5.18 Effect of de-centering distortion on T_y .

Figure 5.19 Effect of de-centering distortion on T_y .

Figure 5.20 Effect of de-centering distortion on T_z .

Figure 5.21 Effect of de-centering distortion on T_z .

Figure 5.22 Effect of de-centering distortion on the magnitude of offset of T .

Figure 5.23 Effect of de-centering distortion on the magnitude of offset of T .

Figure 5.24 Components of T and magnitude of camera position offset due to de-centering distortion.

Rotation Matrix Convergence – The rotation matrix appears to be accurately estimated with the maximum magnitude of rotation matrix convergence as $0.0018 \cong 0$, see Figure 5.25. The rotation matrix is considered to have been accurately estimated.

Mean Error – The value of the mean error for each trial with de-centering distortion, is displayed in Figure 5.26 through Figure 5.30. These figures illustrate a linearly increasing trend with κ_4 and κ_5 demonstrating near equal effects to the mean error. The maximum mean error when back projecting is 19.4mm. The maximum mean error in the inverse perspective projection of the input data is 38.8mm. Figure 5.30 illustrates the mean error for both calibration method and IPP on the same figure and how these error assessment tools line up correctly. When comparing the maximum mean error between the calibration method and IPP, the calibration method appears to remove the radial distortion but does not completely compensate for de-centering distortion, which is expected as Tsai's model does not account for de-centering distortion.

Figure 5.25 Effect of de-centering distortion on R .

Figure 5.26 Effect of de-centering distortion on the mean error using calibration method.

Figure 5.27 Effect of de-centering distortion on the mean error using calibration method.

Figure 5.28 Effect of de-centering distortion on the mean error using inverse perspective projection and input data.

Figure 5.29 Effect of de-centering distortion on the mean error using inverse perspective projection and input data.

Figure 5.30 Comparison of mean error for calibration method and inverse perspective projection of input data when applying de-centering distortion.

Summary of Effects: The maximum mean error for all trials in this study was 19.4mm and the maximum mean error using inverse perspective projection of the input data was

38.8mm. Given the actual camera position to the centre of the world points, a distance of 4.4825m, the percent mean error for trials using the calibration method is 0.43%. The magnitude of error in T was at a maximum of 0.1390m when $\kappa_4=2.0$ and $\kappa_5=0.0$, see Figure 5.23, however, as the values of κ_4 and κ_5 approached being equal, the magnitude of the error in T was reduced and appeared as a “saddle”. This effect is shown in Figure 5.22. As shown in Figure 5.30, Tsai’s method reduced the error that would have occurred from inverse perspective projection by interpreting the majority of the radial distortion from the input data. As de-centering distortion includes radial and tangential distortion elements, the mean error did increase as de-centering distortion coefficients increased. As Tsai’s camera model and calibration method is based solely on radial distortion being present, this is a reasonable result.

5.4 Measurement Error in World Points and the Effect on Camera Parameters

From previous studies in this Chapter, Tsai’s non-coplanar calibration method correctly estimates the intrinsic and extrinsic parameters when input data has no distortion or error sources applied and when radial distortion, up to the second order distortion coefficient, is applied. It is probable that, in addition to radial distortion that is present in all cameras, that there will be error sources that will effect the degree of accuracy in measuring world points.

In this section, a study of measurement error in world point is conducted. The purpose of this study is to evaluate the effect of measurement error in world points on the estimated parameters and the accuracy of the mean error in back projected points.

Measurement error is applied as a truncated normal distribution about each world point with a separate random error value for X and Y axes and a Z axis standard error to all world points in the same plane. The standard plane error in the Z direction is based on using a calibration grid and using a single measurement to establish the elevation value. If an error was made measuring the elevation of the plane, then this would be applied to all points in the plane. As a control, no distortion or other error sources will be applied to the input data except measurement error. Parameter values are the same as in Table 5.1, with radial and de-centering distortion coefficients equal to zero.

Test Name: Effect of measurement error in world points on estimated parameters and mean error in back projection.

Objective: While holding the input values of radial and de-centering distortion equal to zero, apply a random measurement error to world points and observe the values of the estimated parameters and mean error in back projection.

Variables: Error in X and Y directions. The maximum allowable error about a world point in the X and Y directions was limited to a range of 0 to 20mm and a step size of 2mm. The error is applied as a normal distribution and is truncated to a value of ± 1.0 to prevent severe outliers from entering the data.

Error in elevation Z . Maximum error in the elevation of each plane of world points is applied consistently to all world points in the plane. The maximum error in elevation was limited to a range of 0 to 2mm and a step size of 1mm .

The variables were applied in a double loop format with the change in elevation in the outside loop and the change in the maximum allowable error in the XY plane in the inner loop.

Test Results: The results of the tests in this section are presented as surface and two-dimensional plots. Surface plots represent the variables of error in X and Y directions along the x-axis and elevation error in the y-axis. The estimated value of the parameter being investigated or magnitude of error is plotted on the z-axis. A summary of the results of the study is shown in Table 5.3 which provides a range of estimated parameter values. Actual parameter values used to generate the input data are in the actual input column of the table. Two main columns of the table are sub-divided as min/max. The first main column provides the maximum and minimum value of each parameters when the error is in the $X - Y$ plane with no error in elevation of the world points. The second main column is similar except the error in elevation of world points is at the maximum error of 2mm. Note that Tsai's method does not estimate κ_4 and κ_5 and these values are excluded from Table 5.3. The maximum values for changes in the position of the camera from original position are provided as $\max |T_{Error}|$. The maximum error in the rotation matrix convergence is $\max \Delta R$ with a value of 0 indicating no error between the estimated and actual R . The mean error values correspond to the mean error of the back projected points when using the calibration method and the world points. A mean error is also calculated for the IPP of the input data compared to

the world points and used as a comparative tool. The estimated parameters did exhibit a common trend when the estimated values started to diverge from actual parameter values and this was in the 6-8mm range. This suggests that the calibration method may compensate for random measurement errors of up to 6mm and this may be considered as a error range for the calibration method, given a similar calibration setup.

Table 5.3 Summary of measurement error results.

	Actual Input	Estimated Parameter Value with $X - Y$ Error and Error in $Z = 0$		Estimated Parameter Value with Maximum $X - Y$ Error and Error in $Z = 2\text{mm}$	
		Min	Max	Min	Max
κ_1	0	-555.6440	538.8605	-1.1957e+03	31.8807
κ_2	0	-1.808e+07	1.687e+07	-3.625e+06	3.405e+07
f	0.0120m	0.0116m	0.0122m	0.0118m	0.0122m
s_x	1.0	0.9967	1.0074	0.9969	1.0022
T_x	3m	2.9398m	3.0794m	2.9448m	3.0538m
T_y	2m	1.9855m	2.0770m	1.9293m	2.0261m
T_z	4m	3.9329m	4.0264m	3.9745m	4.0434m
$\max T_{Error} $		0.1294m		0.0795m	
$\max \Delta R$		0.0067		0.0053	
Maximum Mean Error Calibration Method		16.5mm		16.5mm	
Maximum Mean Error IPP		14.3mm		15.8mm	

First Order Radial Distortion (κ_1) – The estimated value of κ_1 is illustrated in Figure 5.31 and Figure 5.32 with oscillations about the actual value of κ_1 . The estimated value of κ_1 responds similarly to other parameters in that the increases in the range of oscillations commences at trial 13 of Figure 5.32 when the maximum error in the

XY plane is 8mm. It appears that 8mm of random measurement error for this particular camera set up is a maximum range of allowable error.

Figure 5.31 Measurement error in world points and effect on κ_1 .

Figure 5.32 Measurement error in world points and effect on κ_1 .

Second Order Radial Distortion (κ_2) – The estimated value of κ_2 is illustrated in Figure 5.33 and Figure 5.34. Similar to κ_1 , the estimated values of κ_2 oscillate about the actual value of 0 distortion. The oscillations remain relatively small until trial 13, when maximum error in the XY plane is 8mm and elevation error is 0, as shown in Figure 5.34, and then the oscillations increase in range. This suggests that 8mm of random measurement error for this particular camera set up is a maximum range of allowable error.

Figure 5.33 Measurement error in world points and effect on κ_2 .

Figure 5.34 Measurement error in world points and effect on κ_2 .

Effective Focal Length (f) – The estimated value of f is illustrated in Figure 5.35 and Figure 5.36. The estimated effective focal length value oscillated about the actual value of f , 0.012m, as shown in Figure 5.36 with no significant increase in range until after trial 13 when the maximum error in the XY plane is 8mm. This observation is consistent with other estimated parameters, in particular with T . This suggests that 8mm of random measurement error for this particular camera set up is a maximum range of allowable error.

Figure 5.35 Measurement error in world points and effect on f .

Figure 5.36 Measurement error in world points and effect on f .

Horizontal Scale Factor (s_x) – The estimated value of s_x is illustrated in Figure 5.37 and Figure 5.38. The estimated value of s_x oscillates closely about the actual value of 1.0, up to the period when the measurement error in the XY is 16mm, trial 24, and the range of oscillation increases thereafter, see Figure 5.38. The range of estimated values for s_x is quite small with maximum deviation of 0.0031 from actual value of 1.0. The maximum deviation in the estimated value of s_x is insignificant and this makes sense, since scanning frequencies in the camera and computer frame memory should not depend on random measurement error.

Figure 5.37 Measurement error in world points and effect on s_x .

Figure 5.38 Measurement error in world points and effect on s_r .

Translation Vector (T) – A long series of following figures, from Figure 5.39 to Figure 5.47 illustrates the effect of measurement error on the estimated parameter T . The individual components of T are illustrated separately, as well as the magnitude of offset from the estimated value of T to the actual value of T , the camera position. In general, the estimated camera position was offset a maximum of 57.0cm when both maximum measurement error in the XY plane and an elevation error in Z were applied. When maximum measurement error was applied in the XY plane only, the maximum camera offset was 12.94cm. In either case, the estimated camera position was offset from the actual position. The mean error for the calibration method was 16.5mm and the mean error for the IPP was 15.8mm. The values of T_x are shown in Figure 5.39 and Figure 5.40 where the value for T_x oscillated above and below the actual value of T_x as shown in Figure 5.40. Figure 5.40 also illustrates that the estimated value of T_x remains

relatively consistent until trial number 13 when the oscillations increase in range. The error values applied for trial 13 are an 8mm maximum error in the XY plane and 0 elevation error. Similar responses to increased measurement error were noticed in T_y . Figure 5.41 and Figure 5.42, and T_z . Figure 5.43 and Figure 5.44, respectively. Figure 5.45 and Figure 5.46 illustrate the magnitude of the offset in T and how the error appears to be increasing in a non-linear trend. Figure 5.47 illustrates all components of T and the magnitude of error plotted simultaneously with the trial number. When the offset in individual components T_x and T_y increase, the offset of T_z increases in greater proportion compared to T_x and T_y as shown in Figure 5.47. Figure 5.48 is difficult to follow in detail, however the figure does illustrate that the estimated camera position moves when the maximum measurement error is increased. Three separate paths are followed, grouped to changes in elevation error as outlined in Table 5.4.

Figure 5.39 Measurement error in world points and effect on T_x .

Figure 5.40 Measurement error in world points and effect on T_x .

Figure 5.41 Measurement error in world points and effect on T_y .

Figure 5.42 Measurement error in world points and effect on T_y .

Figure 5.43 Measurement error in world points and effect on T_z .

Figure 5.44 Measurement error in world points and effect on T_z .

Figure 5.45 Measurement error in world points and effect on the magnitude of error in T .

Figure 5.46 Measurement error in world points and effect on the magnitude of error in T .

Figure 5.47 Comparison of error in components of T and offset from actual T .

Figure 5.48 Tracking camera position movements due to measurement error in the world points.

Table 5.4 Legend for Figure 5.48.

Shape	Meaning
Square	Elevation change = 0.0m
Circle	Elevation change = 0.001m
Diamond	Elevation change = 0.002m

Rotation Matrix Convergence – The rotation matrix appears to be accurately estimated with the maximum magnitude of rotation matrix convergence as 0.0067 and the error associated with the estimated value of R is considered insignificant. Figure 5.49 illustrates the error in the estimated value of R compared to the actual value. The value of error in R remains almost zero until the measurement error in the XY plane exceeds 8mm and then the “spikes” in Figure 5.49 commence. This is consistent with the observations of other estimated parameters as well.

Figure 5.49 Measurement error in world points and effect on R .

Mean Error – The mean error for the calibration method is shown in Figure 5.50 and displays a consistent rise from 0 to 16.5mm. The mean error for the IPP of input data is shown in Figure 5.51 and displays a consistent rise from 0 to 15.8mm. In both cases, the maximum mean error is less than the range of maximum allowable measurement error in the world points. The overall scale of error is small, based upon the distance from camera to the world points. From Figure 5.50, it appears that the calibration method incorporates the change in elevation as the mean error values for the calibration method back projection appears consistent across the elevation changes. The IPP of the input data appears sensitive to the elevation changes with an increase in mean error as the elevation error increases, as shown in Figure 5.51. Figure 5.52 illustrates the mean error of both calibration method and IPP plotted together. The calibration method does not significantly reduce the mean error in comparison to the mean error of IPP using the input data.

Figure 5.50 Effect of measurement error in the world points on the mean error using calibration method.

Figure 5.51 Effect of measurement error in the world points on the mean error using inverse perspective projection.

Figure 5.52 Comparison of mean error between calibration method and IPP of input data.

Summary of Effects: Overall, random measurement error in the world points of 2cm in the XY plane results in less than 2cm of error when back projecting using the calibration method estimated parameters or using IPP of the input data. Similarly, a random measurement error in the world points of 2cm in the XY plane and 2mm in elevation results in a maximum mean error of less than 2cm for either back projection. Note however, how the magnitude of error in the translation vector increased as the maximum allowable measurement error increased. This indicates that the estimated position of the camera is “moving” from the actual position, up to 12.94cm, when there is no elevation error in the world points and up to a maximum of 7.95cm when a maximum 2mm error is applied to all world point elevation values. There is some consistency in the estimated parameters and the range of maximum measurement error applied. It appears that κ_1, κ_2, f and T all estimate values close to their actual values until the maximum error in the XY plane is 6-8mm. The mean error of the calibration method is 5.9mm and the

mean error of the IPP is 5.65mm when the maximum measurement error is 6-8mm. This appears to place a maximum allowable measurement error of 6-8mm, given a similar physical camera system as used in this study.

5.5 Resolution Error in Image Point Positions and the Effect on Camera Parameters

Resolution error is a source of error due to the requirements for human interaction with a calibration method, and in this study, with Tsai's calibration method. An operator is required to enter the pixel coordinates of the world point as viewed in the image plane (image point). As discussed in Section 2.7.1, this may not be straight forward. Figure 2.11 is an example of an image point enlarged and the centre of the image point is represented by 5x3 array of pixels. The selection of the centre of the image point is challenging, even when the image is enlarged, and subject to error. Using Figure 5.53 as an example of a calibration grid, the selection of the centre of the world points in the image plane can be off by numerous pixels, and this is represented as resolution error.

To conduct the study, the application of resolution error is in the x and y directions of the image plane. The selection of the maximum number of pixels of error in both x and y directions is preset and then a truncated normal distribution is used to select the actual pixel error for the image point.

There are calibration methods that select image points automatically [12], however, edge detection methods are utilized. Edge detection for the selection of the centre of image points, such as those in Figure 5.53 illustrates that automatic aides to

calibration may not be applicable as the edges of the image points “blend” with the background.

Figure 5.53 Image of a calibration grid used in a camera calibration experiment.

Test Name: Effect of resolution error on Tsai’s non-coplanar calibration method.

Objective: While holding the input values of radial and de-centering distortion equal to zero, apply a resolution error to image points and observe the values of the estimated parameters and mean error of the back projected points.

Variables: Image points are modified in x and y directions by the application of a maximum error in each direction limited to a range of 0 to 10 pixels with a step size of 0.5 pixels. Resolution error is applied using a truncated normal distribution by limiting the variance to 1. All other parameters to generate the input data are keep constant, as in Table 5.1, and no distortion or other sources of error are applied.

Test Results: The results of the tests in this section are presented as 2D plots with the maximum allowed pixel error per trial along the x-axis and estimated parameter value on the y-axis. A summary of the results of the study is shown in Table 5.5 which provides a range of values for the actual input and estimated parameters. Note that κ_4 and κ_5 are input as zero and are not estimated in Tsai's method and therefore excluded from the table. The maximum values for changes in the position of the camera from original position are provided as $\max|T_{Error}|$. The maximum error in the rotation matrix convergence is $\max \Delta R$ with a value of 0 indicating no error between the estimated and actual R . The mean error values correspond to the mean error of the back projected points when using the calibration method and the world points. A mean error is also calculated for the IPP of the input data compared to the world points and used as a comparative tool.

Table 5.5 Summary of resolution error results.

	Input Value	Estimated Parameter Minimum Value	Estimated Parameter Maximum Value
κ_1	0	-4.1462e+03	167.0200
κ_2	0	-8.20e+06	1.16e+08
f	0.0120m	0.0109m	0.0123m
s_x	1.0	0.9825	1.0193
T_x	3.0m	2.8309m	3.0912m
T_y	2.0m	1.8983m	2.2506m
T_z	4.0m	3.8577m	4.0595m
$ T_{Error} $	0	1.0e-13m	0.2907m
ΔR		2.6e-15	0.0218
Maximum Mean Error		1.9e-11mm	51.9mm

	Input Value	Estimated Parameter Minimum Value	Estimated Parameter Maximum Value
Calibration Method			
Maximum Mean Error IPP		3.3e-13mm	50.1mm

First Order Radial Distortion (κ_1) – The estimated value of κ_1 is illustrated in Figure 5.54 and decreases in a linear trend with oscillations. Figure 5.53, and other figures in this section on resolution error, do not suggest a trend where the maximum pixel value has a greater effect on mean error or accuracy in estimating the camera parameters. The values of κ_1 and κ_2 appear to be related as the shape of the respective graphs appear mirrored.

Second Order Radial Distortion (κ_2) – The estimated value of κ_2 is illustrated in Figure 5.55 and reacts in a very similar manner to κ_1 , increasing with a linear trend with oscillation, and mirroring κ_1 .

Figure 5.54 Resolution error in image points and effect on κ_1 .

Figure 5.55 Resolution error in image points and effect on κ_2 .

Effective Focal Length (f) – The estimated value of f is illustrated in Figure 5.56 and oscillates about the actual value of f until the maximum pixel error reaches 5 pixels.

Figure 5.56 Resolution error in image points and effect on f .

Horizontal Scale Factor (s_x) – The estimated value of s_x is illustrated in Figure 5.57 and oscillates about the actual value throughout the range of the study.

Figure 5.57 Resolution error in image points and effect on s_x .

Translation Vector (T) – The error in the estimated values of the individual components of the translation vector are shown in Figure 5.58 through Figure 5.60 and all three components together are shown in Figure 5.61. Similar to the other parameters in this study, the values of each component T oscillate about the actual value but no clear pattern is apparent. The magnitude of the offset in T , difference in actual camera position to estimated position, is shown in Figure 5.62. The magnitude of the offset in T also does not provide clear, distinguishing characteristics from which to base conclusions. Figure 5.63 illustrates the movement of the camera position and does

indicate substantial adjustment of T in an effort to reduce the mean error in back projection.

Figure 5.58 Resolution error in image points and effect on T_x .

Figure 5.59 Resolution error in image points and effect on T_y .

Figure 5.60 Resolution error in image points and effect on T_z .

Figure 5.61 Resolution error in image points and effect on T .

Figure 5.62 Resolution error in image points and effect on the magnitude of error in T .

Table 5.6 Legend for Figure 5.63.

Shape	Meaning
Square	Camera position
Circle	Initial camera position

Figure 5.63 Tracking camera position movements.

Rotation Matrix Convergence – The rotation matrix appears to be quite accurately estimated, as shown in Figure 5.64, with the maximum magnitude of rotation matrix convergence as 0.0218 and the error associated with the estimated value of R is considered insignificant.

Figure 5.64 Resolution error in image points and effect on the magnitude of error in R .

Mean Error – The mean error per change in maximum pixel error is shown in Figure 5.65 and appears to rise linearly with both the calibration method and IPP results exhibiting the similar slope and magnitude. The maximum mean error for the calibration method was 51.9mm and the maximum mean error for IPP of the input data was 50.1mm. The camera parameters do not exhibit any strong trends from which to base any conclusions with regards to resolution error limits. From the linear trend in the increase in mean error, Tsai's calibration method does not interpret randomly applied resolution error and any resolution error will be added to the overall mean error. Therefore, every

effort should be made to minimize resolution error when selecting the centre of image points in the calibration process.

Figure 5.65 Effect of resolution error in the image points on the mean error.

Summary of Effects: Overall, the values of the estimated parameters vary substantially from the actual parameter values. Tsai's calibration method did not correctly assimilate the errors in resolution and the mean error per trial was on average, actually greater than if inverse perspective projection of the input data was used alone. From the linear trend in the increase in mean error, and the lack of consistent trends in estimated camera parameters, Tsai's calibration method does not respond well to randomly applied resolution error and any resolution error would be added to the overall mean error. Therefore, every effort should be made to minimize the error when selecting the centre of image points in the calibration process. Resolution error in the image points appears to have a significant impact on the accurate estimation of camera parameters and the mean error of the calibration.

A technique to assist in the improvement of accuracy for the selection of the image point centre is an averaging technique. The operator would select multiple coordinates as close to the centre of the image point as possible and then use the mean of all the entries. This would allow the operator to select coordinates “around” the true image point centre and should reduce the resolution error. This technique can only reduce the error, not eliminate it.

5.6 Actual Camera Calibration Using Tsai’s Method

In addition to the four studies of Tsai’s method using controlled input data, a study of Tsai’s method with an actual calibration was conducted. The program software and code used in this section is the same as that used for MatCam. This provides confirmation that Tsai’s method is properly implemented for an actual calibration and for MatCam as well.

This study used six trials with the variables being the position and orientation of the camera. As the actual camera parameters are not known, the parameters from the calibration are estimates. The camera position and calibration jig, as shown in Figure 5.66 and Figure 5.67, were measured with a tape measure.

Figure 5.66 View of camera and calibration jig.

The calibration jig is a $\frac{3}{4}$ inch sheet of plywood, reinforced to maintain a consistent planar surface. To elevate the calibration jig for three non-coplanar stages, the blocks, as shown in Figure 5.67, were used. The base level is the jig on the “flat-earth” plane, level one is with the blocks in a horizontal orientation and level two is with the blocks in a vertical orientation. An arbitrary origin of F_w is selected outside the field of view of the camera, and in Figure 5.67, the origin of F_w is behind the camera and to the left. F_w is defined as: X positive right (along the bottom edge of Figure 5.67), Y axis positive forward (along the left edge of Figure 5.67) and the Z axis positive up from the calibration jig. The world points are the circular targets on the plywood, spaced 10x10cm from each other in 12 rows and 12 columns. The diameter of each circular target is 1cm.

Figure 5.67 Calibration jig with elevation blocks.

As the camera used is a "box" camera, as shown in Figure 5.68, it was not possible to measure to the centre of the camera reference frame (inside the camera), and therefore, the "measured" value of T , see Table 5.7, is an approximate value. The rotation matrix is also not known, and difficult to measure for similar reasons and therefore R is not included in the table. Information provided by the manufacturer of the camera is: a camera resolution 640x480, the CCD array size of one-third of an inch, and an approximate focal length of 6mm.

The computer frame memory is 640x480 and the centre of the frame memory was selected as the principal point. C_x is 320. C_y is 240.

Figure 5.68 Box camera on a tripod.

Six trials were conducted with the estimated parameter values as shown in Table 5.7. A description of each trial set up is as follows:

Trial 1. Camera set in a position, as shown in Figure 5.66, with the the tripod legs fully extended, with the camera looking along the centre of the calibration jig, in the base level.

Trial 2. The camera is positioned similar to trial 1, except the elevation of the camera is lowered and the angle of the camera is changed to continue looking at the centre of the calibration jig in the base level.

Trial 3. The camera is positioned similar to trial 2, the elevation remains constant and the angle of the camera is modified such that the optical axis intersects the calibration jig further away from the camera, thereby reducing the number of image points in the top portion of the field of view.

Trial 4. The camera is elevated to the fully extended position of the tripod and the camera angle is a “high, look down” angle. An approximate down angle (pitch) of 60-70° was noted.

Trial 5. The camera elevation and look down angle were maintained similar to trial 4, with an attempt to translate the camera along the X_w axis only, placing more image points on the left side of the image plane than on the right.

Trial 6. The camera position is similar to trial 5 with an attempt to rotate the camera in a panning motion counter-clockwise and creating a more balanced distribution of image points in the field of view.

Table 5.7 Summary of actual calibration trials.

	Trial Number					
	1	2	3	4	5	6
World Points	195	196	167	162	169	160
κ_1	-3194.5	-1741.1	-9904.8	-7910.5	-1271.4	5730.0
κ_2	5.29e08	9.67e08	5.04e08	1.00e09	3.07e08	1.15e08
f	5.8mm	5.4mm	5.3mm	5.4mm	5.2mm	6.3mm
s_r	1.0181	1.0064	0.9864	1.0247	1.0086	0.9934
T (m)	0.9754 0.2937 0.9678	0.6994 0.2351 0.7169	0.7135 0.2601 0.6736	0.8709 0.5933 1.2065	1.0890 0.5077 1.0982	1.0304 0.4636 1.2527
Measured T (m)	0.945 0.287 0.916	0.718 0.276 0.685	0.761 0.255 0.778	0.890 0.545 1.295	1.080 0.545 1.295	1.030 0.545 1.300
$ T_{Error} $	0.060m	0.055m	0.115m	0.103m	0.200m	0.094m
ME	8.9mm	9.6mm	7.5mm	22.2mm	8.0mm	7.9mm

The estimated parameters from the trials are fairly consistent given the quality of resolution of the camera, see Figure 2.9 for an example from the actual calibration. The estimated camera position, as defined by T , is different than the measured position and based upon the studies conducted previously, this is expected. The mean error from each

trial is relatively small, except trial 4. It was noted that during the conduct of the trials, matching image points to world points, as is necessary for the implementation process, can be difficult. On a number of occasions, values were entered incorrectly (mistakes) and this did have a significant effect on the estimated parameter values and mean error. This mis-entering one image point coordinate increased the mean error of the trial by up to 4mm. The data from trial 4 was checked for mistakes and none were found, leaving the large mean error unexplained.

The value of MatCam in a calibration of this type could be used to simulate the estimated parameters and conduct studies on the parameters and camera set up.

5.7 Chapter Summary

The objectives of each of the four main studies conducted were to use the strength of MatCam and evaluate Tsai's calibration method using controlled input data to investigate the effect on estimated parameter values and the effect on the mean error accuracy. In each study conducted, Tsai's model produced excellent mean error accuracy where the maximum mean errors were less than 51mm in all studies. In some of the studies, the variable ranges were extended beyond what would be considered reasonable in an properly manufactured camera system and the error beyond an experienced operator. However, the ability to study extended ranges of input data and error sources provided insight as to how the calibration method reacted to controlled inputs that are not simply replicatable in physical systems.

The radial distortion study confirmed that Tsai's method responds accurately to radial distortion, as would be expected given the manner that Tsai defines his camera model.

The de-centering study was the first study to show how parameter values, such as the translation vector, appear to adjust to compensate for other parameter values, with the goal of minimizing the mean error.

The measurement error study was the only study to provide data that supported a range of maximum measurement error for a given camera set up. An observation from the study was that the maximum error in measuring world points should be 6-8mm for the camera set up as defined in the study. This type of information may be useful if manufacturers provide approximate values for camera parameters and a MatCam simulation could be conducted to test the measurement accuracy required for future camera use.

The study of resolution error in selecting image points indicated that the accurate selection of image point centres is important as any error is directly passed on to the mean error.

The use of MatCam with Tsai's calibration method was valuable to evaluate Tsai's method and also to highlight the capabilities of MatCam as an evaluation tool of camera calibration methods.

Chapter

6 Conclusions and Proposals For Future Work

The development of a camera calibration method is based upon a camera model. The camera model is a mathematical abstraction of a camera. Camera calibration methods rely on this mathematical foundation to relate the 2D image to the 3D world. So long as the camera model can be defined mathematically, MatCam can interface with camera calibration methods to provide a valuable analysis tool to the model developer.

Physical camera systems must be calibrated which involves the estimation of camera parameters that define the camera model. These estimated parameters are required to back project the image points into the 3D world. Currently, the primary assessment of accuracy of a calibration method is the mean error, the difference between world points and back projected points. The assessment of accuracy in the camera parameters has not been a priority in research and is accepted as difficult to achieve. However, valuable information can be gained about a calibration method if actual camera parameter values are known.

MatCam is a simulated computer based camera model, that accurately generates controlled input data for the calibration method with the aim of studying not only the mean error, but the accuracy in estimating camera parameters.

In Section 4.3, MatCam accurately simulated radial and de-centering distortion as well as sources of error that are probable to exist within a physical camera. The key to the accurate simulation of input data is the ability to generate distorted image data from undistorted data. The generation of the input data is based on a Gauss-Newton optimization of undistorted image points into distorted image points. Assessments as to the accuracy of the distorted image data was conducted and proven to be correct.

By carefully controlling the generation of the input data, the calibration method can be evaluated for accuracy in interpreting distortion and the effect of error sources on the calibration method. This was observed during the study of measurement error where a possible maximum measurement error limit was detected.

MatCam is also based on a broad range of camera models that makes it flexible in the analysis of multiple calibration methods. This flexibility is useful in two ways. The first is that a calibration method can be evaluated over a wider range of distortion than would normally be possible with physical camera equipment. Secondly, the calibration method can be evaluated for its reaction to sources of error by studying the estimated parameters and mean error. Flexibility is provided by the ability to apply distortion and sources of error either singularly, or in combination, over a wide range of values and assess calibration methods. During several of the studies conducted, the estimated position of the camera was offset from the actual position used to generate the input data. This observation is important in that the accuracy of the mean error and camera

parameters apply to the current set up only and any change to the set up requires a new calibration process. Observations such as these are difficult to substantiate when the camera parameters are estimated only with no base of comparison.

MatCam provides three major contributions to the camera calibration community: an accurate means to generate synthetic data, a broad range camera model and a useful tool for the analysis and development of camera calibration methods.

As discussed above, the synthetic data, input data, is useful to all calibration methods and can assist in the development of new calibration methods. Prior to this work, there was no known source of accurate synthetic data available for camera calibration studies. During the literature search for this thesis, it was observed that the majority of calibration methods are similar and in particular, the mathematical models used for various components of camera models are the same. For example radial and de-centering distortion, when used in the camera model, were defined the same in all models. The development of a broad range of camera models, based on a strong mathematical foundation, provides flexibility for integration with many calibration methods.

The studies conducted in this thesis are sample studies only. Much more work is recommended with greater analysis into the reaction of calibration methods to distortion and sources of error. It is recommended that monte carlo simulations be utilized to expand the range and depth of simulations.

During the evaluation of Tsai's calibration method, two studies were initiated but were not completed: a study of the effect of incorrect selection of the principal point and a study on image modification. General observations are below.

Principal point offset: In [7] it was stated that the selection of the principal point could be done using 15 different methods, with each method providing a different result. A study was designed and executed where the actual principal point was defined by the numerical average of the CCD array. A simulation was executed that offset the principal point and an analysis of mean error and parameter estimation was conducted. In this study, the camera “movement” was extensive and the mean error was small. All of the estimated parameters did not match the actual parameters and yet the mean error was excellent. This was another indication of how the estimated camera parameters cannot be accepted, but estimated. This study could be pursued as future work.

Image modification: In Tsai’s paper [2] it was stated that an effort should be made to ensure complete field of view coverage of world points in the image plane. An initial study was conducted with radial distortion present. As long as there were the minimum number of calibration points and full image coverage, the estimated parameters were exact. When up to 7/16 of the top rows of the image were removed, the mean point error was less than 7mm. When the top half image points were removed, the mean point error escalated to 7.037e03mm and estimated parameters were similarly incorrect. This study could be completed as future work.

At present, MatCam integrates with Tsai’s non-coplanar calibration method. Future work would be to integrate MatCam with several calibration methods for the purpose of more detailed study with multiple camera models and calibration methods.

References

- [1] C.C. Slama, C. Theurer, S.W. Henriksen, *Manual of Photogrammetry*, American Society of Photogrammetry, Falls Church, 4th edition, 1980.
- [2] R.Y. Tsai, "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, 1987, pp 323-344.
- [3] Y.I. Abdel-Aziz, H.M. Karara, "Direct linear transformation into object space coordinates in close-range photogrammetry," Proceedings of the Symposium on Close-Range Photogrammetry, Falls Church, 1971, pp 1-18.
- [4] K.M. Dawson-Howe, D. Vernon, "Simple Pinhole Camera Calibration," International Journal of Imaging and Technology, Vol. 5, No. 1, 1994, pp 1-6.
- [5] D.C. Brown, "Lens Distortion for Close Range Photogrammetry," Photogrammetric Engineering, Vol. 37, No. 8, 1971, pp 855-866.
- [6] D.C. Brown, "De-centering Distortion of Lenses," Photogrammetric Engineering, Vol. 32, No. 3, 1966, pp 444-462.
- [7] R.G. Wilson, S.A. Shafer, "What is the center of the image?," Journal of the Optical Society of America, Vol. 11, No. 11, 1994, pp 2946-2955.

- [8] L.L. Wang, W.H. Tsai, "*Computing camera parameters using vanishing-line information from a rectangular parallelepiped.*" Machine Vision Applications, Vol 3, No. 3, 1990, pp 129-141.
- [9] I.W. Faig, "Calibration of Close-Range Photogrammetric Systems: Mathematical Formulation," Photogrammetric Engineering and Remote Sensing, Vol. 41, No. 15, 1975, pp 1479-1486.
- [10] J. Weng, P. Cohen, M. Herniou, "*Camera Calibration with Distortion Models and Accuracy Evaluation.*" IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 14, No. 10, 1992, pp 965-980.
- [11] B. Prescott, G.F. McLean, "*Line-Based Correction of Radial Lens Distortion.*" Graphical Models and Image Processing, Vol. 59, No. 1, 1997, pp 39-47.
- [12] J. Heikkilä, O. Silvén, "*A Four-Step Camera Calibration Procedure with Implicit Image Correction.*" IEEE Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition, 1997, pp 1106-1112.
- [13] V. Aitken, K. Russell, J. McFee, "*Geometrical and Optical Calibration of a Vehicle-Mounted IR Imager for Landmine Localization.*" in Proceedings SPIE 14th Annual International Symposium on Aerospace/Defense. Sensing, Simulation and Controls, AeroSense 2000, Detection and Remediation Technologies for Mines and Minelike Targets V, Orlando FL USA, April 24-28, 2000.
- [14] S. Shih, Y. Hung, W. Lin, "*When should we consider lens distortion in camera calibration.*" Pattern Recognition, Vol. 28, No. 3, 1995, pp 447-461.

- [15] J. Lai, "On the sensitivity of camera calibration," Image and Vision Computing, Vol. 11, No. 10, 1993, pp 656-664.
- [16] A. Cumani, A. Guiducci, "Geometric camera calibration: the virtual camera approach," Machine Vision and Applications, Vol. 8, 1995, pp 375-384.
- [17] H. Bacakoglu, M. Kamel, "A Three-Step Camera Calibration Method," IEEE Transactions on Instrumentation and Measurement, Vol. 46, No. 5, 1997, pp 1165-1172.
- [18] C. Chatterjee, V. Roychowdhury, "Algorithms for coplanar calibration," Machine Vision and Applications, Vol. 12, 2000, pp 84-97.
- [19] H. Zhuang, W. Wu, "Camera Calibration with a Near-Parallel (Ill-Conditioned) Calibration Board Configuration," IEEE Transaction on Robotics and Automation, Vol. 12, No. 6, 1996, pp 918-921.
- [20] T.A. Clarke, J.G. Fryer, "The Development of Camera Calibration Methods and Models," Photogrammetric Record, Vol. 91, 1998, pp 51-66.
- [21] J. Batista, H. Araújo, A. de Almeida, "Iterative Multistep Explicit Camera Calibration," IEEE Transactions on Robotics and Automation, Vol. 15, No. 5, 1999, pp 897-917.
- [22] P.F. Sturm, S.J. Maybank, "On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications," IEEE International Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, June 1999, pp 432-437.

- [23] Z. Zhang. "*Flexible Camera Calibration By Viewing a Plane From Unknown Orientation*," IEEE Proceedings of the 7th International Conference on Computer Vision, September 1999, pp 666—673.
- [24] <http://www.kwon3d.com/theory/dlt/dlt.html> , "*DLT Method*".
- [25] <http://www.kwon3d.com/theory/dlt/mdlt.html> , "*Modified DLT*".
- [26] <http://www.cs.berkeley.edu/~laura/cs184/quat/quaternion.html>. "CS184: "*Using Quaternions to Represent Rotation*".
- [27] J. Heikkilä. "*Geometric Camera Calibration Using Circular Control Points*," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 10, 2000, pp 1066-1077.
- [28] Z. Zhang. "*A Flexible New Technique for Camera Calibration*," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 11, 2000, pp 1330-1334.
- [29] D.C. Lay, *Linear Algebra and its Applications*, Addison-Wesley Inc., New York, 2nd edition update, 1996.
- [30] R.K. Lenz, R.Y. Tsai, "*Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology*," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 10, No. 5, 1988, pp 713-720.
- [31] B. Etkin, *Dynamics of Atmospheric Flight*, John Wiley & Sons Inc., Toronto, 1972.
- [32] C.S. Fraser, "*Digital camera self-calibration*," ISPRS Journal of Photogrammetry & Remote Sensing, Vol. 52, No. 4, 1997, pp 149-159.

- [33] S.K Ghosh, *Analytical Photogrammetry*, Pergamon Press, 2nd Edition, 1988, pp 115-117.
- [34] R. Horaud, R. Mohr, B. Lorecki, "On Single-Scanline Camera Calibration," IEEE Transactions on Robotics and Automation, Vol. 9, No. 1, 1993, pp 71-75.
- [35] W.J. Smith, *Modern Optical Engineering*, SPIE Press•McGraw-Hill, New York, 3rd edition, 2000.
- [36] M.A. Penna, "Camera Calibration: A Quick and Easy Way to Determine the Scale Factor," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No. 12, 1991, pp 1240-1245.
- [37] M. Aggarwal, N. Ahuja, "Camera Center Estimation," IEEE Proceedings 15th Annual Conference on Pattern Recognition, 2000, Vol. 1, pp 876-880.
- [38] G.P. Stein, *Internal Camera Calibration using Rotation and Geometric Shapes*, MSc. MIT, 1993.
- [39] *MatLab*, Version 5.3, The MathWorks Inc., Natick MA, 1999.
- [40] V.C.Aitken, "Sliding Mode State Estimation for Nonlinear Discrete-time Systems: Applications in Image Sequence Analysis," PhD Disertation, Department of Systems and Computer Engineering, Carleton Univesity, 1995, pp245-246, 283-284.
- [41] V.C. Aitken, "Calibration, navigation, spatial correspondence and data fusion for the Improved Landmine Detection Project D6300," Defence Research Establishment Suffield, Contract No. W7702-7-R664.

Appendix A

Improvement Methods

Two improvement methods are used in this thesis and are explained below.

Newton's Method

Newton's method is explained in detail in [40]. The goal of Newton's method is to solve, in an iterative process, for a root of the function

$$G(y) \tag{A.1}$$

where (A.1) is defined as continuously differentiable within a defined region and there exists a point y^* , such that

$$G(y^*) = 0 \tag{A.2}$$

The Jacobian of G is

$$J_G(y) = \left[\frac{\partial G_i(y)}{\partial y_j} \right] \tag{A.3}$$

A Newton iterative solution is determined by

$$y^{(n+1)} = y^{(n)} - J_G^{-1}(y^{(n)})G(y^{(n)}) \tag{A.4}$$

where $n = 0, 1, 2, \dots$ such that

$$y^{(n+1)} - y^{(n)} \rightarrow 0 \tag{A.5}$$

then

$$y'' = y^* \quad (\text{A.6})$$

To prevent “over-shooting” the solution during the iterations, a step size may be included to limit the amount of improvement towards the local solution. Equation (A.4) can be written as

$$y^{(n+1)} = y^{(n)} - S J_G^{-1} \left(y^{(n)} \right) G \left(y^{(n)} \right) \quad (\text{A.7})$$

where $S \in (0, 1]$.

Gauss-Newton's Method

Gauss-Newton's method is explained in detail in [40] with the basis of the method as a least squares solution of an equation to minimize a cost function

$$\gamma(\xi) = \frac{1}{2} \xi^T P \xi \quad (\text{A.8})$$

where $P \in \mathbb{R}^{m \times m}$ and $\xi = F(x)$. Similar to Newton's method, the function must be differentiable within a defined region to determine the Jacobian (A.3)

$$J_\gamma(\xi) = \xi^T P \quad \text{and} \quad H_\gamma = P \quad (\text{A.9})$$

where H_γ is the Hessian of the cost function and

$$J_F = \frac{\partial F}{\partial x} \quad (\text{A.10})$$

The inclusion of the the Hessian is the main difference between the Gauss-Newton and Newton's method. The resulting iteration is

$$x^{(j+1)} = x^{(j)} - \left[J_F^T(x^{(j)}) H_\gamma J_F(x^{(j)}) \right]^{-1} J_F^T(x^{(j)}) H_\gamma F(x^{(j)}) \quad (\text{A.11})$$

where $j = 0, 1, 2, \dots$ such that

$$\mathbf{x}^{(j+1)} - \mathbf{x}^{(j)} \rightarrow \mathbf{0} \quad (\text{A.12})$$

Similar to equation(A.7), a step size may be added to (A.11) as

$$\mathbf{x}^{(j+1)} = \mathbf{x}^{(j)} - S \left[\mathbf{J}_F^T(\mathbf{x}^{(j)}) \mathbf{H}_\gamma \mathbf{J}_F(\mathbf{x}^{(j)}) \right]^{-1} \mathbf{J}_F^T(\mathbf{x}^{(j)}) \mathbf{H}_\gamma \mathbf{F}(\mathbf{x}^{(j)}) \quad (\text{A.13})$$

where $S \in (0, 1]$.

Appendix B

Software Code of MatCam

The software code will included in the Appendix in grouped functional areas similar to the diagram below:

Main Code:

Preset Values:

Grid Shapes:

Coordinate Modification:

Camera Model:

Error Analysis:

Main File: MatCam

```

% MatCam.m
%
% Purpose: Main file used to control execution of program.
%           MatCam is a program to develop integrated input data to test/conduct
%           accuracy and parametric studies of camera calibration models.
%
% Variables:
%           Camera Parameters:
%
%           f           effective focal length.
%           Cx          Principal point centre value for x-direction (columns/2) of computer frame
buffer.
%           Cy          Principal point centre value for y-direction (rows/2) of computer frame buffer.
%           sx          Horizontal scale factor.
%           dx          Pixel-to-pixel metric spacing in x-direction (column-to-column).
%           dy          Pixel-to-pixel metric spacing in y-direction (row-to-row).
%           Nex         Number of columns in the resolution of the camera sensor (CCD).
%           Nfx         Number of columns in the resolution of the computer frame buffer.
%           K1          1st order radial distortion coefficient.
%                       Magnitude keep between  $-1.e+03 < K1 < +1.e+03$ 
%                       Negative K1 gives stronger pin-cushion distortion
%                       Positive K1 gives stronger barrel distortion
%           K2          2d order radial distortion coefficient.
%                       Value of K2 must be  $> +$  or  $- 1.e+07$ 
%           K3          Reserved for 3d order radial distortion coefficient (not modeled).
%           K4          First coefficient of tangential distortion.
%                       Magnitude keep between  $-1.0 < K4 < +1.0$ 
%           K5          Second coefficient of tangential distortion.
%                       Magnitude keep between  $-1.0 < K5 < +1.0$ 
%           K1input     Copy of K1 to be used to check distortion of undistorted coordintes.
%           K2input     Copy of K2 to be used to check distortion of undistorted coordintes.
%           K4input     Copy of K4 to be used to check distortion of undistorted coordintes.
%           K5input     Copy of K5 to be used to check distortion of undistorted coordintes.
%           Yes_Noise   Set to 1 to apply Gaussian normal noise error to calibration point data due to
%                       poor resolution of the image point centre.
%           Max_Blunder Sets the max error for point selection due to resolution noise.
%                       Dimesions in meters (i.e. 0.10 = 10 cm error).
%           Blunder_Variance Limits severe outliers to variance of random distribution.
%           Measure_Error Set to 1 to apply Gaussian normal noise error to calibration point data due to
%                       inaccurate measurement of points.
%           Measure_Max  Sets the max error for point selection due to resolution noise.
%                       Dimesions in meters (i.e. 0.10 = 10 cm error).
%           Measure_Var  Limits severe outliers to variance of random distribution.
%           Z_Max        Offset/measurement error of each calibration grid level elevation.
%           Division_Factor Used to modify the image coverage.
%           CCD_Array_Size Resolution of the camera sensor (CCD).
%           Frame_Grabber_Resolution Resolution of the computer frame buffer.
%           JigSize       Defines overall x and y dimesions in meteres.
%           PointSpacing  Defines the distance between points in the grid either
%                       along x or y axis (square) or along the radial arm (spoke).
%           zStage1       Defines the change in elevation between calibration grid 0 and 1.
%           zStage2       Defines the change in elevation between calibration grid 0 and 2.%

```

```

% Ow_w      Origin of world reference frame in world coordinates.
% Twc_w     Translation vector to from world origin to camera frame origin in Fw.
% Twc_wPerfect Copy of Twc_w, used for plotting.
% Twj_w     Centre of calibration grid (jig) in Fc.
%           Grid elevation for level 0 must be z=0!!!
% Tcj_w     Translation vector from camera origin to jig reference frame centre in Fc.
% lw_c      Rotation matrix from world to camera frame.
% TargetPattern Used to select which calibration pattern to use.
% CameraOpticalAxisVector Vector in Fc from origin to centre point of calibration jig in Fc.
% CalibGrid# 3xN array of calibration points for level #.
% JigX       1xN array of square jig x coordinates. Used for plotting.
% JigY       1xN array of square jig y coordinates. Used for plotting.
% lw_j      Rotation from Fw to Fj. Set as identity matrix. Can be modified.
% TgtPts_Level#_InFc 3xN array of calibration points transformed to Fc
% Level#_CCDPnts 2xN array of calibration points transformed to pixel coordinates.
% labels#    1xN array of calibration point reference labels.
% ViewableBoundary Defines the boundary edges of CCD FOV.
% CrossHairX Used for plotting centre point of CCD array FOV.
% CrossHairY Used for plotting centre point of CCD array FOV.
% CalibGrid#_in_FOV 3xN array of calibration points that are within the Field of View.
% Modify_Image User input variable to choose to remove points from visible FOV.
% FirstRow    User input variable select first row of points to remove from visible FOV.
% LastRow     User input variable select last row of points to remove from visible FOV.
% FirstCol    User input variable select first column of points to remove from visible FOV.
% LastCol     User input variable select last column of points to remove from visible FOV.
% Row_Range   User input for the range of rows to be removed (removing points) from the
image plane.
% Colm_Range  User input for the range of column to be removed (removing points) from the
image plane.
% BackProjectPts_Level0 3xN array of input points that have been processed and
%                   projected back to Fw.
% Error        5xN array of error RMS of (x^2+y^2) point differences from original
%                   points and back projected points.
% Error1       2xN array with point labels and rms error in millimeter distance.
% CamToTgtCen 3x2 array of vector from calculated camera position to grid centre in Fw.
% PerfectCamToTgtCen 3x2 array of vector from input camera position to grid centre in Fw.
%
%
%
% Functions Used:
%     Default_Jig_Values
%     RotoW_C
%     Intrinsic_Parameters
%     TargetGridSquare_3_Level
%     TargetGridSpoke_3_Level
%     CCD_ArrayPlane
%     Calibrate_Grid_Points_in_FOV
%     Modify_Image_Coverage
%     Camera_Model Interface to calibration model under investigation.
%     Back_Project
%     Error_Analysis
%
% [JigSize.PointSpacing.zStage1.zStage2]=Default_Jig_Values;
Ow_w=[0;0;0]; % World Origin in the world coordinate frame

```

```

[Twc_w,Twj_w]=Cam_Posn_and_Jig_Cen;
if (Twc_w(3,1)<= Twj_w(3,1)+ zStage2)
    error('Camera position is at or below upper calibration target. Aborting!');
end;
Tcj_w=Twj_w-Twc_w;
[Iw_c]=RotoW_C(Tcj_w);
Twc_wPerfect = Twc_w; % Used later for plotting comparison of camera postions

disp('Calibration Data to obtain Model Data:');
disp('Camera position in World coordinate frame: ');disp(Twc_w);
disp('Calibration jig centre point in World coordinate frame: ');disp(Twj_w);

[f,Cx,Cy,sx,dx,dy,Nex,Nfx,K1,K2,K4,K5,Yes_Noise,Max_Blunder,Blunder_Variance,...
    Measure_Error,Measure_Max,Measure_Var,Z_Max,Division_Factor,...
    CCD_Array_Size,Frame_Grabber_Resolution] = Intrinsic_Parameters;

%
K1input = K1;
K2input = K2;
K4input = K4;
K5input = K5;
%      Change of axis to x-axis as optical axis and y-axis left and z-axis up.
%
disp('Rotation matrix from the world to camera in Fw is: ');disp(Iw_c);
CameraOpticalAxisVector=Iw_c*Tcj_w;

%      Plot of the target array with position of camera and optical axis displayed
disp('Do you wish a square or circular target grid?');
TargetPattern=input('Press 1 and enter for square. Press 2 and enter for circular...');

while (TargetPattern~=1 & TargetPattern~=2)
    disp('Invalid entry. Select either 1 or 2');
    TargetPattern=input('Press 1 and enter for square. Press 2 and enter for circular...');
end;

if TargetPattern==1
    [CalibGrid0,CalibGrid1,CalibGrid2,JigX,JigY]=TargetGridSquare_3_Level...
        (Twj_w,Twc_w,Ow_w,JigSize,PointSpacing,zStage1,zStage2);
else
    [CalibGrid0,CalibGrid1,CalibGrid2]=TargetGridSpoke_3_Level...
        (Twj_w,Twc_w,Ow_w,JigSize,PointSpacing,zStage1,zStage2);
end;

Iw_j=eye(3); % Rotation from Ow to Oj is identity. Can be adjusted if desired later.

[TgtPts_Level0_InFc,TgtPts_Level1_InFc,TgtPts_Level2_InFc,Level0_CCDPoints,labels0,...
    Level1_CCDPoints,labels1,Level2_CCDPoints,labels2,...
    CrossHairX,CrossHairY,ViewableBoundary,Protected_Level0,Distorted_Level0,...
    DistortProlem0,DistortProlem1,DistortProlem2]=...
    CCD_ArrayPlane(Twj_w,Twc_w,Iw_j,Iw_c,Tcj_w,f,Cx,Cy,sx,dx,dy,...
        Nex,Nfx,K1,K2,K4,K5,CCD_Array_Size,CalibGrid0,CalibGrid1,CalibGrid2,Yes_Noise,...
        Max_Blunder,Blunder_Variance,...
        Measure_Error,Measure_Max,Measure_Var,Z_Max,CameraOpticalAxisVector);

disp('Do you wish to delete a selected portion of the image?');
disp('to modify the point coverage within the image frame???');

```

```

Modify_Image = input('Press 1 and enter to modify image coverage...else enter (0...)');

if Modify_Image==1
    disp('Define the range of calibration points to remove you wish to remove. ');
    disp('Row 1, Column 1 are in the top left position. ');
    disp('Array has been divided into the following number of rows/cols: ', disp(Division_Factor));
    FirstRow = input('Enter the first row number to be deleted: ');
    LastRow = input('Enter the last row number to be deleted: ');
    FirstCol = input('Enter the first column number to be deleted: ');
    LastCol = input('Enter the last column number to be deleted: ');
    if (FirstRow<1 | FirstRow>Division_Factor) | (LastRow<=FirstRow |
LastRow>Division_Factor) |...
        (FirstCol<1 | FirstCol>Division_Factor) | (LastCol<=FirstCol |
LastCol>Division_Factor);
        disp('Error in selection of values, exceed limits. Image Modified Aborted. ');
    else
        Row_Range = [FirstRow:LastRow];
        Colm_Range = [FirstCol:LastCol];
        [Level0_CCDPoints,labels0] = Modify_Image_Coverager(Level0_CCDPoints,...
            labels0,Cx,Cy,Division_Factor,Row_Range,Colm_Range);
        [Level1_CCDPoints,labels1] = Modify_Image_Coverager(Level1_CCDPoints,...
            labels1,Cx,Cy,Division_Factor,Row_Range,Colm_Range);
        [Level2_CCDPoints,labels2] = Modify_Image_Coverager(Level2_CCDPoints,...
            labels2,Cx,Cy,Division_Factor,Row_Range,Colm_Range);
        figure
        plot(Level0_CCDPoints(1,:),Level0_CCDPoints(2,:), 'r');
        title('Modified Target Points Within CCD Array Area and Optical Centre');
        xlabel('x-axis');
        ylabel('y-axis');
        hold on
        plot(Level1_CCDPoints(1,:),Level1_CCDPoints(2,:), 'g-');
        plot(Level2_CCDPoints(1,:),Level2_CCDPoints(2,:), 'bd');
        plot(ViewableBoundary(1,:),ViewableBoundary(2,:), 'k-');
        plot(CrossHairX(1,:),CrossHairX(2,:), 'r-');
        plot(CrossHairY(1,:),CrossHairY(2,:), 'r-');
        hold off
        end;
    end;

% Based upon the data returned from CCD_ArrayPlane, modify each level
% of CalibGrid to use only the points from within the Field of View as defined
% by the labels array which cross-references the data.

[CalibGrid0_in_FOV] = Calibrate_Grid_Points_in_FOV(CalibGrid0, labels0);
[CalibGrid1_in_FOV] = Calibrate_Grid_Points_in_FOV(CalibGrid1, labels1);
[CalibGrid2_in_FOV] = Calibrate_Grid_Points_in_FOV(CalibGrid2, labels2);

[Protected_Level0] = Calibrate_Grid_Points_in_FOV(Protected_Level0, labels0);

disp('Default is to run the data through the camera model. ');
PerfectCam = input('To run the Perfect Model, enter 1, or enter 0 for the Camera Model. ');

if PerfectCam == 1
    disp('Data and error analysis are for the Perfect Perspective Model');
    % Calculate the back projected values of CCD array points for Level 0

```



```

[BackProjectPts_Level0]=Back_Project(Level0_CCDPoints.f,dx,dy,sx,Iw_c,Iw_j,Two_w,Tcj_w);
% Compare known calibration to back projected points.
[Error]=Error_Analysis(labels0,BackProjectPts_Level0,CalibGrid0_in_FOV);
else
    % Run the camera model
    [Two_w,Iw_c,sx,K1,K2,BackProjectPts_Level0,Confirm_Distortion_Points]=Camera_Model...
        (Level0_CCDPoints,Level1_CCDPoints,Level2_CCDPoints,...
        CalibGrid0_in_FOV,CalibGrid1_in_FOV,CalibGrid2_in_FOV);
    % Confirm distortion from Camera Model
    % First apply distortion model and calibrated K1 and K2

    Xd = Confirm_Distortion_Points(1,:);
    Yd = Confirm_Distortion_Points(2,:);
    Xu = [];
    Yu = [];

    for i=1:length(Confirm_Distortion_Points)
        Xu_New =
            Xd(1,i)+K1input*Xd(1,i)^3+K1input*Xd(1,i)*Yd(1,i)^2+K2input*Xd(1,i)^5+2*K2input*Xd(1,i)^3*Yd(1,
            i)^2+K2input*Xd(1,i)*Yd(1,i)^4+...
            K4input*Xd(1,i)^2+K4input*Yd(1,i)^2+2*K4input*Xd(1,i)^2+2*K5input*Xd(1,i)*Yd(1,i);
        Xu=[Xu,Xu_New];
        Yu_New =
            Yd(1,i)+K1input*Yd(1,i)^3+K1input*Yd(1,i)*Xd(1,i)^2+K2input*Yd(1,i)^5+2*K2input*Yd(1,i)^3*Xd(1,
            i)^2+K2input*Yd(1,i)*Xd(1,i)^4+...
            2*K4input*Xd(1,i)*Yd(1,i)+K5input*Xd(1,i)^2+K5input*Yd(1,i)^2+2*K5input*Yd(1,i)^2;
        Yu=[Yu,Yu_New];
    end;

    Pt_ErrorX = zeros(1,length(Xu));
    Pt_ErrorY = zeros(1,length(Yu));
    for i=1:length(Pt_ErrorX)
        Pt_ErrorX(1,i) = Xu(1,i)-Protected_Level0(1,i);
        Pt_ErrorY(1,i) = Yu(1,i)-Protected_Level0(2,i);
    end;
    Mag_Dist_Error=[];
    for k = 1:length(Pt_ErrorX)
        This_Error = sqrt(Pt_ErrorX(1,k)^2+Pt_ErrorY(1,k)^2);
        Mag_Dist_Error =[Mag_Dist_Error, This_Error ];
    end;

    Xu2 = [];
    Yu2 = [];
    for i=1:length(Confirm_Distortion_Points)
        Xu_New2 =
            Xd(1,i)+K1*Xd(1,i)^3+K1*Xd(1,i)*Yd(1,i)^2+K2*Xd(1,i)^5+2*K2*Xd(1,i)^3*Yd(1,i)^2+K2*Xd(1,i)*Y
            d(1,i)^4+...
            K4*Xd(1,i)^2+K4*Yd(1,i)^2+2*K4*Xd(1,i)^2+2*K5*Xd(1,i)*Yd(1,i);
        Xu2=[Xu2,Xu_New2];
        Yu_New2 =
            Yd(1,i)+K1*Yd(1,i)^3+K1*Yd(1,i)*Xd(1,i)^2+K2*Yd(1,i)^5+2*K2*Yd(1,i)^3*Xd(1,i)^2+K2*Yd(1,i)*X
            d(1,i)^4+...
            2*K4*Xd(1,i)*Yd(1,i)+K5*Xd(1,i)^2+K5*Yd(1,i)^2+2*K5*Yd(1,i)^2;
        Yu2=[Yu2,Yu_New2];
    end;
    XuYu2 = [Xu2',Yu2'];

```

```

Pt_ErrorX2 = zeros(1,length(Xu2));
Pt_ErrorY2 = zeros(1,length(Yu2));
for i=1:length(Pt_ErrorX2)
    Pt_ErrorX2(1,i) = Xu2(1,i)-Protected_Level0(1,i);
    Pt_ErrorY2(1,i) = Yu2(1,i)-Protected_Level0(2,i);
end;

Mag_Dist_Error2=[];
for k = 1:length(Pt_ErrorX2)
    This_Error2 = sqrt(Pt_ErrorX2(1,k)^2+Pt_ErrorY2(1,k)^2);
    Mag_Dist_Error2 =[Mag_Dist_Error2, This_Error2 ];
end;

figure
plot(labels0(1,:),Mag_Dist_Error(1,:), 'r*');
title('Check on Magnitude of error Due to distortion of points with K1 K2 Original');
xlabel('Point Labels');
ylabel('Magnitude of Point Error');

figure
plot(labels0(1,:),Mag_Dist_Error2(1,:), 'b*');
title('Check on Magnitude of error Due to distortion of points');
xlabel('Point Labels');
ylabel('Magnitude of Point Error');

disp('Error data for the Camera Model');

[Error]=Error_Analysis(labels0,BackProjectPts_Level0,CalibGrid0_in_FOV);
end;
% Error analysis
disp('Overall error per target point is (in metres)')
disp('Label      x-Error      y-Error      Error in m      Error in mm'),disp(Error,:);
Error1=[Error(1,:);Error(5,:);CalibGrid0_in_FOV(1,:);CalibGrid0_in_FOV(2,:)];
disp('Overall error per target point is (in mm) and actual Point Coordinates')
disp('Label      Error in mm      X (m)      Y (m)'),disp(Error1,:);

figure
plot(CalibGrid0_in_FOV(1,:),CalibGrid0_in_FOV(2,:), 'r*')
xlabel('x-axis');
ylabel('y-axis');
title('Back Projected Target Points(*) vs Actual Points(*) and Camera Position');
grid on
hold on
plot(BackProjectPts_Level0(1,:),BackProjectPts_Level0(2,:), 'bo')
plot(Twe_w(1,1),Twe_w(2,1), 'ks')
hold off

if PerfectCam ~= 1
    CamToTgtCen=[Twj_w,:;Twe_w,:];
    PerfectCamToTgtCen=[Twj_w,:;Twe_wPerfect,:];
    CamToTgtCen = CamToTgtCen';
    PerfectCamToTgtCen = PerfectCamToTgtCen';
    figure
    plot3(CalibGrid0(1,:),CalibGrid0(2,:),CalibGrid0(3,:), 'r*');
    xlabel('x-axis');

```

```

ylabel('y-axis');
xlabel('z-axis');
title('Perfect (o) and Calibrated(+) Camera Position Relative to Target Pattern in Fw');
grid on
hold on
view(Twe_w);
plot3(Twe_w(1,1),Twe_w(2,1),Twe_w(3,1),'g+')
plot3(Twe_wPerfect(1,1),Twe_wPerfect(2,1),Twe_wPerfect(3,1),'mo')
hold on
plot3(CamToTgtCen(1,:),CamToTgtCen(2,:),CamToTgtCen(3,:),'b-')
plot3(PerfectCamToTgtCen(1,:),PerfectCamToTgtCen(2,:),PerfectCamToTgtCen(3,:),'r-')
hold off
end;

disp('Maximum single point error (in mm) is:'); disp(max(Error(5,:)));
disp('Mean point error (in mm) is:'); disp(mean(Error(5,:)));
disp('Median point error (in mm) is:'); disp(median(Error(5,:)));

if PerfectCam == 1
    figure
    hist(Error(5,:),40)
    title('Error Distribution for Perspective Projection Model');
    xlabel('Error per calibration point in mm. ');
    ylabel('Frequency');
else
    figure
    hist(Error(5,:),40)
    title('Error Distribution for Camera Model');
    xlabel('Error per calibration point in mm. ');
    ylabel('Frequency');
end;

% Surface plot of error at the actual (X,Y) point
if TargetPattern==1
    Z = zeros(length(JigX),length(JigY));
    for j=1:length(JigX)
        for i=1:length(JigY)
            for k=1:length(CalibGrid0_in_FOV)
                if (CalibGrid0_in_FOV(1,k)== JigX(1,j) & CalibGrid0_in_FOV(2,k)==
JigY(1,i))
                    Z(j,i) = Error(4,k);
                end;
            end;
        end;
    end;
    figure
    surf(JigX(1,:),JigY(1,:),Z);
    xlabel('x-axis');
    ylabel('y-axis');
    zlabel('z-axis');
    title('Plot of Error Relative to Actual Point Location');
    grid on
end;

```

Input Data: Intrinsic Parameters

```

function [f,Cx,Cy,sx,dx,dy,Nex,Nfx,K1,K2,K4,K5,Yes_Noise,Max_Blunder,Blunder_Variance,...
        Measure_Error,Measure_Max,Measure_Var,Z_Max,Division_Factor,...
        CCD_Array_Size,Frame_Grabber_Resolution] = Intrinsic_Parameters;

% Function: Intrinsic Parameters
%
% Purpose: File to contain intrinsic parameter values that are used in the
%          creation of input data for camera models.
%
% Variables:
% f          effective focal length
% Cx         Principal point centre value for x-direction (columns/2) of computer frame
% buffer
% Cy         Principal point centre value for y-direction (rows/2) of computer frame buffer
% sx         Horizontal scale factor
% dx         Pixel-to-pixel metric spacing in x-direction (column-to-column)
% dy         Pixel-to-pixel metric spacing in y-direction (row-to-row)
% Nex        Number of columns in the resolution of the camera sensor (CCD)
% Nfx        Number of columns in the resolution of the computer frame buffer
% K1         1st order radial distortion coefficient
%            Magnitude keep between  $-1.2 \times 10^3 \leq K1 \leq +1.2 \times 10^3$ 
%            Negative K1 gives stronger pin-cushion distortion
%            Positive K1 gives stronger barrel distortion
% K2         2d order radial distortion coefficient
%            Value of K2 must be  $> +$  or  $< -1 \times 10^7$ 
% K3         Reserved for 3d order radial distortion coefficient (not modeled)
% K4         First coefficient of tangential distortion
%            Magnitude keep between  $-1.0 \leq K4 \leq +1.0$ 
% K5         Second coefficient of tangential distortion
%            Magnitude keep between  $-1.0 \leq K5 \leq +1.0$ 
% Yes_Noise  Set to 1 to apply Gaussian normal noise error to calibration point data.
% Max_Blunder Sets the max error for point selection due to resolution noise.
%            Dimensions in meters (i.e. 0.10 = 10 cm error)
% Blunder_Variance Limits severe outliers to variance of random distribution.
% Measure_Error Set to 1 to apply Gaussian normal noise error to calibration point data due to
%               inaccurate measurement of points.
% Measure_Max   Sets the max error for point selection due to resolution noise.
%               Dimensions in meters (i.e. 0.10 = 10 cm error)
% Measure_Var   Limits severe outliers to variance of random distribution
% Z_Max         Offset measurement error of each calibration grid level elevation
% Division_Factor Used to modify the image coverage.
% CCD_Array_Size Resolution of the camera sensor (CCD)
% Frame_Grabber_Resolution Resolution of the computer frame buffer

% Call to a function to retrieve default values required to generate data.
% These values will not normally be modified extensively.
[Cx,Cy,dx,dy,Nex,Nfx,CCD_Array_Size,Frame_Grabber_Resolution] = Common_Intrinsic_Parameters;
% Call to a function to retrieve default values required to generate data.
% These values will normally be modified for study purposes.
% Controlled values are usually calibrated by the camera model under study.
[fx,K1,K2,K4,K5,Yes_Noise,Max_Blunder,Blunder_Variance,Measure_Error,...
    Measure_Max,Measure_Var,Z_Max,Division_Factor] =
Controlled_Intrinsic_Parameters;

```

Input Data: Common Intrinsic Parameters

```
function
[ Cx,Cy,dx,dy,Ncx,Nfx,CCD_Array_Size,Frame_Grabber_Resolution]=Common_Intrinsic_Parameters;
%      Function: Common_Intrinsic_Parameters
%
%      Purpose: File to contain intrinsic parameter values that are used in the
%               creation of input data for camera models.
%      Variables:
%
%      Cx          Principal point centre value for x-direction (columns/2) of computer frame
buffer
%      Cy          Principal point centre value for y-direction (rows/2) of computer frame buffer
%      dx          Pixel-to-pixel metric spacing in x-direction (column-to-column)
%      dy          Pixel-to-pixel metric spacing in y-direction (row-to-row)
%      Ncx         Number of columns in the resolution of the camera sensor (CCD)
%      Nfx         Number of columns in the resolution of the computer frame buffer
%      CCD_Array_Size Resolution of the camera sensor (CCD)
%      Frame_Grabber_Resolution Resolution of the computer frame buffer

CCD_Array_Size = [640 480];
% Frame_Grabber_Resolution = [1024 768];
Frame_Grabber_Resolution = [640 480];

CCD_Array_Size_in_Metres_x = 0.0254/3; % Current array is 1/3 of an inch square
CCD_Array_Size_in_Metres_y = 0.0254/3; % Current array is 1/3 of an inch square

Cx= Frame_Grabber_Resolution(1,1)/2;
Cy= Frame_Grabber_Resolution(1,2)/2;
dx= CCD_Array_Size_in_Metres_x/(CCD_Array_Size(1,1)-1);
dy= CCD_Array_Size_in_Metres_y/(CCD_Array_Size(1,2)-1);
Ncx= CCD_Array_Size(1,1);
Nfx= Frame_Grabber_Resolution(1,1);
```

Input Data: Common Intrinsic Parameters

```

function [f,sx,K1,K2,K4,K5,Yes_Noise,Max_Blunder,Blunder_Variance,Measure_Error,...
    Measure_Max,Measure_Var,Z_Max,Division_Factor]=...
    Controlled_Intrinsic_Parameters;
% Function: Controlled_Intrinsic_Parameters
%
% Purpose: File to contain intrinsic parameter values that are used in the
%          creation of input data for camera models.
% Output:
%
% f          effective focal length
% sx         Horizontal scale factor
% K1         1st order radial distortion coefficient
%            If K1 applied without K2,  $-800 < K1 < 2.e+05$ 
%            Magnitude keep between  $-4.6e+03 < K1 < 4.6e+03$ 
%            Negative K1 gives stronger pin-cushion distortion
%            Positive K1 gives stronger barrel distortion
% K2         2d order radial distortion coefficient
%            Value of K2 should be  $> +$  or  $< -1.e+07$ 
%            K2 tends to apply a smoothing to K1 value
% K3         Reserved for 3d order radial distortion coefficient (not modeled)
%
% K4         First coefficient of tangential distortion
%            Magnitude keep between  $-1.0 < K4 < +1.0$ 
% K5         Second coefficient of tangential distortion
%            Magnitude keep between  $-1.0 < K5 < +1.0$ 
% Yes_Noise  Set to 1 to apply Gaussian normal noise error to calibration point data
% Max_Blunder Sets the max error for point selection due to resolution noise.
%            Dimensions in meters (i.e. 0.10 = 10 cm error).
% Blunder_Variance Limits severe outliers to variance of random distribution
% Measure_Error Set to 1 to apply Gaussian normal noise error to calibration point data due to
%               inaccurate measurement of points.
% Measure_Max  Sets the max error for point selection due to resolution noise
%               Dimensions in meters (i.e. 0.10 = 10 cm error).
% Measure_Var  Limits severe outliers to variance of random distribution.
% Z_Max       Offset/measurement error of each calibration grid level elevation.
% Division_Factor Used to modify the image coverage.
%
% Internal Variables:
%
% Radial      Sets which combination of radial distortion components are used.
% DeCentering Sets which combination of de-centering distortion components are used.
% Level#Z_Max Maximum measurement error for each calibration grid level. Does not
%             have to be the same value for all.
%
f= 0.012;
sx= 1.0;
Radial = 0; % 0=No radial, 1=K1 only, 2=K2 only, 3=Both
if Radial == 1
    K1 = 8000.0;

```

```

        K2 = 0.0;
elseif Radial == 2
    K1 = 0.0;
    K2 = 1.e+12;
elseif Radial == 3
    K1 = 1000.0;
    K2 = 1.e+07;
else
    K1 = 0.0;
    K2 = 0.0;
end;

DeCentering = 0; % 0=No DeCentering, 1=K4 only, 2=K5 only, 3=Both K4 & K5
if DeCentering == 1
    K4 = 0.6;
    K5 = 0.0;
elseif DeCentering == 2
    K4 = 0.0;
    K5 = 0.3;
elseif DeCentering == 3
    K4 = 0.25;
    K5 = 0.1;
else
    K4 = 0.0;
    K5 = 0.0;
end;

Yes_Noise = 0; % if it ==1, turns on noise
Max_Blunder = 0.01; % Limits the maximum error in each direction
Blunder_Variance = 0.3; % Smaller the number, the less distortion. Must >=0.0
Measure_Error = 0; % if it ==1, turns on Measurement error
Measure_Max = 0.005;
Measure_Var = 0.7;
Level0Z_Max = 0.002;
Level1Z_Max = 0.002;
Level2Z_Max = 0.002;
Z_Max = [Level0Z_Max,Level1Z_Max,Level2Z_Max];
Division_Factor = 16;

```

Input Data: Setting the Camera Position and Calibration Grid Position

```
function [Twe_w,Twj_w] = Cam_Posn_and_Jig_Cen:
%       Function: Cam_Posn_and_Jig_Cen
%
%       Purpose: File to contain camera position and centre of calibration grid coordinates,
%               values that are used in the creation of input data for camera models
%
%       Returns to:      MatCam
%       Variables:
%
%       Output:
%
%       Twe_w           Translation vector to from world origin to camera frame origin in Fc
%       Twj_w           Centre of calibration grid (jig) in Fc.
%                       Grid elevation for level 0 must be z=0!"

Twe_w=[3;1;4];
Twj_w=[4;4;0];
```


Input Data: Setting the Dimensions and Characteristics of the Calibration Grid

```

function [JigSize,PointSpacing,zStage1,zStage2]=Default_Jig_Values;
%      Function: Default_Jig_Values
%
%      Purpose:File to contain the size and centre-point location of the
%               calibration grig (jig).Two calibration grids are used.
%               One grid is square and one is a spoke pattern with variable number
%               of spokes.
%
%      Returns to:    MatCam
%      Variables:
%
%      JigSize        Defines overall x and y dimesions in meteres.
%      PointSpacing    Defines the distance between points in the grid either
%                       along x or y axis (square) or along the radial arm (spoke).
%      zStage1        Defines the change in elevation between calibration grid 0 and 1.
%      zStage2        Defines the change in elevation between calibration grid 0 and 2.

JigSize = 4.0;
PointSpacing = 0.5;
% JigSize = 2.0;           % Used to create a circle for spoke pattern
% PointSpacing = 1.0;      % Used to create a circle for spoke pattern

zStage1 = 0.15;
zStage2 = 0.35;

```

Input Data: Calibration Grid - Square

```

function [CalibGrid0,CalibGrid1,CalibGrid2,JigX,JigY]=...
    TargetGridSquare_3_Level(Twj_w,Twe_w,Ow_w,JigSize,PointSpacing,zStage1,zStage2);
%
% Purpose: To build a square grid pattern with rotational and translational
%           degrees of freedom for 3 levels to implement Tsai's camera calibration
%           This will create a square grid with x and y target axis parallel to respective x and
%           y world axis
%           This assumes the grid pattern is planar and parallel to the ground plane.
%           Data returned in "undistorted"
%
% Variables:
%
% Twj_w       Centre of calibration grid (jig) in Fw.
%              Grid elevation for level 0 must be z=0!!!
% Twe_w       Translation vector to from world origin to camera frame origin in Fw.
% Ow_w       Origin of world reference frame in world coordinates
% JigSize     Defines overall x and y dimensions in meters
% PointSpacing Defines the distance between points in the grid either
%              along x or y axis (square) or along the radial arm (spoke).
% zStage1     Defines the change in elevation between calibration grid 0 and 1.
% zStage2     Defines the change in elevation between calibration grid 0 and 2.
%
% Outputs:
%
% CalibGrid#   3xN array of calibration points for level #
% JigX         1xN array of square jig x coordinates. Used for plotting
% JigY         1xN array of square jig y coordinates. Used for plotting
%
% Internal Variables
%
% zStage0      Sets the elevation of the lowest level grid
% JigDimension  Used to evenly split calibration points around the centre point
% BottomLeft   Sets start point for tracking points
% TopRight     Sets end point for tracking points.
%
zStage0=Twj_w(3,1);
zStage1 = zStage1 + zStage0;
zStage2 = zStage2 + zStage0;

% Used to set the jig size
JigDimension=JigSize/2;

BottomLeft=[(Twj_w(1,1)-JigDimension),(Twj_w(2,1)-JigDimension)];
TopRight=[(Twj_w(1,1)+JigDimension),(Twj_w(2,1)+JigDimension)];

x = [BottomLeft(1,1):PointSpacing:TopRight(1,1)];
y = [BottomLeft(1,2):PointSpacing:TopRight(1,2)];
JigX=x;
JigY=y;

cols=length(x)*length(y);
CalibGrid0 = zeros(3,cols);
count=1;

```

```

for i=1:length(x)
    for j=1:length(x)
        CalibGrid0(1,count) = x(1,j);
        count=count+1;
    end;
end;

count=1;
for i=1:length(y)
    for j=1:length(y)
        CalibGrid0(2,count) = y(1,i);
        CalibGrid0(3,count) = zStage0;
        count=count+1;
    end;
end;

% Now build level1 and level2 arrays
CalibGrid1=CalibGrid0;
CalibGrid2=CalibGrid0;
for i=1:length(CalibGrid0)
    CalibGrid1(3,i)=zStage1;
    CalibGrid2(3,i)=zStage2;
end;

CamToTgtCen=[Twj_w';Twe_w'];
CamToTgtCen=CamToTgtCen';

figure
plot3(CalibGrid0(1,:),CalibGrid0(2,:),CalibGrid0(3,:),'r');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
title('Camera Position and Optical Axis Relative to Target Pattern in Fw');
grid on
hold on
plot3(CalibGrid1(1,:),CalibGrid1(2,:),CalibGrid1(3,:),'g+');
plot3(CalibGrid2(1,:),CalibGrid2(2,:),CalibGrid2(3,:),'bd');
view(Twe_w);
plot3(Twe_w(1,1),Twe_w(2,1),Twe_w(3,1),'rh')
hold on
plot3(CamToTgtCen(1,:),CamToTgtCen(2,:),CamToTgtCen(3,:),'b-')
hold off

```

Input Data: Calibration Grid – Spoke

```

function [CalibGrid0,CalibGrid1,CalibGrid2]=...
    TargetGridSpoke_3_Level(Twj_w,Twc_w,Ow_w,JigSize,PointSpacing,zStage1,zStage2);
%
%
%      Purpose: To build a circular grid pattern (future - with rotational and translational
%                degrees of freedom).
%                This will create a circle with x and y target axis parallel to respective x and y
world axis
%                Separation of calibration points is variable
%                The spoke pattern will have eight legs projecting from the centre
%                Spoke 1 will be oriented as if x=1, y=0 and then rotate counter clockwise
%
%      Variables:
%
%      Twj_w      Centre of calibration grid (pg) in Fc.
%                  Grid elevation for level 0 must be z=0!!!
%      Twc_w      Translation vector to from world origin to camera frame origin in Fw.
%      Ow_w      Origin of world reference frame in world coordinates.
%      JigSize     Defines overall x and y dimesions in meters.
%      PointSpacing Defines the distance between points in the grid either
%                  along x or y axis (square) or along the radial arm (spoke)
%      zStage1     Defines the change in elevation between calibration grid 0 and 1.
%      zStage2     Defines the change in elevation between calibration grid 0 and 2.
%
%      Outputs:
%
%      CalibGrid#   3xN array of calibration points for level #
%
%      Internal Variables
%
%      zStage0      Sets the elevation of the lowest level grid.
%      SpokeLength   Used to evenly split calibration points along spoke length.
%      NumSpokeLegs  Sets number of legs in the spoke/circle pattern.
%      PointsPerLeg  Used to track total calibration points
%      TotalPoints   Total calibration points on each level.
%      RadialDistance Used to properly space position points
%      LegAngle      Controls rotation of each spoke with even placement.
%
%      This assumes the pattern is planar and parallel to the ground plane
zStage0=Twj_w(3,1);
zStage1 = zStage1 + zStage0;
zStage2 = zStage2 + zStage0;

SpokeLength=JigSize/2;
NumSpokeLegs = 16.0;
% NumSpokeLegs = 180.0; % Used to create a circle pattern

%      Determine the number of calibration points per leg
PointsPerLeg = SpokeLength/PointSpacing;
TotalPoints = 1 + PointsPerLeg*NumSpokeLegs;

CalibGrid0 = zeros(3,length(TotalPoints));

```

LegAngle = (2*pi)/NumSpokeLegs; % This will assist in the rotation around the spoke

% First put in the centre point

CalibGrid0(1,1) = Twj_w;

alpha=0.0; % This is used to rotate from one leg to the next

ColmNum=2;

for j=1:NumSpokeLegs

RadialDistance=PointSpacing;

for i=1:PointsPerLeg

CalibGrid0(1,ColmNum)=Twj_w(1,1) + RadialDistance*cos(alpha);

CalibGrid0(2,ColmNum)=Twj_w(2,1) + RadialDistance*sin(alpha);

CalibGrid0(3,ColmNum)=zStage0; % No change to z value

RadialDistance = RadialDistance + PointSpacing;

ColmNum=ColmNum+1;

end

alpha=alpha+LegAngle; % Rotates to next spoke

end

% Now build level1 and level2 arrays

CalibGrid1=CalibGrid0;

CalibGrid2=CalibGrid0;

for i=1:max(size(CalibGrid0))

CalibGrid1(3,i)=zStage1;

CalibGrid2(3,i)=zStage2;

end;

CamToTgtCen=[Twj_w';Twe_w'];

CamToTgtCen=CamToTgtCen';

figure;

plot3(CalibGrid0(1,:),CalibGrid0(2,:),CalibGrid0(3,:), 'r*');

xlabel('x-axis');

ylabel('y-axis');

zlabel('z-axis');

title('Camera Position and Optical Axis Relative to Target Pattern in fwh');

grid on

hold on

plot3(CalibGrid1(1,:),CalibGrid1(2,:),CalibGrid1(3,:), 'g+');

plot3(CalibGrid2(1,:),CalibGrid2(2,:),CalibGrid2(3,:), 'bd');

view(Twe_w);

plot3(Twe_w(1,1),Twe_w(2,1),Twe_w(3,1), 'rh');

hold on

plot3(CamToTgtCen(1,:),CamToTgtCen(2,:),CamToTgtCen(3,:), 'b-');

hold off

Main File: Determine Rotation Matrix

```

function [Iw_c]=RotoW_C(Tej_w);
%
%      Purpose: Main file used to control execution of program.
%              MatCam is a program to develop integrated input data to test/conduct
%              accuracy and parametric studies of camera calibration models.
%
%      Returns to: Matcam
%      Variables:
%
%      Tej_w      Translation vector from camera origin to µg reference frame centre in f.c.
%
%      Outputs:
%
%      Iw_c       Rotation matrix from world to camera frame.
%
x=Tej_w(1,1);
y=Tej_w(2,1);
z=Tej_w(3,1);
%      The z value must always be negative. Error/abort test done in main code.
%      Adjusting for proper rotation based on vector from camera to µg centre.
iew=[x; y; z];
jew=[-y; x; 0];
kew=[-z*cos(tan(y/x)); -z*sin(tan(y/x)); sqrt(x.^2 + y.^2)];

%      Determine the length of each vector so it can be normalized
i_len = sqrt(iew(1,1)^2 + iew(2,1)^2 + iew(3,1)^2);
j_len = sqrt(jew(1,1)^2 + jew(2,1)^2 + jew(3,1)^2);
k_len = sqrt(kew(1,1)^2 + kew(2,1)^2 + kew(3,1)^2);
%      Normalize the vectors
iew=iew/i_len;
jew=jew/j_len;
kew=kew/k_len;
%      Check if orthonormal
%      Now create the orthonormal rotation matrix
Iw_c=[iew'; jew'; kew'];
Ident = inv(Iw_c)*(Iw_c);
if Ident ~= I
    error('Rotation matrix is wrong - Check Function RotoW_C.m');
end;

```

Coordinate Modification: Main File CCD_ArrayPlane

```

function [TgtPts_Level0_InFc,TgtPts_Level1_InFc,TgtPts_Level2_InFc,Level0_CCDPoints,labels0,...
Level1_CCDPoints,labels1,Level2_CCDPoints,labels2,...
CrossHairX,CrossHairY,ViewableBoundary,Protected_Level0,Distorted_Level0,...
DistortProlem0,DistortProlem1,DistortProlem2]=...
CCD_ArrayPlane(Twj_w,Twe_w,Iwj_j,Iwc_c,Tcj_w,...
f,Cx,Cy,sx,dx,Ncx,Nfx,K1,K2,K4,K5,CCD_Array_Size,...
CalibGrid0,CalibGrid1,CalibGrid2,Yes_Noise,...
Max_Blunder,Blunder_Variance,...
Measure_Error,Measure_Max,Measure_Var,Z_Max,CameraOpticalAxisVector);
% Purpose: Transform Fw calibration points to Fc and digitize these points in Fc
%          Fc coordinate system (right hand system).
%          X-axis=optical axis.
%          Y-axis=positive left
%          Z-axis=positive up
%
% Returns to:   MatCam
% Variables:
%
%          Camera Parameters:
%
%          f          effective focal length
%          Cx          Principal point centre value for x-direction (columns,2) of computer frame
buffer
%          Cy          Principal point centre value for y-direction (rows,2) of computer frame buffer
%          sx          Horizontal scale factor
%          dx          Pixel-to-pixel metric spacing in x-direction (column-to-column)
%          dy          Pixel-to-pixel metric spacing in y-direction (row-to-row)
%          Ncx         Number of columns in the resolution of the camera sensor (CCD)
%          Nfx         Number of columns in the resolution of the computer frame buffer
%          K1          1st order radial distortion coefficient.
%                     Magnitude keep between  $-1.e+03 \leq K1 \leq +1.e+03$ 
%                     Negative K1 gives stronger pin-cushion distortion.
%                     Positive K1 gives stronger barrel distortion.
%          K2          2d order radial distortion coefficient.
%                     Value of K2 must be  $> +$  or  $-1.e+07$ 
%          K3          Reserved for 3d order radial distortion coefficient (not modeled).
%
%          K4          First coefficient of tangential distortion.
%                     Magnitude keep between  $-1.0 \leq K4 \leq +1.0$ 
%          K5          Second coefficient of tangential distortion.
%                     Magnitude keep between  $-1.0 \leq K5 \leq +1.0$ 
%          Yes_Noise   Set to 1 to apply normal noise error to image point data due to
%                     poor resolution of the image point centre
%          Measure_Error Set to 1 to apply normal noise error to image point data due to
%                     inaccurate measurement of points.
%          Measure_Max  Sets the max error for point selection due to resolution noise.
%                     Dimensions in pixels.
%          Measure_Var  Limits blunder to variance of random distribution
%          Z_Max        Offset measurement error of each calibration grid level elevation.
%          CCD_Array_Size Resolution of the camera sensor (CCD).
%          Frame_Grabber_Resolution Resolution of the computer frame buffer
%
%          Twj_w        Translation vector to from world origin to camera frame origin in Fc.

```

```

% Twl_w      Centre of calibration grid (jig) in Fc.
%            Grid elevation for level 0 must be z=0!!!
% Tel_w      Translation vector from camera origin to µg reference frame centre in Fc.
% Tw_c       Rotation matrix from world to camera frame.
% Tw_l       Rotation from Fw to Fj. Set as identity matrix. Can be modified.
% CalibGrid# 3xN array of calibration points for level #.
% Max_Blunder Sets the max error for point selection due to resolution noise.
%            Dimensions in pixels
% Blunder_VarianceLimits blunders to variance of random distribution.
% CameraOpticalAxisVector Vector in Fc from origin to centre point of calibration µg in Fc.
%
% Outputs:
%
% TgtPts_Level#_InFc 3xN array of calibration points transformed to Fc.
% Level#_CCDPnts    2xN array of calibration points transformed to pixel coordinates.
% labels#           1xN array of calibration point reference labels.
% CrossHairX        Used for plotting centre point of CCD array FOV.
% CrossHairY        Used for plotting centre point of CCD array FOV.
% ViewableBoundary Defines the boundary edges of CCD FOV.
% Protected_Level0 Nx2 array of distorted coordinates in image coordinates
%
% Internal Variables:
%
% Oe_l         Sets the Origin of Fc.
% TgtPts_Level#_InFc_P 3xN array of Pure calibration points transformed.
%               to Fc, no distortion of any kind applied
% labels#_P     1xN array of pure/undistorted calibration point reference labels.
% CalibGrid#_in_FOV 3xN array of calibration points that are within the Field of View
% CamPosnInFc   Redundant variable for origin of Fc.
% TgtCenInFc    Optical axis vector in Fc (along x-axis of Fc).
% Xu_Level#     Undistorted x coordinate point for level #
% Yu_Level#     Undistorted y coordinate point for level #
% Xd_Level#     Distorted x coordinate point for level #.
% Yd_Level#     Distorted y coordinate point for level #.
% Xu_Level#_P   Pure/undistorted x coordinate point for level #.
% Yu_Level#_P   Pure/undistorted y coordinate point for level #.
% DistortPolem# Indicates a problem with these points for level# when distorting them
% GroupPureData 6xN array used for plotting of undistorted points with metric values
% GroupNoiseData 6xN array used for plotting of points with metric values that
%               may have resolution noise distortion.
% CCD_Level#    2xN array of distorted 2D pixel coordinates for level#.
%               Coordinates are from centre of the array vice top left.
% Protected_Level0 Nx2 array of undistorted coordinates in image coordinates.
% CCD_Level#_U  2xN array of undistorted 2D pixel coordinates for level#.
%               Coordinates are from centre of the array vice top left.
% xLimit        Number of pixels from centre of array to left and right boudary edges of CCD
FOV.
% yLimit        Number of pixels from centre of array to upper and lower boudary edges of
CCD FOV.
% ViewableBoundary Defines the boundary edges of metric FOV.
% CrossHairL_X   Used for plotting centre point of metric array FOV.
% CrossHairL_Y   Used for plotting centre point of metric array FOV.
% Level#_CCDPnts 2xN array of CCD coordinates of distorted points.
% Level#_U_CCDPnts 2xN array of CCD coordinates of undistorted points.
% GpData         Waste variable.
% garb1,garb2    Waste variables.

```



```

% Yes_Pixel_Adjust Used to choose to apply CCD distortion due to digitization of data.
%
% Functions Used:
%     Apply_Noise Applies Gaussian noise to coordinate points (noise resolution).
%     Convert_to_2D_Image Create 2D image points from 3D points in Fe.
%     Convert_To_CCD_Digitize points from Fe to 2D image plane.
%     Distort_Grid_Pts Apply radial and/or tangential distortion.
%     CCD_Points_in_FOV_1 Limit input data to only points within camera field of view.
%     CCD_Points_in_FOV_2 (Used for plotting metric data. Not required otherwise).
%     Before_CCD_Conversion (Used for plotting metric data. Not required otherwise).
%     See_Distortion Plots distorted and undistorted points, as visual confirmation.
%     CCD_Pixel_Adjust For adjusting CCD coordinates due to analog to digital conversion
%                       and for visualization of error due to digitization.
%
Oe_c=[0;0;0]; % Sets the Origin of Fe

% Translate the target point arrays to Oe in world reference frame (Fw)
% Also create a labels array for each level for tracking and dealing with
% points that are on or behind the camera image plane
labels0 = zeros(1, length(CalibGrid0));
labels1 = zeros(1, length(CalibGrid1));
labels2 = zeros(1, length(CalibGrid2));

% Apply a Measurement Noise to the calibration points as though the
% points were not measured with strict care.
% Deviations will be mainly in x and y and general offset in z.

[Measured_Points0] = Apply_Measurment_Error(Measure_Error,...
      CalibGrid0.Measure_Max.Measure_Var.Z_Max(1,1));
[Measured_Points1] = Apply_Measurment_Error(Measure_Error,...
      CalibGrid1.Measure_Max.Measure_Var.Z_Max(1,2));
[Measured_Points2] = Apply_Measurment_Error(Measure_Error,...
      CalibGrid2.Measure_Max.Measure_Var.Z_Max(1,3));

% Translate the pure undistorted calibration points to origin Fe in world coordinates.
for i=1:length(CalibGrid0)
    CalibGrid0(:,i)=CalibGrid0(:,i)-Twe_w;
end;
for i=1:length(CalibGrid1)
    CalibGrid1(:,i)=CalibGrid1(:,i)-Twe_w;
end;
for i=1:length(CalibGrid2)
    CalibGrid2(:,i)=CalibGrid2(:,i)-Twe_w;
end;

% Translate the distorted calibration points to origin Fe in world coordinates.
for i=1:length(CalibGrid0)
    Measured_Points0(:,i)=Measured_Points0(:,i)-Twe_w;
end;
for i=1:length(CalibGrid1)
    Measured_Points1(:,i)=Measured_Points1(:,i)-Twe_w;
end;
for i=1:length(CalibGrid2)
    Measured_Points2(:,i)=Measured_Points2(:,i)-Twe_w;
end;

```

```

%      Now transform (rotate) the undistorted target points into Fe at Oc_c.
%      Consider these as Pure points, control points for comparison later.

TgtPts_Level0_InFc_P = Iw_c*CalibGrid0;
TgtPts_Level1_InFc_P = Iw_c*CalibGrid1;
TgtPts_Level2_InFc_P = Iw_c*CalibGrid2;

%      Now transform (rotate) the distorted target points into Fe at Oc_c.

TgtPts_Level0_InFc = Iw_c*Measured_Points0;
TgtPts_Level1_InFc = Iw_c*Measured_Points1;
TgtPts_Level2_InFc = Iw_c*Measured_Points2;

CamPosnInFc=[Oc_c];

TgtCenInFc=[Iw_c*Tej_w];
CamToTgtCen=[CamPosnInFc,TgtCenInFc];
figure;
plot3(TgtPts_Level0_InFc(1,:),TgtPts_Level0_InFc(2,:),TgtPts_Level0_InFc(3,:), 'r');
title('Camera Position and Optical Axis Relative to Target Pattern in Fe');
xlabel('x-axis');
ylabel('y-axis');
zlabel('z-axis');
grid on
hold on
plot3(TgtPts_Level1_InFc(1,:),TgtPts_Level1_InFc(2,:),TgtPts_Level1_InFc(3,:), 'g+');
plot3(TgtPts_Level2_InFc(1,:),TgtPts_Level2_InFc(2,:),TgtPts_Level2_InFc(3,:), 'bd');
plot3(CamPosnInFc(1,1),CamPosnInFc(2,1),CamPosnInFc(3,1), 'gh');
hold on
plot3(CamToTgtCen(1,:),CamToTgtCen(2,:),CamToTgtCen(3,:), 'b-');
hold off

%      Xu and Yu are values of points to get 2D image coordinate points.
[Xu_Level0,Yu_Level0,labels0] = Convert_to_2D_Image(TgtPts_Level0_InFc,labels0,f,1);
[Xu_Level1,Yu_Level1,labels1] = Convert_to_2D_Image(TgtPts_Level1_InFc,labels1,f,2);
[Xu_Level2,Yu_Level2,labels2] = Convert_to_2D_Image(TgtPts_Level2_InFc,labels2,f,3);

if (Yes_Noise==1)
%      Applying Gaussian Noise to points in Fe
[Xu_Level0,Yu_Level0] = Apply_Noise(Xu_Level0,Yu_Level0,Max_Blunder,Blunder_Variance,dx,dy);
[Xu_Level1,Yu_Level1] = Apply_Noise(Xu_Level1,Yu_Level1,Max_Blunder,Blunder_Variance,dx,dy);
[Xu_Level2,Yu_Level2] = Apply_Noise(Xu_Level2,Yu_Level2,Max_Blunder,Blunder_Variance,dx,dy);
end;

[Xu_Level0_P,Yu_Level0_P,labels0_P] = Convert_to_2D_Image(TgtPts_Level0_InFc_P,labels0,f,1);
[Xu_Level1_P,Yu_Level1_P,labels1_P] = Convert_to_2D_Image(TgtPts_Level1_InFc_P,labels1,f,2);
[Xu_Level2_P,Yu_Level2_P,labels2_P] = Convert_to_2D_Image(TgtPts_Level2_InFc_P,labels2,f,3);

% GroupNoiseData = [Xu_Level0;Yu_Level0;Xu_Level1;Yu_Level1;Xu_Level2;Yu_Level2];
% GroupPureData =
% Xu_Level0_P;Yu_Level0_P;Xu_Level1_P;Yu_Level1_P;Xu_Level2_P;Yu_Level2_P];

Protected_Level0 = [Xu_Level0_P;Yu_Level0_P];

%      Inserting a radial and tangential distortion model here to metric coordinates.
[Xd_Level0,Yd_Level0,DistortProlem0]=Distort_Grid_Pts(Xu_Level0,Yu_Level0,sx,K1,K2,K4,K5);

```

```
[Xd_Level1,Yd_Level1,DistortProlem1]=Distort_Grid_Pts(Xu_Level1,Yu_Level1,sx,K1,K2,K4,K5);
[Xd_Level2,Yd_Level2,DistortProlem2]=Distort_Grid_Pts(Xu_Level2,Yu_Level2,sx,K1,K2,K4,K5);
```

```
Distorted_Level0 = [Xd_Level0;Yd_Level0]; % Used in SeeDistortion function.
```

```
[garb1,garb2]=See_Distortion(Distorted_Level0,Protected_Level0,...
                             K1,K2,K4,K5,Yes_Noise,Measure_Error);
```

```
% garb1,garb2 are not used anywhere.
```

```
% Digitize the converted points to CCD array coordinates
```

```
[CCD_Level0]=Convert_To_CCD(Xd_Level0,Yd_Level0,sx,dx,dy);
```

```
[CCD_Level1]=Convert_To_CCD(Xd_Level1,Yd_Level1,sx,dx,dy);
```

```
[CCD_Level2]=Convert_To_CCD(Xd_Level2,Yd_Level2,sx,dx,dy);
```

```
% Digitize the Pure undistorted points to CCD array coordinates
```

```
[CCD_Level0_U]=Convert_To_CCD(Xu_Level0_P,Yu_Level0_P,sx,dx,dy);
```

```
[CCD_Level1_U]=Convert_To_CCD(Xu_Level1_P,Yu_Level1_P,sx,dx,dy);
```

```
[CCD_Level2_U]=Convert_To_CCD(Xu_Level2_P,Yu_Level2_P,sx,dx,dy);
```

```
figure
```

```
plot(CCD_Level0(1,:),CCD_Level0(2,:), 'r');
```

```
hold on
```

```
plot(CCD_Level1(1,:),CCD_Level1(2,:), 'g+');
```

```
plot(CCD_Level2(1,:),CCD_Level2(2,:), 'bd');
```

```
title('CCD Array Pattern for Target Points');
```

```
xlabel('x-axis');
```

```
ylabel('y-axis');
```

```
hold off
```

```
% Assume the centre of the array is coincident with the optical axis, for now
```

```
% Define the viewable area of the CCD Array.
```

```
% Used to eliminate calibration points outside the camera lens field of view (FOV)
```

```
xLimit = CCD_Array_Size(1,1)/2;
```

```
yLimit = CCD_Array_Size(1,2)/2;
```

```
ViewableBoundary = [-xLimit,-xLimit,xLimit, xLimit, -xLimit;
```

```
                  -yLimit, yLimit,yLimit,-yLimit,-yLimit];
```

```
ViewableBoundary1 = [-xLimit*dx,-xLimit*dx,xLimit*dx, xLimit*dx, -xLimit*dx;
```

```
                  -yLimit*dy, yLimit*dy,yLimit*dy,-yLimit*dy,-yLimit*dy];
```

```
CrossHairX=[-xLimit,xLimit;0,0];
```

```
CrossHairY=[0,0;yLimit,-yLimit];
```

```
CrossHair1_X=[-xLimit*dx,xLimit*dx;0,0];
```

```
CrossHair1_Y=[0,0;yLimit*dy,-yLimit*dy];
```

```
% [GpData] = Before_CCD_Conversion(GroupNoiseData,...
```

```
% CrossHair1_X,CrossHair1_Y,ViewableBoundary1,xLimit,yLimit,dx,dy);
```

```
% GpData: Waste variable
```

```
dispt('Do you wish to see the viewable CCD area overlaid on the target points in Fe?);
```

```
CCDPattern = input('To see the viewable CCD area, press 1 and enter ... else press 0 and enter ...');
```

```
if CCPattern==1
```

```
    figure
```

```
    plot(CCD_Level0(1,:),CCD_Level0(2,:), 'r');
```

```
    title('All Target Points With Viewable CCD Array Area and Optical Centre');
```

```

xlabel('x-axis');
ylabel('y-axis');
hold on
plot(CCD_Level1(1,:),CCD_Level1(2,:),'g+');
    plot(CCD_Level2(1,:),CCD_Level2(2,:),'bd');
plot(ViewableBoundary(1,:),ViewableBoundary(2,:),'k-');
plot(CrossHairX(1,:),CrossHairX(2,:),'r-');
hold on
plot(CrossHairY(1,:),CrossHairY(2,:),'r-');
hold off
end;
%      Determine which distorted and undistorted calibration points are
%      within the field of view

% [Level0_CCDPoints,labels0]=CCD_Points_in_FOV_1(CCD_Level0,labels0,xLimit,yLimit);
% [Level1_CCDPoints,labels1]=CCD_Points_in_FOV_1(CCD_Level1,labels1,xLimit,yLimit);
% [Level2_CCDPoints,labels2]=CCD_Points_in_FOV_1(CCD_Level2,labels2,xLimit,yLimit);

[Level0_CCDPoints,Level0_U_CCDPoints,labels0]=...
    CCD_Points_in_FOV_2(CCD_Level0,CCD_Level0_U,labels0,xLimit,yLimit);
[Level1_CCDPoints,Level1_U_CCDPoints,labels1]=...
    CCD_Points_in_FOV_2(CCD_Level1,CCD_Level1_U,labels1,xLimit,yLimit);
[Level2_CCDPoints,Level2_U_CCDPoints,labels2]=...
    CCD_Points_in_FOV_2(CCD_Level2,CCD_Level2_U,labels2,xLimit,yLimit);
disp('Do you wish to see the viewable CCD points only?');
PointsinArray = input('To see the viewable CCD points, press 1 and enter ...   else press 0 and enter ...');
if PointsinArray==1
    figure
    plot(Level0_CCDPoints(1,:),Level0_CCDPoints(2,:),'r-');
    title('Viewable Target Points Within CCD Array Area and Optical Centre');
    xlabel('x-axis');
    ylabel('y-axis');
    hold on
    plot(Level1_CCDPoints(1,:),Level1_CCDPoints(2,:),'g+');
    plot(Level2_CCDPoints(1,:),Level2_CCDPoints(2,:),'bd');
    plot(ViewableBoundary(1,:),ViewableBoundary(2,:),'k-');
    plot(CrossHairX(1,:),CrossHairX(2,:),'r-');
    hold on
    plot(CrossHairY(1,:),CrossHairY(2,:),'r-');
    hold off
end;
%      CCD Array will adjust the position of the image points due
%      to the actual CCD array pixels
disp('Do you wish to consider CCD Array digitization error of points at this time?')
Yes_Pixel_Adjust = input('To see the error with digitization, press 1 and enter ...   else press 0 and enter ...');
if Yes_Pixel_Adjust == 1
[Level0_CCDPoints]=CCD_Pixel_Adjust(Level0_CCDPoints,0,xLimit,yLimit,dx,dy,...
    ViewableBoundary,CrossHairX,CrossHairY);
[Level1_CCDPoints]=CCD_Pixel_Adjust(Level1_CCDPoints,1,xLimit,yLimit,dx,dy,...
    ViewableBoundary,CrossHairX,CrossHairY);
[Level2_CCDPoints]=CCD_Pixel_Adjust(Level2_CCDPoints,2,xLimit,yLimit,dx,dy,...
    ViewableBoundary,CrossHairX,CrossHairY);
end;

```

Coordinate Modification: Measurement Error

```

function [New_Grid_Data] = Apply_Measurment_Error(Measure_Error,...
    Grid_Data,Measure_Max,Measure_Var,Z_Max);
% Purpose: Apply a Measurement Noise to the calibration points as though the
%           points were not measured with strict care.
%           Deviations will be mainly in x and y and general offset in z. Function to
%           apply noise to input data based upon a truncated normal Gaussian distribution
%
% Returns to:   CCD_ArrayPlane
%
% Inputs:
%
% Measure_Error Set to 1 to apply Gaussian normal noise error to calibration point data due to
%               inaccurate measurement of points.
% Grid_Data     3xN array of calibration points for level #
% Measure_Max   Sets the max error for point selection due to resolution noise.
%               Dimensions in meters (ie. 0.10 = 10 cm error)
% Measure_Var   Limits severe outliers to variance of random distribution.
% Z_Max         Z_Max Offset/measurement error of each calibration grid level elevation
%
% Variables:
% Xn=[]         Array to hold updated coordinates for x value of [x,y,z] 3w coordinates
% Yn=[]         Array to hold updated coordinates for y value of [x,y,z] 3w coordinates.
% X_Adjust      Value from random Gaussian normal function call.
% Y_Adjust      Value from random Gaussian normal function call.
%
% Outputs
%
% New_Grid_Data Output array of adjusted coordinates.
%
if (Measure_Error==1)
    if (abs(Measure_Var)>=1)
        error('Abort - Range for variance of measurement error >= 1.0. Check
variable. ');
    end;
    for i=1:length(Grid_Data)
        X_Adjust = randn;
        % While loop allows for a truncated normal distribution to avoid tails
        % of distribution curve.
        while (X_Adjust <= -Measure_Var | X_Adjust >= Measure_Var)
            X_Adjust = randn;
        end;
        Grid_Data(1,i) = (Grid_Data(1,i) + X_Adjust*Measure_Max);
        Y_Adjust = randn;
        while (Y_Adjust <= -Measure_Var | Y_Adjust >= Measure_Var)
            Y_Adjust = randn;
        end;
        Grid_Data(2,i) = (Grid_Data(2,i) + Y_Adjust*Measure_Max);
        Grid_Data(3,i) = (Grid_Data(3,i) + Z_Max);
    end;
    New_Grid_Data = Grid_Data;
else
    New_Grid_Data = Grid_Data;
end;

```

Coordinate Modification: Convert to Camera Reference Frame

```
function [CCD_Pts]=Convert_To_CCD(Xu,Yu,sx,dx,dy);
```

```
% Purpose: Main file used to control execution of program.
%           MatCam is a program to develop integrated input data to test/conduct
%           accuracy and parametric studies of camera calibration models
%
% Returns to:   CCD:ArrayPlane.
% Variables:
%           Camera Parameters:
%
% Xu           2D image coordinates.
% Yu           2D image coordinates.
% f            effective focal length.
% sx           Horizontal scale factor
% dx           Pixel-to-pixel metric spacing in x-direction (column-to-column)
% dy           Pixel-to-pixel metric spacing in y-direction (row-to-row).
%
% Outputs:
%
% CCD_Pts      2xN array of image points converted to pixel coordinates with the
%              centre of the optical axis as the centre of the CCD points.
```

```
CCD_Pts = zeros(2, length(Xu));
```

```
for i=1:length(Xu)
```

```
    CCD_Pts(1,i) = (Xu(1,i)*sx)/dx;
```

```
    CCD_Pts(2,i) = Yu(1,i)/dy;
```

```
end;
```

Coordinate Modification: Applying Resolution Noise

function [Xu,Yu] = Apply_Noise(Xu,Yu,Max_Blunder,Blunder_Variance,dx,dy):

```

% Purpose: Function to apply resolution noise to input data based upon
%          a truncated normal Gaussian distribution.
%
% Returns to: CCD_ArrayPlane
%
% Inputs:
%
% Grid_Data      3xN array of calibration points for level #
% Yes_Noise      Set to 1 to apply normal noise error to image point data due to
%                poor resolution of the image point centre in x and y-axis.
% Max_Blunder    Sets the max error for point selection due to resolution noise.
%                Dimensions in pixels.
% Blunder_Variance Limits blunders to variance of random distribution.
%
% Variables:
% Yn={}          Array to hold updated coordinates for y value of [x,y,z] 3c coordinates
% Zn={}          Array to hold updated coordinates for z value of [x,y,z] 3c coordinates
% maxVector      Determines maximum distance vector from all points in the level
%                Used to scale larger error error for points further away
% minVector      Determines minimum distance vector from all points in the level.
%                Not currently used. For future use
% Avg_Vector_Length Used to scale larger error error for points further away.
% X_Adjust       Value from random normal function call
% Y_Adjust       Value from random normal function call
%
% Outputs:
%
% New_Grid_Data  Output array of adjusted coordinates.
%
%
% if (Blunder_Variance==0)
%     error('Abort - Range for variance of distortion == 0. Check variable.');
```

end;

```

for i=1:length(Xu)
    X_Adjust = randn;
    % While loop allows for a truncated normal distribution to avoid tails
    % of distribution curve

    while (X_Adjust <= -Blunder_Variance | X_Adjust >= Blunder_Variance)
        X_Adjust = randn;
    end;
    Xu(i,i) = Xu(i,i)+X_Adjust*Max_Blunder*dx;
    Y_Adjust = randn;
    while (Y_Adjust <= -Blunder_Variance | Y_Adjust >= Blunder_Variance)
        Y_Adjust = randn;
    end;
    Yu(i,i) = Yu(i,i)+Y_Adjust*Max_Blunder*dy;
end;
```

Coordinate Modification: Convert 3D Points to 2D Image Points

```

function [Xu,Yu,New_Labels] = Convert_to_2D_Image(TgtPts_In_Fc,labels,f,level);
%
% Purpose: Convert 3D calibration points in Fc to 2D image coordinates.
%          If points end up on or behind the image plane, it causes problems so
%          a message is sent but that point is removed from the calibration process.
%
% Returns to:   CCD_ArrayPlane
% Input Variables:
% TgtPts_In_Fc  3xN array of calibration points in Fc coordinates
% f             effective focal length
% labels        1xN array of calibration point reference labels.
% level         Confirms which level of points are being converted.
%
% Outputs:
%
% Xu,Yu        1xn array of new 2D image coordinates
% New_Labels
%
% Internal Variables:
%
% Xu_Level     Sequential building of array elements for x coordinates
% Yu_Level     Sequential building of array elements for y coordinates
% BadPoints    Array to keep track of points that are either in line with
%              the image plane (x=0) or behind the image plane.

Xu_Level = zeros(1, length(TgtPts_In_Fc));
Yu_Level = zeros(1, length(TgtPts_In_Fc));

BadPoints = [];

for i=1:length(TgtPts_In_Fc)
    if (TgtPts_In_Fc(1,i)<=0)
        disp('Problem with point: ').disp(i)
        disp('on Calibration level: ').disp(level)
        disp('Point has been adjusted but will not be used in calibration. ')
        BadPoints = [BadPoints,i];
        Xu_Level(1,i) = Xu_Level(1,i);
        Yu_Level(1,i) = Yu_Level(1,i);
    else
        Xu_Level(1,i)=f*(TgtPts_In_Fc(2,i)/TgtPts_In_Fc(1,i)); % Here x is the optical axis
        Yu_Level(1,i)=f*(TgtPts_In_Fc(3,i)/TgtPts_In_Fc(1,i));
    end;
end;

Xu = [];
Yu = [];
New_Labels = [];
j = 1; % Counter for points that are in or behind image plane.
for i=1:length(TgtPts_In_Fc)
    if j <= length(BadPoints)
        if labels(1,i) == BadPoints(1,j)
            Xu=[Xu];
            Yu=[Yu];

```



```

                                j=j+1;
else
    Xu=[Xu,Xu_Level(1,i)];
    Yu=[Yu,Yu_Level(1,i)];
    New_Labels =[New_Labels,i];
end;
else
    Xu=[Xu,Xu_Level(1,i)];
    Yu=[Yu,Yu_Level(1,i)];
    New_Labels =[New_Labels,i];
end;
end;
end;
```

Coordinate Modification: Apply Distortion to Image Points

```

function [Xd,Yd,XdYdProblem]=Distort_Grid_Pts(Xu,Yu,sx,K1,K2,K4,K5);
%
% Purpose: Applies Distortion to 2D image points.
%
% Returns to: CCD_ArrayPlane.
% Variables:
%
% Xu      2D image coordinates
% Yu      2D image coordinates.
% sx      Horizontal scale factor
% K1      1st order radial distortion coefficient.
%          Magnitude keep between  $-1.e+03 \leq K1 \leq +1.e+03$ 
%          Negative K1 gives stronger pin-cushion distortion
%          Positive K1 gives stronger barrel distortion
% K2      2d order radial distortion coefficient.
%          Value of K2 must be  $> +$  or  $< -1.e+07$ 
% K3      Reserved for 3d order radial distortion coefficient (not modeled).
%
% K4      First coefficient of tangential distortion.
%          Magnitude keep between  $-1.0 \leq K4 \leq +1.0$ 
% K5      Second coefficient of tangential distortion.
%          Magnitude keep between  $-1.0 \leq K5 \leq +1.0$ 
%
% Outputs
%
% Xd      Distorted 2D image coordinates
% Yd      Distorted 2D image coordinates
%
% Functions Used:
%          Undistorted_To_Distorted Actual working part of the distortion.
%
Xd=[];
Yd=[];
XdYdProblem=[];
for i=1:length(Xu)
    [Xnew,Ynew,problem,difference] = Undistorted_To_Distorted(Xu(i),Yu(i),K1,K2,K4,K5);
    if problem==1
        XdYdProblem=[XdYdProblem,i]
    end;
    Xd = [Xd, Xnew];
    Yd = [Yd, Ynew];
end;

```

Coordinate Modification: Convert Undistorted Points to Distorted

```

function [Xd,Yd, NormError,error] = Undistorted_To_Distorted(Xu,Yu,K1,K2,K4,K5):
%
%   Purpose: Applies Distortion to 2D image points by using the Jacobian and a
%           Gauss-Newton iterative optimization.
%
%   Returns to:   Distort_Grid_Pts.
%   Variables:
%
%           Camera Parmeters:
%
%   Xu           2D image coordinates
%   Yu           2D image coordinates.
%   K1           1st order radial distortion coefficient.
%               Magnitude keep between -1.e+03< K1 < +1.e+03
%               Negative K1 gives stronger pin-cushion distortion
%               Positive K1 gives stronger barrel distortion
%   K2           2d order radial distortion coefficient.
%               Value of K2 must be > = or < -1.e+07
%   K3           Reserved for 3d order radial distortion coefficient (not modeled)
%
%   K4           First coefficient of tangential distortion
%               Magnitude keep between -1.0 < K4 < +3.0
%   K5           Second coefficient of tangential distortion.
%               Magnitude keep between -1.0 < K5 < +1.0
%
%   Outputs:
%
%   Xd           Distorted 2D image coordinates
%   Yd           Distorted 2D image coordinates
%
%   Internal Variables
%
%   iterations   Number of cycles for the optimization.
%   step         Relative size for improvement of solution to prevent jumping over
%   f_0          Mathematical expression of the function being used.
%   x_k          Temporary 2x1 array of image coordinates.
%   f_k          Temporary 2x1 array of recalculated image coordinates
%   dldx        Derivative expression used for Jacobian.
%   error        Used as a check to absolute difference below a threshold value

Max_iterations = 2000;
step = 0.7;

Xd = Xu;
Yd = Yu;
f_0 = [Xd+K1*Xd^3+K1*Xd*Yd^2+K2*Xd^5+2*K2*Xd^3*Yd^2+K2*Xd*Yd^4+...
      K4*Xd^2+K4*Yd^2+2*K4*Xd^2+2*K5*Xd*Yd-Xu;
      Yd+K1*Yd^3+K1*Yd*Xd^2+K2*Yd^5+2*K2*Yd^3*Xd^2+K2*Yd*Xd^4+...
      2*K4*Xd*Yd+K5*Xd^2+K5*Yd^2+2*K5*Yd^2-Yu];

x_k = [Xd;Yd];
f_k = f_0;

```

```

count=0; % Used in loop for control
error = norm(f_k);

while (abs(error)>1e-14 | count==Max_iterations)
    Xd = x_k(1);
    Yd = x_k(2);
    dfdx =
[ 1+3*K1*Xd^2+K1*Yd^2+5*K2*Xd^4+6*K2*Xd^2*Yd^2+K2*Yd^4+2*K4*Xd+4*K4*Xd+2*K5*Yd...
.
    2*K1*Xd*Yd+4*K2*Xd^3*Yd+4*K2*Xd*Yd^3+2*K4*Yd+2*K5*Xd:...
    2*K1*Yd*Xd+4*K2*Yd^3*Xd+4*K2*Yd*Xd^3+2*K4*Yd+2*K5*Xd....

    1+3*K1*Yd^2+K1*Xd^2+5*K2*Yd^4+6*K2*Yd^2*Xd^2+K2*Xd^4+2*K4*Xd+2*K5*Yd+4*
K5*Yd];

    x_k = x_k - step*inv(dfdx)*f_k;
    Xd = x_k(1);
    Yd = x_k(2);
    f_k = [Xd+K1*Xd^3+K1*Xd*Yd^2+K2*Xd^5+2*K2*Xd^3*Yd^2+K2*Xd*Yd^4+...
            K4*Xd^2+K4*Yd^2+2*K4*Xd^2+2*K5*Xd*Yd-Xu;
            Yd+K1*Yd^3+K1*Yd*Xd^2+K2*Yd^5+2*K2*Yd^3*Xd^2+K2*Yd*Xd^4+...
            K4*Xd^2+K4*Yd^2+2*K4*Xd^2+2*K5*Xd*Yd-Yu];
    count=count+1;
    error = norm(f_k);
end;

Xd = x_k(1);
Yd = x_k(2);
error = norm(f_k);
NormError=0;
if (count==Max_iterations)
    NormError=1;
    disp('Check U2D again')
    disp('Difference in normalization error is: '), disp(error);
end;

```

Coordinate Modification: Convert to CCD Coordinates

```
function [CCD_Pts]=Convert_To_CCD(Xu,Yu,sx,dx,dy);
```

```
% Purpose: Main file used to control execution of program.
%           MatCam is a program to develop integrated input data to test/conduct
%           accuracy and parametric studies of camera calibration models.
%
% Returns to:   CCD-ArrayPlane.
% Variables:
%           Camera Parmeters:
%
% Xu           2D image coordinates.
% Yu           2D image coordinates.
% f            effective focal length.
% sx           Horizontal scale factor.
% dx           Pixel-to-pixel metric spacing in x-direction (column-to-column).
% dy           Pixel-to-pixel metric spacing in y-direction (row-to-row).
%
% Outputs:
%
% CCD_Pts      2xN array of image points converted to pixel coordinates with the
%              centre of the optical axis as the centre of the CCD points.
```

```
CCD_Pts = zeros(2, length(Xu));
for i=1:length(Xu)
    CCD_Pts(1,i) = (Xu(1,i)*sx)/dx;
    CCD_Pts(2,i) = Yu(1,i)/dy;
end;
```

Coordinate Modification: Determine Points in Camera Field of View (Part 1)

```

function [CCD_Points_Field_of_View,New_Labels]=CCD_Points_in_FOV_1(CCD_Points,...
    labels,xLimit,yLimit);
%
% Purpose: Using the CCD array resolution limits, remove those calibration points
%          that would be outside the field of view.
%
% Returns to:    CCD_ArrayPlane
% Input Variables:
% CCD_Points    2xN array of all calibration points in CCD coordinates (origin at centre of FOV)
% labels        1xN array of calibration point reference labels.
% xLimit        Number of pixels from centre of array to left and right boudary edges of CCD
%               FOV.
% yLimit        Number of pixels from centre of array to upper and lower boudary edges of
%               CCD FOV.
%
% Outputs:
%
% CCD_Points_Field_of_View Updated 2xN array of all calibration points only in the FOV
% New_Labels              Updated 1xN array of calibration point reference labels in FOV
%
% Internal Variables:
%
% CCDxLeft        Left edge of CCD array boundary.
% CCDxRight       Right edge of CCD array boundary.
% CCDyTop         Upper edge of CCD array boundary.
% CCDyBottom      Lower edge of CCD array boundary.
% xValue          Expanding array of updated x coordinates
% yValue          Expanding array of updated y coordinates
%
CCDxLeft=-xLimit;
CCDxRight=xLimit;
CCDyTop=yLimit;
CCDyBottom=-yLimit;

xValue = [];
yValue = [];
New_Labels = [];

for i=1:length(CCD_Points)
    if (CCD_Points(1,i)<(CCDxLeft) | CCD_Points(1,i)>(CCDxRight) ...
        | CCD_Points(2,i)>(CCDyTop) | CCD_Points(2,i)<CCDyBottom)
        % No change to data set
    else
        xValue = [xValue, CCD_Points(1,i)];
        yValue = [yValue, CCD_Points(2,i)];
        New_Labels = [New_Labels, labels(1,i)];
    end;
end;

CCD_Points_Field_of_View = [xValue; yValue];

```

Coordinate Modification: Determine Points in Camera Field of View (Part 2)

```

function [CCD_Points_Field_of_View.New_UnDist_Pts.New_Labels]=...
    CCD_Points_in_FOV_2(CCD_Points.UnDist_Pts.labels,xLimit,yLimit);
%
% Purpose: Using the CCD array resolution limits, remove those calibration points
%           that would be outside the field of view.
%           Modified version of Function: CCD_Points_in_FOV_1.
%           Alternate function to CCD_Points_in_FOV_1.
%           This is used only for plotting of data only. Not necessary for overall
%           program.
%
% Returns to:   CCD_ArrayPlane
% Input Variables:
% CCD_Points    2xN array of all calibration points in CCD coordinates (origin at centre of FOV)
% UnDist_Pts    2xN array of all undistorted calibration points in CCD coordinates
% labels        1xN array of calibration point reference labels.
% xLimit        Number of pixels from centre of array to left and right boudary edges of CCD
%               FOV.
% yLimit        Number of pixels from centre of array to upper and lower boudary edges of
%               CCD FOV.
%
% Outputs:
%
% CCD_Points_Field_of_View Updated 2xN array of all calibration points only in the FOV
% New_UnDist_Pts Updated 2xN array of all undistorted calibration points only in the FOV
% New_Labels Updated 1xN array of calibration point reference labels in FOV.
%
% Internal Variables
%
% CCDxLeft      Left edge of CCD array boundary.
% CCDxRight     Right edge of CCD array boundary.
% CCDyTop       Upper edge of CCD array boundary.
% CCDyBottom    Lower edge of CCD array boundary.
% xValue        Expanding array of updated x coordinates.
% yValue        Expanding array of updated y coordinates.
%
CCDxLeft=-xLimit;
CCDxRight=xLimit;
CCDyTop=yLimit;
CCDyBottom=-yLimit;

xValue = [];
yValue = [];
xdValue = [];
ydValue = [];
New_Labels = [];

for i=1:length(CCD_Points)
    if (CCD_Points(1,i)<(CCDxLeft) | CCD_Points(1,i)>(CCDxRight) ...
        | CCD_Points(2,i)>(CCDyTop) | CCD_Points(2,i)<CCDyBottom)
        % No change to data set
    else
        xValue = [xValue, CCD_Points(1,i)];
        yValue = [yValue, CCD_Points(2,i)];

```

```
    xdValue = [xdValue, UnDist_Pts(1,i)];  
    ydValue = [ydValue, UnDist_Pts(2,i)];  
    New_Labels = [New_Labels, labels(1,i)];  
end;  
end;  
  
CCD_Points_Field_of_View = [xValue; yValue];  
New_UnDist_Pts = [xdValue; ydValue];
```


Coordinate Modification: Adjust CCD Coordinates for Digitization Error

```

function [Distorted] = CCD_Pixel_Adjust(CCDPoints,level,xLimit,yLimit,dx,dy,...
    ViewableBoundary,CrossHairX,CrossHairY);
%
% Purpose: CCD Array will 'adjust' the position to distort CCD coordinates
%          to integer value to simulate digitization.
%
% Returns to:    CCD_ArrayPlane
% Input Variables:
% CCDPoints      2xN array of all calibration points in CCD coordinates (origin at centre of FOV).
% level          Used for plotting which level of calibration points adjusted
% xLimit         Number of pixels from centre of array to left and right boudary edges of CCD
%                FOV.
% yLimit         Number of pixels from centre of array to upper and lower boudary edges of
%                CCD FOV.
% dx             Pixel-to-pixel metric spacing in x-direction (column-to-column).
% dy             Pixel-to-pixel metric spacing in y-direction (row-to-row).
% CrossHairX     Used for plotting centre point of CCD array FOV.
% CrossHairY     Used for plotting centre point of CCD array FOV.
% ViewableBoundary Defines the boundary edges of CCD FOV.
%
% Outputs:
%
% Distorted      Updated 2xN array of all digitized distorted coordinates only in the FOV.
%
% Internal Variables
%
% Undistorted    Copy of CCDPoints for plotting purposes
%
% xValue         Expanding array of updated x coordinates.
% yValue         Expanding array of updated y coordinates.
% xNew           Temporary variable for adjusted x integer pixel coordinate.
% yNew           Temporary variable for adjusted y integer pixel coordinate.
%
Undistorted = CCDPoints;

xValue = [];
yValue = [];

for i=1:length(CCDPoints)
    if (CCDPoints(1,i)<=0)
        xNew = fix(CCDPoints(1,i));
        xValue = [xValue, xNew];
        if (CCDPoints(2,i)>=0)
            yNew = fix(CCDPoints(2,i));
            yValue = [yValue, yNew];
        else
            yNew = fix(CCDPoints(2,i));
            yValue = [yValue, yNew];
        end;
    else
        xNew = fix(CCDPoints(1,i));
        xValue = [xValue, xNew];
        if (CCDPoints(2,i)>=0)

```

```

        yNew = fix(CCDPoints(2,i));
        yValue = [yValue, yNew];
    else
        yNew = fix(CCDPoints(2,i));
        yValue = [yValue, yNew];
    end;
end;
end;

Distorted = [xValue; yValue];

figure
if level == 0
    title('Before Digitization Error (°) vs Digitization (°) for Level 0');
    plot(Distorted(1,:),Distorted(2,:), 'g+');
elseif level == 1
    title('Before Digitization Error (°) vs Digitization (°) for Level 1');
    plot(Distorted(1,:),Distorted(2,:), 'g+');
elseif level == 2
    title('Before Digitization Error (°) vs Digitization (°) for Level 2');
    plot(Distorted(1,:),Distorted(2,:), 'g+');
else
    title('No Digitization Error - Disregard Plot');
end;
xlabel('CCD Array x-axis');
ylabel('CCD Array y-axis');
hold on
plot(Undistorted(1,:),Undistorted(2,:), 'r');
plot(ViewableBoundary(1,:),ViewableBoundary(2,:), 'k');
plot(CrossHairX(1,:),CrossHairX(2,:), 'r');
plot(CrossHairY(1,:),CrossHairY(2,:), 'r');
hold off

```

Camera Model: Main Camera Model

```

function [Twe_w,Ie_w,f,sx,K1,K2,Rv,Confirm_Distortion_Points]=...
    Camera_Model(Level0_CCDPoints,Level1_CCDPoints,...
        Level2_CCDPoints,w_pts0,w_pts1,w_pts2);

% Copyright the Crown as represented by the Department of National Defence/
% Defence Research Establishment Suffield
%
% Purpose: Camera calibration model to interface with main control program.
%          M-file based on Tsai's camera calibration algorithm with a
%          modification made by Prof. A. C. Attkin, Carleton University.
%
% Returns to:            MatCam
% Input Variables:
%
% Level#_CCDPoints 2xN array of calibration points in FOV in CCD coordinates (origin at centre of
FOV)
% w_pts0                3xN array of calibration points in FOV in Fw.
%
% Outputs:
%
% Twe_w                Translation vector from world origin to camera frame origin in Fw
% Ie_w                Rotation matrix from camera to world frame
% f                    Effective focal length
% sx                   Horizontal scale factor
% K1                   1st order radial distortion coefficient
% K2                   2nd order radial distortion coefficient
% Rv                   2xN array of back projected points based on calibrated parameters.
Confirm_Distortion_Points Used to view and confirm accuracy of distortion model
%
% Internal Variables:
%
% Cx                   Principal point centre value for x-direction (columns,2) of computer frame
buffer
% Cy                   Principal point centre value for y-direction (rows,2) of computer frame buffer
% dx                   Pixel-to-pixel metric spacing in x-direction (column-to-column)
% dy                   Pixel-to-pixel metric spacing in y-direction (row-to-row)
% Nex                  Number of columns in the resolution of the camera sensor (CCD)
% Ntx                  Number of columns in the resolution of the computer frame buffer
% CCD_Array_Size Resolution of the camera sensor (CCD)
% Frame_Grabber_Resolution Resolution of the computer frame buffer
% dxp                   Adjusted value of dx when Nex != Ntx and accounts for when sx != 1
% Ivg                   Rotation matrix from a vehicle to a ground reference plane. For future use.
% Frame_ScaleX          Adjusts for variation in array resolutions.
% Frame_ScaleY          Adjusts for variation in array resolutions.
% cam_parms            Groups calibrated and known parameters to send to a function
% Igv_v                Translation vector from a vehicle to a ground reference plane. For future use.
% w_pts                Grouping of all levels of calibration points in Fw
% Im_test_pts           Takes im_pts0 and uses it in the back project function (b_get3D)
%
% Functions Used:
%                    Common_Intrinsic_Parameters Common values required for calibration.
%                    AdjustCCD_Points_for_Model Resets CCD coordinates from centre to top left

```

```

% Tsai_cal Main work of calibration model done here.
% Back_Project_Model_Data Back projection of points.

[Cx,Cy,dx,dy,Nex,Nfx,CCD_Array_Size,Frame_Grabber_Resolution]=Common_Intrinsic_Parameters;
%
% Need to adjust input data for Cx and Cy to match Camera model
% Frame buffer origin is at top left vice bottom left. This will adjust this.
% Also will adjust for frame buffer resolution where Levelx_CCDPoints comes in
% CCD array resolution.

[im_pts0]=AdjustCCD_Points_for_Model(Level0_CCDPoints,...
    CCD_Array_Size,Frame_Grabber_Resolution);
[im_pts1]=AdjustCCD_Points_for_Model(Level1_CCDPoints,...
    CCD_Array_Size,Frame_Grabber_Resolution);
[im_pts2]=AdjustCCD_Points_for_Model(Level2_CCDPoints,...
    CCD_Array_Size,Frame_Grabber_Resolution);

im_pts = [im_pts0; im_pts1; im_pts2];

w_pts0 = w_pts0'; % Transpose to get Nx3 matrices
w_pts1 = w_pts1';
w_pts2 = w_pts2';

w_pts = [w_pts0;w_pts1;w_pts2];
%
% Key Function to calibrate the camera system

[Two_w,lew,f,sx,K1,K2,dxp]=Tsai_cal(im_pts,w_pts,Nex,Nfx,dx,dy,Cx,Cy);
%
% Need to set some constants due to not using a vehicle reference frame
Ivg = eye(3); % No use to current application. Future option development
% Tgv_y corrects for the height of the calibration rig level 0 to allow back
% projection to the world plane where it is assumed z=0
Tgv_v = [0,0,0,00]';
%
Im_test_pts = im_pts0;

Confirm_Distortion_Points = im_pts0;
Frame_ScaleX = CCD_Array_Size(1,1)/(Frame_Grabber_Resolution(1,1));
Frame_ScaleY = CCD_Array_Size(1,2)/(Frame_Grabber_Resolution(1,2));
for i=1:length(im_pts0)
    Confirm_Distortion_Points(1,i) = ((Confirm_Distortion_Points(1,i)-Cx)*dxp/sx)*Frame_ScaleX;
    Confirm_Distortion_Points(2,i) = ((Confirm_Distortion_Points(2,i)-Cy)*dy)*Frame_ScaleY;
end;

cam_parms = [Nex,Nfx,dx,dy,sx,Cx,Cy,K1,K2,f];
%
Rx = Back_Project_Model_Data(Im_test_pts,cam_parms,Ivg,lew,Tgv_v,Two_w);

disp(Nex'),disp(Nex)
disp(Nfx'),disp(Nfx)
disp(dx'),disp(dx)
disp(dy'),disp(dy)
disp(sx/Horizontal scale factor'),disp(sx)
disp(Cx'),disp(Cx)
disp(Cy'),disp(Cy)

```

```

disp('K1/first order radial distortion:').disp(K1)
disp('K2/second order radial distortion:').disp(K2)
disp('1/Effective focal length:').disp(f)
disp('Rotation matrix from Camera to World'). disp(lew)
disp('Translation Vector from World to Camera'). disp(Twe_w)
%
%          Rv is the Nx3 matrix of the back projected points, must turn into 3xN
Rv = Rv';

```

Camera Model: Adjust CCD Coordinate Origin to Top Left

```

function
[Adjusted_Points]=AdjustCCD_Points_for_Model(CCDPoints,CCD_Array_Size,Frame_Grabber_Resolution);
%
% Purpose: Used to convert the CCD points which come in camera array resolution
%           to Frame Grabber resolution where Origin is top LEFT!!!!
%
% Returns to: Camera_Model
% Input Variables:
%
% CCDPoints      2xN array of calibration points in FOV in CCD coordinates (origin at centre of
% FOV).
% CCD_Array_Size Resolution of the camera sensor (CCD)
% Frame_Grabber_Resolution Resolution of the computer frame buffer
%
% Outputs
%
% Adjusted_Points 2xN array with CCD coordinate origin top left.
%
% Internal Variables:
%
% Adjust_Centre_X To conduct effect of decentering the principal point
%                 Will offset the data from principal point centre.
% Adjust_Centre_Y To conduct effect of decentering the principal point.
%                 Will offset the data from principal point centre.
% dx              Pixel-to-pixel metric spacing in x-direction (column-to-column)
% dy              Pixel-to-pixel metric spacing in y-direction (row-to-row)
% Nx              Number of columns in the resolution of the camera sensor (CCD).
% Ny              Number of columns in the resolution of the computer frame buffer.
%
% Used for study on principal point misalignment.
% global Adjust_Centre_X Adjust_Centre_Y;

Adjust_Centre_X = 0.0;
Adjust_Centre_Y = 0.0;

for i=1:length(CCDPoints)
    CCDPoints(1,i) = CCDPoints(1,i)*(Frame_Grabber_Resolution(1,1)/CCD_Array_Size(1,1));
    CCDPoints(2,i) = CCDPoints(2,i)*(Frame_Grabber_Resolution(1,2)/CCD_Array_Size(1,2));
end;

% Need to "flip" the Y-coordinate to match up to how the CCD array captures the data in
% real life.
Adjusted_Points = zeros(2,length(CCDPoints));

for i=1:length(Adjusted_Points)
    Adjusted_Points(1,i) = CCDPoints(1,i) + (Frame_Grabber_Resolution(1,1)/2) +
    Adjust_Centre_X;
    Adjusted_Points(2,i) = CCDPoints(2,i) + (Frame_Grabber_Resolution(1,2)/2) +
    Adjust_Centre_Y;
end;
Adjusted_Points = Adjusted_Points'; % Creates a Nx2 matrix

```

Camera Model: Main Calibration of Parameters

```

function [Twe_w,Ic_w,sx,K1,K2,dxp]=Tsai_cal(Im_pts,W_pts,Ncx,Nfx,dx,dy,Cx,Cy);

% Copyright the Crown as represented by the Department of National Defence/
% Defence Research Establishment Suffield.
%
% Purpose:      Execute Tsai's non-coplanar camera calibration algorithm.
%               Originally written by Professor V. Aitken.
%
% Returns to:   MatCam
% Input Variables:
%
% Im_pts       Nx2 matrix of frame buffer coordinates of calibration points
%               [col,row] for each point
% W_pts        Nx3 matrix of calibration points in world coordinate system
%               in units of metres.
% Ncx          Number of pixel elements in horizontal-direction for CCD array
% Nfx          Number of pixel elements in x-direction of frame buffer
% dx           Pixel-to-pixel metric spacing in x-direction (column-to-column)
% dy           Pixel-to-pixel metric spacing in y-direction (row-to-row)
% Cx           Principal point centre value for x-direction (columns/2) of computer frame
buffer
% Cy           Principal point centre value for y-direction (rows/2) of computer frame buffer
%
% Outputs:
%
% Twe_w        Translation vector to/from world origin to camera frame origin in f/w
% Ic_w         Rotation matrix from camera to world frame
% t            Effective focal length
% sx           Horizontal scale factor
% K1           1st order radial distortion coefficient
% K2           2d order radial distortion coefficient
% dxp          Adjusted value of dx when Ncx ~= Nfx and accounts for when sx ~= 1
%
% Internal Variables
%
% tcale        Rotation from Tsai's camera frame to x-axis as optical axis
% N_iterations  Number of Gauss-Newton iterations for minimization.
% N            Number of calibration points used. Normally used for a real calibration.
% Indp_pts     Adjusted CCD coordinates for dxp.
% A            Matrix A of Ax=b.
% b            Matrix b of Ax=b.
% a            Least squares estimate of solution
% mag_Ty       Magnitude of Ty element from determined translation vector.
% r#           Element of 3x3 rotation matrix.
% rcal_w       Column vector for rotation matrix.
% rcal_w       Column vector for rotation matrix.
% kcal_w       Column vector for rotation matrix.
%
% Functions Used
%
% sharp        Assists in determining a 3x3 orthonormal matrix

```

```

Icale = [0, 0, 1; % Rotation from Tsai's camera frame to x-axis as optical axis.
         1, 0, 0;
         0, 1, 0];
N_iterations = 200; % Number of Gauss-Newton iterations for minimization
%
% Start Tsai's method
%
N = max(size(Im_pts)); % get number of calibration points
if (N <= 7),
    error('Not enough calibration points...aborting');
end;
%
% Stage 1, Orientation, Position (x,y), and Scale Factor
%
% Part a)
% Note: In the Monoview Noncoplanar calibration (Part G of Section II),
%       sx is initially set to be a perfect 1

dxp = dx*Nex/Nfx; % dx_prime in Tsai's report (Equation 6d)
% Compute matrix of Xd_prime and Yd_prime values
% Note type in Tsai's report...
%
Imdp_pts = (Im_pts - ones(N,1)*[Cx,Cy])*[dxp,0;0,dy];
%
% Part b) -setup
%
A = zeros(N,7);
b = zeros(N,1);
for (i=1:N),
    A(i,:) = [Imdp_pts(i,2)*W_pts(i,1),...
             Imdp_pts(i,2)*W_pts(i,2),...
             Imdp_pts(i,2)*W_pts(i,3),...
             Imdp_pts(i,2),...
             -Imdp_pts(i,1)*W_pts(i,1),...
             -Imdp_pts(i,1)*W_pts(i,2),...
             -Imdp_pts(i,1)*W_pts(i,3)];
    b(i) = Imdp_pts(i,1);
end;
a = inv(A'*A)*A'*b; % Least squares estimate of solution
%
% Now execute improvement to obtain a 1:3 orthogonal to
% a(5:7)...See notes June 22nd...use Gauss-Newton's method
% to solve Ricatti-type equation
xn = [a;0];
an = a;
Q = [zeros(3,4),eye(3);zeros(4,7)];
step = 0.8;
for (i=1:1000),
    Fn = [(A'*A + xn(8)*Q')*an-A'*b; an'*Q*an];
    Hn = [A'*A+xn(8)*Q', Q'*an; 2*an'*Q, 0];
    xn = xn - step*inv(Hn)*Fn;
    an = xn(1:7);
    error1 = norm(an - inv(A'*A+xn(8)*Q')\A'*b);
    error2 = an'*Q*an;
end;

```



```

a = xn(1:7);
%
% Part c)...compute |Ty|
%
mag_Ty = 1/sqrt(a(5)^2 + a(6)^2 + a(7)^2);
%
% Part d)...compute sign of Ty
%
Ty = mag_Ty;
r1 = a(1)*Ty;
r2 = a(2)*Ty;
r3 = a(3)*Ty;
r4 = a(5)*Ty;
r5 = a(6)*Ty;
r6 = a(7)*Ty;
Txp = a(4)*Ty;
[val,index] = max(diag(Imdp_pts*Imdp_pts')); % Pick point farthest from
% image centre
u = r1*W_ptst(index,1) + r2*W_ptst(index,2) + r3*W_ptst(index,3) + Txp;
v = r4*W_ptst(index,1) + r5*W_ptst(index,2) + r6*W_ptst(index,3) + Ty;
if ((sign(u) == sign(Imdp_pts(index,1))) & ...
    (sign(v) == sign(Imdp_pts(index,2)))).
    Ty_sign = 1;
else
    Ty_sign = -1;
end;
Ty = Ty_sign*mag_Ty;
%
% Part e)...determine scale factor
%
sx = sqrt(a(1)^2 + a(2)^2 + a(3)^2)*mag_Ty;
%
% Part f)...Rotation matrix and x-translation
%
a
ical_w = a(1:3)*Ty/sx;
jcal_w = a(5:7)*Ty;
keal_w = sharp(ical_w)*jcal_w;
keal_w = keal_w/norm(keal_w);
icalw = [ical_w,jcal_w,keal_w]; % Tsai's rotation matrix
Tx = a(4)*Ty/sx;
%
% Stage 2...Effective focal length, distortion coefficients and
% Tx, Ty, Tz
%
% Part g)...Initialize iterations for f and Tz
%
A = zeros(N,2);
b = zeros(N,1);
for (i=1:N),
    y = jcal_w*W_pts(i,:) + Ty;
    w = keal_w*W_pts(i,:);
    Y = Im_pts(i,2)-Cy;
    A(i,:) = [y,-dy*Y];
    b(i) = w*dy*Y;
end;

```

```

x0_fTz = inv(A'*A)*A'*b;
%
% Part g)...Minimize to find f,Tz, K1, and K2
%
% ...Why not use both X and Y equations ...
% ... Using both together seems to give singular Jn - Jn
% ... but this occurs because of poor scaling
%
c1 = 100000; % Variable to correct for scaling of Jacobian matrix

xn = [x0_fTz(2);x0_fTz(1);0;0]; % Initialize for iterations with
% xn = [Tz,f,K1,K2]
Fn = zeros(2*N,1); % Storage for function F
Jn = zeros(2*N,4); % Storage for Jacobian of F
step = 0.8;
for (n=1:N_iterations), % Loop for Gauss-Newton iterations
    for (i=1:N),
        alf_x = ical_w*W_pts(i,:) + Tx;
        alf_y = jcal_w*W_pts(i,:) + Ty;
        beta = kcal_w*W_pts(i,:);
        X = lm_pts(i,1)-Cx;
        Y = lm_pts(i,2)-Cy;
        xi_x = (1/sx)*dyp*X;
        xi_y = dy*Y;
        r_2 = xi_x^2 + xi_y^2;
        r_4 = r_2^2;
        zeta_x = xi_x*r_2;
        zeta_y = xi_y*r_2;
        mu_x = xi_x*r_4;
        mu_y = xi_y*r_4; % Now build Fn and Jn
        Fn(2*i-1:2*i) = [xn(2)*alf_x/(beta+xn(1))-xi_x*c1*zeta_x*xn(3)-c1*c1*mu_x*xn(4);...
            xn(2)*alf_y/(beta+xn(1))-xi_y*c1*zeta_y*xn(3)-c1*c1*mu_y*xn(4)];
        Jn(2*i-1:2*i,1) = [-xn(2)*alf_x/(beta+xn(1))^2;...
            -xn(2)*alf_y/(beta+xn(1))^2];
        Jn(2*i-1:2*i,2) = [alf_x/(beta+xn(1));...
            alf_y/(beta+xn(1))];
        Jn(2*i-1:2*i,3:4) = [-c1*zeta_x*c1*c1*mu_x*c1*zeta_y*c1*c1*mu_y];
    end;
    xn = xn - step*inv(Jn'*Jn)*Jn'*Fn; % Gauss-Newton iteration
end;
%
% Collect output data
%
Tz = xn(1);
f = xn(2);
K1 = c1*xn(3); % Must correct K1 and K2
K2 = c1*c1*xn(4);
Tealw_cal = [Tx;Ty;Tz];
lew = lealw*leale';
Twe_w = lealw*(I-Tealw_cal);

```

Camera Model: Create Orthogonal Matrix

function A = sharp(a);

```
% Copyright the Crown as represented by the Department of National Defence/
% Defence Research Establishment Suffield.
%
% Purpose: This function returns the matrix cross product operator of the
%          3-element vector a in the matrix A.
%          Originally written by Prof. A.C. Aitken, Carleton University
%
% Returns to: Tsai, cal
% Input Variables: (Distance unit is metres)
%
% a          Elements of a 1x3 column vector.
%
% Output:
%
% A          3x3 matrix cross product.
%
A = [0, -a(3), a(2);
     a(3), 0, -a(1);
     -a(2), a(1), 0];
```

Camera Model: Back Project Image Points with Calibrated Parameters

```

function Rv = Back_Project_Model_Data(z,cam_parms,Ivg,Iev,Tgv_v,Tvc_v);

%
% Copyright the Crown as represented by the Department of National Defence/
% Defence Research Establishment Sufield.
% Purpose: Convert image frame buffer coordinates in z to
%          3D position vectors Rv with respect to vehicle or other reference frame.
%          Original use was with a vehicle reference plane and ground plane.
%          Have attempted to leave this option available for future use.
%          Originally written by Prof. V.C. Atken, Carleton University as BB_Get3D.
%
% Returns to: Camera_Model
% Input Variables:
%
% z          Image frame buffer measurements, each row contains (col,row) = (Xt,Yt)
% cam_parms  [Nex,Nfx,dx,dy,sx,Cx,Cy,K1,K2,L]
% Ivg        Rotation from Fv to Fg (v = vehicle, g = ground)
% Iev        Rotation from Fe to Fv
% Tgv_v      Translation from Og to Ov in Fv
% Tvc_v      Translation from Ov to Oc in Fv
%
% Outputs:
%
% Rv         Nx3 array where each row contains rv = 3D position vector of
%            corresponding point in z with respect to vehicle or other reference frame.
%
% Internal Variables
%
% e3         Used to hold the value of Z elevation stable.
% L          Effective focal length.
% Nd         Temporary variable for distorted x component.
% Yd         Temporary variable for distorted y component.
% Xu         Temporary variable for undistorted x component.
% Yu         Temporary variable for undistorted y component.
% re_tmp     Ray from origin of image plane to point coordinates in image frame.
% re_bar     Normalizes re_tmp.
% lambda     Scalar to allow intercept re_bar with object plane where Z of Fw = 0
% EachVector 3D coordinates of the intercept point of back projected ray and
%            world coordinate object plane.

e3 = [0;0;1];
Nex = cam_parms(1);
Nfx = cam_parms(2);
dx = cam_parms(3);
dy = cam_parms(4);
sx = cam_parms(5);
Cx = cam_parms(6);
Cy = cam_parms(7);
K1 = cam_parms(8);
K2 = cam_parms(9);
L = cam_parms(10);
dxdp = dx*Nex/Nfx;
%

```

```

[num_pts,garb] = size(z);
Rv = zeros(num_pts,3);

for (i=1:num_pts),
    Xd = (z(i,1)-Cx)*dyp/sx;
    Yd = (z(i,2)-Cy)*dy;
    xi_sqr = Xd*Xd + Yd*Yd;
    xi_sqr2 = xi_sqr*xi_sqr;
    Xu = Xd*(1+K1*xi_sqr+K2*xi_sqr2);
    Yu = Yd*(1+K1*xi_sqr+K2*xi_sqr2);
    re_tmp = [L;Xu;Yu];
    re_bar = re_tmp/norm(re_tmp);
    lambda = (-e3'*Ivg*(Tgv_v+Tvc_v))/(e3'*Ivg*Icv*re_bar);
    rv = Tvc_v + lambda*Icv*re_bar;
    Rv(i,:) = rv';
end;

```

Main program: Adjust Known Points With Calibrated for Comparison

```

function [CalibGrid_in_FOV] = Calibrate_Grid_Points_in_FOV(CalibGrid, labels);
%
%      Purpose: Used for amending the list of calibration points in the FOV due to any
%               changes made by Funtion "Modify_Image_Coverage".
%
%      Returns to:      MatCam
%      Input Variables:
%      CalibGrid       2xN array of all calibration points in CCD coordinates (origin at centre of FOV).
%      labels          1xN array of calibration point reference labels.
%
%      Outputs:
%
%      CalibGrid_in_FOV Updated 2xN array of all calibration points remaining in the FOV
%
%      Internal Variables
%
CalibGrid_in_FOV = [];
j = 1;
for i=1:length(CalibGrid)
    if (j <= length(labels))
        if (i == labels(1,j))
            CalibGrid_in_FOV = [CalibGrid_in_FOV; CalibGrid(i,:)];
            j = j+1;
        end;
    end;
end;

```

Main program: Back Project Image Points with Known Parameters

```

function [BackProjectPts]=Back_Project(CCDPoints,f,dx,dy,sx,Iw_c,Iw_j,Twc_w,Tcj_w);
%
% Purpose: Using the CCD coordinate values (distorted) from either the perspective
%          model or camera model under investigation transform the values into 3D
%          world points such that Z=0 and these points can be compared to the
%          true calibration points within the field of view.
%
% Returns to: MatCam
% Input Variables:
% f          effective focal length.
% dx         Pixel-to-pixel metric spacing in x-direction (column-to-column).
% dy         Pixel-to-pixel metric spacing in y-direction (row-to-row).
% sx         Horizontal scale factor.
% Iw_c       Rotation matrix from world to camera frame.
% Iw_j       Rotation matrix from world to jg frame. Currently set to eye(3).
% Twc_w      Translation vector to from world origin to camera frame origin in Fw.
% Tcj_w      Translation vector from camera origin to jg reference frame centre in Fc. (not
% used now - later)
%
% Outputs:
%
% BackProjectPts Updated 2xN array of all calibration points only in the FOV.
%
% Internal Variables:
%
% e3         Used to hold the value of Z elevation stable.
% Xu         Temporary variable for x component.
% Yu         Temporary variable for y component.
% ImagePlaneRay Ray from origin of image plane to point coordinates in image frame.
% UnitRay     Normalizes ImagePlaneRay.
% lamda      Scalar to allow intercept UnitRay with object plane where Z of Fw =0.
% EachVector  3D coordinates of the intercept point of back projected ray and
%             world coordinate object plane.
%
e3=[0;0;1];

BackProjectPts=zeros(length(CCDPoints),3);
for i=1:length(CCDPoints)
    CCDPoints(1,i) = CCDPoints(1,i)*(dx/sx);
    CCDPoints(2,i) = CCDPoints(2,i)*(dy);
end;

for i=1:length(BackProjectPts)
    Xu=CCDPoints(1,i);
    Yu=CCDPoints(2,i);
    ImagePlaneRay=[f;Xu;Yu];
    UnitRay=ImagePlaneRay/norm(ImagePlaneRay);
    lamda=(-e3'*(Twc_w))/(e3'*Iw_c.*UnitRay);
    EachVector=Twc_w+lamda*Iw_c.*UnitRay;
    BackProjectPts(i,:)=EachVector';
end;
BackProjectPts=BackProjectPts';

```

Main program: Remove Image Points From Calibration (If desired)

```

function [New_CCDPoints,Updated_Labels] = Modify_Image_Coverage(CCDPoints,labels,...
    Cx,Cy,Division_Factor,Row_Range,Colm_Range);

%
% Purpose: Remove "chunks" of image points from the field of view thereby testing the camera
%          model to determine the reaction because the full range of view is not
%          being used in the calibration process.
%
% Returns to:   MatCam
% Input Variables:
% CCD_Points    2xN array of all calibration points in CCD coordinates (origin at centre of FOV)
% labels        1xN array of calibration point reference labels.
% Cx            Principal point centre value for x-direction (columns/2) of computer frame
%               buffer.
% Cy            Principal point centre value for y-direction (rows/2) of computer frame buffer
% Division_Factor Used to modify the image coverage.
% Row_Range     User input for the range of rows to be removed (removing points) from the
%               image plane.
% Colm_Range    User input for the range of column to be removed (removing points) from the
%               image plane.
%
% Outputs:
%
% New_CCDPoints Updated 2xN array of new calibration points only in the modified FOV
% Updated_Labels Updated 1xN array of new calibration point reference labels in the modified
%               FOV
%
% Internal Variables.
%
% Num_Columns   Based upon value of Division_Factor, sets the number of columns
% Num_Rows      Based upon value of Division_Factor, sets the number of rows
% Col_Width     Sets column width in CCD value.
% Row_Height    Sets row height in CCD value.
% Colm_Bins     Creates a numbering system for number of columns.
% Row_Bins      Creates a numbering system for number of rows.
% CCDPoint_Sectors 2xN array of what row and column bin each point is in. Used to cross reference
%               for removal of the points in bins marked for removal.
%
Num_Columns = Division_Factor;
Num_Rows = Num_Columns;

Col_Width = (2*Cx)/Num_Columns;
Row_Height = (2*Cy)/Num_Rows;

Colm_Bins = zeros(3, Division_Factor);
Row_Bins = zeros(3, Division_Factor);

for i=1:length(Colm_Bins)
    j = i;
    Colm_Bins(1,i) = i;
    Colm_Bins(2,i) = (-Cx +(j-1)*Col_Width);
    Colm_Bins(3,i) = (-Cx +(i)*Col_Width);
end;
for i=1:length(Row_Bins)

```



```

j = i;
Row_Bins(1,i) = i;
Row_Bins(2,i) = (Cy -(i-1)*Row_Height);
Row_Bins(3,i) = (Cy -(i)*Row_Height);
end;
% Define the sectors (row/column matching) of calibration to be excluded/removed.
CCDPoint_Sectors = zeros(2,length(CCDPoints));
for i=1:length(CCDPoints)
    for j=1:length(Colm_Bins)
        for k=1:length(Row_Bins)
            if ((CCDPoints(2,i)<= Row_Bins(2,k)) & (CCDPoints(2,i) > Row_Bins(3,k)))
                CCDPoint_Sectors(2,i) = k;
            end;
        end;
        if ((CCDPoints(1,i)>= Colm_Bins(2,j)) & (CCDPoints(1,i) < Colm_Bins(3,j)))
            CCDPoint_Sectors(1,i) = j;
        end;
    end;
end;

Updated_Labels = [];

for i=1:length(CCDPoint_Sectors)
    if (CCDPoint_Sectors(2,i)>=Row_Range(1,1) & CCDPoint_Sectors(2,i)<=Row_Range(1,2))&...
        (CCDPoint_Sectors(1,i)>=Colm_Range(1,1) &
        CCDPoint_Sectors(1,i)<=Colm_Range(1,2))
    else
        Updated_Labels = [Updated_Labels, labels(1,i)];
    end;
end;
% Now update the list of labels and corresponding CCD points
CCDPoints = [CCDPoints;labels];
New_CCDPoints = [];
j = 1;
for i=1:length(CCDPoints)
    if (j<=length(Updated_Labels))
        if (Updated_Labels(1,j) == CCDPoints(3,i))
            New_CCDPoints = [New_CCDPoints; CCDPoints(1,i),CCDPoints(2,i)];
            j = j+1;
        end;
    end;
end;
New_CCDPoints = New_CCDPoints';

```

Main program: Error Analysis

```

function [Error]=Error_Analysist(labels,BackProjectPts,CalibGrid);
%
%      Purpose: Determines the metric error for each point back projected.
%
%
%      Returns to:      MatCam
%      Input Variables:
%      labels           1xN array of calibration point reference labels
%      BackProjectPts   2xN array of back projected coordinates in Fw.
%      CalibGrid        2xN array of matching calibration points with BackProjectPts.
%
%      Outputs:
%
%      Error           5xN array metric data containing:
%      Error(1,:)Point labels for reference.
%      Error(2,:)Magnitude of error in x values (in metres)
%      Error(3,:)Magnitude of error in y values (in metres)
%      Error(4,:)Magnitude of error in radial distance between points (in metres)
%      Error(5,:)Magnitude of error in radial distance between points (in millimetres)
%
%      Internal Variables:
%
%      xError          Array of error values in x coordinates.
%      yError          Array of error values in y coordinates
%
xError=zeros(1,length(BackProjectPts));
yError=zeros(1,length(BackProjectPts));
for i=1:length(xError)
    xError(1,i)=BackProjectPts(1,i)-CalibGrid(1,i);
    yError(1,i)=BackProjectPts(2,i)-CalibGrid(2,i);
end;
Error=zeros(5, length(xError));
%
for i=1:length(Error)
    Error(1,i)=labels(1,i); % Labeling the points
    Error(2,i)=xError(1,i);
    Error(3,i)=yError(1,i);
    Error(4,i)=sqrt(Error(2,i)^2 + Error(3,i)^2);
    Error(5,i)=Error(4,i)*1000; % Convert to mm error
end;

```