The Design of a Digital Photogrammetric Metrology System for the Semi-Automated Surveying and Recording of Pipe Dimensions in Industrial Plants

Submitted to the Department of Surveying and Geodetic Engineering in partial fulfilment of the requirements for the Degree of MSc (Eng) at the University of Cape Town

by

Mark Cammidge

11 September 1996

Abstract

This thesis reports on the design, development and testing of a semi-automated system to aid in the mapping of the interior of industrial plants. The system makes use of digital photogrammetry to assist an operator in locating and identifying components of the plants.

All of the important photogrammetric theory is discussed in the text, and explained in detail in the appendices. Specifically, this system implements various algorithms used for camera calibration, object point intersection, and a method combining the two techniques. Considerable use is made of the iterative least squares method, which is the basis of many of the algorithms employed in this work.

Image processing algorithms are implemented to enhance the digital images, and to ease the identification of objects in the images, and these are fully explained in the text. Adaptive least squares image matching is a method of matching corresponding points in different images and is used to ensure correspondence between points identified by the system operator. A weighted centre of gravity method is used to find the centre of target areas, and an algorithm is implemented to determine the radius, centre and direction of a pipe passing through a number of points.

Various aspects of the system design are discussed and explained. In particular the requirements in terms of hardware and software are presented. In addition, the choices of the operating system and of the compiler are justified. Potential problems with the system, and possible enhancements of it are also described.

Tests were performed to verify the correct operation of all of the algorithms used in the calibration of the cameras. Together with the point intersection routines, these tests calculated the position of various control points, the correct coordinates of which were previously known. The calculated point positions are compared to the known coordinates of the points to determine the accuracy of the various algorithms. Further tests were conducted to demonstrate and verify the ability of the system to measure distance in three dimensions. These tests illustrate that the accuracy achievable is approximately 0.05% of the total distance measured for an object occupying 80% of the width of the image.

The system improves considerably on the method presently used in South Africa and in many industries worldwide which rely on analytical photogrammetry for the determination of object point locations. While the system suffers from reduced accuracy as a result of the use of digital cameras, this problem will become less important as technology and digital camera resolution improve. Possible enhancements include the use of more numerically efficient algorithms, and the introduction of techniques that would partially automate the identification of control points and pipes.

Acknowledgments

I would like to thank Professor Heinz Rüther, my supervisor, for his assistance and guidance during my studies towards my MSc, and for the opportunities with which he provided me to broaden my field of knowledge.

I am also grateful to the other staff members of the Department of Surveying and Geodetic Engineering, to Mrs Val Atkinson who constantly assisted me with numerous administrative difficulties, and to all the postgraduate students in the department.

Finally, thanks to George Tattersfield for his expert proofreading of my thesis.

Contents

List of Figures

List of Tables

List of Symbols

(X, Y, Z): Variables representing the position of a point in object space relative to an arbitrary but fixed origin.

(x, y, z): Variables representing the position of a point in image space relative to the perspective centre.

(X_c, Y_c, Z_c): The object space coordinates of the perspective centre of a camera.

(x_p, y_p, c): The image space coordinates of the principal point on an image.

c (or f): The principal distance of a camera.

ω: Clockwise rotation of the image space axis system around the x-axis as seen from the positive x-axis side of the origin.

κ: Clockwise rotation of the once-rotated image space axis system around the y-axis as seen from the positive y-axis side of the origin.

ϕ: Clockwise rotation of the twice-rotated image space axis system around the z-axis as seen from the positive z-axis side of the origin.

$\mathbf{R}$: The rotation matrix which rotates the image space axis system from a position in which the axes are parallel to the object space axes to its final position.

$r_{11}, \ldots, r_{33}$: The elements of the rotation matrix $\mathbf{R}$.

$L_1, \ldots, L_{11}$: Parameters used by the DLT to describe the orientation of a camera.

dx, dy: Small change in the x and y coordinates respectively of an image point as a result of distortions introduced by the camera.

$\bar{x}$, $\bar{y}$: The x and y image coordinates of a point relative to the principal point on an image.

r: The radial distance between an image point and the principal point.

k_1, k_2, k_3: Parameters describing radial lens distortion.

P_1, P_2: Parameters describing de-centering lens distortion.

s_x: The correction to the scale in the x direction.

a: The correction to the shear of an image in the x direction. The symbol is also used in Smith's explicit space resection to represent the length of the side of a triangle.

θ_x, θ_y: The (anti-clockwise) rotations applied to the x and y axes respectively of an axis system by the affine transformation.

M_x, M_y: The scale factors by which the x and y axes respectively are multiplied as a result of the application of the affine transformation.

(x_t, y_t): The offset added to an axis system by the application of the affine transformation.

A, B, C: Point labels for points in object space.

$\mathbf{A}, \mathbf{B}, \mathbf{C}$: Vectors in object space from the object space origin to points A, B and C respectively. Alternatively these are used to refer to three matrices, each of which has more than one column.

$\mathbf{AB}, \mathbf{BC}$: Vectors in object space from point A to point B and from point B to point C respectively.

$\mathbf{x}, \mathbf{l}$: Matrices with one column, and any number of rows.

$\delta x_p, \delta y_p$, etc: Small corrections to the values of x_p, y_p, etc.

$|\mathbf{A}|$: The length of vector $\mathbf{A}$, or the Euclidean norm or the matrix $\mathbf{A}$.

Chapter 1

Introduction

1.1 Background

There are many companies worldwide which own, run, and maintain plants that are essentially networks of pipes and related equipment. Such plants are especially common in the chemical and the petro-chemical industries, but are also found in other fields.

The design of these plants is initially presented as a set of drawings or, more recently, as CAD models. It is inevitable that during the construction of the plant, the implementation diverges from the original plans due to unforeseen circumstances. Once the plant has been commissioned, repairs, alterations, and upgrades to the plant result in significant changes to its layout. As a result of these factors it has been found that, after many years of operation, the original plans for the plant are no longer suitable for use in the planning and implementation of further modifications and additions.

To expedite alterations to the plant, a three dimensional CAD model is required. It is generally more efficient to generate such a model using analytical photogrammetry to survey the plant, than it is to convert the existing plans to a CAD model. In addition, the use of photogrammetric techniques simplifies the inclusion of previous alterations in the new model.

As part of this thesis, a basic system, using digital photogrammetry in an integrated software package, was designed in response to a request from a local engineering company, where a department specialised in the mapping of chemical plants has been established. The software makes use of a number of digital images of the area being mapped to determine the positions of points identified in the images by the operator.

This basic package was then enhanced to include a variety of photogrammetric, image processing and geometrical techniques to simplify the mapping of plants.

This thesis details the algorithms used, the design of the system and various tests to illustrate and verify the capabilities of the finished product.

1.2 Aim of the Thesis

This thesis aims to create a semi-automated computerised system which will expedite the generation of three dimensional models of chemical and other industrial plants from digital images.

The program should be an interactive, menu driven system which implements:

- Image enhancement algorithms to make target identification easier for the operator;
- Photogrammetric algorithms for calculating the orientation of the cameras from the image space and object space coordinates of a number of points identified by the user;
- Algorithms for locating the positions in space of points identified by the user on the images;
- Algorithms to analyse the geometric relationship between surveyed object points, in order to simplify the identification of objects in the plant.

These features are described in more detail in later chapters.

While the system is aimed at assisting an operator in the creation of a three dimensional CAD model of plants, it should also be feasible to use the software to perform any general purpose photogrammetric measurement using digital images.

1.3 Thesis Outline

The thesis is divided into a further seven chapters, excluding the appendices:

Chapter 2 describes the photogrammetric techniques used in this project to determine camera orientations and point positions.

Chapter 3 deals with the image processing algorithms used to enhance the images in order to simplify the identification of points by the system operator.

Chapter 4 presents a number of other algorithms used in the project which do not fall into either of the above two categories. In particular it describes algorithms used for image matching, target centre location, and for finding circle and line orientations and positions.

Chapter 5 lists some of the problems encountered in the design of the system, and possible solutions to these problems. Where necessary, these solutions were implemented, and the justification for doing so or for leaving the system unchanged is given in each instance.

Chapter 6 describes the design of the system. It lists the hardware and software required by the system, and discusses the choice of programming software. Also described are the options presented by the system to the user for image processing, camera calibration, object point intersection, and circle and line determination.

Chapter 7 reports on a number of tests used to verify the accuracy of the photogrammetric algorithms, and to assess the ability of the system to measure distance in three dimensions.

Chapter 8 draws some conclusions regarding the system and its capabilities, and makes some recommendations for future development of the system.

The appendices describe in detail some of the important algorithms used in the project which are not given in detail in the main text because of their length. Also presented is a detailed listing of the test results which are summarised in chapter 7.

1.4 Related Work

A variety of development work has been done, and is being done, in the field of industrial visualisation and mapping, and the use of analytical photogrammetry for the mapping of industrial plants appears to be widespread.

The company for which the basic system was designed presently makes use of analytical photogrammetry in the mapping of chemical plants. The procedure involves the use of two photographs, with a digitizer being used to identify points on the images. The identified points are used by software designed for this purpose (MCP 1986) to determine the orientation of the cameras, and the object space positions of the points in the photographs.

The generation of three dimensional computer graphics models using analytical photogrammetry is described by Littleworth & Chandler (1995) with particular applications to the mapping of pipes in an oil refinery. The paper also describes the use of photogrammetry in the generation of architectural plans for historic buildings. In (Littleworth, Stirling & Chandler 1992) the same techniques are applied to the mapping of the exterior of an aerodrome and to the generation of a digital terrain map of an industrial complex. Also described is the mapping of an industrial crane.

The role of photogrammetry in the mapping of nuclear environments is de-

scribed by Chapman, Deacon & Hamid (1992). Digital photogrammetry is used to assist in the generation of CAD models of radioactive plants because of the dangers associated with prolonged exposure to radiation. The presence of high levels of radiation precludes the use of standard photography and interferes with some digital imaging systems. Purpose-build imaging systems are therefore used to capture digital images. The system designed implements a variety of photogrammetric techniques, and allows for measurements to be made in object space.

A system using digital photogrammetry with scanned photographs of plants is described in (Hamit 1995). The system is an integrated photogrammetric measurement system and CAD package, which is aimed at the mapping of factories, chemical plants, refineries and nuclear power plants.

While these systems provide the means of measuring the positions of points in object space, few facilities are provided for determining the dimensions and orientations of structures in the plants, or for automating these measurements.

The use of conventional analogue photogrammetric measurements to determine the radius and direction of pipes in industrial plants is described by Krämer & Schöler (1980). A variety of techniques are presented for determining the position and size of pipes using known positions of a number of points on the surface of each pipe. Jones, Chapman & Hamid (1996) derive models describing the relationship between three dimensional primitives and their two dimensional projections, with particular reference to the measurement of pipes in industrial plants.

Research is currently in progress in the Department of Surveying and Geodetic Engineering at the University of Cape Town to develop methods of automating to some extent the determination of the orientation of pipes.

Chapter 2

Photogrammetric Theory

Photogrammetry is the field of study involving techniques to determine the positions and orientations of objects in three dimensions from photographic images. This chapter describes two principal types of algorithms:

- algorithms to determine the orientation and position of each of the cameras used to generate the images of a scene;
- algorithms to determine the position of the points of interest in each of the images.

The former are known as camera calibration algorithms and the latter as intersection algorithms. Only those algorithms which were used in this project will be discussed.

2.1 Camera Calibration Algorithms

Camera calibration is the process of determining the orientation and position of the camera. Nine parameters are used to specify the camera position. These parameters, described in detail by Kraus & Waldhäusl (1993), are illustrated in figure 2.1. The parameters consist of:

1. Three parameters to specify the position of the perspective centre of the camera. The three parameters represent the three space dimensions relative to an arbitrary but fixed origin. The cartesian axes, X, Y, and Z, of the object space coordinate system extend from this origin and, within this system, the perspective centre is identified by (X_c, Y_c, Z_c);
2. One parameter to specify the perpendicular distance from the perspective centre to the imaging surface. This is known as the principal distance.

The principal distance is abbreviated as c in most parts of this text;

3. Two parameters which specify the position on the imaging surface of the intersection of the camera's optical axis with the image plane. This point on the surface is known as the principal point. The symbols x_p and y_p are used to represent the x and y coordinates of the principal point respectively, where x, y and z are the coordinate axes of the image space coordinate system whose origin is at the perspective centre. The principal point of the image is illustrated in figure 2.2. This diagram is an exaggeration of the typical situation, since the principal point is typically very close to the centre of the image.

4. Three parameters which specify the rotation of the image space coordinate system relative to the three object space axes. The parameters are identified by the symbols ω, κ, and ϕ. The image space axes are initially parallel to, but offset from, their corresponding object space axes, and are rotated before images are captured. In this text it is assumed that the rotation around the z axis (ϕ) is performed first, followed in order by κ, the rotation around the once-rotated y axis, and ω, the rotation around the twice-rotated x axis. Both the image space and object space coordinate systems are assumed to be left handed systems. The rotations of the axes are assumed to be clockwise as seen from the positive axis looking towards the origin of the axis system. In figure 2.1 these rotations have already been performed.

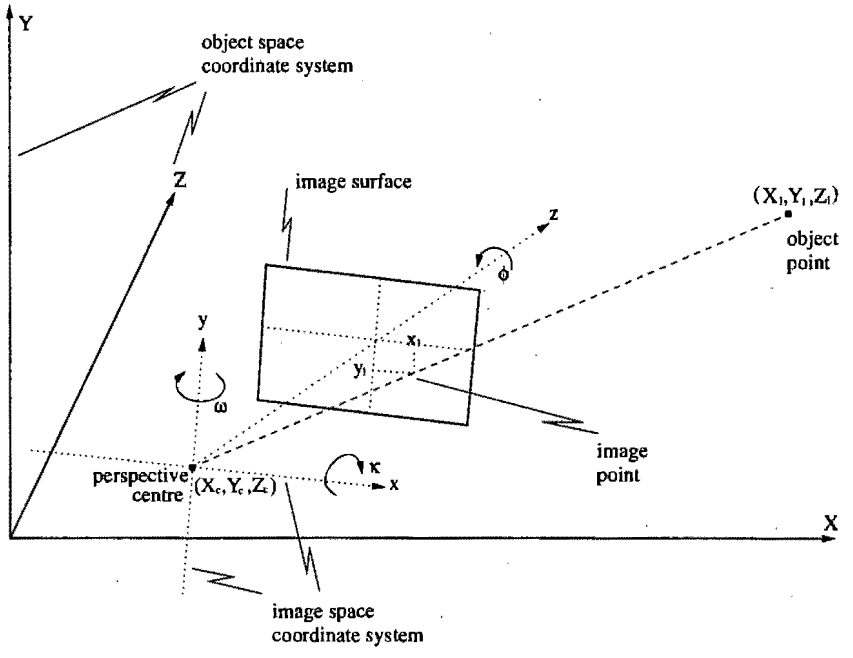

Figure 2.1: Camera Orientation Parameters

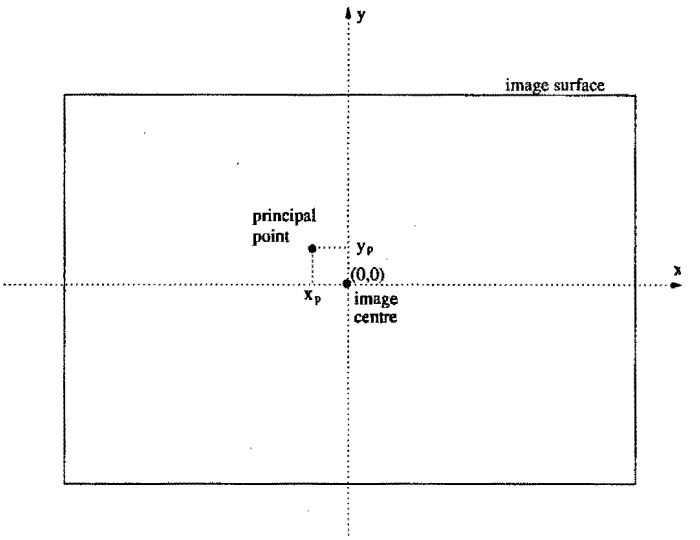

Figure 2.2: Principal Point of an Image

A number of additional parameters can be used to describe imperfections in the camera. These account for distortion of the image introduced by imperfections in the camera lens system. Distortion of the film, or in the case of digital cameras, distortion of the imaging surface can also be accounted for in this manner. The model used to describe these distortions is discussed in more detail in section 2.4.

The principal distance, principal point and any additional parameters used are together known as the interior orientation of the camera, since they are determined entirely by the internal construction of the camera. The remaining parameters are collectively known as the exterior orientation of the camera.

The orientation angles are typically represented by a rotation matrix. This matrix maps coordinates relative to the perspective centre in object space onto coordinates in image space. The rotation matrix will generally be represented by the symbol **R** in this text. It is always a 3×3 matrix, and its elements are represented by the symbols $r_{11}, r_{12}, \ldots, r_{33}$.

Methods for determining the nine camera orientation parameters and the additional parameters are discussed in more detail in the following sections.

2.1.1 Camera Calibration using the Collinearity Equations

The collinearity equations, described in (Karara 1989) and in (Haralick & Shapiro 1993), are the basis for most photogrammetric techniques used in this

project. These equations (equation 2.1 and equation 2.2) describe the relationship between points in three dimensions (X, Y, Z), and the two dimensional images of these points (x, y). They describe mathematically the assumption that a point in object space (three dimensional space), its image, and the perspective centre of the imaging system must be collinear.

$$x - x_p + dx = c\frac{r_{11}(X - X_c) + r_{12}(Y - Y_c) + r_{13}(Z - Z_c)}{r_{31}(X - X_c) + r_{32}(Y - Y_c) + r_{33}(Z - Z_c)} \tag{2.1}$$

$$y - y_p + dy = c\frac{r_{21}(X - X_c) + r_{22}(Y - Y_c) + r_{23}(Z - Z_c)}{r_{31}(X - X_c) + r_{32}(Y - Y_c) + r_{33}(Z - Z_c)} \tag{2.2}$$

In equations 2.1 and 2.2, dx and dy refer to the error in the x and y position of a point in an image as a result of the lens and film distortions. The variable c represents the principal distance of the camera.

Assuming that the distortion terms are zero, this calibration technique requires knowledge of the locations of five points, the coordinates of which are known both in object space and in image space. Using these coordinates, ten equations in the nine unknown parameters can be formulated. This is one more equation than is needed to solve for the nine unknowns, so that only the x or y coordinate of the fifth point is required. If more than nine observations are available, a least squares adjustment is generally used to find a best-fit solution. If additional parameters are used to model the lens and film distortions, one additional equation is required for each additional parameter. This in turn requires one additional observation for each additional parameter.

The equations are generally linearised in order to simplify the solving of the equations and to facilitate finding an optimum solution using the least squares method. Once the equations have been linearised, the parameters can be approximated using an iterative procedure which finds the best approximation for the differential terms in the linearised equations. This does, however, require that initial estimates of the camera orientation parameters are available, and that these estimates are reasonably close to the correct values. If not, the iterative procedure will generally not converge to the correct solution. The initial estimates can be obtained using another camera calibration technique, such as the Direct Linear Transformation as described in 2.1.2. A detailed description of the use of the collinearity equations for solving for the orientation parameters is presented in appendix B.

2.1.2 Camera Calibration using the Direct Linear Transformation

The Direct Linear Transformation (DLT) uses eleven parameters to describe the orientation of the camera, and is used to provide initial estimates for the

camera orientation parameters employed in other camera calibration techniques. Equations 2.3 and 2.4 describe the DLT. A detailed description of the derivation of the DLT, and its use, is presented in (Karara 1989).

$$x - dx = \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z + 1} \tag{2.3}$$

$$y - dy = \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z + 1} \tag{2.4}$$

As is the case with the collinearity equations, a number of additional parameters can be used to describe the distortion introduced by the camera lenses and other camera components.

There are *at least* eleven unknown parameters which need to be determined. Therefore, in order to solve for these parameters, it is required that the positions of at least six control points be known. If no additional parameters are used, only the x or the y value of the last control point need be known. If additional control points are available, a least squares solution is used to determine the best values for the parameters.

In most applications, equations 2.3 and 2.4 are linearised and a least squares method is used to solve for the parameters in an iterative manner. The DLT parameters can also be obtained using a direct method. However, this method is generally not used, because the iterative method makes better use of additional observations to find the *best* parameter values. A detailed description of the solution of the DLT, and of the relationship between the DLT and the collinearity equations, is given in appendix C.

The DLT has the advantage that, unlike the collinearity equations, no initial estimate of the parameter is required. It does however require knowledge of the position of one more control point. Furthermore, a least squares solution for the DLT parameters does not give the best solution of the nine camera parameters used in the collinearity equations, when these are calculated from the DLT parameters. The DLT is thus generally used to provide initial estimates for the exterior and interior orientations, which are then used with the collinearity equations to refine the solutions.

2.1.3 Smith's Explicit Space Resection

Smith's resection is a simple method of determining the exterior orientation of a camera using only four control points. The method also allows interior orientation parameters which are known to be accurate to be incorporated easily into the solution. While the DLT can be altered for these purposes, such alterations are more difficult to implement.

Smith (1965) describes an explicit solution for the space resection of a single image. The method requires no initial estimate of the six exterior orientation parameters. Three control points are required for the resection, which yields a maximum of four solutions. A fourth control point in a suitable position is used to determine which of the four solutions is correct. The interior orientation of the camera is required to solve for the exterior orientation.

As with the DLT, an explicit solution is possible, thus eliminating the need for accurate initial estimates. However, it is not possible to determine the lens and film distortion parameters using this method, and as a result it is generally used as a method of obtaining initial estimates of the camera exterior orientations. Smith describes a way of using four or more points in a least squares adjustment to improve the accuracy of the solution, but simpler methods of including redundant observations exist.

A detailed description of Smith's explicit space resection, and the solution thereof, is provided in appendix D.

2.1.4 Schmid's Iterative Space Resection

Schmid's iterative space resection (Thompson 1966) is a simple solution to implement, and provides a method of using redundant observations to improve the accuracy of the parameters calculated using Smith's explicit resection.

Like Smith's explicit space resection, Schmid's space resection solves only for the exterior orientation of the camera, and requires that the interior orientation of the camera be known. The method is derived from the collinearity equations - equations 2.1 and 2.2. As with the camera calibration using the collinearity equations described in section 2.1.1, the collinearity equations are linearised, and an iterative method is used to update the parameters.

The method requires knowledge of the positions of three control points for a minimal solution. It also requires initial estimates for all of the exterior orientation parameters. An iterative least squares approach is used to produce an optimal solution if the positions of four or more points are known.

A more detailed description of the use of Schmid's iterative resection for solving for the exterior orientation of a camera is given in appendix E.

2.2 Object Point Intersection Methods

The object space coordinates of a point can be determined if the image coordinates of the point are known for two or more images, and if the camera orientation parameters for these images are also known. Methods for performing this calculation are referred to as intersection algorithms. They are generally

based on the same equations as are used in the camera calibration routines, but the values of the object space coordinates X, Y and Z are determined, rather than those of the camera orientation parameters.

Only two intersection methods were investigated. These are:

- Intersection using the Collinearity Equations (equations 2.1 and 2.2)
- Intersection using the Direct Linear Transformation (equations 2.3 and 2.4)

Either method can be used to find object space coordinates, irrespective of the algorithm used to calibrate the cameras. The DLT method should ideally be employed in cases where the DLT is used to calibrate the cameras, and the method based on the collinearity equations when the other calibration techniques described previously are used.

2.2.1 Intersection using the Collinearity Equations

The collinearity equations (2.1 and 2.2) relate the object space coordinates of a point to its image coordinates. The orientation parameters of a camera and the image coordinates of a point on that camera's imaging surface define a line in object space on which the point lies, as illustrated in figure 2.1.

If the object appears in another image, produced by a camera with a different orientation, the image coordinates of the point on this image and the orientation of this camera define a second line in object space, as shown in figure 2.3. This line will also ideally pass through the point in object space.

In virtually all real situations, the lines will pass close to one another near the true object space coordinates of the required point. In general, a least squares approach is used to find the object space coordinates which minimise the sum of the squares of the residuals between the known image coordinates of the point on each image, and the image coordinates calculated using equations 2.1 and 2.2.

At least two images are required to find the point coordinates, but additional images can be used to improve the accuracy of the final estimate of the point position.

Since the values of the camera orientations are held constant, the collinearity equations can be rewritten as linear equations, and the object space coordinates of a point being intersected can be found explicitly. In general, an iterative procedure is used once initial estimates have been found using the explicit method. The standard form of the collinearity equations is linearised using differentials, and the iterative method, as described in appendix B, is used to find the best

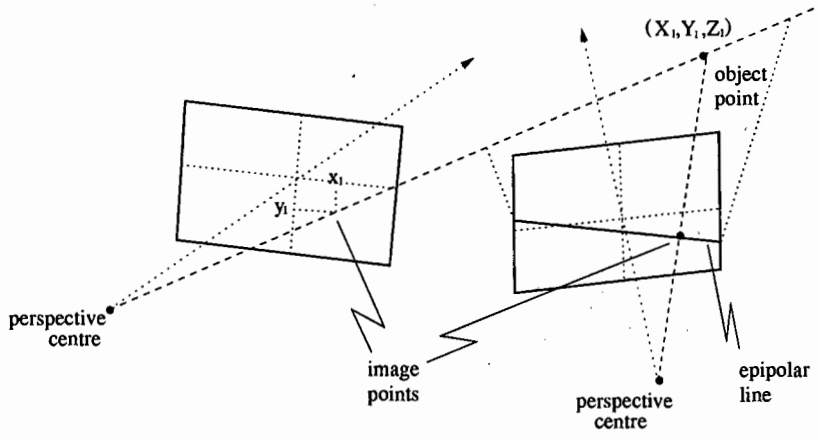

Figure 2.3: Epipolar Lines and Object Point Intersection

solution for the object space coordinates in the least squares sense. This effectively involves finding the best estimate of the point at which a number of lines intersect in object space.

The accuracy of the result is determined by the accuracy to which the image space coordinates of the points are known, the accuracy to which the camera orientations are known, and the positioning of the cameras. A larger number of images increases the number of redundant equations in the system, and thus also increases the accuracy of the solution.

2.2.2 Intersection using the Direct Linear Transformation

It is possible to use the Direct Linear Transformation equations (2.3 and 2.4) to solve for the object space coordinates of a point, if the point has been identified in two or more images. The DLT parameters for a camera and a point on the image obtained with this camera define a range of possible object space coordinates for the point in object space. These coordinates do not necessarily define a line in object space. Using the DLT parameters for a second camera, and the image space coordinates of the point on the image obtained from this camera, together with the corresponding information from the first camera, the object space coordinates of the point can be found.

A least squares solution is used to find the object space coordinates which minimise the sum of the squares of the residuals between the left and right hand sides of equations 2.3 and 2.4. An iterative approach is usually employed, using linearised DLT equations as described in appendix C. Since the camera orientation parameters are known, a linear equation can be formulated from the DLT equations to provide initial estimates of the coordinates.

As with the intersection using the collinearity equations, the accuracy of the object space coordinates is determined by the accuracy to which the positions of the image points are known, the accuracy of the camera orientation parameters, the position of the cameras and the number of images used.

2.3 Combined Methods

It is possible to solve simultaneously for both the camera orientations and the object space coordinates of points with known image coordinates, if more than one camera is present, and if there are a sufficient number of control points. The bundle adjustment is the only such method investigated.

2.3.1 The Bundle Adjustment

If the orientations of the cameras for a number of images are required, then the camera orientation parameters for all cameras can be solved simultaneously. In addition, if the positions of a number of points are to be found, these can be included in the adjustment, and found simultaneously. The adjustment, which is based on the collinearity equations, is then known as a bundle adjustment, a brief description of which is given in (Rüther 1994).

It is possible to leave as many or as few parameters as desired fixed or free, so that the values of accurately known parameters can be used to improve the solution. Moreover, certain parameters can be held fixed for certain cameras, and left free for other cameras.

It is, however, required that some fixed parameters are present in order to fix the scale of the system. Amongst other methods, it is possible to specify a set of distances between control points for this purpose. Granshaw (1980) describes this technique in more detail.

The disadvantage of the bundle adjustment over methods which perform each adjustment separately is that the time taken to solve the system increases dramatically with the number of cameras, and with the number of object points. Chapter 5 discusses the implementation problems which this presents.

A detailed description of the implementation and solution of the bundle adjustment is presented in appendix B.4.

A similar adjustment making use of the DLT is described by Dermanis (1994).

2.4 Modelling Lens and Other Distortions

Any number of parameters can be used to describe the distortion of the image introduced by imperfections in the camera lens, distortion of the imaging surface, and a number of other camera system quantities. When using digital cameras, other parameters can be included to account for errors resulting from electronic imperfections in the camera. Fryer & Brown (1986) and Brown (1966) describe various models of lens distortion for use in photogrammetric applications. Van der Vlught (1995) details the choice of a lens distortion model for digital cameras such as those used for this project.

When using models including a large number of additional parameters, sophisticated statistical testing of the significances of the parameters, as described by Gruen (1978), is required to avoid over-parametrisation of the system. Too many additional parameters in the model will result in unrealistically small error estimates for the camera parameters being found, and in inaccurate results when determining object space coordinates.

A number of simpler models were considered for this project, but the following model was used, since it accounts for distortions specific to digital cameras, and has been successfully used in similar applications:

$$dx = \bar{x}s_x + \bar{y}a + \bar{x}(k_1r^2 + k_2r^4 + k_3r^6) + P_1(r^2 + 2\bar{x}^2) + 2P_2\bar{x}\bar{y} \quad (2.5)$$

$$dy = \bar{x}a + \bar{y}(k_1r^2 + k_2r^4 + k_3r^6) + P_2(r^2 + 2\bar{y}^2) + 2P_1\bar{x}\bar{y} \quad (2.6)$$

where

- dx is the image distortion of a point in the x direction
- dy is the image distortion of a point in the y direction
- $\bar{x}$ is the x image coordinate of a point relative to the principal point
- $\bar{y}$ is the y image coordinate of a point relative to the principal point
- r is the radial distance between an image point and the principal point
- k_1, k_2, k_3 are radial lens distortion parameters
- P_1, P_2 are de-centering lens distortion parameters
- s_x is the correction to the scaling in the x direction
- a is the correction for the shear of the image in the x direction

The use of additional parameters improves the accuracy of the camera orientation parameters determined using either the collinearity equations or other methods. Their use also reduces the error in the object point coordinates calculated using the camera orientation parameters.

Chapter 3

Image Processing Theory

This section describes the image processing algorithms investigated and implemented as part of the system. Two sets of algorithms were implemented. These are:

- Algorithms to alter the colour map of the images in order to improve the ability of an operator to identify points on the images;
- Algorithms to enhance or to mask certain features of the images.

Only those algorithms which were implemented are described here. Many other algorithms can be used to simplify and partially automate the process of point location in images.

3.1 Colour Map Algorithms

3.1.1 Histogram Stretching

This is a standard algorithm which can be used to enhance the contrast in an image which occupies only a small, contiguous section of the available colour map. It involves a simple linear remapping of the colour map to occupy the entire available range. For photogrammetric applications, the colour map generally consists of a number of intensity levels or grey scales.

The effect of this algorithm on the colour histogram of an image is illustrated in figures 3.1 and 3.2. Figure 3.1 shows an example of an image's colour histogram before stretching. The stretched histogram of the same image is illustrated in figure 3.2.

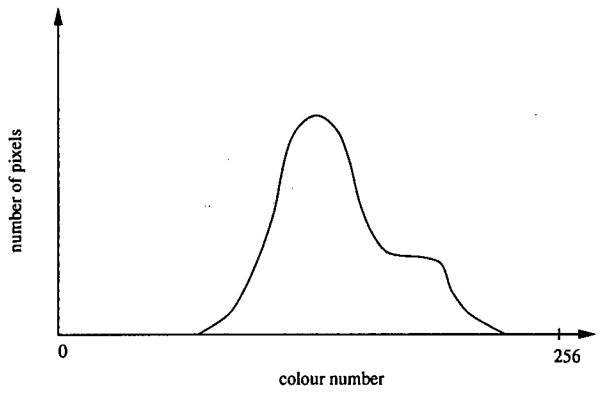

Figure 3.1: Example of Original Image Histogram

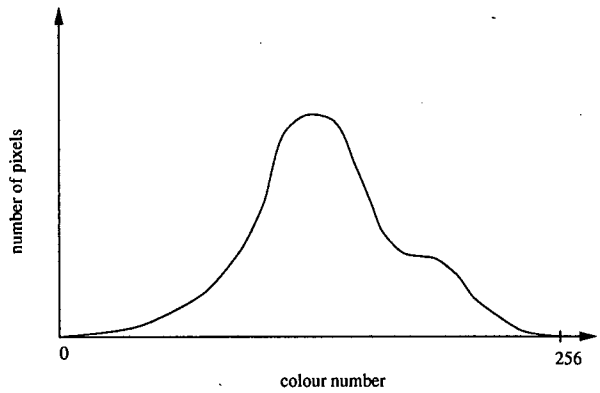

Figure 3.2: Histogram after Stretching

3.1.2 Histogram Equalisation

Histogram equalisation involves remapping the available grey scales to ensure that an equal number of pixels of each grey value exist. The input values to this mapping are the original grey values, and these are clearly discrete values. The mapping will therefore not, in general, produce an entirely flat histogram. The number of pixels in each of a set of grey value bins of equal size is equalised to approximate this mapping.

The effect of this mapping is to improve the contrast in the image, in a similar manner to the histogram stretching. Unlike the histogram stretching, the algorithm is effective even if the original image has a histogram which occupies the entire available colour range.

Histogram equalisation has the disadvantage that the resulting image appears unnatural. This is usually not a serious disadvantage, and the algorithm can be

used effectively to make objects in dark areas of an image more distinguishable.

The effect of histogram equalisation on the example histogram shown in figure 3.1 is illustrated in figure 3.3. Due to the discrete distribution of the grey values in the images, the resulting histogram is usually not completely flat, but exhibits some deviation from the desired histogram.

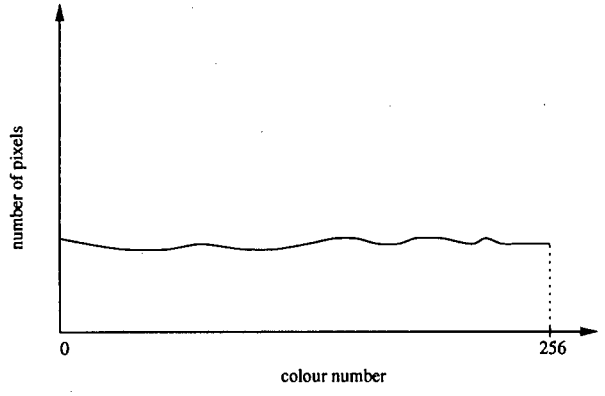

Figure 3.3: Example Image Histogram after Equalisation

3.2 Filtering Algorithms

Filtering algorithms are designed to enhance or to mask certain features of the image. All of the algorithms discussed here are implemented by convolving a small patch of the image with an filter mask of the same size, with the filter mask values (weights) determining the effect of the filter. Examples of filter masks are given in subsection 3.2.1 and subsection 3.2.2.

Convolution involves multiplying corresponding image points and filter values, and adding the resulting matrix of values. The result of this operation becomes the centre pixel of the patch on the filtered image. The filter mask is then shifted by one pixel, and the process is repeated, and in this manner the value of each pixel in the filtered image is calculated. A detailed description of the mathematical methods involved in linear filtering is presented in (Jähne 1991), and (Lim 1984) describes the effects of many standard image filters.

Of particular importance for photogrammetric applications is that filters should not cause any phase shift in the image. A phase shift will introduce a small change in the position of features in the image, introducing inaccuracies to subsequent image coordinates identified in the image. Since phase changes introduced by a filter would cause the image to look unnatural (de Jager 1995), standard image processing filter masks are designed to avoid this problem.

3.2.1 High Pass Filter

The high pass filter enhances high frequency components of the image, causing a sharpening of edges and enhancement of noise. For this thesis, the following high pass filter mask was used:

$$\begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$$

It should be noted that the sum of the filter values is 1 which results in low pass components of the image being retained unaltered, while the high pass components are amplified.

3.2.2 Low Pass Filter

Low pass filters attenuate high frequency components of the image, leaving the low frequency components unaltered. This blurs edges, and reduces the noise in the image.

The standard low pass filter mask used is:

$$\begin{matrix} 0.1 & 0.1 & 0.1 \\ 0.1 & 0.2 & 0.1 \\ 0.1 & 0.1 & 0.1 \end{matrix}$$

As with the high pass filter, the sum of the values of this filter is 1 which ensures that grey values in the filtered image occupy approximately the same range as those in the original image.

3.2.3 Anti-Sharp Mask

Anti-sharp masking is a filter which, as the name implies, masks the low pass components of the image. The effect of this is similar to the effect of a high pass filter, but it is more flexible since the degree of the high pass effect can be controlled using two parameters.

The masking is obtained by subtracting a low pass filtered copy of the image from the original image. The relative contribution of these two components of the filtered image is controlled by two parameters. It is often desirable to remap the histogram once this filter has been used, since the resulting image frequently uses a small range of the available grey values.

Chapter 4

Image Matching, Shape Fitting, and Target Centering

This chapter discusses the theory behind the remainder of the algorithms used in the system. In particular it discusses the least squares image matching algorithm, and the algorithms for finding a circle from three points, and a line from two points in three dimensions. It also describes techniques to locate the centre of the target points used as control points in the images.

4.1 Least Squares Image Matching

A method of obtaining the best possible match between images of the same point identified in two or more digital photographs is required for the photogrammetric determination of points. The least squares matching algorithm, described by Gruen (1985), can be used to obtain such a match, provided an initial estimate of the point position in the second image is known. The accuracy obtained using this algorithm is usually significantly better than one pixel.

The affine transformation, the use of which is described in (Monro & Dudbridge 1995), approximates the mapping of a small section of an image of an object to the image of the same scene as viewed from a different point. This approximation is only valid if the region of the surface being matched is small and approximately flat. In this application, a section of the second image is transformed so as to approximate the view of the same scene as seen in the first image.

Assume that the location of an object in one image is accurately known. The

offset of image coordinates in the first image relative to the object position is given by (x_1, y_1).

Assume that an approximation of the position of the object in a second image has the coordinates (m, n). A number of points around (m, n) comprising a block on the second image are transformed using the the affine transformation. The coordinates of these points relative to (m, n) are given by (x_2, y_2).

These points, and the two blocks of pixels are illustrated in figure 4.1.

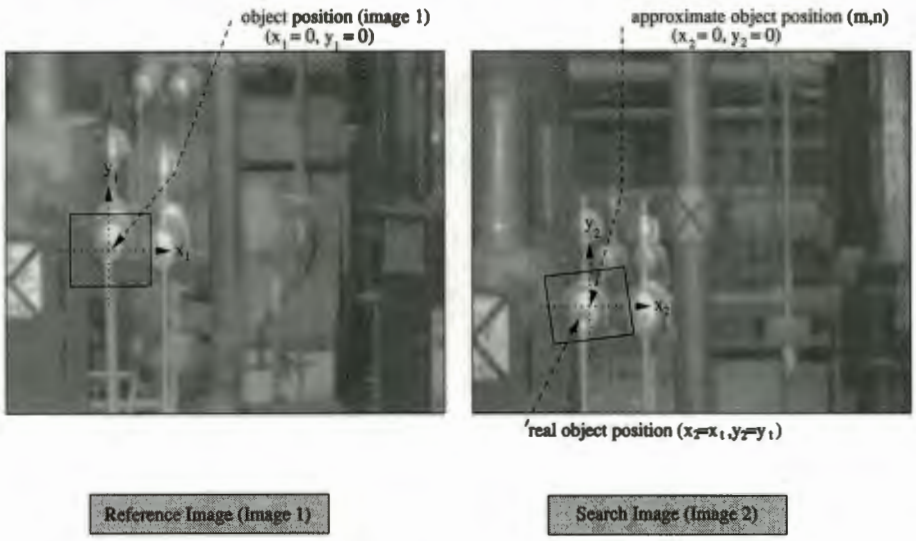

Figure 4.1: Search and Reference Images

It is assumed that a block of pixels surrounding the object on the first image is approximately the same as the affinely transformed block of pixels on the second image.

The affine transformation can be written as:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} M_x \cos\theta_x & -M_y \sin\theta_y \\ M_x \sin\theta_x & M_y \cos\theta_y \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{4.1}$$

The constants M_x and M_y are scale factors for the X and Y axes respectively, while θ_x and θ_y represent the rotations applied to the X and Y axes respectively. A translation is added to the coordinates after scaling and rotation of the coordinates, and this translation is represented by the constants x_t and y_t. The original axes, and the affinely transformed axes are illustrated in figure 4.2 and figure 4.3 respectively. The origin of the coordinate system becomes (x_t, y_t) after the transformation, as illustrated in figure 4.3. Since equation 4.1 contains six independent parameters, it can be rewritten as:

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} x_t \\ y_t \end{bmatrix} \tag{4.2}$$

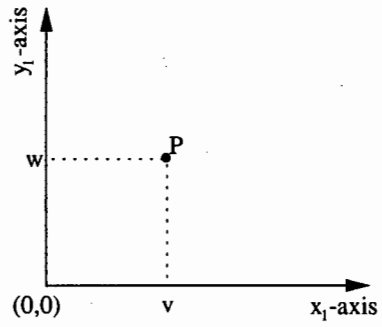

Figure 4.2: Axes Prior to Transformation

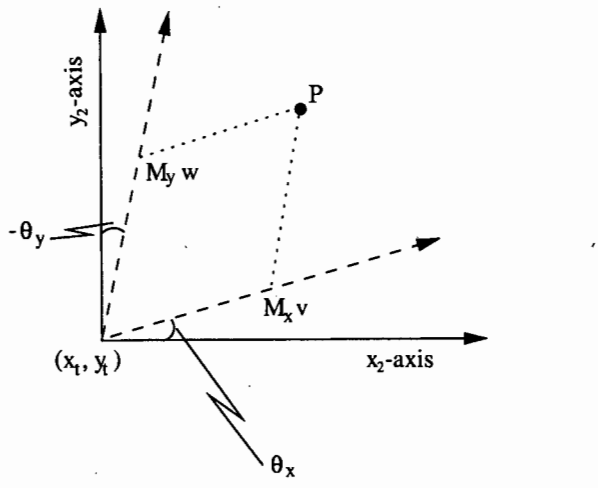

Figure 4.3: Axes After Affine Transformation

Equation 4.2 can be linearised and solved using an iterative linear least squares approach similar to the method used to solve the collinearity and DLT equations.

The first image, on which the original point has been identified, is known as the reference image. The second image, on which a match is being sought, is referred to as the search image.

It is assumed that each image can be represented as a function of horizontal and vertical position, relative to a fixed origin, in the image. Define a function $f(x_1, y_1)$ as the function representing the reference image, the value of which is the grey value of the image at a pixel in column x_1 and row y_1. Similarly, define a function $g(x_2, y_2)$ representing the search image. The value of (x_1, y_1) and that of (x_2, y_2) are restricted to positions within the search and reference patches respectively.

It is assumed that:

$$f(x_1, y_1) \approx g(x_2, y_2) \tag{4.3}$$

Linearising equation 4.3 in order to represent $g(x_2, y_2)$ as a function of x_1 and y_1 it becomes:

$$f(x_1, y_1) \approx g(x_1, y_1) + \delta g \tag{4.4}$$

where

$$\delta g = \frac{\partial g(x_1, y_1)}{\partial x_1}\delta x_1 + \frac{\partial g(x_1, y_1)}{\partial y_1}\delta y_1 \tag{4.5}$$

and

$$\delta x_1 = \delta x_t + x_1\delta a_1 + y_1\delta a_2 \tag{4.6}$$

$$\delta y_1 = \delta y_t + x_1\delta b_1 + y_1\delta b_2 \tag{4.7}$$

Equation 4.4 is linear in parameters δx_t, δy_t, δa_1, δa_2, δb_1 and δb_2, and x_t, y_t, a_1, a_2, b_1 and b_2 can therefore be found using an iterative least squares approach, provided good initial estimates of the parameters are available. The values of x_t and y_t are initialised to 0, while the following approximations can be used for the remaining quantities:

$$a_1 = b_2 = 1$$
$$a_2 = b_1 = 0$$

The values for affine transformation parameters which minimise the sum of the squares of the error between $f(x_1, y_1)$ and $g(x_2, y_2)$ for each pixel in the two image patches are now determined. The values of x_t and y_t are then used as offsets to correct the coordinates of the approximate object position (m, n) in the second image.

Other parameters can be added to equation 4.3 to account for radiometric differences between the similar patches in the images. The radiometric shift can also be corrected for before applying the matching algorithm.

It should be noted that this algorithm is not guaranteed to converge. A number of reasons for possible non-convergence exist. The most common of these are the following:

1. In some situations, the assumption that a small patch on one image is an affinely transformed copy of a small patch on another is not valid. This is a result of the fact that the affine transformation is only an approximation to the projective transformation.

2. The object identified in the reference image is sometimes occluded in the search image, making matching impossible.

3. If the initial estimates of the six unknown parameters are not sufficiently close to their actual values, the linearisation of the affine transformation as described by equations 4.6 and 4.7 will be inaccurate, and the algorithm will not converge to the correct solution.

4. In certain situations, a number of similar patches will surround the correct target. An example of this is occurs where there is a line on an image. In most cases, nothing distinguishes one part of the line from any other part of the line. The algorithm will thus continue to search along the line, never converging to a correct answer, since all sections of the line appear to yield equally valid matches. A possible solution to this problem is described below.

5. If a similar, but distinct image patch is close to the selected point, it is possible that the algorithm will converge to the wrong image patch, and yield an apparently successful match. It is sometimes possible to detect incorrect matches by comparing the actual and expected value of the sum of the squares of the residuals between the patches.

A number of solutions to the above problems have been found. These methods also have the potential to improve the accuracy of the matches. One of the most common methods is the multi-photo geometrically constrained matching algorithm, an algorithm presented by Gruen & Baltsavias (1988). This algorithm minimises the differences between affinely transformed patches of a number of images simultaneously. In addition, the six parameters for each image are limited to values on the epipolar line in the search image. This prevents the algorithm from converging to similar but distinct patches adjacent to the correct object position, since these will, in general, not lie on the epipolar line. The position of the epipolar line, illustrated in figure 2.3, on the search image is determined by the relative orientations of the cameras, and the position of the object in the reference image. The orientations of the cameras are assumed to have been calculated previously. This algorithm can be used to solve simultaneously for the object space coordinates of the point being matched.

Neither this improved algorithm nor any other enhancement to the least squares image matching routine was implemented, since the potential problems were found to occur relatively seldom in this application, and the accuracy was found to be sufficient when using the standard algorithm.

4.2 Circle Location

Given the positions of three points in space, it is possible to find the radius and centre of the circle, and the direction of the normal to the plane of the circle passing through these points. Standard vector geometry is used to calculate these values.

Using the coordinates of the three points, **A**, **B** and **C**, two chords of the circle can be found. These chords are vectors **AB** and **BC**. The directions of the chords are found by subtracting **A** from **B** and **B** from **C** respectively. The cross product of vector **AB** and **BC** defines the direction of the normal to the plane of the circle. Finding the cross product of the directions of the chords will give the same result. The direction of the normal, **D** is therefore:

$$\mathbf{D} = \mathbf{AB} \times \mathbf{BC} \tag{4.8}$$

Figure 4.4 illustrates the points **A**, **B**, and **C**, and the vectors **AB** and **BC**. A vector with the same direction as the normal to the plane of the circle (direction **D**), and passing through the midpoint of the circle (**M**) is also shown.

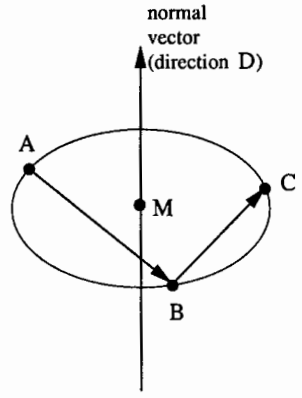

Figure 4.4: Circle with Selected Circumference Points

Two vectors $\mathbf{R_1}$ and $\mathbf{R_2}$ from the midpoints of the the chords to the centre of the circle can now be found. The directions, $\mathbf{Rd_1}$ and $\mathbf{Rd_2}$, of these vectors can be determined from the equations

$$\mathbf{Rd_1} = \mathbf{D} \times \mathbf{AB} \tag{4.9}$$

$$\mathbf{Rd_2} = \mathbf{D} \times \mathbf{BC} \tag{4.10}$$

since $\mathbf{R_1}$ and $\mathbf{R_2}$ must be perpendicular to both the direction of the normal, and the chords. If $\mathbf{R_1}$ and $\mathbf{R_2}$ are each assumed to pass through the centre of one of the chords, it is clear that they will pass through the centre of the circle, in the plane of the circle. The centre points of the chords can be calculated using

$$\mathbf{Rp_1} = \mathbf{A} + \frac{1}{2}\mathbf{AB} \tag{4.11}$$

$$\mathbf{Rp_2} = \mathbf{B} + \frac{1}{2}\mathbf{BC} \tag{4.12}$$

The vectors $\mathbf{R_1}$ and $\mathbf{R_2}$ are illustrated in figure 4.5.

In order to locate the centre of the circle, the intersection of $\mathbf{R_1}$ and $\mathbf{R_2}$ is required. Ideally these vectors will intersect at the centre of the circle. However,

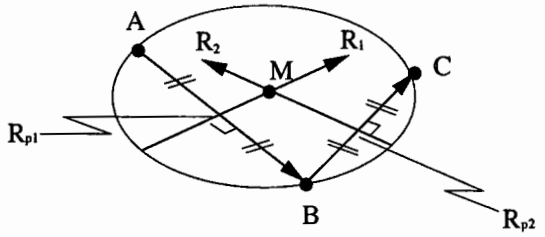

Figure 4.5: Circle with Radius Vectors

as a result of earlier rounding errors, it is probable that the two vectors will approach each other, but will not intersect at the midpoint of the circle. The following equations can be written:

$$s_1\mathbf{Rd_1} + \mathbf{Rp_1} = \mathbf{M} + \epsilon_1 \tag{4.13}$$

$$s_2\mathbf{Rd_2} + \mathbf{Rp_2} = \mathbf{M} + \epsilon_2 \tag{4.14}$$

where $\mathbf{M}$ is the centre of the circle, and s_1 and s_2 are unknown scale factors. The terms ϵ_1 and ϵ_2 describe the vector between $\mathbf{R_1}$ and $\mathbf{R_2}$ vectors, and the centre of the circle at the point of closest approach. The vector describing the difference between $\mathbf{R_1}$ and $\mathbf{R_2}$ at the point of closest approach is given by $\epsilon_1 - \epsilon_2$. Using equations 4.13 and 4.14, the values of s_1 and s_2 can be found. In order to find these values, the distance between $\mathbf{R_1}$ and $\mathbf{R_2}$ at closest approach, $|\epsilon_1 - \epsilon_2|$, must be minimised. Writing equations for each of the vector components in matrix notation gives the equations

$$\begin{bmatrix} (\mathbf{Rd}_x)_1 & -(\mathbf{Rd}_x)_2 \\ (\mathbf{Rd}_y)_1 & -(\mathbf{Rd}_y)_2 \\ (\mathbf{Rd}_z)_1 & -(\mathbf{Rd}_z)_2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} (\mathbf{Rp}_x)_2 - (\mathbf{Rp}_x)_1 \\ (\mathbf{Rp}_y)_2 - (\mathbf{Rp}_y)_1 \\ (\mathbf{Rp}_z)_2 - (\mathbf{Rp}_z)_1 \end{bmatrix} \tag{4.15}$$

Since equation 4.15 is an overdetermined system, the least squares method is used to determine the best values of s_1 and s_2, the two scalars which multiply $\mathbf{Rd_1}$ and $\mathbf{Rd_2}$. The midpoint of the shortest vector between $\mathbf{R_1}$ and $\mathbf{R_2}$ is the best estimate of the centre of the circle, and this point can be calculated using the following equation:

$$\mathbf{M} = \frac{1}{2}[(\mathbf{Rp}_1 + s_1\mathbf{Rd}_1) + (\mathbf{Rp}_2 + s_2\mathbf{Rp}_2)] \tag{4.16}$$

Once the midpoint of the circle has been found, the magnitude of the radius of the circle can be found using one of the following equations

$$\begin{aligned} R &= |\mathbf{M} - \mathbf{A}| \\ R &= |\mathbf{M} - \mathbf{B}| \\ R &= |\mathbf{M} - \mathbf{C}| \end{aligned}$$

The average of these values is generally found to determine the radius of the circle.

The direction of the normal to the plane of the circle is normalised to a unit vector **D**. These three parameters, R, **D**, and **M** then represent the circle passing through **A**, **B**, and **C**.

4.3 Location of a Line in Space

Given two points on a line, the direction of the line is given by $\mathbf{D} = \mathbf{P_1} - \mathbf{P_2}$, where $\mathbf{P_1}$ and $\mathbf{P_2}$ are the given points on the line. The direction is normalised to a unit vector. One of the points and the line direction are then used to represent the line. The distance between the two points is also calculated. This is useful in determining the orientation, position and length of a pipe in an industrial plant.

4.4 Target Centre Location

Reflective targets of approximately 1 cm diameter were used to provide control points in the images. These appear as clusters of high intensity pixels on the images. Since the centres of these targets are used as the control points, a method of finding the centre of each target is required.

A number of methods are described by van der Vlugt (1995) and Rubenstein (1990). The simplest and one of the most accurate and numerically efficient methods available is the use of a weighted centre of gravity to find the centroid of the cluster of pixels. This is done separately to locate the x and y image positions of the centroid as follows:

$$x_{centroid} = \frac{1}{\sum_{i=0}^{N} g(i)} \sum_{i=0}^{N} g(i)^p x_i \tag{4.17}$$

$$y_{centroid} = \frac{1}{\sum_{i=0}^{N} g(i)} \sum_{i=0}^{N} g(i)^p y_i \tag{4.18}$$

where N is the total number of pixels, $g(i)$ is the grey value at the ith pixel, and x_i and y_i are the x and y coordinates of the ith pixel. The variable p is the power to which the grey value is raised.

In order to reduce the effect of surrounding pixels on the calculated value of the centroid, the image is thresholded at a suitable value to remove the pixels which are not part of the target cluster. This assumes that the target area is significantly more reflective than the surrounding areas, which is generally the case. It was found that using the squares of the weights of each pixel

($p = 2$) provided a more reliable estimate of the centroid of the target than the standard centre of gravity method, with $p = 1$. The weighted centre of gravity of the squares of the pixel grey values was used to locate the centre of targets throughout this project.

Another common method of locating the target centroid is the use of grey scale gradients. The gradient of the grey scales at a number of points surrounding the target cluster is calculated. These gradients then define vectors in the plane of the image. By finding the intersection of the gradient vectors, the centroid of the target cluster can be found. This method often provides marginally more accurate results than the centre of gravity method for targets which appear nearly circular in images, but it involves significantly more calculation and was therefore not implemented. Moreover, since targets generally appear as ellipses in images, and the slope vectors of an ellipse do not intersect at one point, this method is not suitable for many targets. Nevertheless, due to the symmetrical distribution of pixels with large grey value gradients around the target images, the method provides reasonable results for target images which are approximately circular.

Another possibility is the fitting of an ellipse to the edge of the target using the least squares method. The edge of the target must be determined to sub-pixel accuracy. This method provides good centroid estimates for well defined targets.

A number of other methods are detailed in (van der Vlugt 1995).

Chapter 5

Algorithm Implementation Issues

This chapter describes some aspects of the implementation of the algorithms which resulted in unexpected difficulties, or where significant improvements are possible. The solutions to these problems are presented, although not all were implemented. The reasons for using or not using these solutions are briefly discussed.

5.1 Matrix Inversion

5.1.1 Numerical Accuracy

In many cases, the matrices used in the photogrammetric algorithms were found to be poorly conditioned, partly because the camera parameter values differ by five or more orders of magnitude. According to (HP 1989) the accuracy of the solution to the matrix equation,

$$\mathbf{A}\mathbf{x} = \mathbf{l} \tag{5.1}$$

where the solution is given by

$$\mathbf{x} = \mathbf{A}^{\mathbf{T}}\mathbf{l} \tag{5.2}$$

where $\mathbf{A}$ is a non-singular matrix, when calculated using finite precision arithmetic, is given by:

$$n_c \geq n_d - \log(|\mathbf{A}||\mathbf{A}^{-1}|) - \log(10n) \tag{5.3}$$

where n is the row and column size of $\mathbf{A}$, n_c is the number of correct digits in each element of the solution vector, n_d is the number of digits in the numerical

representation of the elements of **A**, **x** and **l**. The norm given by $|\mathbf{A}|$ is the standard Euclidean norm.

In order to reduce the errors in the final values of the elements of **x**, an iterative approach was taken to solving for the differentials used in the least squares solutions. A standard iterative refinement method using residuals, as described in (Rice 1981), was used to improve the solution to the equation. More than one iteration using the method of residuals was found to be unnecessary, since the solution converged within one iteration. These iterations are distinct from the iterations involved in the solution of non-linear equations using the iterative least squares approach.

The residual is given by:

$$\mathbf{R} = \mathbf{l} - \mathbf{A}\mathbf{z} \tag{5.4}$$

where **z** is an approximation to **x** calculated using matrix inversion. A correction to **z** can be calculated by solving the equation

$$\mathbf{A}\mathbf{e} = \mathbf{R} \tag{5.5}$$

The calculated value of **e** is added to the value of **z** which becomes a better approximation to **x**.

The speed penalty involved in this calculation is minimal, since most of the time involved in calculating least squares solutions involves matrix multiplications, and matrix inversions. The matrix inversion is performed only once and the inverted matrix stored for further use in other calculations, hence minimising the extra calculation involved.

An alternative approach to reducing numerical errors resulting from the inversion of poorly conditioned matrices involves pre-weighting the equation coefficients. Each row of the matrix **A** is normalised to a standard value by multiplication with a weight before the matrix inversion is performed. The resulting value in vector **x** is then divided by the weight by which the corresponding row of **A** was multiplied.

5.1.2 Calculation Speed

Two important methods of improving the computational efficiency of the calculations involved in the solution of photogrammetric systems are described here. Neither was implemented, since the calculation speed was found to be satisfactory for the application to which the system was being applied, provided a sufficiently fast computer were available. Since present base-line computer systems use Pentium processors, this speed improvement was considered to be of secondary importance relative to other improvements to the system. The improvements may be implemented during further development of the program.

Cholesky Factorisation

If a matrix $\mathbf{B}$ is symmetric and positive definite, it can be factorised into the product of a matrix $\mathbf{Q}$ and its transpose, $\mathbf{Q}^{\mathsf{T}}$, where $\mathbf{Q}$ is a real non-singular lower triangular matrix. The matrix equation

$$\mathbf{Bx} = \mathbf{l} \tag{5.6}$$

can then be written as:

$$\mathbf{QQ^T x} = \mathbf{l} \tag{5.7}$$

According to Jeffrey (1990), solution of equation 5.6 is equivalent to the solution of

$$\mathbf{Qy} = \mathbf{l} \tag{5.8}$$

for $\mathbf{y}$, followed by

$$\mathbf{Q^T x} = \mathbf{y} \tag{5.9}$$

for $\mathbf{x}$. Since in equation 5.8, $\mathbf{Q}$ is lower triangular, and in equation 5.9, $\mathbf{Q^T}$ is upper triangular, these equation can be solved by forward and back substitution respectively.

The Cholesky factorisation can also be used to invert the matrix $\mathbf{B}$, provided $\mathbf{B}$ is positive definite and symmetric. It can be shown that this is the case in many of the photogrammetric algorithms used for this thesis. Rüther (1993) states that, since $\mathbf{B} = \mathbf{QQ^T}$, and using the rules of matrix algebra, the inverse of $\mathbf{B}$ is given by:

$$\mathbf{B^{-1}} = (\mathbf{Q^{-1}})^{\mathbf{T}}\mathbf{Q^{-1}} \tag{5.10}$$

The advantage of this method compared to the use of Gaussian reduction for matrix inversion is that it reduces the number of multiplications required to invert the matrices. The reduction in the number of multiplications reduces the calculation time, and also reduces the accumulated numerical errors in the resulting matrix elements.

A detailed explanation of the derivation of the Cholesky factorisation algorithm and the use thereof is presented by Jeffrey (1990).

Sparse Matrices

In the least squares calculations involved in the bundle adjustment, the matrices involved contain large blocks of zeros. The presence of these zeros lends itself to a number of factorisation and partitioning techniques, such as those used by (van der Vlugt 1995), which can improve the speed of the necessary matrix inversions. A number of these schemes are implemented in the public domain *Sparse* mathematical library, the use of which is explained in (Kundert & Sangiovanni-Vincentelli 1988).

5.2 Polynomial Root Finding

Smith's explicit resection (appendix D) requires that the roots of a 5th degree polynomial be found. An algorithm commonly used in commercial and public domain numerical simulation packages was implemented for this purpose. The method involves finding the complex eigenvalues of the companion matrix of the polynomial and yields all the real and complex roots of the equation. The formulation of companion matrices is described in (Anton 1984), and (Lipshutz 1987) outlines the theory of eigenvalues and eigenvectors required to derive this method.

It was found that, in certain situations, the algorithm incorrectly finds two complex conjugate roots rather than two nearby real roots. This was found to occur only very infrequently, but it is possible that errors of this type might affect the results obtained. However, since the resection using Smith's method is followed by an iterative resection, the magnitude of such inaccuracies is not important. It is possible to use iterative methods such as Newton's method, with the values calculated using the eigenvalue method as initial estimates, to improve the estimates of the roots of the equation.

5.3 Limitations of the Least Squares Algorithm

According to Weisberg (1985), the least squares method does not depend on any assumption concerning the residuals between the observations and the model. It is, however, assumed that the model used is appropriate to the observations, and an outlier in the observation set can, in fact, have a significant effect on the least squares estimates of the model parameters.

Other methods are available to overcome this problem. A method based on robust M-estimates is described in (Haralick & Shapiro 1993) and (Weisberg 1985). The method minimises a function of the residuals other than the sum of their squares, and it can be tuned to minimise the effects of perturbations experienced in a particular application.

A method for fitting a model to experimental data, known as *Random Sample Consensus*, is presented by Fischler & Bolles (1981). This is a method of smoothing data containing a significant percentage of gross errors. It is particularly suited to image analysis based on data provided by error-prone automatic feature detectors.

None of these alternative algorithms was implemented, since the accuracy of the results obtained suggests that the use of the least squares method is justified.

Chapter 6

System Design

This chapter presents an overview of some of the hardware and software design aspects of the system.

6.1 Computer Hardware

While the system was designed to use standard computer hardware, it should be noted that software adaptations as mentioned in later sections could make it possible to use other hardware.

The software system developed for this project is designed to run on a computer with an Intel 486 processor and a floating point unit (maths coprocessor).

At least 2 megabytes of hard disk space is required in addition to the storage requirements for the images being used. The software requires at least 4 megabytes of RAM, but more RAM will improve the speed of operation of the system.

A high resolution video card and complementary monitor are recommended, with a minimum resolution of 1024 by 768 pixels, and a colour depth of 8 bits or 256 colours.

6.2 Image Acquisition Hardware

This project did not include development of techniques or hardware to acquire the images. Images can be acquired by any of a number of techniques mentioned below:

1. The use of a CCD camera together with a commercial frame-grabber card. The frame-grabber card can be used to acquire images from the CCD camera, and to store the images in a common image file format.
2. The use of a (digital) video stills camera. A number of low resolution and high resolution cameras are available, which store captured images directly on a portable hard disk, from where they can later be downloaded to a PC. This method has the advantage of making image acquisition possible in situations where installing a PC is not feasible.
3. Standard photographs can be scanned using a flat bed scanner. This provides very high resolution digital images, but has the disadvantage that distortion is introduced during the developing and the scanning of the photographs.

6.3 Software

This section describes the software used by the system, and the compiler used to develop the programs.

6.3.1 Operating System

The system was implemented as an MS-DOS program, using version 5.0 or greater of MS-DOS. At present, it is not possible to run the software under Windows 95, Windows 3.1 or Windows NT.

6.3.2 Compiler

The software was compiled using DJGPP version 1, an implementation of the GNU C/C++ compiler, version 2.6.3 for MS-DOS. DJGPP was used since it offers a number of advanced features which overcome many of the traditional limitations of MS-DOS. These include:

- allowing access to more than 640k of memory, and blocks of memory larger than 64k;
- 32 bit processing of integer data;
- aggressive optimisation of code, and
- improved portability of the code.

DJGPP includes a DOS extender, which allows programs to access directly up to 128 megabytes of main memory. The DOS extender will swap up to

128 megabytes of data to disk to simulate RAM, although this results in a considerable decrease in the speed of operation of the system.

In addition to the above advantages, GCC is available free from a number of sites around the world, and for a number of different hardware and software platforms.

6.3.3 Portability

Although the software will run only under DOS at present, an effort was made to ensure future portability of the software, allowing it to operate on other platforms, and on computers using different processors.

Using a newer version of DJGPP, it is possible to compile the software to run under DOS, and in a DOS window on all Intel versions of Microsoft operating systems. This should be possible with minimal changes to the code.

If necessary, changes can be made to the code to allow recompiling and running the program as a native Windows or X Windows applications on a Windows or Unix platform respectively. The code was designed to minimise the effort required for such a change, but these changes were not implemented.

6.3.4 Coding

The software was developed as a number of C++ classes to handle images, points, point lists, matrices and menus. Stand-alone functions were written to use these classes. This automated much of the required memory management, and simplified the handling of data in the system. Matrix classes were particularly useful since overloaded functions for addition, division, multiplication and subtraction simplify the writing of the matrix manipulations which are regularly required in the photogrammetric algorithms.

6.3.5 Software Generated Errors

In describing the system, it is assumed that the software operates correctly. While the system was rigorously tested during development, the possibility of coding errors remains. It is, however, also possible that poor compiler optimisation methods can cause numerical inaccuracies and system failure. A variety of compiler errors are known and described in (GNU 1994), and attempts were made to avoid invoking these during the development phase.

6.4 Program Operation

This section provides an overview of the operation of the software, and of the facilities offered to the user of the software.

6.4.1 Image File Loading

The system prompts the user for a project name, project number, and the number of images to be used for a particular session. These inputs are used to determine the file names, and the files are then loaded. The file names are stored to expedite possible reloading of the images in future sessions.

At present only raw image files with 256 grey scale colours, and 1 byte per pixel are supported. A utility is provided to convert from standard 256 colour, uncompressed Microsoft Windows Bitmap format files to raw data files.

6.4.2 Camera Calibration

The system offers three options for camera calibration. These are:

1. Resection using the Smith method, followed by the Schmid's iterative method;
2. Calibration using the Direct Linear Transformation, and
3. Calibration using the DLT followed by the collinearity equations. This calibration is performed on each image, and not as a combined adjustment.

The resection requires that the interior orientation of the camera be known, and be supplied to the program in a file. An estimate of the precision of the orientation parameters is calculated when the collinearity equations are used to calibrate the cameras. The calibration using the collinearity equations is also the only calibration method which offers the possibility of using lens distortion parameters to improve the estimates of the camera orientation parameters, and hence to refine the accuracy of the points found subsequently by object point intersection.

The user is required to identify interactively the positions of the control points on the images. A weighted centre of gravity routine can be used in cases where control points are marked with circular targets. This improves the accuracy with which the control points can be identified by the system operator.

The software also provides the means for the operator to zoom in on an image region, and to identify the position of a point to sub-pixel accuracy manually.

The camera calibration algorithms are described in section 2.1.

6.4.3 Object Point Intersection

A number of routines are provided for the location and viewing of object points.

Routines for finding the object space coordinates of points prompt the operator to identify the point in each of the images. If no point is identified in a certain image, the object space coordinates of the point will be calculated on the basis of the coordinates identified on the remaining images. The point's positions in at least two images are required to find its position in object space.

Once the point has been identified on an image, the epipolar line is drawn on subsequent images to simplify the location of the point. This is of particular value where the control point field is very dense. Once a point has been identified by the operator, an attempt is made to find a match between a block of pixels around the selected point and a block of pixels around the point identified in the first image. The selected point is used as an initial estimate for an iterative least squares match, described in section 4.1. This improves the correspondence between the points identified.

Once all of the image coordinates for a particular point have been identified, the intersection is performed using the collinearity equations. An estimate of the *relative* precision of the object space coordinates is also calculated. The lens distortion parameters are used to improve the estimate if these were calculated during the camera calibration.

The object point intersection algorithms are described in section 2.2.

6.4.4 Image Processing Routines

The program provides routines which perform a number of image processing functions on the images. These are:

1. Histogram Stretching
2. Histogram Equalisation
3. Low Pass Filtering
4. Sharpening
5. Anti-sharp Masking

These algorithms and their effects are described in chapter 3. An option is provided to allow viewing of the images.

6.4.5 Bundle Adjustments

An option is provided to allow the user to run a bundle adjustment on the calibrated cameras, and the intersected points. This finds the solution for all of the object space coordinates, and for all of the camera orientation parameters so as to minimise the residuals in the collinearity equations in the least squares sense. In addition, this option calculates estimates of the precision associated with each of the parameters and object space coordinates. Since the bundle adjustment is computationally intensive, it is intended that this option be used only once for a specific project as a final step before any use is made of the object space coordinates calculated, and before exiting the program.

The bundle adjustment is described in section 2.3.

6.4.6 Other Routines

A number of other utilities are provided. These are described briefly below. Their implementation is described in detail in chapter 4.

Circle Location

The user is prompted to identify three points, the object space coordinates of which have already been calculated. These points are assumed to lie on a circle, and the radius, centre, and direction of the circle are calculated using the algorithm described in section 4.2. It is possible to output these results to a file.

Line Location

The user is prompted to identify two points, the object space coordinates of which have already been calculated. The direction, and the length of the line are then calculated using the method described in section 4.3. The results of this calculation, and a position of a point on the line, are displayed on screen, and can be written to a file if required.

Chapter 7

System Testing

The tests that were conducted to verify the accuracy and correctness of the algorithms used in this thesis are described in this chapter. Four types of tests were performed:

- Tests aimed at ensuring that all of the photogrammetric algorithms implemented work as desired;
- Tests of the ability of the system to measure objects in three dimensions accurately;
- Testing of the circle location algorithm;
- Testing of the line determination routine.

7.1 Algorithm Testing

The DLT and the collinearity equations were employed to calibrate cameras used for generating three images of a rigid metal frame on which control points were marked using small, circular, retro-reflective, white targets. The frame is shown in figure 7.1. A set of images was obtained using different cameras for each image, which therefore gave a different set of interior orientation parameters for each camera.

Another set of images of the interior of an industrial plant were used to verify the correctness of the resection algorithm implementations. Figure 7.2 is one of the images used for this test. Each control point is marked using a black cross on a white background. The control point label is written on the background. A control point labelled '43' is shown in figure 7.3. A bundle adjustment was performed on the same set of images for the sake of comparison. The circle location algorithm was also tested on this set.

Figure 7.1: Control Frame Used for Algorithm Testing

It was necessary to use a different set of images to test the resection algorithms since they require prior knowledge of the interior orientation of the cameras, and these parameters are not accurately known for the cameras used to produce the control frame images. The interior orientation of the camera used to obtain the images of the industrial plant was known, and was held fixed during the photography of the set of images.

Another image set, an example of which is shown in figure 7.4, was used as a further demonstration of the expected accuracy and the reliability of the bundle adjustment. The line location algorithm was also demonstrated using this set of images.

7.1.1 Camera Calibration using the Control Frame

This subsection describes the results of tests using the control frame shown in figure 7.1 to verify the correctness of some of the various camera calibration algorithms, and of the point intersection algorithm.

Camera Calibration using the Bundle Adjustment

The bundle adjustment was performed for a number of points and three cameras, with the control points held fixed.

In addition, the control point image coordinates and the camera orientations were used to calculate the control point positions in object space. Ideally, the

Figure 7.4: Industrial Plant Scene 2

sub-millimeter accuracy in most cases. It is also clear from this test that the presence of additional parameters in the adjustment improves the accuracy of the model, and of results obtained from the model.

Points	A.P.	RMS X dev. (mm)	RMS Y dev. (mm)	RMS Z dev. (mm)
Control points	present	0.23	0.26	0.52
Other points	present	0.28	0.28	0.58
Control points	not present	0.42	0.62	0.81
Other points	not present	0.74	0.41	1.35

Table 7.1: Results from Camera Calibration Using the Bundle Adjustment

Complete results for these tests with and without additional parameters are listed in table F.1 and table F.2 respectively.

Camera Calibration using the Collinearity Equations

The test used to verify the bundle adjustment was repeated for the camera calibration using the collinearity equations. Intersections were also performed using the collinearity equations. The results of the tests are presented in table 7.2. The test was first performed using seven additional parameters, and then repeated without additional parameters.

The results indicate a sub-millimeter accuracy in the X and Y directions, but not always in the Z direction. The poorer accuracy in the Z direction is a result of the positioning of the camera when the images were captured. A greater separation of the cameras in the X or Y directions would improve the accuracy in the Z direction.

Figure 7.2: Industrial Plant Scene 1

Figure 7.3: A Control Point in an Industrial Plant

calculated control point coordinates should coincide with the known control point positions, but because of modelling errors, and errors in the input data, there were small differences between the two sets of coordinates. Table 7.1 shows the RMS values of the deviations in the X, Y and Z directions.

Two other points were intersected to verify the operation of the algorithm on points which did not coincide with control points. These two points were included in the bundle adjustment. It should be noted that in certain cases a large deviation in one of the coordinates of these points resulted in a large RMS deviation being recorded since only two points were involved in the calculation. The control point deviations, calculated on the basis of 25 points in this case, provide a more realistic indication of the accuracy of the algorithm.

The test was first performed using seven additional parameters to model lens distortions, and then repeated without these parameters. The results show

The presence of additional parameters in this case shows definite evidence of improving the accuracy of the solution. An increase in the number of control points used would improve the accuracy to which the additional parameters are calculated, and thus the accuracy with which other points can be found.

Points	A.P.	RMS X dev. (mm)	RMS Y dev. (mm)	RMS Z dev. (mm)
Control points	present	0.22	0.26	0.52
Other points	present	0.29	0.23	0.03
Control points	not present	0.47	0.70	0.97
Other points	not present	0.70	0.42	1.41

Table 7.2: Results from Camera Calibration Using the Collinearity Equations

Complete results for these tests with and without additional parameters are listed in table F.3 and table F.4 respectively.

Camera Calibration using the Direct Linear Transformation

The test used to verify the correctness of the calibration using the collinearity equations was also used to test and verify the accuracy of the DLT. No use was made of additional parameters, and the intersections were performed using the collinearity equations. The results of the test are presented in table 7.3.

The test yielded similar accuracy to that provided by the calibration using the collinearity equations. The difference in accuracy is a result of the conversion of DLT parameters to standard camera orientation parameters.

Points	A.P.	RMS X dev. (mm)	RMS Y dev. (mm)	RMS Z dev. (mm)
Control points	not present	0.69	0.63	1.04
Other points	not present	0.17	0.23	0.83

Table 7.3: Results from Camera Calibration Using the DLT

Complete results for the test without additional parameters are listed in table F.5.

7.1.2 Camera Calibration using Industrial Plant Images

This subsection describes the results of tests to verify the correctness of the camera calibrations using Smith's and Schmid's resections, and using the bundle adjustment. The tests were performed using images of an industrial plant that are similar to the image shown in figure 7.2.

Camera Calibration using Smith's and Schmid's Resections

In order to verify the correct functioning of the resection using Smith's method, followed by Schmid's iterative resection, the set of images of the plant shown in figure 7.2 was used. The control points in this image are marked by black crosses on white target squares.

The camera orientations obtained by these methods were used to determine the control point locations, using the collinearity equations to intersect the points. Additional parameters were not used. The procedure was repeated using the bundle adjustment to provide a reference accuracy with which to compare the results. The bundle adjustment was performed without additional parameters to model lens distortions, since too few control points were available to determine these parameters.

The results of the tests are shown in table 7.4. The accuracy of the two methods is similar, although the bundle adjustment provides better estimates of the X and Y positions of the control points. It should be noted that the interior orientation used in the resections was determined separately using a control frame, and is thus likely to be better than the interior orientation calculated using the bundle adjustment, since more control points were used on the control frame.

Calibration Method	RMS X dev. (mm)	RMS Y dev. (mm)	RMS Z dev. (mm)
Bundle Adjustment	2.01	1.27	7.78
Resections	2.49	2.62	7.43

Table 7.4: Results from Camera Calibration Using Smith's Resection Followed by Schmid's Resection

Complete results for the tests using Smith's and Schmid's resections, and using the bundle adjustment are listed in table F.6.

7.1.3 Bundle Adjustment for Second Factory Scene

An additional test was done to verify the correctness of the bundle adjustment algorithm.

The bundle adjustment was used to calibrate the three cameras which captured a set of images similar to figure 7.4, and to find the location of three points. The position of the control points was also determined as part of the bundle adjustment, and compared to the known coordinates of these points. The results of this test are presented in table 7.5. Additional parameters were not used for this test, because too few control points were available.

Points	RMS X dev. (mm)	RMS Y dev. (mm)	RMS Z dev. (mm)
Control points	19.9	18.2	58.3

Table 7.5: Results from Camera Calibration Using the Bundle Adjustment for Industrial Plant Scene 2

The complete results are presented in table F.9. The results for this test are less accurate than those obtained using the bundle adjustment on the first industrial scene. The reduced accuracy is a result of the larger distances between the control points and the cameras. This is particularly evident in the errors in the most distant control points, point 48 and point 49 (shown in table F.9).

7.2 Photogrammetric Measurement Testing

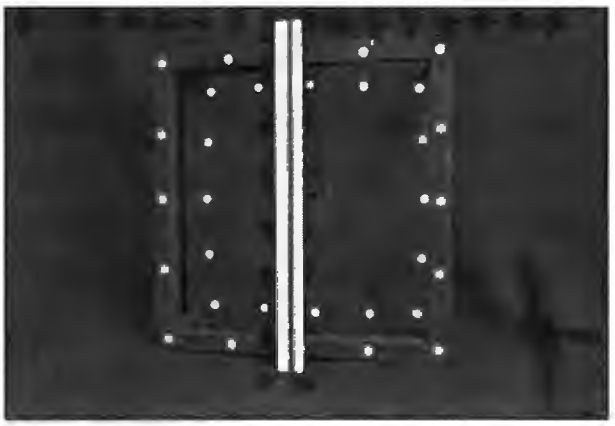

Figure 7.5: Image Used for Photogrammetric Measurements

In order to test the ability of the system to measure distances in space, a set of three images of the scene shown in figure 7.5 were used. A small control frame was used to provide control points for calibration of the cameras. The relative orientations of these control points had previously been calculated for use in other projects.

A standard 30 cm ruler was used to test the accuracy of the distances between points calculated using the system.

The RMS deviations of the intersected control points from the known coordinates of the control points are given in table 7.6. The larger deviation in the Z direction is, as in all examples presented in this chapter, a result of the positioning of the cameras. All these deviations are smaller than those for the tests involving the calibration frame shown in figure 7.1 as a result of the higher

RMS X dev. (mm)	RMS Y dev. (mm)	RMS Z dev. (mm)
0.05	0.06	0.10

Table 7.6: Results from Measurement Test Using the Bundle Adjustment with Additional Parameters

resolution images, and the smaller distance between the frame and the cameras used for the measurement test. Table F.8 in the appendices lists all the results, including the calculated standard deviations for each of the calculated values.

Five points on the ruler were identified, and their coordinates were calculated. The distances between these points were then also calculated, and compared to their known values. The calculated distances and the actual distances measured are presented in table 7.7. The distances calculated are all within $\frac{1}{2}$mm of the true values. There are four principle sources of error:

1. Accuracy of control point image coordinates: Control points which are poorly identified in images result in inaccuracy in the camera orientation parameters, and in the object point coordinates calculated using these parameters.

2. Poor lens distortion model: The lens distortion model used is necessarily an approximation of the true distortion model, and will thus introduce small errors.

3. Accuracy of object point identification in images: The lines marking millimeters on the ruler occupy approximately 2 or 3 pixels on the images, and manual identification of the centre of the lines, and the endpoints thereof is difficult. This is probably the most significant cause of errors in the test.

4. Inaccurate control point coordinates: The accuracy of the control point coordinates is unknown, and it is assumed that the given coordinates are absolutely accurate, which is clearly not true.

Point 1 (cm marking)	Point 2 (cm marking)	measured distance (mm)	error %
0	10	99.95	-0.05
0	15	149.96	-0.02
0	20	200.06	0.03
0	30	300.11	0.04

Table 7.7: Distances Calculated for the Measurement Test

These measurements illustrate the ability of the system to measure distances with an accuracy better than 0.05% of the total distance, when the object being measured occupies approximately 80% of the height of the image.

7.3 Circle Location

The first set of industrial plant images (figure 7.2) were used for this test, which involved calculating the radius, orientation, and centre of a circle passing through three points on the edge of a large drum.

The test was repeated using a second set of three points on the same drum edge, and the radius, orientation, and centre of the circle calculated using these points was compared to those calculated using the first three points. The two sets of three points are illustrated in figure 7.6.

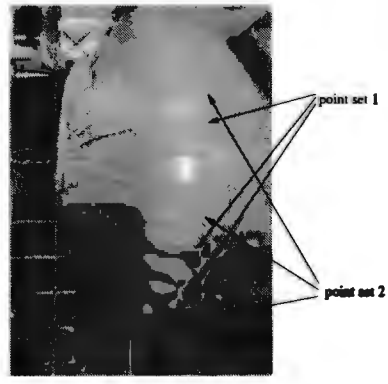

Figure 7.6: Point Sets for Circle Location Tests

The results of these two tests are presented in table 7.8.

circle	parameter	value
1	midpoint (X, Y, Z)(mm)	(208151, 16500, -76539)
	normal direction (X, Y, Z)	(-0.9968, -0.0238, -0.0767)
	radius (mm)	1100
2	midpoint (mm)	(208207, 16479, -76529)
	normal direction	(-0.9997, -0.0086, -0.0242)
	radius (mm)	1104

Table 7.8: Results from Circle Location Algorithm

The two circles identified are expected to be identical, since both pass though points on the rim of the tank as illustrated in figure 7.6. The radii differ by 4 mm, and the angle between the two normal directions is 3.1°. The distance between the two calculated centre points is 60.4 mm.

A number of factors cause errors in the calculated circles. The most important of these is the precision of the calculated positions of points in object space, which is of the order of 3 cm for this test. The effect of the errors in the object space coordinates is larger if all three points used to locate the circle are located on the same side of the pipe, as is the case for this test. In addition, it should be noted that the object to which a circle is being fitted is not necessarily perfectly circular. Image point identification by the operator is also imperfect as a result of the difficulty involved in determining the exact position of features on the images.

7.4 Line Location

The second set of industrial plant images (figure 7.4) were used for this test. To demonstrate the line location procedure, the ends of two approximately perpendicular pipes were identified. These lines are illustrated in figure 7.7.

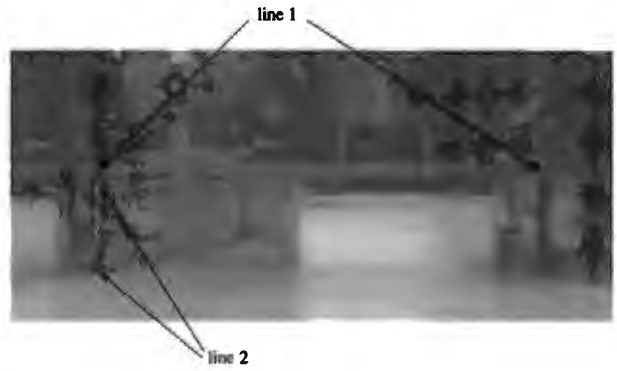

Figure 7.7: Lines Located in Industrial Plant Scene 2

The results of this test are presented in table 7.9. As a result of rounding errors, the length of the direction vector is not exactly 1 mm.

line	parameter	value
1	base point (X, Y, Z)(mm)	(-83777, 14346, -205122)
	direction (X, Y, Z)	(1.0000, -0.0011, 0.0019)
	length (mm)	2463.8
2	base point (mm)	(-83784, 13965, -205048)
	direction	(0.0145, 0.9993, -0.0348)
	length (mm)	325.4

Table 7.9: Results from Line Determination Procedure

The angle between line 1 and the horizontal is less than 0.2°, and between line 2 and the vertical is less than 2°. These deviations result from a combination

of real deviations in the layout of the plant, and the limited accuracy of the calculated object point positions. The angle between the two lines is 89.23°, indicating that the lines are approximately perpendicular, as expected.

Chapter 8

Conclusions

The aim of this thesis was to develop a system capable of expediting the process of mapping the interior of industrial plants by implementing a number of photogrammetric and image processing algorithms in an integrated software system, and through the use of digital images.

The system designed is successful in achieving these aims. The ability of the system to measure accurately in three dimensions the positions of points on digital images has been demonstrated. The system improves on analogue photogrammetric techniques in a number of ways:

- It decreases the time taken from image capture to object point location. By eliminating the need to capture, develop, and enlarge standard photographs, a significant amount of time is saved. In addition, the use of digital images has meant that some of the distortion typically introduced during the numerous processing stages involved in analogue photography has been avoided.
- Transfer of object point locations from the photogrammetric system to the CAD model is eased since point coordinates are stored digitally, and are marked on images. Simple alterations will make it possible to produce output in any desired CAD format.
- The number of images used, which to some extent determines the accuracy of the object points located, is limited only by available memory. Most current systems are restricted to the use of two images.
- Project management is simplified considerably by the ability to store complete projects in digital format on a computer, rather than storing photographs as is currently done. Moreover, the transmission of projects between sites can be achieved at significantly higher speed and lower cost using computer networks or modem links.

Among the disadvantages of the system compared to current techniques are:

- The use of digital images in place of standard photographs reduces the resolution to which points on images can be determined, since the resolution of high quality photographs is generally significantly higher than that of images produced by digital cameras. As computer displays improve, and as digital camera technology advances, this disadvantage will become less significant.
- The requirements of the software, in terms of computer data storage capacity, RAM, and video resolution, are significant compared to current systems which require only text mode displays and minimal storage capacity. However, this is not a major disadvantage, since computers which are currently considered entry-level systems are sufficiently powerful to run the new software.

Apart from those improvements mentioned in chapter 5 which were not implemented, a number of other possible enhancements which would be of particular value are:

- Improved tolerance to system- and user-generated errors.
- Portability of the system. This would be of particular use in allowing the system to be run side-by-side with the CAD package on an advanced operating system to allow the easy transfer of data between packages.
- The integration of other related photogrammetric software designed by postgraduate students of the Department of Surveying and Geodetic Engineering at the University of Cape Town.
- Improved image processing capabilities. Existing algorithms to automate the location of targets in images should be included, and the possibility of implementing algorithms to identify and follow pipes in images should be considered.
- Improved circle, ellipse, line and square location algorithms to facilitate the identification of characteristic features of pipes and other objects commonly found in the images.

Bibliography

Anton, H. (1984), *Elementary Linear Algebra*, fourth edn, John Wiley and Sons, New York, p. 293.

Brown, D. C. (1966), 'Decentering distortion of lenses', *Photogrammetric Engineering* pp. 444–462.

Chapman, D., Deacon, A. & Hamid, A. (1992), 'CAD modelling of radioactive plant: The role of digital photogrammetry in hazardous nuclear environments', *International Archives of Photogrammetry and Remote Sensing, Part B5* **XXIX**, 741–753.

de Jager, G. (1995), Course Notes: Multidimensional Digital Signal and Image Processing, EEE591S.

Dermanis, A. (1994), 'Free network solutions with the DLT method', *ISPRS Journal of Photogrammetry and Remote Sensing* **49**(2), 2–12.

Fischler, M. A. & Bolles, R. C. (1981), 'Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography', *Communications of the Association for Computing Machinery* **24**(6), 381–395.

Fryer, J. G. & Brown, D. C. (1986), 'Lens distortion for close range photogrammetry', *Photogrammetric Engineering and Remote Sensing* **52**(1), 51–58.

GNU (1994), 'GNU C/C++ version 2.6.3 documentation'.

Granshaw, S. I. (1980), 'Bundle adjustment methods in engineering photogrammetry', *Photogrammetric Record* **56**(10), 181–207.

Gruen, A. (1978), 'Accuracy, reliability and statistics in close-range photogrammetry'. Presented Paper, Inter-Congress Symposium, Commission V.

Gruen, A. W. (1985), 'Adaptive least squares correlation - a powerful image matching technique'. Presented Paper to ACSM-ASP Convention, Washington D.C.

Gruen, A. W. & Baltsavias, E. P. (1988), 'Geometrically constrained multi-photo matching', *Photogrammetric Engineering and Remote Sensing* **54**(5), 633–641.

Hamit, F. (1995), 'As-built industrial visualization CAD meets digital photo and document', *Advanced Imaging* pp. 75–77.

Haralick, R. M. & Shapiro, L. G. (1993), *Computer and Robot Vision*, Vol. II, Addison-Wesley Publishing Company, Reading, Massachusetts.

Hirvonen, R. A. (1971), *Adjustment by Least Squares in Geodesy and Photogrammetry*, Frederick Ungar Publishing Co.

HP (1989), *Advanced Scientific Calculator, Reference Manual, HP28S*, 5 edn.

Jähne, B. (1991), *Digital Image Processing, Concepts, Algorithms and Scientific Applications*, second edn, Springer-Verlag.

Jeffrey, A. (1990), *Linear Algebra and Ordinary Differential Equations*, Blackwell Scientific Publications Inc, Massachusetts.

Jones, M. A., Chapman, D. P. & Hamid, A. A. (1996), 'Close range photogrammetry using geometric primitives for efficient cad modelling of industrial plant', *International Archives of Photogrammetry and Remote Sensing, Part B5* **XXXI**, 284–289.

Karara, H., ed. (1989), *Non-Topographic Photogrammetry*, American Society for Photogrammetry and Remote Sensing, pp. 37–57.

Krämer, J. & Schöler, H. (1980), 'Photogrammetric measurement of piping systems'. Presented Paper, 14th International Congress of ISP, Commission V, Working Group, Hamburg.

Kraus, K. & Waldhäusl, P. (1993), *Photogrammetry, Fundamentals and Standard Processes*, Vol. 1, Ferd. Dümmlers Verlag, Bonn.

Kundert, K. S. & Sangiovanni-Vincentelli, A. (1988), *Sparse User's Guide, A Sparse Linear Equation Solver.*

Lim, J. S. (1984), *Image Enhancement: Digital Image Processing Techniques*, Academic Press, Florida.

Lipshutz, S. (1987), *Theory and Problems of Linear Algebra*, McGraw Hill, New York, chapter 9.

Littleworth, R. M. & Chandler, J. H. (1995), 'Three dimensional computer graphics models by analytical photogrammetry', *Photogrammetric Record* **85**(15), 65–76.

Littleworth, R. M., Stirling, D. M. & Chandler, J. H. (1992), 'Three-dimensional mapping and as-built computer modelling by analytical photogrammetry', *International Archives of Photogrammetry and Remote Sensing, Part B5* **XXIX**, 754–760.

MCP (1986), *Micro Computer-Aided Photogrammetry System.*

Milton, J. S. & Arnold, J. C. (1986), *Probability and Statistics in the Engineering and Computing Sciences*, McGraw-Hill Book Company, New York.

Monro, D. M. & Dudbridge, F. (1995), 'Rendering algorithms for deterministic fractals', *IEEE Computer Graphics and Applications* pp. 32–40.

Nicholson, W. K. (1990), *Elementary Linear Algebra With Applications*, second edn, PWS-KENT Publishing Company, Boston.

Rice, J. R. (1981), *Matrix Computations and Mathematical Software*, McGraw-Hill Book Company, New York.

Rubenstein, M. (1990), Assessing target centring algorithms for use in near real time photogrammetry, Master's thesis, University of Cape Town.

Rüther, H. (1993), Course Notes - The Theory of the Adjustment of Observations : SUR321F.

Rüther, H. (1994), Course Notes - Advanced Engineering Surveying : SUR409F.

Smith, A. D. N. (1965), 'The explicit solution of the single picture resection problem, with a least squares adjustment to redundant control', *Photogrammetric Record* **5**(26), 113–121.

Thompson, M. M., ed. (1966), *Manual of Photogrammetry*, Vol. 1, third edn, American Society of Photogrammetry, pp. 50–53.

van der Vlugt, G. (1995), Algorithms and design aspects of an Automated Vision Based 3-D Surface Measurement System, PhD thesis, University of Cape Town.

Weisberg, S. (1985), *Applied Linear Regression*, John Wiley and Sons, New York.

Zill, D. G. & Cullen, M. R. (1992), *Advanced Engineering Mathematics*, PWS Publishing Company, Boston.

Appendix A

Least Squares Model Fitting

The least squares algorithm for model fitting is a simple method of finding the parameters for a linear model in order to ensure the *best* fit between the model and a number of observations. The *best* parameters are assumed to be those parameters for the model which minimise the sum of the squares of the differences between the values estimated using the model, and the observed values of the quantity. These differences are usually referred to as residuals. The method, described in (Rice 1981) and (Nicholson 1990), can only be used for equations which are linear in the unknown model parameters, represented by the vector $(\beta_1, \beta_2, \ldots, \beta_k)$ where these are coefficients of variables $(x_1, x_2, \ldots, x_k)$, where x_1 may be constant.

A.1 Least Squares Solutions

Let the value of the model be given by l where:

$$l = \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k \tag{A.1}$$

If a number, n, of observations of l and x_i for all $i \in [1, k]$ are available, an equal number of equations in the unknown model parameters can be written in matrix form as:

$$\begin{bmatrix} x_{11} & x_{21} & \ldots & x_{k1} \\ x_{12} & x_{22} & \ldots & x_{k2} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1n} & x_{2n} & \ldots & x_{kn} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_k \end{bmatrix} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_n \end{bmatrix} \tag{A.2}$$

which can be abbreviated as:

$$\mathbf{Ab} = \mathbf{l} \tag{A.3}$$

where $\mathbf{A}$ is a matrix containing values of x_i, $\mathbf{b}$ is a matrix containing values of β_i and $\mathbf{l}$ (not $\mathbf{I}$) is the matrix of values of l_i.

If n is equal to the number of unknown parameters (k) and each of the observation sets is independent, matrix inversion can be used to solve for the values of $(\beta_1, \beta_2, \ldots, \beta_k)$. That is:

$$\mathbf{b} = \mathbf{A}^{-1}\mathbf{l} \tag{A.4}$$

If however, as is frequently true, n is larger than the number of unknown parameters, there is generally no solution for $\mathbf{b}$. A common measure of the *best* values of $(\beta_1, \beta_2, \ldots, \beta_k)$ is the set of values which minimises the sum of the squares of the differences between $\mathbf{l}$ and $(\beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k)$. It can be shown, geometrically (Nicholson 1990), or by the use of calculus (Zill & Cullen 1992), that solving the following equation will find the unknown parameters which satisfy this criterion:

$$\mathbf{b} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{l} \tag{A.5}$$

If the relative accuracies of all the observations are known, then these can be used to influence the result using the equation:

$$\mathbf{b} = (\mathbf{A}^T\mathbf{P}\mathbf{A})^{-1}\mathbf{A}^T\mathbf{P}\mathbf{l} \tag{A.6}$$

where the matrix $\mathbf{P}$ is a diagonal matrix with the weights representing the accuracy of each of the observations on the diagonal.

A.2 Standard Deviation Estimates

The value of the standard deviation of the calculated vector of parameters from the correct values for these parameters is derived in (Milton & Arnold 1986). The standard deviation is calculated as follows: the standard deviation of unit weight is estimated by:

$$\sigma_0 = \sqrt{\frac{\mathbf{v}^T\mathbf{v}}{n-k}} \tag{A.7}$$

where $\mathbf{v}$ is the vector of residuals:

$$\mathbf{v} = \mathbf{A}\mathbf{b} - \mathbf{l} \tag{A.8}$$

An estimate of the standard deviation of the individual parameters is then given by

$$\sigma_{x_m} = \sigma_0\sqrt{((\mathbf{A}^T\mathbf{A})^{-1})_{mm}} \tag{A.9}$$

where $((\mathbf{A}^T\mathbf{A})^{-1})_{mm}$ is the m-th diagonal element of $(\mathbf{A}^T\mathbf{A})^{-1}$.

A.3 Least Squares Methods and Non-Linear Models

The least squares method can be used on non-linear models, provided sufficiently accurate estimates of the parameters are available. A detailed description of the use of this technique is given in (Haralick & Shapiro 1993), and Hirvonen (1971) describes its use in photogrammetric applications.

The model is linearised using a single term of the Taylor expansion of the non-linear function. The standard least squares approach is then used to find the best values of the differentials of the parameters. The differentials are added to the initial estimates of the parameters to yield improved estimates. The estimates of standard deviation derived above apply to parameter differentials, and thus also to the parameters.

A more detailed overview of the use of the least squares approach on non-linear models is given in the descriptions of individual algorithms.

Appendix B

The Collinearity Equations

The collinearity equations, the use of which in photogrammetric applications is described in (Karara 1989), are:

$$x - x_p + dx = c\frac{r_{11}(X - X_c) + r_{12}(Y - Y_c) + r_{13}(Z - Z_c)}{r_{31}(X - X_c) + r_{32}(Y - Y_c) + r_{33}(Z - Z_c)} \tag{B.1}$$

$$y - y_p + dy = c\frac{r_{21}(X - X_c) + r_{22}(Y - Y_c) + r_{23}(Z - Z_c)}{r_{31}(X - X_c) + r_{32}(Y - Y_c) + r_{33}(Z - Z_c)} \tag{B.2}$$

where r_{11}, r_{12}, ... , r_{33} are the elements of the rotation matrix R such that:

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{B.3}$$

This matrix is referred to throughout this chapter as the rotation matrix. The terms dx and dy are terms which describe the distortion in the x and y directions as a result of lens and image surface imperfections. For the derivations which follow, these terms are assumed to be zero. Section B.3 explains the changes needed to incorporate the distortion parameters in the solution.

The angles ω, κ, and ϕ are the rotations around the x, y, and z axes respectively. The rotations are applied to the image space axis system whose axes are initially parallel to their corresponding object space axes. The rotation ϕ, is followed by κ, and then ω. This sequence of rotations is important, and is assumed throughout this chapter. Other rotation sequences are possible, but are not used here.

The elements of the rotation matrix are functions of the three rotation angles, ω, κ, and ϕ. The functions are:

$$r_{11} = \cos\kappa\cos\phi \tag{B.4}$$

$$r_{12} = -\cos\kappa\sin\phi \tag{B.5}$$

$$r_{13} = \sin\kappa \tag{B.6}$$
$$r_{21} = \cos\omega\sin\phi + \sin\omega\sin\kappa\cos\phi \tag{B.7}$$
$$r_{22} = \cos\omega\cos\phi - \sin\omega\sin\kappa\sin\phi \tag{B.8}$$
$$r_{23} = -\sin\omega\cos\kappa \tag{B.9}$$
$$r_{31} = \sin\omega\sin\phi - \cos\omega\sin\kappa\cos\phi \tag{B.10}$$
$$r_{32} = \sin\omega\cos\phi + \cos\omega\sin\kappa\sin\phi \tag{B.11}$$
$$r_{33} = \cos\omega\cos\kappa \tag{B.12}$$

These equations are obtained from the individual rotation matrices which perform rotation around each of the three axes. These three rotation matrices are:

$$\mathbf{R_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix} \tag{B.13}$$

$$\mathbf{R_y} = \begin{bmatrix} \cos\kappa & 0 & \sin\kappa \\ 0 & 1 & 0 \\ -\sin\kappa & 0 & \cos\kappa \end{bmatrix} \tag{B.14}$$

$$\mathbf{R_z} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.15}$$

The matrices $\mathbf{R_x}$, $\mathbf{R_y}$ and $\mathbf{R_z}$ are multiplied in order to obtain matrix $\mathbf{R}$.

Using equations B.1 and B.2 we can define

$$F_x = x - x_p - c\frac{R_1}{R_3} = 0 \tag{B.16}$$

$$F_y = y - y_p - c\frac{R_2}{R_3} = 0 \tag{B.17}$$

where

$$R_1 = r_{11}(X - X_c) + r_{12}(Y - Y_c) + r_{13}(Z - Z_c)$$
$$R_2 = r_{21}(X - X_c) + r_{22}(Y - Y_c) + r_{23}(Z - Z_c)$$
$$R_3 = r_{31}(X - X_c) + r_{32}(Y - Y_c) + r_{33}(Z - Z_c)$$

B.1 Camera Calibration

For the calibration of cameras, X, Y, Z, x and y are known quantities, and x_p, y_p, c, ω, ϕ, κ, X_c, Y_c and Z_c are the nine unknown variables.

Using one term of the Taylor expansion of F_x, it can be shown that:

$$\begin{aligned}\delta F_x \approx\ & F_x(x_p + \delta x_p, y_p + \delta y_p, \ldots, Y_c + \delta Y_c, Z_c + \delta Z_c) \\ & - F_x(x_p, y_p, \ldots, Y_c, Z_c)\end{aligned}$$

where δx_p, δy_p, δY_c, etc are assumed to be small changes in the variables x_p, y_p, Y_c, etc. If this equation is being used iteratively to refine a value of F_x then it can be formulated as:

$$(F_x)_{t+1} = (F_x)_t + (\delta F_x)_t$$

where t is the iteration number. From equations B.16 and B.17, it is clear that the value of F_x and of F_y should ideally be 0. In practice F_x and F_y are small but non-zero, due to system errors when the correct orientation parameters are used in the collinearity equations. Since $(F_x)_{t+1}$ is a better estimate of F_x than $(F_x)_t$, and since it is *desired* that $(F_x)_{t+1} = 0$, an equation

$$(F_x)_t = -(\delta F_x)_t \tag{B.18}$$

can be written for each observation set. A corresponding set of equations can be written for F_y. If more than nine equations exist, it is generally not possible to find a set of parameters which simultaneously solves all of these equations. The values of F_x and F_y will thus generally *never* be zero in practice. The set of parameters which minimises the sum of the squares of the differences between the left and the right hand sides of the equations is then found, and this is considered the *best* solution, in the least squares sense.

According to Taylor's theorem

$$\begin{aligned}\delta F_x \approx \quad & + \frac{\partial F_x}{\partial x_p}\delta x_p + \frac{\partial F_x}{\partial y_p}\delta y_p + \frac{\partial F_x}{\partial c}\delta c \\ & + \frac{\partial F_x}{\partial \omega}\delta\omega + \frac{\partial F_x}{\partial \kappa}\delta\kappa + \frac{\partial F_x}{\partial \phi}\delta\phi \\ & + \frac{\partial F_x}{\partial X_c}\delta X_c + \frac{\partial F_x}{\partial Y_c}\delta Y_c + \frac{\partial F_x}{\partial Z_c}\delta Z_c\end{aligned} \tag{B.19}$$

and

$$\begin{aligned}\delta F_y \approx \quad & + \frac{\partial F_y}{\partial x_p}\delta x_p + \frac{\partial F_y}{\partial y_p}\delta y_p + \frac{\partial F_y}{\partial c}\delta c \\ & + \frac{\partial F_y}{\partial \omega}\delta\omega + \frac{\partial F_y}{\partial \kappa}\delta\kappa + \frac{\partial F_y}{\partial \phi}\delta\phi \\ & + \frac{\partial F_y}{\partial X_c}\delta X_c + \frac{\partial F_y}{\partial Y_c}\delta Y_c + \frac{\partial F_y}{\partial Z_c}\delta Z_c\end{aligned} \tag{B.20}$$

Assuming that n control points are available, with x_i and y_i representing the x and y image coordinates of the ith control point, and X_i, Y_i and Z_i representing the object space coordinates of the ith control point, then equations B.19, B.20 and B.18 can be rewritten as a matrix equation for a particular iteration, with the subscript t having been dropped:

$$\begin{bmatrix} -\frac{\partial F_x}{\partial x_p}_1 & \cdots & -\frac{\partial F_x}{\partial Y_c}_1 & -\frac{\partial F_x}{\partial Z_c}_1 \\ -\frac{\partial F_y}{\partial x_p}_1 & \cdots & -\frac{\partial F_y}{\partial Y_c}_1 & -\frac{\partial F_y}{\partial Z_c}_1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{\partial F_x}{\partial x_p}_i & \cdots & -\frac{\partial F_x}{\partial Y_c}_i & -\frac{\partial F_x}{\partial Z_c}_i \\ -\frac{\partial F_y}{\partial x_p}_i & \cdots & -\frac{\partial F_y}{\partial Y_c}_i & -\frac{\partial F_y}{\partial Z_c}_i \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta x_p \\ \delta y_p \\ \vdots \\ \vdots \\ \delta X_c \\ \delta Y_c \\ \delta Z_c \end{bmatrix} = \begin{bmatrix} (F_x)_1 \\ (F_y)_1 \\ \vdots \\ (F_x)_i \\ (F_y)_i \\ \vdots \end{bmatrix} \tag{B.21}$$

This matrix equation cannot generally be solved when more than the minimum number of control points are used.

Using the least squares method, described in appendix A, the *best* values of δx_p, δy_p, δc, ..., δX_c, δY_c and δZ_c are found. These are used to update the orientation parameters by adding the changes $(\delta x_p, \delta y_p, \delta c, \ldots, \delta X_c, \delta Y_c, \delta Z_c)$ to the present values of $(x_p, y_p, c, \ldots, X_c, Y_c, Z_c)$.

Since in practice $\delta x_p, \delta y_p, \delta c, \ldots, \delta X_c, \delta Y_p$ and δZ_c are not vanishingly small, it is necessary that equation B.21 be applied iteratively, updating the camera parameters after each iteration. The values of the parameters will converge to give the least squared residual between the various values of F_x, F_y and zero. This is the equivalent of minimising the sum of the squares of the differences between the observed control point image coordinates, and the control point image coordinates calculated using the object space control point coordinates together with the camera orientation parameters.

The partial derivatives in equation B.21 are:

$$\frac{\partial F_x}{\partial x_p} = -1 \tag{B.22}$$

$$\frac{\partial F_x}{\partial y_p} = 0 \tag{B.23}$$

$$\frac{\partial F_x}{\partial c} = -\frac{R_1}{R_3} \tag{B.24}$$

$$\frac{\partial F_x}{\partial \omega} = \frac{c\,R_1 R_2}{R_3^2} \tag{B.25}$$

$$\frac{\partial F_x}{\partial \kappa} = -\frac{c}{R_3^2}(a_1 R_3 - a_2 R_1) \tag{B.26}$$

$$\frac{\partial F_x}{\partial \phi} = -\frac{c}{R_3^2}(a_3 R_3 - a_4 R_1) \tag{B.27}$$

$$\frac{\partial F_x}{\partial X_c} = c\left[\frac{r_{11} R_3 - r_{31} R_1}{R_3^2}\right] \tag{B.28}$$

$$\frac{\partial F_x}{\partial Y_c} = c\left[\frac{r_{12} R_3 - r_{32} R_1}{R_3^2}\right] \tag{B.29}$$

$$\frac{\partial F_x}{\partial Z_c} = c\left[\frac{r_{13} R_3 - r_{33} R_1}{R_3^2}\right] \tag{B.30}$$

and

$$\frac{\partial F_y}{\partial x_p} = 0 \tag{B.31}$$

$$\frac{\partial F_y}{\partial y_p} = -1 \tag{B.32}$$

$$\frac{\partial F_y}{\partial c} = -\frac{R_2}{R_3} \tag{B.33}$$

$$\frac{\partial F_y}{\partial \omega} = \frac{c(R_3^2 + R_2^2)}{R_3^2} \tag{B.34}$$

$$\frac{\partial F_y}{\partial \kappa} = -\frac{c}{R_3^2}(a_5 R_3 - a_6 R_2) \quad \text{(B.35)}$$

$$\frac{\partial F_y}{\partial \phi} = -\frac{c}{R_3^2}(a_7 R_3 - a_8 R_2) \quad \text{(B.36)}$$

$$\frac{\partial F_y}{\partial X_c} = c\left[\frac{r_{21} R_3 - r_{31} R_2}{R_3^2}\right] \quad \text{(B.37)}$$

$$\frac{\partial F_y}{\partial Y_c} = c\left[\frac{r_{22} R_3 - r_{32} R_2}{R_3^2}\right] \quad \text{(B.38)}$$

$$\frac{\partial F_y}{\partial Z_c} = c\left[\frac{r_{23} R_3 - r_{33} R_2}{R_3^2}\right] \quad \text{(B.39)}$$

where

$$a_1 = -\sin\kappa\cos\phi(X - X_c) + \sin\kappa\sin\phi(Y - Y_c) + \cos\kappa(Z - Z_c) \quad \text{(B.40)}$$

$$a_2 = -\cos\omega\cos\kappa\cos\phi(X - X_c) + \cos\omega\cos\kappa\sin\phi(Y - Y_c) - \cos\omega\sin\kappa(Z - Z_c) \quad \text{(B.41)}$$

$$a_3 = r_{12}(X - X_c) - r_{11}(Y - Y_c) \quad \text{(B.42)}$$

$$a_4 = r_{32}(X - X_c) - r_{31}(Y - Y_c) \quad \text{(B.43)}$$

$$a_5 = \sin\omega\cos\kappa\cos\phi(X - X_c) - \sin\omega\cos\kappa\sin\phi(Y - Y_c) + \sin\omega\sin\kappa(Z - Z_c) \quad \text{(B.44)}$$

$$a_6 = -\cos\omega\cos\kappa\cos\phi(X - X_c) + \cos\omega\cos\kappa\sin\phi(Y - Y_c) - \cos\omega\sin\kappa(Z - Z_c) \quad \text{(B.45)}$$

$$a_7 = r_{22}(X - X_c) - r_{21}(Y - Y_c) \quad \text{(B.46)}$$

$$a_8 = r_{32}(X - X_c) - r_{31}(Y - Y_c) \quad \text{(B.47)}$$

B.2 Object Point Intersection

In order to find the intersection of two lines defined by points on images from two cameras, the orientations of which are known, the camera orientation parameters are held constant and the object space coordinates of the point are treated as variables.

For the intersection, the small changes δF_x and δF_y are given by:

$$\delta F_x \approx \frac{\partial F_x}{\partial X}\delta X + \frac{\partial F_x}{\partial Y}\delta Y + \frac{\partial F_x}{\partial Z}\delta Z \quad \text{(B.48)}$$

and

$$\delta F_y \approx \frac{\partial F_y}{\partial X}\delta X + \frac{\partial F_y}{\partial Y}\delta Y + \frac{\partial F_y}{\partial Z}\delta Z \quad \text{(B.49)}$$

where the partial derivatives are evaluated using the estimates of the camera parameters.

Since in the *ideal* case, $F_x = -\delta F_x$ and $F_y = -\delta F_y$, the following matrix equation can be written:

$$\begin{bmatrix} -\frac{\partial F_x}{\partial X}_1 & -\frac{\partial F_x}{\partial Y}_1 & -\frac{\partial F_x}{\partial Z}_1 \\ -\frac{\partial F_y}{\partial X}_1 & -\frac{\partial F_y}{\partial Y}_1 & -\frac{\partial F_y}{\partial Z}_1 \\ \vdots & \vdots & \vdots \\ -\frac{\partial F_x}{\partial X}_j & -\frac{\partial F_x}{\partial Y}_j & -\frac{\partial F_x}{\partial Z}_j \\ -\frac{\partial F_y}{\partial X}_j & -\frac{\partial F_y}{\partial Y}_j & -\frac{\partial F_y}{\partial Z}_j \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \end{bmatrix} = \begin{bmatrix} (F_x)_1 \\ (F_y)_1 \\ \vdots \\ (F_x)_j \\ (F_y)_j \\ \vdots \end{bmatrix} \tag{B.50}$$

where j is the index of the camera being used.

Equation B.50 can generally not be solved, and the least squares estimate is used as the best approximation of the values of δX, δY and δZ. An iterative implementation of the least squares method is used to refine estimates of X, Y and Z. As the solution converges to the *best* value, F_x and F_y become smaller, although they generally *never* reach 0 due to the small errors in the system.

A good initial estimate of the coordinates is required in order for the iterative algorithm to converge to the correct value. Since the camera orientation parameters are held constant for the object space intersection, the collinearity equations can be manipulated into a linear form without the use of differentials. This method can be used to provide initial estimates of the coordinates of a point. The equations can be represented in matrix form as:

$$\begin{bmatrix} (\Delta x r_{31} - c r_{11})_1 & (\Delta x r_{32} - c r_{12})_1 & (\Delta x r_{33} - c r_{13})_1 \\ (\Delta y r_{31} - c r_{21})_1 & (\Delta y r_{32} - c r_{22})_1 & (\Delta y r_{33} - c r_{23})_1 \\ (\Delta x r_{31} - c r_{11})_2 & (\Delta x r_{32} - c r_{12})_2 & (\Delta x r_{33} - c r_{13})_2 \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix}$$

$$= \begin{bmatrix} ((\Delta x r_{31} - c r_{11})X_c + (\Delta x r_{32} - c r_{12})Y_c + (\Delta x r_{33} - c r_{13})Z_c)_1 \\ ((\Delta y r_{31} - c r_{21})X_c + (\Delta y r_{32} - c r_{22})Y_c + (\Delta y r_{33} - c r_{23})Z_c)_1 \\ ((\Delta x r_{31} - c r_{11})X_c + (\Delta x r_{32} - c r_{12})Y_c + (\Delta x r_{33} - c r_{13})Z_c)_2 \end{bmatrix} \tag{B.51}$$

where Δx and Δy are:

$$\Delta x = x - x_p \tag{B.52}$$

$$\Delta y = y - y_p \tag{B.53}$$

The outer subscripts in equation B.51 refer to the image number. Two observations are used from the first camera (one point), and one observation from the second camera (the x coordinate only). This is a satisfactory method of obtaining the initial estimates of the object space coordinates of a point, but since the equations do not minimise the values of F_x and F_y as described in equations B.16 and B.17, this method is unsatisfactory for obtaining the best values of X, Y and Z, and the iterative method described above should therefore be used to obtain the object space coordinates of the point.

B.3 Accounting for Lens Distortion

The solutions described above ignore any possible lens or film distortion parameters which could be relevant. To include these parameters, it is sufficient to replace equations B.16 and B.17 with

$$F_x = x - x_p + dx - c\frac{R_1}{R_3} \tag{B.54}$$

and

$$F_y = y - y_p + dy - c\frac{R_2}{R_3} \tag{B.55}$$

where dx and dy here represent the total lens distortion in the x and y directions respectively.

Using the model described in section 2.4, dx and dy are:

$$\begin{aligned} dx &= \bar{x}s_x + \bar{y}a + \bar{x}(k_1r^2 + k_2r^4 + k_3r^6) \\ &\quad + P_1(r^2 + 2\bar{x}^2) + 2P_2(\bar{x}\bar{y}) \end{aligned} \tag{B.56}$$

$$\begin{aligned} dy &= \bar{x}a + \bar{y}(k_1r^2 + k_2r^4 + k_3r^6) \\ &\quad + P_2(r^2 + 2\bar{y}^2) + 2P_1(\bar{x}\bar{y}) \end{aligned} \tag{B.57}$$

in which

$$\begin{aligned} \bar{x} &= x - x_p \\ \bar{y} &= y - y_p \\ r &= \sqrt{\bar{x}^2 + \bar{y}^2} \end{aligned}$$

The partial derivatives of dx and dy with respect to each of the additional parameters can be shown to be:

$$\frac{\partial dx}{\partial k_1} = r^2\bar{x} \tag{B.58}$$

$$\frac{\partial dx}{\partial k_2} = r^4\bar{x} \tag{B.59}$$

$$\frac{\partial dx}{\partial k_3} = r^6\bar{x} \tag{B.60}$$

$$\frac{\partial dx}{\partial P_1} = r^2 + 2\bar{x}^2 \tag{B.61}$$

$$\frac{\partial dx}{\partial P_2} = 2(\bar{x}\bar{y}) \tag{B.62}$$

$$\frac{\partial dx}{\partial s_x} = \bar{x} \tag{B.63}$$

$$\frac{\partial dx}{\partial a} = \bar{y} \tag{B.64}$$

and

$$\frac{\partial dy}{\partial k_1} = r^2\overline{y} \quad \text{(B.65)}$$

$$\frac{\partial dy}{\partial k_2} = r^4\overline{y} \quad \text{(B.66)}$$

$$\frac{\partial dy}{\partial k_3} = r^6\overline{y} \quad \text{(B.67)}$$

$$\frac{\partial dy}{\partial P_1} = 2(\overline{x}\,\overline{y}) \quad \text{(B.68)}$$

$$\frac{\partial dy}{\partial P_2} = r^2 + 2\overline{y}^2 \quad \text{(B.69)}$$

$$\frac{\partial dy}{\partial s_x} = 0 \quad \text{(B.70)}$$

$$\frac{\partial dy}{\partial a} = \overline{x} \quad \text{(B.71)}$$

The additional parameters can be solved for during the camera calibration, and will frequently improve the accuracy of subsequent object point intersections.

The lens distortion model used in this chapter is described by (van der Vlugt 1995). Detailed discussions of lens distortion models for use in photogrammetry can be found in (Fryer & Brown 1986), (Brown 1966) and (Karara 1989).

B.4 The Bundle Adjustment

The bundle adjustment, described by van der Vlugt (1995) and by Granshaw (1980), is based on the collinearity equations. The object space coordinates of a number of points and the camera orientation parameters are initially unknown. The camera orientations, and the object space coordinates which best minimise the sum of the squares of the differences between the given image space coordinates of the points and those calculated using the model, may be found as follows:

Equation B.19 then becomes:

$$\delta F_x \approx \frac{\partial F_x}{\partial x_p}\delta x_p + \frac{\partial F_x}{\partial y_p}\delta y_p + \frac{\partial F_x}{\partial c}\delta c \quad \text{(B.72)}$$

$$+\frac{\partial F_x}{\partial \omega}\delta\omega + \frac{\partial F_x}{\partial \kappa}\delta\kappa + \frac{\partial F_x}{\partial \phi}\delta\phi \quad \text{(B.73)}$$

$$+\frac{\partial F_x}{\partial X_c}\delta X_c + \frac{\partial F_x}{\partial Y_c}\delta Y_c + \frac{\partial F_x}{\partial Z_c}\delta Z_c \quad \text{(B.74)}$$

$$+\frac{\partial F_x}{\partial X}\delta X + \frac{\partial F_x}{\partial Y}\delta Y + \frac{\partial F_x}{\partial Z}\delta Z \quad \text{(B.75)}$$

with δX, δY and δZ, and partial derivatives of F_x with respect to the object space coordinates (X, Y, Z) having been included in the equation.

The approximation can be written in matrix form as:

$$
\begin{bmatrix}
\frac{\partial F_x}{\partial C_1}(1,1) & 0 & \dots & 0 & \frac{\partial F_x}{\partial P_1}(1,1) & 0 & \dots & 0 \\
\frac{\partial F_y}{\partial C_1}(1,1) & 0 & \dots & 0 & \frac{\partial F_y}{\partial P_1}(1,1) & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_x}{\partial C_1}(1,n) & 0 & \dots & 0 & 0 & \dots & 0 & \frac{\partial F_x}{\partial P_n}(1,n) \\
\frac{\partial F_y}{\partial C_1}(1,n) & 0 & \dots & 0 & 0 & \dots & 0 & \frac{\partial F_y}{\partial P_n}(1,n) \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & \frac{\partial F_x}{\partial C_m}(m,1) & \frac{\partial F_x}{\partial P_1}(m,1) & 0 & \dots & 0 \\
0 & \dots & 0 & \frac{\partial F_y}{\partial C_m}(m,1) & \frac{\partial F_y}{\partial P_1}(m,1) & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & \frac{\partial F_x}{\partial C_m}(m,n) & 0 & 0 & \dots & \frac{\partial F_x}{\partial P_n}(m,n) \\
0 & \dots & 0 & \frac{\partial F_y}{\partial C_m}(m,n) & 0 & 0 & \dots & \frac{\partial F_y}{\partial P_n}(m,n) \\
\frac{\partial F_x}{\partial C_1}(1,n+1) & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\
\frac{\partial F_y}{\partial C_1}(1,n+1) & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\frac{\partial F_x}{\partial C_1}(1,n+k) & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\
\frac{\partial F_y}{\partial C_1}(1,n+k) & 0 & \dots & 0 & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & \frac{\partial F_x}{\partial C_m}(m,n+1) & 0 & \dots & \dots & 0 \\
0 & \dots & 0 & \frac{\partial F_y}{\partial C_m}(m,n+1) & 0 & \dots & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & \frac{\partial F_x}{\partial C_m}(m,n+k) & 0 & \dots & \dots & 0 \\
0 & \dots & 0 & \frac{\partial F_y}{\partial C_m}(m,n+k) & 0 & \dots & \dots & 0
\end{bmatrix}
\begin{bmatrix}
\delta \mathbf{C_1} \\
\vdots \\
\delta \mathbf{C_m} \\
\delta \mathbf{P_1} \\
\vdots \\
\delta \mathbf{P_n}
\end{bmatrix}
$$

$$
= \begin{bmatrix}
-(F_x)_{(1,1)} \\
-(F_y)_{(1,1)} \\
\vdots \\
\vdots \\
-(F_x)_{m,n} \\
-(F_y)_{m,n} \\
-(F_x)_{m,n+1} \\
-(F_y)_{m,n+1} \\
\vdots \\
\vdots \\
-(F_x)_{m,n+k} \\
-(F_y)_{m,n+k}
\end{bmatrix} \qquad \text{(B.76)}
$$

where

- $\frac{\partial F_x}{\partial \mathbf{C}_i}$ represents the vector $(\frac{\partial F_x}{\partial x_p}, \frac{\partial F_x}{\partial y_p}, \ldots, \frac{\partial F_x}{\partial Y_c}, \frac{\partial F_x}{\partial Z_c})$ for camera i in the set of m cameras.
- $\frac{\partial F_x}{\partial \mathbf{P}_j}$ represents the vector $(\frac{\partial F_x}{\partial X}, \frac{\partial F_x}{\partial Y}, \frac{\partial F_x}{\partial Z})$ for the point j in the set of n (non-control) points.
- $\delta\mathbf{C}_i$ represents the vector $(\delta x_c, \delta y_c, \ldots, \delta Y_c, \delta Z_c)$ for the ith camera
- $\delta\mathbf{P}_j$ represents the vector $(\delta X, \delta Y, \delta Z)$ for the jth point.

The outer subscripts represent the sets of parameters used to evaluate the functions and the partial derivatives. The first parameter in the subscript pair represents the camera orientation parameter set, and the second parameter in the subscript pair represents the point coordinates used. Thus, for example, $\frac{\partial F_x}{\partial \mathbf{C}_{i\,(i,l)}}$ is the vector of partial derivatives of F_x with respect to each of the camera parameters for camera i, evaluated using the camera parameters for camera i, and the point coordinates (image and object space coordinates) for point l. The last k object space - image space coordinate pairs are the control points. The first matrix on the left hand side of the equation above is referred to as the design matrix.

As with the camera calibration using the collinearity equations, the bundle adjustment requires estimates of the initial value of each of the parameters for which a solution is being sought. In addition, it is required that constraints be added to the matrix equation in order to set the scale of the solutions. The scale can be set by adding a set of equations defining the distances between certain points. Alternatively, the object space and image space coordinates of certain points can be considered fixed. The second approach is used above, with the control point image space and object space coordinates being considered constants in the last $2 \times k$ rows of the matrix equation.

Once the small changes have been found, using a least squares approach if the system is over-constrained, these are added to the estimates of the camera parameters for each of the cameras, and to the point coordinates for each of the points, other than the control points. The process is repeated until the results converge to a solution, which is the solution which minimises the sum of the squares of the differences between the left and right hand sides of the collinearity equations for each of the points. An efficient method of implementing the bundle adjustment is discussed by van der Vlugt (1995).

Appendix C

The Direct Linear Transformation

The direct linear transformation, described in (Karara 1989), is defined by the equations:

$$x - dx = \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z + 1} \tag{C.1}$$

$$y - dy = \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z + 1} \tag{C.2}$$

As with the collinearity equations, the terms dx and dy describe the distortion introduced by imperfections in the lens system and the image carrier. These terms are assumed to be zero in the derivations which follow below. Section C.3 explains the alterations needed to include additional parameters.

The method used to solve these equations is similar to that used to solve the collinearity equations, and steps which are common to the two methods are omitted here.

Equations C.1 and C.2 are rewritten as:

$$F_x = x - \frac{B_1}{B_3} \tag{C.3}$$

and

$$F_y = y - \frac{B_2}{B_3} \tag{C.4}$$

where

$$\begin{aligned} B_1 &= L_1X + L_2Y + L_3Z + L_4 \\ B_2 &= L_5X + L_6Y + L_7Z + L_8 \\ B_3 &= L_9X + L_{10}Y + L_{11}Z + 1 \end{aligned}$$

C.1 Camera Calibration

The camera orientation parameters in the DLT model are variables, and the differential δF_x is defined, using Taylor's theorem, as:

$$\begin{aligned}\delta F_x \approx & \frac{\partial F_x}{\partial L_1}\delta L_1 + \frac{\partial F_x}{\partial L_2}\delta L_2 + \frac{\partial F_x}{\partial L_3}\delta L_3 + \frac{\partial F_x}{\partial L_4}\delta L_4 \\ & + \frac{\partial F_x}{\partial L_9}\delta L_9 + \frac{\partial F_x}{\partial L_{10}}\delta L_{10} + \frac{\partial F_x}{\partial L_{11}}\delta L_{11}\end{aligned} \tag{C.5}$$

A corresponding equation exists for δF_y.

Since in the *ideal* case $F_x + \delta F_x = 0$ and $F_y + \delta F_y = 0$ the equations

$$-\delta F_x = F_x \tag{C.6}$$

$$-\delta F_y = F_y \tag{C.7}$$

can be written for each observation set. If more than 11 equations exist, there is generally no set of DLT parameters which will satisfy all of the equations, and F_x and F_y will thus generally *not* be zero as a result of small measurement errors and modelling imperfections. The parameter set which minimises the sum of the squares of the differences between the left and right hand sides of these equations is used as the best approximation to the correct solution.

Writing these equations in a matrix form, with x_i and y_i representing the x and y image coordinates of the ith control point, and with X_i, Y_i and Z_i representing the X, Y and Z object space coordinates of the ith control point, yields the matrix equation:

$$\begin{bmatrix} -\frac{\partial F_x}{\partial L_1}_1 & \cdots & -\frac{\partial F_x}{\partial L_{10}}_1 & -\frac{\partial F_x}{\partial L_{11}}_1 \\ -\frac{\partial F_y}{\partial L_1}_1 & \cdots & -\frac{\partial F_y}{\partial L_{10}}_1 & -\frac{\partial F_y}{\partial L_{11}}_1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{\partial F_x}{\partial L_1}_i & \cdots & -\frac{\partial F_x}{\partial L_{10}}_i & -\frac{\partial F_x}{\partial L_{11}}_i \\ -\frac{\partial F_y}{\partial L_1}_i & \cdots & -\frac{\partial F_y}{\partial L_{10}}_i & -\frac{\partial F_y}{\partial L_{11}}_i \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \delta L_1 \\ \delta L_2 \\ \vdots \\ \delta L_{10} \\ \delta L_{11} \end{bmatrix} = \begin{bmatrix} (F_x)_1 \\ (F_y)_1 \\ \vdots \\ (F_x)_i \\ (F_y)_i \\ \vdots \end{bmatrix} \tag{C.8}$$

The partial derivatives in equation C.8 can be shown to be:

$$\begin{aligned} \frac{\partial F_x}{\partial L_1} &= -\frac{X}{B_3} \\ \frac{\partial F_x}{\partial L_2} &= -\frac{Y}{B_3} \\ \frac{\partial F_x}{\partial L_3} &= -\frac{Z}{B_3} \\ \frac{\partial F_x}{\partial L_4} &= -\frac{1}{B_3} \\ \frac{\partial F_x}{\partial L_5} &= 0 \end{aligned}$$

$$\begin{aligned}
\frac{\partial F_x}{\partial L_6} &= 0 \\
\frac{\partial F_x}{\partial L_7} &= 0 \\
\frac{\partial F_x}{\partial L_8} &= 0 \\
\frac{\partial F_x}{\partial L_9} &= \frac{XB_1}{{B_3}^2} \approx \frac{Xx}{B_3} \\
\frac{\partial F_x}{\partial L_{10}} &= \frac{YB_1}{{B_3}^2} \approx \frac{Yx}{B_3} \\
\frac{\partial F_x}{\partial L_{11}} &= \frac{ZB_1}{{B_3}^2} \approx \frac{Zx}{B_3}
\end{aligned}$$

and

$$\begin{aligned}
\frac{\partial F_y}{\partial L_1} &= 0 \\
\frac{\partial F_y}{\partial L_2} &= 0 \\
\frac{\partial F_y}{\partial L_3} &= 0 \\
\frac{\partial F_y}{\partial L_4} &= 0 \\
\frac{\partial F_y}{\partial L_5} &= -\frac{X}{B_3} \\
\frac{\partial F_y}{\partial L_6} &= -\frac{Y}{B_3} \\
\frac{\partial F_y}{\partial L_7} &= -\frac{Z}{B_3} \\
\frac{\partial F_y}{\partial L_8} &= -\frac{1}{B_3} \\
\frac{\partial F_y}{\partial L_9} &= \frac{XB_2}{{B_3}^2} \approx \frac{Xy}{B_3} \\
\frac{\partial F_y}{\partial L_{10}} &= \frac{YB_2}{{B_3}^2} \approx \frac{Yy}{B_3} \\
\frac{\partial F_y}{\partial L_{11}} &= \frac{ZB_2}{{B_3}^2} \approx \frac{Zy}{B_3}
\end{aligned}$$

The same iterative approach used for the collinearity equations is used here to find the best values for the parameters.

C.2 Object Point Intersection

As with the collinearity equations, minor alterations are needed to solve for the object space coordinates of a point, given the image coordinates of the point in

two or more images. The object space coordinates are variables for this purpose, and the camera orientation parameters are constants.

The value δF_x is given by Taylor's theorem to be:

$$\delta F_x \approx \frac{\partial F_x}{\partial X}\delta X + \frac{\partial F_x}{\partial Y}\delta Y + \frac{\partial F_x}{\partial Z}\delta Z \tag{C.9}$$

A corresponding equation exists for δF_y.

Since it is desired that $F_x = -\delta F_x$ and $F_y = -\delta F_y$, the following matrix equation can be written:

$$\begin{bmatrix} \left(\frac{\partial F_x}{\partial X}\right)_1 & \left(\frac{\partial F_x}{\partial Y}\right)_1 & \left(\frac{\partial F_x}{\partial Z}\right)_1 \\ \left(\frac{\partial F_x}{\partial X}\right)_1 & \left(\frac{\partial F_x}{\partial Y}\right)_1 & \left(\frac{\partial F_x}{\partial Z}\right)_1 \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ \left(\frac{\partial F_x}{\partial X}\right)_j & \left(\frac{\partial F_x}{\partial Y}\right)_j & \left(\frac{\partial F_x}{\partial Z}\right)_j \\ \left(\frac{\partial F_x}{\partial X}\right)_j & \left(\frac{\partial F_x}{\partial Y}\right)_j & \left(\frac{\partial F_x}{\partial Z}\right)_j \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \end{bmatrix} \begin{bmatrix} \delta X \\ \delta Y \\ \delta Z \end{bmatrix} = \begin{bmatrix} -(F_x)_1 \\ -(F_y)_1 \\ \vdots \\ -(F_x)_j \\ -(F_y)_j \\ \vdots \end{bmatrix} \tag{C.10}$$

In practice, equation C.10 can generally not be solved as a result of small measurement errors and imperfect modelling. The *best* values of δX, δY, and δZ are found by the iterative least squares approach used previously.

The iterative method described above requires initial estimates for the values of X, Y, and Z. Since the camera orientation parameters are known, and are constants, the DLT equations can be written as a linear equation without the use of differentials. This linear equation can be used to solve for the values of the point coordinates. The equations can be written in matrix form as:

$$\begin{bmatrix} (\Delta x\, L_9 - L_1)_1 & (\Delta x\, L_{10} - L_2)_1 & (\Delta x\, L_{11} - L_3)_1 \\ (\Delta y\, L_9 - L_4)_1 & (\Delta y\, L_{10} - L_5)_1 & (\Delta y\, L_{11} - L_6)_1 \\ (\Delta x\, L_9 - L_1)_2 & (\Delta x\, L_{10} - L_2)_2 & (\Delta x\, L_{11} - L_3)_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} (L_4 - \Delta x)_1 \\ (L_8 - \Delta y)_1 \\ (L_4 - \Delta x)_2 \end{bmatrix} \tag{C.11}$$

where Δx and Δy are:

$$\Delta x = x + dx \tag{C.12}$$

$$\Delta y = y + dy \tag{C.13}$$

$$\tag{C.14}$$

It is assumed here that dx and dy are 0, but this will not generally be the case if additional parameters are used to account for lens distortion. The outer subscripts in equation C.11 refer to the image number from which the x and y coordinates have been determined. This equation can be solved by matrix inversion to yield an initial estimate of the object space coordinates of the point being determined.

C.3 Accounting for Lens Distortions

If lens distortion parameters are required, these can be included in the DLT equations as is described for the collinearity equations in section B.3.

C.4 Conversion of DLT Parameters to Standard Camera Orientation Parameters

The DLT parameters can be used to find the standard camera orientation parameters as follows:

The perspective centre is found by solving the matrix equation:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 \\ L_5 & L_6 & L_7 \\ L_9 & L_{10} & L_{11} \end{bmatrix} \begin{bmatrix} L_4 \\ L_8 \\ 1 \end{bmatrix} \tag{C.15}$$

The x and y coordinates of the principal point are:

$$x_p = (L_1L_9 + L_2L_{10} + L_3L_{11})L^2 \tag{C.16}$$

$$y_p = (L_5L_9 + L_6L_{10} + L_7L_{11})L^2 \tag{C.17}$$

where

$$L = \frac{-1}{\sqrt{L_9^2 + L_{10}^2 + L_{11}^2}} \tag{C.18}$$

The principal distance is given by

$$c = \frac{c_x + c_y}{2} \tag{C.19}$$

where

$$c_x = \sqrt{(L_1^2 + L_2^2 + L_3^2)L^2 - x_p} \tag{C.20}$$

$$c_y = \sqrt{(L_5^2 + L_6^2 + L_7^2)L^2 - y_p} \tag{C.21}$$

The rotation angles can be determined from:

$$\kappa = \sin^{-1}(r_{13}) \tag{C.22}$$

$$\omega = \tan^{-1}\frac{(-r_{23})}{(r_{33})} \tag{C.23}$$

$$\phi = \tan^{-1}\frac{(-r_{12})}{(r_{11})} \tag{C.24}$$

where

$$r_{11} = L^2\left[\frac{L_6L_{11} - L_7L_{10}}{c_y}\right] \quad \text{(C.25)}$$

$$r_{12} = L^2\left[\frac{L_7L_9 - L_5L_{11}}{c_y}\right] \quad \text{(C.26)}$$

$$r_{13} = L^2\left[\frac{L_5L_{10} - L_6L_9}{c_y}\right] \quad \text{(C.27)}$$

$$r_{21} = L^2\left[\frac{L_3L_{10} - L_2L_{11}}{c_x}\right] \quad \text{(C.28)}$$

$$r_{22} = L^2\left[\frac{L_1L_{11} - L_3L_9}{c_x}\right] \quad \text{(C.29)}$$

$$r_{23} = L^2\left[\frac{L_2L_9 - L_1L_{10}}{c_x}\right] \quad \text{(C.30)}$$

$$r_{31} = L_9L \quad \text{(C.31)}$$

$$r_{32} = L_{10}L \quad \text{(C.32)}$$

$$r_{33} = L_{11}L \quad \text{(C.33)}$$

The values of r_{31}, r_{32} and r_{33}, and the angles calculated using these values, should be calculated using L, and then using $-L$. This yields two possible sets of angles, of which only one set is valid. The correct solution is determined by recalculating the rotation matrix using ω, κ and ϕ. The correct solution yields a rotation matrix which matches the matrix calculated using equations C.25 - C.33.

Appendix D

Smith's Explicit Space Resection

This resection, presented by Smith (1965), is an explicit method of determining the exterior orientation of a camera using four control points.

Assume that the image and object space coordinates of three points, A, B and C are known for a given image. Let **A**, **B** and **C** be the object space coordinates and $\mathbf{A_1}$, $\mathbf{B_1}$ and $\mathbf{C_1}$ the image coordinates (in vector form) of points A, B and C respectively. The image coordinates of A, B and C are (x_1, y_1), (x_2, y_2) and (x_3, y_3) respectively, while the object space coordinates of the points are (X_1, Y_1, Z_1), (X_2, Y_2, Z_2) and (X_3, Y_3, Z_3) respectively. The perspective centre of the camera is labeled P in this discussion. These points are illustrated in figure D.1. The symbol f is used to represent the principal distance to avoid confusion with c which is used here for other purposes.

The object space coordinates of the three points define a triangle in object space, $\triangle$ABC. The sides opposite A, B and C have lengths a, b and c respectively. The angles α, β and γ are subtended at P by $\triangle$ABC, and are given by:

$$\cos\alpha = \frac{x_2x_3 + y_2y_3 + f^2}{\sqrt{(x_2^2 + y_2^2 + f^2)(x_3^2 + y_3^2 + f^2)}} \tag{D.1}$$

$$\cos\beta = \frac{x_3x_1 + y_3y_1 + f^2}{\sqrt{(x_3^2 + y_2^3 + f^2)(x_1^2 + y_1^2 + f^2)}} \tag{D.2}$$

$$\cos\gamma = \frac{x_1x_2 + y_1y_2 + f^2}{\sqrt{(x_1^2 + y_1^2 + f^2)(x_2^2 + y_2^2 + f^2)}} \tag{D.3}$$

Let the lengths of **PA**, **PB** and **PC** be u, v, and w respectively. It can be

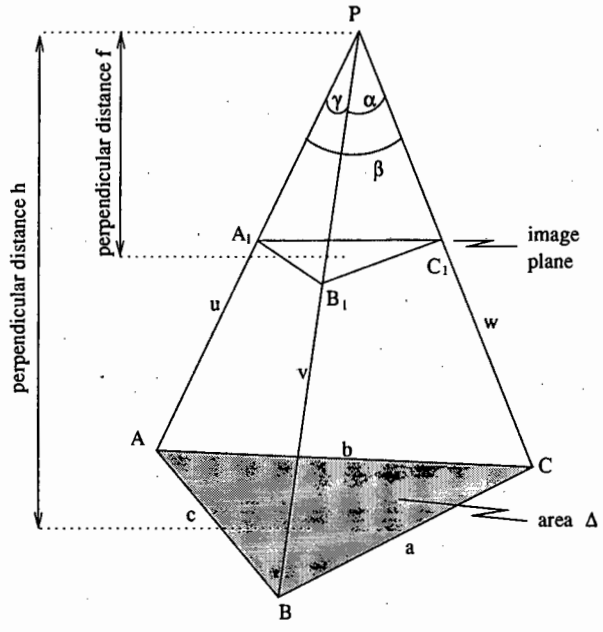

Figure D.1: Points and Angles in Smith's Resection

shown that:

$$u = \frac{1}{\sqrt{c(1 - 2m_1 \cos\gamma + m_1^2)}} \tag{D.4}$$

$$v = m_1 u \tag{D.5}$$

$$w = m_2 u \tag{D.6}$$

where m_1 can be found by solving for the roots of the equation

$$\mathcal{A}m_1^4 + \mathcal{B}m_1^3 + \mathcal{C}m_1^2 + \mathcal{D}m_1 + \mathcal{E} = 0 \tag{D.7}$$

where

$$\mathcal{A} = (b^2 + c^2 - a^2)^2 - 4b^2c^2\cos^2\alpha \tag{D.8}$$

$$\mathcal{B} = \mathcal{K} - 2\mathcal{A}\cos\gamma \tag{D.9}$$

$$\mathcal{C} = \mathcal{A} + \mathcal{E} - 2\mathcal{K}\cos\gamma + 4c^4(\cos^2\gamma + \cos^2\beta + \cos^2\gamma - 2\cos\gamma\cos\beta\cos\gamma - 1) \tag{D.10}$$

$$\mathcal{D} = \mathcal{K} - 2\mathcal{E}\cos\gamma \tag{D.11}$$

and where

$$\mathcal{K} = 2(b^2 + c^2 - a^2)(a^2 + c^2 - b^2)\cos\gamma + 4c^2(a^2 + b^2 - c^2)\cos\gamma\cos\beta \tag{D.12}$$

$$\mathcal{E} = (a^2 + c^2 + b^2)^2 - 4c^2a^2\cos^2\beta \tag{D.13}$$

The value of m_2 can be found from

$$m_2 = \frac{a^2 + c^2 - b^2 + 2(b^2 - a^2)m_1 \cos\gamma - (b^2 + c^2 - a^2)m_1^2}{2c^2(\cos\beta - m_1 \cos\gamma)} \qquad \text{(D.14)}$$

once the value of m_1 is known.

The polynomial in equation D.7 can be solved using any of a number of standard methods. Smith suggests the use of Bairstow's method for finding roots of a polynomial. A maximum of four real roots to equation D.7 exist. A maximum of four sets of values for (u, v, w) can thus be found. Equation D.4 is assumed to be positive, since u is a distance, as are v and w. All these sets of distances are geometrically possible, since α, β and γ may be smaller or larger than 90°. A fourth control point, D, is required to discriminate between the correct solution and the other solutions. The method used to discriminate between solution sets is presented later.

The position of P can be calculated from the values u, v and w (assumed to be correct) as follows:

Let Δ be the area of $\triangle$ABC, which is calculated using the formula:

$$\Delta = \sqrt{s(s-a)(s-b)(s-c)} \qquad \text{(D.15)}$$

where $2s = a + b + c$

Let the length of the perpendicular from P to the plane on which points A, B and C lie be h. The volume of the tetrahedron PABC can be shown to be:

$$V = \frac{\Delta h}{3} \qquad \text{(D.16)}$$

It can also be shown that:

$$V = \frac{uvw}{6}\sqrt{1 + 2\cos\alpha\cos\beta\cos\gamma - \cos^2\alpha - \cos^2\beta - \cos^2\gamma} \qquad \text{(D.17)}$$

From these it follows that

$$h = \frac{uvw}{2\Delta}\sqrt{1 + 2\cos\alpha\cos\beta\cos\gamma - \cos^2\alpha - \cos^2\beta - \cos^2\gamma} \qquad \text{(D.18)}$$

The direction cosines, λ_1, λ_2 and λ_3, of the normal to plane ABC are given by:

$$\lambda_1 = \frac{\Delta_1}{\Delta} \qquad \text{(D.19)}$$

$$\lambda_2 = \frac{\Delta_2}{\Delta} \qquad \text{(D.20)}$$

$$\lambda_3 = \frac{\Delta_3}{\Delta} \qquad \text{(D.21)}$$

where Δ_1, Δ_2 and Δ_3 are the areas of the triangles found by projecting $\triangle$ABC onto the planes $X = 0$, $Y = 0$ and $Z = 0$ respectively. These are given by:

$$\Delta_1 = \pm\frac{1}{2}(Y_1(Z_2 - Z_3) + Y_2(Z_3 - Z_1) + Y_3(Z_1 - Z_2)) \qquad \text{(D.22)}$$

$$\Delta_2 = \pm\frac{1}{2}(Z_1(X_2 - X_3) + Z_2(X_3 - X_1) + Z_3(X_1 - X_2)) \quad \text{(D.23)}$$

$$\Delta_3 = \pm\frac{1}{2}(X_1(Y_2 - Y_3) + X_2(Y_3 - Y_1) + X_3(Y_1 - Y_2)) \quad \text{(D.24)}$$

The signs of the directions cosines should be either all positive or all negative, depending upon the side of plane ABC on which point P lies. Which set is chosen will be determined by the particular situation.

Let X_P, Y_P, and Z_P be the coordinates of P. The plane ABC has the equation:

$$\lambda_1(X - X_1) + \lambda_2(Y - Y_1) + \lambda_3(Z - Z_1) = 0 \quad \text{(D.25)}$$

From this the length h can be determined:

$$h = \lambda_1(X_P - X_1) + \lambda_2(Y_P - Y_1) + \lambda_3(Z_P - Z_1) \quad \text{(D.26)}$$

The coordinates of P can therefore be found by solving the following system of linear equations:

$$h = \lambda_1(X_P - X_1) + \lambda_2(Y_P - Y_1) + \lambda_3(Z_P - Z_1) \quad \text{(D.27)}$$

$$v^2 - u^2 = 2X_P(X_1 - X_2) + 2Y_P(Y_1 - Y_2) + 2Z_P(Z_1 - Z_2) + X_2^2 + Y_2^2 + Z_2^2 - X_1^2 - Y_1^2 - Z_1^2 \quad \text{(D.28)}$$

$$w^2 - u^2 = 2X_P(X_1 - X_3) + 2Y_P(Y_1 - Y_3) + 2Z_P(Z_1 - Z_3) + X_3^2 + Y_3^2 + Z_3^2 - X_1^2 - Y_1^2 - Z_1^2 \quad \text{(D.29)}$$

By definition, the rotation matrix $\mathbf{R}$ is defined such that:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \mathbf{R} \begin{bmatrix} X - X_P \\ Y - Y_P \\ Z - Z_P \end{bmatrix} \quad \text{(D.30)}$$

where x, y and z are the image space coordinates of a point relative to the image space origin which is at the perspective centre of the image. The terms X, Y and Z are the object space coordinates of the same point. The matrix $\mathbf{R}$ is the 3×3 rotation matrix. The perspective centre P, the image point (x_1, y_1, z_1), and the point, (X_1, Y_1, Z_1) must be collinear. Similar statements hold for points B, C and D. The distance from P to point A is given by u and so:

$$\frac{X_1}{x_1} = \frac{Y_1}{y_1} = \frac{Z_1}{f} = \frac{u}{\sqrt{x_1^2 + y_1^2 + f^2}} \quad \text{(D.31)}$$

The elements of the rotation matrix can be found by solving the following system of linear equations:

$$r_{11}(X_1 - X_P) + r_{12}(Y_1 + Y_P) + r_{13}(Z_1 - Z_P) = \frac{ux_1}{\sqrt{x_1^2 + y_1^2 + f^2}} \quad \text{(D.32)}$$

$$r_{11}(X_2 - X_P) + r_{12}(Y_2 - Y_P) + r_{13}(Z_2 - Z_P) = \frac{ux_2}{\sqrt{x_2^2 + y_2^2 + f^2}} \quad \text{(D.33)}$$

$$r_{11}(X_3 - X_P) + r_{12}(Y_3 + Y_P) + r_{13}(Z_3 - Z_P) = \frac{ux_3}{\sqrt{x_3^2 + y_3^2 + f^2}} \quad \text{(D.34)}$$

The other elements of $\mathbf{R}$ can be found using similar equations.

In order to select values for u, v and w from the four or fewer possible solutions, a fourth control point is required. Using the fourth control point's image coordinates, and the rotation matrix and perspective centre calculated for each of the possible sets of distances (u, v, w), the object space coordinates for the fourth control point are calculated. The set (u, v, w) which minimises the distance between the calculated point D and the vertical through D is selected as the correct solution.

The equation of the line PD in image space is given by:

$$\frac{x}{x_4} = \frac{y}{y_4} = \frac{z}{f} \quad \text{(D.35)}$$

Transforming this equation into an equation in object space coordinates gives:

$$\begin{aligned} & \frac{r_{11}(X - X_P) + r_{12}(Y - Y_P) + r_{13}(Z - Z_P)}{x_4} \\ = & \frac{r_{21}(X - X_P) + r_{22}(Y - Y_P) + r_{23}(Z - Z_P)}{y_4} \\ = & \frac{r_{31}(X - X_P) + r_{32}(Y - Y_P) + r_{33}(Z - Z_P)}{f} \end{aligned}$$

Manipulating these ratios gives

$$\begin{aligned} & \frac{X - X_P}{r_{11}x_4 + r_{21}y_4 + r_{31}f} \\ = & \frac{Y - Y_P}{r_{12}x_4 + r_{22}y_4 + r_{32}f} \\ = & \frac{Z - Z_P}{r_{13}x_4 + r_{23}y_4 + r_{33}f} \end{aligned}$$

The vertical line through D is given by

$$\begin{aligned} X &= X_4 \\ Y &= Y_4 \end{aligned}$$

and the minimum distance between the line PD and this line can be shown to be d, where

$$d = \left| \frac{(r_{11}x_4 + r_{21}y_4 - r_{31}f)(Y_P - Y_4) - (r_{12}x_4 + r_{22}y_4 - r_{32}f)(X_P - X_4)}{\sqrt{(r_{11}x_4 + r_{21}y_4 - r_{31}f)^2 + (r_{12}x_4 + r_{22}y_4 - r_{32}f)^2}} \right| \quad \text{(D.36)}$$

The value of d is calculated for all possible solutions, and the solution which yields the minimum value of d is assumed to be correct. This assumption relies on the point positions being accurately identified.

Appendix E

Schmid's Iterative Space Resection

This method of solving for the exterior orientation was developed at the same time as the bundle adjustment. It also relies on the collinearity equations (B.1 and B.2), but assumes that the interior orientation of the camera is known, and is held constant. The algorithm, described in (Thompson 1966), uses determinants to represent the differential terms, which are derived as in appendix B. A least squares approach is used to find the best solutions for the differential terms, and these differentials are used to update the initial estimates of the exterior orientation parameters.

Since $z = c$ for all points on an image, where c is the principal distance of the camera, the following two equations can be written:

$$\begin{aligned} x\,((X-X_c)r_{31} + (Y-Y_c)r_{32} + (Z-Z_c)r_{33}) & \\ = \; z\,((X-X_c)r_{11} + (Y-Y_c)r_{12} + (Z-Z_c)r_{13}) & \qquad \text{(E.1)} \\ y\,((X-X_c)r_{31} + (Y-Y_c)r_{32} + (Z-Z_c)r_{33}) & \\ = \; z\,((X-X_c)r_{21} + (Y-Y_c)r_{22} + (Z-Z_c)r_{23}) & \qquad \text{(E.2)} \end{aligned}$$

where $r_{11}, r_{12}, \ldots, r_{33}$ are the elements of the rotation matrix $\mathbf{R}$. The position of the perspective centre is given by (X_c, Y_c, Z_c). The coordinates (x, y) and (X, Y, Z) are the image space and object space coordinates of an arbitrary control point respectively.

Setting

$$\mathbf{R}_i = \begin{bmatrix} r_{i1} & r_{i2} & r_{i3} \end{bmatrix} \qquad \text{(E.3)}$$

and

$$\mathbf{X} = \begin{bmatrix} X-X_c \\ Y-Y_c \\ Z-Z_c \end{bmatrix} \qquad \text{(E.4)}$$

equation E.1 and equation E.2 can be written in determinant form as

$$\begin{vmatrix} x & z \\ \mathbf{R}_1\mathbf{X} & \mathbf{R}_3\mathbf{X} \end{vmatrix} = 0 \tag{E.5}$$

and

$$\begin{vmatrix} y & z \\ \mathbf{R}_2\mathbf{X} & \mathbf{R}_3\mathbf{X} \end{vmatrix} = 0 \tag{E.6}$$

Setting $\mathbf{R}_1\mathbf{X} = r$ and $\mathbf{R}_3\mathbf{X} = q$ and writing out the determinant for equation E.5 yields the equation which should be satisfied:

$$qx - rz = 0 \tag{E.7}$$

Setting

$$F_x = qx - rz \tag{E.8}$$

and using a single term of the Taylor expansion of F_x gives:

$$F_{xnew} = F_{xold} + \delta F_x \tag{E.9}$$

Since ideally $F_{xnew} = 0$

$$F_{xold} = -\delta F_x \tag{E.10}$$

The value of δF_x can be found from:

$$\delta F_x = \frac{\partial F_x}{\partial x}\delta x + \frac{\partial F_x}{\partial \omega}\delta\omega + \ldots + \frac{\partial F_x}{\partial X_c}\delta X_c + \frac{\partial F_x}{\partial Y_c}\delta Y_c + \frac{\partial F_x}{\partial Z_c}\delta Z_c \tag{E.11}$$

where x is considered a variable.

The value of each of the differentials can be calculated from determinants as shown below:

$$\frac{\partial F_x}{\partial x} = q \tag{E.12}$$

$$\frac{\partial F_x}{\partial \omega} = \begin{vmatrix} x & z \\ \frac{\partial R_1}{\partial \omega}\mathbf{X} & \frac{\partial R_3}{\partial \omega}\mathbf{X} \end{vmatrix} \tag{E.13}$$

$$\frac{\partial F_x}{\partial \kappa} = \begin{vmatrix} x & z \\ \frac{\partial R_1}{\partial \kappa}\mathbf{X} & \frac{\partial R_3}{\partial \kappa}\mathbf{X} \end{vmatrix} \tag{E.14}$$

$$\frac{\partial F_x}{\partial \phi} = \begin{vmatrix} x & z \\ \frac{\partial R_1}{\partial \phi}\mathbf{X} & \frac{\partial R_3}{\partial \phi}\mathbf{X} \end{vmatrix} \tag{E.15}$$

$$\frac{\partial F_x}{\partial X_c} = \begin{vmatrix} x & z \\ r_{11} & r_{31} \end{vmatrix} \tag{E.16}$$

$$\frac{\partial F_x}{\partial Y_c} = \begin{vmatrix} x & z \\ r_{12} & r_{32} \end{vmatrix} \tag{E.17}$$

$$\frac{\partial F_x}{\partial Z_c} = \begin{vmatrix} x & z \\ r_{13} & r_{33} \end{vmatrix} \tag{E.18}$$

where $\frac{\partial \mathbf{R}_1}{\partial \omega}$ for example is the first row of the matrix $\frac{\partial \mathbf{R}}{\partial \omega}$. The values of the partial derivatives of $\mathbf{R}$ are:

$$\frac{\partial \mathbf{R}}{\partial \omega} = \begin{bmatrix} 0 & 0 & 0 \\ -r_{31} & -r_{32} & -r_{33} \\ r_{21} & r_{22} & r_{23} \end{bmatrix}$$

$$\frac{\partial \mathbf{R}}{\partial \kappa} = \begin{bmatrix} -\sin\kappa\cos\phi & \sin\kappa\sin\phi & \cos\kappa \\ \sin\omega\cos\kappa\cos\phi & -\sin\omega\cos\kappa\sin\phi & \sin\omega\sin\kappa \\ -\cos\omega\cos\kappa\cos\phi & \cos\omega\cos\kappa\sin\phi & -\cos\omega\sin\kappa \end{bmatrix}$$

$$\frac{\partial \mathbf{R}}{\partial \phi} = \begin{bmatrix} r_{12} & -r_{11} & 0 \\ r_{22} & -r_{21} & 0 \\ r_{32} & -r_{31} & 0 \end{bmatrix}$$

Since a solution requires that $\delta F_x = -F_{x\,old}$, equation E.11 can be rewritten as:

$$-\delta x = \frac{1}{q}\frac{\partial F_x}{\partial \omega}\delta\omega + \ldots + \frac{1}{q}\frac{\partial F_x}{\partial Z_c}\delta Z_c + \frac{1}{q}F_{x\,old} \qquad \text{(E.19)}$$

In order to minimise the value of δx, the required correction to the value of x to satisfy equation E.10, a least squares method is used to find the best solution (in the least squares sense), to the equation:

$$\frac{1}{q}\frac{\partial F_x}{\partial \omega}\delta\omega + \ldots + \frac{1}{q}\frac{\partial F_x}{\partial Z_c}\delta Z_c + \frac{1}{q}F_{x\,old} = 0 \qquad \text{(E.20)}$$

and the corresponding equation for F_y. The equations for F_y are obtained by replacing $\mathbf{R}_1$ and x with $\mathbf{R}_2$ and y respectively in equations E.7 to E.18.

The initial estimates of the exterior orientation parameters are updated using the values of $\delta\omega$, $\delta\kappa$, $\delta\phi$, δX_c, δY_c and δZ_c calculated above. The process is repeated using the updated values for the exterior orientation parameters, until these values converge.

Appendix F

System Test Results

This appendix lists complete results for the system tests described in chapter 7.

Table F.1: Results for Calibration Frame Using the Bundle Adjustment With Additional Parameters

point number	X(mm)	Y(mm)	Z(mm)
format	measured coordinate calculated coordinate difference		
Non-Control Points			
6	1005.45	929.09	999.39
	1005.58	928.74	998.97
	-0.13	0.35	0.42
27	737.53	795.97	1078.76
	737.15	796.16	1078.05
	0.38	-0.19	0.71
Control Points			
1	1003.34	1002.14	999.45
	1003.07	1001.85	998.96
	0.27	0.29	0.49
2	929.50	1002.96	999.52
	929.53	1002.93	999.60
	-0.03	0.03	-0.08
3	859.58	1003.54	999.51
	859.66	1003.48	999.56
	-0.08	0.06	-0.05
4	790.61	1003.23	999.48
	790.62	1003.42	999.90
	-0.01	-0.19	-0.42

5	716.38	1002.48	999.20
	716.42	1002.31	998.87
	-0.04	0.17	0.33
7	715.70	928.43	999.00
	715.49	928.66	999.04
	0.21	-0.23	-0.04
8	1005.46	863.20	999.29
	1005.54	863.24	999.42
	-0.08	-0.04	-0.13
9	715.81	860.50	998.61
	715.51	860.60	998.55
	0.30	-0.10	0.06
10	1005.85	790.26	998.90
	1005.68	790.56	998.65
	0.17	-0.30	0.25
11	716.10	791.38	998.26
	716.79	791.50	998.27
	-0.69	-0.12	-0.01
12	1002.38	718.53	996.95
	1002.60	718.23	997.46
	-0.22	0.30	-0.51
13	930.91	715.84	997.95
	930.84	715.75	997.54
	0.07	0.09	0.41
14	856.98	715.04	997.73
	856.94	714.94	997.46
	0.04	0.10	0.27
15	786.88	715.43	997.33
	786.56	715.40	998.27
	0.32	0.03	-0.94
16	717.62	718.52	997.14
	717.69	718.57	996.99
	-0.07	-0.05	0.15
17	979.59	981.46	1079.60
	979.80	981.60	1079.88
	-0.21	-0.14	-0.28
18	921.48	983.88	1079.29
	921.76	984.02	1080.06
	-0.28	-0.14	-0.77
19	860.75	984.55	1079.15
	860.85	984.61	1079.40
	-0.10	-0.06	-0.25
20	800.56	981.18	1078.98
	800.37	981.12	1078.72
	0.19	0.06	0.26

21	739.71	976.59	1078.60
	739.72	976.76	1078.58
	-0.01	-0.17	0.02
22	984.21	924.80	1080.05
	984.66	925.59	1081.48
	-0.45	-0.79	-1.43
23	736.40	921.80	1078.73
	736.33	921.37	1078.38
	0.07	0.43	0.35
24	987.00	860.15	1080.05
	986.84	859.78	1078.82
	0.16	0.37	1.23
25	734.94	859.65	1078.72
	734.96	859.36	1078.55
	-0.02	0.29	0.17
26	985.34	796.66	1079.88
	985.28	797.08	1080.30
	0.06	-0.42	-0.42

Table F.2: Results for Calibration Frame Using the Bundle Adjustment Without Additional Parameters

point number	X (mm)	Y (mm)	Z (mm)
format	measured coordinate calculated coordinate difference		
Non-Control Points			
6	1006.43	928.55	999.68
	1005.45	929.09	999.39
	0.98	-0.54	0.29
27	737.88	796.16	1076.88
	737.53	795.97	1078.76
	0.35	0.19	-1.88
Control Points			
1	1003.01	1000.39	999.00
	1003.34	1002.14	999.45
	-0.33	-1.75	-0.45
2	929.84	1002.34	998.65
	929.50	1002.96	999.52
	0.34	-0.62	-0.88
3	859.56	1003.24	998.51
	859.58	1003.54	999.51
	-0.02	-0.30	-1.00
4	790.25	1002.72	998.85
	790.61	1003.23	999.48
	-0.36	-0.51	-0.63
5	716.59	1001.32	999.79
	716.38	1002.48	999.20
	0.21	-1.16	0.59
7	714.98	928.22	998.81
	715.70	928.43	999.00
	-0.72	-0.21	-0.19
8	1006.29	863.14	999.21
	1005.46	863.20	999.29
	0.83	-0.06	-0.08
9	714.62	860.44	998.39
	715.81	860.50	998.61
	-1.19	-0.06	-0.22
10	1006.26	790.26	998.49
	1005.85	790.26	998.90
	0.41	0.00	-0.41
11	716.16	791.47	998.14
	716.10	791.38	998.26
	0.06	0.09	-0.12

12	1002.28	718.72	998.00
	1002.38	718.53	996.95
	-0.10	0.19	1.05
13	931.00	715.77	996.67
	930.91	715.84	997.95
	0.09	-0.07	-1.28
14	856.96	714.60	996.71
	856.98	715.04	997.73
	-0.02	-0.44	-1.02
15	786.51	715.43	997.23
	786.88	715.43	997.33
	-0.37	0.00	-0.10
16	717.89	719.57	998.29
	717.62	718.52	997.14
	0.27	1.05	1.15
17	979.93	981.43	1080.37
	979.59	981.46	1079.60
	0.34	-0.03	0.77
18	922.13	984.76	1080.94
	921.48	983.88	1079.29
	0.65	0.88	1.65
19	860.88	985.56	1080.54
	860.75	984.55	1079.15
	0.13	1.01	1.39
20	800.18	981.85	1079.84
	800.56	981.18	1078.98
	-0.38	0.67	0.86
21	739.68	976.38	1078.48
	739.71	976.59	1078.60
	-0.03	-0.21	-0.12
22	984.47	924.90	1079.95
	984.21	924.80	1080.05
	0.26	0.10	-0.10
23	736.15	921.81	1078.12
	736.40	921.80	1078.73
	-0.25	0.01	-0.61
24	987.19	860.44	1079.61
	987.00	860.15	1080.05
	0.19	0.29	-0.44
25	734.77	859.61	1077.55
	734.94	859.65	1078.72
	-0.17	-0.04	-1.17
26	985.08	797.56	1079.40
	985.34	796.66	1079.88
	-0.26	0.90	-0.48

Table F.3: Results for Calibration Frame Using the Collinearity Equations With Additional Parameters

point number	X (mm)	Y (mm)	Z (mm)
format	measured coordinate calculated coordinate difference		
Non-Control Points			
6	1005.45	929.09	999.39
	1005.05	928.82	999.42
	0.40	0.27	-0.03
27	737.53	795.97	1078.76
	737.62	795.80	1078.78
	-0.09	0.17	-0.02
Control Points			
1	1003.34	1002.14	999.45
	1003.07	1001.85	998.96
	0.27	0.29	0.49
2	929.50	1002.96	999.52
	929.53	1002.93	999.60
	-0.03	0.03	-0.08
3	859.58	1003.54	999.51
	859.66	1003.48	999.56
	-0.08	0.06	-0.05
4	790.61	1003.23	999.48
	790.62	1003.42	999.90
	-0.01	-0.19	-0.42
5	716.38	1002.48	999.20
	716.42	1002.31	998.87
	-0.04	0.17	0.33
7	715.70	928.43	999.00
	715.49	928.66	999.04
	0.21	-0.23	-0.04
8	1005.46	863.20	999.29
	1005.54	863.24	999.42
	-0.08	-0.04	-0.13
9	715.81	860.50	998.61
	715.51	860.60	998.55
	0.30	-0.10	0.06
10	1005.85	790.26	998.90
	1005.68	790.56	998.65
	0.17	-0.30	0.25
11	716.10	791.38	998.26
	716.79	791.50	998.27
	-0.69	-0.12	-0.01

12	1002.38	718.53	996.95
	1002.60	718.23	997.46
	-0.22	0.30	-0.51
13	930.91	715.84	997.95
	930.84	715.75	997.54
	0.07	0.09	0.41
14	856.98	715.04	997.73
	856.94	714.94	997.46
	0.04	0.10	0.27
15	786.88	715.43	997.33
	786.56	715.40	998.27
	0.32	0.03	-0.94
16	717.62	718.52	997.14
	717.69	718.57	996.99
	-0.07	-0.05	0.15
17	979.59	981.46	1079.60
	979.80	981.60	1079.88
	-0.21	-0.14	-0.28
18	921.48	983.88	1079.29
	921.76	984.02	1080.06
	-0.28	-0.14	-0.77
19	860.75	984.55	1079.15
	860.85	984.61	1079.40
	-0.10	-0.06	-0.25
20	800.56	981.18	1078.98
	800.37	981.12	1078.72
	0.19	0.06	0.26
21	739.71	976.59	1078.60
	739.72	976.76	1078.58
	-0.01	-0.17	0.02
22	984.21	924.80	1080.05
	984.66	925.59	1081.48
	-0.45	-0.79	-1.43
23	736.40	921.80	1078.73
	736.33	921.37	1078.38
	0.07	0.43	0.35
24	987.00	860.15	1080.05
	986.84	859.78	1078.82
	0.16	0.37	1.23
25	734.94	859.65	1078.72
	734.96	859.36	1078.55
	-0.02	0.29	0.17
26	985.34	796.66	1079.88
	985.28	797.08	1080.30
	0.06	-0.42	-0.42

Table F.4: Results for Calibration Frame Using the Collinearity Equations Without Additional Parameters

point number	X (mm)	Y (mm)	Z (mm)
format	measured coordinate calculated coordinate difference		
Non-Control Points			
6	1005.45	929.09	999.39
	1006.25	928.57	1000.24
	-0.80	0.52	-0.85
27	737.53	795.97	1078.76
	738.11	796.25	1076.95
	-0.58	-0.28	1.81
Control Points			
1	1003.34	1002.14	999.45
	1003.04	1000.38	998.78
	0.30	1.76	0.67
2	929.50	1002.96	999.52
	929.85	1002.38	998.77
	-0.35	0.58	0.75
3	859.58	1003.54	999.51
	859.62	1003.26	998.64
	-0.04	0.28	0.87
4	790.61	1003.23	999.48
	790.26	1003.08	999.59
	0.35	0.15	-0.11
5	716.38	1002.48	999.20
	716.58	1001.25	999.68
	-0.20	1.23	-0.48
7	715.70	928.43	999.00
	714.90	928.34	998.85
	0.80	0.09	0.15
8	1005.46	863.20	999.29
	1006.41	863.03	999.16
	-0.95	0.17	0.13
9	715.81	860.50	998.61
	714.67	860.40	997.98
	1.14	0.10	0.63
10	1005.85	790.26	998.90
	1006.25	790.31	998.58
	-0.40	-0.05	0.32
11	716.10	791.38	998.26
	716.24	791.42	997.96
	-0.14	-0.04	0.30

12	1002.38	718.53	996.95
	1002.35	718.80	998.02
	0.03	-0.27	-1.07
13	930.91	715.84	997.95
	931.02	715.53	996.45
	-0.11	0.31	1.50
14	856.98	715.04	997.73
	856.86	714.52	996.06
	0.12	0.52	1.67
15	786.88	715.43	997.33
	786.29	715.42	997.66
	0.59	0.01	-0.33
16	717.62	718.52	997.14
	718.12	719.38	997.51
	-0.50	-0.86	-0.37
17	979.59	981.46	1079.60
	980.07	981.66	1080.96
	-0.48	-0.20	-1.36
18	921.48	983.88	1079.29
	922.09	984.61	1080.96
	-0.61	-0.73	-1.67
19	860.75	984.55	1079.15
	860.83	985.36	1080.13
	-0.08	-0.81	-0.98
20	800.56	981.18	1078.98
	800.09	981.61	1079.25
	0.47	-0.43	-0.27
21	739.71	976.59	1078.60
	739.80	976.57	1078.80
	-0.09	0.02	-0.20
22	984.21	924.80	1080.05
	984.96	926.11	1082.14
	-0.75	-1.31	-2.09
23	736.40	921.80	1078.73
	736.22	921.39	1077.81
	0.18	0.41	0.92
24	987.00	860.15	1080.05
	987.09	860.45	1079.25
	-0.09	-0.30	0.80
25	734.94	859.65	1078.72
	734.95	859.48	1077.29
	-0.01	0.17	1.43
26	985.34	796.66	1079.88
	985.24	798.19	1081.03
	0.10	-1.53	-1.15

Table F.5: Results for Calibration Frame Using the DLT Without Additional Parameters

point number	X (mm)	Y (mm)	Z (mm)
format	measured coordinate calculated coordinate difference		
Non-Control Points			
6	1005.45	929.09	999.39
	1005.37	928.83	999.10
	0.08	0.26	0.29
27	737.53	795.97	1078.76
	737.29	795.78	1077.62
	0.24	0.19	1.14
Control Points			
1	1003.34	1002.14	999.45
	1003.07	1000.49	999.22
	0.27	1.65	0.23
2	929.50	1002.96	999.52
	929.77	1002.47	999.24
	-0.27	0.49	0.28
3	859.58	1003.54	999.51
	859.44	1003.33	999.14
	0.14	0.21	0.37
4	790.61	1003.23	999.48
	790.00	1003.12	1000.11
	0.61	0.11	-0.63
5	716.38	1002.48	999.20
	716.22	1001.28	1000.23
	0.16	1.20	-1.03
7	715.70	928.43	999.00
	714.54	928.53	999.37
	1.16	-0.10	-0.37
8	1005.46	863.20	999.29
	1006.44	863.06	998.53
	-0.98	0.14	0.76
9	715.81	860.50	998.61
	714.29	860.74	998.59
	1.52	-0.24	0.02
10	1005.85	790.26	998.90
	1006.33	790.49	997.62
	-0.48	-0.23	1.28
11	716.10	791.38	998.26
	715.80	791.88	998.76
	0.30	-0.50	-0.50

12	1002.38	718.53	996.95
	1002.50	719.19	996.93
	-0.12	-0.66	0.02
13	930.91	715.84	997.95
	931.19	715.97	995.96
	-0.28	-0.13	1.99
14	856.98	715.04	997.73
	856.92	714.99	996.18
	0.06	0.05	1.55
15	786.88	715.43	997.33
	786.11	715.88	998.31
	0.77	-0.45	-0.98
16	717.62	718.52	997.14
	717.60	719.86	998.64
	0.02	-1.34	-1.50
17	979.59	981.46	1079.60
	980.44	981.44	1080.92
	-0.85	0.02	-1.32
18	921.48	983.88	1079.29
	922.21	984.41	1081.03
	-0.73	-0.53	-1.74
19	860.75	984.55	1079.15
	860.67	985.17	1080.28
	0.08	-0.62	-1.13
20	800.56	981.18	1078.98
	799.66	981.42	1079.47
	0.90	-0.24	-0.49
21	739.71	976.59	1078.60
	739.07	976.38	1079.13
	0.64	0.21	-0.53
22	984.21	924.80	1080.05
	985.45	925.50	1081.73
	-1.24	-0.70	-1.68
23	736.40	921.80	1078.73
	735.48	921.13	1078.14
	0.92	0.67	0.59
24	987.00	860.15	1080.05
	987.60	859.70	1078.49
	-0.60	0.45	1.56
25	734.94	859.65	1078.72
	734.19	859.13	1077.70
	0.75	0.52	1.02
26	985.34	796.66	1079.88
	985.79	797.24	1079.88
	-0.45	-0.58	-0.00

Table F.6: Results for Factory Scene 1 Using Smith's Resection Followed by Schmid's Resection.

point number	X (mm)	Y (mm)	Z (mm)
format	measured coordinate calculated coordinate difference		
Control Points			
39	208603.0	14965.6	-78327.5
	208602.0	14965.7	-78326.7
	1	-0.1	-0.8
40	207978.0	14560.9	-75885.7
	207978.5	14563.2	-75881.4
	-0.5	-2.3	-4.3
41	206700.0	15620.6	-75519.6
	206703.5	15619.7	-75537.1
	-3.5	0.9	17.5
42	205824.0	14237.9	-76853.2
	205819.9	14233.3	-76849.0
	4.1	4.6	-4.2
43	206159.0	17002.1	-77610.5
	206155.6	17005.0	-77607.7
	3.4	-2.9	-2.8
44	207694.0	17142.9	-76923.6
	207694.8	17139.8	-76923.3
	-0.8	3.1	-0.3
45	207299.0	17208.1	-75546.0
	207301.1	17211.2	-75536.9
	-2.1	-3.1	-9.1
46	206214.0	15422.9	-78003.0
	206215.6	15422.7	-78000.3
	-1.6	0.2	-2.7

Table F.7: Results for Factory Scene 1 Using Bundle Adjustment.

point number	X (mm)	Y (mm)	Z (mm)
Control Points			
format	measured coordinate calculated coordinate difference		
39	208602.0 208602.0 0.0	14966.1 14965.7 0.4	-78325.6 -78326.7 1.1
40	207977.0 207978.5 -1.5	14563.4 14563.2 0.2	-75892.5 -75881.4 -11.1
41	206701.0 206703.5 -2.5	15621.1 15619.7 1.4	-75521.3 -75537.1 15.8
42	205821.0 205819.9 1.1	14233.0 14233.3 -0.3	-76847.1 -76849.0 1.9
43	206160.0 206155.6 4.4	17003.7 17005.0 -1.3	-77612.2 -77607.7 -4.5
44	207694.0 207694.8 -0.8	17141.2 17139.8 1.4	-76927.2 -76923.3 -3.9
45	207301.0 207301.1 -0.1	17208.8 17211.2 -2.4	-75545.3 -75536.9 -8.4
46	206214.0 206215.6 -1.6	15421.6 15422.7 -1.1	-78001.5 -78000.3 -1.0

Table F.8: Results for Calibration Frame and 30 cm Ruler

point number	X (mm)	Y (mm)	Z (mm)
Non-Control Points			
format	calculated coordinate standard deviation		
m0	827.04 0.03	988.91 0.05	948.87 0.15
m10	827.28 0.04	888.96 0.05	948.04 0.19
m20	827.91 0.04	788.86 0.04	947.62 0.19
m30	828.47 0.03	688.82 0.05	946.79 0.15
m15	827.44 0.04	838.95 0.04	947.59 0.20
Control Points			
format	measured coordinate calculated coordinate difference standard deviation		
1	973.25 973.17 0.08 0.05	972.08 972.04 0.04 0.05	969.48 969.38 0.10 0.17
2	901.63 901.71 -0.08 0.03	972.89 972.88 0.02 0.05	969.55 969.67 -0.12 0.15
4	766.90 766.86 0.05 0.03	973.15 973.13 0.02 0.05	969.51 969.50 0.01 0.15
5	694.90 694.92 -0.02 0.05	972.41 972.52 -0.10 0.05	969.24 969.25 -0.01 0.17
6	975.30 975.31 -0.00 0.04	901.22 901.21 0.01 0.04	969.42 969.39 0.03 0.15
7	694.23 694.24 -0.01 0.04	900.59 900.58 0.01 0.03	969.04 968.95 0.09 0.15

8	975.31	837.31	969.32
	975.31	837.32	969.40
	0.00	-0.01	-0.08
	0.04	0.03	0.16
9	694.34	834.69	968.66
	694.39	834.72	968.58
	-0.05	-0.02	0.08
	0.05	0.03	0.15
10	975.68	766.56	968.94
	975.62	766.65	968.84
	0.06	-0.09	0.10
	0.04	0.03	0.16
11	695.47	767.64	968.14
	695.46	767.69	968.02
	0.02	-0.05	0.12
	0.04	0.03	0.15
12	972.85	696.15	968.12
	972.85	696.07	968.12
	-0.00	0.08	0.01
	0.05	0.05	0.17
13	902.99	694.37	968.03
	903.01	694.25	968.17
	-0.02	0.13	-0.14
	0.03	0.04	0.16
15	763.28	693.97	967.41
	763.28	693.91	967.62
	0.00	0.07	-0.20
	0.04	0.04	0.15
16	696.10	696.97	967.23
	696.17	697.02	967.21
	-0.07	-0.06	0.02
	0.05	0.05	0.16
17	950.21	952.03	1047.23
	950.29	952.10	1047.31
	-0.08	-0.07	-0.08
	0.04	0.05	0.19
18	893.85	954.37	1046.93
	893.91	954.40	1047.05
	-0.07	-0.03	-0.12
	0.03	0.05	0.18
20	776.55	951.76	1046.63
	776.55	951.73	1046.59
	-0.00	0.03	0.04
	0.04	0.05	0.18

	717.53	947.30	1046.25
21	717.41	947.15	1046.45
	0.11	0.15	-0.20
	0.05	0.05	0.19
	954.69	897.07	1047.66
22	954.69	897.05	1047.59
	0.00	0.02	0.07
	0.04	0.04	0.19
	714.32	894.15	1046.39
23	714.30	894.16	1046.43
	0.02	-0.01	-0.04
	0.04	0.04	0.18
	957.40	834.35	1047.66
24	957.43	834.39	1047.58
	-0.03	-0.03	0.08
	0.04	0.03	0.19
	712.90	833.87	1046.38
25	712.90	833.87	1046.32
	-0.01	-0.00	0.06
	0.05	0.03	0.18
	955.79	772.77	1047.50
26	955.75	772.86	1047.42
	0.04	-0.09	0.08
	0.04	0.03	0.19
	715.41	772.10	1046.42
27	715.41	772.14	1046.28
	0.01	-0.04	0.14
	0.04	0.03	0.18
	950.35	716.55	1047.38
28	950.31	716.61	1047.21
	0.04	-0.06	0.17
	0.04	0.04	0.20
	890.80	714.92	1047.10
29	890.77	714.93	1047.19
	0.03	-0.01	-0.09
	0.03	0.04	0.19
	771.70	712.97	1046.48
31	771.75	712.96	1046.49
	-0.05	0.01	-0.01
	0.04	0.04	0.18
	720.35	714.21	1046.20
32	720.34	714.15	1046.20
	0.01	0.06	-0.00
	0.05	0.04	0.19

Table F.9: Results of Test for Factory Scene 2 Using Bundle Adjustment.

point number	X (mm)	Y (mm)	Z (mm)
Non-Control Points			
format	measured coordinate standard deviation		
m7	-83777.0	14345.8	-205122
	6.1	5.3	21
m2	-81313.2	14343.1	-205118
	5.5	5.7	20
m3	-83783.9	13964.9	-205048
	6.3	6.1	22
m4	-83779.2	14290.1	-205060
	6.2	5.5	22
Control Points			
format	measured coordinate calculated coordinate difference standard deviation		
48	-83062.4	17771.6	-202824
	-83063.3	17755.6	-202863
	1.1	16.0	39
	9.8	16.0	47
49	-79993.5	16884.1	-202838
	-79977.5	16892.3	-202796
	16.0	8.2	42
	8.9	7.6	32
50	-81127.4	17277.3	-202073
	-81134.7	17275.7	-202079
	7.3	1.6	6
	8.1	13.7	50
51	-85906.4	14308.4	-206404
	-85907.0	14309.0	-206407
	0.6	1.0	3
	10.0	5.7	19
52	-80413.2	14587.5	-204912
	-80415.5	14586.4	-204908
	2.3	1.1	4
	7.0	5.7	22

53	-82430.9	17472.3	-206397
	-82431.5	17471.7	-206399
	0.6	0.6	2
	4.6	6.6	17
54	-84247.1	15376.5	-204839
	-84238.2	15378.1	-204846
	8.9	1.6	7
	6.8	4.8	23