12-2008

Development of a Mathematical Model for 3D Reconstruction of Target Objects by Photogrammetry

Keith F. Blonquist
Utah State University

DEVELOPMENT OF A MATHEMATICAL MODEL FOR 3D RECONSTRUCTION OF TARGET OBJECTS BY PHOTOGRAMMETRY

by

Keith F. Blonquist

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Civil and Environmental Engineering

Approved:

Dr. Robert T. Pack Major Professor	Dr. David G. Tarboton Committee Member
Dr. Kevin C. Womack Committee Member	Dr. Christopher M. U. Neale Committee Member

Dr. Byron R. Burnham
Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2008

ABSTRACT

Development of a Mathematical Model for 3D Reconstruction of

Target Objects by Photogrammetry

by

Keith F. Blonquist, Master of Science

Utah State University, 2008

Major Professor: Dr. Robert T. Pack
Department: Civil and Environmental Engineering

This thesis outlines the development of a mathematical model which can be used to perform 3D reconstruction of a target object from surveillance images. 3D reconstruction is a common procedure in photogrammetry, but performing 3D reconstruction from surveillance images can be more difficult than typical photogrammetry applications.

Surveillance images are generally captured in an unsystematic manner because there is no control over the target that is being photographed. Surveillance images can have a wide variety of fields of view, are often captured with uncalibrated cameras, and typically the targets are objects for which there is no other *a priori* information. For these reasons, performing 3D reconstruction from surveillance images may not be possible using standard photogrammetric methods, especially when the angular fields of view of the images are rather narrow.

Several alternative methods and algorithms have been developed in photogrammetry to handle some of the complications mentioned above. For example, close-range photogrammetry methods are designed to deal with situations where images are captured from varying and random aspects. However, the majority of these models were not designed to accommodate images with a narrow angular field of view. In satellite imagery, sensor models have been created which are well-suited for narrow angular fields of view, but these models generally assume images that were captured in a systematic manner with available ground control information. Hence, existing models and methods may not be adequate to perform 3D reconstruction from surveillance images in all situations.

The model developed herein is a robust model based on principles from close-range photogrammetry, satellite imagery, and computer vision. Previous work has been relied upon, and routines from several areas have been tied together to form a comprehensive algorithm that is capable of accurate 3D reconstruction in a wide variety of circumstances. The flexibility and precision of the model are demonstrated using several sets of actual surveillance images and a series of synthetic images.

(188 pages)

ACKNOWLEDGMENTS

I wish to thank my family for their encouragement and support. To a great extent, I attribute my skill in engineering to my father, who has always been somewhat of a "fix-it" man—a true problem solver. I also owe a great debt to my parents for their work ethic and knack for teaching me responsibility at a young age; and for buying me a lot of Legos to encourage my creative interests in engineering. Along with my family, I would like to recognize my extended family and friends.

I wish to thank my major professor, Dr. Robert Pack, for taking a chance on hiring me after only a short interview. I also thank him for his continued guidance and suggestions. Without his help this research and thesis would not have been possible.

I would like to thank the other members of my committee as well, Dr. Neale, Dr. Tarboton, and Dr. Womack, for their suggestions and insight. Their help has improved my work significantly.

I wish to thank NAVAIR for their financial contributions to this project and their continued interest in my research. Thanks as well to Dean Cook at NAVAIR.

Finally, I wish to thank my co-workers: Dan, Mike (both Mike L. and Mike M.), John, Ren, Tiffany, Chris, Rob (both Rob H. and Rob G.), Paul (both Paul I. and Paul D.), Ky, Adam, Shuo, Brandon, Ken, Brad (both Brad P. and Brad L.), Randy, Jed, Bretton, Scott, and Rees. I also apologize if there is anyone that has been left out.

Keith F. Blonquist

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

NOTATION & ACRONYMS

x, y	image coordinates
X, Y, Z	3D object coordinates
Xc, Yc, Zc	Perspective center of a camera
f	Principal distance of a camera
xo, *yo*	Principal point of a camera
ω, φ, κ	Rotation angles of a camera
$m_{11}, m_{12}, ...$	Rotation matrix terms
r_x, r_y	Image residuals
$\sigma_X, \sigma_Y, \sigma_Z$	1-sigma errors
x', y'	Back-projected image coordinates
$A_1, A_2, ..., A_8$	Affine projection parameters
s	Parallel projection scale
$\Delta x, \Delta y$	Parallel projection shift terms
$\hat{x}, \hat{y}$	Shifted and scaled image coordinates
X', Y', Z'	Calculated 3D coordinates
xp, *yp*	Perspective-corrected image coordinates
$\theta, \psi, \varepsilon$	Alternate convention of rotation angles
d	Distance from target to a camera
Xo, *Yo*, *Zo*	Point on target toward which a camera is pointed

DLT	Direct Linear Transformation
EOP	Exterior Orientation Parameters
IOP	Interior Orientation Parameters

CHAPTER 1

PRIMER

1.1 Problem Statement

Photogrammetry is defined as "the art, science, and technology of obtaining reliable information about physical objects and the environment through processes of recording, measuring, and interpreting photographic images…" (Wolf and Dewitt, 2000) One common application of photogrammetry is determining the 3D geometry of a target object from images of the target object. This process is typically referred to as 3D reconstruction.

The process of 3D reconstruction is important for many applications because it allows one to accurately determine the 3D geometry of an object without having to come in direct contact with the object. This is used to determine the topography of landscapes from aerial images and satellite imagery, to measure the movement of unstable hill slopes from a distance, to make precise measurements in industry and production, to document crime scenes and archeological or historical sites, and for various other purposes.

Most of the above mentioned applications involve 3D reconstruction of objects that can also be modeled and measured by other means. For example, measurements made by surveying can assist in determining topography and monitoring hill slopes, and direct measurements can generally be made in industry, production, and other sites. Also, quite often the imagery for these applications is captured in a somewhat controlled environment using calibrated sensors.

There has been a great deal of research and development in photogrammetry in order to perform 3D reconstruction. This research and development has led to several different algorithms and methods for 3D reconstruction. The primary method is the bundle triangulation algorithm, which will be discussed in this chapter. Other common photogrammetry algorithms used to assist in 3D reconstruction are the relative orientation of two images, and the Direct Linear Transformation (DLT), which are also discussed later in this chapter.

One interesting application, and the subject of this thesis, is photogrammetric 3D reconstruction using surveillance images. There are cases where it is desired to determine the 3D geometry of a target object for which there is no *a priori* information available, except surveillance imagery. Because surveillance involves capturing images without control over the target object and without direct access to the target object, 3D reconstruction from surveillance images poses some difficulties that are not encountered in some of the applications mentioned above. Sets of surveillance images generally have very diverse properties and will often be collected using uncalibrated cameras, the images will likely not be captured in any systematic fashion, and no additional information about the target will generally be available.

There are some limitations of the photogrammetry methods currently in use. Some of these shortcomings become apparent when trying to perform 3D reconstruction using surveillance images. Difficulty when using surveillance images generally arises from four key factors:

1) The surveillance images may have been captured from various aspects and angles and from different locations with respect to the target. There is generally no systematic manner in which the images were captured.

2) The images may have been captured at various distances from the target and therefore have a wide variety of angular fields of view. A narrow angular field of view typically leads to difficulty.

3) The images may have been captured with cameras and lenses which are uncalibrated or possibly from imaging systems for which the interior orientation parameters (IOP) are unknown.

4) The target object will most likely be an object for which there is no *a priori* knowledge available. The only accessible information will generally be the surveillance images alone.

This large diversity of image information and lack of target information causes problems when using many of the standard photogrammetric algorithms. The purpose of this thesis is to develop a mathematical model and algorithm that facilitates the process of 3D reconstruction from surveillance images having very diverse properties. The desired result is a photogrammetric model which can perform accurate 3D reconstruction in a variety of circumstances and from widely varying surveillance image sets.

This chapter will outline some of the fundamental principles of photogrammetry, and outline some of the standard methods currently in use in photogrammetry. Next, some of the specific difficulties that arise when applying standard photogrammetric methods to surveillance imagery will be discussed. Finally, some other methods found in research in different fields of study will be discussed.

1.2 Introduction to Photogrammetry

An introduction to some of the fundamental principles and topics in photogrammetry is given here. The perspective projection, collinearity equations, relative orientation of two images, bundle triangulation algorithm, and the Direct Linear Transformation (DLT) are discussed.

1.2.1 Perspective projection

An image of an object is a 2D representation of a 3D object. The mathematical model which relates the position of an object in 3D space to its position on a 2D image is called a projection. Most standard photogrammetric algorithms are based on the perspective projection. The perspective projection approximates the imaging process and it is sometimes referred to as the pin-hole camera model.

The perspective projection models the sensor as a plane (referred to as the image plane) and a point (referred to as the perspective center). It is assumed that a point on the target in 3D space, its corresponding point on the image plane, and the perspective center all lie on a straight line. Or in other words, it is assumed that all light rays from the target pass in a straight line from the target, through the perspective center to the image plane. Figure 1.1 shows this geometry.

The distance from the image plane to the perspective center is referred to as the principal distance. The image plane shown in Figure 1.1 captures a negative image, or mirror image of the target object. For this reason it is generally referred to as the negative image plane. It is often convenient to consider another image plane, known as the positive image plane.

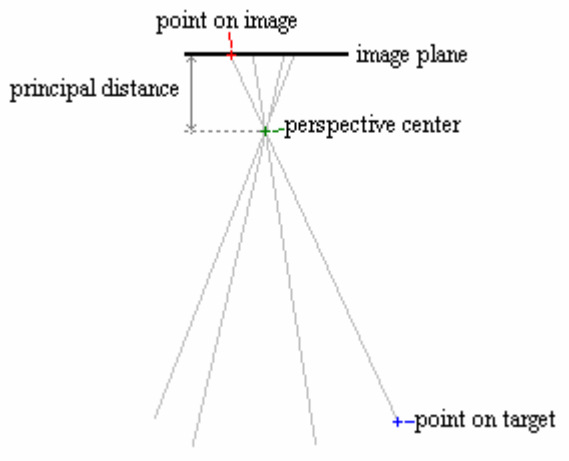

Figure 1.1. Geometry of the perspective projection.

The positive image plane and negative image plane are parallel to each other. The focal axis is the line perpendicular to the image planes which passes through the perspective center. The point where the focal axis intersects the positive image plane is known as the principal point of an image. Figure 1.2 shows the negative image plane and the positive image plane.

While the negative image plane is where the negative of an image is physically captured in a camera, when the negative is developed it produces an image as though it were captured on the positive image plane. In the discussion, figures, and equations that follow the positive image plane will be referred to and the negative image plane will be disregarded. In the remaining discussion "image plane" will refer to the positive image plane. Figure 1.3 is a 3D depiction of the perspective projection.

As shown in Figure 1.3, the target point in 3D space (X, Y, Z), the point on the image (x,y), and the perspective center (Xc, Yc, Zc) all lie on a straight line. The principal distance is denoted by f and the principal point of the image is denoted by (xo, yo). The **XYZ** coordinate system is the coordinate system for the 3D points (X, Y, Z). The xyz

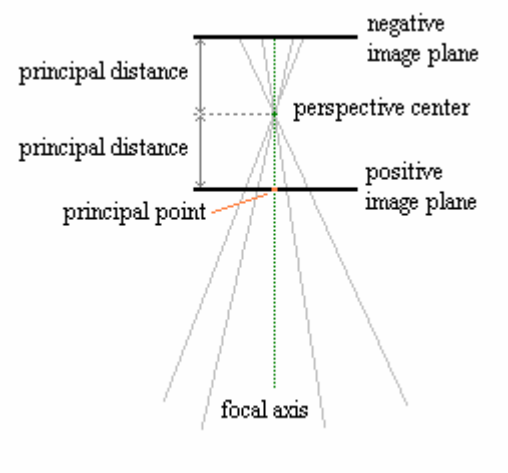

Figure 1.2. Positive image plane and negative image plane.

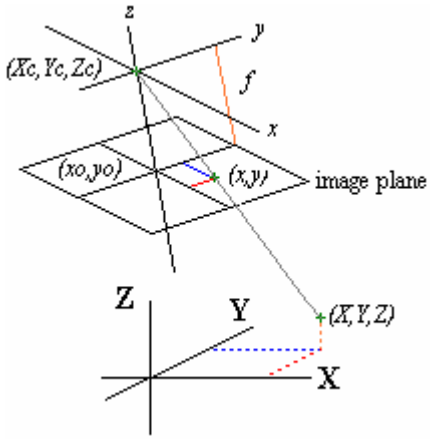

Figure 1.3. 3D depiction of the perspective projection.

coordinate system is the coordinate system of the image; the *z*-axis is the focal axis. The 2D coordinate system on the image plane (the coordinate system in which image coordinates are measured) has axes parallel to the *x*- and *y*-axes of the image coordinate system, and the focal axis passes through the image plane at the principal point (*xo*, *yo*).

There is a rotation of the image coordinate system (*xyz*) with respect to the 3D coordinate system (**XYZ**). This rotation can be described by three ordered rotations about the **X**-axis, **Y**-axis, and **Z**-axis, respectively. These rotations will be denoted by $(\omega, \varphi, \kappa)$ which is standard in photogrammetry. This rotation can also be described by a 3 x 3 rotation matrix. (For additional information on rotation matrices see Appendix A.1.) The six parameters $(\omega, \varphi, \kappa, Xc, Yc, Zc)$ are referred to as the exterior orientation parameters (EOP) because they describe the position and orientation of the camera in 3D space. The three parameters (*f*, *xo*, *yo*) are referred to as the interior orientation parameters (IOP) because they describe geometry within the camera.

1.2.2 Collinearity equations and the coplanarity equation

The collinearity equations (1.1) are the fundamental equations describing the perspective projection (Wolf and Dewitt, 2000). They express the 2D image coordinates (*x*, *y*) of a point with 3D coordinates (*X*, *Y*, *Z*) as a function of the EOP (perspective center coordinates and three rotation angles) of an image denoted by $(Xc, Yc, Zc, \omega, \varphi, \kappa)$ and the IOP (principal distance and principal point coordinates) of an image denoted by (f, xo, yo):

$$
\begin{aligned}
x &= xo - f\left(\frac{m_{11}(X - Xc) + m_{12}(Y - Yc) + m_{13}(Z - Zc)}{m_{31}(X - Xc) + m_{32}(Y - Yc) + m_{33}(Z - Zc)}\right) \\
y &= yo - f\left(\frac{m_{21}(X - Xc) + m_{22}(Y - Yc) + m_{23}(Z - Zc)}{m_{31}(X - Xc) + m_{32}(Y - Yc) + m_{33}(Z - Zc)}\right)
\end{aligned}
\qquad (1.1)
$$

In these equations (1.1), the *m* terms are from the rotation matrix which describes the rotation of the image coordinate system with respect to the 3D coordinate system.

These nine matrix terms can be expressed as a function of the three ordered rotation angles $(\omega, \varphi, \kappa)$; the explicit equations for these terms are given in the appendix (equation A.6). The collinearity equations are used extensively in photogrammetry and form the basis of the bundle triangulation algorithm.

The coplanarity equation is an extension of the collinearity equations when two images are considered. The coplanarity condition states that for a given point in 3D space (*X*, *Y*, *Z*), the point lies on a common plane with the two perspective centers of the two images $(Xc, Yc, Zc)_1$ and $(Xc, Yc, Zc)_2$ and the two image points $(x, y)_1$ and $(x, y)_2$. This is shown in Figure 1.4.

The coplanarity equation (1.2 and 1.3) is expressed as a vector triple product (Mikhail, Bethel, and McGlone, 2001). The three vectors in the triple product are the baseline vector which is the vector from one perspective center to the other, and then one vector for each image which points from the perspective center to the image point. These vectors are expressed mathematically as:

$$\begin{bmatrix} b_X \\ b_Y \\ b_Z \end{bmatrix} = \begin{bmatrix} Xc_2 - Xc_1 \\ Yc_2 - Yc_1 \\ Zc_2 - Zc_1 \end{bmatrix} \qquad \begin{bmatrix} u_1 \\ v_1 \\ w_1 \end{bmatrix} = M_1^T \begin{bmatrix} x - xo \\ y - yo \\ -f \end{bmatrix}_1 \qquad \begin{bmatrix} u_2 \\ v_2 \\ w_2 \end{bmatrix} = M_2^T \begin{bmatrix} x - xo \\ y - yo \\ -f \end{bmatrix}_2 \tag{1.2}$$

where for each image (*x*, *y*) is the image point, *f* is the principal distance, *xo* and *yo* are the coordinates of the principal point, and *M* is the rotation matrix. The coplanarity equation is then:

$$\begin{vmatrix} b_X & b_Y & b_Z \\ u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \end{vmatrix} = 0 \tag{1.3}$$

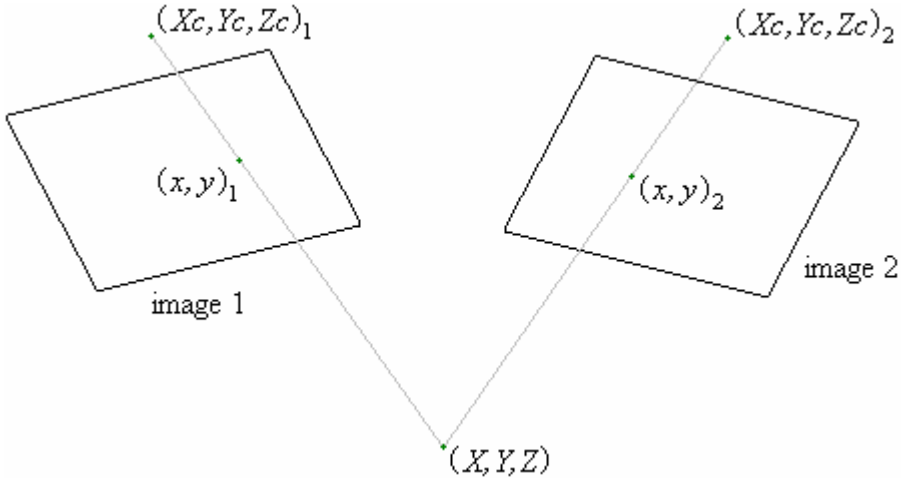

Figure 1.4. The coplanarity condition.

The coplanarity equation can also be rearranged to form the *essential* matrix *A* (Mikhail, Bethel, and McGlone, 2001):

$$\begin{bmatrix} x - xo & y - yo & -f \end{bmatrix}_1 \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} x - xo \\ y - yo \\ -f \end{bmatrix}_2 = 0 \qquad (1.4)$$

These two forms of the coplanarity equation (1.2 – 1.4) will be used later in the discussion of methods for the relative orientation of two images.

1.2.3 Bundle triangulation algorithm

The bundle triangulation algorithm (also known as bundle block adjustment, or multi-image triangulation) is a standard procedure in photogrammetry. It is the primary algorithm in photogrammetry for 3D reconstruction. It is a robust algorithm capable of simultaneously solving for the EOP, the IOP, and the 3D coordinates of target points. It also provides statistical information regarding the precision of the calculated solution. The standard bundle triangulation algorithm is based on perspective projection, and is derived from the collinearity equations.

The primary input for the bundle triangulation algorithm is a set of point correspondences across the images which are being analyzed. A point correspondence is when a particular point on the target object in 3D space is located in several images—i.e. the image coordinates across several images are recorded for a given point on the target. Figure 1.5 shows point correspondences across two images. The bundle triangulation algorithm requires that a set of point correspondences has been collected.

The second major input for the bundle triangulation algorithm is a set of initial approximations of all unknowns. The bundle triangulation algorithm is an iterative procedure that begins with the initial approximations and updates these values with each iteration. The initial approximations must be fairly close to the actual values in order for the bundle triangulation algorithm to converge to an accurate solution. These initial approximations can be obtained in several different ways. As will be seen later, obtaining reliable initial approximations can prove to be the most problematic step of the 3D reconstruction process.

Given a set of point correspondences and a set of satisfactory initial approximations, the bundle triangulation algorithm can solve for the EOP and the IOP of the images and the 3D geometry of the target points, and give statistical information regarding these values. This makes the bundle triangulation algorithm a valuable tool in photogrammetry. Two special cases of the bundle triangulation algorithm based on some of the assumptions for surveillance imagery are outlined in the appendix. Also, for a more comprehensive discussion of the bundle triangulation algorithm, see (Luhmann et al., 2006; Wolf and Dewitt, 2000; Mikhail, Bethel, and McGlone, 2001).

Figure 1.5. Point correspondences across two images.

1.2.4 Relative orientation of two images

Relative orientation is the process of finding the rotation and translation of one image with respect to another image. There are several methods currently in use in photogrammetry to solve for the relative orientation of two images. If the relative

orientation of two images is solved for, the two images can be used to perform 3D reconstruction.

Generally in relative orientation, the EOP of the first image (three rotation angles and perspective center) are "fixed" along with one of the coordinates of the perspective center of the second image. This leaves five unknowns—the three rotation angles of the second image and two of the coordinates of the perspective center of the second image. Also, it is generally assumed that the IOP (principal distance and principal point) of the images are known.

Two standard methods for finding the relative orientation of two images will be discussed here. Both of these methods are based on the coplanarity equation.

The first method to solve for the relative orientation of two images is analytical relative orientation based on the coplanarity equation. The coplanarity equation is linearized in terms of the five unknowns (three rotation angles and two perspective center coordinates for the second image); then an iterative least-squares method is used to solve for the five unknowns. This method requires a set of point correspondences between the two images and initial approximations of the five unknowns. The specific equations for this method and a more in-depth discussion can be found in (Mikhail, Bethel, and McGlone, 2001).

The second method for the relative orientation of two images comes from research in the field of computer vision and is also based on the coplanarity equation. Using a series of point correspondences between the two images, the terms of the *essential* matrix (equation 1.4) can be solved for. Once the terms of the matrix have been obtained, the five unknown orientation parameters of the second image can be derived

from these matrix terms. The equations for this method and a more in-depth discussion are found in (Pan, Brooks, and Newsam, 1995; Pan, Huynh, and Hamlyn, 1995).

1.2.5 Direct linear transformation

The Direct Linear Transformation (DLT) (Abdel-Aziz and Karara, 1971) is a standard photogrammetry method to solve for the EOP and the IOP of a single image relative to a set of 3D coordinates. This method requires that the 3D coordinates of several points on the target are known and are visible in the image.

The DLT can be derived from the collinearity equations. The EOP and the IOP of the camera are described by 11 *L*-terms. The *L*-terms can be solved for from a set of control points with known XYZ coordinates. The set up is as follows:

$$\begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ \dots \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 & -x_1Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1X_1 & -y_1Y_1 & -y_1Z_1 \\ X_2 & Y_2 & Z_2 & 1 & 0 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 & -x_2Z_2 \\ 0 & 0 & 0 & 0 & X_2 & Y_2 & Z_2 & 1 & -y_2X_2 & -y_2Y_2 & -y_2Z_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \\ L_5 \\ L_6 \\ L_7 \\ L_8 \\ L_9 \\ L_{10} \\ L_{11} \end{bmatrix} \quad (1.5)$$

In equation (1.5), (x_1, y_1) represent the image coordinates of the target point (X_1, Y_1, Z_1) etc… The *L* terms can be solved for by inverting the matrix on the right side. Once the *L* terms have been derived from the control information, the EOP

$(Xc, Yc, Zc, \omega, \varphi, \kappa)$ and IOP (*f*, *xo*, *yo*) can then be solved for from the *L* terms. A more in depth discussion of the DLT can be found in (Mikhail, Bethel, and McGlone, 2001; Luhmann et al., 2006).

1.3 Surveillance Imagery

While the standard photogrammetry techniques outlined above are sufficient to perform 3D reconstruction for many applications, they are not suitable to produce reliable 3D models of target objects in all situations. These methods are based on certain assumptions which may not always be applicable to surveillance imagery. Some of the specific challenges faced when using surveillance imagery will be discussed in detail.

1.3.1 Narrow angular field of view

As noted above, the great majority of photogrammetry algorithms are based on perspective projection. This model works well for many imaging systems. However, there is one primary drawback to the perspective projection—as the distance to the target being imaged increases, the angular field of view of the relevant portion of the resulting image decreases. This is shown in Figure 1.6.

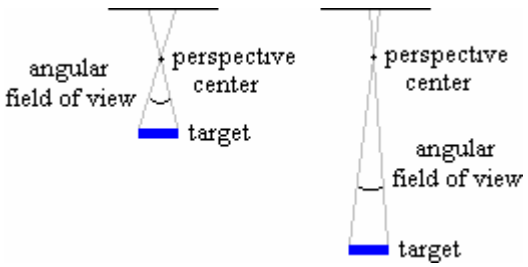

Figure 1.6. Decreasing angular field of view with increasing distance to target.

As the distance to the target increases and the angular field of view decreases, the perspective projection is susceptible to instability and low precision. Note that as the distance to the target increases, the angular field of view becomes narrower and the light rays from the target become more parallel.

Another demonstration of this is shown by the synthetic images of a unit cube in Figure 1.7. The image on the left is a synthetic image of a unit cube generated as though the sensor were 3 units away. The image on the right is a synthetic image of the same unit cube generated as though the sensor were 15 units away (the distance to target and principal distance have been multiplied 5x). Note that the edges of the cube appear nearly parallel in the image generated at a further distance. This is the primary effect of a narrowing angular field of view. The two images are shown superimposed in Figure 1.8.

In situations where the angular field of view of an image (or the relevant portion of an image) is too narrow, the algorithms based on perspective projection commonly fail to produce satisfactory results due to numerical instability and near linear dependencies between camera orientation parameters (both EOP and IOP) (Gruen and Li, 2003; Grodecki and Dial, 2003). In particular, the bundle triangulation algorithm, relative orientation algorithms, and the DLT are prone to instability in circumstances of narrow angular field of view. In these situations, other methods may need to be employed in order to reach an accurate solution.

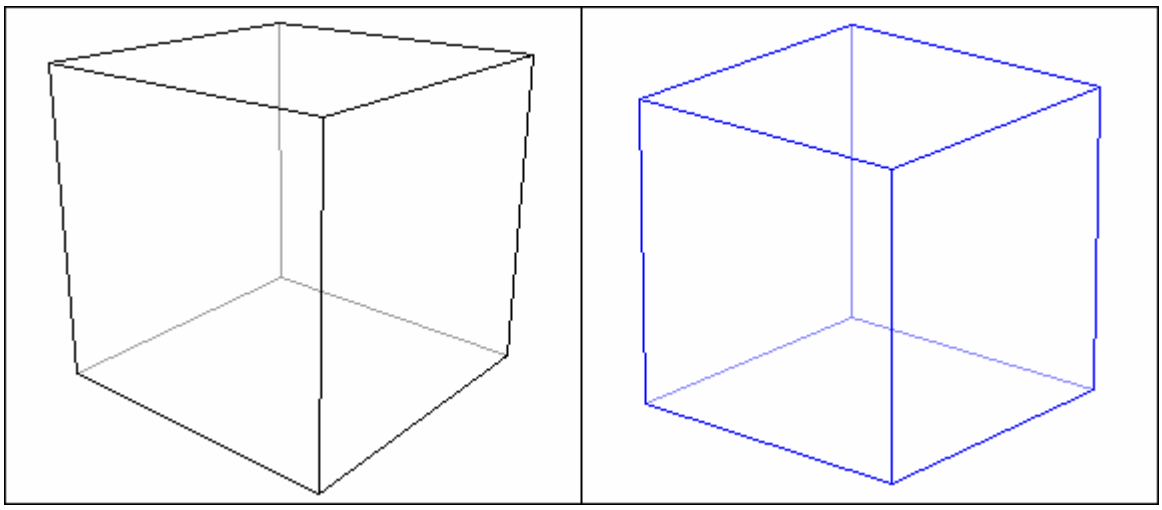

Figure 1.7. Synthetic images of a unit cube.

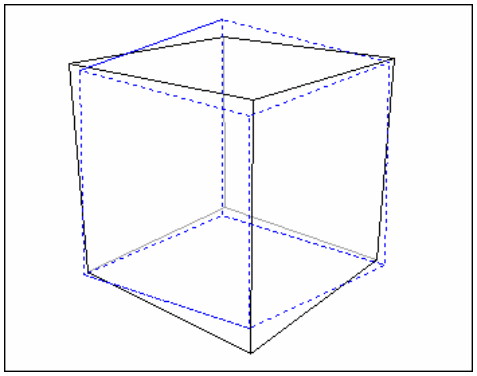

Figure 1.8. Synthetic cube images superimposed.

1.3.2 Lack of ground control and sensor orientation information

In most photogrammetry applications there is spatial information available which describes the geometry of the target or the position and orientation of the sensor with respect to the target. For example, in most aerial photogrammetry and satellite imagery applications there are ground control points with known coordinates visible in the images. Or, coordinates of points which are visible in the images can be obtained

through surveying. Also, most aerial photography aircraft and remote sensing satellites are outfitted with GPS or inertial navigation equipment which describes the position and orientation of the sensor at the time the image was captured. In many other photogrammetric applications, such as monitoring hill slope movement, or in production and industry, the orientation of images can be measured by other means when the images are captured. Also, in many of these cases, the general geometry of the object being imaged can be measured by other means to provide an initial approximation of its 3D geometry.

Most photogrammetric algorithms are dependent to some extent on ground control and sensor orientation information. Specifically, the bundle triangulation algorithm requires initial approximations of the EOP, the IOP, and target coordinates; the analytical relative orientation algorithm requires an initial approximation of the EOP and the IOP; and the DLT requires that the 3D coordinates of several points are known. For the bundle triangulation algorithm and the analytical relative orientation algorithm, these initial approximations are generally derived from the ground control and sensor orientation data. The measured coordinates required by the DLT are also generally provided by ground control.

As noted above, in most surveillance applications, the images are captured without control over the target object, without direct access to the target object, and from variable locations. Hence, there will be no control information available for the images or the target object. When this information is not available, as will be the case with most surveillance imagery, initial approximations for the bundle triangulation algorithm and analytical orientation algorithm will have to be derived in another manner. Also, in order

to use the DLT, 3D coordinates of several points on the target will have to be obtained differently.

For this thesis it is desired to develop a robust 3D reconstruction algorithm that arrives at an accurate solution in the absence of ground control information and sensor orientation information. Thus, additional methods or algorithms will have to be developed to bypass this problem.

1.3.3 Camera calibration

One important step in most photogrammetry applications is the calibration of the camera/lens. This calibration gives information about the sensor, such as the principal distance, and location of the principal point (the IOP). It may also be desired to solve for lens distortion parameters. There are many calibration methods available to find the IOP for a given camera/lens, most of which are done in a laboratory. In some cases, however, one may wish to work with images captured with an unknown or uncalibrated sensor. For example, a set of surveillance images may have been captured with uncalibrated or unknown sensors. Hence, it is desired to develop a model which is not vulnerable to camera calibration.

In the case where an uncalibrated sensor has been used it is necessary to either approximate the IOP, or solve for them during the 3D reconstruction process. The IOP can be included as unknowns and solved for in the bundle triangulation algorithm (known as self-calibration). However, in the case of a narrow field of view sensor this may not be possible due to the numerical instabilities described in section 1.3.1. Also, camera calibration often requires that image coordinates are collected over the whole of the field

of view of an image. In many surveillance images, the target will not fill a sufficient portion of the field of view in order to perform an accurate calibration. Hence, camera calibration can introduce additional difficulties.

1.3.4 Varying aspects

Many photogrammetric algorithms were developed for determining the topography of the ground from aerial photographs. These aerial photographs are generally captured as an airplane or satellite travels in a straight path over the target (the ground) with the camera pointed downward. This type of photogrammetry is referred to as topographic photogrammetry. For other applications, images are captured at various aspects around the target. This is generally referred to as non-topographic or close-range photogrammetry. Figure 1.9 shows the difference between image collection in topographic photogrammetry and non-topographic photogrammetry.

The uniformity in the manner of image collection in topographic photogrammetry is utilized in many of the photogrammetry applications and algorithms in use. For example, this regular method of capturing images can provide very good initial approximations of the EOP for both the bundle triangulation algorithm and the

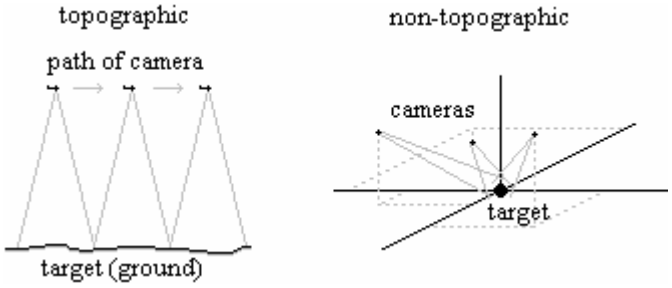

Figure 1.9. Difference in imaging geometries between topographic and non-topographic photogrammetry.

analytical relative orientation algorithm. Because the camera remains pointed downward, there is very little rotation between two images, and because the camera is generally aligned with the direction of flight, there is only an appreciable translation in one direction between two images.

Surveillance images will not typically be collected in a uniform manner. They will be captured from various locations, sometimes from moving platforms. Also, in surveillance there is no control over the location of the target object. So any methods in topographic photogrammetry which rely on the above assumptions are generally invalid when using surveillance images.

In cases like surveillance imagery where the images were not collected in a uniform manner, there is often rotation about all three axes and translation in all three directions from one image to another. The field of non-topographic photogrammetry has been developed to handle these situations. Some of the primary algorithms used in non-topographic photogrammetry are analytical relative orientation, bundle triangulation, and the DLT. Note however, that when these algorithms are applied to non-topographic photogrammetry it is often more difficult to obtain initial approximations of the EOP than for topographic photogrammetry.

1.3.5 Summary of complications of surveillance imagery

Surveillance images therefore provide a unique combination of challenges. In particular, there is the unusual mix of close-range photogrammetry with narrow angular field of view. The differing aspects from which the images are captured incline one to rely upon the methods of close-range photogrammetry. However, due to the fact that

images may be captured at great distances from the target and have a narrow angular field of view, it becomes necessary to employ methods from other areas of photogrammetry, such as satellite imagery.

1.4 Research

Several areas of research will be discussed which may provide insight as to how 3D reconstruction can be performed from surveillance imagery. Research in three areas will be discussed: non-topographic photogrammetry, satellite imagery, and computer vision.

1.4.1 Research in non-topographic photogrammetry

The first area of interest is the field of non-topographic, or close-range, photogrammetry. As mentioned before, a large percentage of photogrammetry applications and developments are in the field of topographic (aerial) photogrammetry. The field of non-topographic photogrammetry deals with image sets that were captured from a variety of distances and aspects around the target. This less-structured geometry is the manner in which most surveillance images will be captured. Thus, methods developed for non-topographic applications may be more useful when working with surveillance images. Two specific methods have been researched and will be mentioned here: the orthogonal projection model and single image orientation based on parallel projection.

First, an orthogonal projection model has been developed by Ono and Hattori (2002) in order to measure movement of cliff faces from long distances. This model

makes use of parallel (orthogonal) projection in order to accommodate images with a narrow angular field of view and considers images taken from different aspects. While the model does accommodate non-topographic imaging geometry and images with a narrow angular field of view, the model they present is dependent upon some prior knowledge of the target—specifically this model requires that the coordinates of several control points are known in order to develop initial approximations. For the purposes of using surveillance imagery it is desired to derive a model which does not rely upon ground control or prior knowledge of the target, so the model outlined by Ono and Hattori does not fully solve the problems presented by surveillance images. However, this model provides a good example of the use of parallel projection in a non-topographic setting. This model will be discussed in more detail later and serves as the basis for many of the relations that will be drawn between the parallel projection and perspective projection in sections 3.3 and 3.4.

Second, an algorithm was developed by Kyle (2004) to orient a single image with respect to an object. It serves essentially the same purpose as the DLT, except it is based on parallel projection instead of perspective projection. This algorithm is a good candidate for images that were captured at varying aspects and have a narrow angular field of view. However, this algorithm is dependent upon some prior knowledge of the target and is only valid for one image individually. Thus, the algorithm presented is not adequate alone to solve the problems faced when using several surveillance images. This algorithm, however, does give insight into the parallel projection and will be used later as part of the method to orient surveillance images for 3D reconstruction.

1.4.2 Research in satellite imagery

Another branch of photogrammetry in which research has been conducted is satellite imagery. Satellite imagery is a form of topographic photogrammetry because the sensor travels essentially in a straight line over the target with the sensor pointed downward. Thus, many of the assumptions made in topographic photogrammetry apply to satellite imagery, but will not apply to surveillance images. However, imaging sensors that have been deployed on satellites generally have a very narrow angular field of view. The narrow angular field of view introduces additional problems which are not solved by typical photogrammetry techniques. The surveillance imagery used for this project will benefit from the research which has been done for satellite imagery due to the possibility of images which have a narrow angular field of view.

Morgan (2004) researched the epipolar resampling of linear array scanner scenes using principles of parallel projection. The work by Morgan includes a rigorous mathematical development of parallel projection geometry. However, his work is primarily concerned with topographic photogrammetry and may not be directly applicable to non-topographic applications such as 3D reconstruction from surveillance images. Also, his work assumes that ground control is available. Hence, the work presented by Morgan is not sufficient by itself to solve the problem at hand, but the parallel projection model outlined therein provides insight on the parallel projection.

1.4.3 Research in computer vision

Computer vision is a field in which computers are used to automatically analyze images. Applications include object recognition and navigation, among others. To a

certain degree, research in computer vision has been conducted independent of research in photogrammetry and some work has been duplicated. As a result, there have been some developments made in computer vision which have not been widely known or used in the photogrammetry community. One particular algorithm from computer vision will be discussed here.

Huang and Lee (1989) have detailed an algorithm for use in computer vision which solves for the motion and structure of a target object from three orthographic (parallel) images. Their algorithm assumes that the input images are orthographic images; also, the algorithm they present is limited to exactly three images. Surveillance images which are used for 3D reconstruction will generally not be truly orthographic images, but depending on the field of view, they may be approximately orthographic. Also, it is will often be desired to work with more than, or perhaps less than, three images. Thus, this algorithm does not provide a full solution to the problems introduced by surveillance images. However, this algorithm provides a key starting point as it does not require any additional knowledge about the target—only a set of point correspondences.

CHAPTER 2

TESTS OF STANDARD PHOTOGRAMMETRY TECHNIQUES USING ACTUAL SURVEILLANCE IMAGES

This chapter discusses some tests that were conducted using three sets of actual surveillance images. Standard photogrammetric techniques were used to perform 3D reconstruction to reveal any possible weaknesses of the standard methods.

2.1 Test Procedure

The procedure for the tests will be discussed here. This procedure is a series of steps involving several standard photogrammetry techniques. The two primary inputs that are generally available from surveillance imagery are:

1) a set of point correspondences (x, y) across the images

2) approximations of the IOP (f, xo, yo) for each of the images

The first step is to begin with two of the images and perform relative orientation. Relative orientation can be carried out following the essential matrix method or by analytical relative orientation (these methods are described in section 1.2.4). The essential matrix method requires the set of point correspondences for the two images, and the approximate IOP. Analytical relative orientation requires these two inputs plus one additional input: the approximate relative orientation of the first two images (the approximate EOP). These algorithms result in an initial approximation of the relative orientation of these two images. Next 3D reconstruction is performed using equations (A.44, A.45) in the appendix; this results in initial approximations for the 3D coordinates

of the points on the target. Given these initial approximations of the EOP for two images, the IOP, and 3D point coordinates, the bundle triangulation algorithm is performed to refine the initial approximations. The bundle triangulation algorithm produces 3D coordinates, along with the EOP of the two images, residuals, 1-sigma errors, and back-projected image coordinates. The residuals, 1-sigma errors, and back-projected image coordinates give a measure of the precision of the solution. The flow chart for this procedure is shown in Figure 2.1 (assuming the essential matrix algorithm).

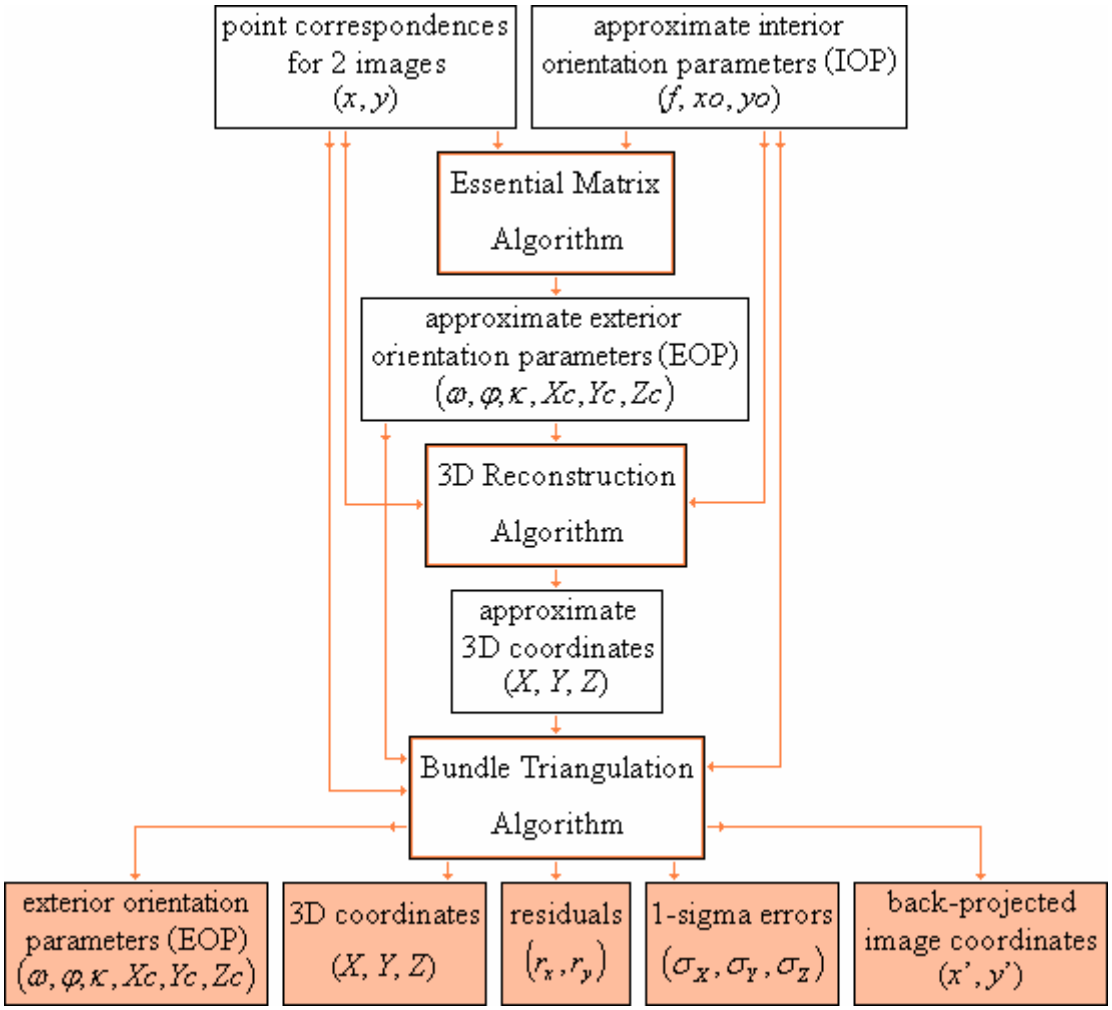

Figure 2.1. Flow chart for relative orientation of two images and bundle triangulation using standard photogrammetry methods.

Next, each remaining image is added in turn to the solution by the following steps: First, the DLT is used to obtain an initial approximation of the EOP—this is possible because initial approximations of the 3D coordinates have already been found. Note that generally when using the DLT actual control point coordinates are used, however because the DLT is being used for approximations only, approximate 3D coordinates can be used. After using the DLT to get an initial approximation of the EOP, the solution is refined using single image resection as explained in the appendix (section A.2.2). Next, the 3D coordinates are re-calculated by equations (A.44, A.45) including the newly oriented image. Then, the bundle triangulation algorithm is performed to refine the solution. The flow chart for this process is shown in Figure 2.2.

By repeating the process outlined in Figure 2.2, each image is added to the solution and the initial approximations are updated until the final image is added and a final result is obtained. As a final step, the resulting model can be scaled. This can be done if the approximate distance to the target is known for one of the images, or if the approximate distance between two points on the target is known.

2.2 Haddock Images at Close Range with the CANON 5D

The first set of images used to test the standard algorithms was a series of eight images of the Coast Guard Vessel Haddock, taken in San Diego Harbor, CA. The images were taken with a 12.7 megapixel CANON 5D digital camera with a 400 mm lens at distances ranging from approximately 300 m – 450 m. The eight images that were used are shown in Figure 2.3.

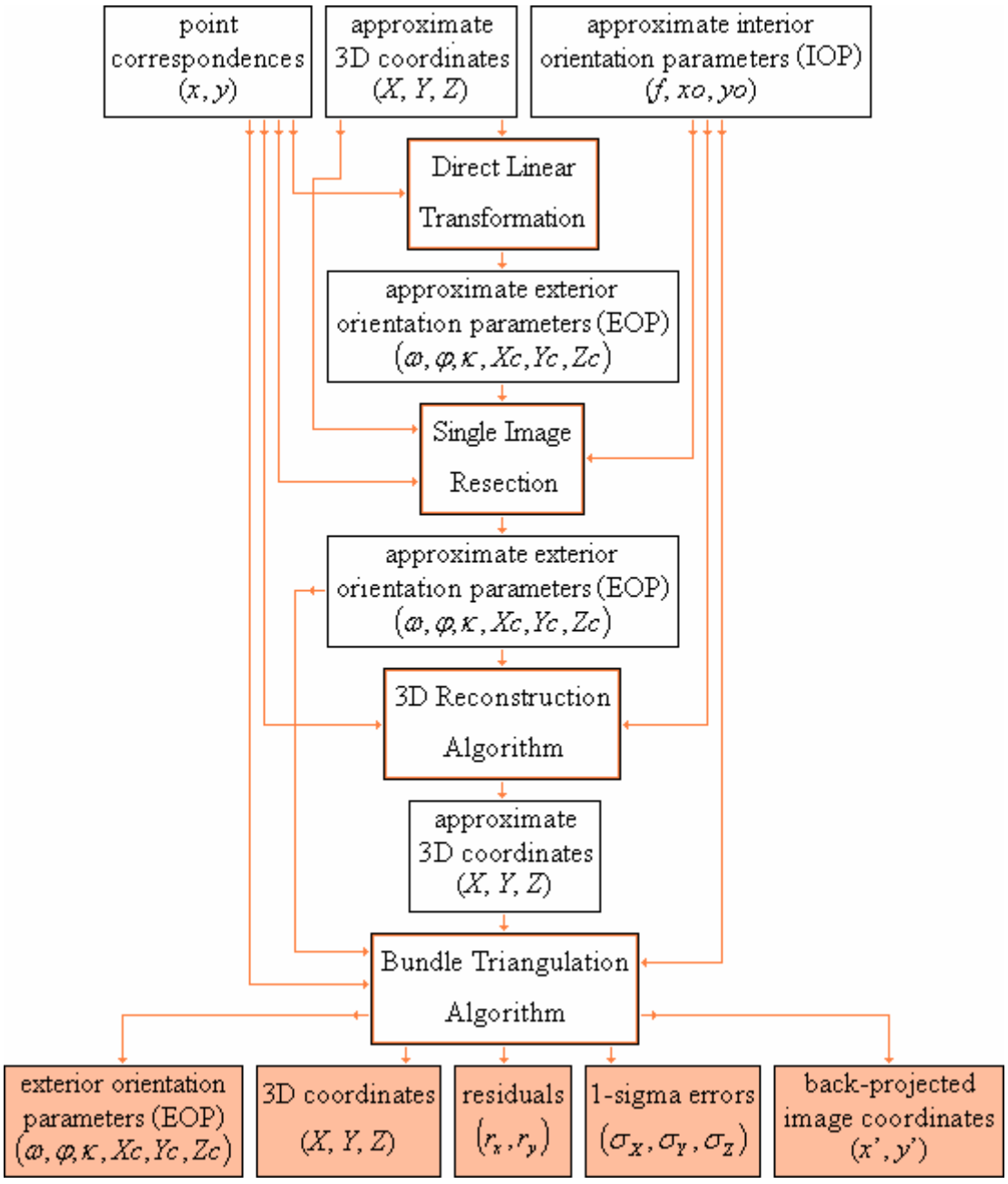

Figure 2.2. Flow chart for additional images using standard photogrammetry methods.

Figure 2.3. Eight images of the Haddock at close range with the CANON 5D.

This dataset is, in some respects, a good representative of surveillance imagery. The images were taken from several points of view so each of the images has an aspect which is quite different from the other images. While the images were taken roughly in a circle around the target, there was no strict pattern or order in the image collection as is the case with topographic photogrammetry. There was no control information recorded for the image locations. Also, the target did move somewhat between images and the images were captured from a moving platform.

The images were taken with a camera for which no calibration had been performed so the precise IOP were unknown. Good approximations were available however, which may not be the case with all surveillance imagery.

The target has an unknown geometry—there was no control information given for the target.

The images do have one characteristic that will not be the case with all sets of surveillance imagery—the images were captured at a modest range with the vessel largely filling the field of view of the camera. So, these images were expected to cause only minor limitations due to narrow angular field-of-view. Also, because the target nearly fills the field of view for several of the images, an accurate self-calibration to solve for the IOP may be possible.

Thus, this test does not fully investigate the problems that can arise from surveillance imagery, but it does provide a good starting point.

Sixty-five point correspondences were located on the images. The point correspondences were manually selected and point coordinates were measured in pixels. Not all 65 points were visible in all images. Several of these point correspondences are

shown on one of the images in Figure 2.4. This set of point correspondences was the first input to the standard algorithms.

The next input was the approximate IOP. The images were taken with a CANON 5D digital camera with a 400 mm lens. A nominal value of 400 mm was chosen as the approximate value for the principal distance. While the focal length is a good approximation of the principal distance, it is not necessarily the true principal distance. This value had to be converted into units of pixels to match the units of the image coordinates that were collected. The CANON 5D has an imaging element which is 35.8 mm x 23.9 mm (4368 pixel x 2912 pixel). Thus, the value for the principal distance was 400 mm * (4368 pixel / 35.8 mm) = 48770 pixel. Because the principal points of the images were not known exactly, it was assumed that the center pixel of each image

Figure 2.4. Point correspondences on the Haddock.

(2184, 1456) was the principal point. As noted above, the vessel largely fills the field of view for several of the images, so a self-calibration to solve for the IOP may be possible with these images. For these tests, self-calibration was not performed.

With these inputs defined, the standard methods were carried out following the procedure described above. The final results consisted of a set of 3D coordinates, the EOP for each image, residuals, 1-sigma errors, and back-projected image coordinates. The scale for the final model was chosen by approximating the distance between two points on the target (specifically, two of the points on the stern of the ship were compared with the water depth markers on the hull of the ship near the stern) and then scaling the model to match this approximate distance. Because the units of the model are not exactly (m), the units will be designated as (~m).

The results from this set of Haddock images were very informative. The first output of the algorithms was the set of XYZ coordinates of the target. These were plotted and were shown to visually resemble the 3D geometry of the points on the actual target. This gave some initial confidence in the solution from the standard photogrammetric methods.

The next output of the algorithms was the set of image residuals. For each point on each image the total residual was calculated using equations (A.29 – A.31). The mean of the total residuals across all eight images was 0.88 pixels with a maximum of 3.27 pixels. These values were favorable when considering that each point coordinate was only collected to roughly the nearest pixel.

The final output that was considered for these images was the set of 1-sigma errors. The 1-sigma errors were calculated with equations (A.32 – A.37). The mean of

the total 1-sigma errors for the 65 points was 0.006 ~m (0.020 ft) and the maximum total 1-sigma error for the 65 points was 0.015 ~m (0.048 ft). These 1-sigma errors are calculated as part of the bundle triangulation algorithm and are based on the assumption that the residuals follow a normal distribution. The 1-sigma errors obtained for this test were very favorable.

As a second test, analytical relative orientation was used to find the relative orientation of the first two images instead of the essential matrix approach. This method requires an initial approximation for the relative orientation of the first two images. An initial approximation was made by assuming that the first two images were taken from the same distance and by guessing at the vertical and horizontal angular difference between the two points of view for the first two images. Using these initial approximations for the relative orientation of the first two images, analytical relative orientation was performed. Then, the procedure continued in the same manner as explained for the first test. The 3D reconstruction equations (A.44, A.45), bundle triangulation algorithm, and DLT were used to add each of the additional images. The results from this second test were nearly identical to the results from the first test.

Therefore, for this set of surveillance images, the standard photogrammetric algorithms were capable of performing accurate 3D reconstruction.

2.3 Haddock Images at Medium Range with the CANON 20D

The second set of images used to test the standard photogrammetric algorithms was a series of six images, also of the Coast Guard Vessel Haddock, taken in San Diego Harbor, CA. The images were taken with an 8.2 megapixel CANON 20D digital camera

with a 400 mm lens at distances ranging from approximately 1100 m – 1700 m with an average of 1475 m. The six images that were used are shown in Figure 2.5.

This dataset is also a good representative of surveillance imagery. The imagees were captured from different points of view. There was no control information available for the target or the EOP. The images were taken with an uncalibrated camera. Also, the

Figure 2.5. Six images of the Haddock at medium range with the CANON 20D.

images were taken at a greater range than the previous images resulting in a narrower field of view. The vessel does not fill a significant portion of the field of view for any of the images, making self-calibration unlikely.

For the first input, a total of 46 point correspondences were collected from the six images, again with some points missing on some images. The IOP were approximated to roughly match the CANON 20D. The CANON 20D has a 22.5 mm x 15.0 mm (3504 pixel x 2336 pixel) imaging sensor. Thus, a value of 62293 pixel was chosen as the principal distance (62293 pixel = 400 mm * 3504 pixel / 22.5 mm), and the principal point was chosen as the center of the images (1752 pixel, 1168 pixel).

With all of the inputs defined, the standard photogrammetric algorithms could be carried out in the same manner as for the previous image set. As before, the first step was to calculate the relative orientation of the first two images using the algorithm outlined in (Pan, Brooks, and Newsam, 1995; Pan, Huynh, and Hamlyn, 1995). After using this algorithm to obtain an initial approximation of the orientation of the first two images, 3D reconstruction was performed with equation (A.44, A.45). However, at this point it was noted that the residuals for this 3D reconstruction were extremely high (on the order of 7,000 pixel). Also, by inspecting the 3D coordinates which resulted from this 3D reconstruction it was apparent that they did not resemble the geometry of the actual points on the target. Thus, the approximation of relative orientation resulting from the essential matrix method was inaccurate. Upon attempting to continue with the process, the bundle triangulation algorithm failed to refine the solution, indicating that the initial approximations were not sufficiently close to the true solution to lead to convergence.

Finally, upon attempting to add additional images using the DLT, the solution did not converge to a better result.

This deficiency of the standard methods was to be expected due to the narrow angular field of view of the images. As mentioned above, as the angular field of view decreases, the light rays from the target become more parallel, and the calculations based on the perspective projection become unstable. For example, Fraser and Yamakawa (2004, p. 277) mention,

> [W]ell known shortcomings of over-parameterization … are encountered in the application of collinearity-based models for satellite imagery, and … can also be experienced with the DLT. These arise from projective coupling between parameters of sensor position and spatial orientation.

In an attempt to improve the results, several different pairs of images were put into the essential matrix algorithm in order to find a pair of images which did lead to convergence. Eventually, it was observed that by beginning with the first and sixth images, an accurate orientation was achieved. Then, continuing with the procedure, the second and fifth images were added using the DLT and bundle triangulation algorithm. At this point, it should be noted that the solution from these four images had a mean total residual of 0.87 pixels (comparable to the first test) and a mean total 1-sigma error of 0.067 ~m (0.220 ft). Thus, the standard methods were successful in performing 3D reconstruction from four of the six images. However, upon attempting to add the third and fourth images using the DLT, the solution became unstable.

The complication with the third and fourth images was most likely due to the fact that the target covers a smaller portion of these two images and hence there is a narrower angular field of view for the relevant portion of these images than the other images.

After realizing that these images could not be added to the current solution by using the DLT, attempts were made to orient these images using the essential matrix approach. Some experimentation revealed that by beginning with the sixth image and fourth image, the essential matrix approach was capable of producing an accurate initial approximation for the relative orientation of these two images. Then by following the procedure as before, the first, second, fifth, and third images were added in order. The final results were a mean total residual of 1.16 pixels and a maximum total residual of 3.41 pixel. The mean total 1-sigma error was 0.036 ~m (0.117 ft) with a maximum of 0.172 ~m (0.564 ft). This represented an accurate 3D reconstruction, although not without some effort.

As a double-check of the results from this test, matching points on the target from the eight-image test were compared with corresponding points on the target from the six-image test. There were 38 corresponding points on the two models. A 3D rigid coordinate transformation (similarity transformation) was used to convert both models into a common coordinate frame. The RMSE between corresponding points was calculated for each point; the mean was 0.049 ~m (0.161 ft). Note that the mean 1-sigma error for the first model was 0.006 ~m (0.020 ft) and the mean 1-sigma for the second model was 0.036 ~m (0.117 ft). So, the correlation between the two models is on the same order as the 1-sigma errors for the second model. This indicates an added degree of confidence in the results of the tests.

Next, it was desired to perform 3D reconstruction by beginning with analytical relative orientation. This method requires an initial approximation for the relative orientation of the first two images, so an initial approximation was made by assuming that the first two images were taken from the same distance and by guessing at the

horizontal and vertical angular difference between the two images. It was necessary to make several trial-and-error guesses before the relative orientation step converged, but eventually an accurate 3D reconstruction was obtained from the first two images. The remaining images were then added using the method described above. The resulting model had a mean total residual of 1.33 pixel and a mean total 1-sigma error of 0.035 ~m (0.115 ft). This model was compared with the model from the first part of the test using a 3D rigid coordinate transformation. The RMSE between the two models was computed for each point; the mean was 0.033 ~m (0.108 ft). Thus, the models were equivalent up to the order of precision of the individual models. Hence, an accurate 3D reconstruction was also obtained by beginning with analytical relative orientation, despite the initial trial and error.

In conclusion, this test revealed that the standard methods began to show some instability due to narrow angular field of view, but were still able to produce an accurate 3D reconstruction.

2.4 Petrel Images at Further Range with the CANON 1Ds

The third set of images used to test the standard photogrammetric algorithms was a series of four images of the Coast Guard Vessel Petrel, taken in San Diego Harbor, CA. The Petrel is the same class of vessel as the Haddock, and therefore should very closely match the Haddock. The images were taken with a 16.7 megapixel CANON 1Ds digital camera with an 800 mm lens at distances ranging from approximately 3000 m – 4300 m with an average range of 3550 m. The four images that were used are shown in Figure 2.6.

Figure 2.6. Four images of the Petrel at further range with the CANON 1Ds.

This dataset is also a good representative of surveillance imagery. The images were captured from different points of view. There was no control information available for the target or the EOP. The images were taken with an uncalibrated camera. Also, the images were taken at a greater range than the previous images resulting in a narrower angular field of view. The target only filled a very small portion of the field of view for each image.

For the first input, 46 point correspondences were collected from the four images which matched the 46 point correspondences collected in the previous Haddock test (assuming that points on the Petrel match points on the Haddock). The IOP were approximated to roughly match the CANON 1Ds. The CANON 1Ds has a 36 mm x 24

mm (4992 pixel x 3328 pixel) imaging sensor. Thus, a value of 110934 pixel was chosen as the principal distance (110934 pixel = 800 mm * 4992 pixel / 36 mm), and the principal point was chosen as the center of the images (2496 pixel, 1664 pixel).

With all of the inputs defined, the standard photogrammetric algorithms could be carried out in the same manner as for the previous image set. As before, the first step was to calculate the relative orientation of the first two images using the essential matrix algorithm outlined in (Pan, Brooks, and Newsam, 1995; Pan, Huynh, and Hamlyn, 1995). Once the relative orientation was found, 3D reconstruction was performed using these two images. The residuals for this calculation were very high (on the order of 7,000 pixel). When proceeding with the bundle triangulation algorithm, the solution did not improve. It was also noted that the resulting geometry of the 3D coordinates did not closely match the geometry of the actual points. Finally, upon attempting to add additional images using the DLT, the solution did not converge to a better result. These results were nearly identical to the initial attempts for the previous six-image test.

As was the case with the previous test, to obtain an accurate result, the essential matrix algorithm, 3D reconstruction, and bundle triangulation were performed on all possible image pairs (images 1 & 3, images 1 & 4, images 2 & 3, images 2 & 4, images 3 & 4). For each of these image pairs, the residuals, 1-sigma errors, and resulting 3D geometry of points were considered. A couple of the image pairs did lead to low residuals (on the order of 1 pixel) and low 1-sigma errors (on the order of 0.03 ~m). However, upon inspection of the geometry of the resulting 3D points, it was determined that none of the image pairs led to an accurate 3D reconstruction of points. Figure 2.7 shows two resulting sets of 3D points (as viewed from above the target). The black dots

are the results obtained from the six-image Haddock test—these points represent an accurate approximation of the true geometry of the points. The red x's are the results from the relative orientation. The points on the left were obtained from images 1 & 2; the points on the right were obtained with images 1 & 4. These two sets are typical of the results that were obtained using other image pairs. In all cases, the resulting geometry and relative orientation were not close enough to the true geometry to cause the subsequent steps of the algorithm to converge. When attempting to add additional images to a solution using the DLT and single image resection, the results did not improve.

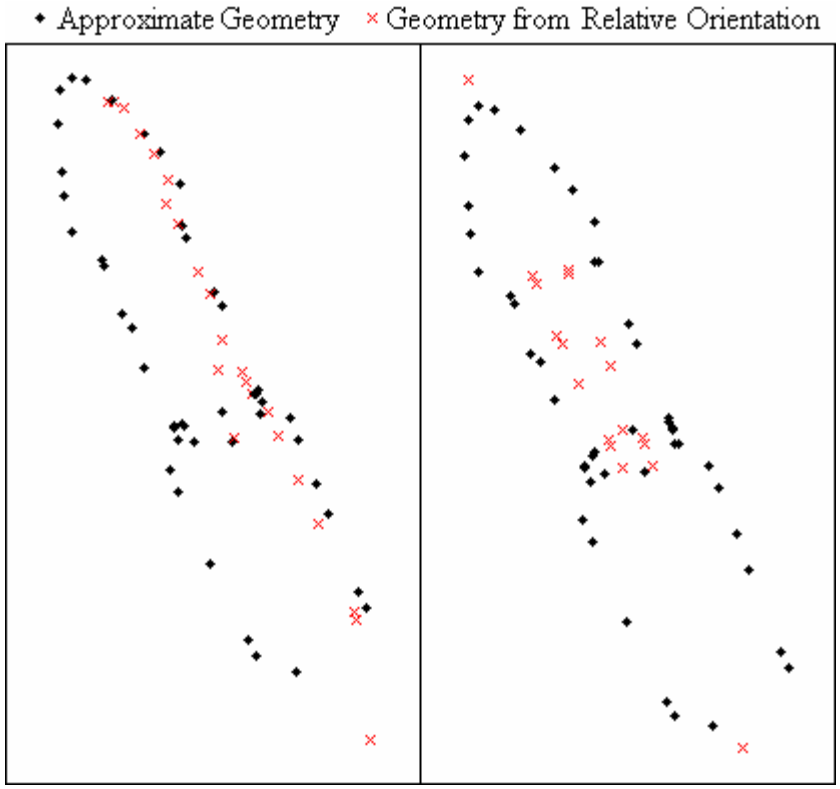

Figure 2.7. Two sets of point geometries resulting from essential matrix relative orientation.

It should also be noted that in many cases when attempting to perform bundle triangulation, the matrix inversion routine returned a warning due to a near singular or rank deficient matrix. This indicates a very high linear dependency between variables and an unstable solution.

The second attempt to obtain an accurate model by standard photogrammetric methods involved beginning with the analytical relative orientation algorithm. This requires an initial guess at the relative orientation of the first two images. As before, it was assumed that the first two images were taken at equal distances from the target and a guess was made at the horizontal and vertical angular differences between the first two images. After performing analytical relative orientation, a 3D reconstruction was performed from the first two images. After some trial and error using different initial approximations, a partial convergence was obtained. For this partial convergence, the residuals and 1-sigma errors were quite low (residuals about 0.5 pixel, 1-sigma errors about 0.03 ~m). However, as with the previous attempts, the resulting 3D geometry of the points deviated noticeably from the actual geometry of the points. The third image was added using the DLT and single image resection as detailed above. After adding the third image, the residuals and 1-sigma errors were still rather good (residuals about 1.5 pixel, 1-sigma errors about 0.15 ~m), but the geometry of points did not improve. When attempting to add the fourth image the solution became unstable and the residuals and 1-sigma errors increased dramatically.

In an effort to reach a stable solution, it was attempted to add the fourth image first, and then add the third image. Once again, the solution remained marginally stable when adding the fourth image, but became unstable when attempting to add the third

image. Figure 2.8 shows the resulting 3D geometries from images 1 & 2, images 1, 2, & 3, and images 1, 2, & 4 along with the solution from the six-image Haddock tests.

Analytical relative orientation was attempted using some of the other image pairs. However, similar results were obtained from these other image pairs. In some cases it was possible to obtain a marginally stable solution with the first two images; in some cases it was possible to add one additional image to the solution and maintain a degree of accuracy, however an accurate result from all four images was not reached. The point geometries represented in Figure 2.8 are characteristic of the results that were obtained.

Thus, the standard photogrammetry techniques were unable to accurately solve for the 3D geometry of the target for these four images. The primary complications come from the fact that the images have a narrow angular field of view, were captured from

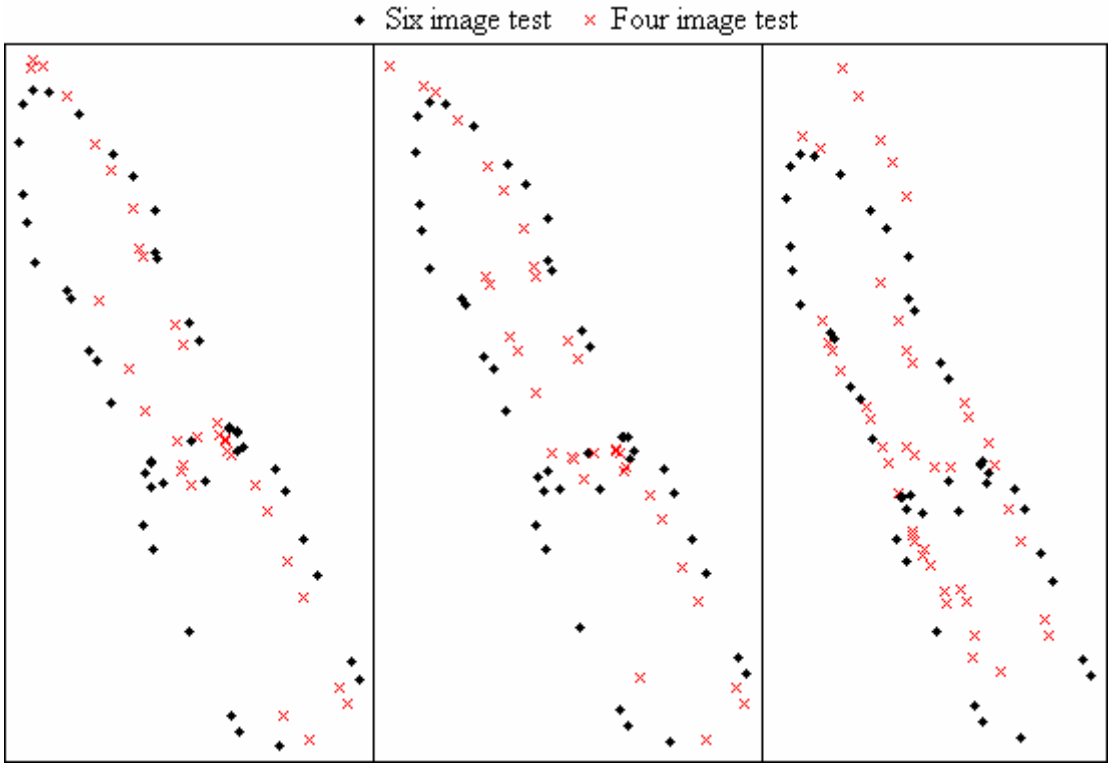

Figure 2.8. Three sets of point geometries resulting from analytical relative orientation.

various and unknown aspects, and are images of a target for which no other information is available.

2.5 Conclusions and Recommendations

The above tests were instructive in pointing out some of the limitations of the standard photogrammetry methods when using surveillance images for 3D reconstruction.

First, it was shown that the current methods used for obtaining initial approximations (the essential matrix algorithm and analytical relative orientation) can become unstable when using images that have a narrow angular field of view. It was also shown that because these algorithms can only accommodate two images at a time, it was necessary to try different pairs of images until convergence was reached. Further, analytical relative orientation requires an initial approximation of the relative orientation of the two images. In the absence of ground control or sensor orientation information this initial approximation has to simply be a guess at the angular difference between the two images and the relative distance to the target for the two images.

The acquisition of accurate initial approximations is crucial to the remainder of the 3D reconstruction process, specifically for bundle triangulation. Hence, it is desired to develop an algorithm which is more reliable at calculating initial approximations when using images that may have a narrow angular field of view. It is also desired that this algorithm has the capacity to utilize more than two images at a time, removing the necessity to perform calculations on several pairs of images. It is also desired that this

algorithm, like the essential matrix algorithm, should rely on point correspondences only—not ground control or an initial approximation of the relative orientation.

Second, it was shown that when trying to add another image to a given network (find an initial approximation of the EOP and IOP of a single image relative to a set of 3D coordinates), the DLT was not consistent—especially when using images with a narrow angular field of view. Thus, it is desired to develop a more robust algorithm to find an initial approximation of the EOP of a single image relative to a set of 3D coordinates.

Third, the perspective projection in general has the tendency to become unstable as the angular field of view of an image decreases. The standard bundle triangulation algorithm is based on the perspective projection, and hence loses stability as the angular field of view of the images decreases. Because the bundle triangulation algorithm is such a powerful tool for 3D reconstruction in various circumstances, it is desired to develop a form of the bundle triangulation algorithm which is more dependable when using images with a narrow angular field of view.

CHAPTER 3

PARALLEL PROJECTION AND PERSPECTIVE-CORRECTED PARALLEL PROJECTION

This chapter gives an explanation of the parallel projection and perspective-corrected parallel projection. In particular, the parallel projection bundle triangulation algorithm is outlined and some important relationships between parallel projection and perspective projection will be derived. Much of the material presented here will be referenced in Chapter 5 where the comprehensive algorithm is outlined.

3.1 Parallel Projection

Parallel projection, like the perspective projection, is a mathematical description of how a point in 3D space is projected onto a 2D image. Parallel projection is a simpler model than the perspective projection, involves fewer parameters, and is based on a different geometry. Because of this, the parallel projection is able to overcome some of the problems that plague the perspective projection in situations where images have a narrow angular field of view. The parallel projection is able to deliver high precision in such situations and is far more robust (Morgan, 2004; Fraser and Yamakawa, 2004).

The parallel projection is a specific form of an affine projection. The general affine projection represents a point in an image (x, y) as a linear combination of the coordinates of its corresponding point in 3D space (X, Y, Z):

$$\begin{aligned} x &= A_1 X + A_2 Y + A_3 Z + A_4 \\ y &= A_5 X + A_6 Y + A_7 Z + A_8 \end{aligned} \tag{3.1}$$

The eight *A* terms $(A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8)$ are referred to as the affine projection parameters, or parallel projection parameters. Throughout the remainder of this report, these *A* terms will be referred to as affine projection parameters. They describe both the EOP and the IOP. Generally these eight parameters are treated as eight independent parameters. However, the parallel projection model outlined here will constrain the eight affine projection parameters so that they can be expressed as six parallel projection parameters.

The parallel projection model that is developed herein will be based on the assumption that all light rays from the target travel parallel to the focal axis of the camera until they intersect the image plane. It will also be assumed that the image plane is perpendicular to the focal axis. It is possible to allow the image plane to deviate from being orthogonal to the focal axis, as is the case with the model developed in (Morgan, 2004). In this case, there are eight degrees of freedom and the full eight-parameter affine projection model is used. However, herein the projection will be constrained to be parallel to the focal axis, resulting in a six-degree-of-freedom model. The two degrees of freedom which are lost correspond to image skew, and non-uniform scaling in the *x*- and *y*-directions in the image.

The 3D geometry of the parallel projection model is shown in Figure 3.1. For comparison, it may be instructive also to recall Figure 1.3, which is a 3D representation of the perspective projection.

As with the perspective projection, the 3D coordinate system is denoted by **XYZ**, the image coordinate system is denoted by *xyz*, and image coordinates are measured in a 2D coordinate system on the image plane. As with the perspective projection, there is a

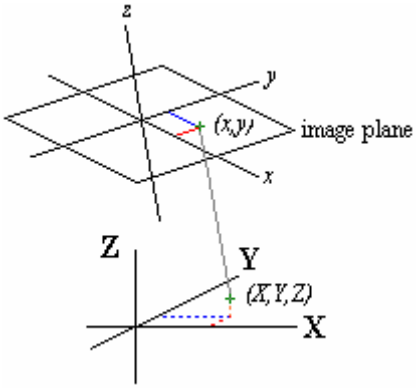

Figure 3.1. 3D depiction of the parallel projection.

rotation of the image coordinate system with respect to the 3D coordinate system which can be described by a 3 x 3 rotation matrix. Note that Figure 3.1 does not account for a scale change. In the parallel projection there is a difference in scale between the object and the image. Conceptually, this scale change can occur in 3D space before the target points are projected onto the image plane, or it can occur in the image after the target points are projected onto the image plane.

When scaling is considered, the parallel projection can conceptually be accomplished by two steps—projection and rescaling. First the points on the target object are projected parallel to the focal axis onto a plane (the 'average plane') which is parallel to the image plane and perpendicular to the focal axis. Next, the image of the object on the average plane is rescaled to the size of the image. Figure 3.2 shows this process and is based on the material presented by Ono and Hattori (2002). This illustration will be helpful later to visualize the similarity between parallel projection and perspective projection.

Mathematically parallel projection can be accomplished in four steps—rotation, projection, rescaling, and translation. First, the 3D coordinates of the target are rotated

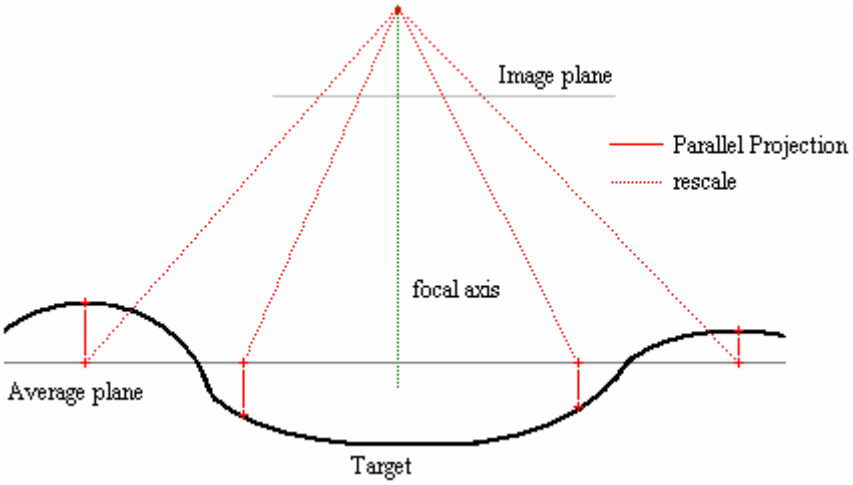

Figure 3.2. Parallel projection and rescaling.

from the **XYZ**-system to align with the image coordinate system *xyz*. Then by disregarding the *z*-coordinate (i.e. by letting all points have a *z*-coordinate of 0), the object is projected parallel to the focal axis onto the *xy*-plane of the image coordinate system. Then the coordinates are re-scaled to the size of the image. Finally, the remaining *x* and *y* coordinates can be shifted to align with the 2D image coordinate system on the image plane.

Therefore, given the assumption that the parallel projection is perpendicular to the focal axis, the parallel projection is defined by six parameters: three rotation angles $(\omega, \varphi, \kappa)$, a scale *s*, and two shift terms $(\Delta x, \Delta y)$. These $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$ will be referred to as the parallel projection parameters.

3.1.1 Parallel projection equations

The following equation (3.2) expresses the image coordinates (*x*, *y*) of a point in 3D space (*X*, *Y*, *Z*) as a function of the parallel projection parameters $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$:

$$\begin{bmatrix} x-\Delta x \\ y-\Delta y \\ * \end{bmatrix} = s \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.2}$$

In this equation (3.2), the [m] matrix is a rotation matrix based on the three rotation angles. The top two rows of this equation can be written as:

$$\begin{aligned} x &= sm_{11}X + sm_{12}Y + sm_{13}Z + \Delta x \\ y &= sm_{21}X + sm_{22}Y + sm_{23}Z + \Delta y \end{aligned} \tag{3.3}$$

These are the fundamental equations for the parallel projection, just as the collinearity equations are the fundamental equations for the perspective projection. These equations (3.3) can be made to match the general affine projection equation (3.1) with the following substitutions:

$$\begin{aligned} A_1 &= sm_{11} \quad & A_2 &= sm_{12} \quad & A_3 &= sm_{13} \quad & A_4 &= \Delta x \\ A_5 &= sm_{21} \quad & A_6 &= sm_{22} \quad & A_7 &= sm_{23} \quad & A_8 &= \Delta y \end{aligned} \tag{3.4}$$

Therefore, the parallel projection model outlined here is a specific form of an affine projection. By constraining the image plane to be perpendicular to the focal axis, the eight affine projection parameters $(A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8)$ have been reduced to six parallel projection parameters $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$, which describe the EOP and the IOP of the image. The major motivation for this constraint is to maintain a high similarity to the perspective projection. This constraint is further explained and other advantages of constraining the eight parameters to six parameters are discussed by Ono and Hattori (2002). This results in a different form of the model outlined by Morgan (2004) which is an eight-parameter model.

As with the perspective projection equations, there are several forms of the parallel projection equations which are helpful. The forms expressed in (3.2) and (3.3)

will be used along with equation (3.5) which expresses the shifted and scaled image coordinates $(\hat{x}, \hat{y})$ as the rotated 3D coordinates (*X*,*Y*,*Z*):

$$\begin{bmatrix} \hat{x} \\ \hat{y} \\ * \end{bmatrix} = \begin{bmatrix} (x - \Delta x)/s \\ (y - \Delta y)/s \\ * \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.5}$$

3.1.2 3D Reconstruction by parallel projection

Because the parallel projection equations (3.2) can be written in a linear manner, 3D reconstruction is a straightforward calculation. Given a point in 3D space, a set of *I* images for which the parallel projection parameters are known, and the image coordinates of the point in each of the *I* images, a matrix equation can be written:

$$\begin{bmatrix} x_A - \Delta x_A \\ y_A - \Delta y_A \\ x_B - \Delta x_B \\ y_B - \Delta y_B \\ \cdots \\ x_I - \Delta x_I \\ y_I - \Delta y_I \end{bmatrix} = \begin{bmatrix} s_A m_{11,A} & s_A m_{12,A} & s_A m_{13,A} \\ s_A m_{21,A} & s_A m_{22,A} & s_A m_{23,A} \\ s_B m_{11,B} & s_B m_{12,B} & s_B m_{13,B} \\ s_B m_{21,B} & s_B m_{22,B} & s_B m_{23,B} \\ \cdots & \cdots & \cdots \\ s_I m_{11,I} & s_I m_{12,I} & s_I m_{13,I} \\ s_I m_{21,I} & s_I m_{22,I} & s_I m_{23,I} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.6}$$

In equation (3.6) (x_I, y_I) represent the image coordinates in image *I*, and $(s_I, \Delta x_I, \Delta y_I, m_{11,I}, m_{12,I}, ...)$ represent the parallel projection parameters for image *I*; recall that the matrix terms *m* can be written in terms of the angles $(\omega_I, \varphi_I, \kappa_I)$. This equation is of the form $L = AX$ and the least-squares solution is $X = (A^T A)^{-1}(A^T L)$. This gives the 3D coordinates of a point by parallel projection.

Next, the back-projected image coordinates (x', y') can be calculated:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = s \begin{bmatrix} m_1 & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3.7}$$

Where (X', Y', Z') are the calculated 3D coordinates of the point. The residuals can then be calculated for each point on each image by:

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} x'-x \\ y'-y \end{bmatrix} \tag{3.8}$$

The magnitude of these residuals provides a metric for determining the precision of each calculation. They may also be used to detect blunders in the image coordinates that were measured from the images. It also may also be helpful to consider the total residual r for each point on each image:

$$r = \sqrt{r_x^2 + r_y^2} \tag{3.9}$$

While image residuals are very helpful, they do not directly provide a measure of precision in 3D space—they give the degree of accuracy in 2D image space. Since the primary concern is an accurate model of the target in 3D space, 1-sigma errors are used to determine the precision of the calculations in 3D space. The process for calculating the 1-sigma errors begins by considering equation (3.6), which is of the form $L = AX$; the least-squares solution of this equation is $X = \left(A^T A\right)^{-1}\left(A^T L\right)$.

Note that the residuals for each point on each image (equivalent to equation 3.8) can be calculated as a vector by:

$$R = AX - L \tag{3.10}$$

This vector of residuals is then used to calculate the standard deviation of unit weight:

$$\sigma_0 = \sqrt{\frac{R^T R}{n}} \tag{3.11}$$

In equation (3.11) *n* represents the degrees of freedom and is given by:

n = (the number of equations) – (the number of unknowns) (3.12)

The standard deviation of unit weight σ_0 is then used to calculate the 1-sigma errors:

$$\sigma_X = \sigma_0 \sqrt{Q_{11}} \qquad \sigma_Y = \sigma_0 \sqrt{Q_{22}} \qquad \sigma_Z = \sigma_0 \sqrt{Q_{33}} \tag{3.13}$$

In equation (3.13) Q_{xx} represents the diagonal elements of the cofactor matrix:

$$Q = \left(A^T A\right)^{-1} \tag{3.14}$$

A further explanation of this calculation of 1-sigma errors is given in (Mikhail, Bethel, and McGlone, 2001; Luhmann et al., 2001). The total 1-sigma error of a single point may also be calculated by:

$$\sigma = \sqrt{\sigma_X^2 + \sigma_y^2 + \sigma_Z^2} \tag{3.15}$$

The 1-simga errors derived above are based on the assumption that the errors in the calculations follow a standard normal distribution. The 1-sigma error values represent one standard deviation of the standard normal distribution. This means that for a given point *p*, there is a 68% probability that the true value of X_p lies in the interval $\left(X_p' - \sigma_X, X_p' + \sigma_X\right)$, and a 95% probability that the true value of X_p lies in the interval $\left(X_p' - 2\sigma_X, X_p' + 2\sigma_X\right)$. While the error may not actually follow a standard normal distribution, these 1-sigma errors still give a metric whereby the precision of the calculations in 3D space can be determined.

3.2 Parallel Projection Bundle Triangulation

As noted above, it is desired to develop a form of the bundle triangulation which is more reliable when using images that may have a narrow angular field of view. One of the contributions of this thesis is the outline of the parallel projection bundle triangulation algorithm. It follows the pattern of the standard bundle triangulation algorithm, but is based on the parallel projection equations as opposed to the perspective projection equations. The use of both algorithms will provide greater accuracy and consistency over a wider range of fields of view.

3.2.1 Linearization of the parallel projection equations

The parallel projection equations (3.2) which describe the specific six-term parallel projection model are non-linear in terms of the six unknowns $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$. By taking partial derivatives of the parallel projection equations in the same manner as with the perspective projection equations, they can be linearized by a first order Taylor's theorem approximation. This will be for the parallel projection bundle triangulation algorithm. Consider the partial derivatives (3.16 – 3.17) of x and y in equation (3.2) with respect to the unknowns $(s, \omega, \varphi, \kappa, \Delta x, \Delta y, X, Y, Z)$:

$$
\begin{aligned}
&\frac{\partial x}{\partial s} = m_{11}X + m_{12}Y + m_{13}Z \qquad \frac{\partial x}{\partial \omega} = s\left(-m_{13}Y + m_{12}Z\right) \\
&\frac{\partial x}{\partial \varphi} = s\left[\left(-\sin\varphi\cos\kappa\right)X + \left(\sin\omega\cos\varphi\cos\kappa\right)Y + \left(-\cos\omega\cos\varphi\cos\kappa\right)Z\right] \\
&\frac{\partial x}{\partial \kappa} = s\left(m_{21}X + m_{22}Y + m_{23}Z\right) \qquad \frac{\partial x}{\partial \Delta x} = 1 \qquad \frac{\partial x}{\partial \Delta y} = 0 \\
&\frac{\partial x}{\partial X} = sm_{11} \qquad \frac{\partial x}{\partial Y} = sm_{12} \qquad \frac{\partial x}{\partial Z} = sm_{13}
\end{aligned}
\tag{3.16}
$$

$$\frac{\partial y}{\partial s} = m_{21}X + m_{22}Y + m_{23}Z \qquad \frac{\partial y}{\partial \omega} = s\left(-m_{23}Y + m_{22}Z\right)$$
$$\frac{\partial y}{\partial \varphi} = s\left[\left(\sin\varphi\sin\kappa\right)X + \left(-\sin\omega\cos\varphi\sin\kappa\right)Y + \left(\cos\omega\cos\varphi\sin\kappa\right)Z\right]$$
$$\frac{\partial y}{\partial \kappa} = s\left(-m_{11}X - m_{12}Y - m_{13}Z\right) \qquad \frac{\partial y}{\partial \Delta x} = 0 \qquad \frac{\partial y}{\partial \Delta y} = 1 \tag{3.17}$$
$$\frac{\partial y}{\partial X} = sm_{21} \qquad \frac{\partial y}{\partial Y} = sm_{22} \qquad \frac{\partial y}{\partial Z} = sm_{23}$$

Now, if initial approximations $s_0, \omega_0, \varphi_0, \kappa_0, \Delta x_0, \Delta y_0, X_0, Y_0, Z_0$ are available for the variables, then the equations (3.2, 3.16, and 3.17) can be calculated at these initial approximations. When this is done, by Taylor's theorem the first-order approximations of the image coordinates (*x*, *y*) are:

$$x \approx x'_0 + \left(\frac{\partial x}{\partial s}\right)_0 ds + \left(\frac{\partial x}{\partial \omega}\right)_0 d\omega + \left(\frac{\partial x}{\partial \varphi}\right)_0 d\varphi + \left(\frac{\partial x}{\partial \kappa}\right)_0 d\kappa + \ldots$$
$$\ldots + \left(\frac{\partial x}{\partial \Delta x}\right)_0 d\Delta x + \left(\frac{\partial x}{\partial \Delta y}\right)_0 d\Delta y + \left(\frac{\partial x}{\partial X}\right)_0 dX + \left(\frac{\partial x}{\partial Y}\right)_0 dY + \left(\frac{\partial x}{\partial Z}\right)_0 dZ$$
$$y \approx y'_0 + \left(\frac{\partial y}{\partial s}\right)_0 ds + \left(\frac{\partial y}{\partial \omega}\right)_0 d\omega + \left(\frac{\partial y}{\partial \varphi}\right)_0 d\varphi + \left(\frac{\partial y}{\partial \kappa}\right)_0 d\kappa + \ldots \tag{3.18}$$
$$\ldots + \left(\frac{\partial y}{\partial \Delta x}\right)_0 d\Delta x + \left(\frac{\partial y}{\partial \Delta y}\right)_0 d\Delta y + \left(\frac{\partial y}{\partial X}\right)_0 dX + \left(\frac{\partial y}{\partial Y}\right)_0 dY + \left(\frac{\partial y}{\partial Z}\right)_0 dZ$$

Note that equations (3.18) are linear in terms of the set of nine unknowns $\left(ds, d\omega, d\varphi, d\kappa, d\Delta x, d\Delta y, dX, dY, dZ\right)$; six for the exterior orientation of the image, and three for the 3D coordinate of the point. Also note that $\left(x'_0, y'_0\right)$ are calculated by equation (3.7) using the initial approximations.

3.2.2 Parallel projection bundle triangulation algorithm

The parallel projection bundle triangulation algorithm is patterned after the standard bundle triangulation. A special case of the parallel projection bundle triangulation algorithm will be considered by assuming some constraints (as is the case with the standard bundle triangulation algorithm outlined in the appendix). It is noted however, that the parallel projection triangulation could be extended to accommodate many different circumstances in the same manner that the standard bundle triangulation algorithm is used.

First, it is assumed that there are *I* images (image *A*, image *B*,…, image *I*) with unknown orientations $(s_A, \omega_A, \varphi_A, \kappa_A, \Delta x_A, \Delta y_A)$, $(s_B, \omega_B, \varphi_B, \kappa_B, \Delta x_B, \Delta y_B)$, …, $(s_I, \omega_I, \varphi_I, \kappa_I, \Delta x_I, \Delta y_I)$. It is assumed further that there are *n* target object points with unknown coordinates $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), ..., (X_n, Y_n, Z_n)$ which appear in the *I* images. It is also assumed that the set of image coordinates of the *n* points in the *I* images are known and are denoted as:

$$\begin{matrix} (x_{A,1}, y_{A,1}) & (x_{B,1}, y_{B,1}) & ... & (x_{I,1}, y_{I,1}) \\ (x_{A,2}, y_{A,2}) & (x_{B,2}, y_{B,2}) & ... & (x_{I,2}, y_{I,2}) \\ ... & ... & ... & ... \\ (x_{A,n}, y_{A,n}) & (x_{B,n}, y_{B,n}) & ... & (x_{I,n}, y_{I,n}) \end{matrix} \tag{3.19}$$

For this algorithm, all *n* points do not need to be visible in all *I* images—however, it is necessary for each point to be visible in at least two images.

In the parallel projection, for each image there are six unknowns—an image scale, three rotation angles and two shift terms $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$. Also, for each 3D point there are three unknowns (X, Y, Z). It will be assumed that there are initial approximations for

each of the unknowns. Hence, the set of inputs will be: the set of image coordinates (point correspondences) and the initial approximations of the unknowns.

As is the case with the perspective projection bundle triangulation algorithm, there is no initial absolute orientation of the 3D coordinate system because it has not been defined by the inputs. So, defining an absolute orientation of the 3D coordinate system is the first step. To do this, a rotation, an origin, and a scale for the 3D coordinate system must be defined. There are several ways to define the orientation of 3D space, but here the rotation of 3D space will be defined by fixing the three rotation angles of image *A* at the values of the initial approximations. Next, the scale of 3D space will be defined by fixing the scale of image *A* to be the initial approximation for the scale of image *A*. To fix the origin of 3D space, the *x*- and *y*- shift terms for image *A* and the *x*- shift term for image *B* will be "fixed" at the initial approximations:

$$\begin{array}{llll} s_A = (s_A)_0 & \omega_A = (\omega_A)_0 & \varphi_A = (\varphi_A)_0 & \kappa_A = (\kappa_A)_0 \\ \Delta x_A = (\Delta x_A)_0 & \Delta y_A = (\Delta y_A)_0 & \Delta x_B = (\Delta x_B)_0 & \end{array} \tag{3.20}$$

By fixing the shift terms for image *A*, a ray in 3D space parallel to the *z*-axis of image *A* has been defined, and by fixing the *x*-shift term for image *B*, a plane in 3D space perpendicular to the image plane of image *B* and parallel to the *z*-axis of image *B* has been defined. The intersection of this ray with this plane becomes the origin of 3D space.

Now, for each image (except image *A* and image *B*) there are six unknowns $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$, which need to be solved for. For image *A* there are no unknowns, and for image *B* there are five unknowns $(s, \omega, \varphi, \kappa, \Delta y)$. There are also three unknowns for each target point in 3D space.

Now, a system of linear equations is set up. For a given point *p* on a given image *K*, the following two equations can be written:

$$
\begin{aligned}
x_{K,p} &\approx \left(x'_{K,p}\right)_0 + \left(\frac{\partial x_{K,p}}{\partial s_K}\right)_0 ds_K + \left(\frac{\partial x_{K,p}}{\partial \omega_K}\right)_0 d\omega_K + \left(\frac{\partial x_{K,p}}{\partial \varphi_K}\right)_0 d\varphi_K + \left(\frac{\partial x_{K,p}}{\partial \kappa_K}\right)_0 d\kappa_K + \ldots \\
&\ldots + \left(\frac{\partial x_{K,p}}{\partial \Delta x_K}\right)_0 d\Delta x_K + \left(\frac{\partial x_{K,p}}{\partial \Delta y_K}\right)_0 d\Delta y_K + \left(\frac{\partial x_{K,p}}{\partial X_p}\right)_0 dX_p + \left(\frac{\partial x_{K,p}}{\partial Y_p}\right)_0 dY_p + \left(\frac{\partial x_{K,p}}{\partial Z_p}\right)_0 dZ_p \\
y_{K,p} &\approx \left(y'_{K,p}\right)_0 + \left(\frac{\partial y_{K,p}}{\partial s_K}\right)_0 ds_K + \left(\frac{\partial y_{K,p}}{\partial \omega_K}\right)_0 d\omega_K + \left(\frac{\partial y_{K,p}}{\partial \varphi_K}\right)_0 d\varphi_K + \left(\frac{\partial y_{K,p}}{\partial \kappa_K}\right)_0 d\kappa_K + \ldots \\
&\ldots + \left(\frac{\partial y_{K,p}}{\partial \Delta x_K}\right)_0 d\Delta x_K + \left(\frac{\partial y_{K,p}}{\partial \Delta y_K}\right)_0 d\Delta y_K + \left(\frac{\partial y_{K,p}}{\partial X_p}\right)_0 dX_p + \left(\frac{\partial y_{K,p}}{\partial Y_p}\right)_0 dY_p + \left(\frac{\partial y_{K,p}}{\partial Z_p}\right)_0 dZ_p
\end{aligned}
\tag{3.21}
$$

Like equations (3.18), these equations (3.21) are linear in the unknowns $\left(ds_K, d\omega_K, d\varphi_K, d\kappa_K, d\Delta x_K, d\Delta y_K, dX_p, dY_p, dZ_p\right)$. If these two equations are written for each of the points which appear in each of the images, then a linear system of equations with the following unknowns is obtained:

$$
\begin{aligned}
&\left(ds_B, d\omega_B, d\varphi_B, d\kappa_B, d\Delta y_B\right), \left(ds_C, d\omega_C, d\varphi_C, d\kappa_C, d\Delta x_C, d\Delta y_C\right), \ldots \\
&\ldots, \left(ds_I, d\omega_I, d\varphi_I, d\kappa_I, d\Delta x_I, d\Delta y_I\right) \\
&\left(dX_1, dY_1, dZ_1\right), \left(dX_2, dY_2, dZ_2\right), \ldots, \left(dX_n, dY_n, dZ_n\right)
\end{aligned}
\tag{3.22}
$$

Note that the above set of unknowns (3.22) does not include the following terms:

$$
\begin{aligned}
&ds_A = 0 \qquad d\omega_A = 0 \qquad d\varphi_A = 0 \qquad d\kappa_A = 0 \\
&d\Delta x_A = 0 \qquad d\Delta y_A = 0 \qquad d\Delta x_B = 0
\end{aligned}
\tag{3.23}
$$

These terms (3.23) correspond to variables that were fixed in order to define an absolute 3D coordinate system. Hence, these variables are not included in the set of unknowns.

So, by considering the equations for each point in each image a linear system can be written in matrix-form $L = AX$ with L, A, and X as follows:

$$L = \begin{bmatrix} x_{A,1} - \left(x'_{A,1}\right)_0 \\ y_{A,1} - \left(y'_{A,1}\right)_0 \\ \cdots \\ x_{A,n} - \left(x'_{A,n}\right)_0 \\ y_{A,n} - \left(y'_{A,n}\right)_0 \\ x_{B,1} - \left(x'_{B,1}\right)_0 \\ y_{B,1} - \left(y'_{B,1}\right)_0 \\ \cdots \\ x_{B,n} - \left(x'_{B,n}\right)_0 \\ y_{B,n} - \left(y'_{B,n}\right)_0 \\ \cdots \\ x_{I,1} - \left(x'_{I,1}\right)_0 \\ y_{I,1} - \left(y'_{I,1}\right)_0 \\ \cdots \\ x_{I,n} - \left(x'_{I,n}\right)_0 \\ y_{I,n} - \left(y'_{I,n}\right)_0 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \\ \\ \\ \\ \} n \text{ points in image } A \\ \\ \\ \\ \} n \text{ points in image } B \\ \\ \\ \\ \} n \text{ points in image } I \end{matrix} \tag{3.24}$$

$$A = \begin{bmatrix} 0_{2n\ x\ 5} & 0_{2n\ x\ 6} & \cdots & 0_{2n\ x\ 6} & A_{XYZ} \\ B & 0_{2n\ x\ 6} & \cdots & 0_{2n\ x\ 6} & B_{XYZ} \\ 0_{2n\ x\ 5} & C & \cdots & 0_{2n\ x\ 6} & C_{XYZ} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ 0_{2n\ x\ 5} & 0_{2n\ x\ 6} & \cdots & I & I_{XYZ} \end{bmatrix} \begin{matrix} \} n \text{ points in image } A \\ \} n \text{ points in image } B \\ \} n \text{ points in image } C \\ \\ \} n \text{ points in image } I \end{matrix} \tag{3.25}$$

The terms in equation (3.25) are:

$$B = \begin{bmatrix} \frac{\partial x_{B,1}}{\partial s_B} & \frac{\partial x_{B,1}}{\partial \omega_B} & \frac{\partial x_{B,1}}{\partial \varphi_B} & \frac{\partial x_{B,1}}{\partial \kappa_B} & \frac{\partial x_{B,1}}{\partial \Delta y_B} \\ \frac{\partial y_{B,1}}{\partial s_B} & \frac{\partial y_{B,1}}{\partial \omega_B} & \frac{\partial y_{B,1}}{\partial \varphi_B} & \frac{\partial y_{B,1}}{\partial \kappa_B} & \frac{\partial y_{B,1}}{\partial \Delta y_B} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial x_{B,n}}{\partial s_B} & \frac{\partial x_{B,n}}{\partial \omega_B} & \frac{\partial x_{B,n}}{\partial \varphi_B} & \frac{\partial x_{B,n}}{\partial \kappa_B} & \frac{\partial x_{B,n}}{\partial \Delta y_B} \\ \frac{\partial y_{B,n}}{\partial s_B} & \frac{\partial y_{B,n}}{\partial \omega_B} & \frac{\partial y_{B,n}}{\partial \varphi_B} & \frac{\partial y_{B,n}}{\partial \kappa_B} & \frac{\partial y_{B,n}}{\partial \Delta y_B} \end{bmatrix} \tag{3.26}$$

$$C = \begin{bmatrix} \frac{\partial x_{C,1}}{\partial s_C} & \frac{\partial x_{C,1}}{\partial \omega_C} & \frac{\partial x_{C,1}}{\partial \varphi_C} & \frac{\partial x_{C,1}}{\partial \kappa_C} & \frac{\partial x_{C,1}}{\partial \Delta x_C} & \frac{\partial x_{C,1}}{\partial \Delta y_C} \\ \frac{\partial y_{C,1}}{\partial s_C} & \frac{\partial y_{C,1}}{\partial \omega_C} & \frac{\partial y_{C,1}}{\partial \varphi_C} & \frac{\partial y_{C,1}}{\partial \kappa_C} & \frac{\partial y_{C,1}}{\partial \Delta x_C} & \frac{\partial y_{C,1}}{\partial \Delta y_C} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial x_{C,n}}{\partial s_C} & \frac{\partial x_{C,n}}{\partial \omega_C} & \frac{\partial x_{C,n}}{\partial \varphi_C} & \frac{\partial x_{C,n}}{\partial \kappa_C} & \frac{\partial x_{C,n}}{\partial \Delta x_C} & \frac{\partial x_{C,n}}{\partial \Delta y_C} \\ \frac{\partial y_{C,n}}{\partial s_C} & \frac{\partial y_{C,n}}{\partial \omega_C} & \frac{\partial y_{C,n}}{\partial \varphi_C} & \frac{\partial y_{C,n}}{\partial \kappa_C} & \frac{\partial y_{C,n}}{\partial \Delta x_C} & \frac{\partial y_{C,n}}{\partial \Delta y_C} \end{bmatrix} \tag{3.27}$$

$$I = \begin{bmatrix} \frac{\partial x_{I,1}}{\partial s_I} & \frac{\partial x_{I,1}}{\partial \omega_I} & \frac{\partial x_{I,1}}{\partial \varphi_I} & \frac{\partial x_{I,1}}{\partial \kappa_I} & \frac{\partial x_{I,1}}{\partial \Delta x_I} & \frac{\partial x_{I,1}}{\partial \Delta y_I} \\ \frac{\partial y_{I,1}}{\partial s_I} & \frac{\partial y_{I,1}}{\partial \omega_I} & \frac{\partial y_{I,1}}{\partial \varphi_I} & \frac{\partial y_{I,1}}{\partial \kappa_I} & \frac{\partial y_{I,1}}{\partial \Delta x_I} & \frac{\partial y_{I,1}}{\partial \Delta y_I} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial x_{I,n}}{\partial s_I} & \frac{\partial x_{I,n}}{\partial \omega_I} & \frac{\partial x_{I,n}}{\partial \varphi_I} & \frac{\partial x_{I,n}}{\partial \kappa_I} & \frac{\partial x_{I,n}}{\partial \Delta x_I} & \frac{\partial x_{I,n}}{\partial \Delta y_I} \\ \frac{\partial y_{I,n}}{\partial s_I} & \frac{\partial y_{I,n}}{\partial \omega_I} & \frac{\partial y_{I,n}}{\partial \varphi_I} & \frac{\partial y_{I,n}}{\partial \kappa_I} & \frac{\partial y_{I,n}}{\partial \Delta x_I} & \frac{\partial y_{I,n}}{\partial \Delta y_I} \end{bmatrix} \tag{3.28}$$

$$0_{2n \;\; x \;\; 5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.29}$$

$$0_{2n \;\; x \;\; 6} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{3.30}$$

$$A_{XYZ} = \begin{bmatrix} \frac{\partial x_{A,1}}{\partial X_1} & \frac{\partial x_{A,1}}{\partial Y_1} & \frac{\partial x_{A,1}}{\partial Z_1} & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ \frac{\partial y_{A,1}}{\partial X_1} & \frac{\partial y_{A,1}}{\partial Y_1} & \frac{\partial y_{A,1}}{\partial Z_1} & 0 & 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial x_{A,2}}{\partial X_2} & \frac{\partial x_{A,2}}{\partial Y_2} & \frac{\partial x_{A,2}}{\partial Z_2} & ... & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial y_{A,2}}{\partial X_2} & \frac{\partial y_{A,2}}{\partial Y_2} & \frac{\partial y_{A,2}}{\partial Z_2} & ... & 0 & 0 & 0 \\ ... & ... & ... & ... & ... & ... & ... & ... & ... & ... \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & \frac{\partial x_{A,n}}{\partial X_n} & \frac{\partial x_{A,n}}{\partial Y_n} & \frac{\partial x_{A,n}}{\partial Z_n} \\ 0 & 0 & 0 & 0 & 0 & 0 & ... & \frac{\partial y_{A,n}}{\partial X_n} & \frac{\partial y_{A,n}}{\partial Y_n} & \frac{\partial y_{A,n}}{\partial Z_n} \end{bmatrix} \quad (3.31)$$

In equation (3.25) $B_{XYZ}, C_{XYZ}, \ldots, I_{XYZ}$ follow the same format as A_{XYZ} and represent the partial derivatives for image *B*, image *C*,…, image *I*.

$$X = \begin{bmatrix} ds_B \\ d\omega_B \\ d\varphi_B \\ d\kappa_B \\ d\Delta y_B \\ ds_C \\ d\omega_C \\ d\varphi_C \\ d\kappa_C \\ d\Delta x_C \\ d\Delta y_C \\ \ldots \\ ds_I \\ d\omega_I \\ d\varphi_I \\ d\kappa_I \\ d\Delta x_I \\ d\Delta y_I \\ dX_1 \\ dY_1 \\ dZ_1 \\ \ldots \\ dX_n \\ dY_n \\ dZ_n \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \end{matrix}\right\} \text{image } B \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \text{image } C \\ \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\} \text{image } I \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{point } 1 \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} \text{point } n \end{matrix} \tag{3.32}$$

In equations (3.24 – 3.32) note that the row index of L is equal to the row index of A (each row corresponds to the x- or y-coordinate of one point on one image) and the column index of A is equal to the row index of X (each column of A corresponds to one unknown in X).

Now, by beginning with the initial approximations for the unknown EOP and 3D points, the matrix system (equations 3.24 – 3.32) is set up. Then the least-squares

solution for the unknowns (3.22) is found by $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. Then the unknowns ($X$-vector) are added to the initial approximations to produce improved approximations. The improved approximations become the new initial approximations and this process is repeated until the magnitudes of the corrections X are below a desired threshold.

Once the solution is found, residuals and 1-sigma errors can be obtained in the same manner as for the standard bundle triangulation algorithm (equations A.30 – A.37).

3.3 Relationships Between Perspective Projection and Parallel Projection

Often it will be desired to use both parallel and perspective projection. The perspective projection is widely used and accurate for many circumstances, and the parallel projection provides a good alternative for images with a narrow angular field of view. So that both projections can be utilized in one comprehensive method, a series of relationships between the perspective projection and the parallel projection will be derived here. The primary reason for constraining the parallel projection to a six parameter $\left(s, \omega, \varphi, \kappa, \Delta x, \Delta y\right)$ model, instead of the general eight parameter affine model is to define the following relationships.

Figure 3.3 shows the difference between parallel projection and perspective projection. At first it would appear that the parallel projection and perspective projection have very little in common, but Figures 3.4 and 3.5 show a stronger resemblance between the two models. These two figures are based on the material presented in (Ono and Hattori, 2002). In Figure 3.4, the solid lines represent the perspective projection where light rays travel in straight lines from the target through the image plane to the perspective center. The dashed lines represent the parallel projection (similar to Figure

3.2). Note that the dashed lines first project the target onto a plane (the "average plane" which is at an average distance from the camera) which is parallel to the image plane and perpendicular to the focal axis. Then, the points are scaled down to the size of the image by using the ratio of the principal distance to the distance to the average plane.

Figure 3.5 shows that as the distance to the target becomes large in comparison to the depth of the target and the principal distance, the two models become more similar.

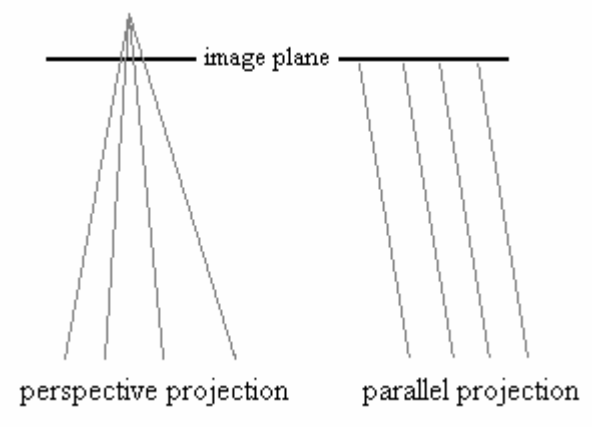

Figure 3.3. Perspective projection vs. parallel projection.

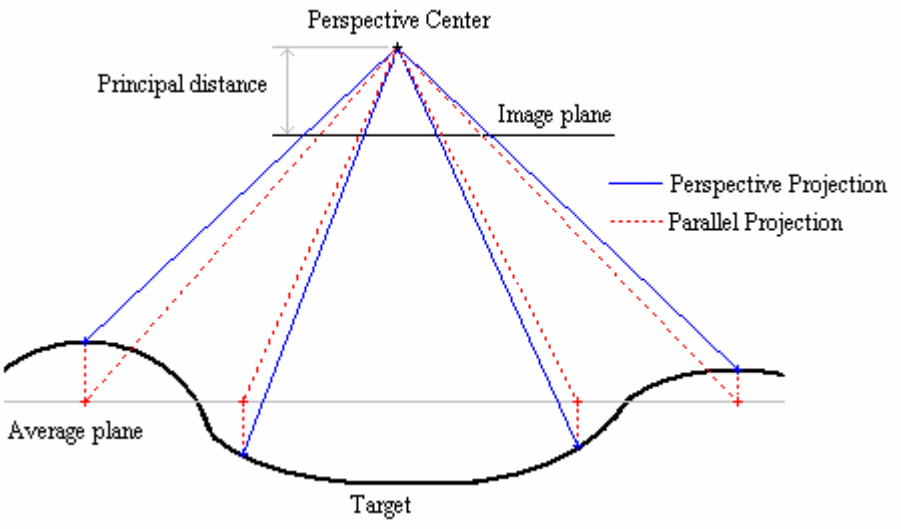

Figure 3.4. Similarity of perspective projection and parallel projection.

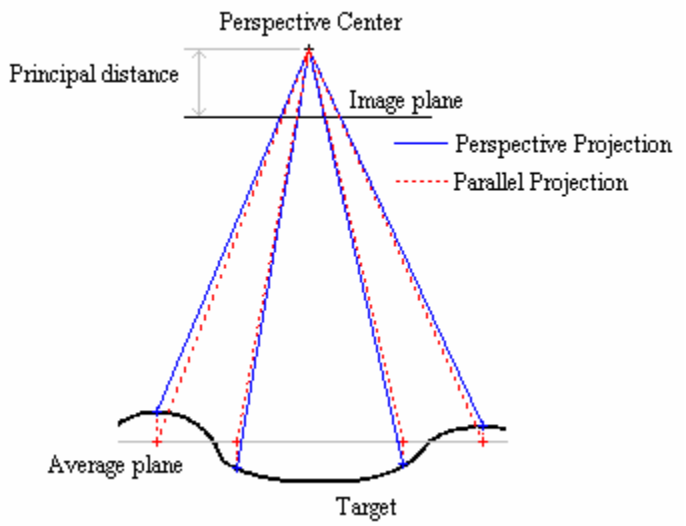

Figure 3.5. Similarity of perspective projection and parallel projection at a greater distance.

Fraser and Yamakawa (2004, p. 277) remark, "[T]he affine model departs from a central-perspective model, though as the field of view of the imaging sensor becomes narrower, the similarity with a parallel projection becomes more apparent …" and "… with imaging systems … with narrow fields of view … the assumption that the projection is parallel has been shown to stand up quite well in practical tests." Thus, as the angular field of view of a sensor becomes narrower, the parallel projection model becomes more valid, even though the perspective projection model is a more rigorous model of the actual imaging geometry. They also comment that the parallel projection model is generally more robust than the perspective projection model when working with images that have a narrow field-of-view. Thus by defining some relations between the two projections both can be employed in a single algorithm. The projection which provides the best result will depend on the field of view.

3.3.1 Equations relating parallel and perspective projections

A series of equations which relate the two models will now be developed. These are based on material presented by Ono and Hattori (2002) and Morgan (2004). The perspective projection equations will be considered first. From equation (1.1) the distance d_{XYZ} in the *z*-direction in 3D space from (*Xc*, *Yc*, *Zc*) to a target point (*X*, *Y*, *Z*) is:

$$d_{XYZ} = (Xc - X)m_{31} + (Yc - Y)m_{32} + (Zc - Z)m_{33} \tag{3.33}$$

This is the length of the vector from (*X*,*Y*,*Z*) to (*Xc*,*Yc*,*Zc*) projected onto the *z*-axis. Now, if there is a set of *p* points in 3D space $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), ..., (X_p, Y_p, Z_p)$, let (X_c, Y_c, Z_c) denote the centroid of all of these *p* points:

$$(X_c, Y_c, Z_c) = \left(\frac{X_1 + X_2 + ... + X_p}{p}, \frac{Y_1 + Y_2 + ... + Y_p}{p}, \frac{Z_1 + Z_2 + ... + Z_p}{p} \right) \tag{3.34}$$

Then the distance (denoted as d_{avg}) from (*Xc*, *Yc*, *Zc*) to the centroid of points (X_c, Y_c, Z_c) in the *z*-direction is:

$$d_{avg} = (Xc - X_c)m_{31} + (Yc - Y_c)m_{32} + (Zc - Z_c)m_{33} \tag{3.35}$$

With some manipulation it can be shown that the distance in the *z*-direction to the centroid (d_{avg}) is equal to the average of distances $d_1, d_2, ..., d_p$ in the *z*-direction to each point. Now, an image scale *s* is defined by considering equation (A.39) for the centroid (X_c, Y_c, Z_c); a similar equation is used by Kyle (2004).

$$\begin{aligned} s = \lambda(X_c, Y_c, Z_c) &= \frac{-f}{m_{31}(X_c - Xc) + m_{32}(Y_c - Yc) + m_{33}(Z_c - Zc)} = ... \\ ... &= \frac{f}{m_{31}(Xc - X_c) + m_{32}(Yc - Y_c) + m_{33}(Zc - Z_c)} = \frac{f}{d_{avg}} \end{aligned} \tag{3.36}$$

Note that this image scale s is the ratio of the principal distance to the average distance to the target in the z-direction. In Figures 3.2, 3.4, and 3.5 this was the factor used to rescale the parallel projection image on the "average plane" to the image plane.

Now consider equations (3.2) and (A.40); these two equations begin to show the strong resemblance between the parallel projection and the perspective projection:

$$\begin{bmatrix} x-\Delta x \\ y-\Delta y \\ * \end{bmatrix} = s\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \text{(parallel)}$$
$$\begin{bmatrix} x-xo \\ y-yo \\ -f \end{bmatrix} = \lambda(X,Y,Z)\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} X-Xc \\ Y-Yc \\ Z-Zc \end{bmatrix} \quad \text{(perspective)} \tag{3.37}$$

If the distance to the target in the z-direction is large with respect to the depth of the target it follows that $\lambda(X,Y,Z) \approx \lambda(X_c,Y_c,Z_c) = s$ for all (X, Y, Z). Then by substituting into the second equation of (3.37):

$$\begin{bmatrix} x-xo \\ y-yo \\ -f \end{bmatrix} = \lambda(X,Y,Z)\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} X-Xc \\ Y-Yc \\ Z-Zc \end{bmatrix} \approx s\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}\begin{bmatrix} X-Xc \\ Y-Yc \\ Z-Zc \end{bmatrix} \tag{3.38}$$

With some manipulation it can easily be shown from equation (3.38) that:

$$\begin{aligned} x &\approx s(m_{11}X + m_{12}Y + m_{13}Z) - s(m_{11}Xc + m_{12}Yc + m_{13}Zc) + xo \\ y &\approx s(m_{21}X + m_{22}Y + m_{23}Z) - s(m_{21}Xc + m_{22}Yc + m_{23}Zc) + yo \end{aligned} \tag{3.39}$$

From the above it follows that:

$$\begin{aligned} x &\approx A_1X + A_2Y + A_3Z + A_4 \\ y &\approx A_5X + A_6Y + A_7Z + A_8 \end{aligned} \tag{3.40}$$

$$\begin{aligned} A_1 &= sm_{11} \quad A_2 = sm_{12} \quad A_3 = sm_{13} \qquad A_4 = -s(m_{11}Xc + m_{12}Yc + m_{13}Zc) + xo \\ A_5 &= sm_{21} \quad A_6 = sm_{22} \quad A_7 = sm_{23} \qquad A_8 = -s(m_{21}Xc + m_{22}Yc + m_{23}Zc) + yo \end{aligned} \tag{3.41}$$

These equations (3.40 and 3.41) show the strong relation between the perspective projection and parallel projection models when the distance to the target is large in comparison to the depth of the target. These equations also provide a means of solving for the parallel projection parameters $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$ in terms of the EOP and IOP $(Xc, Yc, Zc, \omega, \varphi, \kappa, f, xo, yo)$.

It is noted also that if the parallel projection parameters $(s, \omega, \varphi, \kappa, \Delta x, \Delta y)$ have been solved for and the approximate IOP (f, xo, yo) are known, then the approximate coordinates of the perspective center (Xc, Yc, Zc) can be solved for. Because there is no true perspective center in the parallel projection, the point which lies a distance f along the focal axis (z-axis) in the positive direction from (xo, yo) can be used as the perspective center. In the image coordinate system, the coordinates of this point are $(xo - \Delta x, yo - \Delta y, f)$. Now, the coordinates of this point are converted into the 3D coordinate system to get the perspective center:

$$\begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix} = \frac{1}{s} \begin{bmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} \begin{bmatrix} xo - \Delta x \\ yo - \Delta y \\ f \end{bmatrix} \tag{3.42}$$

Thus, approximations of the perspective projection EOP can be found from the parallel projection parameters and approximate IOP.

3.4 Perspective-Corrected Parallel Projection

The perspective-corrected parallel projection model is based on both the parallel projection and the perspective projection and further bridges the gap between the two models. The model developed here is very similar to the orthogonal projection model

developed by Ono and Hattori (2002). The mathematical derivation of the perspective-corrected parallel projection model which follows is based on the same set of assumptions made in the derivation of the orthogonal projection model. While the set of assumptions underlying both models are essentially the same, some of the resulting equations and nomenclature are different. Also, in the implementation of the perspective-corrected parallel projection model, approximations of the EOP, the IOP, and 3D coordinates will be used to perspective-correct the image-coordinates. In the orthogonal projection model it is assumed that ground control is available in order to make this correction.

The perspective-corrected parallel projection model transforms the initial measured image coordinates (x, y) to perspective-corrected coordinates denoted by (xp, yp). This adjustment for perspective transforms the image from a central perspective image to an orthogonal image. Then the perspective-corrected image can be analyzed using the parallel projection model without losing any of the precision of the perspective model. This hybrid model has the potential for accuracy at a large range of imaging distances and over a wide range of fields of view.

3.4.1 Perspective-corrected parallel projection equations

The fundamental equations for the perspective-corrected parallel projection are derived here. Begin with the top two rows of equation (A.40):

$$\begin{bmatrix} x - xo \\ y - yo \end{bmatrix} = \lambda(X, Y, Z) \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X - Xc \\ Y - Yc \\ Z - Zc \end{bmatrix} \tag{3.43}$$

Now, multiply both sides of equation (3.43) by $\frac{s}{\lambda(X,Y,Z)}$:

$$\frac{s}{\lambda(X,Y,Z)}\begin{bmatrix} x-xo \\ y-yo \end{bmatrix} = s\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix}\begin{bmatrix} X-Xc \\ Y-Yc \\ Z-Zc \end{bmatrix} \tag{3.44}$$

The perspective-corrected image coordinates (xp, yp) are defined as follows:

$$\begin{bmatrix} xp-xo \\ yp-yo \end{bmatrix} = \frac{s}{\lambda(X,Y,Z)}\begin{bmatrix} x-xo \\ y-yo \end{bmatrix} = s\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix}\begin{bmatrix} X-Xc \\ Y-Yc \\ Z-Zc \end{bmatrix} \tag{3.45}$$

Recall that:

$$\lambda(X,Y,Z) = \frac{-f}{m_{31}(X-Xc)+m_{32}(Y-Yc)+m_{33}(Z-Zc)} \tag{3.46}$$

Now, by manipulating (3.45):

$$\begin{bmatrix} xp \\ yp \end{bmatrix} = s\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} - s\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix}\begin{bmatrix} Xc \\ Yc \\ Zc \end{bmatrix} + \begin{bmatrix} xo \\ yo \end{bmatrix} \tag{3.47}$$

Which reduces to:

$$\begin{aligned} xp &= A_1X + A_2Y + A_3Z + A_4 \\ yp &= A_5X + A_6Y + A_7Z + A_8 \end{aligned} \tag{3.48}$$

where:

$$\begin{aligned} A_1 &= sm_{11} \quad A_2 = sm_{12} \quad A_3 = sm_{13} \qquad A_4 = -s(m_{11}Xc + m_{12}Yc + m_{13}Zc) + xo \\ A_5 &= sm_{21} \quad A_6 = sm_{22} \quad A_7 = sm_{23} \qquad A_8 = -s(m_{21}Xc + m_{22}Yc + m_{23}Zc) + yo \end{aligned} \tag{3.49}$$

These equations (3.45 – 3.49) constitute the perspective-corrected parallel projection model. It follows the same form as the simple parallel projection model, but

the image coordinates (xp, yp) have been corrected from the original image coordinates (x, y) such that none of the precision of the perspective projection is lost.

Thus, if the perspective projection EOP and IOP $(Xc, Yc, Zc, \omega, \varphi, \kappa, f, xo, yo)$ and the 3D coordinates of each point (X,Y,Z) are known (or adequate approximations), the image coordinates (x, y) can be transformed into perspective-corrected coordinates (xp, yp) by equation (3.45) and the images can be analyzed with the perspective-corrected equations (3.48 and 3.49). The benefit of this is that the perspective-corrected parallel projection equations are simpler than the collinearity equations.

In sections 5.6 and 5.7 this will be explored further. In particular, the parallel projection will be used to generate initial approximations for the XYZ coordinates and EOP; then these initial approximations will be used to do a perspective-correction. The perspective-corrected coordinates lead to improved approximations. These can then serve as approximations to the standard perspective projection model.

3.4.2 3D reconstruction by perspective-corrected parallel projection

The perspective-corrected parallel projection 3D reconstruction is carried out in much the same manner as for the parallel projection. For a point in 3D space, if the image coordinates of the point are collected in I images for which the perspective-corrected parallel projection parameters are known, then the perspective-corrected image coordinates can be calculated with equation (3.45). Next, equation (3.50) can be set up which is of the form $L = AX$ and has the least-squares solution $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. This gives the 3D coordinates of a point by perspective-corrected parallel projection.

$$\begin{bmatrix} xp_A - \Delta x_A \\ yp_A - \Delta y_A \\ xp_B - \Delta x_B \\ yp_B - \Delta y_B \\ \cdots \\ xp_I - \Delta x_I \\ yp_I - \Delta y_I \end{bmatrix} = \begin{bmatrix} s_A m_{11,A} & s_A m_{12,A} & s_A m_{13,A} \\ s_A m_{21,A} & s_A m_{22,A} & s_A m_{23,A} \\ s_B m_{11,B} & s_B m_{12,B} & s_B m_{13,B} \\ s_B m_{21,B} & s_B m_{22,B} & s_B m_{23,B} \\ \cdots & \cdots & \cdots \\ s_I m_{11,I} & s_I m_{12,I} & s_I m_{13,I} \\ s_I m_{21,I} & s_I m_{22,I} & s_I m_{23,I} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.50}$$

In order to determine the residuals and back-projected image coordinates, two steps have to be performed. First, the back-projected perspective-corrected image coordinates (*xp*', *yp*') are found by:

$$\begin{bmatrix} xp' \\ yp' \end{bmatrix} = s \begin{bmatrix} m_1 & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{3.51}$$

Note that these coordinates are in the perspective-corrected image coordinate system. In order to compute residuals in the original image coordinate system, the perspective-corrected back-projected image coordinates must be transformed into the original image coordinate system. The back-projected image coordinates in the original image coordinate system (*x*', *y*') are calculated by the inverse of the perspective-correction equation:

$$\begin{bmatrix} x' - xo \\ y' - yo \end{bmatrix} = \frac{\lambda(X, Y, Z)}{s} \begin{bmatrix} xp' - xo \\ yp' - yo \end{bmatrix} \tag{3.52}$$

Where:

$$\lambda(X, Y, Z) = \frac{-f}{m_{31}(X - Xc) + m_{32}(Y - Yc) + m_{33}(Z - Zc)} \tag{3.53}$$

Then the back-projected image coordinates (x', y') can be used to find the residuals in the same manner as for the parallel projection (equations 3.7). The 1-sigma errors are also found in the same manner as for the parallel projection (equations 3.10 – 3.15).

CHAPTER 4

INITIAL APPROXIMATIONS BY PARALLEL PROJECTION

This chapter contains the rigorous mathematical derivation of the equations used to obtain initial approximations for 3D reconstruction of a target object using surveillance images. The first section of this chapter contains the derivation of equations for an algorithm which is an extension of the work of Huang and Lee (1989). The resulting algorithm will solve for the initial approximations of the EOP based on point correspondences alone. The second section contains the equations for an algorithm which is similar to the algorithm presented by Kyle (2004). This algorithm will solve for an initial approximation of the orientation of a single image relative to a set of 3D coordinates. Much of the material presented here will be referenced in Chapter 5 where the comprehensive algorithm is outlined.

4.1 Equations for Initial Approximations by Parallel Projection

In practice, the bundle triangulation algorithm will be the final step in the 3D reconstruction process. However, initial approximations of the EOP and the 3D coordinates need to be obtained before the bundle triangulation algorithm can be carried out. As noted above, the algorithms found in the literature (Ono and Hattori, 2002; Kyle, 2004; Morgan, 2004) rely on ground control points to obtain initial approximations for the EOP. Because it is desired to perform 3D reconstruction in the absence of ground control, algorithms which use only point correspondences will provide the initial approximations for the bundle triangulation algorithm.

The essential matrix algorithm outlined earlier is a method based on perspective geometry that has been used to obtain initial approximations from point correspondences alone. However, as shown by the tests in Chapter 2, there are two primary drawbacks to this algorithm. First, the perspective geometry can become unstable when working with images that have a narrow angular field of view. Secondly, the essential matrix algorithm is only capable of utilizing point correspondences across two images—only the relative orientation of two images is solved for.

The equations presented in this chapter are the fundamental equations used in an algorithm which will solve for initial approximations of *I* images simultaneously using parallel projection. The resulting algorithm will provide a robust alternative to the essential matrix algorithm. The equations (4.4 – 4.76) are an extension of the work by Huang and Lee (1989) developed for computer vision applications. Many of the equations found there are repeated here though some of the notation has been altered. Also, several of the steps of the algorithm have been altered and extended. The step for solving for relative shift of images has been modified (equations 4.4 – 4.9), a step to solve for the relative scales of each image has been added (equations 4.10 – 4.31), and the relative rotation step (equations 4.32 – 4.71) has been expanded to accommodate more than three images.

4.1.1 Assumptions

The algorithm presented below follows a set of assumptions. It will be assumed that there are *I* images (image *A*, image *B*,…, image *I*) with unknown orientations. The orientations are denoted by equation (4.1).

$$\begin{matrix}(s_A, \omega_A, \varphi_A, \kappa_A, \Delta x_A, \Delta y_A) \\ (s_B, \omega_B, \varphi_B, \kappa_B, \Delta x_B, \Delta y_B) \\ \ldots \\ (s_I, \omega_I, \varphi_I, \kappa_I, \Delta x_I, \Delta y_I)\end{matrix} \tag{4.1}$$

For each image, these six unknowns describe a scaled-shifted rotation. In the development of the required equations which follows, each of these three steps (shift, scale, and rotation) will be considered individually. It is necessary to have at least three images so it is assumed $I \geq 3$.

It is also assumed that there are *n* points with unknown 3D coordinates:

$$(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), \ldots, (X_n, Y_n, Z_n) \tag{4.2}$$

It is assumed further that the set of point correspondences of the *n* points in *I* images have been measured and are given as:

$$\begin{matrix}(x_{A,1}, y_{A,1}) & (x_{B,1}, y_{B,1}) & \ldots & (x_{I,1}, y_{I,1}) \\ (x_{A,2}, y_{A,2}) & (x_{B,2}, y_{B,2}) & \ldots & (x_{I,2}, y_{I,2}) \\ \ldots & \ldots & \ldots & \ldots \\ (x_{A,n}, y_{A,n}) & (x_{B,n}, y_{B,n}) & \ldots & (x_{I,n}, y_{I,n})\end{matrix} \tag{4.3}$$

For this algorithm all *n* points do not need to be visible in all *I* images—however, it is necessary that each point is visible in at least two images. For this algorithm it is also necessary that at least one point is visible in all images in order to calculate the relative shift of the images. It is also necessary that when considering any two images, there are at least four common points on those two images. While these constraints may seem limiting, it will be shown later that these constraints will only have to be satisfied for three of the images—subsequent images can be added to the solution after the initial approximations are obtained using the method outlined in section 4.2.

It is not necessary to know or approximate the IOP.

It is necessary to input a scale. If the approximate distance to one of the images is known, then this scale is equal to the approximate principal distance divided by the approximate distance (see equation 3.36). If this information is not known, an arbitrary scale, such as 1 can be chosen.

It is also necessary to choose a keypoint. The keypoint is a point in the first image which is closer to the camera than the other points. This is discussed in more detail later.

As was the case in the bundle triangulation algorithm, there is no absolute orientation for 3D space to begin with, so the rotation, origin, and scale of the 3D coordinate system will have to be defined. The origin of 3D space will be defined in the shift step of the algorithm by selecting a point and assigning it to be the origin of 3D space. The scale of 3D space will be set by "fixing" the scale of one of the images (image *A*) during the scale step of the algorithm. The rotation of 3D space will be defined by choosing the rotation angles of image *A* to be 0 (i.e. image *A* will have no rotation with respect to 3D space).

4.1.2 Approximating the shift

This portion of the algorithm is valid for the general affine projection model. Hence, the equations here are expressed in terms of the affine projection parameters $(A_1, A_2, ..., A_8)$. First the centroid of the set of points which appear in all *I* images is defined as the origin of 3D space. In the algorithm by Huang and Lee (1989) a single point which is visible in all images is chosen as the origin. Some advantages of choosing the centroid of points are discussed in (Kyle, 2004).

Suppose that there are *p* points in 3D space (X_1,Y_1,Z_1), (X_2,Y_2,Z_2), … (X_p,Y_p,Z_p) which appear in all *I* images. This is possible due to the assumption there must be at least one point which is visible in all images. The centroid of these *p* points in 3D space will be denoted as (X_c,Y_c,Z_c). Note that (X_c,Y_c,Z_c) is not the same as (*Xc*,*Yc*,*Zc*) which is used to denote the perspective center of an image. Then define this point as the origin of 3D space:

$$(X_c,Y_c,Z_c)=\left(\frac{X_1+X_2+...+X_p}{p},\frac{Y_1+Y_2+...+Y_p}{p},\frac{Z_1+Z_2+...+Z_p}{p}\right)=(0,0,0) \tag{4.4}$$

Let $(x_{K,c}, y_{K,c})$ denote the image coordinates of the centroid (X_c,Y_c,Z_c) in image *K*. Now, the parallel projection equations for the centroid reduce to:

$$\begin{aligned} x_{K,c} &= A_{K,1}X_c + A_{K,2}Y_c + A_{K,3}Z_c + A_{K,4} = A_{K,1}0 + A_{K,2}0 + A_{K,3}0 + A_{K,4} = \Delta x_K \\ y_{K,c} &= A_{K,5}X_c + A_{K,6}Y_c + A_{K,7}Z_c + A_{K,8} = A_{K,5}0 + A_{K,6}0 + A_{K,7}0 + A_{K,8} = \Delta y_K \end{aligned} \tag{4.5}$$

Recall that the fourth and eighth affine projection parameters $(A_{K,4}, A_{K,8})$ are equal to the parallel projection shift terms $(\Delta x_K, \Delta y_K)$. Equations (4.5) show that the image coordinates of the centroid $(x_{K,c}, y_{K,c})$ are equal to the shift terms that are sought. Note however that the centroid of 3D space is a mathematically calculated point and it cannot be directly observed in the images. So these image coordinates must be calculated. By substituting equation (4.4) into equations (4.5) the following is obtained:

$$\begin{aligned} x_{K,c} &= A_{K,1}\left(\frac{X_1+...+X_p}{p}\right) + A_{K,2}\left(\frac{Y_1+...+Y_p}{p}\right) + A_{K,3}\left(\frac{Z_1+...+Z_p}{p}\right) + A_{K,4} \\ y_{K,c} &= A_{K,5}\left(\frac{X_1+...+X_p}{p}\right) + A_{K,6}\left(\frac{Y_1+...+Y_p}{p}\right) + A_{K,7}\left(\frac{Z_1+...+Z_p}{p}\right) + A_{K,8} \end{aligned} \tag{4.6}$$

By rearranging and gathering like terms the result is:

$$\begin{aligned} x_{K,c} &= \frac{1}{p}\left(\left(A_{K,1}X_1 + A_{K,2}Y_1 + A_{K,3}Z_1\right) + \ldots + \left(A_{K,1}X_p + A_{K,2}Y_p + A_{K,3}Z_p\right)\right) \\ y_{K,c} &= \frac{1}{p}\left(\left(A_{K,5}X_1 + A_{K,6}Y_1 + A_{K,7}Z_1\right) + \ldots + \left(A_{K,5}X_p + A_{K,6}Y_p + A_{K,7}Z_p\right)\right) \end{aligned} \tag{4.7}$$

which reduces to:

$$\begin{aligned} x_{K,c} &= \frac{1}{p}\left(x_{K,1} + x_{K,2} + \ldots + x_{K,p}\right) \\ y_{K,c} &= \frac{1}{p}\left(y_{K,1} + y_{K,2} + \ldots + y_{K,p}\right) \end{aligned} \tag{4.8}$$

Note that this is the centroid of the *p* image coordinates.

Equations (4.5 and 4.8) show that the shift terms $\left(\Delta x_K, \Delta y_K\right)$ can be found by taking the centroid of the image coordinates of the *p* points. In other words, the centroid of the image coordinates is equal to the image coordinates of the centroid of the 3D coordinates. Thus:

$$\left(x_{K,c}, y_{K,c}\right) = \left(\frac{x_{K,1} + \ldots + x_{K,p}}{p}, \frac{y_{K,1} + \ldots + y_{K,p}}{p}\right) = \left(\Delta x_K, \Delta y_K\right) \tag{4.9}$$

At this point, the origin of 3D space has been defined and an approximation of the shift of the images with respect to 3D space has been obtained.

4.1.3 Approximating the scale

Now that the shift of each image with respect to 3D space has been approximated, the next step is to determine the scale of each image. The following equations (4.10 – 4.20) are very similar to those outlined by derivations in Morgan (2004) and Huang and Lee (1989). Because the shift has been accounted for, the remaining terms describe a

scaled rotation. Because each image has a unique scale, an equation can be written in the form of equation (3.2) for each image. Consider two images, image *J* and image *K*:

$$\begin{bmatrix} x_J - \Delta x_J \\ y_J - \Delta y_J \\ * \end{bmatrix} = s_J \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$
$$\begin{bmatrix} x_K - \Delta x_K \\ y_K - \Delta y_K \\ * \end{bmatrix} = s_K \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.10}$$

Since an absolute orientation of 3D space has not been defined, 3D space can be defined to have any orientation. If 3D space is temporarily defined to have no rotation with respect to image *J* (i.e. $\omega_J = 0, \varphi_J = 0, \kappa_J = 0$) then the first equation of (4.10) becomes:

$$\begin{bmatrix} x_J - \Delta x_J \\ y_J - \Delta y_J \\ * \end{bmatrix} = s_J \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.11}$$

Equation (4.11) can be rewritten as:

$$\begin{bmatrix} x_J - \Delta x_J \\ y_J - \Delta y_J \end{bmatrix} = s_J \begin{bmatrix} X \\ Y \end{bmatrix} \tag{4.12}$$

And to have a compatible form, the bottom equation of (4.10) can be rewritten as:

$$\begin{bmatrix} x_K - \Delta x_K \\ y_K - \Delta y_K \end{bmatrix} = s_K \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix} + s_K \begin{bmatrix} k_{13} \\ k_{23} \end{bmatrix} Z \tag{4.13}$$

If $\begin{bmatrix} X \\ Y \end{bmatrix}$ is eliminated from these two equations (4.12 and 4.13):

$$\begin{bmatrix} x_K - \Delta x_K \\ y_K - \Delta y_K \end{bmatrix} = \frac{s_K}{s_J} \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \begin{bmatrix} x_J - \Delta x_J \\ y_J - \Delta y_J \end{bmatrix} + s_K \begin{bmatrix} k_{13} \\ k_{23} \end{bmatrix} Z \tag{4.14}$$

Then if the top and bottom equation of (4.14) are combined and Z is eliminated:

$$\begin{aligned} &k_{13}\left(y_K-\Delta y_K\right)=... \\ &...k_{23}\left(x_K-\Delta x_K\right)+\frac{s_K}{s_J}\left(k_{13}k_{21}-k_{11}k_{23}\right)\left(x_J-\Delta x_J\right)+\frac{s_K}{s_J}\left(k_{22}k_{13}-k_{12}k_{23}\right)\left(y_J-\Delta y_J\right) \end{aligned} \tag{4.15}$$

Because the cross-product of any two rows (columns) of a rotation matrix is equal to the other row (column) (see equation A.3 in the appendix):

$$\begin{aligned} \left(k_{13}k_{21}-k_{11}k_{23}\right)&=k_{32} \\ \left(k_{13}k_{22}-k_{12}k_{23}\right)&=-k_{31} \end{aligned} \tag{4.16}$$

If these (4.16) are substituted into equations (4.15):

$$k_{13}\left(y_K-\Delta y_K\right)=k_{23}\left(x_K-\Delta x_K\right)+\frac{s_K}{s_J}k_{32}\left(x_J-\Delta x_J\right)-\frac{s_K}{s_J}k_{31}\left(y_J-\Delta y_J\right) \tag{4.17}$$

At this point four new variables $R_1, R_2, R_3,$ and R_4 are defined:

$$R_1=k_{13} \qquad R_2=k_{23} \qquad R_3=\frac{s_K}{s_J}k_{32} \qquad R_4=\frac{s_K}{s_J}k_{31} \tag{4.18}$$

and by substituting these (4.18) into equation (4.17):

$$R_1\left(y_K-\Delta y_K\right)=R_2\left(x_K-\Delta x_K\right)+R_3\left(x_J-\Delta x_J\right)-R_4\left(y_J-\Delta y_J\right) \tag{4.19}$$

Now if a set of p point correspondences over images J and K have been collected then the following set of equations can be formed:

$$\begin{aligned} R_1\left(y_{K,1}-\Delta y_K\right)&=R_2\left(x_{K,1}-\Delta x_K\right)+R_3\left(x_{J,1}-\Delta x_J\right)-R_4\left(y_{J,1}-\Delta y_J\right) \\ R_1\left(y_{K,2}-\Delta y_K\right)&=R_2\left(x_{K,2}-\Delta x_K\right)+R_3\left(x_{J,2}-\Delta x_J\right)-R_4\left(y_{J,2}-\Delta y_J\right) \\ &... \\ R_1\left(y_{K,p}-\Delta y_K\right)&=R_2\left(x_{K,p}-\Delta x_K\right)+R_3\left(x_{J,p}-\Delta x_J\right)-R_4\left(y_{J,p}-\Delta y_J\right) \end{aligned} \tag{4.20}$$

Note that this set of equations (4.20) is a linear-homogeneous system in four unknowns $\left(R_1, R_2, R_3, R_4\right)$. While this system cannot be solved uniquely (because it is

homogeneous) it can be solved to within a scale factor as long as $p \geq 4$. (Recall from earlier that it was assumed that when considering any two images there are at least four points which appear in both images.) If an arbitrary value is assigned to R_1 (for this derivation $R_1 = 1$ is chosen), R_2, R_3 and R_4 can be solved for in terms of R_1 by using a linear least-squares approach. So by taking $R_1 = 1$ equation (4.20) can be set up as a system of equations as follows:

$$\begin{bmatrix} y_{K,1} - \Delta y_K \\ y_{K,2} - \Delta y_K \\ \cdots \\ y_{K,p} - \Delta y_K \end{bmatrix} = \begin{bmatrix} x_{K,1} - \Delta x_K & x_{J,1} - \Delta x_J & -y_{J,1} + \Delta y_J \\ x_{K,2} - \Delta x_K & x_{J,2} - \Delta x_J & -y_{J,2} + \Delta y_J \\ \cdots & \cdots & \cdots \\ x_{K,p} - \Delta x_K & x_{J,p} - \Delta x_J & -y_{J,p} + \Delta y_J \end{bmatrix} \begin{bmatrix} R_2 \\ R_3 \\ R_4 \end{bmatrix} \qquad p \geq 4 \tag{4.21}$$

This equation (4.21) is of the form $L = AX$ and the least squares solution is is $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. For each pair of images R_2, R_3 and R_4 can be solved for by least squares using the above approach. So, for each pair of images there will be a R_2, R_3 and R_4. Next, note that in a rotation matrix the norms of rows and columns are equal to 1 (equation A.1), so it follows that:

$$1 - k_{33}^2 = k_{13}^2 + k_{23}^2 = k_{31}^2 + k_{32}^2 \Rightarrow R_1^2 + R_2^2 = \left(\frac{s_J}{s_K}\right)^2 \left(R_3^2 + R_4^2\right) \tag{4.22}$$

By rearranging the resulting equation (4.22) the relation between the scale of image *J* and image *K* is given by:

$$s_K^2 = s_J^2 \left(\frac{R_3^2 + R_4^2}{R_1^2 + R_2^2}\right) \tag{4.23}$$

Now, by considering all pairs of two images, there are $\binom{I}{2}$ two-image pairs. If the above process (equations 4.21 – 4.23) is carried out for each of these two-image pairs, each pair results in one equation. The result is the following system of equations:

$$\begin{aligned}
s_B^2 &= s_A^2\left(\frac{R_{3AB}^2 + R_{4AB}^2}{R_{1AB}^2 + R_{2AB}^2}\right) \\
s_C^2 &= s_A^2\left(\frac{R_{3AC}^2 + R_{4AC}^2}{R_{1AC}^2 + R_{2AC}^2}\right) \\
&\cdots \\
s_I^2 &= s_A^2\left(\frac{R_{3AI}^2 + R_{4AI}^2}{R_{1AI}^2 + R_{2AI}^2}\right) \\
s_C^2 &= s_B^2\left(\frac{R_{3BC}^2 + R_{4BC}^2}{R_{1BC}^2 + R_{2BC}^2}\right) \\
&\cdots
\end{aligned} \tag{4.24}$$

This system (4.24) contains $\binom{I}{2}$ equations in I unknowns $(s_A, s_B, ..., s_I)$. This system is homogeneous and hence it does not have a unique solution. However, a unique solution can be found by "fixing" one of the unknowns to a constant value. Recall that the scale of 3D space has not been defined and in order to fully define 3D space, a scale must be defined. The scale of 3D space is defined by fixing the value of one of the image scales—recall that this is one of the inputs of the algorithm. Without loss of generality, let this scale be the scale of image A, s_A.

By fixing the value of s_A, the number of unknowns in the system of equations (4.24) has been reduced to $(I - 1)$ and the system is no longer homogeneous. Thus all of the remaining scales can be solved for. A solution to the system is found by linearizing

equations (4.24) by a first-order Taylor Series approximation and then solving by linear least-squares. First, for ease of notation the following definition is made:

$$\left(\frac{R_{3AB}^2 + R_{4AB}^2}{R_{1AB}^2 + R_{2AB}^2}\right) = R_{AB} \tag{4.25}$$

Implicit differentiation of one of these equations (4.24) which includes s_A (remember that s_A is now a constant) yields:

$$2(s_B)dS_B = s_A^2 R_{AB} \tag{4.26}$$

By implicitly differentiating an equation (4.24) which does not include s_A (i.e. both scales in the equation are treated as unknowns):

$$2(s_C)ds_C = 2(s_B)ds_B R_{BC} \tag{4.27}$$

For this solution, an initial guess must be made for each of the unknowns $(s_B,...,s_I)$. The value of s_A will be used as an initial approximation for each of the unknown scales. This assumption is generally a valid assumption as long as all of the image coordinates are in the same units and as long as each of the images were taken at similar distances. If some of the images have coordinates which are in units that differ by several orders of magnitude from the others (i.e. some image coordinates in pixels, some image coordinates in normalized pixels), or if some images were taken at distances that are several orders of magnitude different from the others this assumption may not be valid. So, assuming the scale of image *A* as an initial approximation:

$$(s_B)_0 = s_A \quad ... \quad (s_I)_0 = s_A \tag{4.28}$$

The first-order Taylor series approximation of an equation (4.26) which contains s_A is:

$$(s_B)_0^2 + 2(s_B)_0 ds_B = s_A^2 R_{AB} \tag{4.29}$$

And the first-order Taylor series approximation of an equation (4.27) which does not contain s_A is:

$$\left(s_C\right)_0^2 + 2\left(s_C\right)_0 ds_C = \left(s_B\right)_0^2 R_{BC} + 2\left(s_B\right)_0 R_{BC} ds_B \tag{4.30}$$

By considering these equations (4.29 and 4.30) for each of the equations in (4.24) a linear system of equations in matrix form can be set up:

$$\begin{bmatrix} s_A^2 R_{AB} - \left(s_B\right)_0^2 \\ s_A^2 R_{AC} - \left(s_C\right)_0^2 \\ s_A^2 R_{AD} - \left(s_D\right)_0^2 \\ s_A^2 R_{AE} - \left(s_E\right)_0^2 \\ \dots \\ \left(s_B\right)_0^2 R_{BC} - \left(s_C\right)_0^2 \\ \left(s_B\right)_0^2 R_{BD} - \left(s_D\right)_0^2 \\ \left(s_B\right)_0^2 R_{BE} - \left(s_E\right)_0^2 \\ \dots \\ \left(s_C\right)_0^2 R_{CD} - \left(s_D\right)_0^2 \\ \left(s_C\right)_0^2 R_{CE} - \left(s_E\right)_0^2 \\ \dots \end{bmatrix} = \begin{bmatrix} 2\left(s_B\right)_0 & 0 & 0 & 0 & \dots \\ 0 & 2\left(s_C\right)_0 & 0 & 0 & \dots \\ 0 & 0 & 2\left(s_D\right)_0 & 0 & \dots \\ 0 & 0 & 0 & 2\left(s_E\right)_0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ -2\left(s_B\right)_0 R_{BC} & 2\left(s_C\right)_0 & 0 & 0 & \dots \\ -2\left(s_B\right)_0 R_{BD} & 0 & 2\left(s_D\right)_0 & 0 & \dots \\ -2\left(s_B\right)_0 R_{BE} & 0 & 0 & 2\left(s_E\right)_0 & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & -2\left(s_C\right)_0 R_{CD} & 2\left(s_D\right)_0 & 0 & \dots \\ 0 & -2\left(s_C\right)_0 R_{CE} & 0 & 2\left(s_E\right)_0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} ds_B \\ ds_C \\ ds_D \\ ds_E \\ \dots \end{bmatrix} \tag{4.31}$$

This equation is of the form $L = AX$ and has the least squares solution $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. The values of *X* are added to the initial guesses to achieve improved approximations. The improved approximations are then used to solve for the system again, resulting in even better approximations. This process is repeated until the magnitudes of the correction terms *X* are below a desired level. This gives the approximation of the image scales.

4.1.4 Approximating the relative rotations

Now that the shifts and scales of each of the images have been approximated, the rotation of the images with respect to each other will be approximated. In a later step, the

rotation of 3D space with respect to one of the images will be defined and the relative rotations of images will be used to find their rotation with respect to 3D space.

The first step is to calculate the shifted-scaled image coordinates $(\hat{x}, \hat{y})$ for each image. For image *J* these are calculated by:

$$\begin{bmatrix} \hat{x}_J \\ \hat{y}_J \end{bmatrix} = \frac{1}{s_J} \begin{bmatrix} x_J - \Delta x_J \\ y_J - \Delta y_J \end{bmatrix} \tag{4.32}$$

In equation (4.32) the parallel projection equations are written in terms of shifted-scaled image coordinates following the form in equation (3.5). As was the case when solving for approximations of the image scales, two images will be considered, image *J* and image *K*:

$$\begin{bmatrix} \hat{x}_J \\ \hat{y}_J \\ * \end{bmatrix} = \begin{bmatrix} j_{11} & j_{12} & j_{13} \\ j_{21} & j_{22} & j_{23} \\ j_{31} & j_{32} & j_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$
$$\begin{bmatrix} \hat{x}_K \\ \hat{y}_K \\ * \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{4.33}$$

By eliminating $\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$:

$$\begin{bmatrix} \hat{x}_K \\ \hat{y}_K \\ * \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} j_{11} & j_{21} & j_{31} \\ j_{12} & j_{22} & j_{32} \\ j_{13} & j_{23} & j_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_J \\ \hat{y}_J \\ * \end{bmatrix} \tag{4.34}$$

Equation (4.34) describes the rotation of image *J* relative to image *K*. Because the multiplication of two rotation matrices is itself a rotation matrix a new rotation matrix $[jk]$ is defined by equation (4.35). The left side the equation (4.35) is the matrix $[jk]$

which describes the rotation from image *J* to image *K*. This notation will be used for all two-image matrices, i.e. for any two images $[jk]=[k][j]^T$.

$$\begin{bmatrix} jk_{11} & jk_{12} & jk_{13} \\ jk_{21} & jk_{22} & jk_{23} \\ jk_{31} & jk_{32} & jk_{33} \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} j_{11} & j_{21} & j_{31} \\ j_{12} & j_{22} & j_{32} \\ j_{13} & j_{23} & j_{33} \end{bmatrix} \tag{4.35}$$

At this point it is important to note that because the rotation of 3D space has not been defined the matrices $([ab],[ac],...[ai],[bc],..,[bi],...)$ which describe the relative rotation of images will be solved for, and not the matrices $([a],[b],...,[i])$ which describe the rotation of images with respect to 3D space. So by substituting (4.35) into (4.34):

$$\begin{bmatrix} \hat{x}_K \\ \hat{y}_K \\ * \end{bmatrix} = \begin{bmatrix} jk_{11} & jk_{12} & jk_{13} \\ jk_{21} & jk_{22} & jk_{23} \\ jk_{31} & jk_{32} & jk_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_J \\ \hat{y}_J \\ * \end{bmatrix} \tag{4.36}$$

The top two equations of this matrix multiplication are combined by eliminating the third element (*) of the vector on the right side:

$$jk_{13}\hat{y}_K = jk_{23}\hat{x}_K + (jk_{13}jk_{21} - jk_{11}jk_{23})\hat{x}_J + (jk_{13}jk_{22} - jk_{12}jk_{23})\hat{y}_J \tag{4.37}$$

Note that this derivation is entirely analogous to the derivation that was used to determine the image scales (equations 4.14 – 4.17). By the properties of a rotation matrix this reduces to:

$$(jk_{13})\hat{y}_K = (jk_{23})\hat{x}_K + (jk_{32})\hat{x}_J - (jk_{31})\hat{y}_J \tag{4.38}$$

If *p* point correspondences over these two images are considered then a system of equations is obtained:

$$\begin{aligned}
(jk_{13})\hat{y}_{K,1} &= (jk_{23})\hat{x}_{K,1} + (jk_{32})\hat{x}_{J,1} - (jk_{31})\hat{y}_{J,1} \\
(jk_{13})\hat{y}_{K,2} &= (jk_{23})\hat{x}_{K,2} + (jk_{32})\hat{x}_{J,2} - (jk_{31})\hat{y}_{J,2} \\
&\ldots \\
(jk_{13})\hat{y}_{K,p} &= (jk_{23})\hat{x}_{K,p} + (jk_{32})\hat{x}_{J,p} - (jk_{31})\hat{y}_{J,p}
\end{aligned} \tag{4.39}$$

As before, this is a linear-homogeneous system which cannot be solved uniquely, but can be solved to within a scale factor given $p \geq 4$. (Recall that it was assumed that there are at least four point correspondences on each set of two images.) An arbitrary value is assigned to jk_{13} (denoted by jk_{13}'; for simplicity $jk_{13}' = 1$ is chosen) and then jk_{23}', jk_{31}', and jk_{32}' are solved for in terms of jk_{13}'. This linear least-squares solution is found in the same manner as equations (4.20 and 4.21).

The actual values of $jk_{13}, jk_{23}, jk_{31}$, and jk_{32} are related to the arbitrary values $jk_{13}', jk_{23}', jk_{31}'$, and jk_{32}' by the matrix scale factor $\xi\kappa$ (recall that the arbitrary values were solved for assuming $jk_{13}' = 1$). Thus:

$$(jk_{13}, jk_{23}, jk_{31}, jk_{32}) = \xi\kappa(jk_{13}', jk_{23}', jk_{31}', jk_{32}') \tag{4.40}$$

So $\xi\kappa$ must be determined in order to calculate the actual terms of the rotation matrix from the arbitrary terms.

It has been proven by Huang and Lee (1989) that when using parallel projection, two images are insufficient to determine the relative orientation of images—three images are necessary. It has also been shown that when a parallel projection is used, much higher precision is obtained when using more than two images (Ono and Hattori, 2002). Thus, a series of equations will be derived by considering three images at a time and then these equations will be applied to all three-image triplets from the set of *I* images.

Consider three images (say image *A*, image *B*, and image *C*). By following the above, the relations between images are (like equation 4.36):

$$\begin{bmatrix} \hat{x}_B \\ \hat{y}_B \\ * \end{bmatrix} = \begin{bmatrix} ab_{11} & ab_{12} & ab_{13} \\ ab_{21} & ab_{22} & ab_{23} \\ ab_{31} & ab_{32} & ab_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_A \\ \hat{y}_A \\ * \end{bmatrix}$$
$$\begin{bmatrix} \hat{x}_C \\ \hat{y}_C \\ * \end{bmatrix} = \begin{bmatrix} ac_{11} & ac_{12} & ac_{13} \\ ac_{21} & ac_{22} & ac_{23} \\ ac_{31} & ac_{32} & ac_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_A \\ \hat{y}_A \\ * \end{bmatrix} \tag{4.41}$$
$$\begin{bmatrix} \hat{x}_C \\ \hat{y}_C \\ * \end{bmatrix} = \begin{bmatrix} bc_{11} & bc_{12} & bc_{13} \\ bc_{21} & bc_{22} & bc_{23} \\ bc_{31} & bc_{32} & bc_{33} \end{bmatrix} \begin{bmatrix} \hat{x}_B \\ \hat{y}_B \\ * \end{bmatrix}$$

Note that the rotation from image *A* to image *C* is equal to the rotation from image *A* to image *B* multiplied by the rotation from image *B* to image *C*:

$$\begin{bmatrix} ac_{11} & ac_{12} & ac_{13} \\ ac_{21} & ac_{22} & ac_{23} \\ ac_{31} & ac_{32} & ac_{33} \end{bmatrix} = \begin{bmatrix} bc_{11} & bc_{12} & bc_{13} \\ bc_{21} & bc_{22} & bc_{23} \\ bc_{31} & bc_{32} & bc_{33} \end{bmatrix} \begin{bmatrix} ab_{11} & ab_{12} & ab_{13} \\ ab_{21} & ab_{22} & ab_{23} \\ ab_{31} & ab_{32} & ab_{33} \end{bmatrix} \tag{4.42}$$

As before, consider two images at a time and use equations (4.39) to calculate $(ab_{13}', ab_{23}', ab_{31}', ab_{32}'), (ac_{13}', ac_{23}', ac_{31}', ac_{32}')$ and $(bc_{13}', bc_{23}', bc_{31}', bc_{32}')$. This requires three least-squares systems. In doing so, three unknown matrix scale factors $\alpha\beta, \alpha\gamma,$ and $\beta\gamma$ are introduced. These are related by (like equation 4.40):

$$\begin{aligned} (ab_{13}, ab_{23}, ab_{31}, ab_{32}) &= \alpha\beta(ab_{13}', ab_{23}', ab_{31}', ab_{32}') \\ (ac_{13}, ac_{23}, ac_{31}, ac_{32}) &= \alpha\gamma(ac_{13}', ac_{23}', ac_{31}', ac_{32}') \\ (bc_{13}, bc_{23}, bc_{31}, bc_{32}) &= \beta\gamma(bc_{13}', bc_{23}', bc_{31}', bc_{32}') \end{aligned} \tag{4.43}$$

To find the true values of the rotation matrix terms the unknown matrix scale factors $\alpha\beta, \alpha\gamma,$ and $\beta\gamma$ must be found. By considering the constraint equation (4.42) two relationships for the matrix scale factors can be developed. First multiply out equation

(4.42) to yield four equations (one equation for the upper left 2 x 2 portion of the matrix, one equation for the top two terms in the third column, one equation for the first two terms in the third row, and one equation for the third term in the third column):

$$\begin{bmatrix} ac_{11} & ac_{12} \\ ac_{21} & ac_{22} \end{bmatrix} = \begin{bmatrix} bc_{11} & bc_{12} \\ bc_{21} & bc_{22} \end{bmatrix} \begin{bmatrix} ab_{11} & ab_{12} \\ ab_{21} & ab_{22} \end{bmatrix} + \begin{bmatrix} bc_{13} \\ bc_{23} \end{bmatrix} \begin{bmatrix} ab_{31} & ab_{32} \end{bmatrix} \tag{4.44}$$

$$\begin{bmatrix} ac_{13} \\ ac_{23} \end{bmatrix} = \begin{bmatrix} bc_{11} & bc_{12} \\ bc_{21} & bc_{22} \end{bmatrix} \begin{bmatrix} ab_{13} \\ ab_{23} \end{bmatrix} + \begin{bmatrix} bc_{13} \\ bc_{23} \end{bmatrix} ab_{33} \tag{4.45}$$

$$\begin{bmatrix} ac_{31} & ac_{32} \end{bmatrix} = \begin{bmatrix} bc_{31} & bc_{32} \end{bmatrix} \begin{bmatrix} ab_{11} & ab_{12} \\ ab_{21} & ab_{22} \end{bmatrix} + bc_{33} \begin{bmatrix} ab_{31} & ab_{32} \end{bmatrix} \tag{4.46}$$

$$ac_{33} = \begin{bmatrix} bc_{31} & bc_{32} \end{bmatrix} \begin{bmatrix} ab_{13} \\ ab_{23} \end{bmatrix} + bc_{33} ab_{33} \tag{4.47}$$

Premultiply both sides of (4.45) by $\begin{bmatrix} bc_{23} & -bc_{13} \end{bmatrix}$:

$$\begin{bmatrix} bc_{23} & -bc_{13} \end{bmatrix} \begin{bmatrix} ac_{13} \\ ac_{23} \end{bmatrix} = \begin{bmatrix} bc_{23}bc_{11} - bc_{13}bc_{21} & bc_{23}bc_{12} - bc_{13}bc_{22} \end{bmatrix} \begin{bmatrix} ab_{13} \\ ab_{23} \end{bmatrix} \tag{4.48}$$

By the properties of a rotation matrix this reduces to:

$$\begin{bmatrix} bc_{23} & -bc_{13} \end{bmatrix} \begin{bmatrix} ac_{13} \\ ac_{23} \end{bmatrix} = \begin{bmatrix} -bc_{32} & bc_{31} \end{bmatrix} \begin{bmatrix} ab_{13} \\ ab_{23} \end{bmatrix} \tag{4.49}$$

Now by substituting equations (4.43):

$$\alpha\gamma\left(bc_{23}'ac_{13}' - bc_{13}'ac_{23}'\right) = \alpha\beta\left(bc_{31}'ab_{23}' - bc_{32}'ab_{13}'\right) \tag{4.50}$$

In a similar manner, postmultiply both sides of (4.48) by $\begin{bmatrix} ab_{32} \\ -ab_{31} \end{bmatrix}$ which, with some manipulation, gives:

$$\alpha\gamma\left(ac_{31}'ab_{32}' - ac_{32}'ab_{31}'\right) = \beta\gamma\left(bc_{32}'ab_{13}' - bc_{31}'ab_{23}'\right) \tag{4.51}$$

Now, rearrange equations (4.50) and (4.51) to give the relation of matrix scale factors:

$$\begin{aligned}\frac{\alpha\beta}{\alpha\gamma} &= \frac{bc_{23}'ac_{13}'-bc_{13}'ac_{23}'}{bc_{31}'ab_{23}'-bc_{32}'ab_{13}'} \\ \frac{\beta\gamma}{\alpha\gamma} &= \frac{ac_{31}'ab_{32}'-ac_{32}'ab_{31}'}{bc_{32}'ab_{13}'-bc_{31}'ab_{23}'}\end{aligned} \tag{4.52}$$

For simplification, define two new terms ABC_1 and ABC_2:

$$\begin{aligned}ABC_1 &= \frac{\alpha\beta}{\alpha\gamma} = \frac{bc_{23}'ac_{13}'-bc_{13}'ac_{23}'}{bc_{31}'ab_{23}'-bc_{32}'ab_{13}'} \\ ABC_2 &= \frac{\beta\gamma}{\alpha\gamma} = \frac{ac_{31}'ab_{32}'-ac_{32}'ab_{31}'}{bc_{31}'ab_{13}'-bc_{32}'ab_{23}'}\end{aligned} \tag{4.53}$$

(this notation will be used for all sets of three images, i.e. $IJK_1 = \frac{\iota\xi}{\iota\kappa}$).

Next premultiply both sides of (4.45) by $\begin{bmatrix} bc_{13} & bc_{23} \end{bmatrix}$ to get:

$$\begin{bmatrix} bc_{13} & bc_{23} \end{bmatrix}\begin{bmatrix} ac_{13} \\ ac_{23} \end{bmatrix} = \begin{bmatrix} bc_{13}bc_{11}+bc_{23}bc_{21} & bc_{13}bc_{12}+bc_{23}bc_{22} \end{bmatrix}\begin{bmatrix} ab_{13} \\ ab_{23} \end{bmatrix} + \begin{bmatrix} bc_{13}^2+bc_{23}^2 \end{bmatrix}ab_{33} \tag{4.54}$$

By the properties of rotation matrices this reduces to:

$$\begin{bmatrix} bc_{13} & bc_{23} \end{bmatrix}\begin{bmatrix} ac_{13} \\ ac_{23} \end{bmatrix} = \begin{bmatrix} -bc_{33}bc_{31} & -bc_{33}bc_{32} \end{bmatrix}\begin{bmatrix} ab_{13} \\ ab_{23} \end{bmatrix} + \begin{bmatrix} bc_{13}^2+bc_{23}^2 \end{bmatrix}ab_{33} \tag{4.55}$$

Then by substituting (4.43) into (4.55):

$$\left(bc_{13}'ac_{13}'+bc_{23}'ac_{23}'\right) = ABC_2\left(bc_{13}'^2+bc_{23}'^2\right)ab_{33} - ABC_1\left(bc_{31}'ab_{13}'+bc_{32}'ab_{23}'\right)bc_{33} \tag{4.56}$$

Similar to equations (4.54 – 4.56), if both sides of (4.46) are post-multiplied by $\begin{bmatrix} ab_{31} \\ ab_{32} \end{bmatrix}$:

$$\left(ac_{31}'ab_{31}'+ac_{32}'ab_{32}'\right) = -ABC_2\left(bc_{31}'ab_{13}'+bc_{32}'ab_{13}'\right)ab_{33} + ABC_1\left(ab_{31}'^2+ab_{32}'^2\right)bc_{33} \tag{4.57}$$

For simplification three additional terms are introduced:

$$\begin{aligned} ABC_3 &= bc_{13}'ac_{13}'+bc_{23}'ac_{23}' \\ ABC_4 &= bc_{31}'ab_{13}'+bc_{32}'ab_{23}' \\ ABC_5 &= ac_{31}'ab_{31}'+ac_{32}'ab_{32}' \end{aligned} \tag{4.58}$$

(this notation is also used for all three-image sets, i.e. $IJK_3 = jk_{13}'ik_{13}'+jk_{23}'ik_{23}'$ etc…)

Note that for a rotation matrix:

$$ab_{13}^2 + ab_{23}^2 = ab_{31}^2 + ab_{32}^2 = 1 - ab_{33}^2 \tag{4.59}$$

Hence the right side term and middle term of equation (4.59) can be used interchangeably in any expression. Because of this fact it should follow that in the numerical calculations:

$$ab_{13}'^2+ab_{23}'^2 \approx ab_{31}'^2+ab_{32}'^2 \tag{4.60}$$

So, the right side and left side of equation (4.60) can also be used interchangeably. Thus an additional term which is the average of the two terms in (4.60) is defined:

$$AB_1 = \frac{ab_{13}'^2+ab_{23}'^2+ab_{31}'^2+ab_{32}'^2}{2} \approx ab_{13}'^2+ab_{23}'^2 \approx ab_{31}'^2+ab_{32}'^2 \tag{4.61}$$

In the calculations that follow the term (AB_1) is used in place of the terms $ab_{13}'^2+ab_{23}'^2$ and $ab_{31}'^2+ab_{32}'^2$. As with other notations, this notation is used hereafter for all three-image sets:

$$\begin{aligned} AC_1 &= \frac{ac_{13}'^2+ac_{23}'^2+ac_{31}'^2+ac_{32}'^2}{2} \approx ac_{13}'^2+ac_{23}'^2 \approx ac_{31}'^2+ac_{32}'^2 \\ BC_1 &= \frac{bc_{13}'^2+bc_{23}'^2+bc_{31}'^2+bc_{32}'^2}{2} \approx bc_{13}'^2+bc_{23}'^2 \approx bc_{31}'^2+bc_{32}'^2 \end{aligned} \tag{4.62}$$

By substituting terms from equations (4.58, 4.61 and 4.62) into equations (4.56 and 4.57):

$$\begin{aligned} ABC_3 &= (ABC_2BC_1)ab_{33} - (ABC_1ABC_4)bc_{33} \\ ABC_5 &= -(ABC_2ABC_4)ab_{33} + (ABC_1AB_1)bc_{33} \end{aligned} \tag{4.63}$$

Thus by considering the constraint equation (4.42) $[ac]=[bc][ab]$ two linear equations (4.63) in two unknowns (ab_{33}, bc_{33}) have been developed.

Next consider the reordering of the constraint equation (4.42) as $[bc]=[ac][ab]^T$ and $[ab]=[bc]^T[ac]$. By considering these two reorderings, four additional linear equations are obtained in the same manner that (4.63) was obtained:

$$\begin{aligned}
ABC_3 &= \left(\frac{1}{ABC_2}AC_1\right)ab_{33} - \left(\frac{ABC_1}{ABC_2}ABC_5\right)ac_{33} \\
ABC_4 &= -\left(\frac{1}{ABC_2}ABC_5\right)ab_{33} + \left(\frac{ABC_1}{ABC_2}AB_1\right)ac_{33} \\
ABC_4 &= \left(\frac{ABC_2}{ABC_1}BC_1\right)ac_{33} - \left(\frac{1}{ABC_1}ABC_3\right)bc_{33} \\
ABC_5 &= -\left(\frac{ABC_2}{ABC_1}ABC_3\right)ac_{33} + \left(\frac{1}{ABC_1}AC_1\right)bc_{33}
\end{aligned} \tag{4.64}$$

By putting these six equations (4.63) and (4.64) together there are six equations in three unknowns ab_{33}, ac_{33}, and bc_{33}. Such is the case with all three-image sets.

Now, consider all three-image sets from the set of *I* images; there are $\binom{I}{3}$ three-image sets. Each of these sets produces 6 equations (like equation 4.63 and 4.64). So, by considering the set of all equations, there are 6*$\binom{I}{3}$ linear equations in $\binom{I}{2}$ unknowns $(ab_{33}, ac_{33}, ad_{33}, ..., ai_{33}, bc_{33}, bd_{33}, ...bi_{33}, cd_{33}, ...)$. These unknowns will be referred to as the 33-terms of the matrices (their row and column index is 33). These can be set up and solved for using a least squares approach. This is set up as shown in equation (4.65).

Once again, the system (4.65) is of the form $L = AX$ the solution to this system is $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. The residual R for this calculation can be computed by $R = AX - L$.

$$\begin{bmatrix} ABC_3 \\ ABC_5 \\ ABC_3 \\ ABC_4 \\ ABC_4 \\ ABC_5 \\ ABD_3 \\ ABD_5 \\ ABD_3 \\ ABD_4 \\ ABD_4 \\ ABD_5 \\ \dots \end{bmatrix} = \begin{bmatrix} ABC_2BC_1 & 0 & 0 & \dots & -ABC_1ABC_4 & 0 & \dots \\ -ABC_2ABC_4 & 0 & 0 & \dots & ABC_1AB_1 & 0 & \dots \\ \frac{AC_1}{ABC_2} & \frac{-ABC_1ABC_5}{ABC_2} & 0 & \dots & 0 & 0 & \dots \\ \frac{-ABC_5}{ABC_2} & \frac{ABC_1AB_1}{ABC_2} & 0 & \dots & 0 & 0 & \dots \\ 0 & \frac{ABC_2BC_1}{ABC_1} & 0 & \dots & \frac{-ABC_3}{ABC_1} & 0 & \dots \\ 0 & \frac{-ABC_2ABC_3}{ABC_1} & 0 & \dots & \frac{AC_1}{ABC_1} & 0 & \dots \\ ABD_2BD_1 & 0 & 0 & \dots & 0 & -ABD_1ABD_4 & \dots \\ -ABD_2ABD_4 & 0 & 0 & \dots & 0 & ABD_1AB_1 & \dots \\ \frac{AD_1}{ABD_2} & 0 & \frac{-ABD_1ABD_5}{ABD_2} & \dots & 0 & 0 & \dots \\ \frac{-ABD_5}{ABD_2} & 0 & \frac{ABD_1AB_1}{ABD_2} & \dots & 0 & 0 & \dots \\ 0 & 0 & \frac{ABD_2BD_1}{ABD_1} & \dots & 0 & \frac{-ABD_3}{ABD_1} & \dots \\ 0 & 0 & \frac{-ABD_2ABD_3}{ABD_1} & \dots & 0 & \frac{AD_1}{ABD_1} & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix} \begin{bmatrix} ab_{33} \\ ac_{33} \\ ad_{33} \\ \dots \\ bc_{33} \\ bd_{33} \\ \dots \end{bmatrix} \tag{4.65}$$

Note that equation (4.65) has $\binom{I}{2}$ unknowns because the relative rotations between each of the images taken two-at-a-time were considered as unknown. In deriving this equation (4.65), only the constraint equation (4.42) for images taken three-at-a-time was considered. However, the constraint from considering all I images simultaneously has not been fully enforced:

$$[ai] = [hi] * [gh] * \dots * [cd] * [bc] * [ab] \tag{4.66}$$

The constraint equation (4.66) is due to the fact that the rotation from image A to image I is just the composite of all of the rotations between images from image A to

image *I*. In other words, there are really only *I* unknown rotations, not $\binom{I}{2}$ unknown rotations. There are only *I* rotations that need to be solved for and the remaining rotations can be found from (4.66) and (4.42). Hence, equation (4.65) has more degrees of freedom than necessary because all of the constraints have not been fully enforced. Because this algorithm will be used to produce initial approximations and not a rigorous final solution, the additional degrees of freedom can be overlooked in this calculation.

Now that the 33-term for each matrix has been obtained, the remaining terms can be solved for. Before proceeding however, note that because the least-squares solution (4.65) is not a fully constrained system it may result in rotation matrix terms which have an absolute value greater than 1. If this occurs, the value of the terms should be replaced with a value near to, but less than, 1. A value slightly less than one (i.e. 0.99) is chosen to avoid numerical problems with ambiguous rotation angles at singularities.

So, recall that the 13, 23, 31, and 32 terms (matrix terms with row and column indices 13, 23, 31, and 32) are known to within the matrix scale factor (equation 4.43). Now, by using the properties of a rotation matrix and equation (4.43):

$$1 - ab_{33}^2 = ab_{13}^2 + ab_{23}^2 = \alpha\beta^2\left(ab_{13}'^2 + ab_{23}'^2\right) = \alpha\beta^2 * AB_1 \tag{4.67}$$

Rearranging this equation (4.67) yields:

$$\alpha\beta = \pm\sqrt{\frac{1 - ab_{33}^2}{AB_1}} \tag{4.68}$$

This equation (4.68) gives the absolute value of the matrix scale factors, but it does not provide the sign. If one matrix scale factor is arbitrarily assigned to be positive (or negative) the sign of the remaining matrix scale factors can be solved for by using

equations (4.52). Two sets of matrix scale factors will be defined—the first will be denoted as the (+) set and will be based on the positive root of $\alpha\beta$, the other set will be denoted as the (-) set and will be based on the negative root of $\alpha\beta$. Note that these two sets have equal absolute values and opposite signs. The existence of two solutions is due to the nature of parallel projection. This is discussed in detail in (Huang and Lee, 1989; Kyle, 2004). One of the solutions is the true solution, while the other is a mirror-image. A keypoint will be used later to resolve this ambiguity. For now, both the (+) set and the (-) set will be kept.

Now that the matrix scale factors have been obtained, the 13, 23, 31, and 32 terms of the rotation matrices can be solved for using equation (4.43). There will be two sets of 13, 23, 31, and 32 terms, one corresponding to the (+) set and one corresponding to the (-) set. Again, the (+) set and (-) set will have equal absolute values and opposite signs.

It is important to note that the 13, 23, 31, 32, and 33 matrix terms were solved for using least-squares solutions that were not fully constrained. Hence, the resulting terms are not exact rotation matrix terms and may not form true orthonormal matrices.

Now that the 33-terms and the 13, 23, 31, and 32 terms of the rotation matrices have been found, the remaining terms can be calculated by using the properties of rotation matrices outlined in section A.1. By using the cross-product property of rotation matrices, one equation for each of the 13, 23, 31, and 32 terms can be written:

$$\begin{aligned} ab_{13} &= ab_{21}ab_{32} - ab_{22}ab_{31} \\ ab_{23} &= ab_{12}ab_{31} - ab_{11}ab_{32} \\ ab_{31} &= ab_{12}ab_{23} - ab_{13}ab_{22} \\ ab_{32} &= ab_{13}ab_{21} - ab_{11}ab_{23} \end{aligned} \tag{4.69}$$

Also, from the dot-product property of rotation matrices:

$$\begin{aligned} ab_{11}ab_{31} + ab_{12}ab_{32} + ab_{13}ab_{33} &= 0 \\ ab_{21}ab_{31} + ab_{22}ab_{32} + ab_{23}ab_{33} &= 0 \\ ab_{11}ab_{13} + ab_{21}ab_{23} + ab_{31}ab_{33} &= 0 \\ ab_{12}ab_{13} + ab_{22}ab_{23} + ab_{32}ab_{33} &= 0 \end{aligned} \tag{4.70}$$

These can be put in matrix-form to solve for the 11, 12, 21, 22 terms (matrix terms with row and column indices 11, 12, 21, and 22) as follows:

$$\begin{bmatrix} ab_{13} \\ -ab_{23} \\ ab_{31} \\ -ab_{32} \\ -ab_{13}ab_{33} \\ -ab_{23}ab_{33} \\ -ab_{31}ab_{33} \\ -ab_{32}ab_{33} \end{bmatrix} = \begin{bmatrix} 0 & 0 & ab_{32} & -ab_{31} \\ ab_{32} & -ab_{31} & 0 & 0 \\ 0 & ab_{23} & 0 & -ab_{13} \\ ab_{23} & 0 & -ab_{13} & 0 \\ ab_{31} & ab_{32} & 0 & 0 \\ 0 & 0 & ab_{31} & ab_{32} \\ ab_{13} & 0 & ab_{23} & 0 \\ 0 & ab_{13} & 0 & ab_{23} \end{bmatrix} \begin{bmatrix} ab_{11} \\ ab_{12} \\ ab_{21} \\ ab_{22} \end{bmatrix} \tag{4.71}$$

This represents an eight-equation four-unknown linear system $L = AX$ which can be solved by least squares $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. This gives the remaining rotation matrix entries $ab_{11}, ab_{12}, ab_{21}$, and ab_{22}. Note that the 11, 12, 21, and 22 terms for the (+) set are essentially the same as those for the (-) set, so the computation (4.71) does not have to be carried out for both sets. If the calculations above resulted in exact rotation matrices then the 11, 12, 21, 22 terms from the (+) set and (-) set would be exactly equal. But, because the 13, 23, 31, 32, 33 terms came from least-squares solutions that were not fully constrained they are not exact rotation matrix terms and there will be a minimal amount of difference in the 11, 12, 21, 22 terms from the (+) set and those from the (-) set.

4.1.5 Approximating the absolute rotations

Now that the shift and scale of each image with respect to 3D space has been determined along with the rotation for each image relative to the other images, only the three rotation angles $(\omega, \varphi, \kappa)$ of each image with respect to 3D space remain to be solved for. Recall that each image is related to 3D space by the following (like equation 4.10):

$$\begin{bmatrix} x_A - \Delta x_A \\ y_A - \Delta y_A \\ * \end{bmatrix} = s_A \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$
$$\begin{bmatrix} x_B - \Delta x_B \\ y_B - \Delta y_B \\ * \end{bmatrix} = s_B \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (4.72)$$
...
$$\begin{bmatrix} x_I - \Delta x_I \\ y_I - \Delta y_I \\ * \end{bmatrix} = s_I \begin{bmatrix} i_{11} & i_{12} & i_{13} \\ i_{21} & i_{22} & i_{23} \\ i_{31} & i_{32} & i_{33} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

This is an absolute orientation to 3D space. However, the rotation of 3D space has not yet been defined. So, to fully define 3D space and find the rotation angles, 3D space will be "fixed" to one of the images. Then the orientation of each image relative to the "fixed" image will give the absolute orientation of that image. Without loss of generality image *A* will be selected to have no rotation with respect to 3D space:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (4.73)$$

Which then gives:

$$\begin{bmatrix} x_A - \Delta x_A \\ y_A - \Delta y_A \\ * \end{bmatrix} = s_A \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad (4.74)$$

Next, (from equation 4.44):

$$\begin{bmatrix} ab_{11} & ab_{12} & ab_{13} \\ ab_{21} & ab_{22} & ab_{23} \\ ab_{31} & ab_{32} & ab_{33} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} a_{11} & a_{21} & a_{31} \\ a_{12} & a_{22} & a_{32} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} = \ldots$$
$$\ldots = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} = \begin{bmatrix} ab_{11} & ab_{12} & ab_{13} \\ ab_{21} & ab_{22} & ab_{23} \\ ab_{31} & ab_{32} & ab_{33} \end{bmatrix} \tag{4.75}$$

And likewise for the other images:

$$\begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix} = \begin{bmatrix} ac_{11} & ac_{12} & ac_{13} \\ ac_{21} & ac_{22} & ac_{23} \\ ac_{31} & ac_{32} & ac_{33} \end{bmatrix}$$
$$\ldots$$
$$\begin{bmatrix} i_{11} & i_{12} & i_{13} \\ i_{21} & i_{22} & i_{23} \\ i_{31} & i_{32} & i_{33} \end{bmatrix} = \begin{bmatrix} ai_{11} & ai_{12} & ai_{13} \\ ai_{21} & ai_{22} & ai_{23} \\ ai_{31} & ai_{32} & ai_{33} \end{bmatrix} \tag{4.76}$$

Thus by fixing the orientation of 3D space to image *A* the relative orientation has been converted into an absolute orientation and the values that were calculated $([ab],[ac],\ldots,[ai])$ give the approximate rotation matrices.

Next, the three rotation angles can be derived from the approximate matrix with equations (A.7 – A.12). Once the three rotation angles are obtained the calculation of the parallel projection parameters is finished. It should be noted that there are two sets of parallel projection parameters, the (+) set and the (-) set. Note also that the two sets are equal, except that ω and φ have opposite signs. This concludes the initial approximation algorithm for parallel projection parameters.

4.2 Adding an Additional Image

One additional algorithm that will be considered is an algorithm for adding an additional image to a previous solution by parallel projection. This algorithm will solve for initial approximations of the parallel projection parameters for the additional image from a set of 3D coordinates and image coordinates. Recall from Chapter 2 that the DLT can be used to add another image to a solution, but it loses stability as the angular field of view becomes narrower. The method developed here is based on parallel projection in order to add images with a narrow field of view. It is similar to the algorithm outlined by Kyle (2004). Kyle (2004, p. 48) reported,

> It is possible to apply the simple mathematics of a parallel projection to conventionally imaged objects of known dimensions, in order to calculate approximate values for object position and angular orientation relative to the camera or vice versa. The conditions are that the imaging bundle for the targets used in the calculation should be relatively narrow … it is further assumed that camera internal geometry is well known …

Note however that in the development of the algorithm here, the dimensions of the target object do not need to be known precisely, only approximations. Because approximations of the actual target geometry are used, more point correspondences than is necessary are collected resulting in an over-determined system with a least-squares solution.

For this algorithm it is assumed that there *n* points $(X_1, Y_1, Z_1),...,(X_n, Y_n, Z_n)$ for which the coordinates are known approximately. It is assumed that there is an image *J* which it is desired to orient relative to these points. Assume that a set of image coordinates of a subset of the set of the *n* points have been collected on image *J*:

$$(x_{J,1}, y_{J,1}),...,(x_{J,p}, y_{J,p}) \tag{4.77}$$

It is not necessary for all n points to be visible on image J, but there must be at least four points on image J (i.e. $p \geq 4$).

Now, writing the affine projection equations (3.1) in matrix form:

$$\begin{bmatrix} x_{J,1} \\ x_{J,2} \\ \dots \\ x_{J,p} \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \dots & \dots & \dots & \dots \\ X_p & Y_p & Z_p & 1 \end{bmatrix} \begin{bmatrix} A_{J,1} \\ A_{J,2} \\ A_{J,3} \\ A_{J,4} \end{bmatrix} \qquad \begin{bmatrix} y_{J,1} \\ y_{J,2} \\ \dots \\ y_{J,p} \end{bmatrix} = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \\ X_2 & Y_2 & Z_2 & 1 \\ \dots & \dots & \dots & \dots \\ X_p & Y_p & Z_p & 1 \end{bmatrix} \begin{bmatrix} A_{J,5} \\ A_{J,6} \\ A_{J,7} \\ A_{J,8} \end{bmatrix} \tag{4.78}$$

Equations (4.78) are of the form $L = AX$ and have the least squares solutions $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. By solving these two equations initial approximations for the affine projection parameters for image J $\left(A_{J,1},...,A_{J,8}\right)$ are obtained.

Now, consider equations (3.4 and A.6), which express the general affine projection parameters in terms of the six parallel projection parameters. By the properties of a rotation matrix (discussed in the appendix):

$$m_{11}^2 + m_{12}^2 + m_{13}^2 = 1 \tag{4.79}$$

And hence the image scale s can be solved from the affine projection parameters by:

$$s_J = s_J \sqrt{m_{11}^2 + m_{12}^2 + m_{13}^2} = \sqrt{\left(sm_{11}\right)^2 + \left(sm_{12}\right)^2 + \left(sm_{13}\right)^2} = \sqrt{A_{J,1}^2 + A_{J,2}^2 + A_{J,3}^2} \tag{4.80}$$

Similarly, $s_J = \sqrt{A_{J,5}^2 + A_{J,6}^2 + A_{J,7}^2}$. Because least-squares numerical calculations were used to get the eight affine projection parameters $\left(A_{J,1}, A_{J,2},..., A_{J,8}\right)$, the two expressions for s_J will not be equal. Thus, in order to utilize all of the data, s_J can be calculated as the average of the two terms:

$$s = \frac{\sqrt{A_{J,1}^2 + A_{J,2}^2 + A_{J,3}^2} + \sqrt{A_{J,5}^2 + A_{J,6}^2 + A_{J,7}^2}}{2} \tag{4.81}$$

Then the first six matrix terms are obtained by:

$$\begin{aligned} m_{11} &= \frac{A_{J,1}}{s} \qquad m_{12} = \frac{A_{J,2}}{s} \qquad m_{13} = \frac{A_{J,3}}{s} \\ m_{21} &= \frac{A_{J,5}}{s} \qquad m_{22} = \frac{A_{J,6}}{s} \qquad m_{23} = \frac{A_{J,7}}{s} \end{aligned} \tag{4.82}$$

Then by the properties of rotation matrices the remaining matrix terms are found by:

$$\begin{aligned} m_{31} &= m_{12}m_{23} - m_{13}m_{22} \\ m_{32} &= m_{13}m_{21} - m_{11}m_{23} \\ m_{33} &= m_{11}m_{22} - m_{12}m_{21} \end{aligned} \tag{4.83}$$

Note that the resulting rotation matrix will not be truly orthogonal because each of the terms were obtained using least-squares calculations and all of the constraints were not fully enforced. In the algorithm presented by Kyle (2004), cross-products are used to construct an orthogonal rotation matrix from the approximate rotation matrix. However, because it is desired to solve for the three rotation angles from the matrix, these can be found using equations (A.7 – A.12). Then, an orthogonal matrix can be constructed from these angles using equations (A.6).

CHAPTER 5

OUTLINE OF COMPREHENSIVE ALGORITHM

To this point, several equations and models have been outlined. This chapter explains how all of this can be streamlined into one comprehensive algorithm for the 3D reconstruction of a target from surveillance imagery. The equations for this algorithm have been developed above (Chapters 3-4), or are given in the appendix, so in the outline that follows the equations will not be repeated, just referenced.

5.1 Introduction

The comprehensive algorithm utilizes three models: parallel projection, perspective-corrected parallel projection, and perspective projection. By using all three projections, accuracy and stability across a wide range of angular fields of view is possible. Also, because all three projections are used, there are essentially three sets of results that are produced. Each of these results can be compared and the most accurate result can be used as the final output. The comprehensive algorithm can be broken down into eight steps:

Step 1: Collect Inputs

Step 2: Parallel Projection Initial Approximation

Step 3: Initial Parallel Projection 3D Reconstruction

Step 4: Parallel Projection Bundle Triangulation

Step 5: Perspective-Corrected Parallel Projection 3D Reconstruction

Step 6: Perspective-Corrected Parallel Projection Bundle Triangulation

Step 7: Perspective Projection 3D Reconstruction

Step 8: Perspective Projection Bundle Triangulation

From the above, it can be seen that parallel projection is used to obtain initial approximations. These approximations are refined through bundle triangulation resulting in a final output from parallel projection. Then perspective-correction is performed on this result and it serves as an initial approximation for the perspective-corrected parallel projection. A bundle triangulation is performed to refine these approximations and a final result from perspective-corrected parallel projection is output. Finally the perspective-corrected parallel projection results are used as an initial approximation for the standard perspective projection. Bundle triangulation refines these results and an output is obtained from perspective projection.

Each of the above steps is outlined in detail in this chapter, including references to the specific equations that are used. A flow chart is also given in Figure 5.1 on the next page. It may be helpful to follow the flow chart when considering each step.

5.2 Step 1: Collect Inputs

The first step is to collect the inputs for the algorithm. The first input is a series of point correspondences on the images. This is generally done manually. It is assumed that there are n points with unknown 3D coordinates. It is assumed further that there are I images (denoted by image A, image B,…, image I) with unknown EOP. It is necessary that $I \geq 3$. (This is due to the fact that the parallel projection algorithm requires at least three images to solve for relative rotation of images.) It is also assumed that the n points appear in the I images. It not necessary that all n points are visible in all I images—some

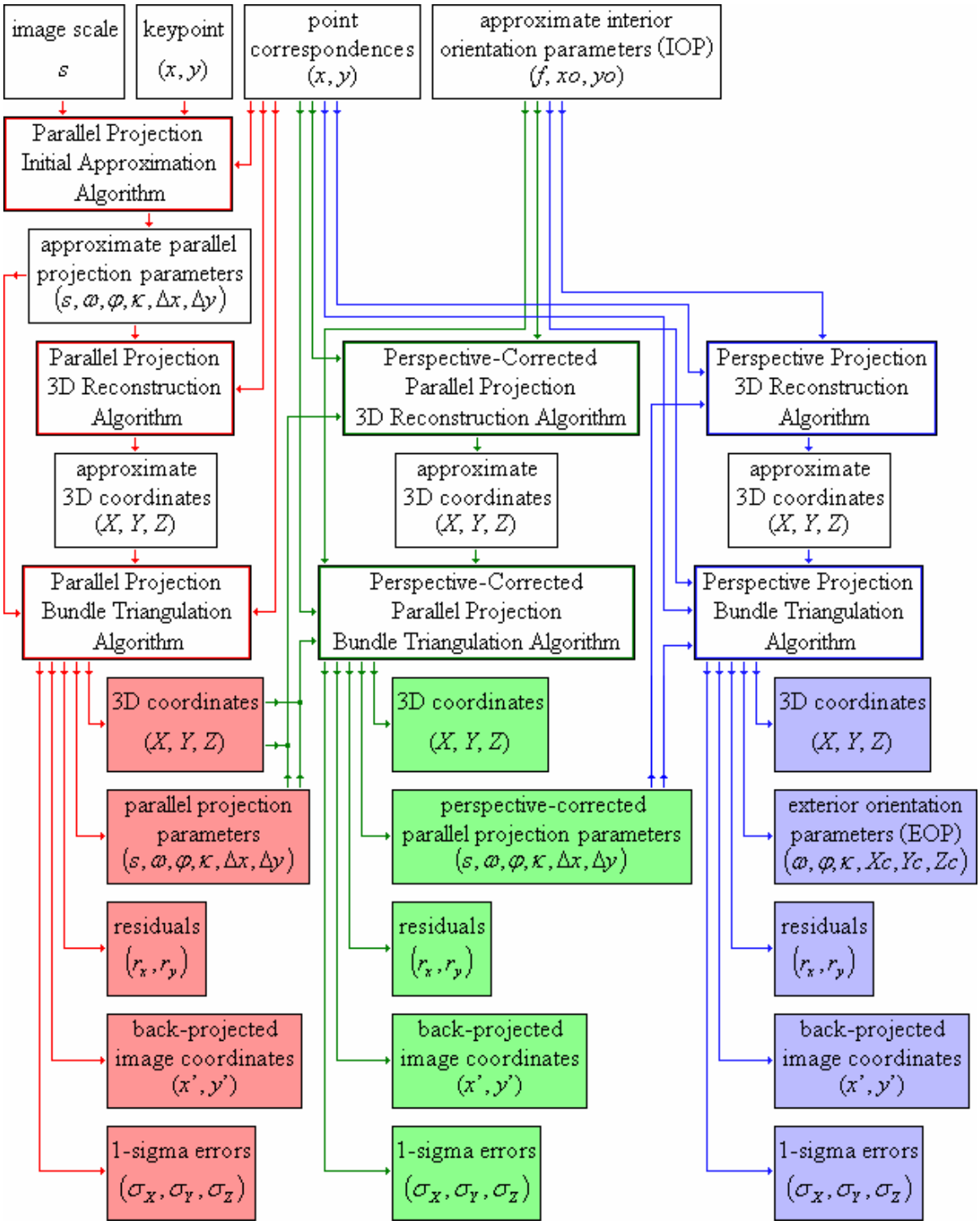

Figure 5.1. Flow chart of comprehensive algorithm.

of the coordinates may be missing. It is necessary however that each point is visible in at least two images. It is also necessary to find a subset of the *I* images which has at least three images and fulfills the following two requirements:

(1) there is at least one point that appears on all of the images in the subset

(2) for any pair of images in the subset there must be at least four common points on the two images

This subset of images will be used to find the initial approximations. If it is possible to fulfill these two requirements with all of the images, then all of the images can be processed with the parallel projection initial approximation algorithm. If these two requirements are not met by the full image set, then a smaller subset of images must be chosen to be used in the initial approximation step, and the remaining images can be added later using the algorithm to add another image (section 4.2).

The second input for the algorithm is the approximate IOP(f, xo, yo). The focal length of a particular lens is generally a good approximation of the principal distance (*f*). The center of an image is generally a good approximation of the principal point (*xo*, *yo*).

There are cases where images may have been captured with an unknown sensor. In these cases it will generally not be possible to approximate the IOP. If such images are used, the parallel projection steps of the procedure can still be used without the initial approximations, but the perspective projection steps may have to be ignored for these images. Or, a standard method such as the DLT may supply the approximate IOP, depending on the angular field of view.

The third input is the scale for one of the images. This scale defines the scale of the final 3D coordinates. This scale is approximately equal to the principal distance of

the image divided by the average distance from the image to the target (see equation 3.36). If the scale of the final model is not important or if the approximate distance to the target is unknown, this input can be set to an arbitrary value such as 1. In this case, the units of the final model are arbitrary units.

The final input is the keypoint. As has been mentioned earlier, the parallel projection algorithm produces two solutions. These two solutions are mirror images of each other. In order to determine which solution is the correct solution, and which is the mirror image, one additional input is defined. A point on the target which is visible in the first image (image *A*) and which is closer to the camera in the *z*-direction (the focal axis of the camera) than the centroid of the target is selected as the keypoint. Because of the way that 3D space is defined[1] the keypoint should have a positive *Z*-coordinate. This will allow the correct solution from the parallel projection algorithm to be determined.

5.3 Step 2: Parallel Projection Initial Approximation

The parallel projection initial approximation algorithm is the first major step in the comprehensive algorithm. It is used to get initial approximations of the EOP by parallel projection.

5.3.1 Inputs, assumptions and outputs

There are three inputs for the algorithm discussed above: image coordinates of point correspondences, the scale of one image, and a keypoint.

[1] The origin is the approximately the centroid of the target object, and the positive *Z*-direction points toward the camera.

There is one output of this algorithm: a set of initial parallel projection parameters. This set of approximate camera orientations includes the three rotation angles $(\omega, \varphi, \kappa)$, the image scale *s*, and the shift terms $(\Delta x, \Delta y)$ for each image.

5.3.2 Step-by-step outline

There are five major steps of the algorithm:

(1) determine the shift of each image

(2) select the points to be used to calculate the image scales and rotation matrices

(3) calculate the image scales

(4) calculate the rotation angles

(5) 3D reconstruction of the keypoint.

Each of these steps will be considered individually.

The first step (1) is to find the approximate shift of each image. This portion of the algorithm is outlined in section 4.1.2 and is based on the equations (4.4 – 4.9). The origin of 3D space is defined as the centroid of the set of points which appear in all *I* images. Then, the centroid of those image coordinates in each image becomes the shift for that image. So, the steps for this portion of the algorithm are: a) find the set of points which appear in all images and get the image coordinates of these points for each image and b) take the centroid of those coordinates in each image (4.8).

The *x*-coordinate of the centroid in each image is Δx for that image, and the *y*-coordinate of the centroid in each image is Δy for that image (equation 4.5).

The next step (2) is to select which points will be used to calculate image scales and rotations. One failure mode of the parallel projection algorithm is when a set of

coplanar points is used as point correspondences when determining the relative rotations of images (Huang and Lee, 1989; Kyle, 2004). It has been found by experiment that if a large number of points which are nearly coplanar are used to calculate the image scales and rotations the algorithm is subject to failure due to matrices which are nearly singular. Hence, a smaller sub-set of points which are not coplanar should be used to determine the image scales and rotations. Recall, however, that it is necessary that when any two images are considered, there must be at least four points which appear on both images.

This portion of the algorithm is up to the discretion of the user, but from experiment it has been determined that favorable results are obtained when determining the rotations and image scales from a set of points which satisfies four conditions: 1) the points appear in all of the images, 2) the points are not nearly coplanar, 3) the points are spread over a large portion of the target, and 4) there are at least four such points.

If point-correspondences are chosen such that the above four conditions are satisfied, then this step could be executed as: a) find the set of points which are on all images, b) save this smaller set of points for the calculation of rotations and image scales, and c) check to make sure that there are a sufficient number of points (i.e. ensure that there are at least four points which appear in all images).

In the event that no such set of points exists, subsequent steps can be carried out with the full set of points, but as mentioned above, the parallel projection algorithm is subject to failure when the set of points are nearly coplanar. For subsequent steps, it will be assumed that this step has been performed as described above.

The next step (3) is to solve for the scales of the images. This portion of the algorithm is based on section 4.1.3 and the equations (4.10 – 4.31) outlined above. This step proceeds as follows:

a) Consider each set of two images and compute the four *R* terms for each set using equation (4.21). Note that the points from the smaller set which appear in both images, and have been shifted are used for this calculation. There will be $\binom{I}{2}$ two-image sets, and hence there will be $\binom{I}{2}$ sets of *R* terms that have to be solved for.

b) Compute a fifth *R* term for each two image set using equation (4.24). Then perform the iterative least squares computation to solve for the image scales using equation (4.31). Recall that the scale of image *A*, s_A (which was input), is used as an initial approximation for each image scale. This computation may require several iterations as each iteration improves the initial guesses. The convergence criteria is up to the discretion of the user. Generally the iteration is stopped when the magnitude of the corrections *X* is below a desired threshold.

The next step (4) solves for the three rotation angles for each image. This step follows sections 4.1.4 and 4.1.5 and equations (4.32 – 4.71 and 4.72 – 4.76) outlined above. This step is carried out as follows:

a) Begin with the set of points which are on all images. Then consider each two-image set and compute the four terms for each set using equation (4.39). These are saved for each two-image set (recall that these equations can be set up and solved in the same

manner as 4.21.) Note that the points used for this calculation have been shifted and scaled.

b) Consider each three-image set and compute the eight coefficients defined in equations (4.53, 4.58, 4.61, and 4.62). So, there will be $\binom{I}{3}$ sets of eight coefficients. These eight coefficients are based on the four terms computed in step a) from the two-image sets. For example, for the three-image set (image *A*, image *B*, image *C*) there are three sets of two-image terms (image *A* and image *B*), (image *A* and image *C*), and (image *B* and image *C*), which are used to solve for the eight coefficients. By using the eight coefficients from each three-image set, solve for the 33-terms of the rotation matrices using a least squares computation. This is outlined in equation (4.65).

c) At this point, an error check on the 33-terms is also performed. In the numerical calculations, it is possible to have a 33-term with an absolute value greater than 1. If this happens, change the absolute value of the term to 0.99 since no term in a rotation matrix can have an absolute value greater than 1.

d) Next, for each rotation matrix, calculate the matrix scale factor for the 13, 23, 31, 32 matrix terms (equation 4.43) using equation (4.68). Note that for each term, there is a (+/-) sign. By default, choose the positive root of $\alpha\beta$ and then solve for the signs of the remaining terms using equation (4.52) for each three-image set. This will be denoted as the (+) set; the (-) set consists of the same terms with opposite signs. It will be determined later which set is the correct set.

e) Next, solve for the remaining terms of each rotation matrix. The (13, 23, 31, 32) terms are found by multiplying the (13', 23', 31', 32') coefficients by the matrix scale

factors. The (13, 23, 31, 32) terms for the (+) set and (-) set will have equal absolute values, but opposite signs. At this point error-correct the 31-terms—if the absolute value of these are greater that 1 then they are replaced with an absolute value of 0.99. Then, the (11, 12, 21, 22) terms of each matrix are solved for using equation (4.71). Note that this can be done for either the (+) set or the (-) set—these terms are equal for each set.

f) Finally, solve for the three rotation angles for each matrix using equations (A.7 – A.12). By default, use the (+) set. Then recalculate each matrix from the three rotation angles using equations (A.6). This is done because the rotation matrices that result from steps a) – e) above are not necessarily orthogonal rotation matrices—they were found by least-squares and were not fully constrained. Solve for rotation matrices directly from three rotation angles so that the matrices are exact.

The final step (5) is the 3D reconstruction of the keypoint. This step allows for the determination of which set of matrix terms, the (+) set or the (-) set, is the correct solution. Set up an equation to solve for the 3D coordinates of the keypoint like equation (3.6). Once the 3D coordinates of the keypoint have been solved for, check the sign of the *Z*-coordinate. If the *Z*-coordinate is positive, then the correct set (the (+) set) was chosen and the rest of the algorithm can be carried out. If the *Z*-coordinate is negative, the (-) set must be used. This is equivalent to changing the signs on the 13, 23, 31, 32 terms of the rotation matrices, and changing the signs of ω and φ.

5.4 Step 3: Initial Parallel Projection 3D Reconstruction

The parallel projection 3D reconstruction algorithm solves for the 3D coordinates of target-points. There are two inputs for this algorithm: the set of image coordinates

collected in step 1 and the set of initial parallel projection parameters solved for in step 2. There is one major output of this algorithm: the 3D coordinates of target points.

The step-by-step procedure for this step is as follows: 3D reconstruction of each point is performed by setting up an equation like (3.6). If it is desired, the vector of residuals, the total residuals, the covariance matrix, the 1-sigma errors, and the back-projected image coordinates can also be calculated (3.7 – 3.15).

5.5 Step 4: Parallel Projection Bundle Triangulation

The parallel projection bundle triangulation algorithm is discussed in section 3.2 and the outline here follows the discussion there very closely. The bundle triangulation algorithm solves for the 3D coordinates of the target points and the EOP simultaneously. It also solves for residuals, 1-sigma errors, and back-projected image coordinates. Note that the outline given here one particular implementation of the bundle triangulation algorithm and some constraints have been assumed that are not necessary in all implementations of the bundle triangulation algorithm.

There are three inputs into the parallel projection bundle triangulation algorithm: 1) the set of image coordinates collected in step 1, 2) the set of initial approximations for the 3D coordinates of the target points calculated in step 3, 3) the set of parallel projection parameters calculated in step 2. All of these are input into the bundle triangulation algorithm.

There are five outputs from the parallel projection bundle triangulation algorithm: 1) a set of 3D coordinates, 2) residuals, 3) 1-sigma errors, 4) back-projected image coordinates, and 5) the final parallel projection parameters.

The step-by-step procedure for this step is as follows: First, the partial derivatives (3.21) are evaluated at the initial approximations. These values are then put into matrix and vector form (3.24 – 3.31). These are solved to get the *X* vector which is added to the initial approximations to get improved approximations. For each iteration the residuals, 1-sigma errors, and back-projected image coordinates are also calculated. This process is repeated until the corrections *X* become negligible.

5.6 Step 5: Perspective-Corrected Parallel Projection 3D Reconstruction

The perspective-corrected parallel projection 3D reconstruction algorithm solves for the 3D coordinates of target-points.

There are four inputs for this algorithm: 1) the set of image coordinates collected in step 1, 2) the approximate IOP collected in step 1, 3) the set of parallel projection parameters calculated in step 4, and 4) the set of 3D coordinates calculated in step 4.

The output of this algorithm is a set of 3D coordinates. The residuals, 1-sigma errors, and back-projected image coordinates can also be calculated if desired.

The 3D reconstruction algorithm consists of two steps: 1) perspective-correction, 2) perform 3D reconstruction.

The first step is to perform perspective-correction. The image coordinates are perspective-corrected using equation (3.45). Note that (*X*,*Y*,*Z*), *s*, $(\Delta x, \Delta y)$, and the *m*-terms are from the parallel projection parameters, and *f*, *xo*, *yo* are the approximate IOP. Also, (*Xc*, *Yc*, *Zc*) can be calculated with equation (3.42).

The second step is to perform 3D reconstruction of the perspective-corrected coordinates using equation (3.50). Note that the parallel projection parameters are used

for this calculation with one exception: the two shift terms $(\Delta x, \Delta y)$ for each image should be perspective-corrected using equation (3.45), with (X, Y, Z) = (0, 0, 0).

If it is desired, the back-projected image coordinates, residuals, and 1-sigma errors can also be calculated using equations (3.51 – 3.53 and 3.7 – 3.15).

5.7 Step 6: Perspective-Corrected Parallel Projection Bundle Triangulation

The perspective-corrected parallel projection bundle triangulation algorithm is very similar to the parallel projection bundle triangulation algorithm discussed in section 3.2.

There are five inputs into the perspective-corrected parallel projection bundle triangulation algorithm: 1) the set of image coordinates collected in step 1, 2) the approximate IOP collected in step 1, 3) the approximate 3D coordinates calculated in step 5, 4) the parallel projection parameters from step 4, and 5) the 3D coordinates from parallel projection from step 4.

There are five outputs of the perspective-corrected parallel projection bundle triangulation algorithm: 1) 3D coordinates, 2) the perspective-corrected parallel projection parameters, 3) residuals, 4) 1-sigma errors, and 5) back-projected image coordinates.

The first step in the algorithm is to perform perspective-correction on the image coordinates. This is performed using the 3D coordinates of the target points and the parallel projection parameters using (3.50) in the same manner that was done in the perspective-corrected parallel projection 3D reconstruction algorithm.

The remainder of the algorithm follows the same procedure as the parallel projection bundle triangulation algorithm with a few small changes. The parallel projection parameters are used as initial approximations, with the shift terms perspective-corrected in the same manner as for the perspective-corrected parallel projection 3D reconstruction algorithm. Also, when calculating residuals, 1-sigma errors, and back-projected image coordinates, the inverse-perspective correction must be performed (equations 3.51 – 3.53). Note that perspective-correction and inverse perspective-correction are always performed using the parallel projection 3D coordinates and the parallel projection parameters.

5.8 Step 7: Perspective Projection 3D Reconstruction

The perspective projection 3D reconstruction algorithm solves for the 3D coordinates of target-points.

There are three inputs for this algorithm: 1) the set of image coordinates collected in step 1, 2) the approximate IOP collected in step 1, and 3) the perspective-corrected parallel projection parameters from step 6. Note that the perspective center can be found from the perspective-corrected parallel projection parameters with equation 3.42.

There is one output of the algorithm: 3D coordinates of target points. If desired, the residuals, 1-sigma errors, and back-projected image coordinates can also be output.

In this step, 3D reconstruction is performed for each point with (A.43 and A.44). If desired, the residuals, 1-sigma errors, and back-projected image coordinates can be found with equations (A.29 – A.37).

5.9 Step 8: Perspective Projection Bundle Triangulation

The perspective projection bundle triangulation algorithm is discussed in the appendix and the outline here follows the discussion there very closely.

There are four inputs into the perspective projection bundle triangulation algorithm: 1) the set of image coordinates collected in step 1, 2) the approximate IOP collected in step 1, 3) the set of initial approximations for the 3D coordinates of the target points calculated in step 7, 4) the perspective-corrected parallel projection parameters found in step 6.

There are five outputs of the perspective projection bundle triangulation algorithm: 1) 3D coordinates, 2) residuals, 3) 1-sigma errors, 4) back-projected image coordinates, and 5) the perspective projection EOP.

The step-by-step outline for this step is as follows: First, the partial derivatives (A.17) are evaluated at the initial approximations. These values are then put into matrix and vector form (A.20 – A.28). The *X* vector is calculated and added to the initial approximations to get improved approximations. At each iteration the residuals, 1-sigma errors, and back-projected image coordinates are calculated with equations (A.29 – A.37). This is repeated until the corrections *X* become negligible.

CHAPTER 6

TESTS OF THE COMPREHENSIVE ALGORITHM USING ACTUAL SURVEILLANCE IMAGES

This chapter discusses three tests of the comprehensive algorithm using actual surveillance images. The same image sets that were used to test the standard algorithms were used.

6.1 Haddock Images at Close Range with the CANON 5D

The first set of images used to test the comprehensive algorithm was the series of eight images of the Coast Guard Vessel Haddock that was used to test the standard methods. Recall that the images were taken with a 12.7 megapixel CANON 5D digital camera with a 400 mm lens at distances ranging from approximately 300 m – 450 m. The eight images that were used are shown in Figure 2.3 in Chapter 2.

Recall that the test conducted on these images using standard techniques resulted in an accurate 3D model. This was due to the fact that these images were captured at a modest range to the target, so complications due to narrow angular field of view were not manifest.

Due to the modest angular field of view, for these tests it was expected that the parallel projection would not provide results as accurate as those for the perspective projection. In fact, it was expected that using parallel projection as a means of obtaining initial approximations may lead to difficulty. For this reason, this dataset represented a good test of the robustness of the parallel projection initial approximation algorithm. It

also served as a good test of the algorithm's workflow—using parallel projection as an initial step towards a solution for the perspective projection.

The first input for the algorithm was the set of point correspondences that were collected across the eight images. Recall from Chapter 2 that image coordinates for 65 points were manually collected.

The second input for the algorithm was the approximate IOP. The same approximate values as were used for the first test were used (f = 48770 pixel, xo = 2184 pixel, yo = 1456 pixel).

The third input for the algorithm was the approximate scale for the first image. This scale was chosen by trial-and-error using the same two points on the target that were used for the first test.

The final input was a keypoint. Recall that the keypoint is used to resolve the mirror image ambiguity resulting from the parallel projection. The keypoint is a point on the first image that is closer to the camera than the majority of the other points.

The test was carried out following the procedure outlined in Chapter 5. Initial approximations were obtained by using the parallel projection initial approximation algorithm. Then, 3D reconstruction and bundle triangulation for the parallel projection were performed. Next, the results from the parallel projection were used as initial approximations for the perspective-corrected parallel projection. From these initial approximations, 3D reconstruction and bundle triangulation were performed for perspective-corrected parallel projection. Finally, the results from perspective-corrected parallel projection were used as initial approximations for the perspective projection. For the perspective projection, 3D reconstruction and bundle triangulation were performed.

The results of the test will now be discussed. These outputs include three sets of 3D coordinates, 1-sigma errors, and residuals.

First, the 3D coordinates are considered. The results from the comprehensive algorithm were compared with the results from the tests using standard methods using a 3D rigid transformation. For each projection, the mean error between the sets was considered. The mean error between the parallel projection and standard method was 0.090 ~m (0.294 ft), the mean error between the perspective-corrected parallel projection and standard method was 0.003 ~m (0.011 ft), and the mean error between the perspective projection and standard method was 0.003 ~m (0.009 ft). This shows that for the perspective-corrected parallel projection and perspective projection, the algorithm produced results that were equivalent to the results from the standard methods up to the order of precision. Recall that the mean 1-sigma error for the results from the standard methods was 0.006 ~m (0.020 ft). The results for the parallel projection were decidedly less accurate than the results from perspective-corrected parallel projection and perspective projection, as expected.

The second output to consider is the sets of 1-sigma errors. The average 1-sigma error for the parallel projection was 0.062 ~m (0.203 ft) with a maximum of 0.202 ~m (0.663 ft). The average 1-sigma error for the perspective-corrected parallel projection was 0.006 ~m (0.020 ft) with a maximum of 0.016 ~m (0.052 ft). The average 1-sigma error for the perspective projection was 0.006 ~m (0.021 ft) with a maximum of 0.014 ~m (0.045 ft). Note that the values from the perspective-corrected parallel projection and perspective projection are nearly identical to the results in Chapter 2 obtained by using the standard methods (mean of 0.006 ~m and max of 0.015 ~m). Note also that the

values from the parallel projection were the least accurate of the three, as expected. These values are given in Table 6.1.

The third output to consider is the set of residuals. The average total residual (as calculated with equations 3.9 and A.31) for the parallel projection was 9.45 pixels with a maximum of 56.47 pixels. The average total residual for the perspective-correction was 0.88 pixels with a maximum of 3.40 pixels. The average total residual for the perspective projection was 0.93 pixels with a maximum of 3.34 pixels. These total residuals give a measure of the overall precision of each point in each image. Note that once again, the results from the perspective-corrected parallel projection and perspective projection are nearly identical to the results using standard methods (mean residual of 0.88 pixel, max residual of 3.27 pixel), and the parallel projection results were the least accurate. This data is summarized in Table 6.1 along with the results of the test using standard methods.

In addition to the total residuals, the set of *x*- and *y*- residuals were also considered (as calculated with equation 3.8 and A.30). Because these residuals are not calculated by a square-root function, they are distributed about zero. Figure 6.1 shows a histogram of the residuals, along with a normal distribution with the means and standard deviations given in Table 6.2.

Table 6.1. 1-sigma errors and residuals from the eight-image test with the CANON 5D

	1-sigma mean (~m)	1-sigma max (~m)	total residual mean (pixel)	total residual max (pixel)
Standard Methods	0.006	0.015	0.88	3.27
Parallel Projection	0.062	0.202	9.45	56.47
Perspective-Corrected Parallel Projection	0.006	0.016	0.88	3.40
Perspective Projection	0.006	0.014	0.93	3.34

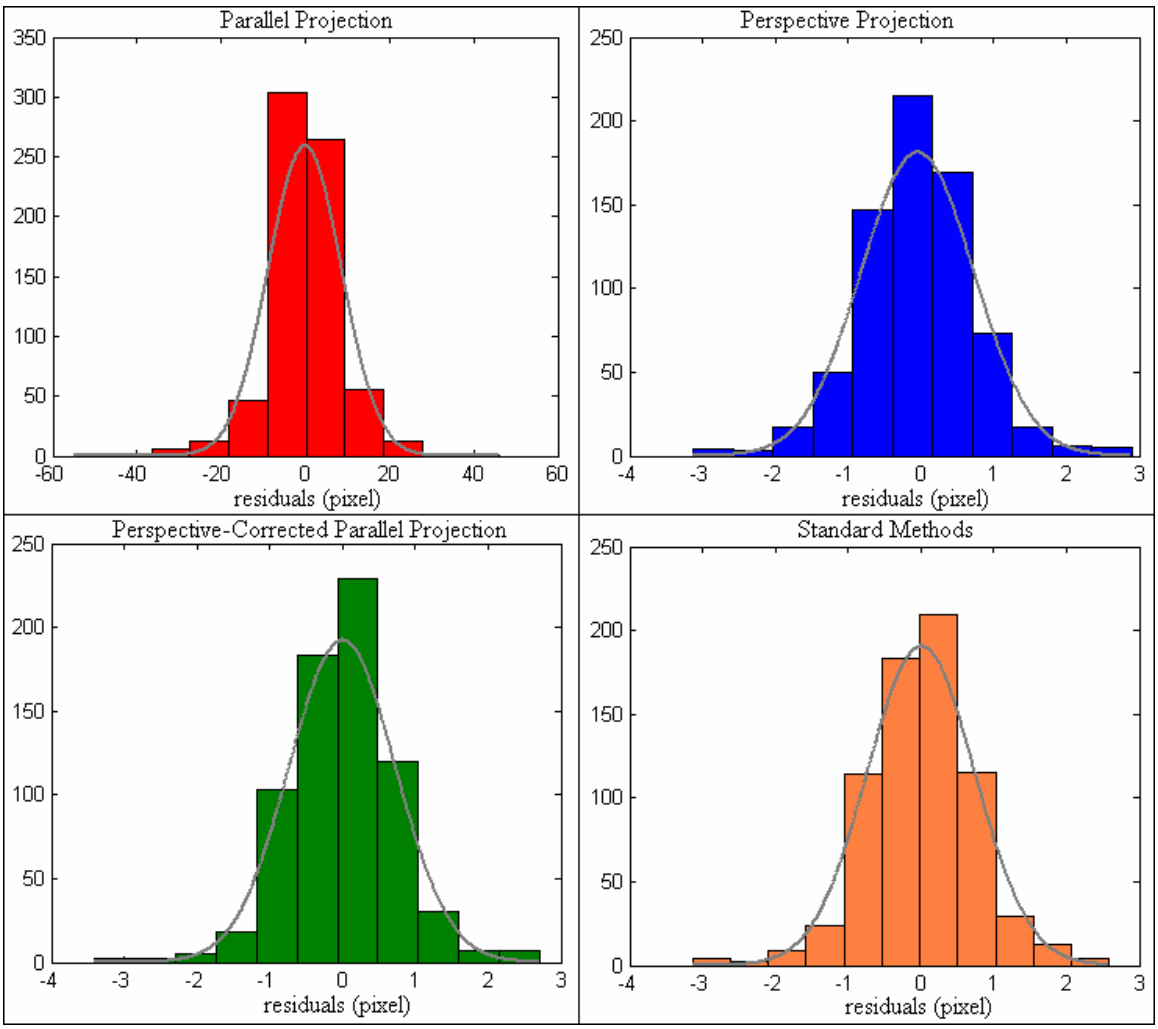

Figure 6.1. Histograms of residuals for eight-image Haddock test.

Table 6.2. Mean and standard deviation of residuals from the eight-image test with the CANON 5D

	Mean (pixel)	Standard Deviation (pixel)
Standard Methods	0.01	0.74
Parallel Projection	0.00	8.69
Perspective-Corrected Parallel Projection	0.00	0.73
Perspective Projection	-0.04	0.78

Figure 6.1 reveals that the residuals for each projection follow a normal distribution very well. Recall that in the calculation of the 1-sigma errors it was assumed that the residuals follow a normal distribution. Figure 6.1 gives confidence in that assumption.

Given the above, the comprehensive algorithm was able to produce a 3D model with comparable precision to the model created using standard methods.

6.2 Haddock Images at Medium Range with the CANON 20D

The second set of images used to test the comprehensive algorithm was the series of six images, also of the Coast Guard Vessel Haddock, used in the earlier tests. Recall that the images were taken with an 8.2 megapixel CANON 20D digital camera with a 400 mm lens at distances ranging from approximately 1100 m – 1700 m with an average of about 1475 m. The six images that were used are shown in Figure 2.5 in Chapter 2.

Recall that these images were taken at a greater range than the previous images resulting in a narrower angular field of view. Recall also that when using the standard procedures on this image set, some complications were encountered. There was some instability of the initial approximation algorithms depending on the pair of images that were used. Also, the order in which subsequent images were added using the DLT affected the solution. Despite the complications, an accurate solution was eventually obtained. However, for this image set it was desired to increase the reliability of the initial approximations and the process for adding images to a previous solution.

The inputs from this image set were as before; there was a set of 46 correspondences collected manually across the six images, the IOP (*f, xo*, *yo*) were

approximated as before, the scale was chosen to match the size of the model obtained from the eight-image test, and a keypoint was selected. The procedure of the algorithm was carried out in the same manner as for the eight-image test.

The resulting 3D coordinates from each projection were compared by a 3D rigid transformation with the results from the standard methods. The mean error for the parallel projection was 0.035 ~m (0.116 ft), the mean error for the perspective-corrected parallel projection was 0.020 ~m (0.065 ft), and the mean error for the perspective projection was 0.019 ~m (0.063 ft). Thus, the resulting 3D models were similar up to the order of precision—recall that the 1-sigma error for the model from the standard test was 0.036 ~m.

Also, as mentioned above, there were 38 corresponding points between the six-image model and the eight-image model. For each projection, a 3D rigid coordinate transformation was used to compare the two models. For the parallel projection the mean error was 0.100 ~m (0.328 ft), for the perspective-corrected parallel projection the mean error was 0.055 ~m (0.180 ft), and for the perspective projection the mean error was 0.053 ~m (0.174 ft). These show that for each projection the similarity of the two models is on the same order as the precision of the models.

The 1-sigma errors and total residuals of the six-image test are summarized in Table 6.3 along with the results from the test using standard methods. Note that the results from the all three projections are very similar to the results obtained with the standard methods. Thus, the algorithm was able to match the precision of the standard methods without the hassle of trying different image pairs and without the instability of the DLT.

Table 6.3. 1-sigma errors and residuals from the six-image test with the CANON 20D

	1-sigma mean (~m)	1-sigma max (~m)	total residual mean (pixel)	total residual max (pixel)
Standard Methods	0.036	0.172	1.16	3.41
Parallel Projection	0.035	0.113	1.27	4.11
Perspective-Corrected Parallel Projection	0.032	0.121	1.09	3.36
Perspective Projection	0.033	0.123	1.11	3.42

6.3 Petrel Images at Further Range with the CANON 1Ds

The third set of images used to test the comprehensive algorithm was the series of four images, of the Coast Guard Vessel Petrel, used in the earlier tests. Recall that the images were taken with a 16.7 megapixel CANON 1Ds digital camera with an 800 mm lens at distances ranging from approximately 3000 m – 4300 m with an average of about 3550 m. The four images that were used are shown in Figure 2.6 in Chapter 2.

Recall that these images were taken at a greater range than the previous images resulting in a narrower angular field of view. Recall also that when using the standard photogrammetry procedures on this image set an accurate 3D reconstruction was not achieved.

The inputs from this image set were as before; there was a set of 46 correspondences collected manually across the four images, the IOP (*f, xo*, *yo*) were approximated as before, the scale was chosen to match the size of the model obtained from the six-image test, and a keypoint was selected. The procedure of the algorithm was carried out in the same manner as for the eight- and six-image tests.

The comprehensive algorithm was able to produce three sets of 3D coordinates. As mentioned above, the 46 points on the six-image model matched the points for the four-image model. For each projection, a 3D conformal coordinate transformation was used to compare the two models. For the parallel projection the mean error was 0.110 ~m (0.361 ft), for the perspective-corrected parallel projection the mean error was 0.112 ~m (0.367 ft), and for the perspective projection the mean error was 0.113 ~m (0.369 ft). These show that for each projection the similarity of the six-image model and four-image model is on the same order as the precision of the individual models. It is possible that there was some additional error due to the fact that the two targets were not the same vessel, just from the same class. This gives confidence that the 3D reconstruction was accurate.

The 1-sigma errors and total residuals of the four-image test along with the results from the standard method tests are summarized in Table 6.4. These values are very similar to those obtained for the six-image model. Thus, the algorithm was able to perform an accurate 3D reconstruction on the four Petrel images, which was not achieved with the standard methods.

Table 6.4. 1-sigma errors and residuals from the four-image test with the CANON 1Ds

	1-sigma mean (~m)	1-sigma max (~m)	total residual mean (pixel)	total residual max (pixel)
Standard Methods	*	*	*	*
Parallel Projection	0.034	0.109	0.67	3.80
Perspective-Corrected Parallel Projection	0.031	0.111	0.62	3.84
Perspective Projection	0.031	0.113	0.63	3.58

* indicates that an accurate result was not achieved

These tests indicate that the comprehensive algorithm performed equal to, or better than, the standard methods for these particular surveillance image sets. Some further tests using synthetic images are discussed in the next chapter.

CHAPTER 7

TESTS OF THE COMPREHENSIVE ALGOTIHM USING SYNTHETIC IMAGES

Some tests using synthetic image sets will now be discussed. The purpose of these synthetic tests is to enable a prediction of algorithm performances at a variety of distances, with a variety of camera resolutions, and with a variety of measurement errors.

7.1 Test Set-up and Procedure

The first set of synthetic images was designed to match the Haddock eight-image test. Because of the rather small imaging distances it was anticipated that the parallel projection would provide the least accurate result, and that obtaining initial approximations through parallel projection may lead to complications.

First, a set of synthetic 3D points was created. A set of points which roughly form the outline of a boat were created. Figure 7.1 shows these points. The dimensions of the synthetic coordinates were chosen to roughly match the dimensions of the actual Coast Guard vessel Haddock. For the synthetic coordinates the height of the hull (the portion visible above the water) was 1.8 ~m (5.9 ft), the length of hull was 30.6 ~m (100.4 ft), the width of the hull was 6.0 ~m (19.7 ft), the length of the cabin was 10.4 ~m (34.1 ft), the width of the cabin was 4.5 ~m (14.8 ft), the height of the cabin was 5.0 ~m (16.4 ft), and the height of the superstructure was 7.3 ~m (24.0 ft).

Next, the EOP had to be created. For creating these synthetic camera orientations, an alternate relation between the 3D coordinate system (**XYZ**) and image coordinate system (*xyz*) was created. This alternate definition was created in order to more easily

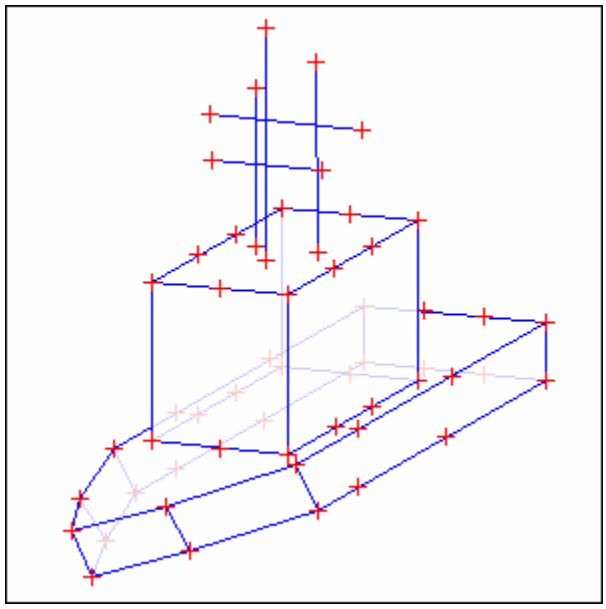

Figure 7.1. 3D synthetic XYZ coordinates.

visualize and create the camera orientations. This alternate definition involves three rotation angles, denoted by $(\theta, \psi, \varepsilon)$ which are different from $(\omega, \varphi, \kappa)$ used earlier, a distance *d*, and a point (*Xo*, *Yo*, *Zo*) in 3D space toward which the camera is pointed. Figures 7.2, 7.3, and 7.4 describe this alternate relation of 3D space to the image coordinate system. The major motivation for this alternate definition between 3D space and the image coordinate system is that the seven parameters $(Xo, Yo, Zo, d, \theta, \psi, \varepsilon)$ are easily visualized and chosen, while the standard EOP $(Xc, Yc, Zc, \omega, \varphi, \kappa)$ can be difficult to visualize—especially the three rotation angles. The point (*Xo*, *Yo*, *Zo*) is a point on the target toward which the camera is pointing, *d* is the distance from the target to the camera, θ describes the horizontal angle to the camera, ψ is the vertical angle to the camera, and ε is the amount of rotation about the *z*-axis of the camera.

In Figure 7.2, **XYZ** denotes the 3D coordinate system, and X'Y'Z' is a shifted coordinate system whose axes are parallel to **XYZ** but with its origin at (*Xo*, *Yo*, *Zo*) in the **XYZ** system. The perspective center of the camera has 3D coordinates (*Xc*, *Yc*, *Zc*). The *z*-axis (focal axis of the camera) extends from (*Xo*, *Yo*, *Zo*) to (*Xc*, *Yc*, *Zc*). The z'-axis lies in the X'Y'-plane and extends from (*Xo*, *Yo*, *Zo*) to the point (*Xc*, *Yc*, *Zo*); i.e. it is the projection of the *z*-axis in the Z' direction onto the X'Y'-plane. The angle θ is the horizontal angle in the X'Y' plane from the Y' axis to the z'-axis. The angle θ is positive when it is measured clockwise[2] from the Y' axis as viewed from the positive end of the Z' axis. The angle ψ is the vertical angle in the z'Z' plane from the z'-axis to the *z*-axis. The angle ψ is positive when measured from the positive z' axis in the positive Z'-direction. The distance between (*Xo*, *Yo*, *Zo*) and (*Xc*, *Yc*, *Zc*) is denoted by *d*.

Now, the orientation of the x'y'*z*-axes are shown in Figure 7.3. Figure 7.3 shows that the perspective center (*Xc*, *Yc*, *Zc*) is the origin of the x'y'*z* coordinate system. The x' axis is perpendicular to the *z*-axis and is parallel to the X'Y' plane. The y' axis is mutually orthogonal to the x' and *z* axes.

Finally, the third rotation angle ε describes the full orientation of the image coordinate system (*xyz*). Figure 7.4 shows the x'y' axes and the *xy* axes as viewed from the positive *z*-axis. The *xy*-axes of the image coordinate system are rotated about the *z*-axis with respect to the x'y' axes. The rotation ε is positive when it is clockwise as viewed from the positive *z*-direction. Thus, the orientation of the image coordinate system (*xyz*) with respect to the **XYZ** coordinate system has been defined in terms of the seven parameters $(Xo, Yo, Zo, d, \theta, \psi, \varepsilon)$.

[2] This is not a standard convention.

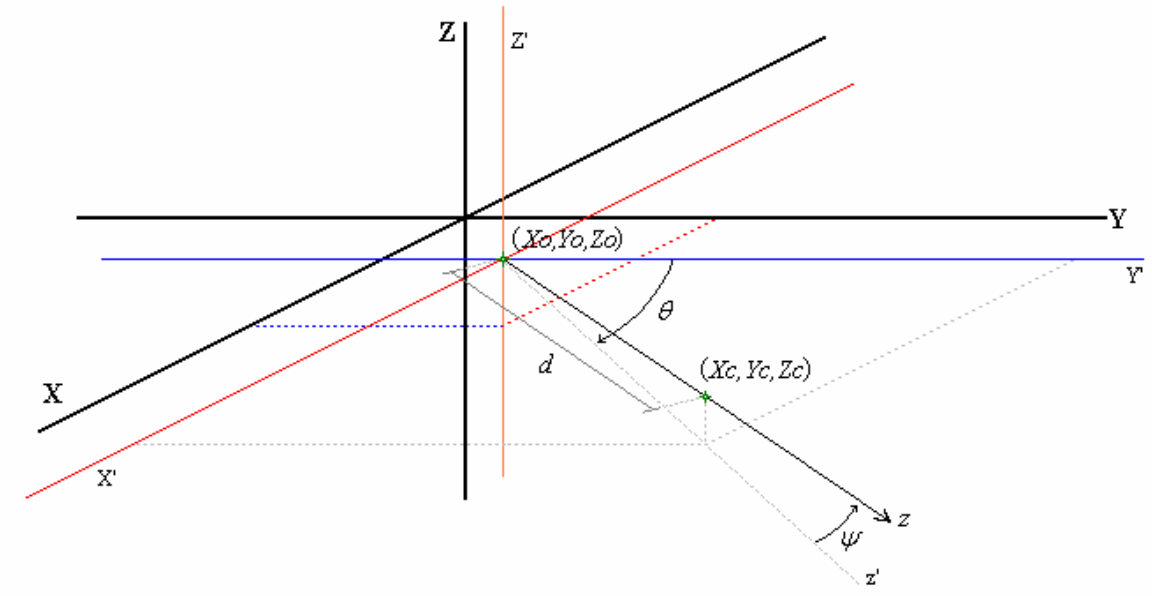

Figure 7.2. Alternate definition of relation between coordinate systems.

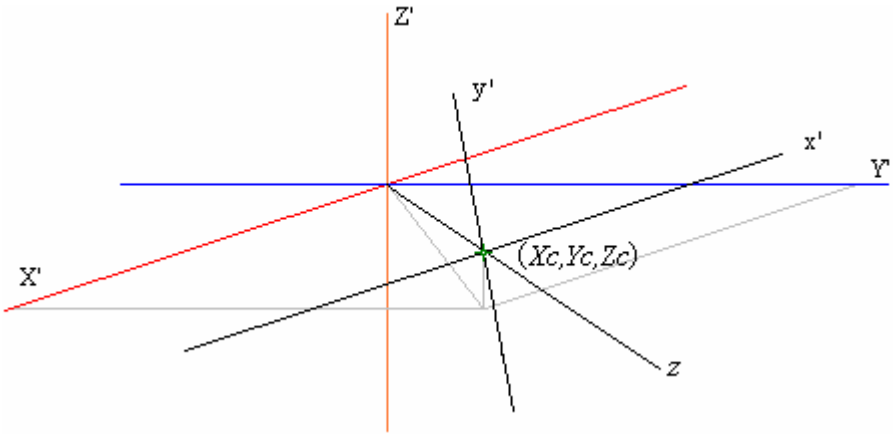

Figure 7.3. The x'y'z-axes in 3D space.

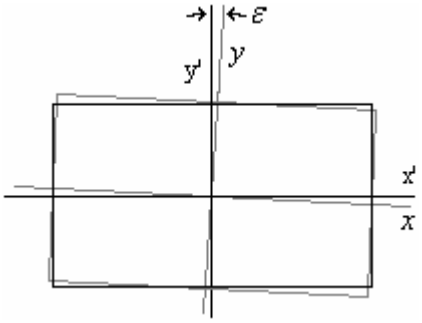

Figure 7.4. Third rotation angle ε.

The image plane is perpendicular to the *z*-axis and parallel to the *xy* plane and intersects the *z*-axis at the principal point. The image plane is a distance of *f* (the principal distance) from the perspective center in the negative *z*-direction. Finally, there is a coordinate system on the image plane which is parallel to the *xy*-plane of the image coordinate system, but has a shift such that the principal point has coordinates (*xo*, *yo*).

Eight synthetic EOP were created. The seven parameters $(Xo, Yo, Zo, d, \theta, \psi, \varepsilon)$ for each image are shown in Table 7.1. The values of θ are at fairly even intervals on a circle around the target, similar to the eight Haddock images. The values of the angle ψ simulate each of the synthetic images being taken from a slightly higher elevation than the origin of 3D space (the origin of 3D space lies on the deck of the ship near the center of the boat). The small values of the angle ε denote that the images were taken with a camera that was held quite level. The values for *Xo*, *Yo*, and *Zo* show that each camera is pointed near the origin of 3D space (near the center of the boat). Finally, the values of *d* match the imaging distances that were calculated from the set of eight Haddock images using the perspective projection EOP. Figure 7.5 shows the synthetic boat and the positions of the perspective centers of the cameras.

Table 7.1. Camera orientations for synthetic image test

	θ (deg)	ψ (deg)	ε (deg)	Xo (m)	Yo (m)	Zo (m)	d (m)
image 1	23.0	3.2	-1.2	0.3	2.2	4.4	337.4
image 2	76.0	2.4	1.4	2.1	-0.2	5.2	422.2
image 3	112.0	3.1	0.8	-1.5	-1.8	3.4	448.4
image 4	161.0	3.6	-0.7	0.7	-2.0	3.9	291.7
image 5	212.0	1.9	1.1	1.4	0.9	3.5	304.9
image 6	248.0	2.2	-1.3	-1.4	-1.8	4.1	379.7
image 7	302.0	3.2	1.5	-0.6	-1.1	4.2	403.4
image 8	331.0	1.8	-0.6	0.7	-3.9	3.8	300.2

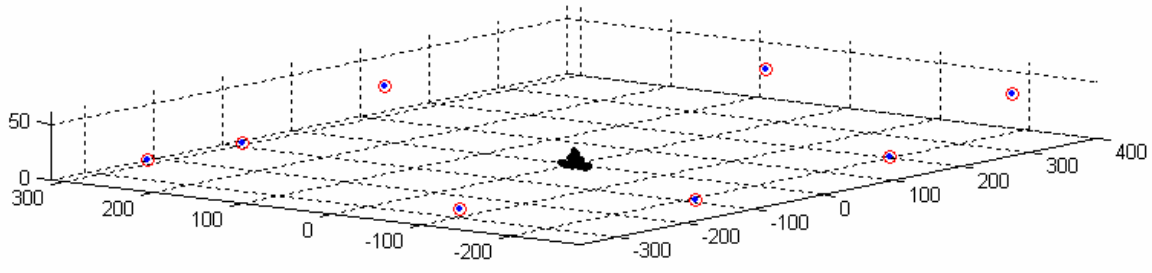

Figure 7.5. 3D representation of synthetic boat and camera positions.

Once these seven parameters were chosen for each image, the coordinates of the perspective center of each camera (Xc, Yc, Zc) could be solved for:

$$\begin{aligned} Xc &= Xo + d\sin(\theta)\cos(\psi) \\ Yc &= Yo + d\cos(\theta)\cos(\psi) \\ Zc &= Zo + d\sin(\psi) \end{aligned} \tag{7.1}$$

The rotation matrix terms were found from the three angles for each of the synthetic images with equations (A.13) in the appendix. It was assumed that each of the synthetic images was created with the same IOP. The IOP were chosen such that they approximated the CANON 5D digital camera which was used to capture the first set of

Haddock images. The IOP were chosen as f = 48770 pixel, xo = 2184, and yo = 1456 pixel.

With all of the terms defined, synthetic image coordinates were calculated by the collinearity equations (1.1).

The next step in creating the synthetic images was to add some random error to account for slight errors in selecting point correspondences and deviation from a true pin-hole camera model. A random error from a normal distribution with mean 0.00 pixel and standard deviation 0.75 pixel was added to each image coordinate. The magnitude of this random error was chosen to match the residuals from the perspective projection results of the set of eight Haddock images. This means that for each image coordinate, there was a 68% probability that the measured coordinate was within 1.50 pixel of the true coordinate, and there was a 95% probability that the measured coordinate was within 3.00 pixel of the true coordinate.

The synthetic images that were created from the set of XYZ coordinates (Figure 7.1) and the EOP (Table 7.1) are shown in Figure 7.6.

Next, in order that the set of synthetic image-coordinates would more closely match the actual image-coordinates, certain image-coordinates were deleted from each image. This is to account for points which are not visible in a given image. Figure 7.7 shows an image with points removed. When deleting points, it was necessary to keep each point visible in at least two images, it was necessary also that when considering any two images there were at least four common points visible in both of the images, and it was necessary to have at least one point which was visible in all images. All of these constraints were met after deleting points.

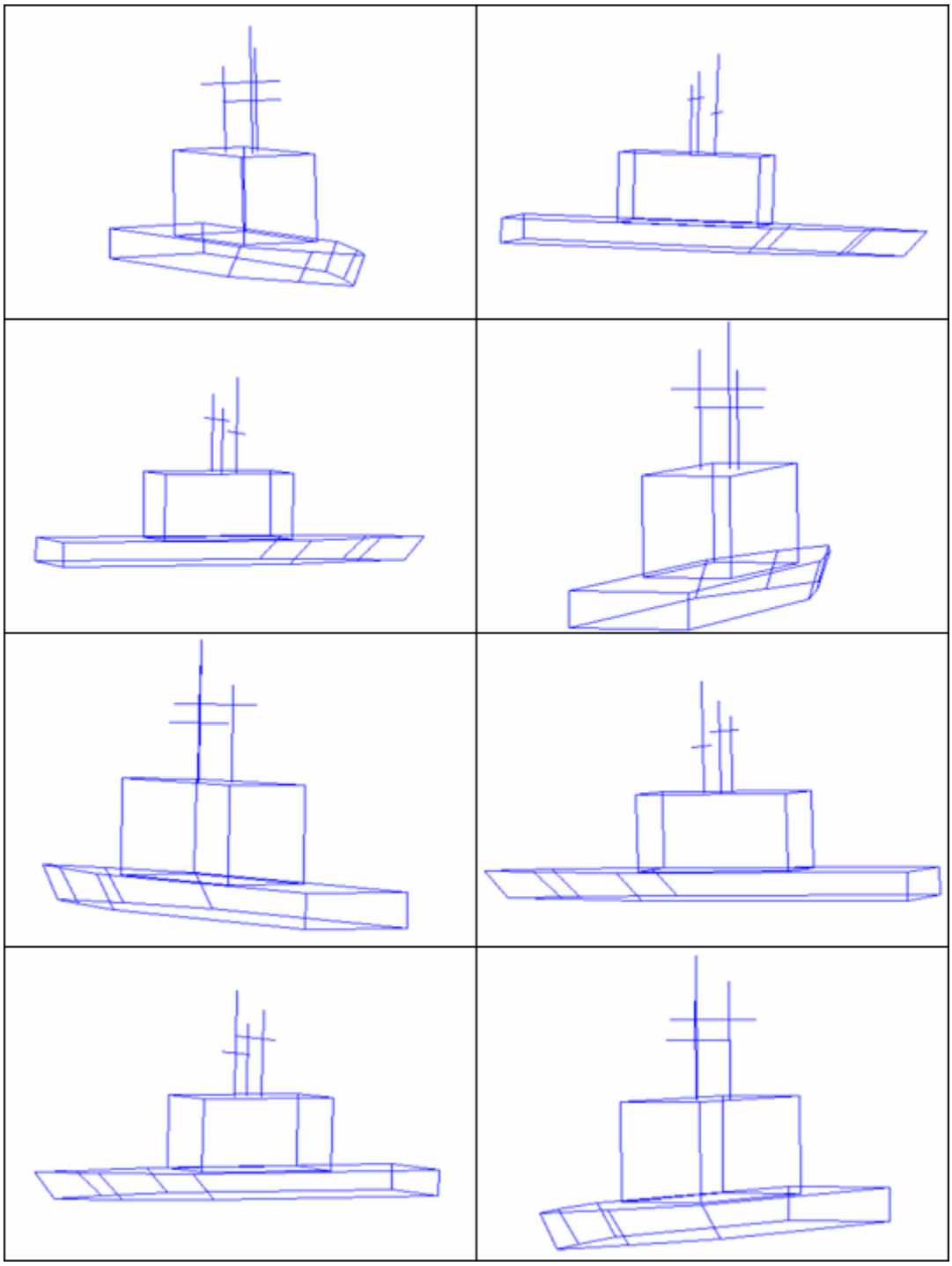

Figure 7.6. Eight synthetic images.

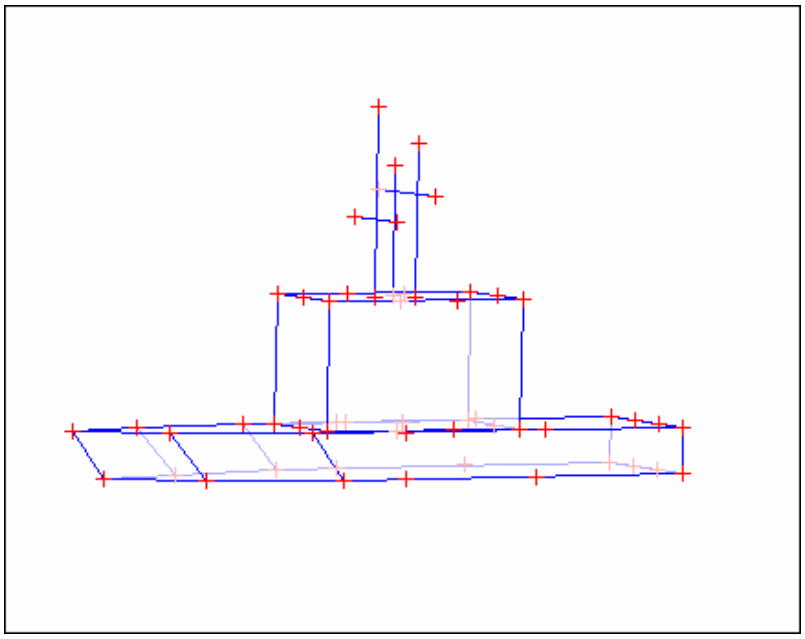

Figure 7.7. Synthetic image with points removed.

Once the set of synthetic image coordinates were created it was desired to perform 3D reconstruction from the images using both standard methods and the comprehensive algorithm. The set of image coordinates (point correspondences) was the first input to the algorithms.

The second input for the algorithms is the approximate IOP. It was assumed that these were known, and hence the values that were used to create the images were used as the approximate IOP: f = 48770 pixel, xo = 2184 pixel, and yo = 1456 pixel.

The third input for the comprehensive algorithm is the scale of the first image. Again, this value determines the scale and units of the final 3D coordinates. It was desired to have the final coordinates in units of (m), and the image coordinates in pixels. Hence, by equation (3.36), this value was chosen as the ratio of the principal distance to the imaging distance for image A (48770 pixel / 337.4 m).

Finally, a keypoint had to be selected. A point which was closer to the camera on the first image than the centroid of the target was chosen.

With all of the inputs defined, the standard photogrammetry methods were carried out as explained in Chapter 2, and the comprehensive algorithm was executed as explained in Chapter 5.

7.2 Test Results

First, a set of XYZ coordinates were calculated using each projection. Because the actual XYZ coordinates are known, a 3D conformal coordinate transformation was used to convert the calculated coordinates into the same reference frame as the actual coordinates. The actual coordinates are shown along with the calculated coordinates for each of the outputs in Figure 7.8.

The root mean squared error (RMSE) between actual coordinates and calculated coordinates was calculated for each point by:

$$\text{RMSE} = \sqrt{(X - X')^2 + (Y - Y')^2 + (Z - Z')^2} \tag{7.2}$$

The RMSE for this test are given in Table 7.2 along with the 1-sigma errors. These RMSE reflect a high degree of precision. They also show that the perspective-corrected parallel projection and the perspective projection were more precise than the parallel projection for this image set.

The second output is the set of 1-sigma errors. Table 7.2 shows a comparison of the RMSE errors and the 1-sigma error estimates. Note that the 1-sigma errors are on the same order as the true errors (RMSE). This reveals that for this synthetic image test, the

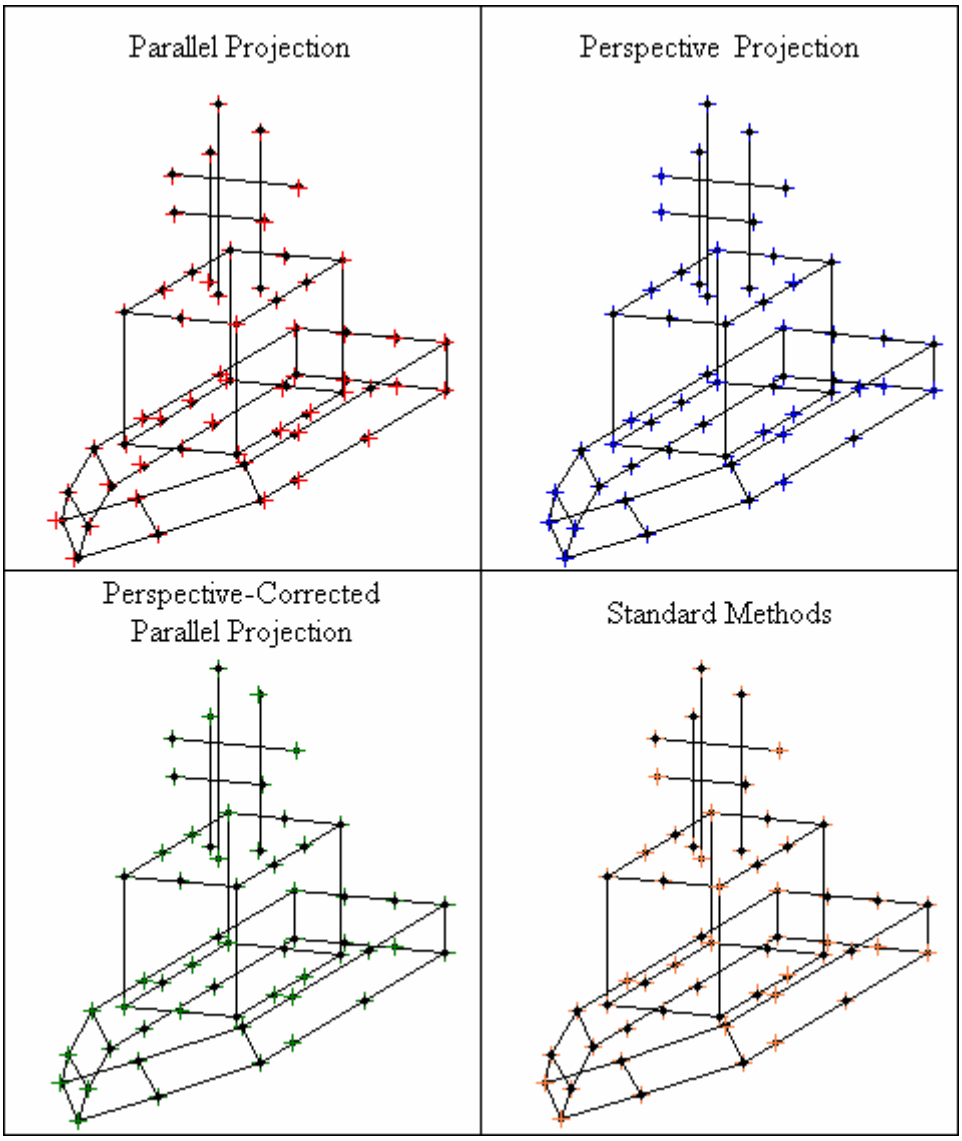

Figure 7.8. 3D plot of synthetic coordinates and calculated coordinates.

calculated 1-sigma errors were indeed a good estimate of the precision of the calculated 3D coordinates. This adds confidence in using 1-sigma errors as a measure of precision of the algorithm. It is noted that the maximum 1-sigma error is about half of the maximum RMSE for each projection. This is because the 1-sigma error is a statistical measure of one standard deviation. If the errors follow a normal distribution, then 68% of the data points should fall within one standard deviation, and 95% of the data should

fall within two standard deviations, etc. Hence, the maximum RMSE (the RMSE for the "worst" point) corresponds roughly to 2 times the highest 1-sigma error estimate.

The next output is the set of residuals and back-projected image coordinates. The histograms of residuals are shown in Figure 7.8 along with the normal distribution. These histograms show that the residuals do follow a normal distribution quite well. Recall that the calculation of 1-sigma errors is based on the assumption that the residuals do follow a normal distribution. The total residuals were also considered. They are summarized in Table 7.3. Recall that the image noise was chosen such that 95% of the measured image coordinates were within 3.00 pixel of the actual image coordinates. The residuals from the parallel projection are higher than the level of noise reflecting the fact that parallel projection is not a valid model for this image set. The residuals from the perspective-corrected parallel projection and perspective projection were less than this noise reflecting that they are very good models for this image set. This also demonstrates the ability of the algorithm to produce a solution which is accurate up to the level of error introduced at the beginning of the test.

Table 7.2. Comparison of true error (RMSE) and 1-sigma error estimates for the synthetic image test

	RMSE		1-sigma error	
	mean (m)	max (m)	mean (m)	max (m)
Standard Methods	0.006	0.015	0.005	0.015
Parallel Projection	0.080	0.255	0.054	0.127
Perspective-Corrected Parallel Projection	0.006	0.014	0.005	0.014
Perspective Projection	0.006	0.015	0.005	0.014

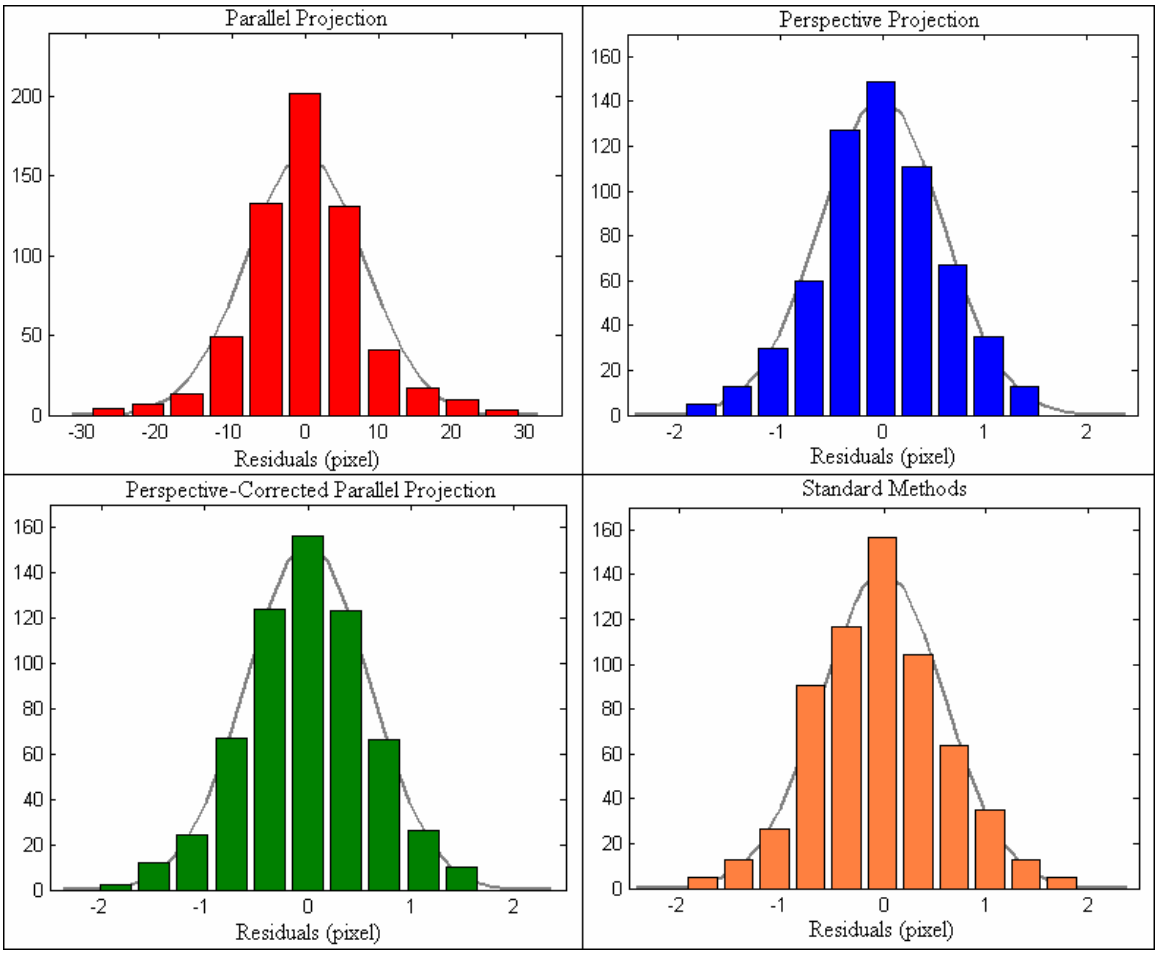

Figure 7.9. Histograms of residuals for synthetic image test.

Table 7.3. Total residuals for the synthetic image test

Total residuals		
	Mean	Max
	(pixel)	(pixel)
Standard Methods	0.73	2.06
Parallel Projection	9.14	47.88
Perspective-Corrected Parallel Projection	0.72	1.95
Perspective Projection	0.74	1.94

7.3 Comparison of Actual Eight-Image Test and Synthetic Image Test

Some of the similarities and differences between the actual eight-image test and the synthetic image test are now discussed.

First, the synthetic images were created by the collinearity equation which assumes an exact pin-hole camera model. This assumes that the light rays travel in perfectly straight lines from the target, through the perspective center to the image. The actual images were created from a camera which does not follow the pin-hole camera model. In a camera, the light rays may not all travel through the same point; i.e., an actual camera may not have a true perspective center. Actual images are also subject to distortions, such as lens distortion, distortion from heat radiation, etc...

In order to try to compensate for the above, random error was added to the synthetic images. This error followed a normal distribution, which may or may not accurately characterize the distortions in the actual images.

Second, point correspondences (image-coordinates) were measured from the actual images by visually selecting points in the images. This process adds additional error to the solution, as coordinates cannot be exactly measured from the images. As mentioned before, coordinates were measured essentially to the nearest pixel. Conversely, the synthetic image coordinates were calculated as exact values. Again, the added random error compensates somewhat, assuming that point correspondence errors follow a random normal distribution.

Third, the IOP for the synthetic images were known exactly and exact values were used in the algorithm. For the actual images, the IOP were not known exactly and

approximations were used. The amount of error between the approximate IOP and true IOP for the actual images is not known.

Fourth, the two sets of images were of different targets with different geometries. While the dimensions of the synthetic target were chosen to roughly match the Haddock, the target geometries were somewhat different. This has an effect on the solution.

Fifth, the EOP were not the same. Both sets of EOP roughly formed a circle around the target, the elevations of the cameras were slightly higher than the deck of the target in both cases, and similar distances were chosen for each test. However, the EOP can have a large effect on the precision of the algorithm.

Considering all of these differences, Table 7.4 is presented which summarizes the results from each of the two sets. Note that the values from both tests are very similar for each model. This demonstrates that despite the differences, the synthetic image test can simulate actual images with a good degree of precision. Based on this fact, several more synthetic image tests will be considered.

Table 7.4. 1-sigma errors and residuals from the actual eight-image test and synthetic eight-image test

	Actual Haddock Images				Synthetic Images			
	1-sigma errors		residuals		1-sigma errors		residuals	
	mean (~m)	max (~m)	mean (pixel)	max (pixel)	mean (m)	max (m)	mean (pixel)	max (pixel)
Standard Methods	0.006	0.015	0.88	3.27	0.005	0.015	0.73	2.06
Parallel Projection	0.062	0.202	9.45	56.47	0.054	0.127	9.14	47.88
Perspective-Corrected Parallel Projection	0.006	0.016	0.88	3.40	0.005	0.014	0.72	1.95
Perspective Projection	0.006	0.014	0.93	3.34	0.005	0.014	0.74	1.94

7.4 Additional Synthetic Image Tests

Based on the above test, 80 sets of synthetic images were generated, with eight images in each set. Five sets of images were created at a series of 16 different base imaging distances *d*. The 16 base distances were: 200 m, 300 m, 400 m, 500 m, 600 m, 800 m, 1000 m, 1200 m, 1400 m, 1600 m, 1800 m, 2000 m, 2400 m, 2800 m, 3400 m, and 4000 m. The major motivation for this series of tests was to show the trend as the distance to the target increases (angular field-of-view becomes narrower).

For all of these image sets, the synthetic boat target outlined above was used. For each image set at each base distance, there were eight images created using the seven camera parameters $(Xo, Yo, Zo, \theta, \psi, \varepsilon, d)$. The values of (*Xo*, *Yo*, *Zo*) were chosen from uniform random distributions which ranged across the dimensions of the target for all images—i.e. it was assumed that the camera was pointed at a random point on the target for each image. The base values of the horizontal angle θ for the eight images were chosen as 0°, 45°, 90°, 135°, 180°, 225°, 270°, and 315° (eight even intervals around a circle). For each individual image set, a random perturbation was added to each θ from a normal random distribution with mean 0° a standard deviation of 15°. This simulates a series of eight images that are roughly in a circle around the target. For each image the vertical angle ψ was chosen from a uniform random distribution from 0° to 10°. This simulates a series of images that are captured either at the same elevation or slightly higher than the target. The third angle ε was chosen from a standard normal distribution with mean 0° and standard deviation 2°. This simulates a situation where the camera is held fairly level for each image. The seventh parameter *d* was chosen from a uniform

random distribution centered at the base distance with a range equal to 80% of the base distance. For example, for the 200 m image sets, the eight imaging distances were chosen from a uniform random distribution with range 160 m centered at 200 m [120 m, 280 m]; for the 300 m image sets, the eight imaging distances were chosen from a uniform random distribution with range 240 m centered at 300 m [180 m, 420 m]; etc…

There were three sets of IOP used for the images. For each eight-image set, three images were based on the approximate IOP of the CANON 5D with a 400 mm lens: f = 48770 pixel, xo = 2184 pixel, and yo = 1456 pixel; three images were based on the approximate IOP of the CANON 20D with a 400 mm lens: f = 62293 pixel, xo = 1752 pixel, yo = 1168 pixel; and two images were based on the approximate IOP of the CANON 1Ds with an 800 mm lens: f = 110934 pixel, xo = 2496 pixel, yo = 1664 pixel. These values were used as the approximate IOP which were input to the algorithms. However, when each image was generated, random perturbations were added to the IOP to simulate a situation where the IOP are not known precisely. A normal random error with mean 0 and standard deviation equal to 2.5% of the principal distance was added to each principal distance. A normal random error with mean 0 and a standard deviation of 10 pixel was added to each principal point coordinate.

For the images with the CANON 1Ds parameters (800 mm lens), the imaging distance was doubled to simulate images captured at greater distances with a longer lens.

With these inputs defined, image coordinates were generated with the collinearity equations as before. After the coordinates were generated, random noise was added to each of the image coordinates and points were removed from each image in the same manner as explained above.

Each of the image sets were considered separately; for each image set, the parallel projection initial approximation algorithm was performed using 6-10 points on four of the images from the image set. Next, the four remaining images were added one-by-one using the algorithm for adding an additional image by parallel projection. Then, once all 8 images had an initial approximation for the EOP, each of the remaining points were added one-by-one. After adding each point the 3D reconstruction and bundle triangulation algorithms were used to continually update the EOP. Also, for each image set, the standard methods were carried out using the essential matrix algorithm as explained in Chapter 2.

For each image set, the following values were considered: the RMSE between the calculated 3D points and the true 3D points, the 1-sigma errors, and the residuals. Figure 7.10 shows a plot of the RMSE for each test plotted against the average imaging distance. In the cases where the standard methods failed to converge, a value of 0 is plotted.

Note that at smaller distances (200 m – 2000 m), the perspective projection and perspective-corrected parallel projection provide more precise results than the parallel projection. However, beyond 2000 m as the field-of-view becomes narrower the three projections produce comparable results. Of greatest importance is the fact that for all 80 image sets, the comprehensive algorithm converged to an accurate result. This demonstrates the ability of the algorithm to solve for the 3D geometry of the target with high precision at a large range of distances, or fields-of-view. It also demonstrates the ability of the algorithm to incorporate images from a variety of sensors.

Note that as the imaging distance increases beyond 1000m the standard results generally did not converge to an accurate solution.

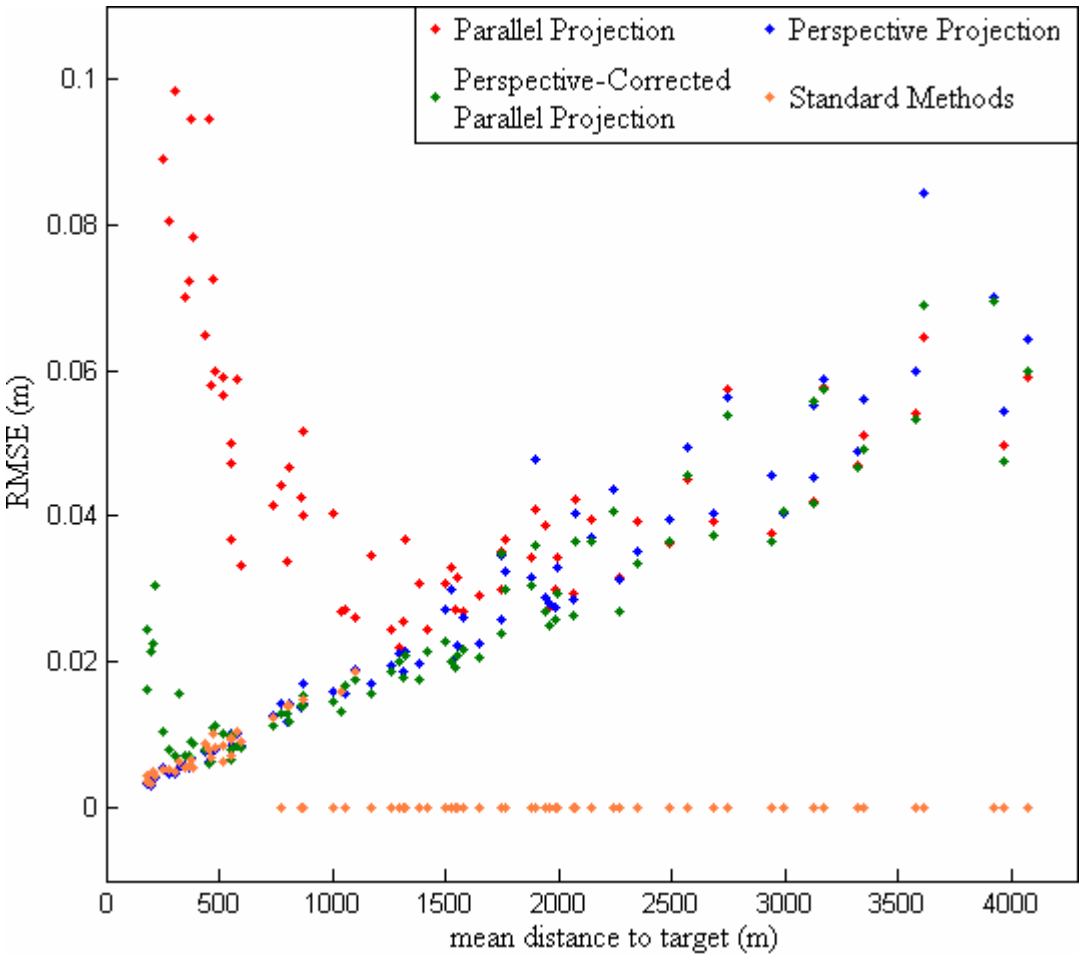

Figure 7.10. RMSE from each projection for 80 synthetic image tests.

Figure 7.11 shows a plot of the 1-sigma errors for each test plotted again the average imaging distance. As before, in the cases where the standard methods failed to converge to a result, a value of 0 is plotted. Note first that by comparing Figure 7.10 with Figure 7.11 the 1-sigma errors from the synthetic test are a good approximation of the actual errors. The same trends that were seen in the RMSE are reflected by the 1-sigma errors. This gives confidence in using the 1-sigma errors as a metric of accuracy for a given model. Also note that the 1-sigma errors from the actual image tests fit this data

very well. Table 7.5 gives the 1-sigma error for those tests. This gives additional confidence in the validity of the synthetic tests.

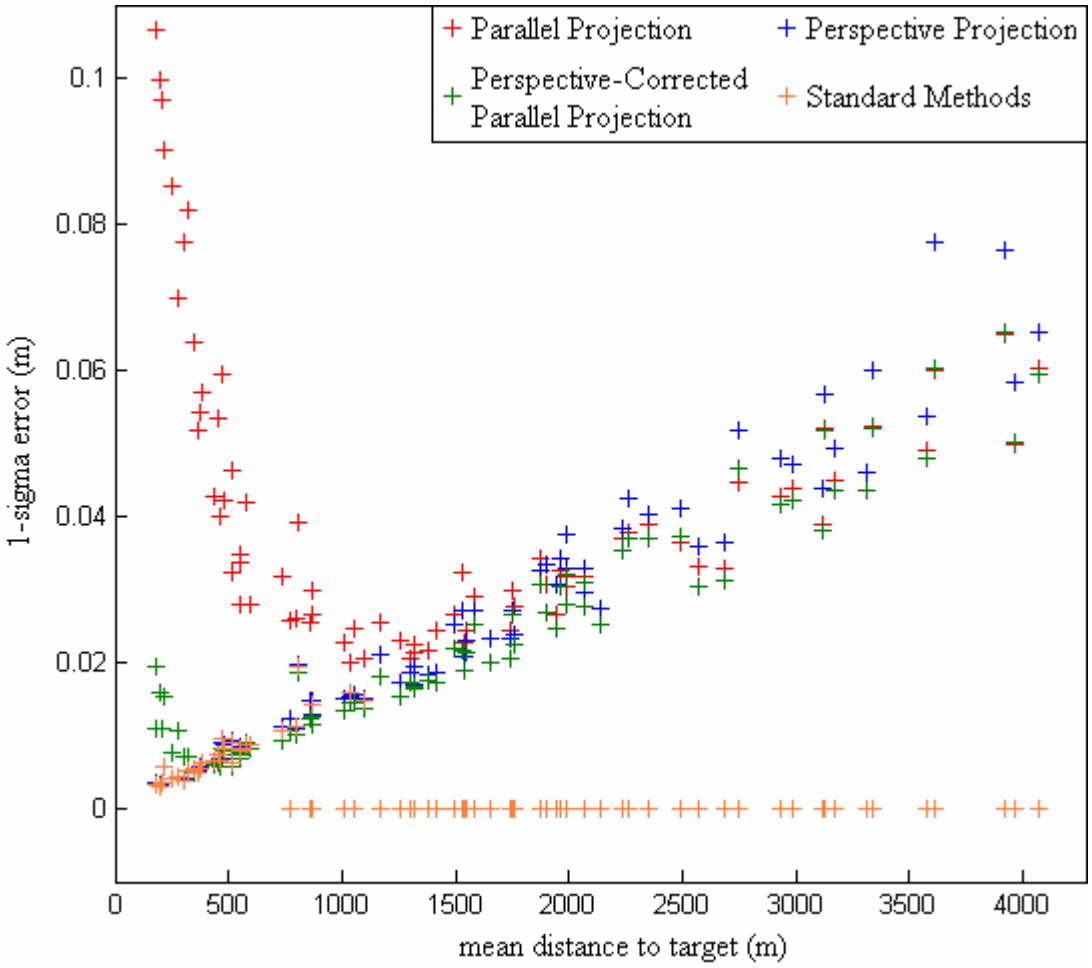

Figure 7.11. 1-sigma errors from each projection for 80 synthetic image tests.

Table 7.5. Mean 1-sigma errors for actual image tests

Mean 1-sigma errors (~m)			
	Eight-image test	Six-image test	Four-image test
	~360 m	~1475 m	~3550 m
Standard Methods	0.006	0.036	*
Parallel Projection	0.062	0.035	0.034
Perspective-Corrected Parallel Projection	0.006	0.032	0.031
Perspective Projection	0.006	0.033	0.031

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

This thesis has outlined a mathematical model which can perform photogrammetric 3D reconstruction of target objects from varying sets of surveillance imagery. A comprehensive algorithm has been developed using principles of parallel projection and perspective projection. The developed model has followed closely, and expanded upon, previous work in photogrammetry and computer vision.

The key problems posed by surveillance imagery are: 1) narrow angular field of view, 2) lack of ground control or sensor orientation information, 3) camera calibration, and 4) images captured from various aspects. These problems have been overcome in the following ways:

1) The problems associated with a narrow angular field of view have been eliminated by adopting the parallel projection as opposed to the perspective projection. The parallel projection is a more robust model for images with a narrow angular field of view (Morgan, 2004). The perspective-corrected parallel projection has also been developed based on the orthogonal projection model by Ono and Hattori (2002). This projection bridges the gap between the parallel projection and perspective projection and allows for a solution from the parallel projection and the standard perspective projection used commonly in photogrammetry.

2) Because most surveillance images lack any form of ground control or sensor orientation information, initial approximations have been obtained from point correspondences alone using a parallel projection algorithm based on previous work in computer vision. The work of Huang and Lee (1989) has been expanded to solve for relative scales of images, and to orient more than three images at a time. When considering single images, the work of Kyle (2004) has provided much insight.
3) It has been shown that when reasonable approximations of the interior orientation parameters (IOP) of a camera are available, the developed algorithm is capable of producing an accurate result. This eliminates the necessity of a rigorous camera calibration. In the case where reasonable approximations are not available, the parallel projection can still be used, as it does not require approximations of the IOP.
4) The irregular manner in which surveillance images are captured does not pose a problem for the algorithm; no assumptions based on a systematic image collection (such as topographic photogrammetry) have been included. The bundle triangulation algorithm is the main tool for 3D reconstruction, and it is commonly used in close-range photogrammetry where random camera orientations are commonplace (Mikhail, Bethel, and McGlone, 2001).

The comprehensive algorithm's versatility has been demonstrated through tests with actual images and synthetic images. For these tests, the results from the comprehensive algorithm have been equal to, or better than, the results obtained using

standard photogrammetry methods. Thus, for the image sets that have been tested, the developed model meets the challenges posed by surveillance imagery.

8.2 Recommendations

There are some remaining questions and recommendations. First, there are some potential improvements that could be made to the comprehensive algorithm to increase the quality of its results. Also, it may be possible to simplify the comprehensive algorithm in several ways to make the computations more efficient. Furthermore, there are some additional test cases that may be instructive at indicating possible weaknesses or shortcomings of the model.

8.2.1 Increasing the quality of results

The comprehensive algorithm has employed the standard bundle triangulation algorithm (based on perspective projection), as well as the parallel projection bundle triangulation algorithm. However, in all of the above tests, these two versions of the bundle triangulation algorithm were used separately. Because of the flexibility of the bundle triangulation algorithm, it is possible to use the parallel projection equations within the algorithm for some images, and the perspective projection equations for other images. For instance, in the case where some images of a given target may have been captured at close range and have a wider angular field of view, and other images were captured at a further range and have a narrower angular field of view, it would likely be beneficial to use the perspective projection equations within the algorithm for the wide angle images and the parallel projection equations for the narrow angle images. This

implementation would be rather straightforward; one could use the parallel projection equation partial derivatives (3.16 – 3.17) for some images, and the perspective projection equation partial derivatives for other images within the bundle triangulation algorithm. This would likely give the best results for image sets with images that have varying fields of view.

The bundle triangulation algorithm has several other features that have not been fully investigated in the test above. For example, the bundle triangulation algorithm allows one to assign weights to each of the input measurements. Image coordinates collected from a higher resolution image (such as images captured with the CANON 1Ds) could be assigned a higher weight than images collected from an image with lower resolution (such as images captured with the CANON 20D). By applying appropriate weights to the measurements, the solution of the algorithm may be more accurate. The distance to the target may also be used as a method of applying weights to each measurement.

8.2.2 Simplifying the algorithm

The equations for the parallel projection model have been developed in terms of a scale *s*, and three rotation angles $(\omega, \varphi, \kappa)$. These four unknowns could be replaced by a quaternion (*qx*, *qy*, *qz*, *qw*), which describes a scaled rotation. There are several advantages to using a quaternion as opposed to rotation angles. The equations for parallel projection in terms of rotation angles involve sine and cosine functions, which are expensive computationally. The equations for parallel projection in terms of a quaternion are more simple and do not involve trigonometric functions. Also, a

quaternion is more robust than rotation angles. When using rotation angles, singularities exist when angles are equal to multiples of 90°. These singularities can be problematic. No such singularities exist for a quaternion.

The parallel projection initial approximation algorithm as developed above solves for the scale and rotation of the images in two separate steps. This requires essentially the same least-squares computation to be made for each pair of images twice (see equations 4.21 and 4.39). It may be possible to solve for the scale and rotation of each image at the same time—especially if the equations are expressed in terms of a quaternion instead of a scale and three rotation angles. This would eliminate the need to consider each pair of images twice, and only one least-squares calculation would have to be performed on each pair of images. This would lead to faster computation time and perhaps improved accuracy.

The parallel projection initial approximation algorithm leads to two possible solutions, referred to earlier as the (+) set and the (-) set, which are mirror images of each other. There is one additional input to the algorithm, the keypoint, to resolve this ambiguity. A more elegant way of resolving this uncertainty which does not require an additional input may be possible.

8.2.3 Additional tests

There are several additional tests that may reveal possible weaknesses of the algorithm. In particular, the algorithm has been tested using images which were taken from widely varying aspects. Some surveillance images sets may contain images which were taken from approximately the same aspect, with only a slight angular change

between images. It is known that the parallel projection can be less stable and less precise when two images are captured from essentially the same aspect. This is because the parallel projection assumes that all light rays for a given image are parallel, and uncertainty in the depth direction is large. The perspective projection is more stable in these situations. Hence, it would be instructive to test the algorithm against standard methods using images with very similar aspects.

As noted above, if images are captured with a sensor for which the approximate IOP are unknown, it is still possible to analyze these images with the parallel projection. However, if these images do not have a narrow angular field of view, the parallel projection will not provide a very precise solution. In these cases it may be better to analyze the images from an unknown sensor with the DLT, which would be capable of solving for the approximate IOP when the field of view is not too narrow. Thus, further testing with images from an unknown sensor would be informative.

REFERENCES

Abdel-Aziz, Y. I., and H. M. Karara. 1971. Direct linear transform from comparator coordinates into object-space coordinates. Proceedings of the ASP Symposium on Close-Range Photogrammetry, Falls Church, VA p. 1-18.

Fraser, C.S., and T. Yamakawa. 2004. Insights into the affine model for high-resolution satellite sensor orientation. ISPRS Journal of Photogrammetry and Remote Sensing 58: 275-288.

Huang, T.S., and C. H. Lee. 1989. Motion and structure from orthographic projections. IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (5): 536-540.

Grodecki, J., and G. Dial. 2003. Block adjustment of high-resolution satellite images described by rational functions. Photogrammetric Engineering and Remote Sensing 69 (1): 59-70.

Gruen, A., and Z. Li. 2003. 3D processing of high-resolution satellite images. Asian Conference on Remote Sensing, November 3-7, 2003, Busan, Korea. CD-ROM.

Kyle, S. 2004. Using parallel projection mathematics to orient an object relative to a single image. The Photogrammetric Record 19 (105): 38-50.

Luhmann, T., S. Robson, S. Kyle, and I. Harley. 2006. Close range photogrammetry principles, methods, and applications. Whittles Publishing, Dunbeath, Scotland, UK. 509 p.

McGlone, J. C. 1989. Analytic data-reduction schemes in non-topographic photogrammetry, p. 37-57. In Non-Topographic Photogrammetry, Second Edition. American Society for Photogrammetry and Remote Sensing, Falls Church, VI.

Mikhail, E. M., J. S. Bethel, and J. C. McGlone. 2001. Introduction to Modern Photogrammetry. John Wiley & Sons, New York, NY. 480 p.

Morgan, M. 2004. Epipolar resampling of linear array scanner scenes. Ph D Dissertation. University of Calgary UCGE Reports. 174 p.

Ono, T., and S. Hattori. 2002. Fundamental principles of image orientation using orthogonal projection model. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 34 (3B): 194-199.

Pan, H-P., M. J. Brooks, and G. N. Newsam. 1995. Image resituation: initial theory. Proceedings of SPIE: Videometrics IV 2598: 162-173.

Pan, H-P., D. Q. Huynh, and G. K. Hamlyn. 1995. Two-image resituation: practical algorithm. Proceedings of SPIE: Videometrics IV 2598: 174-190.

Wolf, P., and B. Dewitt. 2000. Elements of Photogrammetry, 3rd Edition. McGraw Hill, Boston, MA. 608 p.

Zhang, J., and X. Zhang. 2002. Strict geometric model based on affine transformation for remote sensing image with high resolution. International Archives of Photogrammetry and Remote Sensing 34 (B3): 309-312.

APPENDICES

A.1 Rotation Matrices

Rotation Matrices play an important role in photogrammetry. Rotation matrices are a very convenient and simple way of expressing rotations in 3D space. When a set of XYZ coordinates are multiplied by a 3 x 3 rotation matrix, the resulting XYZ coordinates are simply a rigid rotation of the original coordinates about the origin of 3D space. Here the properties of rotation matrices are discussed, the equations for a rotation matrix in terms of three ordered rotation angles $(\omega, \varphi, \kappa)$ about the X- Y- and Z- axes are presented (this is a standard convention in photogrammetry), the equations for a rotation matrix in terms of three ordered rotation angles $(\theta, \psi, \varepsilon)$ are presented (as discussed in Chapter 7) and a method to solve for the three rotation angles from a known rotation matrix is outlined. A more comprehensive review of rotation matrices and the proofs of some of the equations given here are found in the appendix of (Morgan, 2004).

A.1.1 Properties of rotation matrices

A rotation matrix (as used in this thesis) is a 3 x 3 orthonormal matrix. An orthonormal matrix is a matrix in which all rows (columns) have a norm of 1, and all rows (columns) are mutually orthogonal to the other two rows (columns). The properties of rotation matrices will be very helpful in some of the derivations. For convenience these properties will be outlined here.

For the equations that follow, let $\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$ be a rotation matrix; all rows (columns) have a norm of 1:

$$\begin{aligned} m_{11}^2 + m_{12}^2 + m_{13}^2 = 1 \qquad m_{21}^2 + m_{22}^2 + m_{23}^2 = 1 \qquad m_{31}^2 + m_{32}^2 + m_{33}^2 = 1 \\ m_{11}^2 + m_{21}^2 + m_{31}^2 = 1 \qquad m_{12}^2 + m_{22}^2 + m_{32}^2 = 1 \qquad m_{13}^2 + m_{23}^2 + m_{33}^2 = 1 \end{aligned} \tag{A.1}$$

Also, each row (column) is orthogonal to the other rows (columns). This means that the dot product of any two rows (columns) is zero:

$$\begin{aligned} &\begin{bmatrix} m_{11} & m_{12} & m_{13} \end{bmatrix} \bullet \begin{bmatrix} m_{21} & m_{22} & m_{23} \end{bmatrix} = 0 \\ &\begin{bmatrix} m_{11} & m_{12} & m_{13} \end{bmatrix} \bullet \begin{bmatrix} m_{31} & m_{32} & m_{33} \end{bmatrix} = 0 \\ &\begin{bmatrix} m_{21} & m_{22} & m_{23} \end{bmatrix} \bullet \begin{bmatrix} m_{31} & m_{32} & m_{33} \end{bmatrix} = 0 \\ &\begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \bullet \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} = 0, \quad \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \bullet \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = 0, \quad \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} \bullet \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = 0 \end{aligned} \tag{A.2}$$

Also due to orthogonality, the cross product of any two rows (columns) is equal to the remaining row (column) with a positive or negative sign:

$$\begin{aligned} &\begin{bmatrix} m_{11} & m_{12} & m_{13} \end{bmatrix} \times \begin{bmatrix} m_{21} & m_{22} & m_{23} \end{bmatrix} = \begin{bmatrix} m_{31} & m_{32} & m_{33} \end{bmatrix} \\ &\begin{bmatrix} m_{11} & m_{12} & m_{13} \end{bmatrix} \times \begin{bmatrix} m_{31} & m_{32} & m_{33} \end{bmatrix} = -\begin{bmatrix} m_{21} & m_{22} & m_{23} \end{bmatrix} \\ &\begin{bmatrix} m_{21} & m_{22} & m_{23} \end{bmatrix} \times \begin{bmatrix} m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \end{bmatrix} \\ &\begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \times \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} = \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix}, \quad \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \times \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = -\begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix}, \quad \begin{bmatrix} m_{12} \\ m_{22} \\ m_{32} \end{bmatrix} \times \begin{bmatrix} m_{13} \\ m_{23} \\ m_{33} \end{bmatrix} = \begin{bmatrix} m_{11} \\ m_{21} \\ m_{31} \end{bmatrix} \end{aligned} \tag{A.3}$$

The inverse of a rotation matrix is equal to the transpose of the rotation matrix:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}^{-1} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}^{T} = \begin{bmatrix} m_{11} & m_{21} & m_{31} \\ m_{12} & m_{22} & m_{32} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} \tag{A.4}$$

Finally, the product of any two rotation matrices is a rotation matrix which describes both of the rotations performed in sequence.

A.1.2 Solving for a rotation matrix from three angles $(\omega, \varphi, \kappa)$

As mentioned earlier, a rotation matrix can be fully determined by three rotation angles (Wolf and DeWitt, 2000). The angles that will be considered will be three ordered rotations about the X-axis, Y-axis, and Z-axis of 3D space respectively. These three angles will be denoted by $(\omega, \varphi, \kappa)$. The angles will be considered to be positive when they are in the counter-clockwise direction when viewed from the positive end of their respective axes. Using these three rotation angles, the rotation matrix becomes:

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} \cos\kappa & \sin\kappa & 0 \\ -\sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\varphi & 0 & -\sin\varphi \\ 0 & 1 & 0 \\ \sin\varphi & 0 & \cos\varphi \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & \sin\omega \\ 0 & -\sin\omega & \cos\omega \end{bmatrix} \tag{A.5}$$

And the matrix terms are given by:

$$\begin{aligned} m_{11} &= \cos(\varphi)\cos(\kappa) \\ m_{12} &= \sin(\omega)\sin(\varphi)\cos(\kappa) + \cos(\omega)\sin(\kappa) \\ m_{13} &= -\cos(\omega)\sin(\varphi)\cos(\kappa) + \sin(\omega)\sin(\kappa) \\ m_{21} &= -\cos(\varphi)\sin(\kappa) \\ m_{22} &= -\sin(\omega)\sin(\varphi)\sin(\kappa) + \cos(\omega)\cos(\kappa) \\ m_{23} &= \cos(\omega)\sin(\varphi)\sin(\kappa) + \sin(\omega)\cos(\kappa) \\ m_{31} &= \sin(\varphi) \\ m_{32} &= -\sin(\omega)\cos(\varphi) \\ m_{33} &= \cos(\omega)\cos(\varphi) \end{aligned} \tag{A.6}$$

Thus, given the three rotation angles $(\omega, \varphi, \kappa)$, the terms of the rotation matrix can be found. Because trigonometric functions are periodic, some constraints can be placed on these angles. All possible rotations can still be defined if ω is held in the interval $[-\pi, \pi]$, φ is held in the interval $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$, and κ is held in the interval $[-\pi, \pi]$.

A.1.3 Solving for the three angles $(\omega, \varphi, \kappa)$ from a rotation matrix

It may also be necessary to determine the three rotation angles from a given matrix. To do this, first consider the equation $m_{31} = \sin(\varphi)$ from (A.6). From this equation φ is solved for:

$$\varphi = \sin^{-1}(m_{31}) \tag{A.7}$$

Because it was assumed that φ is in the interval $\left[\frac{-\pi}{2}, \frac{\pi}{2}\right]$, this equation has only one solution. Now, by considering equations (A.6) the following are derived:

$$\frac{-m_{32}}{m_{33}} = \frac{\sin(\omega)\cos(\varphi)}{\cos(\omega)\cos(\varphi)} \qquad \frac{-m_{21}}{m_{11}} = \frac{\cos(\varphi)\sin(\kappa)}{\cos(\varphi)\cos(\kappa)} \tag{A.8}$$

Now, if $\cos(\varphi) \neq 0$ (i.e. $\varphi \neq \frac{\pi}{2}, \varphi \neq \frac{-\pi}{2}$), then equations (A.8) reduce to:

$$\frac{-m_{32}}{m_{33}} = \frac{\sin(\omega)}{\cos(\omega)} \qquad \frac{-m_{21}}{m_{11}} = \frac{\sin(\kappa)}{\cos(\kappa)} \tag{A.9}$$

And ω and κ can be found by:

$$\omega = \text{atan2}(-m_{32}, m_{33}) \qquad \kappa = \text{atan2}(-m_{21}, m_{11}) \tag{A.10}$$

where "atan2" denotes the full circle inverse tangent function. In the event that $\cos(\varphi) = 0$ (i.e. $\varphi = \frac{\pi}{2}, \varphi = -\frac{\pi}{2}$) the rotations about the X- and Z- axes can be reduced to one rotation about the X-axis since the rotation of $\pm\frac{\pi}{2}$ about the Y-axis rotates the X-axis into the position of the Z-axis (positive or negative). Hence, it can be assumed $\kappa = 0$ and the matrix terms reduce to:

$$\begin{aligned} m_{12} &= m_{23} = \sin\omega \\ m_{22} &= -m_{13} = \cos\omega \end{aligned} \tag{A.11}$$

And:

$$\omega = \text{atan2}(m_{12}, m_{22}) \tag{A.12}$$

This case is better handled using a quaternion, but will likely not be encountered in practice with the algorithm.

A.1.4 Solving for the rotation matrix terms from three rotation angles $(\theta, \psi, \varepsilon)$

The following formulas give the rotation matrix terms as functions of the three rotation angles $(\theta, \psi, \varepsilon)$ discussed in Chapter 7. Note that these are similar to equations (A.6) which define the rotation matrix terms from $(\omega, \varphi, \kappa)$.

$$\begin{aligned} m_{11} &= -\sin(\theta)\sin(\psi)\sin(\varepsilon) - \cos(\theta)\cos(\varepsilon) \\ m_{12} &= -\cos(\theta)\sin(\psi)\sin(\varepsilon) + \sin(\theta)\cos(\varepsilon) \\ m_{13} &= \cos(\psi)\sin(\varepsilon) \\ m_{21} &= -\sin(\theta)\sin(\psi)\cos(\varepsilon) + \cos(\theta)\sin(\varepsilon) \\ m_{22} &= -\cos(\theta)\sin(\psi)\cos(\varepsilon) - \sin(\theta)\sin(\varepsilon) \\ m_{23} &= \cos(\psi)\cos(\varepsilon) \\ m_{31} &= \sin(\theta)\cos(\psi) \\ m_{32} &= \cos(\theta)\cos(\psi) \\ m_{33} &= \sin(\psi) \end{aligned} \tag{A.13}$$

A.2 Bundle Triangulation Algorithm

Two special cases of the bundle triangulation algorithm will be given here. The first is a general form of the bundle triangulation algorithm used to solve for the EOP of several images along with the 3D coordinates of several target points, the second is a

simpler form used to solve for the EOP of a single image. Both forms assume some limitations which are common in surveillance imagery and are not a complete representative of the many different structures that the bundle triangulation can follow. For a more comprehensive discussion of the bundle triangulation method, see (Luhmann, et al., 2006; Wolf and DeWitt, 2000; Mikhail, Bethel, and McGlone, 2001).

A.2.1 General bundle triangulation algorithm

The first case is a general form of the bundle triangulation algorithm used to solve for the EOP of several images and the 3D coordinates of several target points. It will be assumed that there are *I* images (image *A*, image *B*,…, image *I*) with unknown EOP:

$$(\omega_A, \varphi_A, \kappa_A, Xc_A, Yc_A, Zc_A),\ (\omega_B, \varphi_B, \kappa_B, Xc_B, Yc_B, Zc_B),\ldots\ (\omega_I, \varphi_I, \kappa_I, Xc_I, Yc_I, Zc_I)$$

It will be assumed further that there are *n* target points with unknown 3D coordinates: $(X_1, Y_1, Z_1), (X_2, Y_2, Z_2), \ldots, (X_n, Y_n, Z_n)$

It is assumed that these points appear in the *I* images. It will also be assumed that the set of point correspondences of the *n* target points in the *I* images have been collected and are denoted as:

$$\begin{array}{cccc} (x_{A,1}, y_{A,1}) & (x_{B,1}, y_{B,1}) & \ldots & (x_{I,1}, y_{I,1}) \\ (x_{A,2}, y_{A,2}) & (x_{B,2}, y_{B,2}) & \ldots & (x_{I,2}, y_{I,2}) \\ \ldots & \ldots & \ldots & \ldots \\ (x_{A,n}, y_{A,n}) & (x_{B,n}, y_{B,n}) & \ldots & (x_{I,n}, y_{I,n}) \end{array} \qquad \text{(A.14)}$$

All *n* target points do not need to be visible in all *I* images—some of the above coordinates may be missing. It is necessary for each point to be visible in at least two images. It will be assumed that for each image the approximate principal distance *f* and the approximate coordinates of the principal point (*xo*, *yo*) are known, although with

some surveillance images this may not be the case. It will be assumed further that no survey data in 3D space or information regarding the geometric properties of the target object is available. For each image there are 6 unknowns $(\omega, \varphi, \kappa, Xc, Yc, Zc)$ and for each 3D point there are three unknowns (X, Y, Z).

For the bundle triangulation algorithm it is necessary to have initial approximations for each of the unknowns. It will be assumed that initial approximations of the unknowns have been obtained.

Hence, the set of inputs will be: the set of image coordinates (point correspondences), the approximate IOP *f*, *xo*, and *yo* for each image, and the initial approximations of the unknowns (the EOP and 3D coordinates).

These assumptions are not necessary in all forms of the bundle triangulation method, but for the general form outlined here these assumptions will be made.

Initially, there is no absolute orientation of 3D space because it has not been defined by the inputs. So defining the absolute orientation of 3D space is the first step. To do this, a rotation, an origin, and a scale for the 3D coordinate system can arbitrarily be defined. There are several ways to define the orientation; a standard procedure is to "fix" the 6 unknowns for one of the images. By "fixing" the three rotation angles of one of the images the orientation of 3D space is defined, by "fixing" the perspective center of one of the images the location of the origin of 3D space is defined. Without loss of generality, the EOP for image *A* will be fixed. They can be set to the values of the initial approximations:

$$\begin{aligned} &\omega_A = (\omega_A)_0 \qquad \varphi_A = (\varphi_A)_0 \qquad \kappa_A = (\kappa_A)_0 \\ &Xc_A = (Xc_A)_0 \qquad Yc_A = (Yc_A)_0 \qquad Zc_A = (Zc_A)_0 \end{aligned} \tag{A.15}$$

By fixing the *X*-coordinate of the perspective center for one of the other images the scale of 3D space is defined. Without loss of generality the *X*-coordinate of the perspective center of image *B* will be fixed at the value of the initial approximation:

$$Xc_B = \left(Xc_B\right)_0 \tag{A.16}$$

By "fixing" these seven unknowns, the orientation of 3D space has been defined and the number of unknowns has been reduced to: zero for image *A*, five for image *B*, six for each of the remaining images, and three for each target point. Alternate definitions of 3D space can be used, which will result in a slightly different set of unknowns.

Next, a system of linear equations (A.17) is set up (Wolf and DeWitt, 2000). These linear equations are created by taking a first-order Taylor series approximation of the collinearity equations with respect to each of the unknowns. This equation is for point *p* on image *K* (items with a 0 subscript are initial approximations):

$$\begin{aligned}
x_{K,p} &\approx \left(x'_{K,p}\right)_0 + \left(\frac{\partial x_{K,p}}{\partial \omega_K}\right)_0 d\omega_K + \left(\frac{\partial x_{K,p}}{\partial \varphi_K}\right)_0 d\varphi_K + \left(\frac{\partial x_{K,p}}{\partial \kappa_K}\right)_0 d\kappa_K + \left(\frac{\partial x_{K,p}}{\partial Xc_K}\right)_0 dXc_K + \ldots \\
&\ldots + \left(\frac{\partial x_{K,p}}{\partial Yc_K}\right)_0 dYc_K + \left(\frac{\partial x_{K,p}}{\partial Zc_K}\right)_0 dZc_K + \left(\frac{\partial x_{K,p}}{\partial X_p}\right)_0 dX_p + \left(\frac{\partial x_{K,p}}{\partial Y_p}\right)_0 dY_p + \left(\frac{\partial x_{K,p}}{\partial Z_p}\right)_0 dZ_p \\
y_{K,p} &\approx \left(y'_{K,p}\right)_0 + \left(\frac{\partial y_{K,p}}{\partial \omega_K}\right)_0 d\omega_K + \left(\frac{\partial y_{K,p}}{\partial \varphi_K}\right)_0 d\varphi_K + \left(\frac{\partial y_{K,p}}{\partial \kappa_K}\right)_0 d\kappa_K + \left(\frac{\partial y_{K,p}}{\partial Xc_K}\right)_0 dXc_K + \ldots \\
&\ldots + \left(\frac{\partial y_{K,p}}{\partial Yc_K}\right)_0 dYc_K + \left(\frac{\partial y_{K,p}}{\partial Zc_K}\right)_0 dZc_K + \left(\frac{\partial y_{K,p}}{\partial X_p}\right)_0 dX_p + \left(\frac{\partial y_{K,p}}{\partial Y_p}\right)_0 dY_p + \left(\frac{\partial y_{K,p}}{\partial Z_p}\right)_0 dZ_p
\end{aligned} \tag{A.17}$$

These equations are linear in the unknowns $\left(d\omega_K, d\varphi_K, d\kappa_K, dXc_K, dYc_K, dZc_K\right)$ and $\left(dX_p, dY_p, dZ_p\right)$. In equation (A.17), $\left(x'_{K,p}\right)_0$ denotes the back-projection of point *p* into image *K* based on the initial approximations; this is given in equation (A.29). If these

two equations are written for each of the points which appear in each of the images, then a linear system of equations with the following unknowns is obtained:

$$
\begin{aligned}
&(d\omega_B, d\varphi_B, d\kappa_B, dYc_B, dZc_B), (d\omega_C, d\varphi_C, d\kappa_C, dXc_C, dYc_C, dZc_C), \ldots \\
&\ldots, (d\omega_I, d\varphi_I, d\kappa_I, dXc_I, dYc_I, dZc_I) \\
&(dX_1, dY_1, dZ_1), (dX_2, dY_2, dZ_2), \ldots, (dX_n, dY_n, dZ_n)
\end{aligned}
\tag{A.18}
$$

Note that missing from the unknowns (A.18) are:

$$
\begin{aligned}
&d\omega_A = 0 \qquad d\varphi_A = 0 \qquad dd\kappa_A = 0 \\
&dXc_A = 0 \qquad dYc_A = 0 \qquad dZc_A = 0 \qquad dXc_B = 0
\end{aligned}
\tag{A.19}
$$

This is due to the fact that these variables were fixed in order to define an absolute 3D coordinate system. Hence, these variables are not included in the set of unknowns.

So, by considering the equations for each point in each image, a linear system can be written in matrix-form $L = AX$ with L, A, and X as follows:

$$
L = \begin{bmatrix}
x_{A,1} - (x'_{A,1})_0 \\
y_{A,1} - (y'_{A,1})_0 \\
\ldots \\
x_{A,n} - (x'_{A,n})_0 \\
y_{A,n} - (y'_{A,n})_0 \\
x_{B,1} - (x'_{B,1})_0 \\
y_{B,1} - (y'_{B,1})_0 \\
\ldots \\
x_{B,n} - (x'_{B,n})_0 \\
y_{B,n} - (y'_{B,n})_0 \\
\ldots \\
x_{I,1} - (x'_{I,1})_0 \\
y_{I,1} - (y'_{I,1})_0 \\
\ldots \\
x_{I,n} - (x'_{I,n})_0 \\
y_{I,n} - (y'_{I,n})_0
\end{bmatrix}
\begin{matrix}
\left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\} n \text{ points in image } A \\
\left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\} n \text{ points in image } B \\
\\
\left.\vphantom{\begin{matrix}a\\a\\a\\a\\a\end{matrix}}\right\} n \text{ points in image } I
\end{matrix}
\tag{A.20}
$$

$$A = \begin{bmatrix} 0_{2n\ x\ 5} & 0_{2n\ x\ 6} & \dots & 0_{2n\ x\ 6} & A_{XYZ} \\ B & 0_{2n\ x\ 6} & \dots & 0_{2n\ x\ 6} & B_{XYZ} \\ 0_{2n\ x\ 5} & C & \dots & 0_{2n\ x\ 6} & C_{XYZ} \\ \dots & \dots & \dots & \dots & \dots \\ 0_{2n\ x\ 5} & 0_{2n\ x\ 6} & \dots & I & I_{XYZ} \end{bmatrix} \begin{matrix} \}n \text{ points in image } A \\ \}n \text{ points in image } B \\ \}n \text{ points in image } C \\ \\ \}n \text{ points in image } I \end{matrix} \tag{A.21}$$

Where:

$$B = \begin{bmatrix} \frac{\partial x_{B,1}}{\partial \omega_B} & \frac{\partial x_{B,1}}{\partial \varphi_B} & \frac{\partial x_{B,1}}{\partial \kappa_B} & -\frac{\partial x_{B,1}}{\partial Yc_B} & -\frac{\partial x_{B,1}}{\partial Zc_B} \\ \frac{\partial y_{B,1}}{\partial \omega_B} & \frac{\partial y_{B,1}}{\partial \varphi_B} & \frac{\partial y_{B,1}}{\partial \kappa_B} & -\frac{\partial y_{B,1}}{\partial Yc_B} & -\frac{\partial y_{B,1}}{\partial Zc_B} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial x_{B,n}}{\partial \omega_B} & \frac{\partial x_{B,n}}{\partial \varphi_B} & \frac{\partial x_{B,n}}{\partial \kappa_B} & -\frac{\partial x_{B,n}}{\partial Yc_B} & -\frac{\partial x_{B,n}}{\partial Zc_B} \\ \frac{\partial y_{B,n}}{\partial \omega_B} & \frac{\partial y_{B,n}}{\partial \varphi_B} & \frac{\partial y_{B,n}}{\partial \kappa_B} & -\frac{\partial y_{B,n}}{\partial Yc_B} & -\frac{\partial y_{B,n}}{\partial Zc_B} \end{bmatrix} \tag{A.22}$$

$$C = \begin{bmatrix} \frac{\partial x_{C,1}}{\partial \omega_C} & \frac{\partial x_{C,1}}{\partial \varphi_C} & \frac{\partial x_{C,1}}{\partial \kappa_C} & -\frac{\partial x_{C,1}}{\partial Xc_C} & -\frac{\partial x_{C,1}}{\partial Yc_C} & -\frac{\partial x_{C,1}}{\partial Zc_C} \\ \frac{\partial y_{C,1}}{\partial \omega_C} & \frac{\partial y_{C,1}}{\partial \varphi_C} & \frac{\partial y_{C,1}}{\partial \kappa_C} & -\frac{\partial y_{C,1}}{\partial Xc_C} & -\frac{\partial y_{C,1}}{\partial Yc_C} & -\frac{\partial y_{C,1}}{\partial Zc_C} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial x_{C,n}}{\partial \omega_C} & \frac{\partial x_{C,n}}{\partial \varphi_C} & \frac{\partial x_{C,n}}{\partial \kappa_C} & -\frac{\partial x_{C,n}}{\partial Xc_C} & -\frac{\partial x_{C,n}}{\partial Yc_C} & -\frac{\partial x_{C,n}}{\partial Zc_C} \\ \frac{\partial y_{C,n}}{\partial \omega_C} & \frac{\partial y_{C,n}}{\partial \varphi_C} & \frac{\partial y_{C,n}}{\partial \kappa_C} & -\frac{\partial y_{C,n}}{\partial Xc_C} & -\frac{\partial y_{C,n}}{\partial Yc_C} & -\frac{\partial y_{C,n}}{\partial Zc_C} \end{bmatrix} \tag{A.23}$$

$$I = \begin{bmatrix} \frac{\partial x_{I,1}}{\partial \omega_I} & \frac{\partial x_{I,1}}{\partial \varphi_I} & \frac{\partial x_{I,1}}{\partial \kappa_I} & -\frac{\partial x_{I,1}}{\partial Xc_I} & -\frac{\partial x_{I,1}}{\partial Yc_I} & -\frac{\partial x_{I,1}}{\partial Zc_I} \\ \frac{\partial y_{I,1}}{\partial \omega_I} & \frac{\partial y_{I,1}}{\partial \varphi_I} & \frac{\partial y_{I,1}}{\partial \kappa_I} & -\frac{\partial y_{I,1}}{\partial Xc_I} & -\frac{\partial y_{I,1}}{\partial Yc_I} & -\frac{\partial y_{I,1}}{\partial Zc_I} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \frac{\partial x_{I,n}}{\partial \omega_I} & \frac{\partial x_{I,n}}{\partial \varphi_I} & \frac{\partial x_{I,n}}{\partial \kappa_I} & -\frac{\partial x_{I,n}}{\partial Xc_I} & -\frac{\partial x_{I,n}}{\partial Yc_I} & -\frac{\partial x_{I,n}}{\partial Zc_I} \\ \frac{\partial y_{I,n}}{\partial \omega_I} & \frac{\partial y_{I,n}}{\partial \varphi_I} & \frac{\partial y_{I,n}}{\partial \kappa_I} & -\frac{\partial y_{I,n}}{\partial Xc_I} & -\frac{\partial y_{I,n}}{\partial Yc_I} & -\frac{\partial y_{I,n}}{\partial Zc_I} \end{bmatrix} \tag{A.24}$$

$$A_{XYZ} = \begin{bmatrix} \frac{\partial x_{A,1}}{\partial X_1} & \frac{\partial x_{A,1}}{\partial Y_1} & \frac{\partial x_{A,1}}{\partial Z_1} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ \frac{\partial y_{A,1}}{\partial X_1} & \frac{\partial y_{A,1}}{\partial Y_1} & \frac{\partial y_{A,1}}{\partial Z_1} & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial x_{A,2}}{\partial X_2} & \frac{\partial x_{A,2}}{\partial Y_2} & \frac{\partial x_{A,2}}{\partial Z_2} & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\partial y_{A,2}}{\partial X_2} & \frac{\partial y_{A,2}}{\partial Y_2} & \frac{\partial y_{A,2}}{\partial Z_2} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{\partial x_{A,n}}{\partial X_n} & \frac{\partial x_{A,n}}{\partial Y_n} & \frac{\partial x_{A,n}}{\partial Z_n} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & \frac{\partial y_{A,n}}{\partial X_n} & \frac{\partial y_{A,n}}{\partial Y_n} & \frac{\partial y_{A,n}}{\partial Z_n} \end{bmatrix} \tag{A.25}$$

(B_{XYZ}, C_{XYZ},…, I_{XYZ} have the same format as A_{XYZ} and correspond to image *B*, image *C*,…, image *I*.)

$$0_{2n\ x\ 5} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.26}$$

$$0_{2n\ x\ 6} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{A.27}$$

$$X = \begin{bmatrix} d\omega_B \\ d\varphi_B \\ d\kappa_B \\ dYc_B \\ dZc_B \\ d\omega_C \\ d\varphi_C \\ d\kappa_C \\ dXc_C \\ dYc_C \\ dZc_C \\ \cdots \\ d\omega_I \\ d\varphi_I \\ d\kappa_I \\ dXc_I \\ dYc_I \\ dZc_I \\ dX_1 \\ dY_1 \\ dZ_1 \\ \cdots \\ dX_n \\ dY_n \\ dZ_n \end{bmatrix} \begin{matrix} \\ \end{matrix} \begin{matrix} \left.\begin{matrix} \\ \\ \\ \\ \end{matrix}\right\}\text{image } B \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\}\text{image } C \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \end{matrix}\right\}\text{image } I \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\}\text{point 1} \\ \left.\begin{matrix} \\ \\ \end{matrix}\right\}\text{point } n \end{matrix} \tag{A.28}$$

In equations (A.20 – A.28) note that the row index of L is equal to the row index of A (each row corresponds to the x- or y-coordinate of one point on one image) and the column index of A is equal to the row index of X (each column of A corresponds to one unknown in X). Now, by beginning with the initial approximations for the unknown EOP of the images and 3D points, the matrix system above can be set up by evaluating at the initial approximations. Then the least-squares solution for the unknowns (A.28) is found

by $X = \left(A^T A\right)^{-1}\left(A^T L\right)$. Then the *X*-vector of unknowns is added to the initial approximations to produce improved approximations. The improved approximations become the new initial approximations and this process is repeated until the magnitudes of the corrections *X* are below a desired threshold.

The primary outputs of the bundle triangulation algorithm are the set of calculated 3D coordinates and the set of calculated EOP. These are given by the vector *X* of unknowns. There are three other important outputs of the bundle triangulation algorithm: back-projected image coordinates, residuals, and 1-sigma errors.

Let the measured image coordinates of point *p* be denoted by $\left(x_p, y_p\right)$ and let the calculated 3D coordinates of the point *p* be denoted by $\left(X_p', Y_p', Z_p'\right)$. These coordinates which have been solved for are not exactly equal to the actual 3D coordinates of the point *p* denoted by $\left(X_p, Y_p, Z_p\right)$. The bundle triangulation algorithm also calculates values for the EOP denoted by $\left(\omega, \varphi, \kappa, Xc, Yc, Zc\right)$ which are not exact.

To get the back-projected image coordinates, it is assumed that the calculated 3D points are exact, that the calculated EOP are exact, and that the perspective projection is an exact model. Then the back-projected image coordinates $\left(x_p', y_p'\right)$ are calculated for the perspective projection using the collinearity equations (like equation 1.1):

$$\begin{aligned} x_p' &= xo - f\left(\frac{m_{11}\left(X_p'-Xc\right)+m_{12}\left(Y_p'-Yc\right)+m_{13}\left(Z_p'-Zc\right)}{m_{31}\left(X_p'-Xc\right)+m_{32}\left(Y_p'-Yc\right)+m_{33}\left(Z_p'-Zc\right)}\right) \\ y_p' &= yo - f\left(\frac{m_{21}\left(X_p'-Xc\right)+m_{22}\left(Y_p'-Yc\right)+m_{23}\left(Z_p'-Zc\right)}{m_{31}\left(X_p'-Xc\right)+m_{32}\left(Y_p'-Yc\right)+m_{33}\left(Z_p'-Zc\right)}\right) \end{aligned} \tag{A.29}$$

Recall that the rotation matrix *m* terms in this equation are derived from the three rotation angles. Recall also that it is assumed that the approximate principal distance *f*, and the principal point coordinates (*xo*, *yo*) are known.

These back-projected image coordinates are just the calculated 3D coordinates back-projected onto the images according to the calculated EOP. These back-projected image coordinates can be compared with the measured image coordinates as a measure of the precision of the solution. The residuals (r_x, r_y) are defined as the difference of the back-projected image coordinates (x_p', y_p') and the measured image coordinates (x_p, y_p):

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} x_p' - x_p \\ y_p' - y_p \end{bmatrix} \tag{A.30}$$

The magnitude of these residuals provides a metric for determining the precision of each calculation. They may also be used to detect blunders in the image coordinates that were measured from the images.

It also may also be helpful to consider the total residual *r* for each point:

$$r = \sqrt{r_x^2 + r_y^2} \tag{A.31}$$

While residuals are very helpful, they do not directly provide a measure of precision in 3D space. Since the primary concern is an accurate model of the target in 3D space, 1-sigma errors are used to determine the precision of the calculations in 3D space.

The bundle triangulation equations (A.20 – A.28) are of the form $L = AX$ and the least-squares solution is $X = (A^T A)^{-1}(A^T L)$. The residuals are therefore given in vector form as:

$$R = AX - L \tag{A.32}$$

The standard deviation of unit weight, σ_0, is then calculated as:

$$\sigma_0 = \sqrt{\frac{R^T R}{n}} \tag{A.33}$$

Where *n* is the degrees of freedom and is given by:

n = (the number of equations) – (the number of unknowns) (A.34)

Then, the 1-sigma error for each of the unknowns is given by:

$$\sigma_x = \sigma_0 \sqrt{Q_{xx}} \tag{A.35}$$

Where Q_{xx} represents the diagonal elements of the matrix:

$$Q = \left(A^T A\right)^{-1} \tag{A.36}$$

Note that the dimensions of *Q* will correspond to the number of unknowns. So, the 1-simga error for a given point will consist of three components $\sigma_X, \sigma_Y, \sigma_Z$ corresponding to the three unknowns (*X*, *Y*, *Z*) for each point. The total 1-sigma error of a point may also be calculated by:

$$\sigma = \sqrt{\sigma_X^2 + \sigma_y^2 + \sigma_Z^2} \tag{A.37}$$

The 1-simga errors derived above are based on the assumption that the residuals in the calculations follow a standard normal distribution. The 1-sigma error values represent one standard deviation of the standard normal distribution.

A.2.2 Single image resection

The second form of the bundle triangulation that will be considered is a method for determining the EOP of a single image with respect to a set of *XYZ* coordinates. This is often referred to as resection. This form of the algorithm is very similar to the more

general form, except that only the EOP $(\omega, \varphi, \kappa, Xc, Yc, Zc)$ of the single image are treated as unknowns.

As before, it is assumed that there are *p* points for which the 3D coordinates (*X*, *Y*, *Z*) are known (or approximately known). It is also assumed that these *p* points have been located in the image and the image coordinates (*x*, *y*) have been recorded. It is also necessary to have an initial approximation of the 6 unknown EOP.

The procedure follows as outlined above. The *L* vector contains image coordinates for the image, the *A* matrix has only 6 columns corresponding to the six unknowns, and the *X* vector has only the six unknowns. The process is iterative; at each step updates to the initial approximations are calculated by $X = (A^T A)^{-1}(A^T L)$. The iteration continues until the updates are negligible.

The output of this algorithm is the EOP of the image with respect to the 3D points.

A.3 Perspective Projection 3D Reconstruction

A simple method for performing 3D reconstruction by the perspective projection is given here. It is useful when the IOP and the EOP are known, so the full bundle triangulation algorithm is no necessary.

A.3.1 Alternative forms of the collinearity equations

Equation (1.1) is the standard form of the collinearity equations. There are several other forms of the collinearity equations which are very helpful. They are obtained by rearranging terms in (1.1). One form of the collinearity equations which is

helpful is a matrix-form representation. This form comes from adding a third equation to the two equations of (1.1) and rewriting them in matrix form:

$$\begin{bmatrix} x - xo \\ y - yo \\ -f \end{bmatrix} = \frac{-f}{m_{31}(X - Xc) + m_{32}(Y - Yc) + m_{33}(Z - Zc)} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X - Xc \\ Y - Yc \\ Z - Zc \end{bmatrix} \quad \text{(A.38)}$$

Using this matrix-form representation, a scale term $\lambda(X, Y, Z)$ is define:

$$\lambda(X, Y, Z) = \frac{-f}{m_{31}(X - Xc) + m_{32}(Y - Yc) + m_{33}(Z - Zc)} \quad \text{(A.39)}$$

And by substituting this scale term (A.39) into equation (A.38) the collinearity equations become (Ono and Hattori, 2002; Morgan, 2004):

$$\begin{bmatrix} x - xo \\ y - yo \\ -f \end{bmatrix} = \lambda(X, Y, Z) \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \begin{bmatrix} X - Xc \\ Y - Yc \\ Z - Zc \end{bmatrix} \quad \text{(A.40)}$$

This form represents the shifted image coordinates $\begin{bmatrix} x - xo \\ y - yo \\ -f \end{bmatrix}$ as a scaled rotation of the shifted 3D coordinates $\begin{bmatrix} X - Xc \\ Y - Yc \\ Z - Zc \end{bmatrix}$ where the rotation is $\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$ and the scale $\lambda(X, Y, Z)$ is a function of the 3D coordinates (*X*, *Y*, *Z*). This form of the collinearity equations will be used to develop several relationships between the parallel projection and perspective projection in subsequent chapters.

Another form (A.41) represents a linear combination of (*X*, *Y*, *Z*) as a linear combination of (*Xc*, *Yc*, *Zc*) (McGlone, 1989).

$$\begin{aligned}
&\left(m_{11}+\frac{x-xo}{f}m_{31}\right)X+\left(m_{12}+\frac{x-xo}{f}m_{32}\right)Y+\left(m_{13}+\frac{x-xo}{f}m_{33}\right)Z=...\\
&...=\left(m_{11}+\frac{x-xo}{f}m_{31}\right)Xc+\left(m_{12}+\frac{x-xo}{f}m_{32}\right)Yc+\left(m_{13}+\frac{x-xo}{f}m_{33}\right)Zc\\
&\left(m_{21}+\frac{y-yo}{f}m_{31}\right)X+\left(m_{22}+\frac{y-yo}{f}m_{32}\right)Y+\left(m_{23}+\frac{y-yo}{f}m_{33}\right)Z=...\\
&...=\left(m_{21}+\frac{y-yo}{f}m_{31}\right)Xc+\left(m_{22}+\frac{y-yo}{f}m_{32}\right)Yc+\left(m_{23}+\frac{y-yo}{f}m_{33}\right)Zc
\end{aligned} \tag{A.41}$$

This notation can be simplified as:

$$\begin{aligned}
C_{x1}X+C_{x2}Y+C_{x3}Z&=C_{x1}Xc+C_{x2}Yc+C_{x3}Zc\\
C_{y1}X+C_{y2}Y+C_{y3}Z&=C_{y1}Xc+C_{y2}Yc+C_{y3}Zc
\end{aligned} \tag{A.42}$$

Where:

$$\begin{aligned}
&C_{x1}=m_{11}+\frac{x-xo}{f}m_{31}\quad C_{x2}=m_{12}+\frac{x-xo}{f}m_{32}\quad C_{x3}=m_{13}+\frac{x-xo}{f}m_{33}\\
&C_{y1}=m_{21}+\frac{y-yo}{f}m_{31}\quad C_{y2}=m_{22}+\frac{y-yo}{f}m_{32}\quad C_{y3}=m_{23}+\frac{y-yo}{f}m_{33}
\end{aligned} \tag{A.43}$$

A.3.2 3D reconstruction by perspective projection equations

If there are *I* images for which the IOP and the EOP are known, then for any point which is visible in these image the 3D coordinates (X, Y, Z) can be found by the following linear equation (McGlone, 1989):

$$\begin{bmatrix}
C_{x1A}Xc_A+C_{x2A}Yc_A+C_{x3A}Zc_A\\
C_{y1A}Xc_A+C_{y2A}Yc_A+C_{y3A}Zc_A\\
C_{x1B}Xc_b+C_{x2A}Yc_B+C_{x3B}Zc_B\\
C_{y1B}Xc_B+C_{y2A}Yc_B+C_{y3B}Zc_B\\
...\\
C_{x1I}Xc_I+C_{x2I}Yc_I+C_{x3I}Zc_I\\
C_{y1I}Xc_I+C_{y2i}Yc_I+C_{y3I}Zc_I
\end{bmatrix}=\begin{bmatrix}
C_{x1A} & C_{x2A} & C_{x3A}\\
C_{y1A} & C_{y2A} & C_{y3A}\\
C_{x1B} & C_{x2B} & C_{x3B}\\
C_{y1B} & C_{y2B} & C_{y3B}\\
... & ... & ...\\
C_{x1I} & C_{x2I} & C_{x3I}\\
C_{y1I} & C_{y2I} & C_{y3I}
\end{bmatrix}\begin{bmatrix}X\\Y\\Z\end{bmatrix} \tag{A.44}$$

Where for (image *A*):

$$
\begin{aligned}
C_{x1A} &= (m_{11})_A + \frac{x_A - xo_A}{f_A}(m_{31})_A \\
C_{x2A} &= (m_{12})_A + \frac{x_A - xo_A}{f_A}(m_{32})_A \\
C_{x3A} &= (m_{13})_A + \frac{x_A - xo_A}{f_A}(m_{33})_A \\
C_{y1A} &= (m_{21})_A + \frac{y_A - yo_A}{f_A}(m_{31})_A \\
C_{y2A} &= (m_{22})_A + \frac{y_A - yo_A}{f_A}(m_{32})_A \\
C_{y3A} &= (m_{23})_A + \frac{y_A - yo_A}{f_A}(m_{33})_A
\end{aligned}
\tag{A.45}
$$

Equation (A.44) is of the form $L = AX$ and the least-squares solution is

$X = (A^T A)^{-1}(A^T L)$. The residuals can be calculated by:

$$R = AX - L \tag{A.46}$$

The standard deviation of unit weight, σ_0, is then calculated as:

$$\sigma_0 = \sqrt{\frac{R^T R}{r}} \tag{A.47}$$

Where *r* is the degrees of freedom and is given by:

r = (the number of equations) – (the number of unknowns) (A.48)

Then, the 1-sigma errors for the point are given by:

$$\sigma_X = \sigma_0\sqrt{Q_{11}} \qquad \sigma_Y = \sigma_0\sqrt{Q_{22}} \qquad \sigma_Z = \sigma_0\sqrt{Q_{33}} \tag{A.49}$$

Where Q_{xx} represents the diagonal elements of the matrix:

$$Q = (A^T A)^{-1} \tag{A.50}$$

Note that the dimensions of *Q* will correspond to the number of unknowns. So, the 1-simga error for a given point will consist of three components $\sigma_X, \sigma_Y, \sigma_Z$ corresponding

to the three unknowns (*X*, *Y*, *Z*) for each point. The total 1-sigma error of a point may also be calculated by:

$$\sigma = \sqrt{\sigma_X^2 + \sigma_y^2 + \sigma_Z^2} \tag{A.51}$$

To get the back-projected image coordinates:

$$\begin{aligned} x_p' &= xo - f\left(\frac{m_{11}\left(X_p'-Xc\right)+m_{12}\left(Y_p'-Yc\right)+m_{13}\left(Z_p'-Zc\right)}{m_{31}\left(X_p'-Xc\right)+m_{32}\left(Y_p'-Yc\right)+m_{33}\left(Z_p'-Zc\right)}\right) \\ y_p' &= yo - f\left(\frac{m_{21}\left(X_p'-Xc\right)+m_{22}\left(Y_p'-Yc\right)+m_{23}\left(Z_p'-Zc\right)}{m_{31}\left(X_p'-Xc\right)+m_{32}\left(Y_p'-Yc\right)+m_{33}\left(Z_p'-Zc\right)}\right) \end{aligned} \tag{A.52}$$

The residuals $\left(r_x, r_y\right)$ are then defined as the difference of the back-projected image coordinates $\left(x_p', y_p'\right)$and the measured image coordinates $\left(x_p, y_p\right)$:

$$\begin{bmatrix} r_x \\ r_y \end{bmatrix} = \begin{bmatrix} x_p'-x_p \\ y_p'-y_p \end{bmatrix} \tag{A.53}$$

The magnitude of these residuals provides a metric for determining the precision of each calculation. They may also be used to detect blunders in the image coordinates that were measured from the images.

It also may also be helpful to consider the total residual *r* for each point:

$$r = \sqrt{r_x^2 + r_y^2} \tag{A.54}$$