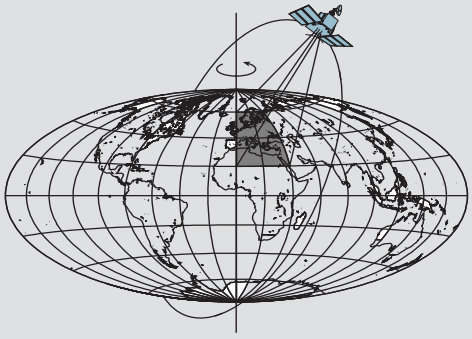
Linear Features in Photogrammetry

by

Ayman Habib
Andinet Asmamaw
Devin Kelley
Manja May



Report No. 450

Geodetic Science and Surveying
Department of Civil and Environmental Engineering and Geodetic Science
The Ohio State University
Columbus, Ohio 43210-1275

January 2000

Linear Features in Photogrammetry

By:
Ayman Habib
Andinet Asmamaw
Devin Kelley
Manja May

Report No. 450

Geodetic Science and Surveying
Department of Civil and Environmental Engineering and Geodetic Science
The Ohio State University
Columbus, Ohio 43210-1275

January, 2000

ABSTRACT

This research addresses the task of including points as well as linear features in photogrammetric applications. Straight lines in object space can be utilized to perform aerial triangulation. Irregular linear features (natural lines) in object space can be utilized to perform single photo resection and automatic relative orientation.

When working with primitives, it is important to develop appropriate representations in image and object space. These representations must accommodate for the perspective projection relating the two spaces. There are various options for representing linear features in the above applications. These options have been explored, and an optimal representation has been chosen.

An aerial triangulation technique that utilizes points and straight lines for frame and linear array scanners has been implemented. For this task, the MSAT (Multi Sensor Aerial Triangulation) software, developed at the Ohio State University, has been extended to handle straight lines. The MSAT software accommodates for frame and linear array scanners.

In this research, natural lines were utilized to perform single photo resection and automatic relative orientation. In single photo resection, the problem is approached with no knowledge of the correspondence of natural lines between image space and object space. In automatic relative orientation, the problem is approached without knowledge of conjugate linear features in the overlap of the stereopair. The matching problem and the appropriate parameters are determined by use of the modified generalized Hough transform. These techniques were tested using simulated and real data sets for frame imagery.

ACKNOWLEDGMENTS

This research project spanned for the last year and half. It is a comprehensive report covering much research done on linear features in the photogrammetry research group. During its long journey, many people contributed ideas and participated in discussions, which was very beneficial to the research. In particular, all members of the photogrammetry research group deserve thanks for their contributions. Had it not been for all of those discussions and contributions, this project would not have been successful.

We are all indebted to Dr. Tony Schenk for providing us with very constructive comments in the research, and for reviewing the final report.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

With the trend towards automatic extraction and recognition of features from digital imagery, it is becoming advantageous to utilize linear features in photogrammetric applications. Linear features can be used to increase redundancy and improve the geometric strength of photogrammetric adjustments.

The perspective relationship between image and object space for distinct points is established through the collinearity equations. The collinearity equations have been extended to accommodate different types of exposure stations: frame cameras, push-broom, three-line and panoramic linear array scanners. These equations are presented and discussed in Chapter 2.

The implementation of linear feature constraints in photogrammetric applications is dependent on the image and object space representations of the linear features. This paper discusses several approaches for representing both straight and natural lines. Chapter 3 discusses these approaches and justifies the optimal representation for this research.

An algorithm has been developed that utilizes linear features in single photo resection. This algorithm solves for the exterior orientation parameters (EOP's) associated with an image, and also accomplishes the matching of corresponding linear features in image and object space. An algorithm has been developed which uses linear features in automatic relative orientation. In its implementation, the correspondence of conjugate linear features in the overlapping images is determined. These matching problems are solved by the applying the modified generalized Hough transform, discussed in Chapter 4.

Chapter 5 presents the mathematical model adopted to include straight lines into aerial triangulation algorithms. The mathematical models used to accommodate for natural lines in single photo resection and automatic relative orientation are also presented. The

experiments and their results are presented in Chapter 6. The analysis of the results and the motivation for future research are discussed in Conclusions.

2. PERSPECTIVE TRANSFORMATION MODEL FOR POINT FEATURES

The collinearity condition is used to establish the relation between image and object space points. The collinearity model is first presented for the case of frame imagery. This model is then extended for linear array scanners, namely three-line, push-broom and panoramic. As a result, general collinearity equations have been developed to establish the relationship between image and object space points frame and linear array scanners.

2.1. COLLINEARITY EQUATION FOR FRAME IMAGERY

To reconstruct the position and shape of objects from imagery, the geometry at the time of exposure must be known. The geometry of frame imagery can be regarded with sufficient accuracy as a perspective or central projection, where the optical center of the camera is the perspective center (Habib, A., Beshah, B., 1997). The relationship between image and object coordinate system is shown in Figure 1.

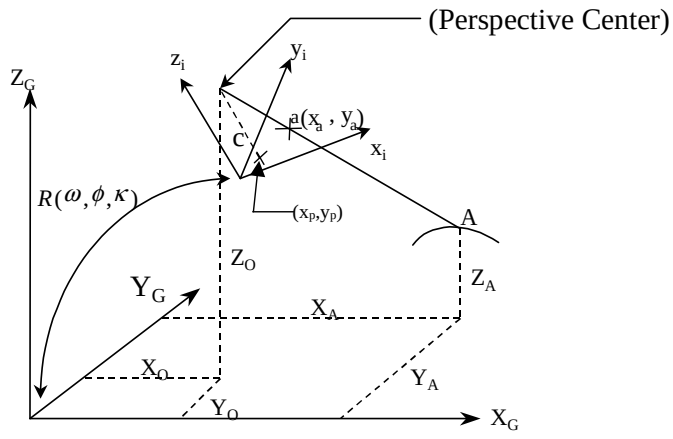


Figure 1: Relationship between the image coordinate system (x_i,y_i,z_i) and the object coordinate system (X_G,Y_G,Z_G) for frame imagery

The collinearity condition states that the vector from the perspective center to a distinct point on the image is a scaled version of the vector from the perspective center to the corresponding object point. The perspective center, **PC,** and a point on the focal plane, **a**, can be represented with respect to the image coordinate system according to Equation (1).

$$PC = \begin{bmatrix} x_p \\ y_p \\ c \end{bmatrix} \qquad a = \begin{bmatrix} x_a \\ y_a \\ 0 \end{bmatrix} \tag{1}$$

The vector, **v**, from the perspective center to a point on the focal plane can be expressed with respect to the image coordinate system as follows:

$$\vec{v} = \begin{bmatrix} x_a - x_p \\ y_a - y_p \\ -c \end{bmatrix} \tag{2}$$

The vector, **V**, from the perspective center to the object point, with respect to the ground coordinate system can be expressed by:

$$\vec{V} = \begin{bmatrix} X_A - X_O \\ Y_A - Y_O \\ Z_A - Z_O \end{bmatrix} \tag{3}$$

The image coordinate system is related to the ground coordinate system by a three-dimensional rotation matrix **R**, which contains the rotations ω,φ and κ necessary to align the two coordinate systems. This rotation matrix relates the vectors **v** and **V**. The collinearity model for each point can be written as:

$$\begin{bmatrix} x_a - x_p \\ y_a - y_p \\ -c \end{bmatrix} = \lambda \; R^T(\omega, \phi, \kappa) \begin{bmatrix} X_A - X_O \\ Y_A - Y_O \\ Z_A - Z_O \end{bmatrix}; \qquad \lambda \ldots \text{ scale} \tag{4}$$

By substituting the focal length **c** from the third row into the first two rows of the collinearity model, one can rewrite the collinearity equations as:

$$x_a = x_p - c\frac{N_x}{D} \qquad (5)$$

$$y_a = y_p - c\frac{N_y}{D}$$

where :

$$N_x = r_{11}(X_A - X_o) + r_{21}(Y_A - Y_o) + r_{31}(Z_A - Z_o)$$

$$N_y = r_{12}(X_A - X_o) + r_{22}(Y_A - Y_o) + r_{32}(Z_A - Z_o)$$

$$D = r_{13}(X_A - X_o) + r_{23}(Y_A - Y_o) + r_{33}(Z_A - Z_o)$$

X_A, Y_A, Z_A: object coordinates of point **A**

x_P, y_P, c: calibrated principal point position and focal length of the camera

$r_{11}, r_{12} \ldots r_{33}$: elements of the rotation matrix **R**

X_0, Y_0, Z_0: object coordinates of the perspective center

2.2. COLLINEARITY EQUATION FOR LINEAR ARRAY SCANNERS

Digital imagery can be acquired through digital cameras (Charged Coupled Device cameras – CCD cameras). These sensors quantize (gray level assignment) and sample (pixel size assignment) the electromagnetic energy incident on the sensor.

To attain the same resolution as frame photography, 20K × 20K staring array sensors would be necessary. The array size of digital sensors that are commercially available at this time allow for images with a size of 4K x 4K, which is not commensurate with the aforementioned resolution. To circumvent this problem, linear array scanners simulate the staring sensor array by a one-dimensional array of sensors in the focal plane and incremental integration of the object space.

Depending on the number of 1D-arrays, the scanning direction and the relation of the sensor with the flight direction, one differentiates between three-line, push-broom and panoramic linear array scanners.

Push-broom scanners have a one-dimensional array of sensors in the image plane perpendicular to the flight direction. Full coverage of a scene is achieved through the movement of the camera platform along the flight direction (Figure 2). The scene is a combination of images where each image has its own perspective center and consequently its own EOP's.

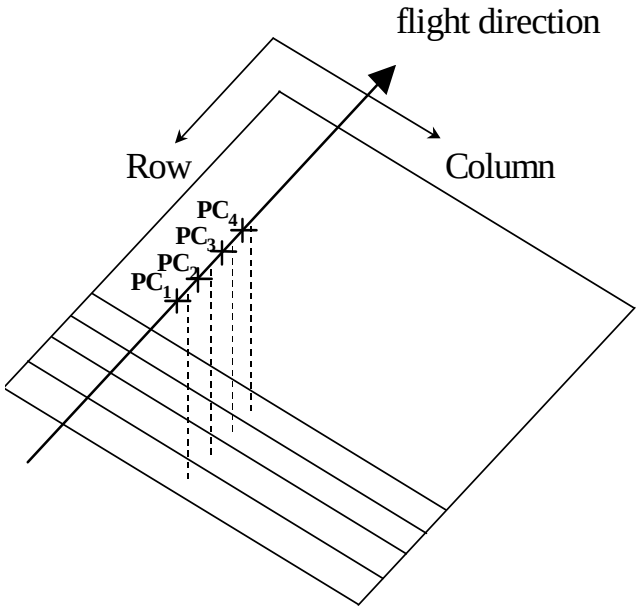


Figure 2: Scene of push-broom scanner

The relationship between image and object coordinate systems for push-broom scanners is shown in Figure 3.

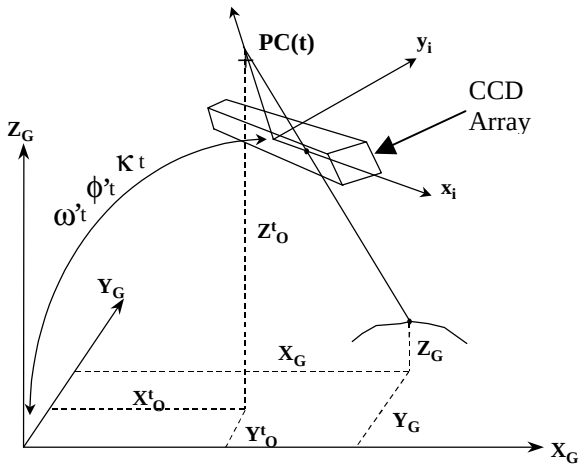


Figure 3: Relationship between image coordinate system (x_i,y_i,z_i) and object coordinate system (X_G,Y_G,Z_G) for push-broom scanners (flight direction parallel to y_i).

Push-broom scanners can only provide multiple coverage of the object space across the track through overlap of successive flight strips. This entails a time lapse between images used to form a stereopair, which can be very long in the case of satellite images. To circumvent this problem, three-line scanners were introduced. Three-line scanners provide triple coverage along the track within a few seconds, depending on sensor configuration and platform speed.

Three-line scanners have three 1-D sensor arrays in the image plane perpendicular to the flight direction. The sensor arrays capture the object space in forward, downward and backward locations respectively, resulting in three scenes (see Figure 4). Each image has its own associated exterior orientation parameters. Again, a scene is composed of a combination of images.

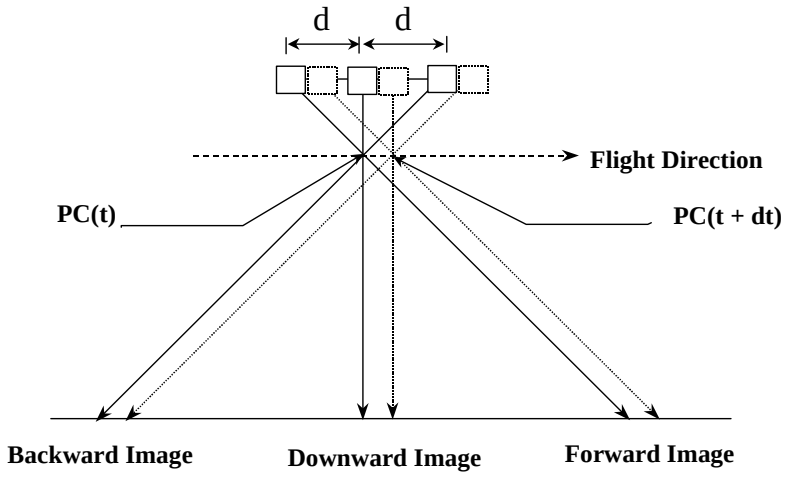


Figure 4: Exposure geometry for three-line scanners

Panoramic linear array scanners have one sensor line in the image plane that is swivel-mounted on a telescope. The sensor array line is parallel to the flight direction and rotates about the perspective center with an angle "α_t" (Figure 5). The swing angle "α_t" of the 1-D sensor array, the column coordinates of the scene and the EOP's of each image are time dependent. A scene is a combination of images that are captured parallel to the flight direction (Figure 6)

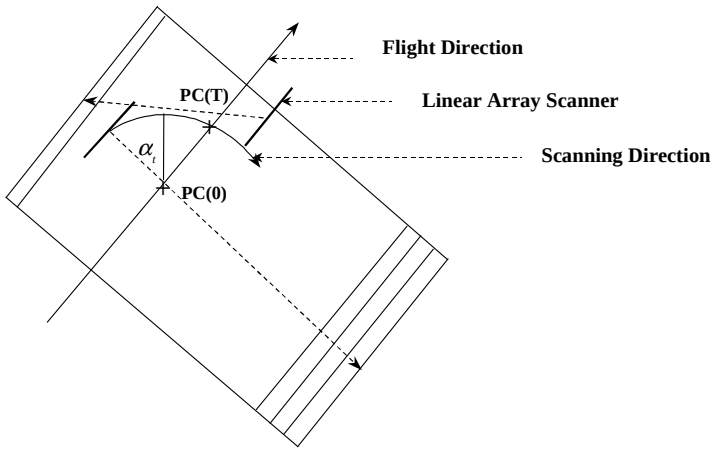


Figure 5: Scene capture with panoramic linear array scanners

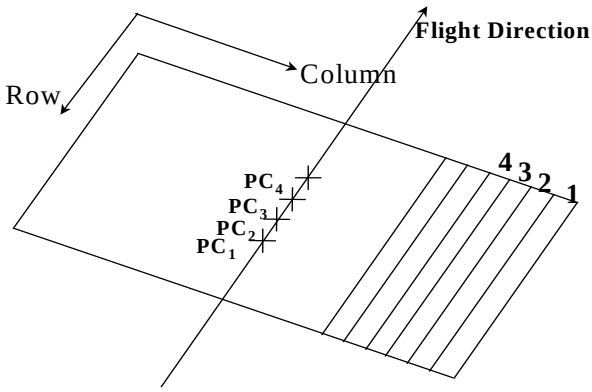


Figure 6: Panoramic linear array scanner scene

For panoramic linear array scanners, one can define the image coordinate system as being oriented in such a way that the x-axis is parallel to the flight direction. The swing angle is applied to the y-z plane and is time dependent (see Figure 7). We refer to this rotated image coordinate system as the telescope coordinate system. The relationship between the image and the object coordinate system is characterized by a three dimensional rotation (ω,φ,κ) and is illustrated in Figure 8.

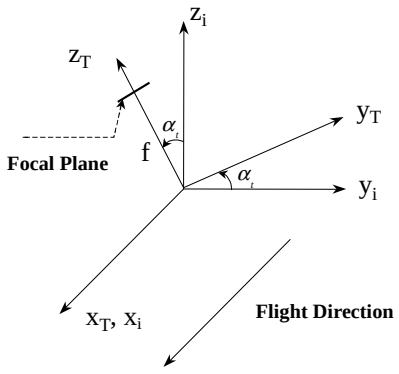


Figure 7: Relation between image and telescope coordinates for panoramic linear array scanners

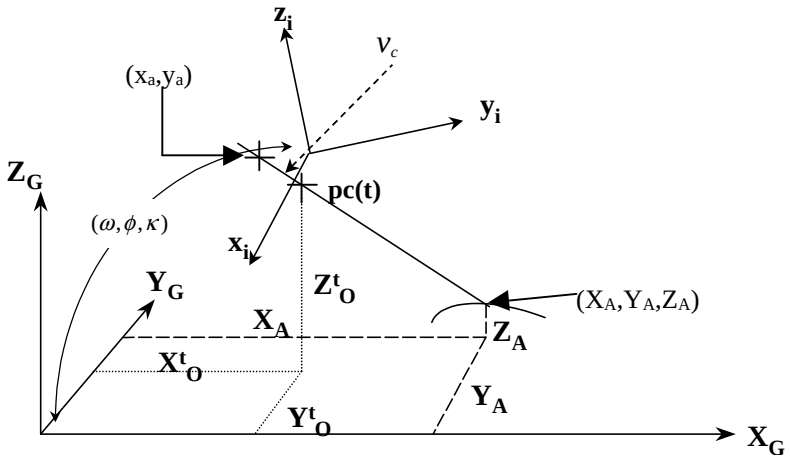


Figure 8: Relationship between image coordinate system (x_i,y_i,z_i) and object coordinate system (X_G,Y_G,Z_G) for panoramic linear array scanners

2.3. THE GENERALIZED COLLINEARITY MODEL

The objective now is to modify the collinearity model used for frame imagery (Equation (4)) in such a way that it is also valid for push-broom, three-line and panoramic linear array scanners. The most general scenario for linear array scanners is that of panoramic linear array scanners. Collinearity models for the other sensors of interest can easily be derived from this model.

For linear array scanner imagery, the vector from the perspective center "**PC(t)**" to an image point "**a**" is time dependent. To better describe this vector, we can include information about image motion compensation[1] (see Equation (6)).

$$PC(t) = \begin{bmatrix} imc(t) + x_p \\ y_p \\ c \end{bmatrix} \tag{6}$$

An image point "**a**" in the image coordinate system can be represented as:

[1] Image motion compensation is often utilized at exposure stations to avoid the image blur and distorted footprints associated with a moving platform.

$$a = \begin{bmatrix} x_a \\ y_a \\ 0 \end{bmatrix} \tag{7}$$

The vector from the perspective center to the object point in the ground coordinate system is time dependent (superscript t):

$$\vec{V} = \begin{bmatrix} X_A - X_0^t \\ Y_A - Y_0^t \\ Z_A - Z_0^t \end{bmatrix} \tag{8}$$

In addition to the rotation matrix **R**, containing the time dependent angles ω^t, ϕ^t and κ^t, a rotation matrix for the swing angle "α_t" in panoramic linear array scanners must be incorporated. $\mathbf{R}_\alpha$ establishes the rotational relationship between the camera and telescope coordinate system:

$$R_\alpha = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_t) & -\sin(\alpha_t) \\ 0 & \sin(\alpha_t) & \cos(\alpha_t) \end{bmatrix}; \qquad \alpha_t \ldots \text{ swing angle at time } \mathbf{t} \tag{9}$$

The general collinearity model for push-broom, three-line and panoramic linear array scanners is now established as:

(10)

$$\begin{bmatrix} x_a^t - imc(t) - x_p \\ y_a^t - y_p \\ -c \end{bmatrix} = \lambda R^T(\alpha_t) R^T(\omega_t, \phi_t, \kappa_t) \begin{bmatrix} X_A - X_0^t \\ Y_A - Y_0^t \\ Z_A - Z_0^t \end{bmatrix}$$

The collinearity equations for frame imagery (Equation (5)) are now modified to accommodate for the aforementioned sensors:

$$x_a^t = x_P + imc(t) - c\frac{N_x}{D} \qquad (11)$$

$$y_a^t = y_P - c\frac{N_y}{D}$$

where

$$N_x = r_{11}^t(X_A - X_0^t) + r_{21}^t(Y_A - Y_0^t) + r_{31}^t(Z_A - Z_0^t)$$

$$N_y = r_{12}^t(X_A - X_0^t) + r_{22}^t(Y_A - Y_0^t) + r_{32}^t(Z_A - Z_0^t)$$

$$D = r_{13}^t(X_A - X_0^t) + r_{23}^t(Y_A - Y_0^t) + r_{33}^t(Z_A - Z_0^t)$$

x_a^t , y_a^t: image coordinate measurement of point **a** at time **t**

X_A, Y_A, Z_A: object coordinates of point **A**

x_P, y_P, c: calibrated principal point position and principal distance of the camera

$r_{11}^t, r_{12}^t ... r_{33}^t$: time dependent elements of the combined rotation matrices $R^T(\alpha_t)R^T(\omega_t, \phi_t, \kappa_t)$

$imc(t)$: image motion compensation at time **t**

X_0^t, Y_0^t, Z_0^t: time dependent object coordinates of the perspective center

A scene of linear array scanner imagery is composed of a combination of images, each having a set of EOP's as unknown parameters. Therefore, each row of push-broom and three-line scanner imagery and each column of panoramic linear array scanner imagery has its own EOP's. This results in a large number of correlated unknown parameters. The objective is to reduce the number of involved parameters to avoid singularities in the solution process.

Reducing the number of unknown EOP's can be achieved through modeling the system trajectory by polynomials or by using the approach of orientation images. Modeling the system trajectory by polynomials determines the change of the EOP's with

time (see Figure 9). The order of the polynomial depends on the smoothness of the trajectory.

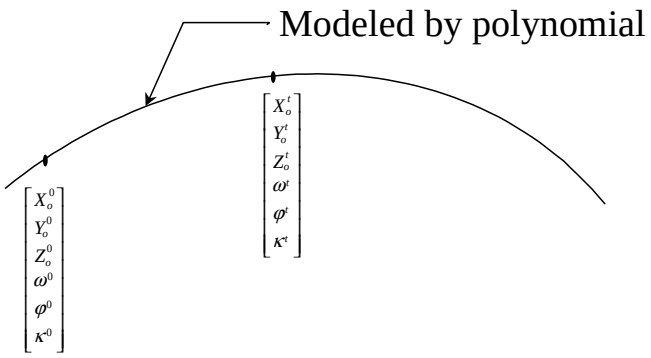


Figure 9: Modeling system trajectory by a polynomial

The approach of reducing the number of EOP's using polynomial modeling of the system trajectory has the following disadvantages:

- The flight trajectory may be too rough to be represented by a polynomial,
- It is difficult to incorporate GPS/INS observations.

A better approach to reduce the number of EOP's is by the use of orientation images. Orientation images are usually designated at equal intervals along the system's trajectory. The EOP's of an image captured at a certain time **t** are then an interpolation of the EOP's of two neighboring images, the so-called orientation images (see Figure 10).

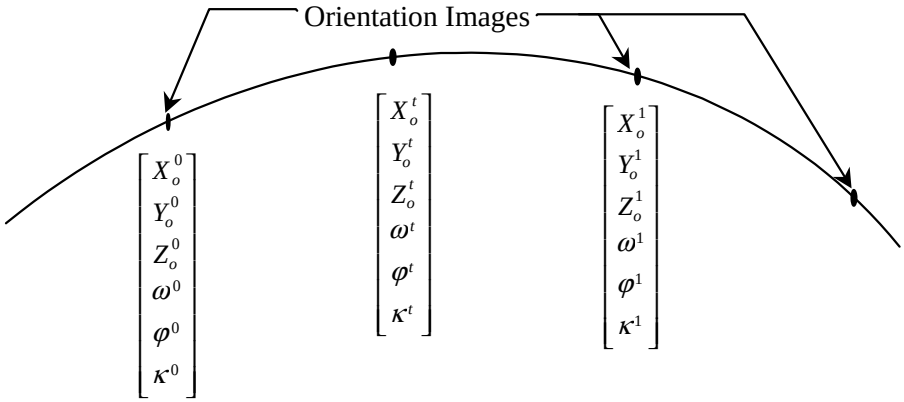
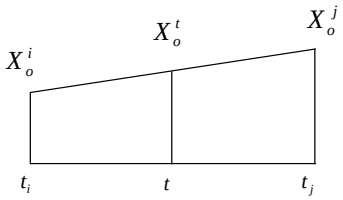


Figure 10: Orientation images and their EOP's

One example of utilizing orientation images is by linear interpolation (Equation (13)). The appropriate interpolation technique depends on the characteristics of the trajectory.



$$X_o^t = \frac{t - t_i}{t_j - t_i} X_o^j + \frac{t_j - t}{t_j - t_i} X_o^i \tag{13}$$

X_o^i … EOP's of the preceding orientation image at time t_i

X_o^j … EOP's of the succeeding orientation image at time t_j

In summary, the scene to image coordinate transformation is dependent on the type of sensor used. The panoramic model is the most general case among the sensors mentioned. The models for frame imagery, three-line and push-broom scanners are derived from this model. In the implementation of this general model, push-broom scanners and three-line scanners have a swing angle of zero degrees, many orientation images in a scene, and time dependency in the scene coordinates. In the case of frame

cameras, the image coordinates are not time dependent, the swing angle is zero, and only one orientation image constitutes a scene.

3. LINEAR FEATURES

The implementation of linear feature constraints in photogrammetric applications is dependent on the image and object space representations of the linear features. This chapter discusses several approaches for representation and justifies the optimal representation for this research.

3.1. LINEAR FEATURES VERSUS DISTINCT POINTS IN PHOTOGRAMMETRY

Most photogrammetric applications are based on the use of distinct points. These points are often obtained through measurements in an analog or digital environment. Recently, more attention has been drawn to linear features. There are several motivations for the utilization of linear features in photogrammetry:

- Points are not as useful as linear features when it comes to higher level tasks such as object recognition.
- Automation of the map making process is one of the major tasks in digital photogrammetry and cartography. It is easier to automatically extract linear features from the imagery rather than distinct points (Kubik, 1991).
- Images of a man made environment are rich with linear features.

The linear features of interest to this research include straight lines and natural lines. The term *natural lines* refers to free form lines. For both kinds of linear features, two issues have to be addressed to successfully include them into existing photogrammetric applications:

1. The representation of linear features in both image and object space,

2. The formulation of the mathematical relationship between image and object space linear features.

There has been a substantial body of research dealing with geometric modeling and perspective transformation of linear features in imagery (Habib, 1997; Mulawa and Mikahil, 1988; Tommaselli and Tozzi, 1996; Tommaselli and Lugnani, 1988, Tommaselli and Tozzi, 1992; Ayache and Faugeras, 1989; Mikhail, 1993; Wilkin, 1992; Kubik 1991).

The next section deals with the first issue by introducing different options of mathematical representations of straight and natural lines. The second issue will be discussed in Chapter 5 by developing the specific models for including straight lines in aerial triangulation, and natural lines in single photo resection and automatic relative orientation. These models are developed for frame cameras as well as linear array scanners.

3.2. MATHEMATICAL REPRESENTATION OF LINEAR FEATURES

The representation of linear features in both image and object space is closely related to the perspective projection. For this reason, we have to discuss the representation issue before attempting to utilize the image to object space perspective transformation.

3.2.1. Representation of straight lines

One way of representing straight 3-D lines is by using two points along the line (six-parameter representation), Figure 11. This representation is not unique, since it can be defined by any two points along the line.

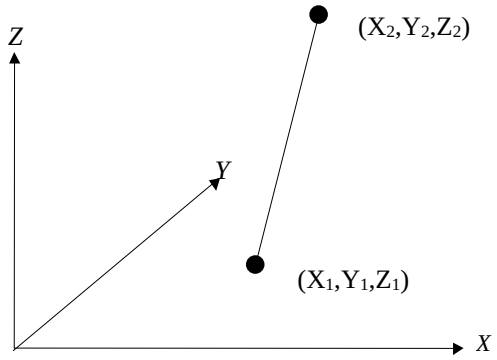


Figure 11: Six parameter representation of a line

Mulawa and Mikhail, (1988) chose to represent the 3-D line by a point along the line and its direction vector (Figure 12). This representation is not unique. Two constraints were applied: the norm of the direction vector was chosen as unity and the point along the line was chosen as the closest point to the origin. They developed the perspective relationship between image and object space using this representation. Tommaselli and Tozzi (1996) also represented the 3-D line using a point and a direction vector. They derived two correspondence equations for space resection, using the linear features as ground control.

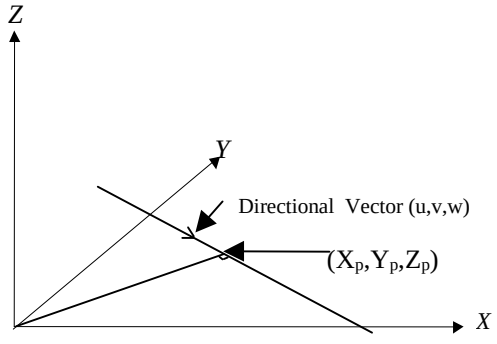


Figure 12: Point and direction vector representation of a line

An optimal, unique representation of 3-D lines would require four parameters. Optimal representations are thought to be more suitable than other representations since they have no redundancy (i.e. they are unique for each line – Roberts, 1988). Ayache and Faugeras, 1989 suggested the representation of 3-D straight lines as the intersection of two planes, one plane parallel to the x-axis and the other parallel to the y-axis. Each of these two planes is defined by two parameters (Equations (14) and (15)).

$$X = a\,Z + p \qquad \text{(Plane parallel to the y-axis)} \qquad (14)$$

$$Y = b\,Z + q \qquad \text{(Plane parallel to the x-axis)} \qquad (15)$$

Using these two constraints, a line can be represented with four parameters rather than with six. Consequently, the 3-D line is defined by the four-dimensional vector (a, b, p, and q). It can be proven that these parameters have the following characteristics:

- the direction of the line is given by the vector (a, b, 1),
- the point of intersection of the line with the XY-plane has the coordinates (p, q, 0) (Figure 13).

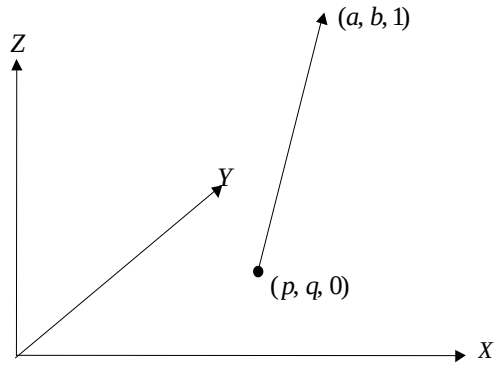


Figure 13: Representation of 3-D lines as the intersection of two planes parallel to x- and y-axes

Since the direction vector of the line is (a, b, 1), this form can not be used to represent lines that are perpendicular to the z-axis (i.e. parallel to the xy-plane). To avoid this singularity, one can represent the 3-D line as the intersection of two planes, one parallel to the z-axis and the other parallel to the y-axis. Once again, this representation can not represent lines that are orthogonal to the x-axis. For these lines, one can use two planes

that are parallel to the x- and z-axes, respectively. Using this representation, Habib (1997) introduced the mathematical model for the back and forth perspective transformation of linear features between image and object space.

The above mentioned representations have the following problems:

- Optimal representations always have singularities (they can not represent all 3-D lines in space).
- They represent infinite lines rather than line segments.
- The covariance matrix resulting from the bundle adjustment does not characterize the uncertainty of the actual line segment under consideration. Rather, it indicates the uncertainty associated with the infinite line. To explain, consider the following: the above mentioned representations required one point along the line (e.g. the intersection point with the xy-plane or the closest point to the origin). Let us assume that we are using the point (p, q, 0). One can see that a segment with a small uncertainty that is far from the xy-plane will have a covariance matrix with a larger trace than a segment with greater uncertainty but closer to the xy-plane. Thus, the uncertainty of the derived parameters will propagate according to the distance of the line segment from the xy-plane, yielding misleading dispersion matrices (Figure 14).

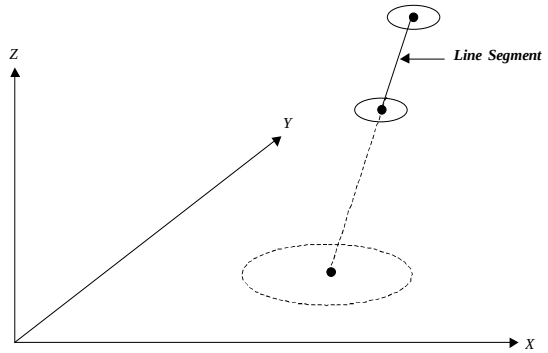


Figure 14: Relationship between the uncertainty of the line segment and that of the representation

For these reasons, a more appropriate representation is desired. In this research, straight lines are represented in the object space by using two points along the line. In

this way, the line segment is well localized in the object space. This representation is attractive since such coordinates can be easily introduced or obtained from a GIS database. In the image space, two points along the line will be used to define the 2-D line in one image while the same line is represented using polar parameters in the overlapping images. The polar parameters are defined by (ρ,θ), where ρ is the normal distance from the origin to the line, and θ is the angle of the normal vector to the line.

3.2.2. Representation of natural lines

In object space, a natural line can be represented by an analytical function or by a sequence of 3-D points along the linear feature. Representing a natural line by an analytical function minimizes the amount of information required to describe the line. Only the parameters of the function itself have to be transferred and stored during the photogrammetric application. However, a natural line in object space can not be faithfully represented by an analytical function that describes its slightest detail without any generalization.

In this research, we represent natural lines as a sequence of 3D-coordinates of points along the object space line. The natural lines in the image space are given as a sequence of 2D-coordinates corresponding to the points along the line. A sequence of 3D-object coordinates corresponding to a linear feature can be captured by a terrestrial mobile mapping system (e.g. GPSVan) or can be obtained from an existing database. The advantage of this representation is that the original data is not modified by a functional description or by interaction of the user.

In this research, single photo resection and automatic relative orientation are performed using natural lines. The adopted algorithms accommodate for the following:

- The solution of the matching problem between:
  - corresponding linear features in image and object space (for single photo resection),

- conjugate linear features in overlapping images (for automatic relative orientation).

- Performing the specific photogrammetric task:
  - determination of the exterior orientation parameters (EOP's) (for single photo resection),
  - determination of the relative orientation parameters (ROP's) of a given stereopair (for automatic relative orientation).

The matching problems were solved through the use of the modified generalized Hough transform. This technique is discussed in the next chapter.

4. HOUGH TRANSFORM TECHNIQUES

4.1. THE HOUGH TRANSFORM

The Hough Transform algorithm is a curve detection and segmentation technique. According to Ballard and Brown (1982), this technique is used if the location of a curve is not known but its shape is known as a parametric curve.

Usually, after automatic point extraction in digital imagery, one has a list of points in image space which are assumed to represent a certain analytical function. The Hough transform searches for the extracted points that satisfy this known function. The parameters of this function are the result of the Hough transform algorithm.

To explain the mechanics of the Hough transform, let us consider the case of straight lines. Assume we are giving a cluster of points and want to find the points that belong to straight lines. The parametric representation of a straight line is:

$$y = mx + q \tag{16}$$

In the application of the Hough transform, the dependent variables are **x** and **y**, while **m** and **q** are the independent parameters. Each point along a straight line in x-y space satisfies Equation (16), where **m** and **q** are constants for any particular line. Each point in

x-y space is represented by a line in m-q space. An intersection of lines in m-q space indicates that the points in x-y space contributing to this intersection belong to a single straight line in x-y space. A similar technique can be applied to other shapes such as circles.

As demonstrated above, the parametric representation of the feature needs to be known. This means that we need to know the analytical function that approximates the linear feature as well as possible. Since we do not want to generalize a natural line by an analytical function, the Hough transform can be altered to deal with its representation as a list of points. This leads to the introduction of the generalized Hough transform.

4.2. THE GENERALIZED HOUGH TRANSFORM

The generalized Hough transform can be used to match a shape in an image or a database model (the template) with instances of this shape in another image without the need for the description of the shape by an analytical function (Ballard and Brown (1982) and Zahran (1997)).

Assume to have a shape described by a list of points. The objective is to compare a template shape with a number of candidate shapes and to determine conjugate shapes. The two matching shapes can differ in scale and orientation.

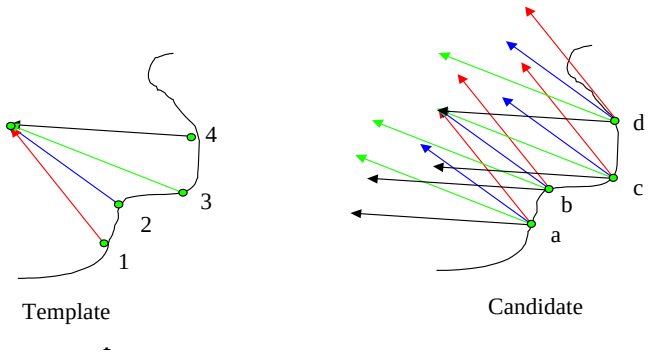


Figure 15: Matching natural lines using generalized Hough transform

The algorithm is outlined as follows:

- For each point of the template shape, the vector (Δx, Δy) to an arbitrary reference point is computed (see Figure 15). The reference point can be chosen as the centroid of the shape.
- An R-table is created for the template shape. The R-table lists the gradient (ϕ) at points along the boundary, together with (Δx, Δy) offsets between these points and the arbitrary reference point (Figure 16).

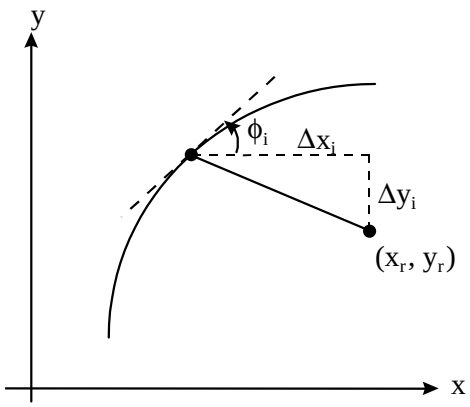


Figure 16: Geometry used to form the R-Table

- A 2-D accumulator array is formed for possible reference points, A (x_{rmin} : x_{rmax}, y_{rmin} : y_{rmax}). The elements of this array should be initialized to zero. If the candidate shape has a different scale and orientation, a 4D-accumulator array which also accommodates for the scale factor **s** and the orientation θ can be applied. In this case, we must have initial approximations for the scale factor and the rotation.
- For each boundary point on the candidate shape, the gradient (ϕ) is computed. The offsets Δx, Δy that correspond to this gradient are acquired from the R-table. The reference point associated with each boundary point can be computed according Equation (17).

$$x_r = x + \Delta x(\phi) \qquad (17)$$

$$y_r = y + \Delta y(\phi)$$

To accommodate for the scale and orientation difference, the reference point can be computed according Equation (18).

$$\begin{bmatrix} x_r \\ y_r \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + s \cdot \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \cdot \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{18}$$

- The accumulator array element corresponding to the reference point $(\mathbf{x_r},\mathbf{y_r})$ is incremented by one.
- The possible center location, as well as the associated scale factor and rotation, is designated by a peak in the accumulator array.
- Conjugate points can be determined by tracking the points that have contributed to the maximum peak in the accumulator array.

4.3. THE MODIFIED GENERALIZED HOUGH TRANSFORM

The modified generalized Hough transform is used to incorporate mathematical models into the matching process. In general, the matching process between data sets can be described as follows:

1. A mathematical model is established relating the involved data sets (see Figure 17). The relation between the data sets can be described as a function of its parameters: $F(x_1, x_2,\ldots x_n)$.
2. An accumulator array is formed for the parameters. The number of parameters to be simultaneously solved will designate the dimension of the accumulator array.
3. Approximations are made for parameters which are not yet to be determined. The cell size of the accumulator array depends on the quality of the initial approximations; poor approximations will require larger cell sizes.
4. Every possible match between the data sets is evaluated, incrementing the accumulator array at the location of each solution.
5. The maximum peak in the accumulator array will indicate the location of the desired solution.

6. After each parameter is determined, the approximations are updated.
7. For the next iteration, decrease the cell size of the accumulator array, and repeat steps 2-6.

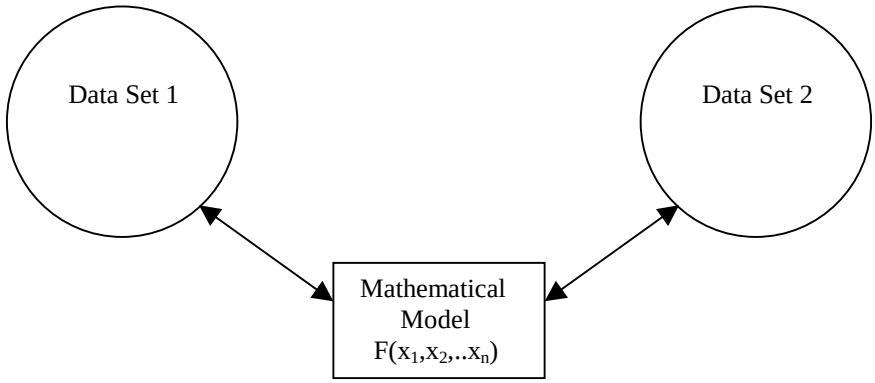


Figure 17: Mathematical model relating data sets.

In the applications of this research, the generalized Hough transform is modified to incorporate well-established mathematical models. For example, in single photo resection, the relationship between matching entities in image and object space is established by the collinearity condition. In automatic relative orientation, the coplanarity condition is utilized for matching conjugate features. The implementation of these techniques is explained in detail in Chapter 5.

5. LINEAR FEATURES IN PHOTOGRAMMETRIC APPLICATIONS

In this research, photogrammetric applications have been developed that utilize straight lines as well as natural lines. Straight lines as well as points are utilized in an aerial triangulation technique. Natural lines have been utilized in single photo resection and automatic relative orientation algorithms. These photogrammetric applications have been developed for frame imagery as well as linear array scanners.

As an added benefit, these techniques also solve the involved matching problems. In the single photo resection algorithm, the matching of linear features between image and object space is automatically accomplished. Also, the matching of conjugate features between the images of a stereopair is achieved as a result of the automatic relative orientation algorithm.

5.1. STRAIGHT LINES IN AERIAL TRIANGULATION

5.1.1. Frame imagery

Assume that we are given straight lines that appear in a group of overlapping images, together with some tie and control points. To define a 2-D line in the image space, we will measure two points along the line in one of the images. In the remaining images, the polar representation (ρ_i, θ_i) will be used to represent this line (see Figure 18). The objective is to combine these measurements in a unified bundle adjustment to solve for the EOP's of the imagery, the ground coordinates of the tie points and the ground coordinates of the points defining the object lines.

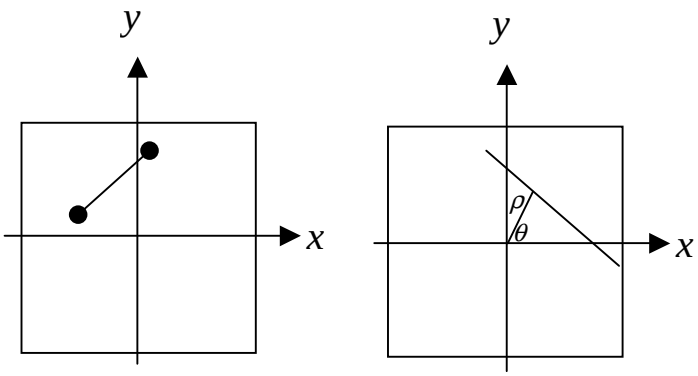


Figure 18: Representation of the image line in overlapping images

In the first image, the relationship between the image coordinates and the object coordinates is given by the collinearity condition. For the remaining images, a constraint is introduced that ensures that the two points defining the line in object space belong to the line represented by (ρ_i, θ_i), when transformed into the i^{th} image space. We define the

object line plane as the plane through the perspective center and the two points defining the line in object space (Figure 19). Also, we can define the *image line plane* as the plane that contains the perspective center and the image line (Figure 20). It should be noted that the image and object line planes (from a single image) that correspond to the same line coincide with each other.

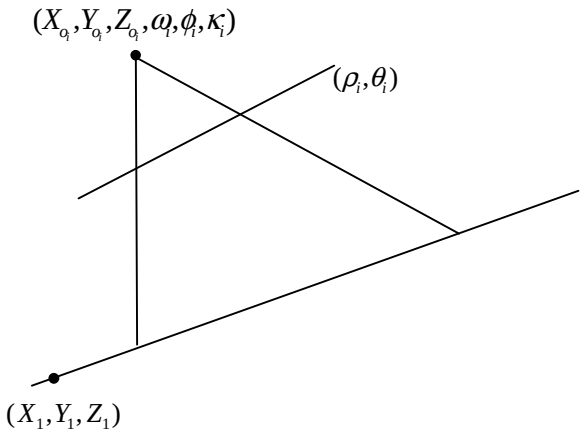


Figure 19: Object line plane

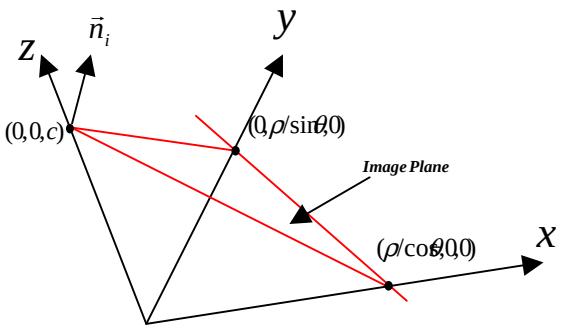


Figure 20: Image line plane

The image line plane associated with image i contains the two vectors (ρ_i / cos θ_i, 0, -c) and (0, ρ_i / sin θ_i, -c). These two vectors connect the perspective center with the intersection points of the image line with the x and y axes of the image coordinate system, respectively. The normal **n** to the image line plane can be defined as the cross product of these two vectors, (Equation (19).) The last term in Equation (19) can be

considered as a scale factor, and therefore can be ignored when defining the normal to the image line plane. This vector is defined relative to the image coordinate system.

$$\vec{n}_i = \begin{bmatrix} \rho_i / \cos\theta_i \\ 0 \\ -c \end{bmatrix} \times \begin{bmatrix} 0 \\ \rho_i / \sin\theta_i \\ -c \end{bmatrix} = \begin{bmatrix} c\ \cos\theta_i \\ c\ \sin\theta_i \\ \rho_i \end{bmatrix} \frac{\sin\theta_i\ \ \cos\theta_i}{\rho_i} \tag{19}$$

The object line plane includes the vector ($X_1 - X_{oi}$, $Y_1 - Y_{oi}$, $Z_1 - Z_{oi}$) from the perspective center to the first object point defining the line. This vector is defined relative to the ground coordinate system. Thus, it can be referenced to the image coordinate system through multiplication with the transpose[2] of the 3-D rotation matrix associated with that image. The dot product of this vector with the normal vector to the image line plane equals zero, Equation (20). This constraint is embedded in the bundle adjustment to ensure the coplanarity of perspective center, the image line and object point. Similarly, another constraint is written for the second object point along the line.

$$R^T_{\omega_i,\phi_i,\kappa_i} \begin{bmatrix} X_1 - X_{o_i} \\ Y_1 - Y_{o_i} \\ Z_1 - Z_{o_i} \end{bmatrix} \bullet \begin{bmatrix} c\ \cos\theta_i \\ c\ \sin\theta_i \\ \rho_i \end{bmatrix} = 0 \tag{20}$$

One can derive Equation (20) in an alternative way as follows: The equation of the image line in the i^{th} image is:

$$x\cos\theta_i + y\sin\theta_i = \rho_i \tag{21}$$

Equation (21) can be written as the dot product of two vectors, namely:

$$\begin{bmatrix} x \\ y \\ -c \end{bmatrix} \bullet \begin{bmatrix} c\cos\theta_i \\ c\sin\theta_i \\ \rho_i \end{bmatrix} = 0 \tag{22}$$

[2] The rotation matrix is orthogonal

The image point that corresponds to the object point (X_1, Y_1, Z_1) in the i^{th} image can be found through the collinearity equations, Equation (23).

$$\begin{bmatrix} x_{1_i} \\ y_{1_i} \\ -c \end{bmatrix} = \lambda \, R^T_{\omega_i, \phi_i, \kappa_i} \begin{bmatrix} X_1 - X_{o_i} \\ Y_1 - Y_{o_i} \\ Z_1 - Z_{o_i} \end{bmatrix} \tag{23}$$

By substituting Equation (23) into Equation (22) and eliminating the scale factor, one gets the constraint in Equation (20). This means that the object point along the linear feature, when transformed into the image space, must satisfy the equation that describes the feature shape in that image. A similar technique can be used to develop constraints for higher order linear features.

In summary, one can describe the suggested algorithm as follows:

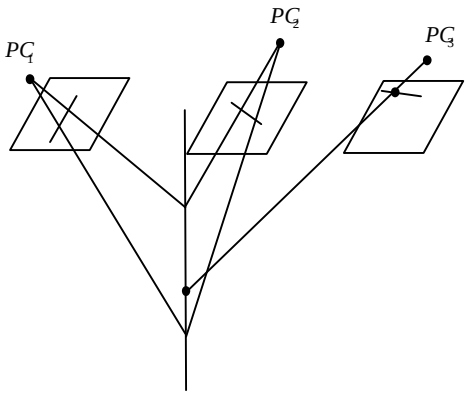


Figure 21: Geometry of aerial triangulation algorithm

1. In one image, two points are measured along the straight line. For each of these image points, the corresponding object point will lie on the straight line defined by the perspective center and the measured image point (using the collinearity equations). These points need not be visible or identifiable in the remaining images (see Figure 21).
2. In subsequent images, one defines the line in terms of polar coordinates. The plane through the perspective center and the image line (the *image line plane*) is

established, using (ρ_i,θ_i) and the position of the perspective center. We define the normal, $\mathbf{n}_i$ to each *image line plane*.

3. A vector is defined from the perspective center of image i to one of the object points defining the line.
4. This vector is then rotated into the i^{th} image, and its dot product with $\mathbf{n}_i$ is constrained to be zero. One should note that the vector and the planes should not be coplanar.

In Equation (20), the unknowns are the ground coordinates defining the object line and the EOP's of the involved images. The image line parameters (ρ_i, θ_i) are the observed quantities. In the case of a stereopair, an additional line will add six unknowns (the ground coordinates of the two points along the line). This line will contribute six equations: four collinearity equations for the two points measured in the first image, and two constraints in the form of Equation (20) in the second image. Thus, straight lines, as opposed to tie points, do not provide any redundancy towards the estimation of the relative orientation parameters. Therefore, it is not advantageous to use straight lines for the relative orientation of a stereopair.

The same constraint, Equation (20), can be used as the mathematical model for introducing linear features as control. In this case, the ground coordinates defining the object line are available and may be treated as observed quantities or as constants. Once again, these control points along the object lines need not be visible or identifiable in the overlapping images. The polar representation will be used to define the linear feature in the overlapping images. Two or more points in the object space will define a control line. Each point will contribute one equation. Thus, for single photo resection a minimum of three control lines is required to solve for the EOP's. Control lines can be easily obtained from an existing GIS database or from data collected by a terrestrial mobile mapping system, such as the GPSVan developed by The Ohio State University (The Center for Mapping, 1991). Consequently, control linear features will be cheaper and easier to obtain than distinct control points.

5.1.2. Linear array scanner imagery

Linear array scanners have one or more 1-D arrays of CCD sensors in the image plane. The electromagnetic energy incident upon these sensors at a given time will constitute an image. Movement of the platform and/or rotation of the lens configuration will enable successive images of the ground. A scene is defined as a sequence of linear array scanner images.

Because of the nature of the linear array scanners, straight lines in object space may not appear as straight lines in the image space. The individual images may be slightly shifted against each other perpendicular to the flight direction. This is due to slight changes in the system trajectory.

In the object space, straight lines are chosen to be represented by two points along the line. The corresponding line in the image space will be represented as a sequence of points. The points along the object line are used to define the line during the bundle adjustment.

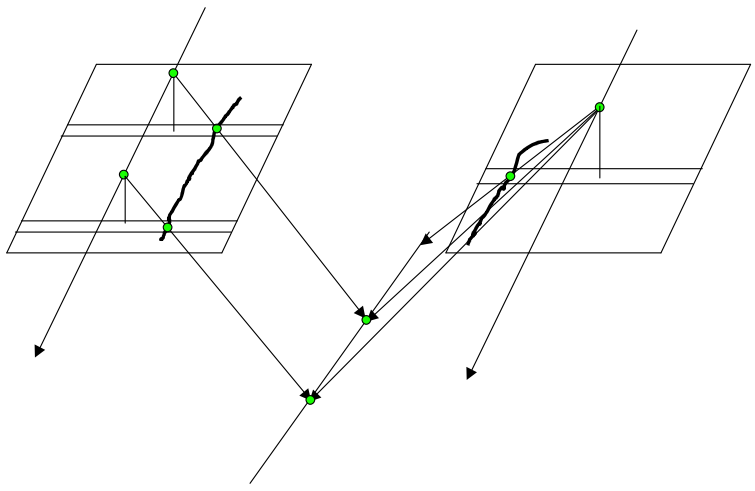


Figure 22: Object space geometry and straight lines in linear array scanners

As shown in Figure 22, two points in one scene constitute a line. If the ground coordinates of these two points are known, it is considered a control line. If only the

image coordinates of the two points are known, this line is called a tie line. The object point, the corresponding image point and the perspective center of the exposure station lie on a single light ray. Therefore, the generalized collinearity equations (Equation (11)) can be applied to each of the two points defining the line. The objective is to increase the accuracy of the bundle adjustment by constraining a straight line in object space to be a straight line in image space.

The vector from the perspective center to any image point along the line can be defined with respect to the ground coordinate system as:

$$\vec{V}_1 = R(\omega^t, \phi^t, \kappa^t) \begin{bmatrix} x - x_p \\ y - y_p \\ -c \end{bmatrix} \tag{24}$$

The multiplication with the rotation matrix **R**, transforms the vector from the image coordinate system into the ground coordinate system.

In another scene, the vector from the perspective center to the first object point along the line is defined as:

$$\vec{V}_2 = \begin{bmatrix} X_1 - X_O^t \\ Y_1 - Y_O^t \\ Z_1 - Z_O^t \end{bmatrix} \tag{25}$$

The vector from the perspective center to the second object point along the line is defined as:

$$\vec{V}_3 = \begin{bmatrix} X_2 - X_O^t \\ Y_2 - Y_O^t \\ Z_2 - Z_O^t \end{bmatrix} \tag{26}$$

Both vectors are given with respect to the ground coordinate system.

As illustrated in Figure 23, the vectors from the perspective center to each scene point along the line should lie on the plane that is defined by the perspective center and the two object points defining the straight line. This condition could be formulated as:

$$(\vec{V}_2 \times \vec{V}_3) \bullet \vec{V}_1 = 0 \tag{27}$$

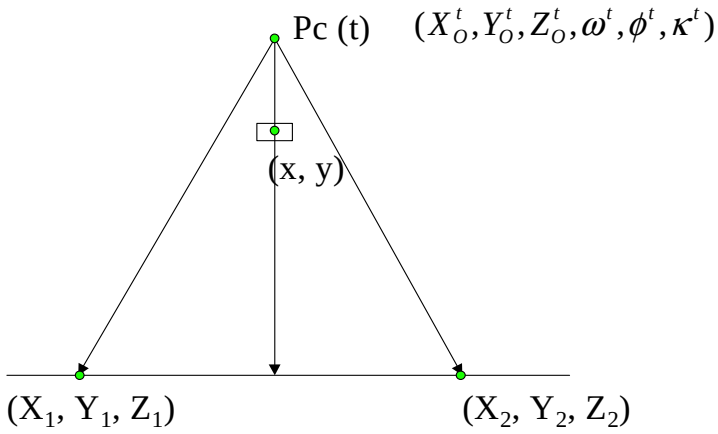


Figure 23: Object plane for linear array scanner imagery

This constraint for straight lines in aerial triangulation is a function of the parameters shown in Equation (28).

$$f(X_1, Y_1, Z_1, X_2, Y_2, Z_2, X_O^t, Y_O^t, Z_O^t, \omega^t, \phi^t, \kappa^t, x, y) = 0 \tag{28}$$

The unknown parameters are the EOP's of the images and the ground coordinates of the two points defining the line. The constraint in Equation (27) can be applied to all points measured along the line in each scene. As a result, the number of constraints will equal the number of measured points along the line. Again, the ground coordinates of the supplemental points along the straight line are not determined during the bundle adjustment. They only contribute to increase the geometric strength of the adjustment.

The constraint Equation (27) for straight lines in linear array scanner imagery was incorporated into an existing bundle adjustment at The Ohio State University (OSU) to allow for the use of points as well as straight lines in aerial triangulation. This constraint can be applied to frame imagery instead of using the constraint suggested in Chapter 5.1.1.

5.2. NATURAL LINES IN SINGLE PHOTO RESECTION

In digital imagery, edges are represented as discontinuities in gray value. Such edges usually correspond to boundaries in the object space, and therefore often distinguish between different objects. The application of an edge detection operator to a digital image will result in an image of edge pixels. This facilitates the representation of boundary lines as a sequence of pixels. Such lines cannot be properly described by an analytical function. For this reason, we chose to represent natural lines as an aggregation of points.

The objective of single photo resection is to determine the six EOP's ($X_0, Y_0, Z_0, \omega, \phi, \kappa$) associated with an image. In the traditional approach, measurements of distinct points are applied to the collinearity equations (Equation (5)). Since each measured point utilizes two collinearity equations, at least three points are necessary to determine the six EOP's of one image. The introduction of more than three points increases the redundancy and strengthens the solution of the single photo resection.

Since extracted natural lines actually contain a large number of points, we would like to incorporate them into single photo resection. In this research, we have lists of points along natural lines in both image and object space. The image space list contains a sequence of 2D-coordinates, while the object space list contains a list of natural lines and the 3D-coordinates of the points associated with them. This information can be acquired from a GIS database, the GPSVan, or as a result of digitization of existing maps or manuscripts. The correspondence between image and object space lines is not known. This matching problem is solved through a modified version of the generalized Hough transform.

The relationship between conjugate image and object points along a line is given through the collinearity model, as illustrated in Figure 24.

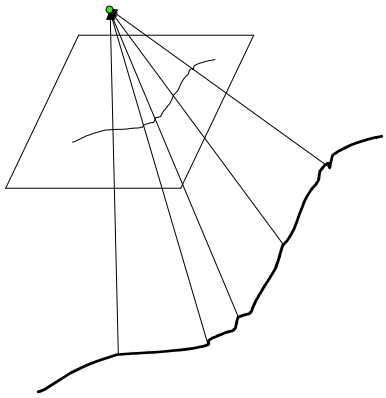


Figure 24: Single photo resection with natural lines

The collinearity model can be applied for all image points:

$$\begin{bmatrix} x_i - x_p \\ y_i - y_p \\ -c \end{bmatrix} = \lambda \; R^T(\omega, \phi, \kappa) \begin{bmatrix} X_I - X_O \\ Y_I - Y_O \\ Z_I - Z_O \end{bmatrix}; \qquad \lambda \ldots \text{ scale} \tag{29}$$

$x_i, y_i, z_{I:}$ image coordinates of i^{th} point

X_I, Y_I, Z_I: object coordinates of i^{th} point

We will examine two approaches for utilizing the modified generalized Hough transform in single photo resection. For both approaches, assume we are given the object coordinates of points along linear features as well as image coordinates corresponding to these points. It is important to emphasize that in these scenarios there is no knowledge of the correspondence between image and object points.

Option 1- Simultaneously solving for the six EOP parameters:

In this technique, an attempted match between three points in the image space with three points in the object space is performed. These three points will allow for six

collinearity equations, which are necessary for simultaneously determining the six EOP's. Utilizing a modified version of the Hough transform, a six dimensional accumulator array corresponding to the six EOP's is incremented by one at the location of each solution. This is done for every possible combination of three image and object points. Given n image points and m object points, the total number of attempted matches performed by this algorithm is: $\frac{n!}{(n-3)!} * \frac{m!}{(m-3)!}$.

Option 2- Solving for two EOP at a time:

In this technique, an attempted match between one point in the image space with one point in the object space is performed. One point allows for two collinearity equations, allowing for the solution of two EOP's at a time. Initial approximations of the remaining four parameters are required. A two-dimensional accumulator array is created for each pair of parameters. The cell size is chosen based on the quality of the initial approximations. For each attempted match, the parameters are solved and the accumulator array is updated. A peak in the accumulator array will indicate the parameter values to be used as approximations for the next iteration. To refine the solution, the cell size of the accumulator array is decreased for each iteration[3]. This is done for every combination of image and object points. Given n image and m object points, the total number of attempted matches performed by this algorithm is: $n * m$.

There is much less computational effort involved in the second technique. Therefore, this technique will now be explained in detail.

Reordering Equation (29) with respect to the ground coordinates and substituting the scale factor λ from the third row into the first two rows of the collinearity model yields Equation (30):

[3] The manipulation of the cell size in the refinement of the solution is analogous to the utilization of image pyramids in matching techniques.

$$X_I - X_0 = \frac{r_{11}(x_i - x_p) + r_{21}(y_i - y_p) + r_{31}(-c)}{r_{13}(x_i - x_p) + r_{23}(y_i - y_p) + r_{33}(-c)}(Z_I - Z_0)$$
$$Y_I - Y_0 = \frac{r_{12}(x_i - x_p) + r_{22}(y_i - y_p) + r_{32}(-c)}{r_{13}(x_i - x_p) + r_{23}(y_i - y_p) + r_{33}(-c)}(Z_I - Z_0) \quad (30)$$

As we are solving for two EOP's at a time, we will assume approximate values for the remaining parameters. Each parameter solution is used to update the approximations. A possible sequence could be for example, to solve for (X_0, Y_0), (Z_0), (ω, ϕ) and (κ) sequentially. The parameter vectors would then be:

$$x_1 = \begin{bmatrix} dX_0 \\ dY_0 \end{bmatrix}, \quad x_2 = [dZ_0], \quad x_3 = \begin{bmatrix} d\omega \\ d\phi \end{bmatrix}, \quad x_4 = [d\kappa] \quad (31)$$

Using this sequence, the modified generalized Hough transform algorithm acts as follows:

Sweep #1:

- Establish the approximations of (Z_0), (ω, ϕ) and (κ).
- Determine the range and the cell size of the accumulator array for (X_0, Y_0), depending on the accuracy of the approximations for the other four parameters.
- Determine x_1 for every point in the object space associated with every point in the image space (using Equation (30)).
- Increment the corresponding element in the accumulator array.
- Select the element with a maximum peak in the array: That particular cell has the most likely values of X_0 and Y_0.

Sweep #2:

- Repeat sweep #1 for (Z_0), (ω, ϕ) and (κ), updating the approximations of the parameters.

Sweep #3:

- Decrease the cell size of the accumulator arrays of (X_0, Y_0), (Z_0), (ω, ϕ) and (κ).
- Repeat sweep #1 – 3 until the iteration converges and all EOP's are fixed.

When using this technique, one can track the indices of the points that lead to a particular solution. In this way, the points associated with a peak in the accumulator array correspond to the conjugate image and object points. Therefore, the matching problem between the natural lines in image and object space is simultaneously solved

5.3. NATURAL LINES IN AUTOMATIC RELATIVE ORIENTATION

Relative orientation establishes the relative relationship between the two images of a stereopair in such a way that it is similar to the relative relationship between the two images at the moment of exposure. The perspective centers associated with the images of a stereopair and a single point on the ground define a plane known as the epipolar plane (Figure 25). Epipolar lines are defined by the intersection of the epipolar plane with the focal planes of the images. The collinearity condition states that the perspective center, an image point and the corresponding object point lie on a straight line. Therefore, the coplanarity constraint confines conjugate points in a stereopair to the epipolar plane. The relative orientation parameters can be determined by using this constraint.

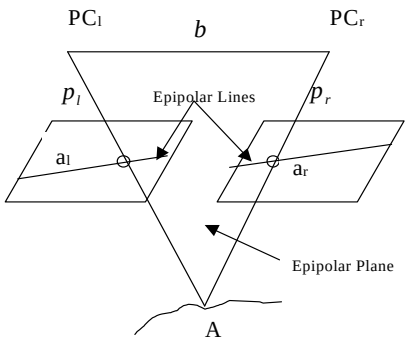


Figure 25: Epipolar geometry of a stereopair

The vector, **b** between the two perspective centers of a stereopair, referred to as the image base, is defined with respect to the ground coordinate system:

$$\vec{b} = \begin{bmatrix} X_{0l} - X_{0r} \\ Y_{0l} - Y_{0r} \\ Z_{0l} - Z_{0r} \end{bmatrix} \tag{32}$$

The vectors $\mathbf{p_l}$ and $\mathbf{p_r}$ from the perspective center to a conjugate point in the left and right images, respectively can be defined with respect to the ground coordinate system as follows:

$$\vec{p}_r = R(\omega_r, \phi_r, \kappa_r) \begin{bmatrix} x_{ar} - x_p \\ y_{ar} - y_p \\ -c \end{bmatrix} \qquad \vec{p}_l = R(\omega_l, \phi_l, \kappa_l) \begin{bmatrix} x_{al} - x_p \\ y_{al} - y_p \\ -c \end{bmatrix} \tag{33}$$

Conjugate light rays intersect in the object space and define an epipolar plane. The coplanarity condition is utilized by observing that the normal vector to the epipolar plane is perpendicular to the image base vector when the stereopair is oriented properly. This condition is defined as follows:

$$(\vec{p}_r \times \vec{p}_l) \bullet \vec{b} = 0 \tag{34}$$

Each pair of conjugate points contributes one coplanarity constraint equation according to Equation (34). At least five conjugate light rays must intersect in object space to define a stereo model. These points are selected at the von Gruber locations. During relative orientation we solve only five out of the twelve EOP's ($X_{0r}, Y_{0r}, Z_{0r}, \omega_r, \phi_r, \kappa_r, X_{0l}, Y_{0l}, Z_{0l}, \omega_l, \phi_l, \kappa_l$). The remaining seven EOP's are later determined through absolute orientation which removes the datum deficiency for that particular stereo model.

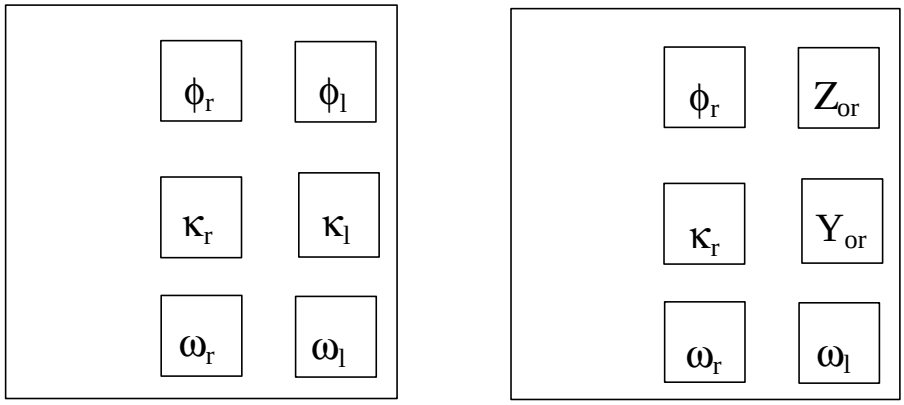
Depending on the choice of parameters to solve in relative orientation, one differentiates between:

- dependent relative orientation: $Y_{0r}, Z_{0r}, \omega_r, \phi_r, \kappa_r$ are unknowns
  $X_{0r}, X_{0l}, Y_{0l}, Z_{0l}, \omega_l, \phi_l, \kappa_l$ are fixed

- independent relative orientation: $\phi_l, \kappa_l, \omega_r, \phi_r, \kappa_r$ are unknowns

  $X_{0r}, Y_{0r}, Z_{0r}, X_{0l}, Y_{0l}, Z_{0l}, \omega_l$ are fixed

The algorithm is well suited for use with extracted image points associated with linear features. In empirical relative orientation, the parameters are determined sequentially by eliminating the y-parallax at the von Gruber locations in a particular sequence. This is due to the fact that some parameters are more sensitive to measurements at certain locations on the image (Figure 26). In our algorithm, the coplanarity condition is applied to image points in a similar way, at one von Gruber location at a time, solving the relative orientation parameters according to this sequence.



Independent RO Dependent RO

Figure 26: Position sensitivity of the unknown parameters during relative orientation

In conventional relative orientation, the user must specify conjugate points in the overlapping images. This is not necessary in automatic relative orientation, where the matching of conjugate points within a stereopair is established through the modified generalized Hough transform.

We will examine two approaches for performing automatic relative orientation that utilize the modified generalized Hough transform.

Option 1- Simultaneously solving for the five relative orientation parameters:

In this technique, an attempted match between five points in the left image with five points in the right image is performed. Using five points will allow for the construction of five coplanarity equations, which are necessary for the simultaneous determination of the ROP's. A five-dimensional accumulator array (corresponding to the ROP's) is incremented by one at the location of the solution. This procedure is repeated for every possible combination of five left image and right image points. Given n points in the left image and m points in the right image, the total number of attempted matches performed by this algorithm is: $\frac{n!}{(n-5)!} * \frac{m!}{(m-5)!}$. The computational effort can be reduced by carefully constraining the candidate matching locations.

Option 2- Solving for one relative orientation parameter at a time:

In this technique, an attempted match between one point in the left image with one point in the right image is performed. One point allows for one coplanarity equation, allowing for the solution of only one ROP at a time. Initial approximations of the remaining five ROP's are required. A one-dimensional accumulator array is utilized for each parameter. Every possible match of conjugate points is evaluated in the determination of each parameter. The accumulator array is updated for each solution. The parameters are calculated in the same sequence as in empirical relative orientation. Each parameter solution is used to update the approximations. To refine the solution, the cell size of the accumulator array is decreased after each iteration. Given n image and m object points, the total number of attempted matches performed by this algorithm is: $n * m$. The implementation of the second option will now be discussed in detail.

The natural lines used in this task must be located in the vicinities of the six von Gruber points of a stereopair. Natural lines are represented as a sequence of 2D-coordinates of image points along the lines. The modified generalized Hough transform algorithm determines conjugate points and consequently conjugate natural lines in the stereopair. The steps for dependent relative orientation are given below. The ROP's are determined with the sequence: Y_{0r}, Z_{0r}, ω_r, ϕ_r, κ_r.

Sweep #1:

- Establish the approximations of $Z_{0r}, \omega_r, \phi_r, \kappa_r$.
- Determine the range and the cell size of the accumulator array for (Y_{0r}), depending on the accuracy of the approximations of the other parameters.
- Solve $Y = Ax_1$, where A is the design matrix according Equation (34). This is done for every point at the proper von Gruber location in the left image associated with every point at the corresponding von Gruber location in the right image. As a result, the relative orientation parameter is determined for every possible match.
- For each evaluated match, the corresponding element in the accumulator array is incremented.
- The element with the maximum peak in the accumulator array has the most likely value of Y_{0r}.

Sweep #2:

- Repeat sweep #1 for $Z_{0r}, \omega_r, \phi_r, \kappa_r$ at their proper von Gruber locations in the two images, updating the approximations of the ROP's from the previous sweep.
- To refine the solution of the parameters, the cell size of the accumulator array can be decreased after each iteration.

By tracking the indices of the points that contribute to the peak, the matching problem of conjugate natural lines is simultaneously solved.

The above algorithms were tested using simulated and real data sets. The results are presented in the next chapter.

6. EXPERIMENTS AND RESULTS

This paper introduced algorithms to incorporate straight lines into aerial triangulation for frame and linear array scanner imagery. In addition, algorithms for performing single photo resection and automatic relative orientation using natural lines were introduced. This chapter will now present the experiments and their results to demonstrate the feasibility of the proposed algorithms.

6.1. AERIAL TRIANGULATION USING STRAIGHT LINES

Existing C and C++ bundle adjustment programs written at the Ohio State University were modified to incorporate straight lines into aerial triangulation for frame and linear array scanner imageries. Simulated and real data were used to test the unified bundle adjustment of points and straight lines, as proposed in Chapter 5.1.

6.1.1. Expected singularities

As outlined in Chapter 5.1, there are two options for including straight lines in aerial triangulation. For the option discussed in Section 5.1.1, we define a vector from the perspective center of an image to a point on the straight line in that image. The underlying principle is to establish the intersection of this vector with the image line planes of the remaining images. If the straight line features are parallel to the flight direction, the image line plane and the vector will be coplanar and will not intersect at a point. A solution will not be possible, as encountered in the following experiment.

6.1.2. Aerial triangulation with simulated data

Figure 27 shows the configuration of image lines in a simulated data set. Control as well as tie points were involved in this experiment. It was found that there is a rank deficiency in the normal equation matrix, due to the fact that the lines ‘a’ and ‘d’ are parallel to the flight direction. The elevation of the four points defining these two lines can not be determined, creating a rank deficiency of four.

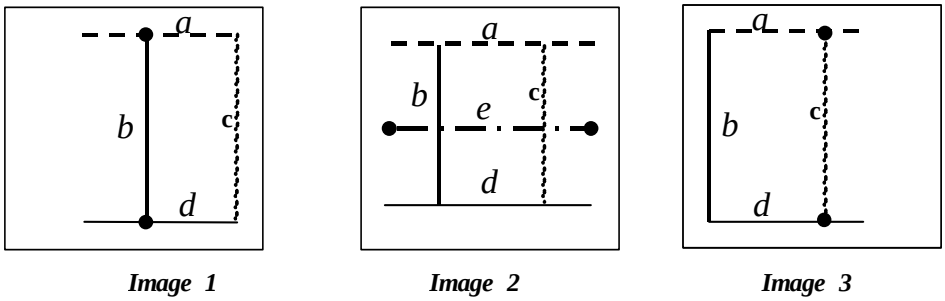


Figure 27: Configuration of image lines using three images

To avoid this problem, two images were added to the configuration, allowing for the necessary intersection (see Figure 28). Repeating this experiment after adding the two images allowed for a solution. The resulting RMS error between the estimated and the true ground coordinates is (0.023m, 0.028m, 0.050m) in the X, Y and Z directions respectively.

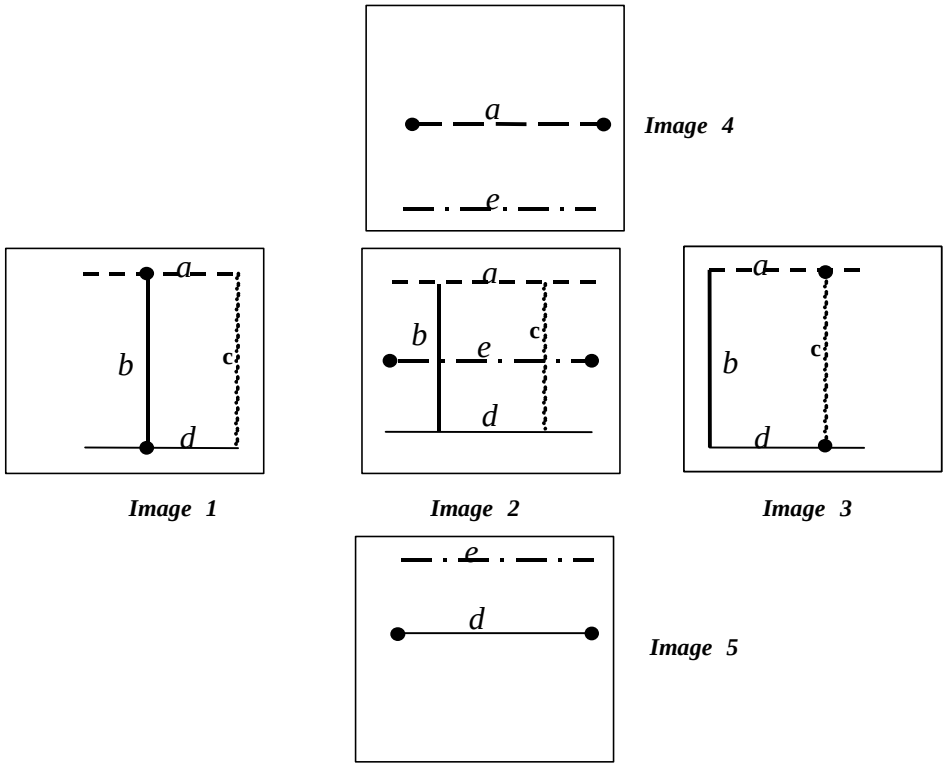


Figure 28: Configuration of image lines using five images

The aerial triangulation algorithm suggested in Chapter 5.1.2. was tested for frame, three-line and panoramic linear array scanner imagery. An overview of the test data configurations is shown in Table 1.

	Number of images	**Number of tie lines**
Frame imagery	12	8
Three-line scanners	6	2
Panoramic linear array scanners	4	7

Table 1: Configuration for bundle adjustment with points and straight lines

For frame imagery, the test field in Figure 29 was used to form a block of twelve images. Eight lines were used in the overlapping area. Aerial triangulation was performed with and without the linear feature constraint. The results are shown in Table 2.

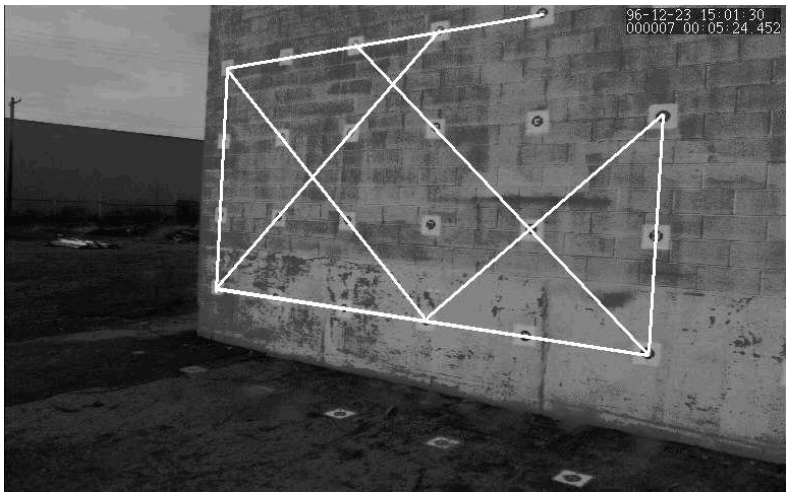


Figure 29: Test field with equally spaced signalized targets

Six images with two tie lines were used to test the algorithm for three-line scanners. Two scenes, each consisting of three images and 60% overlap were used in this experiment. The configuration is shown in Figure 30. GPS information at the exposure stations was utilized in the adjustment. Aerial triangulation was performed with and without the linear feature constraint. The results are shown in Table 3.

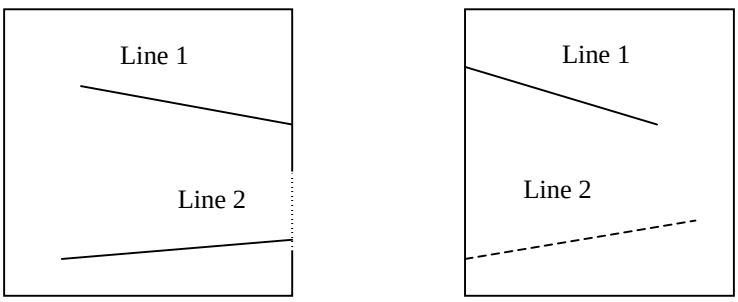


Figure 30: Configuration of three-line scanner imagery with two tie lines

The configuration for panoramic linear array scanner imagery is shown in Figure 31. GPS information at the exposure stations was utilized in the adjustment. Aerial triangulation was performed with and without the linear feature constraint. The results are shown in Table 3.

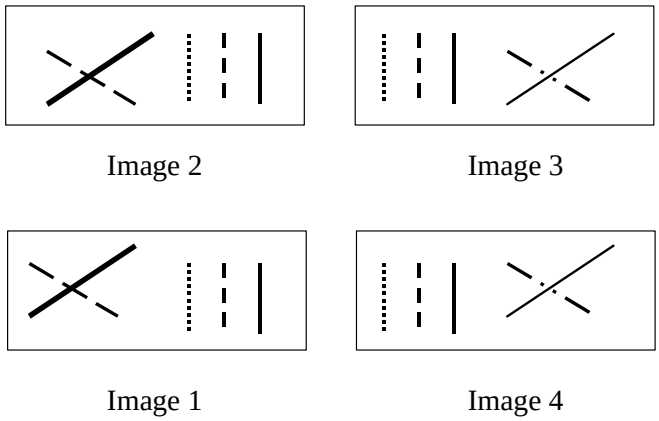


Figure 31: Layout of four panoramic linear array scanner images with seven tie lines

The following processing settings were applied for all test data sets:

- The threshold σ for terminating the iteration process was set to 1.0E-7.
- The correct coordinates of the tie points were disturbed by approximate values of 100m (x, y-components) and 10 m (z-component) to verify their adjustment.

The adjustments were performed with and without the straight-line constraint, using RMS values as a means of comparison: (Table 2 - Table 4).

FRAME IMAGERY	**Without linear features**	**With linear features**
Rms x [m]	0.024	0.040
Rms y [m]	0.031	0.034
Rms z [m]	0.106	0.071

Table 2: Rms-values of the bundle adjustment of frame imagery

THREE-LINE SCANNERS	Without linear features	With linear features
Rms x [m]	2.639	1.688
Rms y [m]	1.292	0.825
Rms z [m]	0.550	0.559

Table 3: Rms-values of the bundle adjustment of three-line scanners

PANORAMIC LINEAR ARRAY SCANNERS	Without linear features	With linear features
Rms x [m]	0.522	0.376
Rms y [m]	0.797	0.300
Rms z [m]	0.935	0.553

Table 4: Rms-values of bundle adjustment of panoramic linear array scanners

As illustrated in the tables, the straight line constraint did not improve the RMS values in the case of frame imagery. However, in the case of linear array scanner imagery, there was a noticeable improvement.

6.1.3. Single photo resection using straight lines

In this experiment, we solved for the EOP's using four straight lines as control. Each line contributes two equations, Equation (20). Thus, there is a redundancy of two. The original, the approximate and the estimated EOP's are shown in Table 5.

	Original EOP	Approximate EOP	Estimated EOP
X_o(m)	-425.00	-575.0	425.008 (±0.049)
Y_o(m)	75.0	-35.0	74.851 (±0.109)
Z_o(m)	1000.0	1170.0	1000.00 (±0.026)
$\omega(^o)$	1.0	11.0	1.07 (±0.005)
$\phi(^o)$	2.0	12.0	1.999 (±0.002)
$\kappa(^o)$	45.0	25.0	44.999 (±0.001)

Table 5: Single photo resection using four control lines

The next experiment involved eight control lines (redundancy of 10). The results of this experiment are shown in Table 6.

	Original EOP	Approximate EOP	Estimated EOP
X_o(m)	-425.00	-575.0	-424.958 (±0.046)
Y_o(m)	75.0	-35.0	74.923 (±0.068)
Z_o(m)	1000.0	1170.0	999.993 (±0.020)
$\omega(^o)$	1.0	11.0	1.004 (±0.003)
$\phi(^o)$	2.0	12.0	2.002 (±0.002)
$\kappa(^o)$	45.0	25.0	45.000 (±0.001)

Table 6: Single photo resection using eight control lines

6.2. SINGLE PHOTO RESECTION USING NATURAL LINES

Single photo resection was performed with natural lines using the modified generalized Hough transform algorithm. An image containing extracted linear features was used in these experiments. The first two experiments demonstrate the algorithm using only one iteration. In other words, the cell size of the accumulator array is not changed. The second two experiments demonstrate the decreasing cell size algorithm, which decreases the cell size of the accumulator array after each iteration.

Single photo resection was performed using natural lines extracted from an image (see Figure 32). Approximately 570 points along the linear features were extracted.

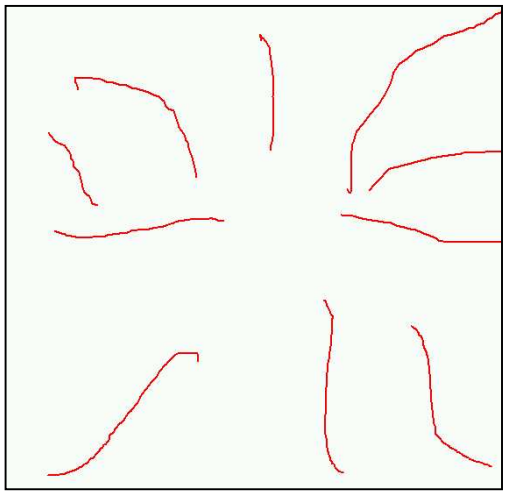
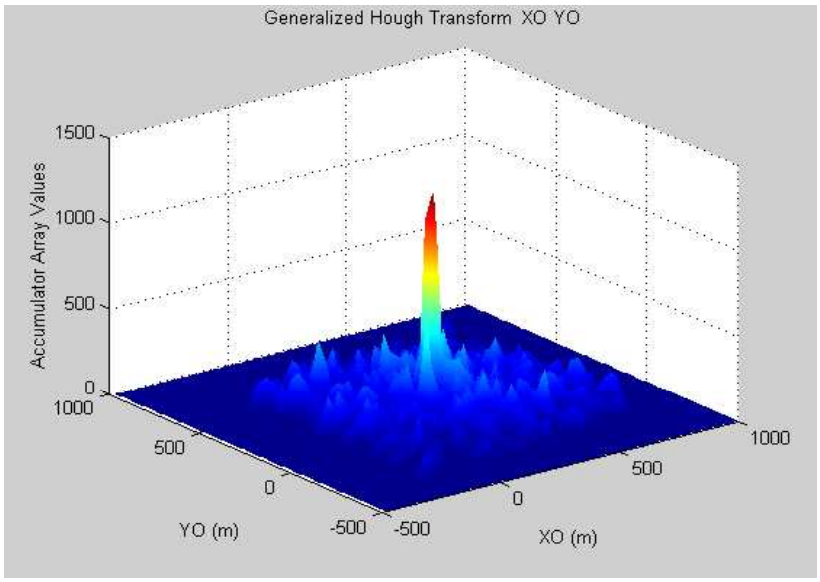


Figure 32: Extracted natural lines used in single photo resection

In the first two experiments, the actual values of the EOP are: $X_0 = Y_0 = Z_o = 250m$ and $\omega = \phi = \kappa = 1^0$. A noise of $\pm 5\mu m$ in image space and $\pm 10cm$ in object space was introduced. The cell size of the accumulator array was chosen according to the quality of the initial approximations. Two experiments were conducted with different approximations and cell sizes.

- $Z_0 = 400m$, $\omega = \phi = \kappa = 3^0$, Cell size = 15m (Figure 33)
- $Z_0 = 500m$, $\omega = \phi = \kappa = 8^0$, Cell size = 40m (Figure 34)

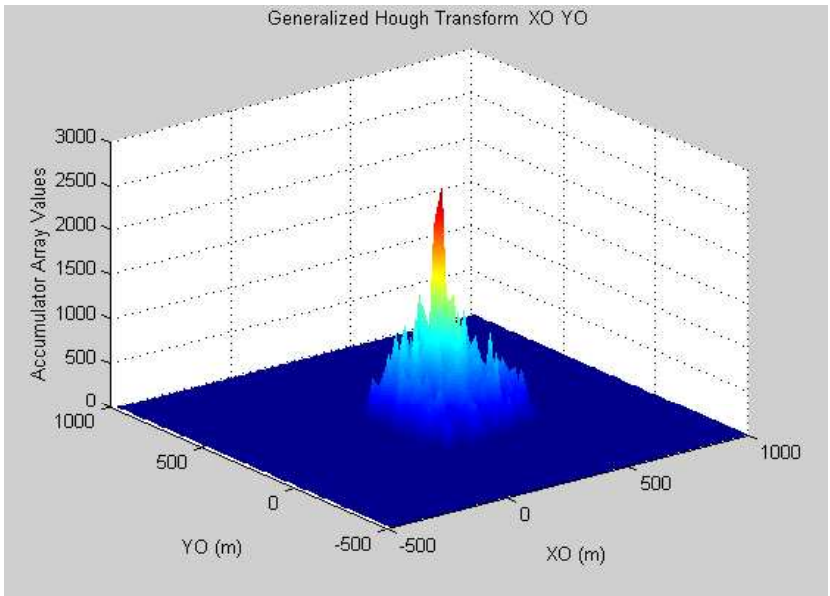
The first sweep of the modified generalized Hough transform was performed for each set of initial approximations. The EOP's, X_0, Y_0, were determined according to the peak of the corresponding accumulator array. The results of this single iteration with constant cell size and different initial values for Z_0, ω, ϕ, κ are given in Figures 33 and 34.



Estimated X_O = 250m

Estimated Y_O = 235m

Figure 33: Accumulator array for pixel size = 15m



Estimated X_O = 250m

Estimated Y_O = 170m

Figure 34: Accumulator array for pixel size = 40m

The next two experiments utilized the decreasing cell size algorithm for the solution of the EOP's (sweeps #1-3 of the modified generalized Hough transform). The settings and results of these experiments are shown in Table 7. In the second experiment, not all points were matched. The remaining points can be matched by local inspection. For this experiment, the convergence of the EOP's is shown graphically in Figure 35.

EXPERIMENT #1	**EXPERIMENT #2**
Actual values for the EOP:	
$(X_O, Y_O, Z_O) = (250, 250, 350)$ m	$(X_O, Y_O, Z_O) = (250, 250, 350)$ m
$(\omega, \phi, \kappa) = (1^o, 1^o, 1^o)$	$(\omega, \phi, \kappa) = (1^o, 1^o, 1^o)$
Assumed values for the EOP:	
$(X_O, Y_O, Z_O) = (400, 400, 550)$ m	$(X_O, Y_O, Z_O) = (400, 400, 550)$ m
$(\omega, \phi, \kappa) = (20.5^o, 20.5^o, 25^o)$	$(\omega, \phi, \kappa) = (20.5^o, 20.5^o, 25^o)$
Pixel size:	
40m & 4^o : 0.5m & 0.01^o	40m & 4^o : 0.5m & 0.01^o
Noise:	
Image space: $\pm 5\mu m$	Image space: $\pm 5\mu m$
Object space: $\pm 10cm$	Object space: $\pm 45cm$
Total number of object points:	
579	458
Total number of image points:	
567	302
Estimated EOP:	
$(X_O, Y_O, Z_O) = (250.01, 250.02, 349.99)$ m	$(X_O, Y_O, Z_O) = (250.27, 249.88, 350.02)$ m
$(\omega, \phi, \kappa) = (0.998^o, 1.0^o, 0.999^o)$	$(\omega, \phi, \kappa) = (1.014^o, 1.037^o, 1.021^o)$
Number of correct matched points:	
All image space points	271 image space points

Table 7: Single photo resection with iterative estimation of the EOP's

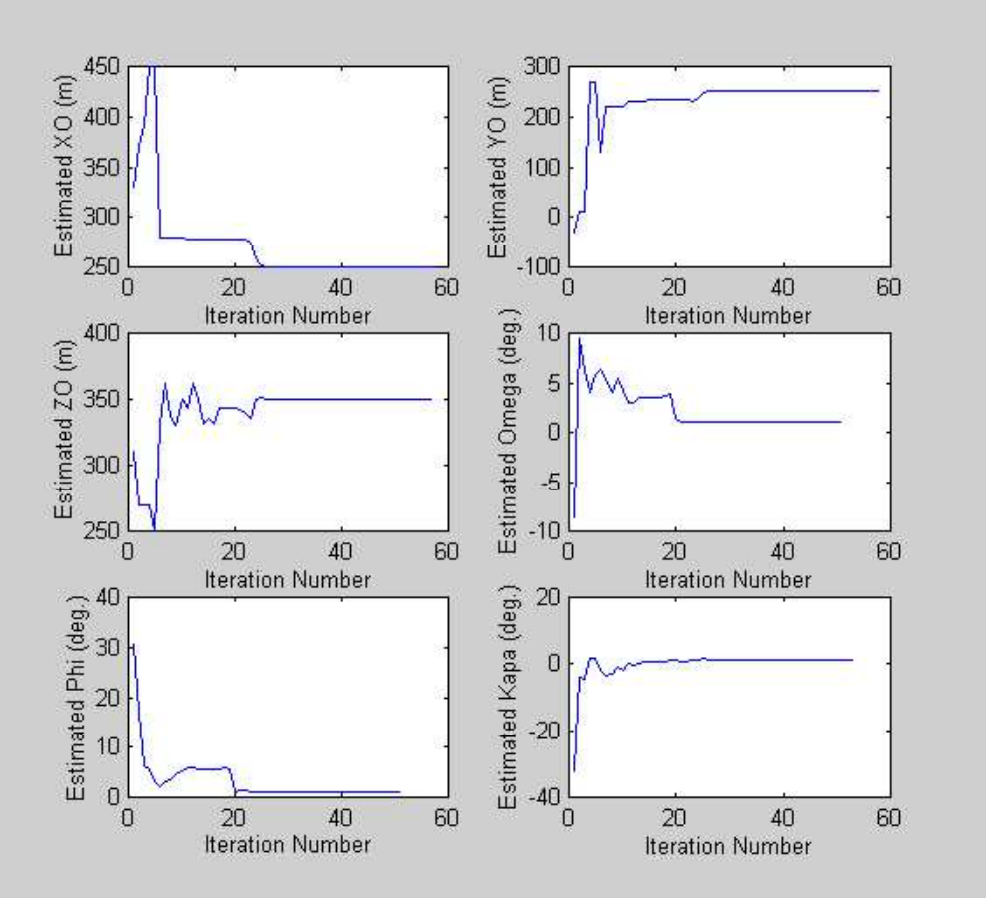
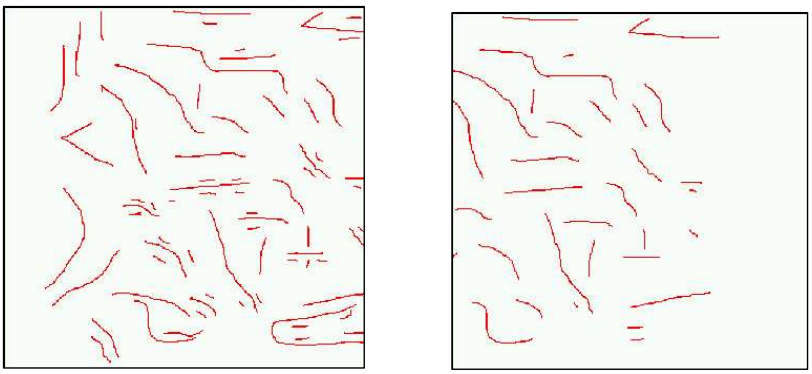


Figure 35: Convergence of the EOP's as a result of decreasing the cell size of the accumulator array

6.2. AUTOMATIC RELATIVE ORIENTATION USING NATURAL LINES

6.3.1. Automatic relative orientation with real data

The automatic relative orientation algorithm of Chapter 5.2.2 was tested using a C-program developed at The Ohio State University. A great number of natural lines were extracted from a stereopair. These lines are represented as a list of 2D-image coordinates. The configuration is shown in Figure 36. The relative orientation technique using the modified generalized Hough transform was performed with the settings and results shown in Table 8.



Left Image Right Image

Figure 36: Extracted natural lines of a stereopair for automatic relative orientation

	Approximated ROP	**Estimated ROP**	**Actual ROP**
$\phi_l(^o)$	19.0^o	0.0^o	0.0^o
$\kappa_l(^o)$	-11.0^o	0.0^o	0.0^o
$\omega_r(^o)$	21.0^o	0.0^o	0.0^o
$\phi_r(^o)$	-25.0^o	0.0^o	0.0^o
$\kappa_r(^o)$	-18.0^o	0.0^o	0.0^o

Table 8: Relative orientation using real data

It should be noted that this automatic relative orientation technique works with linear features as well as with points. Therefore, the incorporation of automatically extracted interest points is straightforward.

6.3.2. Point matching ambiguities

During the matching procedure we are faced with a problem related to the coplanarity model. The epipolar plane intersects the focal planes at the epipolar lines. Conjugate points can be located anywhere on the epipolar lines, still satisfying the coplanarity condition (Equation (34)). As a result, there are multiple solutions for the matching of

conjugate points. Using this algorithm, matched points may be displaced along the epipolar lines (Figure 37). To test this hypothesis, conventional relative orientation was performed using points displaced along the epipolar lines and compared to the solution without displaced points. The ROP's were unaffected by such a displacement. Hence, this point matching ambiguity problem does not affect the solution of the ROP's. The ambiguity problem can be corrected by using additional cues (e.g. area based matching, edge orientation, expected p_x-parallax).

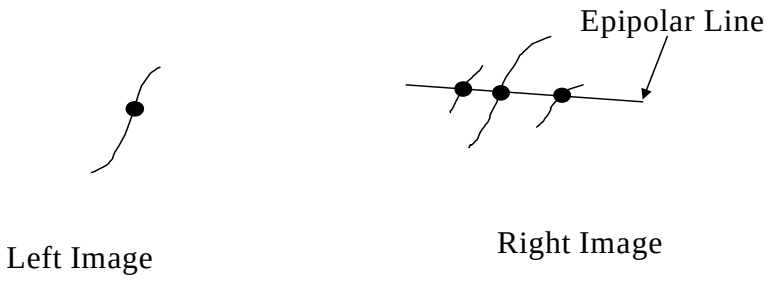


Figure 37: Displacement of conjugate points along epipolar lines

7. CONCLUSIONS AND RECOMMENDATIONS

Linear features can be utilized to provide constraints in photogrammetric applications. Two types of linear features were considered in this research: straight lines and natural lines (free form lines). Straight lines were utilized in aerial triangulation with frame and linear array scanner imageries. Linear features (straight lines and natural lines) were utilized in single photo resection and automatic relative orientation.

7.1. AERIAL TRIANGULATION USING STRAIGHT LINES

For aerial triangulation using straight lines, two options were presented for constraining the solution. For the first option, the object points defining a straight line can be determined by establishing the vector from the perspective center of one image to

an object point on the straight line, and locating the intersection of this vector with the image line planes of the overlapping images. For the second option, the underlying constraint is that the vector passing through the perspective center of an image and an image point on the straight line lies on the plane defined by the perspective center and two object points defining the line. These two object points are considered as tie or control points. This constraint (option two) can accommodate linear array scanner imagery, where straight lines in object space are not necessarily straight lines in image space. In both options, the distinct points defining the straight line in object space need not be present or identifiable in all images.

These algorithms were tested using real and simulated data. It was determined that singularities may occur if the aforementioned intersection point cannot be established due to the orientation of the object space straight line in relation to the flight direction. In this situation, the vector passing through the perspective center and the object point lies on the image line plane, not allowing for an intersection. Real data was used to test this algorithm with frame, panoramic and three line scanner imagery. The adjustments were performed with and without the straight-line constraint, using RMS values as a means of comparison. The results verify the validity of the proposed technique, as the straight-line technique produced comparable, if not better RMS values. In the case of linear array scanner imagery, better results were obtained. This is attributed to the constraints, which help in recovering the EOP's. An overview of the experiments is outlined below:

Option One:

Basic Principle: In the first image, the relationship between the image coordinates and the object coordinates is given by the collinearity condition. For the remaining images, a constraint is introduced that ensures that the two points defining the line in object space belong to the line represented by (ρ_i, θ_i), when transformed into the i^{th} image space.

Representations: The straight line is defined in one image by a pair of image coordinates and is represented in all other images by polar coordinates (ρ, θ).

a) **Using the straight line as a tie line:** In this case, the ground coordinates of the points defining the straight line are unknown.
   Total number of equations per tie line in stereopair = two constraint equations + four collinearity equations = six equations.
b) **Using the straight line as a control line:** In this case, the ground coordinates of the points defining the straight line are known.
   Total number of equations per control line in stereopair is the same as in *a* above.

Adding more equations by considering additional points along the straight line will lead to dependency of equations. However, if camera calibration is included in the aerial triangulation, the additional collinearity equations will be helpful in the calculation of the distortion parameters.

Option Two:

Basic Principle: the vectors from the perspective center to each scene point along the line should lie on the plane that is defined by the perspective center and the two object points defining the straight line.

Representations: The straight line is defined in one image by two points in object space and is represented in other images by image points on the line.

a) **Using the straight line as a tie line:** In this case, the ground coordinates of the points defining the line are unknown.
   Total number of equations per tie line a stereopair = four collinearity equations + two constraint equations = six equations.

b) Using the straight line as a control line: In this case, the ground coordinates of the points defining the line are known.

Total number of equations is the same as in *a* above.

Option two can be used for frame and linear array scanners. In the case of linear array scanner imagery, many EOP's are involved in the adjustment. Added equations will constrain the adjustment and aid in the determination of the EOP's. It is therefore advantageous to evaluate all image points along the straight line.

7.2. SINGLE PHOTO RESECTION WITH NATURAL LINES

A single photo resection algorithm utilizing natural lines has been introduced. For this application, it is assumed that we have the following data available: a list of image coordinates of points along linear features and the 3-D coordinates of these points in the ground coordinate system. The collinearity model is applied to all image points, and the modified generalized Hough transform is utilized to solve for the exterior orientation parameters. By tracking the indices of the points that lead to the solution, the matching of image and object space linear features can be accomplished.

This algorithm has been tested with natural lines using simulated data. The results indicate that this technique produces a high quality solution. The advantage of using this technique over conventional single photo resection is that the correspondence between image and object space points need not be known.

7.3. AUTOMATIC RELATIVE ORIENTATION USING NATURAL LINES

Automatic relative orientation has been implemented utilizing extracted linear features. In its implementation, the coplanarity condition requires that conjugate points lie on the epipolar plane. This constraint is used in conjunction with the modified generalized Hough transform to solve for the relative orientation parameters. By tracking

the indices of the points that lead to a solution (a peak in the accumulator array), conjugate points are identified, hence, linear features are matched. However, due to the geometry of the coplanarity condition, ambiguous matches are possible along epipolar lines. These ambiguities do not affect the solution of the relative orientation parameters. To circumvent this matching problem, area based matching or supplemental cues can be incorporated.

This algorithm was tested using simulated data consisting of extracted natural lines of a stereopair. The results indicate that this technique produces a high quality solution of the relative orientation parameters. Therefore, this technique can utilize point features obtained from the use of interest operators.

7.4. RECOMMENDATIONS FOR FUTURE WORK

Future research will concentrate on the following issues:

- More testing with real data. We would like to use available GIS database and data collected by terrestrial mobile mapping systems, e.g. road networks, to provide control for aerial triangulation.
- Automatic extraction and matching of straight linear features from imagery.
- Generate modules for DEM and Ortho-photo generation based on our approach of automatic relative orientation.
- Automatic selection of triangulation points from imagery (e.g. through interest point operators) to yield an Automatic Aerial Triangulation System.
- Increase the efficiency and speed of the developed algorithms.

8. REFERENCES

Ayache , N., Faugeras, O., 1989. Maintaining representations of the environment of a mobile robot. IEEE Trans. Robotics Automation 5 (6), 804-819.

Ballard, D., Brown, C., 1982. Computer Vision. Department of Computer Science, University of Rochester, Rochester, New York.

Habib, A., 1997. Motion Parameter Estimation by Tracking Stationary Three-Dimensional Straight Lines in Image Sequences. ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 53, pp. 174-182.

Habib, A., Beshah, B., 1997. Modeling Panoramic Linear Array Scanner. Departmental Report No. 443, Geodetic Science and Surveying, Department of Civil and Environment Engineering and Geodetic Science, The Ohio State University, Columbus, Ohio.

Habib, A., Uebbing, R., Novak, K., 1999. Automatic Extraction of Road Signs from Terrestrial Color Imagery

Kubik, K., 1988. Relative and Absolute Orientation Based on Linear Features. . ISPRS Journal of Photogrammetry and Remote Sensing, Vol. 46, pp. 199-204.

Mikhail, E., 1993. Linear Features for Photogrammetric Restitution and Object Completion. SPIE Proceedings – Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision, Volume 1944, Orlando, Fl.

Mulawa, D., and Mikhail, E., 1988. Photogrammetric Treatment of Linear Features. ISPRS 16th Congress, Commission III, Kyoto, Japan.

Roberts, K. 1988. A New Representation for a Line. Proceedings of International Conference on Computer Vision and Pattern Recognition, pp. 635-640.

The Center for mapping, 1991. GPS/Imaging/GIS project. The Ohio State University.

Tommaselli, A., and Tozzi, C., 1992. A Filtering-Based Approach to Eye-In-Hand Robot Vision. ISPRS 17th Congress, Commission V, Washington, DC.

Tommaselli, A., and Tozzi, C., 1996. A Recursive Approach to Space Resection Using Straight Lines. Photogrammetric Engineering and Remote Sensing, Vol. 62, No. 1, pp. 57-66.

Tommaselli, A., and Lugnani, J., 1988. An Alternative Mathematical Model to the Collinearity Equation Using Straight Features. ISPRS 16th Congress, Commission III, Kyoto, Japan.

Wilkin, A., 1992. Robust 3D-Object Representation by Linear Features. ISPRS 17th Congress, Commission V, Washington, DC.

9. APPENDIX 1: MSAT BUNDLE ADJUSTMENT APPLICATION PROGRAM

9.1. MSAT CAPABILITIES

The MSAT (Multi Sensor Aerial Triangulation) is a bundle adjustment software that accommodates for different types of sensors: frame, panoramic and three line scanners. Supplemental data such as GPS/INS can be incorporated. MSAT (version 5) is capable of incorporating a linear feature constraint in the adjustment (see Chapter 5.1.2.). The MSAT program was developed using Microsoft Visual C++. ASCII files are used as input and are described in the help files. Certain input files may not be required for a particular application.

In the MSAT program, all involved parameters can be treated as:

- Constants (not considered as a parameter)
- Parameters with certain accuracy
- Variables (parameters with unknown accuracy)

This is done according to the threshold variances specified in the Project File: (*minCov* and *minCov2*). Any parameter with a dispersion matrix element greater than *minCov* will be treated as variable (unknown accuracy). Parameters with dispersion matrix elements smaller than *minCov2* will be treated as a constant (they will not be involved in the adjustment). If a parameter has a dispersion matrix element lying between the two thresholds then it will be considered as a parameter with certain accuracy (defined by the dispersion matrix). The dispersion of the parameters are to be specified in the GPS File, the Image Coordinate File, the Ground Control Point File, the Linear Feature Coordinate File and the Camera_File.

9.2. MSAT FILE EXAMPLES

Project File (*.prj)

Example:

```
!  1.No of iteration, 2.max sigma, 3.minCov, 4.MinCov2, 5.common alpha,
!  6.GPS and INS availability(0=no,1=yes), 7.linear feature availability(0=no, 1=yes)
10      3E-2   4E-4     5E-12   1   0    0    0    1
```

The above sample input represents a project with no GPS and INS available, and with linear feature constraints. Lines beginning with '!' are comments, and are ignored by the program. This file contains the following parameters (in order):

1. Number of iterations for the bundle adjustment
2. Maximum sigma: the maximum difference between unit weight variances *between each iteration*. If the difference is less than this value, then the iterations will stop.
3. MinCov: MinCov and MinCov2 are best explained in the following paragraph:
4. MinCov2:

   In the MSAT program, all involved parameters can be treated as:

   - Constants (not considered as a parameter)
   - Parameters with certain accuracy
   - Variables (parameters with unknown accuracy)

   This is done according to the threshold variances specified in the project file: (*minCov* and *minCov2*). Any parameter with a dispersion matrix element greater than *minCov* will be treated as variable (unknown accuracy). Parameters with dispersion matrix elements smaller than *minCov2* will be treated as a constant (they will not be involved in the adjustment). If a parameter has a dispersion matrix element lying between the two thresholds then it will be considered as a parameter with certain accuracy (defined by the dispersion matrix).
5. Common alpha: Used in the case of panoramic linear array scanners. This flag should be set to 1 to indicate a common alpha for all photos or each image in panoramic images.

6. GPS/INS Availability: Flag to indicate whether or not GPS and/or INS information at the exposure station is available. If this information is available, one needs to create a GPS File.
7. Linear Feature Availability: Flag to indicate whether or not the straight line constraint will be incorporated into the adjustment. To involve linear features in the adjustment one needs to create a Linear Feature Coordinate File.

GPS File(*.gps)

This file contains the GPS/INS observation and their covariance versus time. The data should be organized in the following format:

- GPS time: time in the orientation image (current implementation depends on each of the orientation images time being in the same time unit as the simplified GPS time)
- Rotation angles: omega, phi, kappa in degrees obtained from the Ins for this time. If INS is not available just input zeros to maintain the format.
- Dispersion of the rotation angles: 3x3 matrix
- Position of perspective center: (Xo, Yo, Zo). Here also if only INS is available then input zeros for Xo, Yo, Zo to maintain the format.
- Dispersion of Xo, Yo, Zo: 3x3 matrix

Example:

```
! GpsTime    omega,     phi,   kappa,   dispersion(opk),    Xo,Yo,Zo, dispersion(Xo,Yo,Zo)
        2.0          -1.64       -1.26      -131.51
1E-6 0.00 0.00
0.00 1E-6 0.00
0.00 0.00 1E-6

    512212.48          215843.73          2933.40
1E-04 0.00 0.00
0.00 1E-04 0.00
0.00 0.00 1E-04
```

```
    2.1        -1.64        -1.26        -131.51
1E-6 0.00 0.00
0.00 1E-6 0.00
0.00 0.00 1E-6

    512205.63          215835.          16 2933.41
1E-04 0.00 0.00
0.00 1E-04 0.00
0.00 0.00 1E-04
```

Image Coordinate File(*.icf)

This file contains the image coordinate measurements. In the current version all the measurements are assumed to be in photo coordinate system for the frame cameras and as raw (row, column) format for the panoramic and three line scanners. The file should contain the data in the following format:

- Photo id: photo in which the point is measured
- Point id
- x: x photo coordinate of the point (or row for digital images)
- y: y photo coordinate of the point (or column for digital images)
- dispersion (2x2 matrix): image coordinate measurement accuracy

Example:

```
! Photo Id     Point Id      x          y              Dispersion
image1         101         -2.991     -102.123        1.0 0.0 0.0 1.0
image1         102         -0.394      3.081          1.0 0.0 0.0 1.0
image3         c1          -9.541      96.288         1.0 0.0 0.0 1.0
```

Ground Control Point File(*.gcp)

This file contains the data for the ground control points and tie points along with their covariance (3x3 matrix). Keep in mind the values of *minCov1* and *minCov2* (specified in the project file) when designating your covariance matrices. The format is:

- Point id
- X, Y, Z: the ground coordinate of the point
- Dispersion (3x3): the dispersion of the point

Example:

!Point id	X	Y	Z	Dispersion
303	6043164.31	2154682.37	44.32	0.01 0.0 0.0 0.0 0.01 0.0 0.0 0.0 0.01
102	6038269.37	2152297.22	80.70	1E+4 0.0 0.0 0.0 1E+4 0.0 0.0 0.0 1E+4

Linear Feature Coordinate(*.lcf) file.

This file consists of

- ID of the photo containing the linear feature
- ID of the linear feature
- ID of the point on the linear feature
- Photo coordinates of the point
- Dispersion matrix
- Flag indicating whether the two identifying points (A and B) of the line are initialized or not in this image
- Flag indicating that point A was initialized in this image
- Flag indicating that point B was initialized in this image

Example:

!Photo id	Linear feature Id	Point ID	Photo coordinates		covariance				Flag	FlagA	FlagB
193	Line1	104	112.5	78.9	1	0	0	1	0	0	0
193	Line1	105	100.5	89.9	1	0	0	1	0	0	0

193	Line1	105	97.9	85.9	1	0	0	1	1	1	0
193	Line1	105	102.5	84.9	1	0	0	1	1	0	1
193	Line2	105	106.5	82.9	1	0	0	1	0	0	0
193	Line2	105	100.5	84.7	1	0	0	1	0	0	0
193	Line2	105	12.5	89.8	1	0	0	1	0	0	0
195	Line1	105	13.5	81.6	1	0	0	1	1	1	0
195	Line1	105	10.5	86.6	1	0	0	1	1	0	1
195	Line1	105	104.5	85.7	1	0	0	1	0	0	0
195	Line1	105	17.5	83.9	1	0	0	1	0	0	0

Camera File (*.cam)

This file contains the cameras involved in the adjustment. For three line scanners, each sensor will be treated as a camera. They should have the names: BACKWARD, FORWARD, DOWN. The data sequence is as follows

- Camera id : camera identification string
- Camera type: FRAME, PANORAMIC, THREELINE
- xp, yp, c: IO parameters
- dispersion of xp, yp and C(3x3 matrix): used to fix or free the camera parameters
- number of fiducials - fiducials: x, y photo coordinates (not used currently)
- number of distortion - distortion parameter value (not used currently)
- GPS antenna offset from the perspective center (dx, dy, dz)
- Gps antenna offset covariance (3x3 matrix)

Example:

```
! Camera id      Type          xp          yp          C
DOWN            THREELINE     -0.003365   0.006972    300.004174
!dispersion of xp, yp and C: (used to fix or free the camera parameters)
1E-6 0.0 0.0
0.0 1E-6 0.0
0.0 0.0 1E-6
!no fiducials fiducials: x, y (not currently used)
0
```

!no distortion and array elements (not currently used)

0

! gps offset: dX, dY, dZ and dispersion

0.0 0.0 0.0

1E-6 0.0 0.0

0.0 1E-6 0.0

0.0 0.0 1E-6

Orientation File (*.ori)

An example of the orientation file:

```
! photoID  camera  sigmaXY  scantime  Degree Opk  Degree Xyz
161         Camera    0.005       0.0      0            0
! alpha coeff and its dispersion
0
! imc coeff and its dispersion
0
! no ori images
1
! ori no  ori time  omega  phi  kappa      X    Y     Z
1     0.0  0.0   0.0   82.146   1821715.816     728473.368  1940.208
! photoID  camera  sigmaXY  scantime  Degree Opk  Degree Xyz
163         Camera    0.005       0.0      0            0
! alpha coeff and its dispersion
0
! imc coeff and its dispersion
0
! no ori images
1
! ori no  ori time  omega  phi  kappa     X     Y     Z
1     0.0  1.0   0.0  85.954  1821789.06  729698.674  1940.208
```

This file contains the following information for each image:

- Photo id
- Sensor type
- Sigma x,y: the image coordinate measurement accuracy
- Scan time: for linear array scanner imagery; the time used to scan the scene.
- Degree OPK, Degree XYZ: For linear array scanner imagery; Degree of the polynomial to model exterior orientation parameters.
- For linear array scanner imagery; The number of coefficients to use to model the scan angle alpha and IMC.
- Number of orientation images
- Orientation time. Should increment for each image in the adjustment.
- Approximations for the exterior orientation parameters.

9.3. MSAT INSTRUCTIONS

The MSAT program requires the use of ASCII files as input data. The structure of these files must be exactly as shown in the examples. For instance, supplemental comments in the files should not be added, and one should pay attention to the carriage returns. The MSAT program does not check for valid files.

To perform the bundle adjustment using MSAT, the first step is to create a Project File. If there is GPS/INS information available at the exposure station, then a GPS File must be included. If the straight line constraint is to be employed, a Linear Feature Coordinate File must be included. All adjustments will require the use of an Image Coordinate File, a Ground Control Point File, and a Camera File.

Once the necessary files are created, click **Bundle Adjustment** on the menu. You are then required to indicate the locations of the involved input files. The user is prompted to designate a file name for the results file (the output file). Next, the bundle adjustment is performed, notifying the user with a message box when the adjustment is complete.

The results of the bundle adjustment are placed in the user defined results file (*.res). The results file contains the following data:

- The estimation of all involved parameters after each iteration
- The dispersion of the parameters and unit weight variance after each iteration.

Also, for interpretation purposes, MSAT (version5) will create a bitmap image of the normal equation martix (for the last iteration). This binary image will designate matrix elements that are zero and non-zero.