A BUNDLE ADJUSTMENT SOLUTION FOR A PERSONAL COMPUTER ENVIRONMENT.

C.L.M. ROMMELAERE

[M.Sc. (Surveying), UCT, 1990]

I hereby declare that this half thesis is
my own work and that it has not been submitted
for a degree at any other university.

Signed by candidate

October 1990 C.L.M. Rommelaere.

a Bundle Adjustment solution for a personal computer environment.

CONTENTS

<u>a Bundle Adjustment solution for a personal computer environment.</u>

INTRODUCTION.

The photogrammetric method.

Photogrammetry is the science and art of using and measuring photographs for various purposes. The word photogrammetry literally means the measuring of something "written" with light.

Various photogrammetric techniques have been developed to solve a wide variety of problems in many fields of endeavour.

The bundle solution is one technique which has proven very useful and versatile.

Bundle adjustment - brief background.

Photogrammetry has developed rapidly with the recent advances in technology. The advent of computers and the personal computer has drastically changed traditional perceptions and the way problems are solved. Faster and large memory capacity computers are now more readily available than in the past. The personal computer commonly known as the PC is fast changing our modern world.

The bundle adjustment solution has been known for over thirty years but has largely been restricted to applications in dedicated analytical photogrammetry systems and mainframes. The scope and versatility of the bundle adjustment solution has now gained more recognition. This half thesis will show that the bundle adjustment solution can be applied in a PC environment. Algorithms are developed for implementation in a full bundle adjustment solution for a personal computer with typical restrictions of RAM and speed limitations.

A bundle adjustment is seen as a full solution encompassing all the data relevant to the solution of the basic photogrammetric problem. Other methods such as block adjustments resort to sequential, partial solutions to the problem, which often rest on mathematical approximations or simplified modelling of the true life situation.

CHAPTER 1

1.1 The bundle solution.

Historically the bundle adjustment solution is a logical conclusion of the basic photogrammetric concepts which evolved since circa 1900. It is a mathematical attempt to model the photographic projective transformation. Analytical solutions gained favour against analogue machines with the advent of the stereocomparator, computers and improved mathematical modelling of least squares solutions. Studies in error theory and error propagation contributed much to our present day acceptance of least squares calculation methods which are applied routinely.

The advent of computer power, the mathematical formulation then known as "The general problem of multiple station analytical stereotriangulation" heralded the technique known today as Bundle Adjustment. Development of this is credited to a number of authors eg. D.C. Brown, H. Schmid and F. Ackermann.

B.G. Müller, A. Grün, E. Gotthardt, S.I. Granshaw and others have also contributed to further research in the bundle adjustment technique. In South Africa research in this field has mainly been done by the department of surveying at the university of Cape Town by L.P. Adams and H. Rüther.

The applications and potential developments involving this technique are wide and may even find a home in a conventional environment traditionally involving theodolites. This application is based on the fact that the theodolite observation can be interpreted as bundles of light rays in space.

The various developments of this technique reported in the literature are based on the same mathematical model but applications thereof often become specialised solutions for the particular problem at hand. This makes it difficult to formulate a specific descriptive definition for the bundle solution of the photogrammetric problem.

Analysis of various publications (by no means exhaustive) suggest the following:

The geometry of the original picture taking is recreated mathematically. This is also done in other solutions, but the difference is that all the photographs here are seen as one simultaneous entity taken into account in the full solution.

The name of the technique implies the use of a bundle of light rays. A bundle of light rays emanating from the object is captured through the photographic process onto an image. Image point position measurements are then used to reproduce the object dimensionally.

The real world geometry of the object being measured is recreated using multiple stations or photo exposures each of which represents a "bundle of light rays" in the mathematical model.

In the case of the bundle solution a least squares adjustment is applied, usually in its most rigorous sense. Other least squares solutions of the photogrammetric restitution problem also employ least squares, but in these cases the least squares model is generally applied sequentially or partially thus resulting in poor modelling and systematic errors distorting the solution.

The bundle solution is an analytical solution mathematically flexible but computationally elaborate as it needs to resolve vast matrices representing simultaneous condition equations. The basic colinearity axiom is used to solve for the unknown values. The linearisation of the condition equations result in an iterative solution.

A particular characteristic and limitation of the bundle adjustment is that good approximations for camera positions and orientations are required.

1.2 Mathematical formulation.

1.2.1 Definition and explanation of photogrammetric terms.

Object space and Image space.

The space outside the camera containing the perspective centre and the object to be measured is known as object space.

The space inside the camera containing the image plane and the principal distance is known as the image space.

Metric and non-Metric cameras.

Specially designed cameras mechanically very stable specifically manufactured for photogrammetry are known as metric cameras. These are calibrated by the manufacturer who supplies values for the principal point position and the principal distance. These values generally remain fixed with time due to the extreme stability of the camera. Very high quality lens systems are used in these cameras with minimal lens distortions.

Non-metric cameras used in photogrammetry are not of such a high stability as metric cameras. These are usually high quality large-format cameras produced by manufacturers such as Rollei, Hasselblat, Mamiya and others. Calibration values for these cameras cannot be considered stable. The lens systems of these cameras usually exhibit large lens distortions.

Fiducial marks and the Fiducial centre.

Fiducial marks are reference marks attached to the corners and/or midpoints of the sides of the image frame of metric cameras. These are sometimes also installed in non-metric cameras. They serve to obtain certain metric parameters directly on the image.

The intersection of the lines connecting opposite fiducial marks is known as the fiducial centre. This is close to the principal point and serves as a reference mark for the principal point. The relation between these two points is supplied with a metric camera.

Principal distance.

The principal distance (PD) of a camera is the computational value used to generate the "best fit" transformation between image distance and object space angles. It is closely related to and often (erroneously) referred to as the focal length of the camera.

The image distance is taken here as the distance measured on the image radially from the fiducial centre to the image point.

The object space angles are the corresponding angles measured in object space at the external perspective centre between the optical axis and the object point ray.

The PD is usually determined as part of a least squares solution of the unknown camera calibration parameters. A change in the PD value results in amending radial distortions.

The principal distance is also known as the camera constant or calibrated focal length of a camera.

Principal point.

The principal point (PP) is the point where a ray entering the camera lens system through the centre of the entrance pupil perpendicular to the image plane intersects with that plane.

This point is usually taken as the reference point for the image space co-ordinate system.

Perspective centre.

The perspective centre, or the centre of projection is the external lens point in space at which all the rays between the object points and associated image points intersect.

In lens systems perspective centres are represented by two points known as the external and internal nodes. In reality the external node of a lens system is taken as the perspective centre.

Interior orientation.

Interior orientation is the geometric definition of camera characteristics. These are the principal distance and the principal point position of a camera.

Relative orientation.

The relative orientation defines the positions of the cameras in a specific photogrammetric setup with respect to each other. The relative orientation of any two cameras is described by any five of the following six elements dX, dY, dZ, dw, do, dk.

The baseline components dX, dY, dZ are the displacements between the perspective centres of the two cameras along the X,Y,Z co-ordinate axes.

The relative orientation angles describe the relative rotations of the two cameras with respect to the X,Y,Z axes. These rotations are denoted as dphi (dw), domega (do), dkappa (dk) in X,Y,Z directions respectively.

The relative orientation does not define the absolute position of the camera in object space.

Absolute orientation.

The absolute orientation positions the camera in object space. The position of the perspective centre and rotation angles for each camera are established in object space. The elements of absolute orientation are Phi (w), Omega (o), Kappa (k), X_O, Y_O, Z_O.

The rotation angle between the camera axes and the object space axes is also defined. This is noted as Phi (w), Omega (o), Kappa (k) for the X,Y,Z axes respectively.

The perspective centre X_O, Y_O, Z_O co-ordinates are established in object space.

The difference between the absolute orientation values of two cameras is the relative orientation. The elements of the interior and the absolute (and by implication the relative) orientations are solved for as part of the bundle solution.

1.2.2 Intuitive grasp of the basic axiom.

It is useful to keep an intuitive description of the bundle solution in mind while developing the rigorous mathematical formulation thereof.

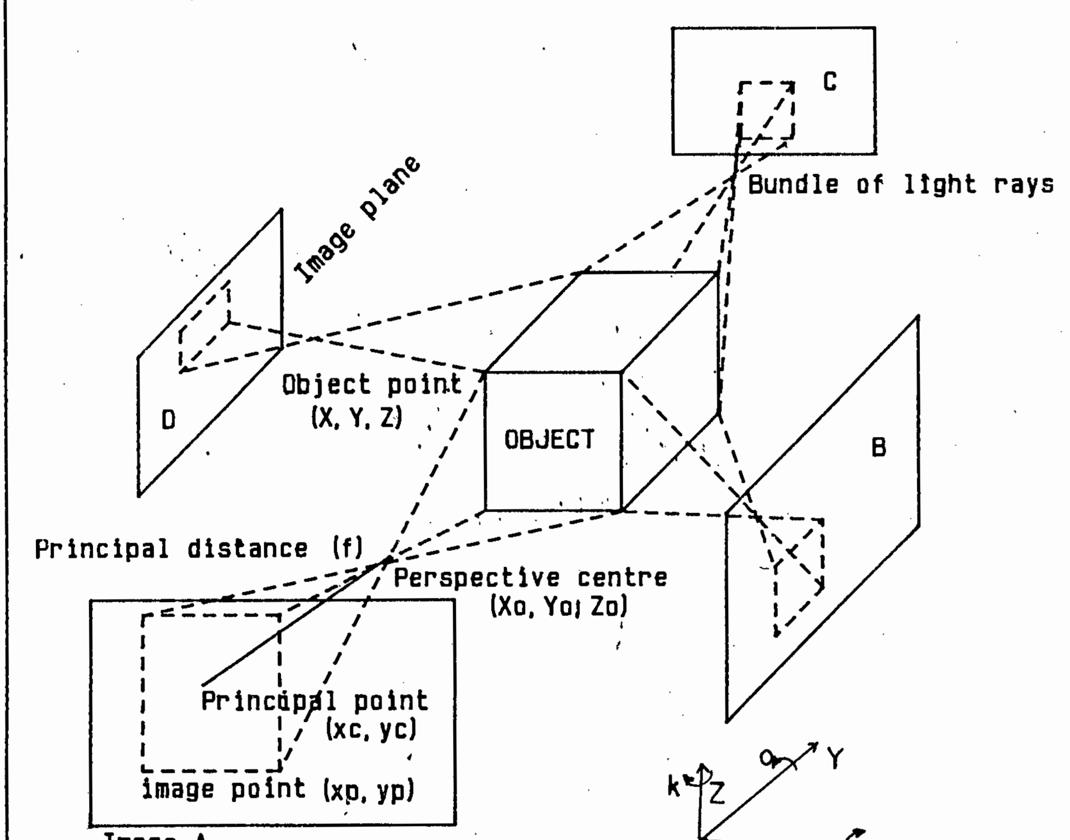
Imagine the taking of a series of pictures from various stations of an object in space. The position of these stations nor that of any or some of the object points need not be known. These pictures are capturing light rays emanating from the object. If enough image points are available it is possible to recreate a mathematical model of the imaged object and thus measure the object dimensions.

The position of a point on the picture is a function of the relative positions of the object point, the perspective centre (camera lens), the camera orientation and the position of the photographic plate. It is obvious that the object point, the perspective centre and the image point are on a light ray (a straight line). Hence every picture can be interpreted as a bundle of light rays through the perspective centre, image point and object point. If one could rearrange these pictures back into their relative but not absolute positions one would be recreating the body (except for scale and position) dimensionally. In order to get actual size at least one distance must be known. The illustrations 1A and 1B attempt to reflect this description.

The basic premise that the perspective centre, image point and object point follow a straight line is an oversimplification of the real life situation. For instance it is well known that light can be refracted, lenses distort the straight line path, the plate used for the photograph may not be a plane. These all have the effect of displacing the image points on the photographs and distorting the collinearity condition. These distortions can be mathematically modelled in terms of the effect they have on the observed plate co-ordinates.

The "images" need not necessarily be taken by a camera since a theodolite observation captures this same light ray. Suffice it to say that one can convert theodolite observations into plate co-ordinates for the purpose of further development of this technique. This is done by artificially creating image (plate) co-ordinates from observed angles.

The observations are either a set of fieldbook azimuth and elevation angles (theodolite) or a set of image (plate) co-ordinates as determined on a comparator. Object points are


C
Bundle of light rays
Image plane
D
Object point
(X, Y, Z)
OBJECT
B
Principal distance (f)
Perspective centre
(Xo, Yo, Zo)
Principal point
(xc, yc)
image point (xp, yp)
Image A
k
Z
Y
X
w

1A

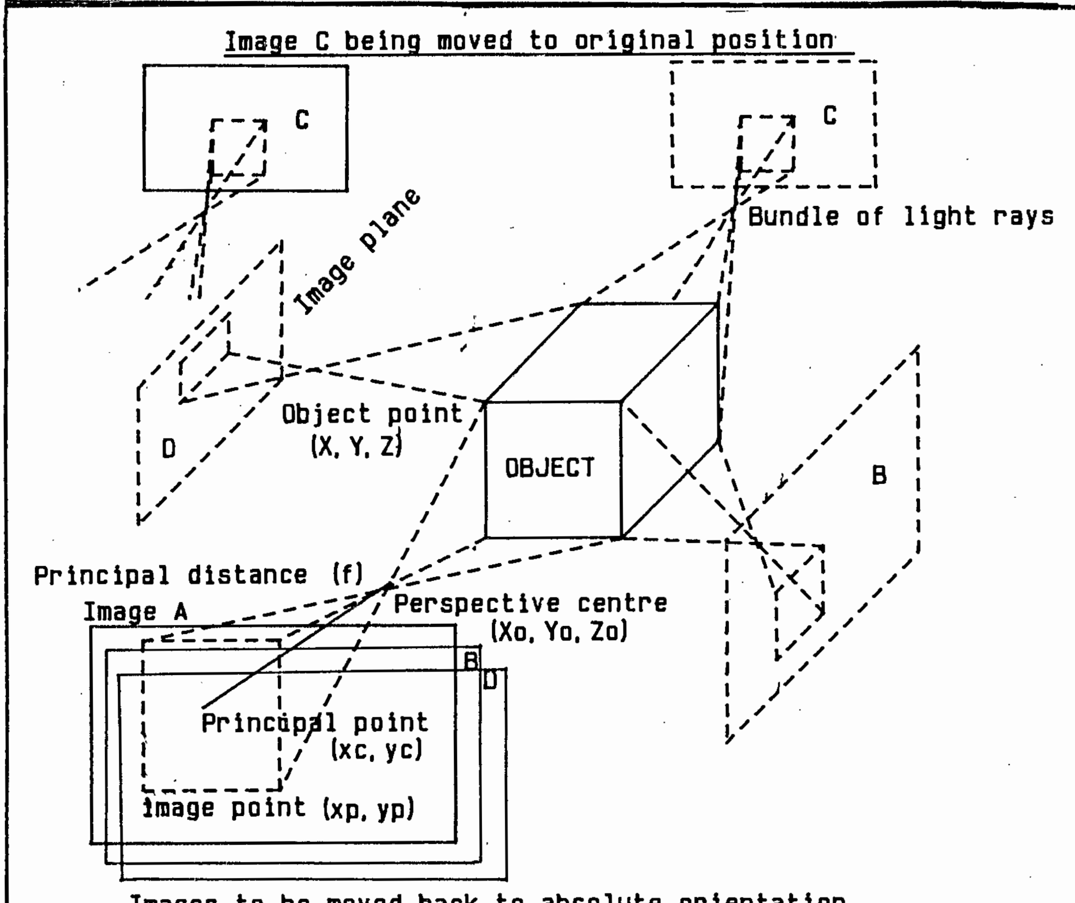


Image C being moved to original position
C
C
Bundle of light rays
Image plane
D
Object point
(X, Y, Z)
OBJECT
B
Principal distance (f)
Image A
Perspective centre
(Xo, Yo, Zo)
B
D
Principal point
(xc, yc)
image point (xp, yp)
Images to be moved back to absolute orientation

1B

defined as X,Y,Z in three dimensional cartesian space. Any combination of object control point co-ordinates can be known. For example a height control point will only have a Z ; a plane ground co-ordinate will only have X,Y or, in some special applications Y,Z or X,Z only are known.

One also needs to establish the perspective centre co-ordinate of the camera (or theodolite) station. This is traditionally referred to as X_o,Y_o,Z_o of image i. This is an unknown co-ordinate in object space.

So, at the minimum a set of images is available. These are used to measure x,y co-ordinates individually on each image of the object points one wishes to recreate in real life. Assume there are i images. Plate co-ordinates of point p in image i are hence referred to as $(x,y)_{pi}$. This basic set of observations requires augmentation by at least one known object space distance for scale. This can be represented in a condition equation in terms of the co-ordinates of the relevant points in object space. Other condition equations can be expressed in terms of known control points on the object, a distance between cameras/stations or a camera to object point distance. Condition equations are also written for object points appearing on more than one image.

It is also necessary to know the parameters of the interior orientation i.e. a relationship between the plate co-ordinates and the perspective centre.

This describes the main points to be resolved.
Usually the unknowns indexed for camera i and point p are;

1. Object points. $(X,Y,Z)_p$
2. Perspective centre points. $(X_o,Y_o,Z_o)_i$
3. Principal distance. (f_i)
4. A set of measurements which by the nature of things are imperfect plate co-ordinates. $(x,y)_p$
5. Principal point on image. (x_c,y_c)

1.3 Rotations and translations.

Imagine the set of photographs stacked like a pack of cards. One plate is kept fixed (basic orientation origin fixed). The bundle adjustment is now trying to move all the other plates about by means of three translations (movements along X,Y,Z) and three rotations (about the axes) to orient and recreate the situation when the set of photographs was taken. In effect this reproduces absolute orientations for all the cameras. These movements are described mathematically in terms of object space co-ordinates as X,Y,Z co-ordinate differences (translations) and an orthogonal rotation matrix for each plate. Traditionally this represents rotations around the primary (usually X), secondary (usually Y) and tertiary (usually Z) axes. Rotations are represented by angles w,o,k respectively around these axes.

A rotation around the X-axis w can be represented by a rotation matrix

$$Rx_w = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos w & -\sin w \\ 0 & \sin w & \cos w \end{vmatrix}$$

Similarly around the Y-axis o:

$$Ry_o = \begin{vmatrix} \cos o & 0 & \sin o \\ 0 & 1 & 0 \\ -\sin o & 0 & \cos o \end{vmatrix}$$

Similarly around the Z-axis k:

$$Rz_k = \begin{vmatrix} \cos k & -\sin k & 0 \\ \sin k & \cos k & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

These previous equations are well known in photogrammetry and need not be elaborated on further.

Multiplication of these individual matrices produce a single combined rotation matrix. The actual final formula depends on the order of rotation applied. Assuming a rotation sequence X, Y, Z one obtains:

$$R = | Rx_w \cdot Ry_o \cdot Rz_k |$$

$$R = \begin{vmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{vmatrix}$$

With

$$R_{11} = \cos o . \cos k$$

$$R_{12} = -\cos o . \sin k$$

$$R_{13} = \sin o$$

$$R_{21} = \sin w . \sin o . \cos k + \cos w . \sin k$$

$$R_{22} = -\sin w . \sin o . \sin k + \cos w . \cos k$$

$$R_{23} = -\sin w . \cos o$$

$$R_{31} = -\cos w . \sin o . \cos k + \sin w . \sin k$$

$$R_{32} = \cos w . \sin o . \sin k + \sin w . \cos k$$

$$R_{33} = \cos w . \cos o$$

Equation 1.1

The full matrix R is the traditional rotation matrix expressed in terms of trigonometric functions.

1.4 Orthogonality in rotation matrices.

Any orthogonal matrix of dimensions 3*3 can be used to describe a three dimensional rotation. This matrix need not contain trigonometric functions. In a sense the trigonometric terms of the matrix R in equation 1.1 are replaced by convenient mathematical unknowns.

The rotation matrix has to be orthogonal since the orthogonality of the object space must be preserved i.e. no distortions of the object are mathematically introduced. It is necessary to prove under which conditions a rotation matrix is orthogonal. The rotation matrix and orthogonality will be used in the mathematical development of the bundle solution.

The following proofs are relevant:

Definitions.

The inner product of vectors X and Y is defined as

$$XY = X^TY = Y^TX$$

The length of a vector X is defined as the square root of the inner product of X and X^T ;

$$|X| = SQR(X^TX)$$

By this is meant that the square root of the sum of the squares of the elements of X is calculated.

Note.

Two vectors X and Y (using inner products) can be written as;

$$XY = 1/2 \{|X+Y|^2 - |X|^2 - |Y|^2\}$$

Proof.

A linear transformation preserves lengths if and only if its transformation matrix is orthogonal:

Taking two vectors X_1 X_2 and their image vectors Y_1 Y_2 under a linear transformation

$$Y_1 = AX_1 \text{ and } Y_2 = AX_2$$

Suppose A is an orthogonal matrix so that

$$A^T A = I \text{ (identity matrix)}$$

then by using inner products of vectors it can be stated that

$$Y_1 \; Y_2 = Y^T{}_1 \; Y_2 = (X^T{}_1 \; A^T)(A \; X_2) = X^T{}_1 \; I \; X_2 = X_1 \; X_2$$

In terms of the lengths of these vectors (using inner products) one can write:

$$X_1 \; X_2 = 1/2 \; \{|X_1+X_2|^2 - |X_1|^2 - |X_2|^2\}$$

and

$$Y_1 \; Y_2 = 1/2 \; \{|Y_1+Y_2|^2 - |Y_1|^2 - |Y_2|^2\}$$

Comparing these two equations left and right sides it is clear that the lengths will only be preserved if inner products are preserved under a linear transformation $Y=AX$

Hence supposing lengths are preserved (which is what is needed in this case) it then follows that

$$Y_1 \; Y_2 = X^T{}_1 (A^T A) X_2 = X^T{}_1 \; X_2 \quad \text{where } A^T A = I$$

Thus proving rotation matrix A must be orthogonal. This then guarantees preservation of vector lengths.

Getting back to the R rotation matrix it is expedient to use a rotation matrix without trigonometric functions (this cuts down on computing time).The Cayley formula, well established in photogrammetry, states briefly that if S is a skew-symmetric matrix then an orthogonal rotation matrix R can be written in terms of a skew-symmetric matrix as follows

$$R = (I+S)(I-S)^{-1} \tag{1.2}$$

If S contains 3 rows and 3 columns (i.e. of the order 3*3) it is written as;

$$S = \begin{vmatrix} 0 & -c & +b \\ +c & 0 & -a \\ -b & +a & 0 \end{vmatrix} \tag{1.3}$$

This yields the orthogonal matrix:

$$R = \frac{1}{1+a^2+b^2+c^2} \cdot \begin{vmatrix} 1+a^2-b^2-c^2 & 2ab-2c & 2ac+2b \\ 2ab+2c & 1-a^2+b^2-c^2 & 2bc-2a \\ 2ac-2b & 2bc+2a & 1-a^2-b^2+c^2 \end{vmatrix}$$

Equation 1.4

Without any trigonometrical functions! Unknowns in these rotations are a,b,c as opposed to rotation angles w,o,k mentioned earlier.

These rather long winded equations (1.4) can be abbreviated as

$$R = \frac{1}{D} \cdot \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \qquad (1.5)$$

1.5 Cayley's formula (Rodrigues terms) and trigonometric function

It is useful to have a connection between the trigonometric functions in terms of w,o,k and Cayley's parameters a,b,c. Equating equivalent terms in equations 1.1 and 1.4 it immediately follows that

Using R_{13}

$$o = \arcsin \frac{2ac+2b}{1+a^2+b^2+c^2} \tag{1.6}$$

Using R_{12} and equation 1.6

$$k = \arcsin \{- \frac{1}{\cos o} \cdot \frac{2ab-2c}{1+a^2+b^2+c^2}\} \tag{1.7}$$

Using R_{23} and equation 1.6

$$w = \arcsin \{- \frac{1}{\cos o} \cdot \frac{2bc-2a}{1+a^2+b^2+c^2}\} \tag{1.8}$$

These equations are useful to derive orientation data for the cameras. Angular rotations are shown here in terms of Rodrigues parameters. Intrinsically the trigonometric function contains angular ambiguities. The solutions to equations 1.6 to 1.8 are not unique. Various values of w,o,k would satisfy the above equations.

It may also be necessary to solve for a,b,c in terms of w,o,k. This may be used to get approximate values for Cayley's parameters from known rotations. This can be done by using ;

$a_{21} - a_{12}$ terms:

$$\frac{4c}{1+a^2+b^2+c^2} = \text{sinw.sino.cosk} + \text{cosw.sink} + \text{coso.sink} \qquad (1.9)$$

$a_{32} - a_{23}$ terms:

$$\frac{4a}{1+a^2+b^2+c^2} = \text{cosw.sino.sink} + \text{sinw.cosk} + \text{sinw.coso} \qquad (1.10)$$

$a_{13} - a_{31}$ terms:

$$\frac{4b}{1+a^2+b^2+c^2} = \text{sino} + \text{cosw.sino.cosk} - \text{sinw.sink} \qquad (1.11)$$

Note however that $D=1+a^2+b^2+c^2$ appears in all these equations. It is a fixed value in terms of the Cayley parameters for each solution. D can be ignored for approximation purposes and one simply accepts the above formulae as a direct ratio between a,b,c and w,o,k in terms of trigonometrical functions.

By adding the terms along the main diagonal of equation 1.4 and with some manipulation thereof one obtains

$$\frac{-\ (\ a_{11}+a_{22}+a_{33}\ -4\ )}{D} = \frac{1+a^2+b^2+c^2}{1+a^2+b^2+c^2} = 1 \qquad (1.12)$$

therefore

$$D = -\ (\ a_{11}+a_{22}+a_{33}\ -4\ )$$

By using the equivalent terms of equation 1.1 for the a_{ii} terms of equation 1.4 one also obtains:

$$-\cos o.\cos k + \sin w.\sin o.\sin k - \cos w.\cos k - \cos w.\cos o + 4 = 1$$

Equation 1.13

These latter equations can serve as checks on the results obtained from a bundle adjustment computation.

1.6 Equations modelling the photogrammetric problem.

It was shown earlier that the light ray which produces the picture on the photograph follows a straight line from the object through the perspective centre onto the photographic plate.This is known as the co-linearity condition written mathematically as

$$\begin{vmatrix} x-x_c \\ y-y_c \\ f \end{vmatrix} = s.R \begin{vmatrix} X-X_o \\ Y-Y_o \\ Z-Z_o \end{vmatrix} \quad \begin{matrix} (1.14.1) \\ (1.14.2) \\ (1.14.3) \end{matrix}$$

Equations 1.14

with s.R representing a scale and rotation matrix as defined. Formulating this in conventional notation (for point p image i) by dividing equation 1.14.1 by 1.14.3 and 1.14.2 by 1.14.3 one obtains;

$$\frac{X_p - X_{oi}}{Z_p - Z_{oi}} = \frac{(x_p - x_{ci})\,A_1 + (y_p - y_{ci})\,A_2 + f_i.D}{(x_p - x_{ci})\,C_1 + (y_p - y_{ci})\,C_2 + f_i.F} \qquad (1.15)$$

and

$$\frac{Y_p - Y_{oi}}{Z_p - Z_{oi}} = \frac{(x_p - x_{ci})\,B_1 + (y_p - y_{ci})\,B_2 + f_i.E}{(x_p - x_{ci})\,C_1 + (y_p - y_{ci})\,C_2 + f_i.F} \qquad (1.16)$$

respectively

where

X_p ,Y_p ,Z_p are object space co-ordinates of point p.

X_{oi},Y_{oi},Z_{oi} are perspective centre co-ordinates in object space of image i

x_p ,y_p are image co-ordinates of point p measured on the photograph

x_{ci},y_{ci} are co-ordinates of the principal point of image i.

f_i is the principal distance of image i.

A_1, A_2, B_1, B_2, C_1, C_2, D, E, F — are the elements of a rotation matrix with scale factor s.R

s.R in equation 1.14 can be written as

$$s.R = \begin{vmatrix} A_1 & B_1 & C_1 \\ A_2 & B_2 & C_2 \\ D & E & F \end{vmatrix} \qquad (1.17)$$

These equations represent a transformation of plate co-ordinates to object space co-ordinates. It is however more convenient for the formation of the least squares condition equations to have all the elements on the one side of the equation and hence this is rewritten as follows:

$$x_c - x \quad + \quad f. \frac{A_1(X-X_o) + B_1(Y-Y_o) + C_1(Z-Z_o)}{D(X-X_o) + E(Y-Y_o) + F(Z-Z_o)} = 0 \qquad (1.18)$$

and

$$y_c - y \quad + \quad f. \frac{A_2(X-X_o) + B_2(Y-Y_o) + C_2(Z-Z_o)}{D(X-X_o) + E(Y-Y_o) + F(Z-Z_o)} = 0 \qquad (1.19)$$

The unknowns are;

the object space co-ordinates $X_p, Y_p, Z_p, X_o, Y_o, Z_o$

the inner orientation parameters x_c, y_c, f

the rotation elements $A_1, A_2, B_1, B_2, C_1, C_2, D, E, F$

The rotation elements of R are correlated and represent three independent unknowns. In some cases some of these elements may be known eg. a control point may have established X,Y,Z values.

On each photograph image co-ordinates x_{pi}, y_{pi} are measured based on an arbitrary photo origin. The mathematical effect of choosing an arbitrary origin is a translation with respect to x_c, y_c. This does not affect the solution for unknown object space co-ordinates and rotations.

The condition equation for point p on image i can therefore be written as

$$-x_{pi} + f_i \cdot \frac{A_{1i}(X_p - X_{oi}) + B_{1i}(Y_p - Y_{oi}) + C_{1i}(Z_p - Z_{oi})}{D_i(X_p - X_{oi}) + E_i(Y_p - Y_{oi}) + F_i(Z_p - Z_{oi})} = 0$$

and

$$-y_{pi} + f_i \cdot \frac{A_{2i}(X_p - X_{oi}) + B_{2i}(Y_p - Y_{oi}) + C_{2i}(Z_p - Z_{oi})}{D_i(X_p - X_{oi}) + E_i(Y_p - Y_{oi}) + F_i(Z_p - Z_{oi})} = 0$$

Equations 1.20

Rotation matrices and Cayley's formula are now directly applied to these equations. The above can be rewritten using the earlier Cayley elements by simple substitution of corresponding elements in these equations ;

As a function in x (Fx)

$$Fx = -x_{p_i} + f_i \cdot \frac{a_{11_i}(X_p - X_{o_i}) + a_{12_i}(Y_p - Y_{o_i}) + a_{13_i}(Z_p - Z_{o_i})}{a_{31_i}(X_p - X_{o_i}) + a_{32_i}(Y_p - Y_{o_i}) + a_{33_i}(Z_p - Z_{o_i})} = 0$$

and as a function in y (Fy)

$$Fy = -y_{p_i} + f_i \cdot \frac{a_{21_i}(X_p - X_{o_i}) + a_{22_i}(Y_p - Y_{o_i}) + a_{23_i}(Z_p - Z_{o_i})}{a_{31_i}(X_p - X_{\ _i}) + a_{32_i}(Y_p - Y_{o_i}) + a_{33_i}(Z_p - Z_{o_i})} = 0$$

Equations 1.21

This is written in matrix form:

$$\begin{vmatrix} x_{p_i} \\ y_{p_i} \\ f_i \end{vmatrix} = s \cdot \begin{vmatrix} a_{11_i} & a_{12_i} & a_{13_i} \\ a_{21_i} & a_{22_i} & a_{23_i} \\ a_{31_i} & a_{32_i} & a_{33_i} \end{vmatrix} \cdot \begin{vmatrix} X_p - X_{o_i} \\ Y_p - Y_{o_i} \\ Z_p - Z_{o_i} \end{vmatrix}$$

Equation 1.22

s in this equation signifies a scale factor which is eliminated as mentioned earlier. This equation is the full equation including the rotation matrix as shown in short in equation 1.14

1.6.1 The least squares solution.

It is expedient to simplify the nomenclature in these equations. The following terms are now introduced:

$$A1p_i = a_{11_i}(X_p - X_{o_i}) + a_{12_i}(Y_p - Y_{o_i}) + a_{13_i}(Z_p - Z_{o_i})$$

$$A2p_i = a_{21_i}(X_p - X_{o_i}) + a_{22_i}(Y_p - Y_{o_i}) + a_{23_i}(Z_p - Z_{o_i})$$

$$A3p_i = a_{31_i}(X_p - X_{o_i}) + a_{32_i}(Y_p - Y_{o_i}) + a_{33_i}(Z_p - Z_{o_i})$$

Equation 1.23

Every observed ray onto an object point yields a pair of equations such as equations 1.21. Since these were taken simultaneously as a set of observations it now needs to be solved as a simultaneous set of condition equations. Since observations are not perfect it is necessary to apply the principles of least squares adjustment and weight each observation realistically. Least squares solutions require linear equations. The above equations 1.21 need to be linearised using Taylor's theory and differentiation.

Taylor's theorem states that a function F(x+h) can be expressed as a linear series expansion ;

$$f(x+h) = f(x) + h.f'(x) + \frac{h^2}{2!}.f''(x) + \frac{h^3}{3!}.f'''(x) + .. + \frac{h^n}{n!}.f^n(x+\theta h)$$

with $(0< \theta <1)$

or for f(x,y);

$$f(x+h,y+k) = f(x,y) + \{hf_x(x,y)+kf_y(x,y)\} + \frac{1}{2!}\{h^2f_{xx}(x,y) + 2hkf_{xy}(x,y) + k^2f_{yy}(x,y)\} + ...$$

with f_x ,f_y ,f_{xx} ,f_{xy} ,f_{yy} denoting partial differentiation

Equation 1.24

This theorem is applied on the non-linear condition equations to linearise these using the first order terms only of the Taylor series. The least squares axiom is then superimposed on this and solved by matrix algebra etc. It is however necessary to iterate this process because of the initial Taylorisation of the non-linear equations introduced in the solution. The iteration procedure converges towards the true values.

The Rodrigues parameters derived above in equation 1.4 have now also been introduced directly in order to fully carry out the requisite partial differentiation denoted here as

$$\frac{\delta Fx}{\delta x} \qquad \text{(first partial derivative } f_x\text{)} \quad \text{(function Fx from eqn. 1.21)}$$

(note that a_i, b_i, c_i here are the Rodrigues parameters for photo i and $a_{11}, a_{12}, a_{13}, a_{21}, a_{22}, a_{23}, a_{31}, a_{32}, a_{33}$ are combinations of Rodrigues terms as defined above)

$$\frac{\delta Fx}{\delta x} = -1$$

$$\frac{\delta Fx}{\delta y} = 0$$

$$\frac{\delta Fx}{\delta a_i} = \frac{2f_i}{A3_{pi}{}^2} * [\{a_i(X_p-X_{oi}) + b_i(Y_p-Y_{oi}) + c_i(Z_p-Z_{oi})\}.A3_{pi} \ldots$$
$$\ldots\ldots - \{c_i(X_p-X_{oi}) + (Y_p-Y_{oi}) - a_i(Z_p-Z_{oi})\}.A1_{pi}]$$

$$\frac{\delta Fx}{\delta b_i} = \frac{2f_i}{A3_{pi}{}^2} * [\{-b_i(X_p-X_{oi}) + a_i(Y_p-Y_{oi}) + (Z_p-Z_{oi})\}.A3_{pi} \ldots$$
$$\ldots\ldots - \{-(X_p-X_{oi}) + c_i(Y_p-Y_{oi}) - b_i(Z_p-Z_{oi})\}.A1_{pi}]$$

$$\frac{\delta Fx}{\delta c_i} = \frac{2f_i}{A3_{pi}{}^2} * [\{-c_i(X_p-X_{oi}) - (Y_p-Y_{oi}) + a_i(Z_p-Z_{oi})\}.A3_{pi} \ldots$$
$$\ldots\ldots - \{a_i(X_p-X_{oi}) + b_i(Y_p-Y_{oi}) + c_i(Z_p-Z_{oi})\}.A1_{pi}]$$

$$\frac{\delta Fx}{\delta X_{oi}} = \frac{-\delta Fx}{\delta X_p} = \frac{f_i}{A3_{pi}{}^2} * (-a_{11}A3_{pi} + a_{31}A1_{pi})$$

$$\frac{\delta Fx}{\delta Y_{oi}} = \frac{-\delta Fx}{\delta Y_p} = \frac{f_i}{A3_{pi}{}^2} * (-a_{12}A3_{pi} + a_{32}A1_{pi})$$

$$\frac{\delta Fx}{\delta Z_{oi}} = \frac{-\delta Fx}{\delta Z_p} = \frac{f_i}{A3_{pi}{}^2} * (-a_{13}A3_{pi} + a_{33}A1_{pi})$$

Equations 1.25

$$\frac{\delta Fy}{\delta x} = 0$$

$$\frac{\delta Fy}{\delta y} = -1$$

$$\frac{\delta Fy}{\delta a_i} = \frac{2f_i}{A3_{pi}{}^2} * [\{b_i(X_p-X_{oi}) - a_i(Y_p-Y_{oi}) - (Z_p-Z_{oi})\}.A3_{pi} \ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots -\{c_i(X_p-X_{oi}) + (Y_p-Y_{oi}) - a_i(Z_p-Z_{oi})\}.A2_{pi}]$$

$$\frac{\delta Fy}{\delta b_i} = \frac{2f_i}{A3_{pi}{}^2} * [\{ a_i(X_p-X_{oi}) + b_i(Y_p-Y_{oi}) + c_i(Z_p-Z_{oi})\}.A3_{pi} \ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots -\{-(X_p-X_{oi}) + c_i(Y_p-Y_{oi}) - b_i(Z_p-Z_{oi})\}.A2_{pi}]$$

$$\frac{\delta Fy}{\delta c_i} = \frac{2f_i}{A3_{pi}{}^2} * [\{ (X_p-X_{oi}) - c_i(Y_p-Y_{oi}) + b_i(Z_p-Z_{oi})\}.A3_{pi} \ldots$$
$$\ldots\ldots\ldots\ldots\ldots\ldots -\{a_i(X_p-X_{oi}) + b_i(Y_p-Y_{oi}) + c_i(Z_p-Z_{oi})\}.A2_{pi}]$$

$$\frac{\delta Fy}{\delta X_{oi}} = \frac{-\delta Fy}{\delta X_p} = \frac{f_i}{A3_{pi}{}^2} * (-a_{21}A3_{pi} + a_{31}A2_{pi})$$

$$\frac{\delta Fy}{\delta Y_{oi}} = \frac{-\delta Fy}{\delta Y_p} = \frac{f_i}{A3_{pi}{}^2} * (-a_{22}A3_{pi} + a_{32}A2_{pi})$$

$$\frac{\delta Fy}{\delta Z_{oi}} = \frac{-\delta Fy}{\delta Z_p} = \frac{f_i}{A3_{pi}{}^2} * (-a_{23}A3_{pi} + a_{33}A2_{pi})$$

Equations 1.26

Taylor's theorem requires a first approximation or absolute term [eg. f(x,y) above].This is combined with the image observations to yield the following:

$$lx = f_i . A1_{pi} / A3_{pi} - x_{pi}$$
$$ly = f_i . A2_{pi} / A3_{pi} - y_{pi}$$

Equation 1.27

These are absolute terms calculated using a first approximation for;

X_{pi}, Y_{pi}, Z_{pi}

X_{oi}, Y_{oi}, Z_{oi}

Rodrigues terms (a, b, c)

f_i and

x_{pi}, y_{pi} values.

1.6.2 Weighting of observations and unknowns.

Consider the weights or relative accuracies of observed plate co-ordinates for each image and even within a particular image. The simplest solution is to accept all measurements to be equal and thus of the same weight (say unit weight). This is however not a good model of the real world. A better idea is to consider that each pair of observations has a specific weight coefficient matrix for point p image i. In practice a co-variance σx and σy to each observed value (x,y) is allocated. The co-variance matrix is then written as

$$Qv = Qxy_{pi} = \begin{vmatrix} Qxx & Qxy \\ Qxy & Qyy \end{vmatrix} = c. \begin{vmatrix} mx^2 & mxy \\ mxy & my^2 \end{vmatrix} \quad (1.28)$$

The control points on the object can also be weighted individually to create a more realistic model. It is suggested here that each control point could be allocated a different weight in keeping with a priori relative accuracies for each control point. A priori values for control points could for instance be obtained from a prior network adjustment when the control was initially established. This will result in keeping these points more stable in the adjustment. Otherwise a generalised weight matrix for any object point can be written as follows:

$$Pu^{-1} = Qu = Qxyz_p = \begin{vmatrix} Qxx & Qxy & Qxz \\ Qyx & Qyy & Qyz \\ Qzx & Qzy & Qzz \end{vmatrix} =c. \begin{vmatrix} mx^2 & mxy & mxz \\ myx & my^2 & myz \\ mzx & mzy & mz^2 \end{vmatrix}$$

Equation 1.29

The following derivation uses weighting of individual points independent from observations and combines all the parameters. This is based on derivations described by Haag.

The Taylor linearisation of the observation equations are split in groups to distinguish between the type of corrections being sought. Thus the linearised observation equation can be written as

$$-v + Au + B\underline{x} + C\underline{y} + l = 0 \qquad (1.30)$$

with covariance matrix as

$$Q = \begin{vmatrix} Qv & 0 \\ 0 & Qu \end{vmatrix}$$

with

v vector of corrections to image co-ordinate observations (x_{pi}, y_{pi}) with covariance matrix Qv

u vector of corrections to the control points (these must appear in more than one image) with covariance matrix Qu

$\underline{x}$ vector of unknown new object point co-ordinates

$\underline{y}$ vector of camera orientation parameters

l vector representing the absolute terms

A,B,C the coefficient matrices relating to the various vectors $u,\underline{x},\underline{y}$ respectively.

The least squares solution requires that the square of the corrections is to be a minimum hence;

$$v^T Q_v^{-1} v + u^T Q_u^{-1} u - 2k^T (-v + A^T u + B\underline{x} + C\underline{y} + l) = \text{minimum}$$

k is correlation vector.

Equation 1.31

This minimum requirement is obtained by equating the partial derivatives of the variables to 0

yielding for dv : $v^T Q_v^{-1} + k^T = 0$ (1.32)

du : $u^T Q_u^{-1} - k^T A^T = 0$ (1.33)

dk^T: $-v + A^T u + B\underline{x} + C\underline{y} + l = 0$ (1.34)

$d\underline{x}$: $k^T B = 0$ (1.35)

$d\underline{y}$: $k^T C = 0$ (1.36)

hence solving for (1.32)

and for (1.33)

$$v = -Q_v . k \qquad (1.37)$$

$$u = Q_u . A . k \qquad (1.38)$$

substituting equations (1.37) and (1.38) in (1.34) above yields

$$(Qv + A^T Q_u A)k + B\underline{x} + C\underline{y} + l = 0 \qquad (1.39)$$

which resolves to

$$k = -(Q_v + A^T Q_u A)^{-1} . (B\underline{x} + C\underline{y} + l) \qquad (1.40)$$

and

$$B^T k = 0 \qquad (1.41)$$

$$C^T k = 0 \qquad (1.42)$$

By substituting k into equations 1.41 and 1.42 there are two equations left. **This gives the solution in terms of the co-ordinates $\underline{x}$ and the orientation elements $\underline{y}$.**

1.6.3 Sparse and diagonal matrices.

Sparse matrices are matrices with many elements being zero. A square matrix containing only non-zero elements along its main diagonal is called a diagonal matrix. Some of the elements along this diagonal may however also be zero. Such matrices as described here have useful properties which simplify matrix algebra in a bundle solution. This results in reducing computing time.

It is beyond the objective of this half thesis to study this vast subject in depth here. Some of the properties of such matrices can be used to advantage in this solution.

Vectors are used to introduce some of the properties mentioned above. The structure of the inner product of a pair of orthogonal vectors with elements equal to zero is of specific significance.

Taking two column vectors a and b each with n elements where at least one of each of the corresponding element i in a and/or b equals 0 (i.e. $a_i=0$ and/or $b_i=0$).

For example ;

$$a = \begin{vmatrix} 0 \\ 0 \\ 4 \end{vmatrix} \qquad b = \begin{vmatrix} 0 \\ 3 \\ 0 \end{vmatrix}$$

It follows that $a^T b = b^T a = 0$ thus proving a and b orthogonal to each other.

This idea expands to a matrix C where in each column only one element is not equal to zero, or all the elements equal zero. This is typically the case in diagonal and sparse matrices. It follows from this that each row vector of this matrix C is orthogonal to the others. Now $F = C\, C^T$ involves the sum of the products of each row vector c_i yielding element d_{ij} thus

$$d_{ij} = c_i^T\, c_j \qquad (1.43)$$

Since these vectors are mutually orthogonal it is clear that for rows i unequal to j one obtains $d_{ij} = 0$. The elements of d_{ii} (ie. i equals j) are $c_i\, c_i^T$. This is the sum of the square of the elements of row i. This latter is not equal to zero unless the whole row vector is a zero vector. It follows that the new matrix F will become a diagonal matrix. A further multiplication with another diagonal matrix (D say, eg. $E = C.D.C^T$) also still yields a diagonal matrix.

Consider the above equations 1.40 to 1.42. These contain the matrix $(Qv + A^T Q_u A)^{-1}$. If one chooses the structure of Qv and Qu as diagonal matrices and the coefficients of A can also be kept mutually orthogonal as discussed above, one would end up with a diagonal matrix for $(Qv + A^T Q_u A)^{-1}$.

This makes it trivial to invert within the calculations for a solution. Matrices of this format can always be set up for uncorrelated unknowns or points.

CHAPTER 2

2 Bundle adjustment - Specific equations for programming.

2.1 Computer hardware.

An attempt is now made to use these derivations in a personal computer (hereafter referred to as PC). The PC environment is undergoing such fast advances in technology that its capacity is expanding continually at a moderate cost to the practitioner. The PC used for this program is an IBM compatible AT machine with 20 megabyte hard disc, 1 megabyte RAM capacity and a maths co-processor. The programming language used was True BASIC.

2.2 Setting up the basic equations.

Three types of corrections are applied.

a) **X** vector of corrections to image plate co-ordinates $(x,y)_{pi}$ with co-variance matrix $\mathbf{Q_X}$.

b) **Y** vector of corrections to the object space points $(X,Y,Z)_p$ with co-variance matrix $\mathbf{Q_Y}$.

c) **Z** vector of corrections to the orientation values of the images otherwise perceived as the translations and rotations necessary to bring each picture mathematically back to where it was when the image was captured on film. This involves the Rodrigues parameters, perspective centre co-ordinates, principal distances and additional parameters (AP) for each image i. The co-variance matrix $\mathbf{Q_Z}$ is associated with this.

The problem is now viewed in a broader context involving vectors. Some of these vectors contain individual co-ordinates as elements.

Note that this choice of grouping makes use of the fact that no direct correlation exists between the sets of corrections. It also logically groups the sets of parameters for which the normal equation system needs to be developed. This means that each set can be entered in the computer independently from the other thus making it easier to amend the data. This data will typically be introduced in the main solution as matrices. It can also be read in as raw data by various means depending on the application. For example image co-ordinates could typically be read in by a comparator electronically linked to the computer, punched in manually or read in from a data storage device obtained from a different source.

This grouping of corrections allows the choice, to directly solve for the object points or the image parameters. The solution can be obtained directly in terms of either set of unknowns. The other unknowns are then solved indirectly by substitution in appropriate formulae.

It is for this reason that all the observations (image plate values $(x,y)_{pi}$) of a particular point are grouped together. This is referred to as a **point group**. All image observations of point 4 are one point group i.e. $(x,y)_{41}$,$(x,y)_{43}$,$(x,y)_{49}$ form a group if point 4 only appears on images 1,3 and 9.

Also all points observed on a particular image form an **image group**. This is used in a separate matrix. This is generally entered naturally as a consequence of the image values being observed one plate at a time (for example in comparator observations).

The number of correlation terms k will be equal to the number of condition equations which equal the number of image observations, one for x and a separate one for y (see equations 1.20 for example). The X vector referred to in a) above will contain the number of x,y image plate co-ordinates measured for each image occurrence of each object point. The Y vector referred to in b) above will contain three times (3*) the number of object points, one for each co-ordinate X,Y,Z. The Z vector referred to in c) above will typically contain six times(6*) the number of images (ni).

Generally speaking in a typical application there will be much less images than object points which in turn will be less than the number of image observations. It is expedient to eliminate the image corrections and the correlation terms (ie. k and X) and then the object point Y parameters for a direct solution. This will reduce the calculations and number crunching in the computer tremendously. It is for this reason that the observation equations are sorted in point groups as described earlier.

The Taylor linearisation of the basic observation equations (eqn.1.20 above) yields in a matrix context;

$$-X + AY + BZ + l = 0 \qquad \text{Co-variance matrix } Q \qquad (2.1)$$

The coefficients A,B are the partial derivatives (equations 1.25 and 1.26). The values for l are derived as per equations.1.27 above.

This new equation is a specific form of equation 1.30. Substitute X for vector v with covariance submatrix Q_X. Vector u and $\underline{x}$ are combined to form Y with covarionce matrix Q_Y. Vector y is replaced by vector Z with its co-variance matrix Q_Z. Vector l is as defined previously. It is necessary to change to this vector notation due to the broader more general basis on which this derivation rests.

The grouping of observations is now done as follows:

Point groups are all image observations of a particular point. These are indexed per point nunber say p. The above matrix equation 2.1 is now made up of n groups (total number of object points = n) indexed p.

Image observations of point p are double indexed that is indexing for point p which appears on image i is written as pi.

This way of grouping results in n groups of equations which together make up a matrix set containing the orientation correction vector Z. For example for 6 points six matrix equations involving the vector Z will be obtained.

$$- \begin{vmatrix} X_1 \\ X_2 \\ \mathbf{X_3} \\ X_4 \\ X_5 \\ X_6 \end{vmatrix} + \begin{vmatrix} A_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & A_2 & 0 & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{0} & \mathbf{A_3} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & 0 & A_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & A_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & A_6 \end{vmatrix} . \begin{vmatrix} Y_1 \\ Y_2 \\ \mathbf{Y_3} \\ Y_4 \\ Y_5 \\ Y_6 \end{vmatrix} + \begin{vmatrix} B_1 \\ B_2 \\ \mathbf{B_3} \\ B_4 \\ B_5 \\ B_6 \end{vmatrix} .\mathbf{Z} + \begin{vmatrix} l_1 \\ l_2 \\ \mathbf{l_3} \\ l_4 \\ l_5 \\ l_6 \end{vmatrix} = \mathbf{0} \qquad (2.2)$$

The one line of this matrix equation has been highlighted to further illustrate the exact make-up of this matrix equation system. This basic set of equations is broken down further. Point p (3 say) observed on all ni images (1 to 4 say) which is noted in equation 2.2 as;

$$- |\mathbf{X_3}| + |0\ 0\ \mathbf{A_3}\ 0\ 0\ 0| . |\mathbf{Y_3}| + |\mathbf{B_3}| . \mathbf{Z} + |\mathbf{l_3}| = 0$$

becomes

$$-\begin{vmatrix} X_{31} \\ \mathbf{X_{32}} \\ X_{33} \\ X_{34} \end{vmatrix} + \begin{vmatrix} A_{31} \\ \mathbf{A_{32}} \\ A_{33} \\ A_{34} \end{vmatrix} . \mathbf{Y_3} + \begin{vmatrix} B_{31} & 0 & 0 & 0 \\ \mathbf{0} & \mathbf{B_{32}} & \mathbf{0} & \mathbf{0} \\ 0 & 0 & B_{33} & 0 \\ 0 & 0 & 0 & B_{34} \end{vmatrix} . \begin{vmatrix} Z_1 \\ \mathbf{Z_2} \\ Z_3 \\ Z_4 \end{vmatrix} + \begin{vmatrix} l_{31} \\ \mathbf{l_{32}} \\ l_{33} \\ l_{34} \end{vmatrix} = 0 \qquad (2.3)$$

The image plate observations $(x,y)_{32}$ of point 3 on image 2 is in itself made up of the basic two linearised observation equations derived from equations 1.20 thus:

$$-|\mathbf{X_{32}}| + |\mathbf{A_{32}}| . \mathbf{Y_3} + |0\ \ \mathbf{B_{32}}\ 0\ \ 0| . |\mathbf{Z_2}| + |\mathbf{l_{32}}| = 0$$

becomes

$$-|x_{32}| + \begin{vmatrix} \frac{\delta Fx}{\delta X_3} & \frac{\delta Fx}{\delta Y_3} & \frac{\delta Fx}{\delta Z_3} \end{vmatrix} . \begin{vmatrix} X_3 \\ Y_3 \\ Z_3 \end{vmatrix} + \begin{vmatrix} \frac{\delta Fx}{\delta a} & \frac{\delta Fx}{\delta b} & \frac{\delta Fx}{\delta c} & \frac{\delta Fx}{\delta Xo} & \frac{\delta Fx}{\delta Yo} & \frac{\delta Fx}{\delta Zo} \end{vmatrix}_2 . \begin{vmatrix} a \\ b \\ c \\ Xo \\ Yo \\ Zo \end{vmatrix}_2 + |lx_{32}| = 0$$

Equation 2.4

and

$$-|y_{32}| + \begin{vmatrix} \frac{\delta Fy}{\delta X_3} & \frac{\delta Fy}{\delta Y_3} & \frac{\delta Fy}{\delta Z_3} \end{vmatrix} . \begin{vmatrix} X_3 \\ Y_3 \\ Z_3 \end{vmatrix} + \begin{vmatrix} \frac{\delta Fy}{\delta a} & \frac{\delta Fy}{\delta b} & \frac{\delta Fy}{\delta c} & \frac{\delta Fy}{\delta Xo} & \frac{\delta Fy}{\delta Yo} & \frac{\delta Fy}{\delta Zo} \end{vmatrix}_2 . \begin{vmatrix} a \\ b \\ c \\ Xo \\ Yo \\ Zo \end{vmatrix}_2 + |ly_{32}| = 0$$

Equation 2.5

This is written in general terms for point p and image i;

$$-|x_{pi}|+\begin{vmatrix}\frac{\delta Fx}{\delta X_p} & \frac{\delta Fx}{\delta Y_p} & \frac{\delta Fx}{\delta Z_p}\end{vmatrix}\cdot\begin{vmatrix}X_p\\Y_p\\Z_p\end{vmatrix}+\begin{vmatrix}\frac{\delta Fx}{\delta a} & \frac{\delta Fx}{\delta b} & \frac{\delta Fx}{\delta c} & \frac{\delta Fx}{\delta Xo} & \frac{\delta Fx}{\delta Yo} & \frac{\delta Fx}{\delta Zo}\end{vmatrix}_i\cdot\begin{vmatrix}a\\b\\c\\Xo\\Yo\\Zo\end{vmatrix}_i+|lx_{pi}|=0$$

Equation 2.6

and

$$-|y_{pi}|+\begin{vmatrix}\frac{\delta Fy}{\delta X_p} & \frac{\delta Fy}{\delta Y_p} & \frac{\delta Fy}{\delta Z_p}\end{vmatrix}\cdot\begin{vmatrix}X_p\\Y_p\\Z_p\end{vmatrix}+\begin{vmatrix}\frac{\delta Fy}{\delta a} & \frac{\delta Fy}{\delta b} & \frac{\delta Fy}{\delta c} & \frac{\delta Fy}{\delta Xo} & \frac{\delta Fy}{\delta Yo} & \frac{\delta Fy}{\delta Zo}\end{vmatrix}_i\cdot\begin{vmatrix}a\\b\\c\\Xo\\Yo\\Zo\end{vmatrix}_i+|ly_{pi}|=0$$

Equation 2.7

The co-variance matrix Q referred to in equation 2.1 is made up of the three submatrices Q_X, Q_Y, Q_Z denoted above.

Q_X is the co-variance matrix for the image plate observations (x,y) which are assumed not to be correlated. No correlation exists between observed image points. This results in Q_X being a diagonal matrix with dimensions 2 rows and 2 columns (denoted as 2*2) submatrices along the main diagonal. This choice is specifically made to obtain a diagonal matrix. The advantage of such a matrix has been discussed earlier in section 1.6.3.

Q_Y is the co-variance matrix for the object space points. For the purpose of this derivation it is accepted that there are no correlations between the object space points either. This results in a diagonal co-variance matrix with submatrices of dimensions 3*3 along the main diagonal.

Similarly provided no correlation exists between the camera stations the Q_Z co-variance matrix results in a diagonal matrix comprising submatrices of dimension 6*6 along the main diagonal.

There exist no direct correlations between the image co-ordinates in the image space, the object space point co-ordinates and the orientation elements of each camera. This means that the co-variance matrix can be written as a diagonal matrix comprising Q_X, Q_Y, Q_Z along the main diagonal. The other terms of the main co-variance matrix Q will equal 0. This matrix can be written as

$$Q = \begin{vmatrix} Q_X & 0 & 0 \\ 0 & Q_Y & 0 \\ 0 & 0 & Q_Z \end{vmatrix} \qquad (2.8)$$

The example for points 3 and 4 on images 2 and 6 may elucidate this further. Q_X for points 3 and 4 together will be diagonal as shown below

$$Q_X = \begin{vmatrix} Q_{X2} & 0 \\ 0 & Q_{X6} \end{vmatrix}$$

This is detailed further for example for image 2 containing points 3 and 4;

$$Q_{X2} = \begin{vmatrix} Q_{X32} & 0 \\ 0 & Q_{X42} \end{vmatrix}$$

which extends for each point to;

$$Q_{X32} = \begin{vmatrix} Q_{xx} & Q_{xy} \\ Q_{xy} & Q_{yy} \end{vmatrix}_3$$

and

$$Q_{X42} = \begin{vmatrix} Q_{xx} & Q_{xy} \\ Q_{xy} & Q_{yy} \end{vmatrix}_4$$

Similarly Q_Y can be written as

$$Q_Y = \begin{vmatrix} \mathbf{Q_{Y3}} & 0 \\ 0 & Q_{Y4} \end{vmatrix}$$

with Q_{Y3} for point 3 being;

$$Q_{Y3} = \begin{vmatrix} Q_{XX} & Q_{XY} & Q_{XZ} \\ Q_{YX} & Q_{YY} & Q_{YZ} \\ Q_{ZX} & Q_{ZY} & Q_{ZZ} \end{vmatrix}_3$$

Matrix Q_Z also combines to form a set of submatrices as follows.

$$Q_Z = \begin{vmatrix} Q_{Z2} & 0 \\ 0 & Q_{Z6} \end{vmatrix}$$

with Q_{Z6} for image 6 being;

$$Q_{Z6} = \begin{vmatrix} Q_{aa} & Q_{ab} & Q_{ac} & Q_{aXo} & Q_{aYo} & Q_{aZo} \\ Q_{ba} & Q_{bb} & Q_{bc} & Q_{bXo} & Q_{bYo} & Q_{bZo} \\ Q_{ca} & Q_{cb} & Q_{cc} & Q_{cXo} & Q_{cYo} & Q_{cZo} \\ Q_{Xoa} & Q_{Xob} & Q_{Xoc} & Q_{XoXo} & Q_{XoYo} & Q_{XoZo} \\ Q_{Yoa} & Q_{Yob} & Q_{Yoc} & Q_{YoXo} & Q_{YoYo} & Q_{YoZo} \\ Q_{Zoa} & Q_{Zob} & Q_{Zoc} & Q_{ZoXo} & Q_{ZoYo} & Q_{ZoZo} \end{vmatrix}_6$$

Co-variance matrices Q_{X32}, Q_{Y3}, Q_{Z6} for example have been written in a more general form showing the various parameters which influence each element of the most basic submatrix. It is clear that usually only the elements along the main diagonals of each of these submatrices will have a value and the others will be null. This is due to the assumption that there are also no correlations between the parameters. For example a and Xo, X_3 and Y_3 and x_4 and y_4 respectively, are uncorrelated.

The unknown parameters are also usually not given any weight at all. This means that many of the terms in these co-variance matrices even along the diagonals will equal 0. It is advisable to weigh only the known points and values as realistically as possible. This ensures results which are compatible with the true situation in the object space.

In section 1.6.2 it was shown that the squares of the corrections is to be a minimum. This is done by combining equations 2.1 and 2.8 yielding the general condition that;

$$X^TQ_X{}^{-1}X + Y^TQ_Y{}^{-1}Y + Z^TQ_Z{}^{-1}Z - 2k^T(-X + AY + BZ + 1) = \text{a minimum}$$

Equation 2.9

with k representing the correlation vector as is customary in a least squares solution.

The derivatives of this formula are to equal 0 to satisfy the above criteria. This results in the following condition equations in matrix form;

$$dX : \quad X^TQ_X{}^{-1} + k^T = 0 \qquad (2.10)$$

$$dY : \quad Y^TQ_Y{}^{-1} + k^TA = 0 \qquad (2.11)$$

$$dZ : \quad Z^TQ_Z{}^{-1} + k^TB = 0 \qquad (2.12)$$

$$dk : \quad -X + AY + BZ + 1 = 0 \qquad (2.13)$$

These 4 equations can be manipulated to suit our needs. The elimination of unknowns is carried out to simplify the solution and avoiding long unwieldy calculations. This elimination of unknowns and other nuisance parameters does not negate their influence. The results obtained can be applied to the appropriate equation to obtain the values of these "eliminated" parameters.

Equation 2.10 results in a solution for X as follows;

$$X = - Q_X . k \qquad (2.14)$$

this is then substituted into equation 2.13 resulting in

$$Q_X . k + AY + BZ + l = 0 \qquad (2.15)$$

Which in turn solves for k

$$k = -Q_X^{-1} (AY + BZ + l) \qquad (2.16)$$

This is now substituted into 2.11 and 2.12 (formulae transposed)

$$Q_Y^{-1} . Y - A^T k = 0 \qquad (2.17)$$

$$Q_Z^{-1} . Z - B^T k = 0 \qquad (2.18)$$

yielding

$$(Q_Y^{-1} + A^T Q_X^{-1} A) . Y + A^T Q_X^{-1} B . Z + A^T Q_X^{-1} l = 0 \qquad (2.19)$$

and

$$B^T Q_X^{-1} A . Y + (Q_Z^{-1} + B^T Q_X^{-1} B) . Z + B^T Q_X^{-1} l = 0 \qquad (2.20)$$

These are a set of normal equations in terms of Y and Z the object points and the image orientations. This is written as follows in simplified form;

$$L . Y + M . Z + T = 0 \qquad (2.19a)$$

and

$$M^T Y + N . Z + U = 0 \qquad (2.20a)$$

Note that M occurs in both equations.

2.3 Closer analysis of these equations.

These matrices from the previous section and the effects of individual elements of these matrices are now studied in detail in order to develop the program further. This will lead to efficient ways of dealing with the vast amount of data this solution requires.

Looking at $L = Q_Y^{-1} + A^T Q_X^{-1} A$;

The A^T matrix consists of vector $A^T{}_p$ in each column. This can be re-arranged in a diagonal matrix. The importance of the axiom proven above (section 1.6.3) that a specific type of matrix multiplied with a diagonal matrix eg. Q_X^{-1} will similarly result in a diagonal matrix is now evident. This is exactly what happens to $A^T Q_X^{-1} A$.

Q_Y^{-1} is also a diagonal matrix linked to the object points with 3*3 submatrices down the centre diagonal as explained in section 2.2.

This yields for group observations on point p;

$$L_p = Q_{Yp}^{-1} + A_p^T Q_{Xp}^{-1} A_p \qquad (2.21)$$

Computers can perform sequential additions very fast. This is usually done by means of loops (a set of commands repeated over and over until a particular preset condition is satisfied). One could do a calculation as set out by the matrix equation 2.21 in the computer by using matrix algebra. This would require setting up of each individual matrix as a basis for the eventual matrix manipulations described by formula 2.21 for example. It is however quite evident that the matrices described earlier are all very large. This means that each matrix will take up a vast amount of memory in the personal computer. Many elements of these matrices are however null values. These matrix manipulations are formulated in terms of additions which in turn will translate to an appropriate loop in the computer program.

For example all the observations of the point p occurring on images i (the point p appears on n images) can be added up to form the A matrix as shown in equation 2.3 for the group observations of point p

Equation 2.21 can be written as an addition for a computer loop ;

$$L_p = Q_{Yp}^{-1} + \Sigma_1^n [A_{pi}^T . Q_{Xpi}^{-1} . A_{pi}] \quad (2.22)$$

This L matrix is made up of submatrices of dimension 3*3 along the main diagonal. The above equation for example means that if point p=4 was measured on images 1,2,4 one would have to add, for group 4

$$L_4 = Q_{Y4}^{-1} + A_{41}^T Q_{X41}^{-1} A_{41} + A_{42}^T Q_{X42}^{-1} A_{42} + A_{44}^T Q_{X44}^{-1} A_{44}$$

Q_{Yp}^{-1} is usually close to or equal to 0 since the weight for a new point is always significantly much smaller or even negligent in comparison with the weight attributed to a control point. This results in effect that for a new point

$$L_p = A_p^T Q_{Xp}^{-1} A_p$$

In the case of a control point, the weight dominates and L_p tends to Q_{Yp}^{-1} for that point.

Examining $M = A^T Q_X^{-1} B$ (and by implication M^T)

This matrix links the object point with the image. Looking at the matrices A and B in equation 2.3 one obtains that

$$M_{pi} = A_{pi}^T . Q_{Xpi}^{-1} . B_{pi} \qquad (2.23)$$

This means that for each observed point p on image i the relevant matrix multiplication is carried out. If point p does not occur on image i then M_{pi} will be a null submatrix. This will result in submatrices of dimension 3 rows and 6 columns (denoted as dimension 3*6).

Examining $N = Q_Z^{-1} + B^T Q_X^{-1} B$

Each column in matrix B^T contains submatrix B_{pi}^T with the other elements null. Q_Z^{-1} is a matrix made up of submatrices (dimension 6*6) along the main diagonal. This results as for L in N containing a series of submatrices dimension 6*6 along its main diagonal. This follows from the axiom as shown in section 1.6.3

above. This summation reveals that for image group i containing point observations p = 1 to m this results in;

$$N_i = Q_{Zi}^{-1} + \Sigma_1^m (B_{pi}^T . Q_{Xpi}^{-1} . B_{pi}) \qquad (2.24)$$

This N matrix is made up of submatrices dimension 6*6 which are the sum of the all the points appearing on image i. For example points p=1,2 and 5 on image i=3 will give;

$$N_3 = Q_{Z3}^{-1} + B_{13}^T . Q_{X13}^{-1} . B_{13} + B_{23}^T . Q_{X23}^{-1} . B_{23} + B_{53}^T . Q_{X53}^{-1} . B_{53}$$

Points not appearing on an image have no influence and do not come into the summation at all. The resultant N_i submatrix will be close to or equal to the Q_Z^{-1} weight matrix if the orientation values of the camera are known. This is due to the fact that known values are weighed heavily against the observations of unknowns.

In contrast herewith N_i will be more heavily influenced by the image point observations and unknown camera parameters when the camera orientations are not known.

Examining $T = A^T Q_X^{-1} l$

Close inspection of this equation shows that for each point p the matrix l is made up of submatrices dimension 2*1. These are multiplied with a diagonal co-variance matrix Q_X which also reflects weights of the observed image co-ordinates. In matrix A the column p which reflects point p will be multiplied out with its appropriate Q_{Xp} and l_p for each of the observed x and y values of point p. There are two such columns for each point observation since x and y form the basis of this derivation. The summation for n images showing point p will yield;

$$T_p = \Sigma_1^n A^T_{pi} Q_{Xpi}^{-1} l_{pi} \qquad (2.25)$$

This matrix multiplication means the addition of all image observations of a particular point. For example point 3 observed on images 1,2 and 4, would yield for $T_p = T_3$;

$$T_3 = A_{31}^T Q_{X31}^{-1} l_{31} + A_{32}^T Q_{X32}^{-1} l_{32} + A_{34}^T Q_{X34}^{-1} l_{34}$$

This yields a submatrix dimension 3*1 for each T_p.
Note that the notation A^T refers to the relevant column of A.

Examining $U = B^T Q_X^{-1} l$

This is very similar to the development of T above. Multiply $Q_X^{-1} l$ with B^T instead of A^T. This means that column p of B is multiplied out as above. Study of a column of B shows that all the point observations p (say p=1,3,5) of image i (say i=2) will be multiplied out with the appropriate Q_X and l and thus summed to yield;

$$U_i = \Sigma_1^n B_{pi}^T Q_{Xpi}^{-1} l_{pi} \tag{2.26}$$

Using the example given here this would result in U_2 being;

$$U_2 = B_{12}^T Q_{X12}^{-1} l_{12} + B_{32}^T Q_{X32}^{-1} l_{32} + B_{52}^T Q_{X52}^{-1} l_{52}$$

This yields a submatrix dimension 6*1 for U_i.

2.4 Further developing of equations.

Equations 2.19 and 2.20 above contain Y (the object point corrections), Z (the orientation unknowns) in conjunction with the actual observations. The X (corrections to image observations) have already been eliminated. Since the 2 sets of matrix equations contain 2 unknowns one can eliminate either Y or Z.

Note that L and N are diagonal matrices. It is much easier to invert a diagonal matrix than other more complex matrices without null values. Both L and N are large matrices. The size of each depends on the number of images and object points solved for.

For each object point 3 corrections are needed. Each camera station requires the solution of 6 parameters. The application of close range photogrammetry usually requires few camera stations to resolve many more object points. The two sets of matrix equations can be manipulated to eliminate the Y values (the object point corrections). This results in the solution of normal equations for camera orientation parameters only.

The matrices involved in this solution will be smaller since few camera stations are required to obtain a large number of unknown object space co-ordinates. This allows for the optimum use of limited random access memory (RAM) available in the PC.

From equation 2.19a one obtains that

$$Y = -L^{-1}.(MZ + T) \quad (2.27)$$

This is used in 2.20a giving

$$[N - M^{T}L^{-1}M].Z + [U - M^{T}L^{-1}T] =0 \quad (2.28)$$

Which in a simplified notation can be written as

$$O.Z + S =0 \quad (2.28a)$$

It is clear from this that since S and O are known one can now solve for Z. The crucial matrices O and S need closer examination.

Looking at O;

Note that

$$O = [N - M^T L^{-1} M] \qquad (2.29)$$

This is made up of a series of submatrices dimension 6*6 linking the point groups with image stations. The matrix multiplication $M^T L^{-1} M$ can be rewritten as an addition of submatrices as follows:

$$- (M^T L^{-1} M)_{ij} = - \Sigma_1^n M_{pi}^T L_p^{-1} M_{pj} \qquad (2.30)$$

This shows that the matrices of all common points occurring on images i and j must be added. This will yield a specific subset of the final O matrix with n as the number of points p which are common to a particular pair of images. The specific pair of images currently involved in this addition are i and j. If either M_{pi} or M_{pj} are 0 that means that there are no points common to images i and j. The summation will then equal 0.

The case where i=j results in a quadratic (multiplication of M matrix with itself and L_p). This will never be equal to a null matrix but result in a set of submatrices dimension 6*6 diagonally about the central diagonal of matrix O.

The submatrix from image i referred to as N_i in equation 2.29 is simply added to the summation of equation 2.30 above in the cases where i=j. This N_i is however ignored where i and j are not equal since N is a diagonal matrix as described earlier.

For the case where i=j ;

$$O_{ii} = N_i - \Sigma_1^n M_{pi}^T L_p^{-1} M_{pi} \tag{2.31}$$

and the case where $i \neq j$ results in;

$$O_{ij} = - \Sigma_1^n M_{pi}^T L_p^{-1} M_{pj} \tag{2.32}$$

It is clear that in most cases, especially terrestrial or close range photogrammetric cases the O matrix will be a full matrix. The only time a submatrix (dimension 6*6) O_{ij} will equal 0 is when no common points occur on images i and j. In the case of a strip survey or aerial strip photography more null matrices are likely. Hence the pattern and layout of O depends on the type of survey.

Special techniques can be applied in such cases where sparse matrices occur. These special techniques fall outside the scope of this half thesis. The personal computer programme envisaged here is specifically designed for general close range photogrammetry applications. This type of work usually requires relatively few image stations (below 10 say) but need to solve a large number of common points. These could even be common to all the images. The inversion of O matrices of smaller sizes will hence be fairly simple and do not need special mathematics to reduce the size of a large sparse matrix.

The ability of the computer to invert an O matrix of a certain size will depend solely on the memory capacity of the personal computer (PC) used. It follows from this that the number of camera stations allowed in a particular solution is a function of the memory capacity of the computer.

Looking at $S = [\ U - M^T L^{-1} T\ ]$;

This matrix is made up of vectors with 6 rows only (dimension 6*1). The summation in this case is;

$$S_i = U_i - \Sigma_1^n M_{pi}^T L_p^{-1} T_p \tag{2.33}$$

This is the summation of matrices involving all the points (image group of n points) p occurring on image i.

The example equation 2.3 of the case involving 4 cameras would now result in;

$$\begin{vmatrix} O_{11} & O_{12} & O_{13} & O_{14} \\ O_{12}^T & O_{22} & O_{23} & O_{24} \\ O_{13}^T & O_{23}^T & O_{33} & O_{34} \\ O_{14}^T & O_{24}^T & O_{34}^T & O_{44} \end{vmatrix} \cdot \begin{vmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{vmatrix} + \begin{vmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{vmatrix} = 0 \tag{2.34}$$

The subscripts here refer to image numbers.
It is this set of equations which must be solved.
Equation 2.28a becomes $O.Z = - S$ which in turn translates to;

$$Z = - O^{-1} S \tag{2.35}$$

This means that O must be inverted. O results in a symmetric matrix due to the particular development technique used in obtaining O. Note that O must also be non-singular.

2.5 Mean square errors and weights.

The set of equations in terms of L,M,N above (equations 2.27 to 2.35) are linked to the co-variance matrices of the unknowns Y,Z and can be written as

$$\begin{vmatrix} QYY & QYZ \\ QYZ^T & QZZ \end{vmatrix} = \begin{vmatrix} L & M \\ M^T & N \end{vmatrix}^{-1} \quad (2.36)$$

This lends itself to the inversion using partitioned matrices to deduce QYY, QZZ and so on. This inversion can be done as follows:

$$\begin{vmatrix} QYY & QYZ \\ QYZ^T & QZZ \end{vmatrix} * \begin{vmatrix} L & M \\ M^T & N \end{vmatrix} = \begin{vmatrix} I & 0 \\ 0 & I \end{vmatrix} \quad (2.37)$$

I is the identity matrix of appropriate column and row size. These can be rewritten as four equations

$$QYY.L \ +QYZ.M^T = I \quad (2.38)$$

$$QYY.M \ +QYZ.N \ = 0 \quad (2.39)$$

$$QYZ^T.M \ +QZZ.N \ = I \quad (2.40)$$

$$QYZ^T.L \ +QZZ.M^T = 0 \quad (2.41)$$

Subtract equation 2.41 from 2.40 and right multiply the result by $-L^{-1}M$. Solving for QZZ thereafter yields;

$$QZZ = (N - M^{T}L^{-1}M)^{-1} = O^{-1} \qquad (2.42)$$

Also using eqn 2.41 directly one can solve

$$QYZ =- L^{-1}M.QZZ. \qquad (2.43)$$

This is the correlation matrix between object points and the image camera orientations. This latter result can now be introduced in eqn 2.38 to solve for the object point co-variances.

$$QYY = L^{-1} + L^{-1}M.QZZ.M^{T}L^{-1} \qquad (2.44)$$

The solution for image orientation data was chosen above other methods because of the economy of effort. It is however only a step in the process to solve for object points represented on the images. Not only is it necessary to solve these, but also some statistical indication of the precision attained from the results is required. The matrix QYY is made up of QZZ which in turn is made up of QZZ_{pq} dimension 6*6 submatrices.

This is multiplied out with M resulting in $M.QZZ.M^T$ being a set of submatrices dimension 3*3 according to the, by now familiar summation formulation;

$$[\ M.QZZ.M^T\ ]_{pq} = \Sigma_{j=1}^{ni} \{ \Sigma_{i=1}^{ni} (M_{pi}\ QZZ_{ij}\ M_{qj}^T)\}$$

Equation 2.45

with p,q a pair of points common to a set of ni images i and j.

In order to obtain the precision of a particular single point the case where the point pair p equals q is relevant. This reduces to a summation of the values for point p appearing on all images picturing point p (point group of p). In the case where point p appears on say images i and j it is only necessary to get the submatrix of QZZ_{ij} for all those images containing point p. The values for QYY of point p alone are hence the diagonal of the reduced QYY matrix of equation 2.44 and is written as

$$QYY_{pp} = L_p^{-1} + L_p^{-1}*[\Sigma_{j=1}^{ni}(\Sigma_{i=1}^{ni}(M_{pi}.QZZ_{ij}.M_{pj}^T)]*L_p^{-1}$$

Equation 2.46

The correlation matrix between points p and q, can also be derived by using a similar summation. In this latter case it is not necessary to add L_p^{-1} since this is a submatrix of L^{-1} which only occurs on the main diagonal. This result can then be utilised in the determination of relative error ellipses. Other statistical precision analysis can also be derived from these matrices.

The study of correlations is a complex field which falls outside the scope of this exposé.

2.6 Comparison with conventional development of normal equations.

The above detailed breakdown of matrices and converting these to summations was done to show how one can obtain the normal equations solution for a bundle adjustment by simply using summations in the computer program. The advantage of this is that the various submatrices can be generated using loops without ever fully forming the matrices contributing to the solution. These are saved on disc and recalled when necessary for the next operation. This results in saving limited random access computer memory (typically 640 Kbyte RAM) for the crucial operation of creating O, the main matrix to be inverted (see eqn. 2.28a). If a computer were available with unlimited memory and speed, these summations would not be necessary. One would then simply use matrix algebra directly from the basic derivation.

The amount of literature on the bundle adjustment is bewildering and sometimes complicated to understand. The complications occur when one has to diverge and study an aspect of the problem in more detail. The reader may well have felt this while following this exposé.

The derivations given thus far seem far removed from the classic solution of the bundle adjustment. The following proves that equation 2.35 is equivalent to the classic solution often quoted in other publications.

The derivation from previous sections resulted in yielding the camera parameters from equations 2.28 and 2.28a

$$Z = - O^{-1} . S$$
$$= - [N - M^T L^{-1} M]^{-1} * [U - M^T L^{-1} T]$$
$$= - \{(Q_z^{-1} + B^T Q_x^{-1} B) - [(B^T Q_x^{-1} A).(Q_y^{-1} + A^T Q_x^{-1} A)^{-1}.(A^T Q_x^{-1} B)]\}^{-1}$$
$$* \{(B^T Q_x^{-1} l) - (B^T Q_x^{-1} A).(Q_y^{-1} + A^T Q_x^{-1} A)^{-1}.(A^T Q_x^{-1} l)\}$$

Equations 2.47

Note that Q_X, Q_Y, Q_Z are diagonal matrices which combined make up the co-variance matrix Q (see equation 2.8). Since these are diagonal they will simply with addition, merge into the multiplication giving

$$Z = -\{(B^TQ^{-1}B) - (B^TQ^{-1}A).(A^TQ^{-1}A)^{-1}.(A^TQ^{-1}B)\}^{-1}$$
$$* \{(B^TQ^{-1}l) - (B^TQ^{-1}A).(A^TQ^{-1}A)^{-1}.(A^TQ^{-1}l)\}$$

Equation 2.48

and regrouping these, replacing Q^{-1} with appropriate P

$$Z= -[B^TPB -B^TPA.(A^TPA)^{-1}.A^TPB]^{-1}* [B^TP -B^TPA.(A^TPA)^{-1}.A^TP].l$$

Equation 2.49

giving

$$Z= -[B^T\{P-PA.(A^TPA)^{-1}.A^TP\}B]^{-1}* [B^T\{P-PA.(A^TPA)^{-1}.A^TP\}].l$$

Equation 2.50

This result is now compared with the classic bundle adjustment case.

An abridged derivation of the classic bundle adjustment case using matrix equations only, follows:

The basic observation equation is according to equation 2.1;

$$X = AY + BZ + l \quad \text{with weight } P$$

The weight P is appropriated to its respective observation. This is classically defined as $P=Q^{-1}$ in terms of the co-variance matrix. This is now written as a matrix equation

$$X = \begin{vmatrix} A & B \end{vmatrix} \cdot \begin{vmatrix} Y \\ Z \end{vmatrix} + l \qquad (2.51)$$

The classic development as described earlier in chapter 1 yields the normal equation system;

$$\begin{vmatrix} A^T \\ B^T \end{vmatrix} P \begin{vmatrix} A & B \end{vmatrix} \cdot \begin{vmatrix} Y \\ Z \end{vmatrix} + \begin{vmatrix} A^TPl \\ B^TPl \end{vmatrix} = 0 \qquad (2.52)$$

Which develops into

$$\begin{vmatrix} A^TPA & A^TPB \\ B^TPA & B^TPB \end{vmatrix} \cdot \begin{vmatrix} Y \\ Z \end{vmatrix} + \begin{vmatrix} A^TPl \\ B^TPl \end{vmatrix} = 0 \qquad (2.53)$$

This can in turn be written out as;

$$A^TPA.Y + A^TPB.Z + A^TPl = 0 \qquad (2.54)$$

and

$$B^TPA.Y + B^TPB.Z + B^TPl = 0 \qquad (2.55)$$

By left multiplying equation 2.54 with $-(B^TPA).(A^TPA)^{-1}$ and adding the new 2.54 equation to equation 2.55 ;

$$[B^TPB - (B^TPA).(A^TPA)^{-1}.(A^TPB)]Z \quad +[B^TP-(B^TPA).(A^TPA)^{-1}.(A^TP)].1=0$$

Equation 2.56

solving for Z

$$Z = -[B^TPB - B^TPA.(A^TPA)^{-1}.A^TPB]^{-1} * [B^TP-B^TPA.(A^TPA)^{-1}.A^TP].1$$

Equation 2.57

regrouping matrices

$$Z = -[B^T\{P-PA.(A^TPA)^{-1}.A^TP\}B]^{-1} * [B^T\{P-PA.(A^TPA)^{-1}.A^TP\}].1$$

Equation 2.58

Comparison with equation 2.50 shows that these are identical. This now proves the equivalence of the derivation in chapter 2 with the classic derivation.

2.7 Magnitude of numerical values.

Equations 2.28 and 2.34 contain matrices representing variables of magnitudes ranging from 0 (eg. zero weight) to a few thousand or bigger. Generally weighting of observations will have a critical effect in the final results. Weight values are typically taken as 0 for "free" points and a comparatively large value for fixed control observations is chosen. If one wishes to force a particular value as invariable its weight might even be kept close to infinity or a very large value in relation to the average weighting of other observations.

It must however be recognised that the introduction of big variations in values may result in mathematical instabilities in the reduction of the normal equations. All the matrices N,L,M,U and T contain a weight matrix. These are most likely to contain big value differences. It is wise to keep weighting of observations to realistic levels. This should avoid divisions by near zero values. The limits of the PC computing accuracies must be kept in mind.

The effects of fixed points and "free " variables has already been mentioned in the analysis of matrices L,N (sections 2.3 and 2.4) which could tend to "normal" size values of $A^T Q_X^{-1} A$ and $B^T Q_X^{-1} B$ or to large values being Q_Y^{-1} and Q_Z^{-1} respectively. This depends on the choice of weights. The matrices containing l tend to "normal" values since these are made up of the approximate image and other observations. These will not show great variations since the order of magnitudes being dealt with are all similar.

A closer look at equation 2.28 shows that this is very much affected by matrices L and N. It has already been shown that these matrices could vary a lot in magnitude. The result obtained from solving equation 2.28 will hence be heavily affected by these very same matrices. It also involves getting L^{-1}. The main diagonal of L is directly influenced by $Q_Y{}^{-1}$. If there are many fixed control points, $Q_Y{}^{-1}$ will generally be of a higher magnitude than the other elements of L not on the main diagonal. The inversion of L with large diagonal values will thus tend to small values nearing 0. It can hence be seen that the observations of fixed control points on the image will not have a great influence on the $M^T L^- M$ part of the O matrix. In such cases O will tend to N.

If there are no common points between image i and j the value of O_{ij} will be zero. Alternately if there are common points O_{ij} will have "normal" values. The diagonal O_{ii} however will vary in size depending on the orientation parameters. If for instance these are well known (eg. a theodolite station) then O_{ii} will be large since N_i will be heavily influenced by $Q_Z{}^{-1}$.

CHAPTER 3

3 Combination of bundle adjustment with geodetic observations.

3.1 General description.

Up to now the purely photogrammetric case has only been discussed. The only "geodetic" values incorporated were fixed control points or fixed camera station parameters. It is just as feasible to look at field measurements taken at the time of photography such as distances, heights and angles between points. These are also subject to imperfections of measurement and hence must be regarded as observations to be incorporated into the adjustment.

Traditionally only (X,Y,Z) co-ordinates of control points were included in the bundle adjustment. These values were obtained by a separate least squares adjustment. The control points adjustment were treated as a separate entity.

The separate adjustment solution for the control point values (X,Y,Z) imposes unnecesary limitations on the bundle adjustment. Control point values can only be obtained from an orderly and coherently observed set of data by making use of an independent adjustment. The photogrammetric solution was then forced to fit the geodetic adjustment. This implies that the photogrammetric case is relegated secondary to the geodetic adjustment.

With modern photogrammetric equipment observations on the image plate could be of a higher order than distance measurements! It is desirable to obtain the simultaneous solution of all the observations.

With this model one can also utilise unrelated distances or directions and heights which could otherwise not have been incorporated in the geodetic control point solution. This will add condition equations to the least squares normal equation system developed in each particular solution.

3.2 Development of normal equations.

Specialised geodetic observations can be incorporated via condition equations in the general bundle adjustment solution. These condition equations are similarly linearised as explained in section 1.6. In general terms the photogrammetric condition equation system can be written as

$$-X + AY + BZ + l = 0 \quad \text{with covariance matrix } Q \qquad \text{(see eqn. 2.1)}$$

The Y matrix is now split into two sets of object space points. The one set contains the points which occur in both the geodetic and the photogrammetric set of observations (Y2). The other set contains those points which only occur in the photogrammetric images (Y1). This results in the expanded equation system;

$$-X + A1Y1 + A2Y2 + BZ + w = 0 \quad \text{with co-variance } Q \text{ as before} \qquad (3.1)$$

Added hereto the linearised geodetic condition equations are generally written as;

$$v - A3Y2 + l = 0 \quad \text{with co-variance matrix } QG \qquad (3.2)$$

In some cases this last equation can be extended by the inclusion of other unknowns Y3 (eg. orientation, refraction and astronomical data). In those cases equation 3.2 can be extended to

$$v - A3Y2 - A4Y3 + l = 0 \quad \text{with } QG \qquad (3.3)$$

The computing capacity of the PC largely dictates the method chosen to solve for equations 3.2 and 3.3. This also influences the choice of unknowns and nuisance parameters which are eliminated to attain a satisfactory solution. All the derivations of such solutions will not be shown here. Equations 3.1 and 3.2 are developed further as a general solution where only object space co-ordinates are solved for:

The terms in the following set of equations are ;

X Image plate observations (ie.$(x,y)_p$ on the image) with co-variance Qx

Y1 Object space co-ordinates occurring only on an image with co-variance Qy1

Y2 Object space co-ordinates occurring on both the photogrammetric image and measured in the geodetic set of observations with co-variance Qy2

Y3 Object space co-ordinates occurring only in the geodetic observations or special conditions with other unknowns with co-variance Qy3.

Z Camera orientation parameters with co-variance Qz

v Vector of residuals of geodetic measurements with co-variance Qg.

k1,k2 Correlation terms arising from the least squares development

l The absolute term similar to the previous examples in equation 1.11.

The least squares condition demands that;

$$X^T Qx^{-1} X + Y1^T Qy1^{-1} Y1 + Y2^T Qy2^{-1} Y2 + Z^T Qz^{-1} Z + v^T Qg^{-1} v \ldots$$

$$\ldots -2k1^T(-X + A1Y1 + A2Y2 + BZ + w) - 2k2^T(v - A3Y2 + l) = \text{Minimum}$$

(a similar equation can be written which includes Y3)

Equation 3.4

The partial derivatives of this must equal 0.

This results in;

$$dX : X^TQx^{-1} + k1^T = 0 \qquad (3.5)$$

$$dY1: Y1^TQy1^{-1} - k1^TA1 = 0 \qquad (3.6)$$

$$dY2: Y2^TQy2^{-1} - k1^TA2 + k2^TA3 = 0 \qquad (3.7)$$

$$dZ : Z^TQz^{-1} - k1^TB = 0 \qquad (3.8)$$

$$dv : v^TQg^{-1} - k2^T = 0 \qquad (3.9)$$

$$dk1: -X + A1Y1 + A2Y2 + BZ + w = 0 \qquad (3.10)$$

$$dk2: v - A3Y2 + l = 0 \qquad (3.11)$$

If Y3 were included the above would be extended with;

$$dY3: Y3^TQy3^{-1} - k2^TA4 = 0 \qquad (3.12)$$

$$dk2: v - A3Y2 - A4Y3 + l = 0 \qquad (3.13)$$

Eliminating from equation 3.5

$$X = -Qx.k1 .$$

This is substituted in equation 3.10

$$dk1:\ Qx.k1 + A1Y1 + A2Y2 + BZ + w = 0 \quad (3.14)$$

Similarly

$$v = Qg.k2.$$

This is used in equation 3.11

$$dk2:\ Qg.k2 - A3Y2 + l = 0 \quad (3.15)$$

This yields for k1 and k2:

$$k1 = -Qx^{-1}(A1Y1 + A2Y2 + BZ + w) \quad (3.16)$$

$$k2 = Qg^{-1}(A3Y2 - l) \quad (3.17)$$

These correlates are now substituted in the remaining equations for dY1,dY2,dZ

$$(Qy1^{T} + A1^{T}Qx^{-1}A1)Y1 + \mathbf{A1^{T}Qx^{-1}A2}.Y2 + A1^{T}Qx^{-1}B.Z + A1^{T}Qx^{-1}.l = 0$$

Equation 3.18

and

$$\mathbf{A2^{T}Qx^{-1}A1}.Y1 + (Qy^{-1} + A2^{T}Qx^{-1}A2 + A3^{T}Qg^{-1}A3)Y2 + A2^{T}Qx^{-1}B.Z \ldots\ldots$$
$$\ldots\ldots + A2Qx^{-1}w - A3^{T}Qg^{-1}l = 0$$

Equation 3.19

and

$$B^TQx^{-1}A1.Y1 + B^TQx^{-1}A2.Y2 + (Qz^{-1} + B^TQx^{-1}B)Z + B^TQx^{-1}w = 0$$

Equation 3.20

The matrix $A1^TQx^{-1}A2$ which refers to the connection between Y1 and Y2 must be a null matrix since none of the points in Y1 occur in Y2. Remember that Y2 reflects only the points common to both the geodetic and the photogrammetric case. No direct connection or correlation exists between these two sets of points. This is a consequential outflow from the deliberate choice to separate geodetically observed points (Y2) from the purely photogrammetric case (Y1). The above equations can now be written in a simplified form

$$L11.Y1 + M1.Z + t1 = 0 \quad (3.21)$$

$$L22.Y2 + M2.Z + t2 = 0 \quad (3.22)$$

$$M1^T.Y1 + M2^T.Y2 + N1.Z + u1 = 0 \quad (3.23)$$

Rewriting equation 3.21;

$$Y1 = -L11^{-1}.(M1.Z + t1) \quad (3.24)$$

Substituted in 3.5

$$M2^T.Y2 + (N1 - M1^T L11^{-1} M1).Z + (u1 - M1^T L11^{-1}.t1) = 0 \quad (3.25)$$

This last equation and the unused eqn 3.4 together can again be rewritten in a simplified form as;

$$L22.Y2 + M2.Z + t2 = 0 \quad (3.26)$$

$$M2^T.Y2 + P1.Z + s1 = 0 \quad (3.27)$$

Eliminate from eqution 3.26

$$Y2 = -L22^{-1}(M2.Z + t2)$$

from equation 3.27

$$- M2^T(L22^{-1}M2.Z + t2) + P1.Z + s1 = 0$$

This yields;

$$(P1 - M2^T L22^{-1} M2)Z - M2^T.t2 + s1 = 0 \quad (3.28)$$

Resulting in the solution for Z:

$$Z = (P1 - M2^T L22^{-1} M2)^{-1}.(M2^T.t2 - s1) \quad (3.29)$$

The other unknowns can then be computed by back substitution in eqn 3.6 and 3.4.

The solution for the case incorporating Y3 follows for completeness sake.

In this case combine the Y1 and Y2 values as Y again ie. all object points are kept under one set.

The fundamental condition equations are written as

Photogrammetric case:

-X +AY +BZ +w =0   covariance matrix Qx,Qy,Qz as before

Geodetic case including for example orientation,refraction unknowns and astronomic co-ordinates:

v +A3Y +B3Y3 +l =0   covariance matrix Qy,Qy3,Qg (for l)

with X,Y,Z and w as defined earlier and Y3 the geodetic (refraction, etc.) unknowns mentioned earlier.

By following a similar process as the derivation above the solution boils down to :

Final equations.

Y3

$$Y3 = -[Qx3^{-1}+(B3^{T}Qg^{-1}B3)]^{-1}[B3^{T}Qg^{-1}l +B3^{T}Qg^{-1}A3.Y]$$

Equation 3.30

Y

$$Y = -[Qy^{-1}+A^{T}Qx^{-1}A+A3^{T}Qg^{-1}A3-(A3^{T}Qg^{-1}B3) * (Qy3^{-1}+B3^{T}Qg^{-1}B3)^{-1}(B3^{T}Qg^{-1}A3)]^{-1} * [A^{T}Qx^{-1}w+A3^{T}Qg^{-1}l-(A3^{T}Qg^{-1}B3) * (Qy3^{-1}+B3^{T}Qg^{-1}B3)^{-1}B3^{T}Qg^{-1}l+A^{T}Qx^{-1}B.Z]$$

Equation 3.31

Z

$$
\begin{aligned}
Z = -[&Qz^{-1}+B^TQx^{-1}B-B^TQx^{-1}A(Qy^{-1}+A^TQx^{-1}A+A3^TQg^{-1}A3-\\
&(A3^TQg^{-1}B3)*(Qy3^{-1}+B3^TQg^{-1}B3)^{-1}).(A^TQx^{-1}B)]^{-1} *\\
&[B^TQx^{-1}w-B^TQx^{-1}A(Qy^{-1}+A^TQx^{-1}A+A3^TQg^{-1}A3-\\
&(A3^TQg^{-1}B3)*(Qy3^{-1}+B3^TQg^{-1}B3)^{-1}).(B3^TQg^{-1}A3)]^{-1} *\\
&[A^TQx^{-1}w+A3^TQg^{-1}l-(A3^TQg^{-1}B3)*\\
&(Qy3^{-1}+B3^TQg^{-1}B3)^{-1}(B3^TQg^{-1}l)]
\end{aligned}
$$

Equation 3.32

This complicated looking set of equations actually has a few patterns in them which tend to repeat. In theory it is possible to program this in a PC with sufficient memory available. Some memory saving will occur due to the repetitive patterns which will allow the programmer to work with a few subroutines.

The solution of this system is done by first calculating Z. The value of Z is then entered into the equation for Y. This in turn is applied to the Y3 equation for the solution of the whole system of equations.

The program presently written for the PC does not include the geodetic solution. This could however be programmed provided enough memory is available in the PC. I have no doubt that this will become a distinct possibility in the foreseeable future as larger and more powerful memory chips become available.

Further work for the solution of a bundle adjustment augmented by geodetic data observations mentioned above, is necessary. The full potential of this technique will then be utilized.

CHAPTER 4

4 Solution for additional physical effects and parameters.(AP)

The discussion up to now has centered on the basic solution of the bundle adjustment model. This is a solution simultaneously involving many different unknowns from one set of image observations. In chapter 3 this was extended further by the inclusion of geodetic conditions.

The strength of this technique rests in its redundancies. A great number of observations is used to solve for space object co-ordinates. These points are observed from many camera positions which generate the redundancies for each point. The solution, in addition, gives the station or camera data.

Besides the above, it is also possible to model and solve other physical effects in cases where a great number of redundancies are available. Deviations from the basic colinearity premise can be mathematically modelled to include effects such as lens distortions, refraction, film or plate deformations due to temperature and other physical effects causing unflatness of the image plate and many others. These can only be solved, if there are enough redundancies in support of the solutions sought. One must also warn against the danger of wishing to solve for too many effects (known as overparametrisation).

Each of the effects caused by some physical condition eg. temperature is usually modelled in isolation. The mathematical formulae are derived from knowledge of the situation excluding all other conditions. It is however still an intellectual interpretation of observed phenomena. One must guard against a false sense of security that the effects have been modelled correctly. The question remains whether the interaction of one effect with another has been described appropriately in the mathematical model used. Most present techniques try to unravel and separate various criteria which inherently may be heavily correlated.

I submit that the basic grassroots mathematical modelling of a problem (in this case the colinearity condition) must be sound. The observations must be set up to result in a great number of redundancies. The sample of observations of object space values taken must be statistically large enough to give a significantly consistent solution. If the above conditions are met, one can expect satisfactory answers even from basic models. It is only justified to bring these extra conditions in, if these particular phenomena are to be studied. One must not lose sight of the possibility that certain effects may inherently be correlated.

If certain significant physical effects are known to apply to a particular situation, they must be included in the solution. In such cases a solution which uses only the basic model will not be acceptable.

It is not appropriate here to embark on a study of all the possible physical effects. Known physical effects and some other possibilities of this technique are described below.

All the formulae in this chapter use;

r	radial distance from principal point
f	is principal distance as defined earlier
k	refractive index
x,y	point position on image (image space co-ordinates)
x_o, y_o	point position of principal point (image space co-ordinates)
A1,A23,P2 etc	are distortion coefficients which need

to be solved.

The coefficients shown here are not the same as those in earlier equations in the other chapters. They refer to the specific distortion effect being discussed.

4.1 Lens distortions.

Lens distortions have always been a major problem in surveying and optical instrumentation. It is manifested in a radial displacement on the image proportional to the radial distance from the principal point. The displacement may be assymetric. Displacements of this nature can also be random. This distortion effect is inherent in any lens. Decentering distortion is a defect attributed to lens construction.

A wide variety of mathematical models has been proposed. These are usually described as a function of displacement on the image in terms of position or radial distance from the principal point. A few models are shown below.

The D. Brown /Conrady model gives the corrections for x and y image plate coordinates due to lens distortions as;

$$dx = x(C1.r^2 + C2.r^4 + C3.r^6) + C4(r^2 + 2x^2) + C5.2xy$$
$$dy = y(C1.r^2 + C2.r^4 + C3.r^6) + C5(r^2 + 2y^2) + C4.2xy$$

Equation 4.1

with

C1, C2, C3 reflecting the radial component
C4, C5 reflecting the decentering component

The Torlegård model uses

$$dx = [\ D3.fr^2 + D4.fr^4\ ]$$

$$dy = [\ D3.fr^2 + D4.fr^4\ ] + D1x + D2y$$

Equation 4.2

with

D1, D2 reflecting decentering
D3, D4 reflecting the radial component

J. Müller model

$$dx = x/r\ (B1.xy + B2.r^3)$$
$$dy = y/r\ (B1.xy + B2.r^3) + B3.y$$

Equation 4.3

with

B1, B3 reflecting decentering
B2 reflecting the radial component

Other authors have also developed various formulae usually in the form of a series or polynomial equation. El Hakim,H. Bauer and G.de Masson d'Autume amongst others come to mind (see bibliography). The criteria used for these formulations are varied and depend on that particular author's interpretation of physical effects and the anticipated magnitude of the systematic error. Some use a priori values for the radial and decentering components based on empirical formulae or laboratory testing. Others allow for a freely adjusted value in the full adjustment.

4.2 Errors due to the image carrier.

This is a vast subject and evidently a cause of many errors in the photogrammetric problem. Basically the image of the object gets displaced from its theoretical colinear position due to inaccuracies or defects of the film emulsion or its surface.

Many possibilities spring immediately to mind. For example the film is not a flat plane surface at time of exposure, the emulsion has a certain thickness thus deviating from the ideal infinitely thin plane, the image gets distorted due to variations in temperature, stretching of the film from handling or the film advance mechanism, effects on the image due to grain thickness of the emulsion, effects from chemical developing and processing. This list is by no means exhaustive.

4.2.1 Film flatness.

The basic effect of the lack of film flatness is a radial displacement of the image from its ideal position on an ideal image plane. This is heavily influenced by the angle of incidence of the light ray on the film. In practice this means that the angular field of the lens used is very important. Wide angle lenses are thus more susceptible to this problem. The film surface is deformed due to camera platten, thickness and the rigidity of the emulsion bearing material (celluloid or glass). These effects have been combined by mathematical modelling using a polynomial of at least 6 coefficients (Brown et al)

$$dx = x/f\{(A13(x^2-y^2)+A14.x^2y^2+A15(x^4-y^4))\} + x(A16.r^4+A17.r^8+A18.r^{12})$$

$$dy = y/f\{(A13(x^2-y^2)+A14.x^2y^2+A15(x^4-y^4))\} + y(A16.r^4+A17.r^8+A18.r^{12})$$

Equation 4.4

with A13, A14, A15 non radial components and
A16, A17, A18 radial and symmetric from the principal point

4.2.2 Film deformations due to processing.

Mechanical, thermal, other physical and chemical influences affect the film itself resulting in distortions of the image. The handling, transportation and care of film before, during and after exposure are important factors. Developing, drying, copying and storage can affect the data stored on an image. The use of fiducial marks contribute to the solution of problems due to these factors. Reseau platten procedures have also been used to obtain higher degrees of interpolation using least squares (Brown Perry et al).

4.2.3 Errors in fiducial markings.

These may not be strictly regarded as film errors but they do appear on the image. The actual measurements of the fiducial marks is thus done from the image. The initial positions of these are usually well established by direct measurements on the actual hardware (camera). However in practice these marks are used from image observations and best fit to the camera calibration. Hence the image is the important factor. Some authors have complained that these marks are too thick to help significantly in resolving film distortions. The grain of the film and other factors such as accuracy of determination in the comparator can influence the eventual corrections applied to the image observations. Care and circumspection must be exercised when applying these corrections.

This error yields a pseudo principal point which is displaced from the true principal point. It results in a translation in x,y of the image space co-ordinates.

The previous comments do not apply in the case of non-metric cameras. I personally prefer to resort to adequate control points to solve for film distortions. The use of fiducial marks to resolve film distortions is especially questioned because of its low redundancy factor (usually only 4 or 8 marks are available per image).

4.2.4 Blurring due to movement.

A slight blurring of the image due to movement (of the object and/or the camera) at the moment of exposure manifests itself. The shutter of the camera stays open a finite time span. Any movement during this crucial exposure period will result in blurring of the image. It is not possible to pinpoint the object on the image as accurately as would otherwise be the case. This effect has not been studied to any great depth as yet.

It is complicated to model adequately since a great variety of factors can contribute to the blurring finally captured on the image. The blurring of the image is a direct function of physical factors such as the direction of camera and object movements (which is not random), time window of exposure, speed of the film, speed and direction of the camera movement (eg. aerial surveys), speed and direction of movement of object and even the type of shutter (focal plane shutter versus other types) mechanism used in the camera.

4.3 Refraction of light ray.

The bending of light rays due to refraction is also a traditional atmospheric effect. It has been studied extensively in applications of conventional surveying using optical equipment. The generally accepted correction formulae are

$$dx = -kx(1+r^2/f^2)$$
$$dy = -ky(1+r^2/f^2)$$

with

k	the refraction coefficient
r,f,x,y	as defined earlier.

4.4 Camera constants and principal point.

Variations of x,y are a function of temperature and pressure differences. These parameters are specifically modelled for each image individually. For example the principal distance f could vary. The effect is a shift in the actual principal point on the image. These are simply modelled as for a general image point;

$$dx = x/f.df$$
$$dy = y/f.df$$

Equations 4.5

Changes in the principal point itself are written as;

$$dx = dx_o$$
$$dy = dy_o$$

Equations 4.6

The original derivation of the bundle solution takes this into account. It is usually ignored as a separate entity to be corrected for. Unless extreme atmospheric or temperature variations are encountered this need not concern one. The pseudo principal point effect mentioned earlier is modelled using equations 4.6.

4.5 Combination of various effects.

A general formula which incorporates some of these abovementioned effects can be utilised.

$$dx = -kx(1+r^2/f^2) + A1.x + A2.xr^2 + A3.xr^4 + P1(y^2+3x^2) + 2P2.xy \; .. \\ + df.x/f + dx_o + A13 + A11.x + A12.y - A31.x^2 - A32.xy + C1(x^3/f) \\ + C2(x^2y/f) + C3(xy^2/f) + C4(x^4/f) + C5(x^3y/f) + C6(x^2y^2/f) \\ + C7(xy^3/f)$$

$$dy = -ky(1+r^2/f^2) + A1.y + A2.yr^2 + A3.yr^4 + P2(x^2+3y^2) + 2P1.xy \; .. \\ + df.y/f + dy_o + A23 + A21.x + A22.y - A31.xy - A32.y^2 + C1(yx^2/f) \\ + C2(xy^2/f) + C3(y^3/f) + C4(yx^3/f) + C5(x^2y^2/f) + C6(x\,y^3/f) \\ + C7(y^4/f)$$

Equations 4.7

4.6 Sequential least squares - Adding to a least squares system.

This is a subject which is presently enjoying the attention of new research. It shows much promise for direct almost real-time solutions of observations. Presently a least squares normal equations system is calculated in one batch. The whole set of equations and observations are read in en bloc and then processed. Traditionally if one then needed to add further observations to this, the whole process would have to be repeated from scratch. The sequential least squares technique allows one to add further observations or unknowns afterwards without having to redo the full number-crunching exercise. The necessary equations simply get added on and inversion of relevant submatrices is carried out. This is analogous to the development of the system as described by the various addition routines in the previous chapter. This has not been used in the programming to date. The subject does warrant further research in the quest to speed up work and get reliable instant results for measurements. I believe this research is presently ingoing at various centres.

It would be very advantageous for direct application in the field. This would mean that once the observations are done, one could then get an immediate answer with full estimates of reliability and accuracies. It may even help to guide the surveyor in adapting his observations to strengthen the accuracy of point fixes where needed. This would eliminate unnecessary extra visits to the site. Results can then also be supplied immediately to clients with full records at one's fingertips.

The general principles and some formulation are given here to encourage the reader to study this further. The linearised least squares equations can be written as

$$lo = Ao.X \tag{4.8}$$

where lo is the observation vector with covariance matrix Q_{lo}

Ao is the coefficient matrix of the unknowns X

The small o subscript refers to initial case before further observations are added.

X is the vector of unknown parameters.

Compare this generalised explanation to the specific development followed in chapters 2 and 3. Note that X here is a general set of unknowns (in terms of the earlier development this X would expand to be X + Y + Z). In this derivation X here must not be confused with earlier definitions.

The least squares solution boils down to

$$Xo = (Ao^T Q_{lo}^{-1} Ao)^{-1} . Ao^T Q_{lo}^{-1} lo \qquad (4.9)$$

As before the Q_{Xo} covariance matrix will be

$$Q_{Xo} = (Ao^T Q_{lo}^{-1} Ao)^{-1} \qquad (4.10)$$

Another set of observations for the same X unknowns with its covariance matrix Q_{l1} is added.

This combined set has normal equations

$$[\ Q_{Xo}^{-1} + A1^{T}Q_{l1}^{-1}A1\ ].X1 = Q_{Xo}^{-1}Xo + A1^{T}Q_{l1}^{-1}l1 \qquad (4.11)$$

Compare this to equation 2.19 which shows remarkable similarity. The solution of these equations is expressed by the SHERMAN-MORRISON formula as

$$[\ Q_{Xo}^{-1} + A1^{T}Q_{l1}^{-1}A1\ ]^{-1} = Q_{Xo} - Q_{Xo}A1^{T}(\ A1Q_{Xo}A1^{T} + Q_{l1}\ )^{-1}A1Q_{Xo}$$

Equation 4.12

Set $K1 = Q_{Xo}A1^{T}(\ A1Q_{Xo}A1^{T} + Q_{l1}\ )^{-1}$

The new solution X1 can now be written as (without proof)

$$X1 = Xo + K1(l1 - A1.Xo) \qquad (4.13)$$

with covariance matrix

$$Q_{X1} = Q_{Xo} - K1.A1.Q_{Xo} \qquad (4.14)$$

This can be redone ad infinatum by resubstitution.

If few observations are added K1 is a small matrix thus making for easy calculation of X1 and Q_{X1}. For example If l1 is a single observation A1 is a row vector, the term in brackets of K1 will be a scalar. One can easily remove an (erroneous) observation by substituting the minus sign with a plus sign in the above equations 4.13 and 4.14.

The above can be generalised by writing

$$Ki = Q_{Xi-1}.Ai^T(Ai.Q_{Xi-1}.Ai^T+Q_{li})^{-1} \qquad (4.15)$$

$$Xi = Xi-1 + Ki(li - Ai.Xi-1) \qquad (4.16)$$

$$Q_{Xi} = Q_{Xi-1} - Ki.Ai.Q_{Xi-1} \qquad (4.17)$$

These equations are known as linear KALMAN filters.

Further reading on the subject is given in the reference section. Note how similar these equations are to those developed in chapter 2. This is in fact basic partitioning of matrices used in reverse. Intuitively one could say that a full set of normal equations have been partitioned for Xi and Xi-1 .

CHAPTER 5

5 Comments and evaluation on dissertation by L. Hinsken.

This is titled;

Algorithms for the establishing of approximate values for the orientation of spatial bundles of light rays.
(German title: Algorithmen zur Beschaffung von Näherungswerten für die Orientierung von beliebig im Raum angeordneten Strahlenbündeln.)

5.1 Background.

A considerable amount of time and effort was put in the study of the methods and derivations explained in this dissertation. It largely follows the trend of derivations explained in previous chapters but there are some distinct differences in the applications of the general bundle adjustment derivation. The derivation is followed in general and elaborated on in detail where it differs materially from derivations previously shown.

This work was studied closely for full implementation and development in the bundle adjustment program on a PC. The developments of the bundle solutions derived thus far are somewhat different from "classic" derivations on the subject. It was decided to use this method in favour of others because of the simplicity in creating the addition loops described earlier. Hinsken's method is very versatile and allows for a wide variety of circumstances. The implementation of Hinsken's method on a small personal computer was found to be impractical due to hardware restrictions of random access memory (RAM). It was for this reason that the implementation thereof was abandoned in favour of a thesis with detailed description of memory saving algorithms for a bundle adjustment solution implemented on a personal computer.

5.2 General development of Hinsken's formulae.

The basic development of the co-linearity equation follows the classic derivation as already set out in my previous chapters. Equation 1.22 can be written in simplified matrix form as;

$$\begin{vmatrix} x_{pi} \\ y_{pi} \\ f_i \end{vmatrix} = sR.\begin{vmatrix} X_p - X_{oi} \\ Y_p - Y_{oi} \\ Z_p - Z_{oi} \end{vmatrix} = s.\begin{vmatrix} p \\ q \\ r \end{vmatrix} \qquad (5.1)$$

This is not a repetition of Hinskens derivations but rather a comment and adaptation thereof following suit with the previous discussion.

The last $|p,q,r|$ matrix is an obvious substitution to simplify the notation of equations. The scale factor s is similar to 1\D of equations 1.5 and 1.22.

By dividing the first and second equations in this matrix formula by the third one shall obtain a pair of equations in terms of x_{pi} and y_{pi}. This also eliminates the unknown scale factor s.
This is rewritten as a new matrix formula

$$\begin{vmatrix} x_{pi} \\ y_{pi} \end{vmatrix} = f_i \begin{vmatrix} p/r \\ q/r \end{vmatrix} \tag{5.2}$$

Once again this is in essence a simplified notation for equations 1.20. Thus far no new notions have been introduced. These equations contain a rotation matrix as described in chapter 1. This is once again an orthogonal matrix.

An approximation of the rotations is established by partial differentiation of equation 5.2. The differentiation is carried out in a similar fashion and for the same reasons as explained in section 1.6.1 yielding;

$$\begin{vmatrix} dx_{pi} \\ dy_{pi} \end{vmatrix} = A \begin{vmatrix} dp \\ dq \\ dr \end{vmatrix} = A \begin{vmatrix} dR \begin{vmatrix} X_p - X_{oi} \\ Y_p - Y_{oi} \\ Z_p - Z_{oi} \end{vmatrix} + R \begin{vmatrix} dX_p - dX_{oi} \\ dY_p - dY_{oi} \\ dZ_p - dZ_{oi} \end{vmatrix} \end{vmatrix}$$

Equation 5.3

Note that the subscripts stand for point p on image i with perspective centre co-ordinates for this image at X_{oi}, Y_{oi}, Z_{oi}. In order to simplify notations of the derivation these subscripts are now omitted. They however remain an integral part of the formulae and must be understood as "unseen" subscripts.

The above equations retain A as;

$$A = \begin{vmatrix} dx/dp & dx/dq & dx/dr \\ dy/dp & dy/dq & dy/dr \end{vmatrix} = f/r \begin{vmatrix} 1 & 0 & -p/r \\ 0 & 1 & -q/r \end{vmatrix} \quad (5.3a)$$

Proof: The derivative of an orthogonal rotation matrix R: .

$$\text{By definition } R.R^T = I$$

and

$$dR.R^T + R.dR^T = 0$$

therefore

$$dR.R^T = -(dR.R^T)^T$$

By virtue of the definition and nature of a skew symmetric matrix one can set this as skew symmetric matrix S. This is written as

$$dR.R^T = S = -S^T$$

with S as in section 1.4. This also implies that

$$dR = S.R \tag{5.4}$$

Note that these rotation equations are in terms of a,b,c (from Cayley's formula) only as in equation 1.4. The object of Hinsken's thesis is to obtain good provisional values of these terms. The a,b,c terms need to be resolved.

Substituting S as derived above into equation 5.3 and associated equations and after a few manipulations one gets;

$$\begin{vmatrix} dx \\ dy \end{vmatrix} = L \begin{vmatrix} a \\ b \\ c \end{vmatrix} + AR \begin{vmatrix} dX - dX_o \\ dY - dY_o \\ dZ - dZ_o \end{vmatrix} \tag{5.5}$$

With

$$L = f/r \begin{vmatrix} pq/r & -(r^2 + p^2)/r & q \\ (r^2 + q^2)/r & -pq/r & -p \end{vmatrix} \tag{5.5a}$$

Note however that as no mathematical connection has been established between |a, b, c| and the rotation matrix.

5.3 The link with a rotation matrix.

It now remains to establish the mathematical link between the general rotation matrix R, the terms $|a, b, c|$ and matrix Q (in other words p, q, r) given in the above equation.

The interesting approach used by Hinsken is to create R into an orthogonal matrix T which contains R. He also makes use of a vector $u=|n, k, l, m|$ with 4 elements, 3 of which are independent, with the dependence of the fourth term defined as

$$n^2 + k^2 + l^2 + m^2 = 1 \qquad (5.6)$$

This means that the vector u is normalised, thus creating homogeneous values. By introducing matrices Pu and Qu as follows;

$$Pu = \begin{vmatrix} n & k & l & m \\ -k & n & m & -l \\ -l & -m & n & k \\ -m & l & -k & n \end{vmatrix} \quad \text{and} \quad Qu = \begin{vmatrix} n & -k & -l & -m \\ k & n & m & -l \\ l & -m & n & k \\ m & l & -k & n \end{vmatrix} \qquad (5.7)$$

It can easily be proven that these matrices are orthogonal. The product of these matrices yields;

$$T = Pu.Qu = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & & & \\ 0 & & R & \\ 0 & & & \end{vmatrix} \tag{5.8}$$

with

$$R = \begin{vmatrix} n^2+k^2-l^2-m^2 & 2(kl+mn) & 2(kn-ln) \\ 2(kl-mn) & n^2-k^2+l^2-m^2 & 2(lm+kn) \\ 2(km+ln) & 2(lm-kn) & n^2-k^2-l^2+m^2 \end{vmatrix} \tag{5.9}$$

Again orthogonality is proving that $T^T.T = I$. From this follows that $R^T R = I$ and is thus also orthogonal. Note that if one transposes R above and the general terms k, l, m, n are substituted with d, a, b, c respectively, one then obtains;

$$R^T = \begin{vmatrix} d^2+a^2-b^2-c^2 & 2(ab-cd) & 2(ac+bd) \\ 2(ab+cd) & d^2-a^2+b^2-c^2 & 2(bc-ad) \\ d^2-a^2-b^2+c^2 & 2(bc+ad) & d^2-a^2-b^2+c^2 \end{vmatrix} \tag{5.10}$$

Substituting d=1 in this latter equation a great similarity can be seen with this equation and equation 1.4 for a rotation matrix R. The R matrix developed here is made up of 3 independent terms. The well known Rodrigues matrix of equation 1.4 is thus a special form of this generalised rotation matrix R developed here.

Values for the terms of the vector u above, are introduced as approximations. These approximations are then updated by re-iteration in the reduction process and converge to the true values for u. This requires Newton-Gauss differentiation as explained earlier. For each iteration, the following substitution formula applies;

$$u_1 = u + du \qquad (5.11)$$

By making use of Taylor's formula and after some manipulations through the fact that the length of the vector u=1 (see equation 5.6), one obtains that;

$$n\ dn + k\ dk + l\ dl + m\ dm = 0 \qquad (5.12)$$

This can be written as

$$\begin{aligned} n' &= 1 + n\ dn + k\ dk + l\ dl + m\ dm \\ k' &= 0 - k\ dn + n\ dk + m\ dl - l\ dm \\ l' &= 0 - l\ dn - m\ dk + n\ dl + k\ dm \\ m' &= 0 - m\ dn + l\ dk - k\ dl + n\ dm \end{aligned}$$

Equations 5.13

Which means that $n' = 1$ and $dn = -(k\ dk + l\ dl + m\ dm)/n$
This is substituted into the last three equations of 5.13 yielding;

$$\begin{vmatrix} k' \\ l' \\ m' \end{vmatrix} = 1/d \begin{vmatrix} n^2+k^2 & kl+mn & km-nl \\ kl-nm & n^2+l^2 & lm+nk \\ km+nl & lm-nk & n^2+m^2 \end{vmatrix} . \begin{vmatrix} dk \\ dl \\ dc \end{vmatrix}$$

Equation 5.14

By equating

$$\mathbf{C} = 2/d \begin{vmatrix} n^2+k^2 & kl+mn & km-nl \\ kl-nm & n^2+l^2 & lm+nk \\ km+nl & lm-nk & n^2+m^2 \end{vmatrix} . \begin{vmatrix} dk \\ dl \\ dc \end{vmatrix}$$

Equation 5.15

and writing

$$\mathbf{w} = \begin{vmatrix} a \\ b \\ c \end{vmatrix} = \mathbf{C} \begin{vmatrix} dk \\ dl \\ dm \end{vmatrix} \qquad (5.16)$$

This is substituted in equation 5.5 giving the result;

$$\mathbf{C}^{-1} = 1/2 \begin{vmatrix} n & -m & l \\ m & n & -k \\ -l & k & n \end{vmatrix} \qquad (5.17)$$

This gives a set of recursion formulae for updating the general elements

$$
\begin{aligned}
n_{i+1} &= n_i - k_i a/2 - l_i b/2 - m_i c/2 \\
k_{i+1} &= k_i + n_i a/2 - m_i b/2 + l_i c/2 \\
l_{i+1} &= l_i + m_i a/2 + n_i b/2 - k_i c/2 \\
m_{i+1} &= m_i - l_i a/2 + k_i b/2 + n_i c/2
\end{aligned}
$$

Equations 5.18

Hinsken proves this derivation by the following short manipulations of the basic matrices and vectors.

$Pu.Pu^T = I$ and $Pu.u = i$ (with $i=[1,0,0,0]^T$)

for normalised elements in Pu (obtained by using equation 5.6)

$Qu.i = u$ (normalised) $= Qu.Pu.u$

This in turn results in $T.u = u$ (normalised). This follows on to $R.o = o$ with o defined as follows;

$$
o = \begin{vmatrix} k \\ l \\ m \end{vmatrix} \tag{5.19}
$$

5.4 Advantages of this method.

The main advantage gained from this derivation is that the orientation elements in the solution are not trigonometric values. This simplifies calculations and saves time in the reduction process.

This was also achieved in the derivations preceding this chapter.

Hinsken claims that due to the rotation matrix R above and by implication **C**, only quadratic terms of k,l,m,n are involved. This results in a rapid convergence of the solution, even with bad initial values of the vector u. This is an advantage in circumstances where less than ideal close range photogrammetry exposures must be taken. The results quoted in his dissertation confirm this.

5.5 Correlation between u and conventional angular rotations.

The angular rotations represented by w,o,k are as defined earlier in section 1.3.

Direct substitution of equivalent terms from a comparison of equation 1.1 with equation 1.4, results in;

$$
\begin{aligned}
\text{coso.sink} &= 2(c-ab) \\
\text{coso.cosk} &= 1+a^2-b^2-c^2 \\
\text{coso.sinw} &= 2(a-bc) \\
\text{coso.cosw} &= 1-a^2-b^2+c^2 \\
\text{sino} &= 2(ac+b)
\end{aligned}
$$

Equations 5.20

This expands to similar formulae using Hinsken's more general formula. Once again the nomenclature used hitherto means that a=k, b=l, c=m are substituted where applicable. The relation d=n is added as the extra more general term. This substitution is done to facilitate direct comparison with the formulae obtained here which are very similar to Hinskens derivation.

$$\cos o . \sin k = 2(dc-ab)$$

$$\cos o . \cos k = d^2+a^2-b^2-c^2$$

$$\cos o . \sin w = 2(da-bc)$$

$$\cos o . \cos w = d^2-a^2-b^2+c^2$$

$$\sin o = 2(ac+db)$$

Equations 5.21

5.6 The applications using this method.

Hinsken uses a special case of this general derivation for a pair of images. This gives him a simplified set of equations using only two images. I respectfully request the reader to refer to this article for the derivation, as this is not a full solution for a bundle adjustment and falls outside the scope of this half thesis. Its usefulness in special cases must not be misjudged.

The examples given by the author show a variety of different circumstances where stereopairs were used under less than ideal conditions. He nevertheless obtained adequate convergence of the orientation parameters.

It is interesting to note that no effort is made in obtaining good approximate values to initiate the solution. The dissertation shows that in general initial values for k,l,m are taken as zero and n is made equal to 1 !

5.6.1 General discussion and comments.

The equations used by Hinsken are fairly simple and certainly less complicated than the full bundle adjustment derivations given earlier. The iterations quoted for u are simple to programme in a personal computer.

This solution leads directly to approximate orientation values for the camera stations only. Once these are solved, one would have to substitute the results in the appropriate formula and solve for X,Y,Z of the object space points. This in turn will provide approximate values for the other parameters.

Weighting of the equations utilised to obtain the values obtained by this method is of importance. The precision obtained will be subject to circumstances and the positions of the cameras at the time of exposure. This has a significant effect in applications where a wide variety of cameras and/or totally different camera configurations are used. The precision will be heavily influenced in cases where camera stations are situated at significant different distances from the object (eg. camera 1 is 10 metres from the object versus camera 2 being 100 metres away).

The simplicity of the iteration formulae do show promise for incorporation in a bundle adjustment program. It will be necessary to apply Hinsken's mathematical solution hand in hand with a rigorous least squares derivation of the bundle solution.

The solution as given here could be useful in a bundle adjustment where preliminary orientation values are required to initiate the full adjustment. Good approximate values always speed up the convergence of the parameters to their final values in a full bundle adjustment solution.

It may be useful to take a sample pair or a few independent pairs of photographs of a project. These pairs may then be subjected to Hinsken's solution to obtain good approximate orientation values for these images. This would be done as a separate preliminary exercise.

The approximations thus obtained are then utilised in the final bundle solution. Orientations thus derived are then applied a priori in the full adjustment. This may assist in obtaining faster convergence of the bundle adjustment. It may also act as safeguard against gross errors. The final bundle adjustment could be compared with the approximations from Hinsken's solution. Significantly large differences will point to gross errors.

5.7 General.

The direct application of this method as a means of obtaining good approximate values, uses much needed computer memory (RAM). The object of this half thesis was to study memory saving algorithms and applications of a bundle adjustment solution for a program on a PC. The Hinsken approach can be seen as a preliminary aid to a full bundle solution. It is not part of a bundle adjustment.

It may be of value to obtain more information from more recent research done on this problem. If this method proves its value it may become a full solution of the bundle adjustment problem in itself, since the orientation values are crucial data elements in every photogrammetric problem.

Alternatively it could be incorporated in a general bundle adjustment program as a pre-program to resolve the approximate orientations for eventual use in a full solution of the bundles involved.

This method was not pursued further. It is beyond the scope of this half thesis to research this particular aspect of the adjustment problem further.

CHAPTER 6

6 The Personal Computer program - General discussion.

The aim of this half thesis was to study and develop algorithms suitable for a bundle adjustment solution on a PC with its typical speed limitations and RAM restrictions.The crucial algorithms for a bundle adjustment program were tested on the IBM compatible Personal Computer. Severe time limitations have restricted the further extensions envisaged in chapters 3, 4 and 5.

The computer used is a MIAD AT computer with 40 megabyte hard disc and one floppy disc drive. The basic memory capacity of the computer is 1 megabyte. a Mathematical coprocessor was also fitted to the computer. The random access memory (RAM) and the speed of the computer are the greatest limiting factors in the development of the bundle adjustment program on a PC.

6.1 Limitations imposed by hardware.

In the past the speed of the computer affected the viability of utilising a small PC for the solution of a big network. The advances in computer technology have however speeded up the personal computer to such an extent that speed is now fast becoming a minor limiting factor.

RAM restricts the size and number of matrices the computer can handle at any specific moment in time. This forces one to temporarily store the other matrices not presently used on a storage device such as a hard disc. Matrix partitioning algorithms are also used for the same reason.

Storing of input and output data could be done on the floppy disc but this generally is very much slower than hard disk storage. Floppy discs are also restricted in memory space. The use of floppy discs limits the number of points and stations which can feasibly be handled at any specific moment in the computer. The hard disc with its far greater memory capacity does not impose such limitations.

As a general guide one can accept that a matrix in the computer will typically be defined as A(m,n) where A is the matrix name with dimensions m*n. Double precision data elements require 8 bytes. This may vary depending on the PC used. In this particular case, it means that for the each array i.e. its name and the elements in the array itself, one will need about 14 bytes plus 8 times the number of elements in the array. The inversion of a matrix in the computer results in two matrices being resident in memory; the main matrix and its inverted counterpart. In order to be able to invert the O matrix, which is the main matrix in the derivation it is necessary to allow for only half the available RAM for the O matrix. The inverted matrix O^{-1} will occupy the remaining RAM. The program needs about 20 kilobytes and allocation of various variables will take about 5 kilobytes. The various household items needed by the operating system also take up memory space.

Assuming 30 kilobytes are required to set up and run the program. With 640K RAM only available to the BASIC compiler this leaves about 610 kilobytes for arrays and other items to be stored in memory. This again must be halved leaving about 305 kilobytes for the main matrix to be inverted. Note that approximations are used here. This means one could expect to be able to invert a square matrix of about 190*190 elements. Since the O matrix is made up of 6*6 submatrices this would allow for about 30 camera stations in the case of a 640 kilobyte computer.

After the solution has been achieved, one needs to solve for inter alia; object points, covariance matrices and other statistical data. It is more realistic to expect the solution of about 20 images or stations. Computing capacity can be increased by careful programming techniques which save memory space. The limits in this program have not been tested nor improved upon due to time restrictions.

Once the O matrix has been inverted, it must be saved on disc. This will release more RAM for the solution of the individual points and any other parameters one needs to obtain. The worst case would be where all the points appear on all the images. This would require space for the L, M, Z,and T matrices simultaneously. For each point the requirement is a total of about 48 elements distributed among the appropriate submatrices of L, M, Z and T.

6.2 Basic flowchart of program. Program sequence and procedure.

The program flowchart is almost linear with a few looping procedures. This deceptively simple looking flowchart and procedure was very difficult to program effectively. With more programming time available and the improvements in PC technology it will be possible to devise a more efficient program covering the full application of the bundle adjustment solution.

The calculation sequence follows:

1 Preparing and reading in of basic parameters and numeric constants.

2 Setting up approximate values to initialise iteration process.

3 Reading in co-ordinates of control points and/or other control data observations such as a fixed distance.

4 Various preliminary sorting of points and setting up of image and point groups.

5 Reading in of image co-ordinate observations and sorting these. Removal of points which only occur once.

6 From data, calculating submatrices till final set-up of O-matrix.

7 Inverting O-matrix and solving camera orientation parameters.

8 Using backward substitution to solve for ground points.

9 Testing iteration against previous values to required accuracy. If iteration not adequate redo loop from item 6.

10 Iteration adequate; get final parameter data and points.

11 Error and Q-matrix calculations, statistical confidences.

This is a simplified program flow summarising the full bundle adjustment procedure.

6.3 More detailed comment on simple bundle adjustment program.

This discussion should be read in conjunction with the listing of the program as reference will be made to the various line numbers or subroutines used.

6.3.1. Setting up of basic parameters.

Option angles is used as traditionally sexagesimal angles are used. Various basic library functions such as Arcsin, Arctan etc. can also be defined here. It is preferable to have these at the beginning of a program for easy future reference. It is always easiest to spot library functions from the outset.

Since the computer has various disk drives, it is necessary to define where the basic input of observations is coming from. Typically this will be drive A. This is a floppy drive which contains data from other sources already read in using associated simple programs.

6.3.2. Setting up approximate values initialising iteration data.

The routine headed "camera provisional orientation parameters" can be a dual routine. It can either use any basic values which are built in the main program or values supplied via an external device such as a keyboard or disk. The program could supply built in default values for the initial parameters of each image. This can result in similar initial values allocated to each image. This practice is rather unsafe and could be the cause of an unsuccessful run. It can also result in a bad first approximation which could give a fluctuating iteration never reaching an accurate solution. The allocation of initial approximate values must be done judiciously. Bad initial approximations might not result in a convergence of the solution. The user will generally have some idea of positioning and camera orientation.

Alternatively the Hinsken solution discussed in chapter 5 could be utilised. Examples of Newton-Raphson approximate differentiation methods based on bad approximations come to mind as an analogy. It may also result in matrices which are difficult to invert due to numerical imbalances. This phenomenon is comparable with division by zero in arithmetic.

The camera orientation parameters are read in as values for a,b,c as described in section 1.4. Note however that a relation has been given linking the Rodrigues parameters to the more traditional rotation angles w,o,k in section 1.5. These can be utilised to compute adequate initialisation parameters.

The program reads in values for the provisional camera parameters Xo,Yo,Zo and the principal distance with a,b,c in single data line. Provision has also been made for the co-ordinate of the principal point on the image in x,y if available.

It is not necessary to tell the computer how many images there are as it will continue reading the data file until no more data is found. This is achieved using a DO WHILE MORE loop. An added advantage is that this also allows the program itself to establish how many images there are. It may be a small matter but it eliminates the possibility of incorrect data input.

As a further check of the correctness of data input and for the convenience of the user, this initial data is printed out as it is read.

The provisional values are stored in data matrix Par(10,NI) where NI is the number of images.

A small safety stop is built in for the case where only one image is read instead of the two or more images required for a bundle adjustment. The program issues a comment and simply stops.

Provisional image data values are now evaluated. A provisional Rodrigues rotation matrix is set up for each image.

6.3.3. Control point data and other measurements.

It has already been mentioned in an earlier section that the implementation of geodetic observations and similar extensions, although they form an important aspect of an extended bundle adjustment program, have not been implemented. This was not within the requirements for this half thesis. Provision is only made for reading in normal control point co-ordinates. If some of the control points contain one or two of the ordinates, the others must be entered as dummy variables in the list being read in. The appropriate weighting of the critical values required, should ensure convergence of the solution.

The reading in of this takes the form of: point number X,Y,Z co-ordinate. Program nomenclature is C(NC), X(NC), Y(NC), Z(NC) respectively, with NC the number of control points. Note that the point number itself is set up in a matrix C. This means that the program will use the position of that control point in matrix C . It enables manipulation of points independent from the point numbering entered in the computer. All programming manipulations of points and camera data is done through repositioning in the appropriate matrices. This has the advantage that the computer cannot confuse point names or "hang" due to accidental duplication of data nomenclature. If the image space data read in is correctly linked to the object space points the solution will be correct despite duplication of data point names on an image.

I chose to separate the co-ordinates in three matrices to simplify matrix cross-referencing and searching of values. Matrices are redimensioned to the total data read in where necessary in order to save memory space.

6.3.4. Image observations: reading in x,y of each image.

Subroutine REIMCON is invoked here. This reads image number, point number, x and y of observed point on the image.

Program nomenclature is CC(N,4) where N is the total number of observations. This is also printed out for verification purposes. Similar reading in procedures are used for this as described in section 6.3.3.

Subroutine REIMCON was specially set up to allow easy conversion and extension of the program if more data is to be read from extra sources. The reading in procedure will be the same and only small adaptations will be required to extend CC or even create a new observation matrix for a comprehensive bundle adjustment program.

6.3.5. Sorting of data, point groups, image groups.

One of the main aspects mentioned in the development of formulae from section 2.3 onwards was, the grouping of points. This is vital for the easy setting up of the various submatrices. The summation procedures shown in section 2.4 and others are applied here.

It is now necessary to set up image groups and point groups from the image space data. The data for each image group is naturally read in since each photograph is traditionally measured and recorded separately. The reading in of data into the bundle adjustment program does not change or upset this image group data. It is useful to sort the image numbers in ascending order. This is mainly for cosmetic reasons, but it does assist with printing and sorting of data.

The point groups as mentioned in section 2.3 also need to be set up. This is done using the fast but easy to program Shell-Metzner sort. This algorithm uses temporary counters but does not need elaborate extra matrices and variables to store data. It hence does not stretch computer memory capacity unduly. A separate matrix will be required to store the point group data.

Matrix CCC(N,2) with N the number of observations, has been set up for this. There are two column values associated with each observation N reflecting the point and image number respectively. This is derived directly from the data of CC above.

The result of these sorts is simply a grouping of point numbers in ascending order with its associated image numbers (CCC). On the other hand CC is sorted in ascending image numbers. Correct image grouping is guaranteed by this procedure.

Line 2410 is the start of a crucial cross referencing list which tells the program when a point group in CC changes. This is vital for the setting up of submatrices as mentioned earlier. It also acts as a check for the case where a point was observed only once in the adjustment. If a point does not appear more than once on an image it cannot be used in the solution, since this mathematically describes a ray to which no other observation can intersect. It cannot be accepted for the solution.

The routine on lines 2580-2630 automatically removes the offending single observation and adjusts the matrix CC set up earlier.

The positions of groups and changepoints is listed in matrix CHAN.

6.3.6 Setting up of matrices for normal equations.

Matrices are initially set up and dimensioned in lines 2720 to 2810. Subroutine WEx is called after the sorting routine mentioned in 6.2.5.

This routine is the setting up of the basic approximations, which get updated with each iteration. This routine uses the camera data, the point observations and control point data to derive the partial derivative values, set out in formulae in chapter 1, for each observed point. The values are stored separately in single row matrices N observations long. In particular the $\delta Fx/\delta x$, $\delta Fy/\delta y$, etc. terms are stored in separate matrices positioned to coincide with the particular camera point pair observation as set out in the sorted CC matrix. These values are the backbone of the following submatrices which are needed for each group.

The loop that follows reads in values for B (2*6 matrix) for point i image j. This is stored on the hard disc. Matrix data for B is cross referenced with point and image number. Refer to chapter 2 for the derivation of the theory.

Once B is established this way, B^T is obtained. These B submatrices are now used in the summation as explained in section 2.4 where N is examined. The N matrix is obtained by summation of appropriate groups. In program nomenclature N_i (i refers to image number i) is described as NNJ. The resultant matrix is then added to the Qz_i matrix for the appropriate group. This N submatrix is then properly cross referenced for each image group and also stored on hard disc for later use.

The next loop is concerned with point groups and the creation of A matrices. Reference to chapter 2 will reveal the structure of A submatrices. The A matrix is primarily related to the point only. The CHAN matrix is now used to search all the observations of point p.

Once again the basic data set up earlier is read directly into A (2*3 matrix) and stored on the hard disc with appropriate cross referencing. A^T is then obtained and also stored as this is also needed later. It is better to store A^T as well than having to read in A itself and transposing that when needed. This way the transpose is done only once and computer time is saved at the expense of more storage space needed on the hard disc.

L_p is directly dependent on A, Qy and Qx. It is now necessary to compute the summation $A^TQx^{-1}A$. This is used immediately and by reading in the appropriate B matrix from earlier, the sum for $A^TQx^{-1}B$ is computed as well. Note that the final addition of the Qy_p^{-1} has to wait till the loop is completed. This sets up the M_{pi} matrix described in section 2.4. Program nomenclature for L_p and M_{pi} is LI and MMIJ respectively

It now remains to compute the T_p and U_i matrices which use the constant l terms. The summation $A^TQx^{-1}l$ and $B^TQx^{-1}l$ is carried out and saved on disc in a similar fashion. Program nomenclature for T_p and U_i is TTI and UUIJ respectively (IJ refers here to a point i and image j observations).

Finally Qy_p is added to the L matrix at the end of this looping procedure. This takes care of almost all the summations required. This, once again, is stored on disc as mentioned before.

This whole looping procedure is repeated for each image/point group. The working matrices in RAM of the PC are reset to zero before each new summation.

The U terms are related to image i and must be summed for each set of points observed on that image. This is accomplished by the next loop (lines 4040 to 4190). Program nomenclature for U_i as finally required is UJ.

The loops described above are the practical implementations of the theoretical description in section 2.4. All that is now required, is to obtain the final normal equation system as set out in section 2.5.

6.3.7 Inverting O matrix, solution of camera orientations.

The structure of the crucial O and S matrices is described as a summation similar to the ones developed above. This summation is achieved by a further loop. Program nomenclature for O_{ij} is PIJ and similarly for S_i it is SI where i and j are now images. Since one is likely to use the O submatrices a few times with different values for each iteration, it is better to reset all stored O matrices equal to zero. This avoids possible corruption of data.

A loop is set up to read all the observations once again. The appropriate M_{pi}, L_p, T_p and U_i matrices are read from disc. These are then multiplied and added as necessary.

M_{pj} is later also read in, multiplied out and eventually added to N_i as described earlier.

The above creates the various submatrices for O and S. The next section now combines this into one big matrix equation set, as described in chapter 2. This matrix O is the main matrix which has to be inverted. Once fully set up, it is simply inverted using the standard matrix calculation routines built in the Basic program compiler.

There are other means of inverting large matrices by reducing the size using special techniques. The possibility to allow for inversion of very large matrices at this point has not been studied here due to time limitations. It is also considered outside the scope of this half thesis.

The solution used here is adequate for most average size networks likely to be encountered by the practitioner employed in the close range photogrammetric field. Expansion into larger networks, inclusion of geodetic control data, other factors such as distortion and so forth are for future research and practical implementation.

6.3.8 Iterations, final results, error calculations.

Items 8 and 9 in the calculation procedure are repetitions of the main program flow. These backward substitution and iteration procedures merely "plug" values in the appropriate section of the program.

In items 10 and 11, the final points are now solved for and the results set out in whatever printout format required or preferred by the user. Various statistical tests can be applied for the results depending on the needs and demands of the job itself.

The solution of a bundle adjustment results in various data items. Each item is clearly defined and can be grouped. It will be advantageous to standardise the storage and data sequence on mass storage devices. The standardised bundle adjustment format thus documented and stored can be utilised and manipulated easily by various other programs (eg. commercial database programs) for a specific application.

7 CONCLUSION.

The bundle adjustment solution is a mathematically elegant but nevertheless highly pragmatic solution of the photogrammetric restitution problem. It is very versatile showing clear practical potential not only in all fields of photogrammetry but also in traditional areas. The basic equations can even be adapted to traditional theodolite observations. Geodetic data can be incorporated. The traditional shortcoming of the bundle adjustment solutions is, its limitation to large computers. This can be overcome as was shown in this half thesis.

The design and encoding of a full bundle adjustment program with various extra possibilities mentioned in earlier chapters, such as lens distortions, the inclusion of simultaneous geodetic data, analysis of object point accuracies and many other applications, is a big task. More practical research in this field should be done. The algorithms and techniques described here, form an excellent basis for a comprehensive bundle adjustment program. Advances in technology and availability of more powerful personal computers will enlarge the scope of this technique.

The use of a bundle adjustment program in photogrammetric applications requires background knowledge in this field and it is unlikely that the bundle adjustment routine will ever be usable as a full "black box" facility in photogrammetry.

It shows promise as a survey three dimensional point fixing tool since the routine can be applied to traditional theodolite observations. Once the basic program and data entering procedures has been accomplished, it can become a "black box" procedure easily used by less skilled persons who can then use the program as a tool to solve dimensional problems.

The taking of a quick complete set of overlapping photographs of an object takes relatively little time. Measuring and data analysis for object point fixing are done in the comfort of a laboratory or office. This technique enables quick results with a high degree of confidence. The surveyor can then spend his time more effectively analysing results and developing more accurate networks and procedures. The tedium of calculating the number crunching routines is left to the computer.

The previous chapters show how this can be achieved. The derivations and special developments of looping formulae allow for easier more economical computing and data storage.

This half thesis has clearly proven that the bundle adjustment solution is within the capabilities of a small personal computer.

The task set for this half thesis has been fulfilled. All the algorithms described in chapter 2 have been successfully tested on a PC. The program is now ready for further development of the bundle adjustment technique in a PC environment employing traditional geodetic methods together with the close range possibilities of photogrammetry. The potential advantages of this have been described in this half thesis.

Bibliography.

1) Adams L.P. (1980)
The use of non-metric cameras in short-range photogrammetry.
Photogrammetria.
36(1981) 51- 60.(1980)

2) American Society of Photogrammetry (1944)
Manual of Photogrammetry.
Pitman publishing corporation.

3) Avery T.E. (1977)
Interpretation of Aerial Photographs.
Third Edition
Burgess Publishing Company
Minneapolis, Minesota
ISBN 0-8087-0130-4 (1977)

4) Ayres F.,jr. (1974)
Schaum's outline series Theory and problems of matrices.
McGraw-Hill book company. (1974)

5) Ayres F.,jr. (1972)
Schaum's outline series Theory and problems of calculus.
McGraw-Hill book company. (1972)

6) Barden W.,jr. (1981)
Programming techniques for level II Basic.
Radio Shack Tandy Corporation Company. (1981)

7) Bauer H. (1973)
Compensation of systematic errors by analytical block adjustment with common image deformation parameters.
Offizielle OEEPE veröff. Nr.8, Frankfurt 1973.

8) Brown D.C. (1958)
A solution to the general problem of multiple station analytical stereotriangulation.
RCA data reduction technical report No. 43.
AFMTC-TR-58-8
Astia document No. 134278.(1958)

9) Brown D.C. (1980)
Application of close-range photogrammetry to measurements of structures in orbit volume 1.
Geodetic Services Incorporated technical report No. 80-012.(1980)

10) de Mason d'Autume G. (1974)
Les erreurs systématiques dans l'aérotriangulation.
Bull. Soc. Franc. de Phot. No. 56, October 1974

11) Earls C.J. (1983)
Accuracy potential of a system for analytical close range photogrammetry.
Photogrammetric Record.
11(62):169-182(October 1983)

12) El-Hakim S.F. (1982)
The general bundle adjustment triangulation (GEBAT) system Theory and applications.
National Research Council Canada
20539.(1982)

13) Granshaw S. (1980)
Bundle adjustment methods in engineering photogrammetry.
Photogrammetric Record.
10(56): 181-207 (October 1980)

14) Grün A. (circa 1985)
Photogrammetrische Punktbestimmung mit der Bündelmethode, Leistungssteigung durch Kompensation systematisher und Elimination großer Fehler.
Institut für Geodäsie und Photogrammetrie an der Eidg. Technischen Hochschule Zürich.
(circa 1985)

15) Haag K. (1977)
Die photogrammetrische Bündelausgleichung und die Auflösung großer Normalgleichungssysteme.
Deutsche Geodätische Kommission.
ISBN 3 7696 92888.(1977)

16) Hakkarainen J. (1985)
Radiale Verzeichnung und die Komponenten der Dezentrierungsverzeichnung in modernen Luftbildkameras.
Deutsche Geodätische Kommission.
ISBN 3 7696 8562 8.(1985)

17) Hinsken L. (1987)
Algorithmen zur Beschaffung von Näherungswerten für die orientierung von beliebig im Raum angeordneten Strahlenbündeln.
Deutsche Geodätische Kommission.
ISBN 3 7696 9382 5.(1987)

18) Kölbl O. (1972)
Selbstkalibrierung von Aufnahmekammern.
Bulletin
31-37 (1/1972)

19) Kupfer G. (referent) (1977)
Teilkalibrierung.
Deutsche Geodätische Kommission.
ISBN 3 7696 9292 6 (1977)

20) Kupfer G. (1985)
Kammer-Kalibrierung in der Photogrammetrischen Praxis - eine Einführung.
Deutsche Geodätische Kommission.
ISBN 3 7696 8562 8.(1985)

21) Lo C.P. (1986)
Applied Remote Sensing.
Longman Scientific & Technical
Longman Group UK Limited
ISBN 0-582_30132-7 (1986)

22) Meier H.-K. (1985)
Uber die Genauigkeit von Laborkaliebrierungen und die Stabilität von Luftbildkammern am Beispiel der Spacelab-RMK A 30/23.
Deutsche Geodätische Kommission.
ISBN 3 7696 8562 8.(1985)

23) Perry L.H. (1978)
An NOS-developed backlighted reseau and its effect on co-ordinate refinement.
Proceedings of the American Society of Photogrammetry, Albuquerque
New Mexico, Oct 1978.

24) Richard M. (1988)
<u>La photogrammetrie industrielle.</u>
Géomètre No.10:36-38. (1988)

25) Rüther H. Doctoral thesis UCT

26) Rüther H.,Adams L.P. (1988)
<u>Lecture Notes University of Cape</u>
Fifth year course in surveying -
Close range photogrammetry.
University of Cape Town. (1988)

27) Schmid H. (1956/1957)
<u>An analytical treatment of the problem of triangulation by stereophotogrammetry.</u>
Photogrammetria Nr.3 (1956/1957)

28) Schöler H. (1979)
<u>On photogrammetric distortion.</u>
S.A. Survey Review
Vol.1 No.1: 1-6 (July 1979)

29) Van Gysen H. (1989)
Least squares algorithms for real-time surveying.
Proceedings of conference of Southern African Surveyors 1989.
Volume 2 Paper No.13.1. (1989)

30) Ziemann H. (1985)
Bewertung von Systemkalibrierung und Selbstkalibrierungverfahren für Senkrechtbilder über Flachem Gelände bei Ausschliesslicher Verwendung Gegebener Passpunkte.
Deutsche Geodätische Kommission.
ISBN 3 7696 8562 8.(1985)

31) Zinndorf S. (1986)
Optimierung der Photogrammetrischen Aufnahmeanordnung.
Deutsche Geodätische Kommission.
ISBN 3 7696 9373 6.(1986)

Acknowledgements:

I thank the department of surveying at the university of Cape Town most sincerely and in particular Prof. H. Rüther for the support and patience in allowing me to finish this half thesis. It was at times very difficult to fit this research in with the busy schedules of lecturing, private practice and the trying times the Survey Department has been going through. It is a tribute to Prof. Rüther's determination that this half thesis got written at all.

I also thank my mentors past and present in the survey department at UCT. The values and knowledge you impart so freely is a benefit to us all. The service to our community is immense.

Most of all I convey my sincerest thank you and love to my wife Lindy for her tremendous patience and tolerance of the time and efforts I had to put in this project.

October 1990.

ANNEXURE

PROGRAM LISTING

```
1000 ! BUNDLE ADJUSTMENT C.ROMMELAERE  Program name:----------BUNBAS----
     --
1010 OPTION ANGLE DEGREES
1020 DECLARE DEF ASIN
1030 DEF ASIN (AS)= ATN(AS/SQR(1-AS^2))
1040 INPUT PROMPT "WHICH DISC DRIVE FOR DATA? eg A": DRIVE$
1050 LET DRIVE$=DRIVE$&":"
1060 !
1070 !   CAMERA PROVISIONAL ORIENTATION PARAMETERS a,b,c,Xo,Yo,Zo,f
1080 !-----------------------------------------------------------------
     ------
1090 DIM Par(10,4),Nuc(4)
1100 LET N=0
1110 INPUT PROMPT "NAME FILE WITH CAMERA PARAMETERS " :CAM$
1120 PRINT
1130 PRINT        "CAM/STAT  a    b    c     Xo    Yo    Zo    f    PP:x,y"
1140 LET N=0
1150 OPEN #1:NAME DRIVE$ & CAM$,CREATE OLD, ACCESS INPUT, ORGANIZATION R
       ECORD,RECSIZE 8
1160 DO WHILE MORE #1
1170    LET N=N+1
1180    PRINT "CAM/STAT No";N,
1190    FOR J = 1 TO 10
1200        READ #1 : Par(J,N)
1210        PRINT Par(J,N);J,N
1220    NEXT J
1230 LOOP
1240 CLOSE #1
1250 LET NI=N                          !No OF IMAGES
1260 IF NI>1 THEN 1290
1270 PRINT "MUST BE 2 OR MORE CAMERA STATIONS RESTART PROGRAM!"
1280 STOP
1290 DIM MPAR(10,20)                   !WORKS OUT ROTATION MATRIX-RODRIGUES P
       ARM.
1300 MAT REDIM MPAR(10,NI)
1310 FOR I=1 TO NI
1320     LET D=1/(1+Par(2,I)^2+Par(3,I)^2+Par(4,I)^2)      !1+a^2+b^2+c^
       2
1330     LET MPAR(1,I)=(1+Par(2,I)^2-Par(3,I)^2-Par(4,I)^2)/D   !a11/D
1340     LET MPAR(2,I)=(2*Par(2,I)*Par(3,I)-2*Par(4,I))/D  !a12/D
1350     LET MPAR(3,I)=(2*Par(2,I)*Par(4,I)+2*Par(3,I))/D  !a13/D
1360     LET MPAR(4,I)=(2*Par(2,I)*Par(3,I)+2*Par(4,I))/D  !a21/D
1370     LET MPAR(5,I)=(1-Par(2,I)^2+Par(3,I)^2-Par(4,I)^2)/D   !a22/D
1380     LET MPAR(6,I)=(2*Par(3,I)*Par(4,I)-2*Par(2,I))/D  !a23/D
1390     LET MPAR(7,I)=(2*Par(2,I)*Par(4,I)-2*Par(3,I))/D  !a31/D
1400     LET MPAR(8,I)=(2*Par(3,I)*Par(4,I)+2*Par(2,I))/D  !a32/D
1410     LET MPAR(9,I)=(1-Par(2,I)^2-Par(3,I)^2+Par(4,I)^2)/D   !a33/D
1420     LET MPAR(10,I)=Par(1,I)  !IMAGE No.
1430 NEXT I
1440 DIM X(20),Y(20),Z(20),C(20)  !CONTROL POINTS
1450 DIM XC(20,1),YC(200,1),ZC(20,1),CC(20,4),CCC(20,2)    !CONTROL POIN
       TS ON IMAGE
1460 DIM L(20,1),WX(20),WY(20),WW(2,1)       !MATRICES FOR LATER REDIM
1470 DIM A(2,3),B(2,6),AT(3,2),BT(6,2),NN(6,6),TT(3,1)
1480 DIM DFXX(15),DFXY(15),DFXZ(15),DFXA(15),DFXB(15),DFXC(15),MM(3,6),M
```

```
      MT(6,3)
1490 DIM DFYX(15),DFYY(15),DFYZ(15),DFYA(15),DFYB(15),DFYC(15),UU(6,1),U
      UT(1,6)
1500 !
1510 !READING IN CONTROL POINTS FROM OUTSIDE FILES PER NO ONLY.
1520 !-------------------------------------------------------------------
      -------
1530 SOUND 600,.2
1540 INPUT PROMPT "CONTROL PTS NO. X Y Z      WHICH FILE? ":CONT$
1550 PRINT       "              Space Co-ords ----CONTROL POINTS"
1560 PRINT       "                        X         Y         Z "
1570 LET NC=0
1580 OPEN #2:NAME DRIVE$ & CONT$,CREATE NEWOLD,ORGANIZATION RECORD,ACCES
      S INPUT,RECSIZE 8
1590 DO WHILE MORE #2
1600    LET NC=NC+1
1610    READ #2:C(NC),X(NC),Y(NC),Z(NC)
1620    LET X(NC)=X(NC)/100
1630    LET Y(NC)=Y(NC)/100        !ARTIFICIAL SCALE FACTOR
1640    LET Z(NC)=Z(NC)/100
1650    PRINT USING "        ---# -----%.%%% ----%.%%% ----%.%%% ":NC,X(
      NC),Y(NC),Z(NC)
1660 LOOP
1670 MAT REDIM X(NC),Y(NC),Z(NC),C(NC)
1680 !
1690 !READING IN x y OF CONTROL IN THE IMAGES.
1700 !-------------------------------------------------------------------
      ---------
1710 SOUND 800,.2
1720 INPUT PROMPT "FILE NAMES OF IMAGES WITH CONTROL x,y ":FILE$
1730 MAT REDIM Nuc(NI)
1740 CALL REIMCON                  !if N matrix is required per photo do
      it here before re-ordering per point
1750 LET NOBS=N
1760 MAT REDIM CCC(N,2)
1770 !                           a SHELL-METZNER SORT REF.PROGRAMMING TE
      CHNIQUES LEVEL 2 BASIC: W. BARDEN
1780 ! SORT OF CC  IN ASCENDING ORDER OF OBS CONTROL POINTS VS CAMERA ST
      ATIONS
1790 !-------------------------------------------------------------------
      -----------
1800 LET M=NOBS
1810 OPEN #6:PRINTER
1820 FOR I=1 TO M                  !CROSS REFERENCE LIST IMAGE vs POINTS
1830     LET CCC(I,1)=CC(I,1)
1840     LET CCC(I,2)=CC(I,2)
1850 NEXT I
1860 PRINT #6:"CC"
1870 MAT PRINT #6: CC
1880 LET M=INT (M/2)
1890 IF M=0 THEN 2080
1900 FOR ST=1 TO M
1910     LET I=ST
1920     LET J=ST+M
1930     LET SW=0
```

```
1940      IF CC(I,2) <= CC(J,2) THEN 2010
1950      LET SW=1
1960      FOR K=1 TO 4
1970          LET SW1=CC(I,K)
1980          LET CC(I,K)=CC(J,K)
1990          LET CC(J,K)=SW1
2000      NEXT K
2010      LET I=J
2020      LET J=J+M
2030      IF J < NOBS+1 THEN 1940
2040      IF SW=0 THEN 2060
2050      GOTO 1910
2060 NEXT ST
2070 GOTO 1880
2080 LET M=NOBS                    ! SORT OF IMAGES IN ASCENDING ORDER
2090 LET M=INT (M/2)
2100 IF M=0 THEN 2290
2110 FOR ST=1 TO M
2120      LET I=ST
2130      LET J=ST+M
2140      LET SW=0
2150      IF CCC(I,1) <= CCC(J,1) THEN 2220
2160      LET SW=1
2170      FOR K=1 TO 2
2180          LET SW1=CCC(I,K)
2190          LET CCC(I,K)=CCC(J,K)
2200          LET CCC(J,K)=SW1
2210      NEXT K
2220      LET I=J
2230      LET J=J+M
2240      IF J < NOBS+1 THEN 2150
2250      IF SW=0 THEN 2270
2260      GOTO 2120
2270 NEXT ST
2280 GOTO 2090
2290 PRINT #6: "CCC MATRIX"
2300 MAT PRINT #6:CCC
2310 !
2320 !SETTING UP Wx AND Wy ELEMENTS OF A & B MATRIX,FROM DATA READ IN
2330 !-----------------------------------------------------------
        ---
2340 DIM LI(3,3),NNJ(6,6),MMIJ(3,6),TTI(3,1),UUJ(6,1)
2350 MAT REDIM WX(NOBS),WY(NOBS)
2360 MAT REDIM DFXX(NOBS),DFYX(NOBS),DFXY(NOBS),DFYY(NOBS),DFXZ(NOBS),DF
        YZ(NOBS)
2370 MAT REDIM DFXA(NOBS),DFXB(NOBS),DFXC(NOBS)
2380 MAT REDIM DFYA(NOBS),DFYB(NOBS),DFYC(NOBS)
2390 DIM QX(2,2),QY(3,3),QZ(6,6)
2400 !SETTING UP A,B MATRICES
2410 !SETS UP WHERE POINT NO CHANGES WITHIN OBSERVATIONS LISTED IN CC-gr
        oups
2420 DIM CHAN(15)
2430 LET CHAN(1)=1
2440 LET M=1
2450 PRINT #6: "SORTED MATRIX CC";     !sorted in ascending order of pts
```

```
        vs. image
2460 MAT PRINT #6: CC
2470 FOR I=1 TO NOBS-1
2480      IF CC(I,2)=CC(I+1,2) THEN 2510
2490      LET M=M+1
2500      LET CHAN(M)=I+1
2510 NEXT I
2520 LET CHAN(M+1)=NOBS+1
2530 MAT REDIM CHAN(M+1)
2540 FOR I=1 TO M
2550      IF CHAN(I+1)-CHAN(I)>1 THEN 2700
2560      SOUND 2000,.5
2570      LET I1=CHAN(I)
2580      PRINT #6: "POINT";CC(I1,2); "IS ONLY OBSERVED ON PHOTO"; CC(I1,
     1);"REJECT THIS OBSERVATION"
2590      FOR K=I TO NOBS-1          !ELIMINATES OFFENDING SINGLE OBS.
2600          FOR J=1 TO 4
2610              LET CC(K,J)=CC(K+1,J)
2620          NEXT J
2630      NEXT K
2640      FOR K=I TO M
2650          LET CHAN(I)=CHAN(I+1)-1
2660      NEXT K
2670      LET NOBS=NOBS-1
2680      LET M=M-1                  ! REDUCE No of observations by 1 for r
     ejected obs.
2690      GOTO 2550
2700 NEXT I
2710 MAT REDIM CHAN(M+1)
2720 CALL WEx
2730 MAT LI=ZER                      !SETTING UP INITIAL MATRICES
2740 MAT NNJ=ZER
2750 MAT NNJ=ZER
2760 MAT MMIJ=ZER
2770 MAT TTI=ZER
2780 MAT UUJ=ZER
2790 MAT QX=IDN
2800 MAT QY=IDN
2810 MAT QZ=IDN
2820 !N= No of obs.      I=Group of point       .J=No of times point app
     ears
2830 CLOSE #1
2840 LET N=1
2850 FOR I=1 TO NOBS
2860      FOR N=1 TO NOBS
2870          IF CCC(I,1)=CC(N,1) AND CCC(I,2)=CC(N,2) THEN
2880             LET B(1,1)=DFXA(N)
2890             LET B(1,2)=DFXB(N)
2900             LET B(1,3)=DFXC(N)
2910             LET B(1,4)=-DFXX(N)    !CREATES Bij TERMS pt i im j
2920             LET B(1,5)=-DFXY(N)
2930             LET B(1,6)=-DFXZ(N)
2940             LET B(2,1)=DFYA(N)
2950             LET B(2,2)=DFYB(N)
2960             LET B(2,3)=DFYC(N)
```

```
2970             LET B(2,4)=-DFYX(N)
2980             LET B(2,5)=-DFYY(N)
2990             LET B(2,6)=-DFYZ(N)
3000             CLOSE #1
3010             OPEN #1:NAME "B"&STR$(CCC(I,2))& STR$(CCC(I,1)),CREATE N
        EWOLD
3020             MAT WRITE #1:B
3030             CLOSE #1
3040             PRINT #6:"POINT"; CCC(I,2),"IMAGE";CCC(I,1)
3050             PRINT #6:"B-TERMS"
3060             MAT PRINT #6:B
3070             MAT BT=TRN(B)
3080             LET O=CC(N,1)      !IMAGE No
3090             MAT NN=BT*QX
3100             MAT NN=NN*B
3110             MAT PRINT #6:NN
3120             IF I=NOBS THEN 3180
3130             IF CCC(I,1)<>CCC(I+1,1) THEN GOTO 3180
3140             MAT NNJ=NNJ+NN
3150             PRINT #6:"NNJ";"IM";O;"PT";CCC(I,2)
3160             MAT PRINT #6:NNJ
3170             GOTO 3290
3180             MAT NNJ=QZ+NNJ
3190             MAT NNJ=NNJ+NN
3200             OPEN #1:NAME "NJ"&STR$(O),CREATE NEWOLD
3210             MAT WRITE #1:NNJ  !CREATES STORES NJ
3220             CLOSE #1
3230             PRINT #6: "NNJ"&STR$(O)
3240             MAT PRINT #6:NNJ
3250             MAT NNJ=ZER
3260          ELSE
3270             GOTO 3290
3280          END IF
3290      NEXT N
3300 NEXT I
3310 LET N=1
3320 FOR I=1 TO M                    !No of groups each point as appears on
          photo
3330      LET K= CHAN(I+1)-CHAN(I)      !No of times a point appears on a
          diff photo
3340      FOR J=1 TO K
3350          LET I1=CC(N,2)        !Pt no i
3360          LET O=CC(N,1)         !Im no j
3370          PRINT #6:"IMAGE";O,"POINT";I1
3380          LET A(1,1)=DFXX(N)
3390          LET A(1,2)=DFXY(N)
3400          LET A(1,3)=DFXZ(N)
3410          LET A(2,1)=DFYX(N)    !Aij  terms individually for point i i
        mage j
3420          LET A(2,2)=DFYY(N)
3430          LET A(2,3)=DFYZ(N)
3440          PRINT #6: "A-TERMS";"PT";CC(N,2);"PHOTO";CC(N,1)
3450          MAT PRINT #6: A
3460          MAT AT=TRN(A)
3470          CLOSE #1
```

```
3480          OPEN #1:NAME "AT"&STR$(I1)&STR$(CC(N,1)),CREATE NEWOLD
3490          MAT WRITE #1 :AT      !ATij MATRIX SAVED FOR LATER pti imj
3500          CLOSE #1
3510          MAT L=AT*QX
3520          MAT L=L*A
3530          MAT LI=LI+L           !For each pt i there is a LI
3540          MAT MM=AT*QX
3550          OPEN #1:NAME "B"&STR$(I1)&STR$(CC(N,1)),ACCESS INPUT
3560          MAT READ #1:B
3570          CLOSE #1
3580          MAT MM=MM*B
3590          PRINT #6:"MM","PT";I1,"IM";O
3600          MAT PRINT #6:MM
3610          OPEN #1:NAME "MMIJ"&STR$(I1)&STR$(O),CREATE NEWOLD
3620          MAT WRITE #1:MM
3630          CLOSE #1
3640          LET WW(1,1)=WX(N)
3650          LET WW(2,1)=WY(N)
3660          MAT TT=AT*QX
3670          MAT TT=TT*WW
3680          PRINT #6:"TT",I1,O
3690          MAT PRINT #6:TT
3700          MAT TTI=TTI+TT
3710          MAT BT=TRN(B)
3720          MAT UU=BT*QX
3730          MAT UU=UU*WW
3740          MAT UUJ=UUJ+UU
3750          OPEN #1:NAME "TTI"&STR$(I1),CREATE NEWOLD
3760          MAT WRITE #1: TTI
3770          CLOSE #1
3780          PRINT #6:"TTI"&STR$(I1)
3790          MAT PRINT #6: TTI
3800          OPEN #1:NAME "UUIJ"&STR$(I1)&STR$(O),CREATE NEWOLD
3810          MAT WRITE #1: UUJ
3820          CLOSE #1
3830          PRINT #6:"UUIJ"&STR$(I1)&STR$(O)
3840          MAT PRINT #6: UUJ
3850          LET N=N+1
3860      NEXT J
3870      MAT LI=QY+LI
3880      OPEN #1:NAME "LI"&STR$(I1),CREATE NEWOLD
3890      MAT WRITE #1:LI               !LI MATRIX OF POINT i
3900      CLOSE #1
3910      PRINT #6: "LI"&STR$(I1)
3920      MAT PRINT #6: LI
3930      MAT TTI=ZER                   !Reset TTI=0 for next group
3940      MAT LI=ZER                    !Reset LI=0 for next group
3950 NEXT I
3960 DIM MT(6,3),PIJ(6,6),PIJI(6,6),LT(3,1),SI(6,1)
3970 MAT MMIJ=ZER
3980 MAT MM=ZER
3990 MAT L=ZER
4000 MAT LI=ZER
4010 MAT UU=ZER
4020 MAT UUJ=ZER
```

```
4030 !WORKING OUT UJ TERMS
4040 FOR I=1 TO NOBS
4050     LET O=CCC(I,1)              !Image no
4060     LET I1=CCC(I,2)             !Pt no
4070     OPEN #1:NAME "UUIJ"&STR$(I1)&STR$(O),ACCESS INPUT
4080     MAT READ #1:UU
4090     CLOSE #1
4100     MAT UUJ=UUJ+UU
4110     PRINT #6: "CUMUL UJ";"IMAGE";O;"PT";I1
4120     MAT PRINT #6: UUJ
4130     OPEN #1:NAME"UJ"&STR$(O),CREATE NEWOLD
4140     MAT WRITE #1:UUJ
4150     CLOSE #1
4160     IF I=NOBS THEN 4190
4170     IF O<>CCC(I+1,1) THEN 4180 ELSE 4190
4180     MAT UUJ=ZER                 !Reset UJ for next group
4190 NEXT I
4200 !INITIALIZING PIJ MATRICES=O
4210 FOR I= 1 TO NI
4220     FOR J= 1 TO NI
4230         MAT PIJ=ZER
4240         OPEN #1: NAME "PIJ"&STR$(I)&STR$(J),CREATE NEWOLD
4250         MAT WRITE #1:PIJ
4260         CLOSE #1
4270     NEXT J
4280 NEXT I
4290 !WORKING OUT PIJ MATRICES
4300 FOR I=1 TO NOBS
4310     FOR J=1 TO NOBS
4320         IF CCC(I,2)=CCC(J,2) THEN
4330             OPEN #1:NAME "MMIJ"&STR$(CCC(I,2))&STR$(CCC(I,1)),ACCESS
         INPUT
4340             MAT READ #1:MM
4350             CLOSE #1
4360             MAT MT=TRN(MM)
4370             OPEN #1:NAME "LI"&STR$(CCC(I,2)),ACCESS INPUT
4380             MAT READ #1:L
4390             CLOSE #1
4400             MAT LI=INV(L)
4410             MAT PIJ=MT*LI
4420             !CREATING SI MATRIX FROM MTLT
4430             OPEN #1 :NAME"TTI"&STR$(CCC(I,2)),ACCESS INPUT
4440             MAT READ #1:TTI
4450             CLOSE #1
4460             MAT SI=PIJ*TTI
4470             OPEN #1:NAME"UJ"&STR$(CCC(I,1)),ACCESS INPUT
4480             MAT READ #1:UUJ
4490             CLOSE#1
4500             MAT SI=UUJ-SI
4510             OPEN #1:NAME "SI"&STR$(CCC(I,2)),CREATE NEWOLD
4520             MAT WRITE #1:SI
4530             CLOSE #1
4540             !CREATING PIJ MATRIX N-MTLM
4550             OPEN #1:NAME "MMIJ"&STR$(CCC(J,2))&STR$(CCC(J,1)),ACCESS
         INPUT
```

```
4560            MAT READ #1: MM
4570            CLOSE #1
4580            MAT PIJ=PIJ*MM
4590            OPEN #1: NAME "PIJ"&STR$(CCC(I,1))&STR$(CCC(J,1)),ACCESS
       OUTIN,CREATE NEWOLD
4600            MAT READ #1:PIJI
4610            CLOSE #1
4620            MAT PIJ=PIJ+PIJI
4630            IF CCC(I,1)=CCC(J,1) THEN
4640               OPEN #1:NAME "NJ"&STR$(CCC(I,1)),ACCESS INPUT
4650               MAT READ #1:NNJ
4660               CLOSE #1
4670               MAT PIJ=NNJ-PIJ
4680               GOTO 4720
4690            ELSE
4700               GOTO 4720
4710            END IF
4720            OPEN #1:NAME "PIJ"&STR$(CCC(I,1))&STR$(CCC(J,1)),CREATE
      NEWOLD
4730            MAT WRITE #1:PIJ
4740            CLOSE#1
4750            PRINT "POINT"; CCC(I,2);"IMAGE1"; CCC(I,1);"IMAGE2"; CCC
      (J,1)
4760            PRINT  "PIJ"
4770            MAT PRINT  PIJ
4780            PRINT "SI"
4790            MAT PRINT  SI
4800            GOTO 4840
4810         ELSE
4820            GOTO 4840
4830         END IF
4840      NEXT J
4850 NEXT I
4860 DIM PF(30,30),PFI(30,30),S(30),ZI(30)
4870 MAT REDIM PF(NI*6,NI*6),PFI(NI*6,NI*6),S(NI*6),ZI(NI*6)
4880 FOR I=1 TO NI
4890      FOR J=1 TO NI
4900          OPEN #1:NAME"PIJ"&STR$(I)&STR$(J),ACCESS INPUT
4910          MAT READ #1:PIJ
4920          CLOSE #1
4930          FOR K=1 TO 6
4940              FOR N=1 TO 6
4950                  LET PF(6*(I-1)+K,6*(J-1)+N)=PIJ(K,N)
4960              NEXT N
4970          NEXT K
4980      NEXT J
4990 NEXT I
5000 MAT PFI=INV(PF)
5010 MAT PRINT PF,PFI
5020 FOR I=1 TO NI
5030      OPEN #1:NAME "SI"&STR$(I),ACCESS INPUT
5040      MAT READ #1:SI
5050      CLOSE #1
5060      FOR K=1 TO 6
5070          LET S(6*(I-1)+K) =-1* SI(K,1)
```

```
5080     NEXT K
5090 NEXT I
5100 MAT PRINT S
5110 MAT ZI = PFI*S
5120 LET IT = IT +1                     !ITERATION COUNTER
5130 PRINT #6: "CAMERA ORIENTATION PARAMETERS Z-MATRIX "
5140 PRINT #6: "ITERATION No.";IT
5150 MAT PRINT #6:ZI
5160 LET N=1
5170 FOR I = 1 TO NI
5180     FOR J = 1 TO 6
5190         LET Par (J+1,NI) = ZI(N)
5200         LET N=N+1
5210     NEXT J
5220 NEXT I
5230 FOR I=1 TO NI
5240     LET D=1/(1+Par(2,I)^2+Par(3,I)^2+Par(4,I)^2)       !1+a^2+b^2+c^
    2
5250     LET MPAR(1,I)=(1+Par(2,I)^2-Par(3,I)^2-Par(4,I)^2)/D   !a11/D
5260     LET MPAR(2,I)=(2*Par(2,I)*Par(3,I)-2*Par(4,I))/D  !a12/D
5270     LET MPAR(3,I)=(2*Par(2,I)*Par(4,I)+2*Par(3,I))/D  !a13/D
5280     LET MPAR(4,I)=(2*Par(2,I)*Par(3,I)+2*Par(4,I))/D  !a21/D
5290     LET MPAR(5,I)=(1-Par(2,I)^2+Par(3,I)^2-Par(4,I)^2)/D   !a22/D
5300     LET MPAR(6,I)=(2*Par(3,I)*Par(4,I)-2*Par(2,I))/D  !a23/D
5310     LET MPAR(7,I)=(2*Par(2,I)*Par(4,I)-2*Par(3,I))/D  !a31/D
5320     LET MPAR(8,I)=(2*Par(3,I)*Par(4,I)+2*Par(2,I))/D  !a32/D
5330     LET MPAR(9,I)=(1-Par(2,I)^2-Par(3,I)^2+Par(4,I)^2)/D   !a33/D
5340     LET MPAR(10,I)=Par(1,I)  !IMAGE No.
5350 NEXT I
5360 GO TO 2720
5370 STOP
5380 SUB WEx
5390     LET M=0
5400     FOR I = 1 TO NOBS
5410         LET M=0
5420         LET CAM=CC(I,1)
5430         LET M=M+1
5440         IF M > NC THEN 5710  !LOOKS FOR CORRESPONDING POINT IN C(M)
5450         IF CC(I,2) <> C(M) THEN 5430   !PT <> TO CONTROL ESCAPE
5460         FOR J= 1 TO NI
5470             IF CAM <> MPAR(10,J) THEN 5700  !CAMERA WITH SAME POINT
     NO
5480             LET DX=X(M)-Par(5,J)  !                         X-Xo
5490             LET DY=Y(M)-Par(6,J)  !                         Y-Yo
5500             LET DZ=Z(M)-Par(7,J)  !                         Z-Zo
5510             LET XB=MPAR(1,J)*DX+MPAR(2,J)*DY+MPAR(3,J)*DZ  !Xbar
5520             LET YB=MPAR(4,J)*DX+MPAR(5,J)*DY+MPAR(6,J)*DZ  !Ybar
5530             LET ZB=MPAR(7,J)*DX+MPAR(8,J)*DY+MPAR(9,J)*DZ  !Zbar
5540             LET WX(I)=Par(8,J)*XB/ZB-CC(I,3)      !Wx
5550             LET WY(I)=Par(8,J)*YB/ZB-CC(I,4)      !Wy
5560             LET FZ2=Par(8,J)/ZB^2
5570             LET DFXX(I)=-(FZ2)*(-MPAR(1,J)*ZB+MPAR(7,J)*XB)      ! d
    Fx/dX
5580             LET DFXY(I)=-(FZ2)*(-MPAR(2,J)*ZB+MPAR(8,J)*XB)      ! d
    Fx/dY
```

```
5590              LET DFXZ(I)=-(FZ2)*(-MPAR(3,J)*ZB+MPAR(9,J)*XB)      ! d
         Fx/dZ
5600              LET DFYX(I)=-(FZ2)*(-MPAR(4,J)*ZB+MPAR(7,J)*YB)      ! d
         Fy/dX
5610              LET DFYY(I)=-(FZ2)*(-MPAR(5,J)*ZB+MPAR(8,J)*YB)      ! d
         Fy/dY
5620              LET DFYZ(I)=-(FZ2)*(-MPAR(6,J)*ZB+MPAR(9,J)*YB)      ! d
         Fy/dZ
5630              LET DFXA(I)=2*FZ2*((Par(2,J)*DX+Par(3,J)*DY+Par(4,J)*DZ
         )*ZB-(Par(4,J)*DX+DY-Par(2,J)*DZ)*XB)      !  dFx/da
5640              LET DFXB(I)=2*FZ2*((-Par(3,J)*DX+Par(2,J)*DY+DZ)*ZB-(-D
         X+Par(4,J)*DY-Par(3,J)*DZ)*XB)  ! dFx/db
5650              LET DFXC(I)=2*FZ2*((-Par(4,J)*DX-DY+Par(2,J)*DZ)*ZB-(Pa
         r(2,J)*DX+Par(3,1)*DY+Par(4,J)*DZ)*XB)      !  dFx/dc
5660              LET DFYA(I)=2*FZ2*((Par(3,J)*DX-Par(2,J)*DY-DZ)*ZB-(Par
         (4,J)*DX+DY-Par(2,J)*DZ)*YB)      !  dFy/da
5670              LET DFYB(I)=2*FZ2*((Par(2,J)*DX+Par(3,J)*DY+Par(4,J)*DZ
         )*ZB-(-DX+Par(4,J)*DY-Par(3,J)*DZ)*YB)      !  dFy/db
5680              LET DFYC(I)=2*FZ2*((DX-Par(4,J)*DY+Par(3,J)*DZ)*ZB-(Par
         (2,J)*DX+Par(3,J)*DY+Par(4,J)*DZ)*YB)      !dFy/dc
5690              PRINT  CC(I,1),CC(I,2),J,I
5700          NEXT J
5710      NEXT I
5720 END SUB
5730 !
5740 SUB REIMCON
5750      PRINT
5760      PRINT  " IMAGE   POINT        X          Y   "
5770      PRINT
5780      LET N=0
5790      OPEN # 3:NAME DRIVE$ & FILE$,create NEWOLD, organization record
         ,ACCESS INPUT,RECSIZE 8
5800      DO WHILE MORE #3
5810         LET N=N+1
5820         READ #3 :J,I,K,KK
5830         LET CC(N,1)=J
5840         LET CC(N,2)=I
5850         LET CC(N,3)=K
5860         LET CC(N,4)=KK
5870         PRINT USING  "   -##  -##   -----.###  -----.###" : J,I,K,KK
5880         IF N => 50 THEN 5910
5890      LOOP
5900      MAT REDIM CC(N,4)
5910      CLOSE #3
5920 END SUB
5930 END
```