REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	18 Dec 2001	Final 22 June 2000 – 31 Dec 2000

4. TITLE AND SUBTITLE

Missile Trainer Workstation Enhancement

5. FUNDING NUMBERS

6. AUTHORS

Stephen J. Dow

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

The University of Alabama in Huntsville
Huntsville, AL 35899

8. PERFORMING ORGANIZATION REPORT NUMBER

5-21046

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

U.S. Army Aviation & Missile Command
Redstone Arsenal, AL 35898

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unlimited

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Terrain models used in training software for several missile systems are built using photogrammetric algorithms in an application program called Range Finder. The initial version of Range Finder, used to create the first set of terrain models for the Javelin training software in late 1999 and early 2000, relied on creating a terrain model in sections, one stereo pair at a time, and then merging these separate models into the completed model for the full panoramic scene. Since that time the Range Finder software has been restructured to handle a complete block of photos together, resulting in a more efficient workflow, improved accuracy of the final terrain model, and better visual quality of the stitched image. This report documents the mathematical methods and some of the operational details of this new block-oriented version of the software.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Terrain modeling, photogrammetry, block adjustment	10
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

PLEASE CHECK THE APPROPRIATE BLOCK BELOW

DAO# ____________

☐ ____________ copies are being forwarded. Indicate whether Statement A, B, C, D, E, F, or X applies.

☑ DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

☐ DISTRIBUTION STATEMENT B:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES ONLY; (indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT C:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND THEIR CONTRACTS (Indicate Reason and Date). OTHER REQUESTS FOR THIS DOCUMENT SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT D:
DISTRIBUTION AUTHORIZED TO DoD AND U.S. DoD CONTRACTORS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT E:
DISTRIBUTION AUTHORIZED TO DoD COMPONENTS ONLY; (Indicate Reason and Date). OTHER REQUESTS SHALL BE REFERRED TO (Indicate Controlling DoD Office).

☐ DISTRIBUTION STATEMENT F:
FUTHER DISSEMINATION ONLY AS DIRECTED BY (Indicate Controlling DoD Office and Date) or HIGHER DoD AUTHORITY.

☐ DISTRIBUTION STATEMENT X:
DISTRIBUTION AUTHORIZED TO U.S. GOVERNMENT AGENCIES AND PRIVATE INDIVIDUALS OR ENTERPRISES ELIGIBLE TO OBTAIN EXPORT-CONTROLLED TECHNICAL DATA IN ACCORDANCE WITH DoD DIRECTIVE 5230.25. WITHHOLDING OF UNCLASSIFIED TECHNICAL DATA FROM PUBLIC DISCLOSURE, 6 Nov 1984 (indicate date of determination). CONTROLLING DoD OFFICE IS (Indicate Controlling DoD Office).

☐ This document was previously forwarded to DTIC on ____________ (date) and the AD number is ____________.

☐ In accordance with provisions of DoD instructions. The document requested is not supplied because:

☐ It will be published at a later date. (Enter approximate date, if known).

☐ Other. (Give Reason)

DoD Directive 5230.24, "Distribution Statements on Technical Documents," 18 Mar 87, contains seven distribution statements, as described briefly above. Technical Documents must be assigned distribution statements.

Stephen J. Dow
Print or Type Name

Stephen J Dow 12/19/01
Authorized Signature/Date

256-824-6406
Telephone Number

Missile Trainer Workstation Enhancement

Stephen J. Dow
Department of Mathematical Sciences
The University of Alabama in Huntsville

December 18, 2001

Final Report

F/DOD/ARMY/AMCOM/Missile Trainer Workstation Enhancement
DAAH01-97-D-R005 D.O. 27

Period of Performance: 6/22/00 to 12/31/00

1. Introduction

The missile trainer workstation is a set of software tools used to create terrain models and target paths for use in training software used for several missile systems. Terrain models are built using photogrammetric algorithms in an application program called Range Finder. The initial version of Range Finder, used to create the first set of terrain models for the Javelin training software in late 1999 and early 2000, relied on creating a terrain model in sections, one stereo pair at a time, and then merging these separate models into the completed model for the full panoramic scene. A previous report [1] by the present author documented that earlier version of Range Finder. Since that time the Range Finder software has been restructured to handle a complete block of photos together, resulting in a more efficient workflow, improved accuracy of the final terrain model, and better visual quality of the stitched image. This report documents the mathematical methods and some of the operational details of this new block-oriented version of Range Finder.

2. Camera Geometry

In using the training software, the gunner sees a simulation of what he or she would see through the actual weapon system, consisting of photographic terrain imagery within which rendered targets and other graphics are inserted. The gunner views the scene from a single vantage point, but can change the line-of-site across a 40 - 60 degree horizontal field-of-view. To create a terrain image covering this field-of-view at the necessary resolution, the workstation software must stitch together several photographs, taken from a single location, rotating the camera between shots. In addition to creating this panoramic image, the software creates a ground model using stereo photogrammetry, which requires that for each of the photos taken at the initial location, another photo covering roughly the same area of the ground is taken from another camera location. The complete set of overlapping photos taken for the purpose of building a single terrain model and panoramic image is called a block. The camera locations are called camera stations; more precisely, the camera station is the location of the center of projection of the camera, where rays of light entering the camera lens meet in a common point. We assume that in rotating the camera on a tripod, the center of projection is on the axis of rotation (i.e. does not move), so that each placement of the tripod corresponds to a single camera station. The initial camera station, where the photos used in constructing the panoramic image are taken, is called the primary camera station,and the others secondary camera stations. For the terrain data collected so far, the secondary camera stations were chosen to be approximately 100 feet from the primary camera station. An example of the typical camera geometry is shown below in Figure 1. In this example there are two camera stations, and three photos taken from each station. The geometry is shown as seen from above, with rays representing the camera lines-of-sight.

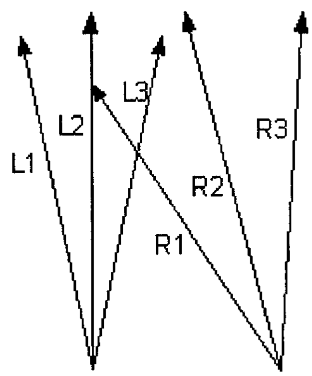


Figure 1: Typical Camera Geometry

The labeling of the rays indicates the association of the photos into three stereo pairs L1-R1, L2-R2, and L3-R3. We call these stereo pairs models; technically any two photos having distinct camera stations within the block may be designated as a model; however we have been forming one model containing each photo at the primary camera station. For a given model, the base line between the two camera stations should be not too far from perpendicular to the lines-of-sight of the two photos (which are themselves within a few degrees of being parallel). Thus if the station on the left in Figure 1 is the primary camera station, and the block is augmented with several more photos from that station to create a larger horizontal field-of-view panorama, it will become necessary at some point to use a third camera station, below the two shown, to maintain the desired geometry for those subsequent models. Estimates of the camera station positions relative to the primary camera station are needed as inputs, so this information should be taken down at the time the pictures are taken; at a minimum the distances between the camera stations are needed.

The photographs are taken with a high-end digital camera. Coordinates within the image file produced by the camera are called image or pixel coordinates; the pixel at the upper left corner of the image has coordinates (0, 0) and the pixel at the lower right corner coordinates (n_c - 1, n_r - 1), where n_c is the number of horizontal columns, n_r the number of vertical rows. Each photograph also defines a 3D camera coordinate system oriented about the camera as it was situated when that photo was taken, having the camera center of projection at the origin, x-axis to the right, y-axis down, and positive z-axis forward along the line of sight of the camera. Units in this coordinate system are meters. Let (x_c, y_c, z_c) denote the camera coordinates of a physical location in the scene. The line joining this point and the origin passes through the plane $z_c = 1$ at $(x', y', 1)$, where $x' = x_c/z_c$, $y' = y_c/z_c$. Coordinates (x', y') are called photo coordinates. We also use the term 3D photo coordinates interchangeably with camera coordinates; thus 3D photo coordinates are converted to 2D photo coordinates by dividing by the z-coordinate.

Photo coordinates (x', y') are related to image coordinates (x, y) through the interior orientation (IO), a set of parameters describing the internal geometry of the camera. We are using a seven parameter model for interior orientation: two scale parameters c_x, c_y, two translation parameters x_0, y_0, and three lens distortion parameters k_1, k_2, and k_3, and the following equations relating photo to image coordinates:

$$x' = c_x(x - x_0)k$$

$$y = c_y(y - y_0)k$$

$$\text{where } k = 1 + k_1 r + k_2 r^2 + k_3 r^3,$$

$$\text{and } r = (x - x_0)^2 + (y - y_0)^2$$

An initial estimate of the IO parameters can be obtained by making the following assumptions: (1) there is no lens distortion, yielding $k_1 = k_2 = k_3 = 0$, (2) the camera center of projection is imaged at the center of the image file, yielding $(x_0, y_0) = ((n_c - 1)/2, (n_r - 1)/2)$, and (3) an estimate f of the camera focal length and the physical size α of a pixel are known, yielding $c_x = c_y = \alpha/f$. This initial estimate of the IO is used as a starting point in the block adjustment described in section 6, in which more precise values of the IO parameters are obtained. Although the software allows separate values of c_x and c_y to be input; the current block adjustment routine assumes that they are equal, so for practical purposes these two should be considered one parameter.

We now turn to the relationship between the camera coordinate systems of the various photos of the block. In the Range Finder data structures, the photos of a block are ordered; and the first photo in this ordering, which must be one of those taken at the primary camera station, is called the reference photo. The geometry of the other photos is expressed in terms of the camera coordinate system of this reference photo, also called the reference coordinate system; coordinates of each camera station are expressed in that coordinate system, and the orientation of each block photo is expressed as a rotation from its camera coordinate system to the reference coordinate system. Here is an example of how the information discussed so far appears in a Range Finder block file:

```
[stations]
    0.00000000      0.00000000      0.00000000
   21.71814677      2.22596730    -21.26963966

[photos]
cp3-L02    0     0.00000000     0.00000000    0.00000000
cp3-L03    0    -0.07401899    -9.39304785    0.23329361
cp3-L04    0    -0.10007923   -18.76682296    0.47444716
cp3-R02    1    -0.31906956     1.38381084    0.49542606
cp3-R03    1    -0.16483240    -8.28314834    0.93992802
```

```
cp3-R04    1      -0.00783966    -17.60389740      1.30013806

[io]
cp3-L02  1517.2553  1047.9045  0.0000686  0.0000686 -0.0000584 -0.00000000473  0.0
cp3-L03  1517.2553  1047.9045  0.0000686  0.0000686 -0.0000584 -0.00000000473  0.0
cp3-L04  1517.2553  1047.9045  0.0000686  0.0000686 -0.0000584 -0.00000000473  0.0
cp3-R02  1528.8749  1016.9314  0.0000714  0.0000714 -0.0000402 -0.00000000816  0.0
cp3-R03  1528.8749  1016.9314  0.0000714  0.0000714 -0.0000402 -0.00000000816  0.0
cp3-R04  1528.8749  1016.9314  0.0000714  0.0000714 -0.0000402 -0.00000000816  0.0

[models]
cp3-L02  cp3-R02
cp3-L03  cp3-R03
cp3-L04  cp3-R04
```

The two camera stations are listed first, each given by its x, y, z coordinates in the reference coordinate system. Then the six photos of the block are listed; on each line are listed the photo's name, then the index (0 or 1) of the camera station for that photo, then the Euler angles e_x, e_y, e_z (in degrees) of the rotation from that photo to the reference photo. Details on this rotation are given in Section 4. Then the IO parameters for each photo are listed beside the photo's name, in the order x_0, y_0, c_x, c_y, k_1, k_2, and k_3. When all photos have the same IO parameters, a single line in the block file is used, with the photo name field replaced by an asterisk (*). Finally the file lists the photo pairs which are designated as models.

3. Range Finder Workflow

A terrain model created for use in the training software consists of a stitched panoramic terrain image, a corresponding IR terrain image, a ground model (triangulated set of ground points), and a range image containing one range value (distance to first visible object) for each pixel in the stitched terrain image. A separate program, Overlay, is used for most of the work involved in creating the IR image. The other terrain files are created using Range Finder. Here are the main operational steps which the user needs to perform to produce these outputs:

1) <u>Block setup</u>. In this step the photos and models of the block are identified, and initial estimates of the IO parameters and camera station coordinates are entered. These values can be entered within Range Finder, and then an initial block file containing them saved, or the values can be entered directly into a new block file using a text editor.

2) <u>Block Adjustment</u>. In this step points are measured on multiple photos of the block, and ranges specified for a few points, and the software then computes values of the parameters described earlier (IO, camera stations, rotations). Here "measuring" a point refers to specifying its pixel coordinates on a photo, which may occur by a mouse click on a displayed image, or by one of several automated commands in the software. The "point" is a physical feature visible in multiple photos. When the point is measured on two photos taken from the same camera station, we may refer to it as a <u>stitch point</u>. When measured on the two photos of a model, we may refer to it as a <u>model point</u>,. In order to do a good job determining the geometry of the block, the software needs

several (at least 3 or 4) stitch points between each consecutive pair of photos at each camera station, and several (at least 6 or 8) model points for each model of the block. In addition, for each consecutive pair of models, there should be at least one or two points common to both models; i.e. measured on all four photos comprising the two models. The user should also enter a few range points (ideally at least one in each model); these are model points for which the actual range (distance from camera station to physical feature) is known and entered by the user. From the stitch points the software is able to compute initial estimates of the rotations. The user can then call on the software to perform a block adjustment, which "adjusts" the values of the all the parameters from their initial estimates to values which more precisely fit the geometric constraints given by the range values and point measurements.

3) Stitching. In this step the IO parameters and rotations between photos computed by the block adjustment are used to combine the photos taken at the primary camera station into a single panoramic image (called the "composite" in the software). The software allows the user to adjust the brightness of the portions of the composite originating from each original photo, to reduce the visible seam where these portions meet. In the setup for this step, the user chooses one of the original photos entering the stitch as the stitch "center photo," whose camera coordinate axes become those of the composite image as well as the coordinate system for the ground model.

4) Ground Model Construction. In this step the user creates a list of 3D points which model the terrain surface across the field-of-view of the composite image. A point may be created and added to this list by measuring it on both photos of a model and transferring the model coordinates to the composite, or by measuring it directly on the composite and specifying its range. The user may also add points whose position is derived from surrounding ground points, and edit ground point coordinates interactively.

5) Range Image Construction. The range file is an image file (stored in PNG format) having the same x and y dimensions as the composite image, but containing a range value for each pixel representing the distance in the scene to the first occluding object at that location. The range r stored for a given pixel is a 16-bit unsigned integer u using the following scheme: If $r < 2000$ (meters), then $u = 10r$ (i.e. u is the range in tenths of a meter). If $2000 \leq r < 47535$, then $u = r + 18000$. All ranges greater than 47535 meters are represented by $u = 65535$. This scheme allows u to represent large ranges (up to 47535 meters at one-meter resolution) while maintaining good resolution at the smaller ranges (one-tenth meter resolution below 2000 meters). Construction of the range image occurs in two steps. First the ground model constructed in step 4 is scan-converted into an initial range image. This image is then edited to add above ground objects and details not present in the ground model. These additional occluding objects are drawn into the range image using tools similar to those in "paint" programs.

More explanation on the procedures for point measurement in both steps 2 and 4, and on the underlying mathematics of the ground model, is given in the earlier report [1]. Information about the automated point matching added since that report is given in a subsequent report [2].

4. Rotations, Relative Orientation, and Stereo Intersection

The rotation parameters (Euler angles) e_x, e_y, e_z for a given block photo are interpreted as follows. Let (x_s, y_s, z_s) be the camera station for the given photo (given in reference coordinates). Let (x, y, z) be coordinates of some object point in the camera coordinate system of the given photo. The coordinates of that point in the reference coordinate system are then computed as

$$[\, x' \;\; y' \;\; z' \,] = [\, x_s \;\; y_s \;\; z_s \,] + [\, x \;\; y \;\; z \,]R, \text{ where}$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(e_x) & \sin(e_x) \\ 0 & -\sin(e_x) & \cos(e_x) \end{bmatrix} \begin{bmatrix} \cos(e_y) & 0 & \sin(e_y) \\ 0 & 1 & 0 \\ -\sin(e_y) & 0 & \cos(e_y) \end{bmatrix} \begin{bmatrix} \cos(e_z) & \sin(e_z) & 0 \\ -\sin(e_z) & \cos(e_z) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

a 3×3 rotation matrix.

The term <u>relative orientation</u> (RO) [3] refers to the rotation and translation relating the two photos of a model. The RO may be represented as a rotation matrix R and a vector **b**. We refer to the two photos as photos 0 and 1. If we regard the model as a two-photo block, with photo 0 as the reference photo, then R is simply the rotation matrix of photo 1 and **b** the camera station of photo 1, as above. If photo 0 is in fact the reference photo of the full block in which we are working, then again R is the rotation matrix of photo 1 and **b** the camera station of photo 1. In the general case, let $\mathbf{p}_0$ and $\mathbf{p}_1$ be the camera stations of the two photos, and R_0 and R_1 their rotation matrices, within the full block. Then the RO for the model is given by $R = R_0^{-1} R_1$, and $\mathbf{b} = (\mathbf{p}_1 - \mathbf{p}_0)R_0^{-1}$.

Given the IO and RO parameters for the two photos of a model, we may use stereo intersection to reconstruct an object point from measurements of it on the two photos. Using the IO parameters we can convert the measurements to photo coordinates (p_x, p_y) on photo 0 and (q_x, q_y) on photo 1. Within the two camera coordinate systems, the lines joining the object point and camera station pass through points $(p_x, p_y, 1)$, $(q_x, q_y, 1)$. Converting these to vectors, both in the coordinate system of photo 0 yields

$$\mathbf{r}_0 = [\, p_x \;\; p_y \;\; 1 \,]$$
$$\mathbf{r}_1 = [\, q_x \;\; q_y \;\; 1 \,]\, R$$

These vectors represent two rays, which intersect in the object point. Because measurement errors result in the rays failing to exactly intersect, we actually take the reconstructed object point to be the midpoint of the line segment joining points of closest approach on the two rays. That segment is perpendicular to the two rays and hence its displacement is a multiple of $\mathbf{r}_0 \times \mathbf{r}_1$, say $\gamma(\mathbf{r}_0 \times \mathbf{r}_1)$. Letting $\alpha\mathbf{r}_0$ and $\beta\mathbf{r}_1$ be vectors from the two camera stations to the points of closest approach, it then follows that

$$\alpha\mathbf{r}_0 + \gamma(\mathbf{r}_0 \times \mathbf{r}_1) = \mathbf{b} + \beta\mathbf{r}_1 .$$

To compute α and γ, take dot products of the equation above with $\mathbf{r}_0 \times \mathbf{r}_1$ and with $\mathbf{r}_1 \times (\mathbf{r}_0 \times \mathbf{r}_1)$, respectively, to obtain

$$\gamma\|\mathbf{r}_0 \times \mathbf{r}_1\|^2 = \mathbf{b} \cdot (\mathbf{r}_0 \times \mathbf{r}_1)$$
$$\alpha\|\mathbf{r}_0 \times \mathbf{r}_1\|^2 = (\mathbf{b} \times \mathbf{r}_1) \cdot (\mathbf{r}_0 \times \mathbf{r}_1)$$

The object point may then be computed as $\alpha \mathbf{r}_0 + (\gamma/2)(\mathbf{r}_0 \times \mathbf{r}_1)$. Coordinates calculated in this way are displayed in Range Finder when the user selects the "Model Coordinates" option on the Point List dialog.

5. Displaying and Editing Photo Parameters

Range Finder uses the two dialogs shown below for display and editing of photo parameters.

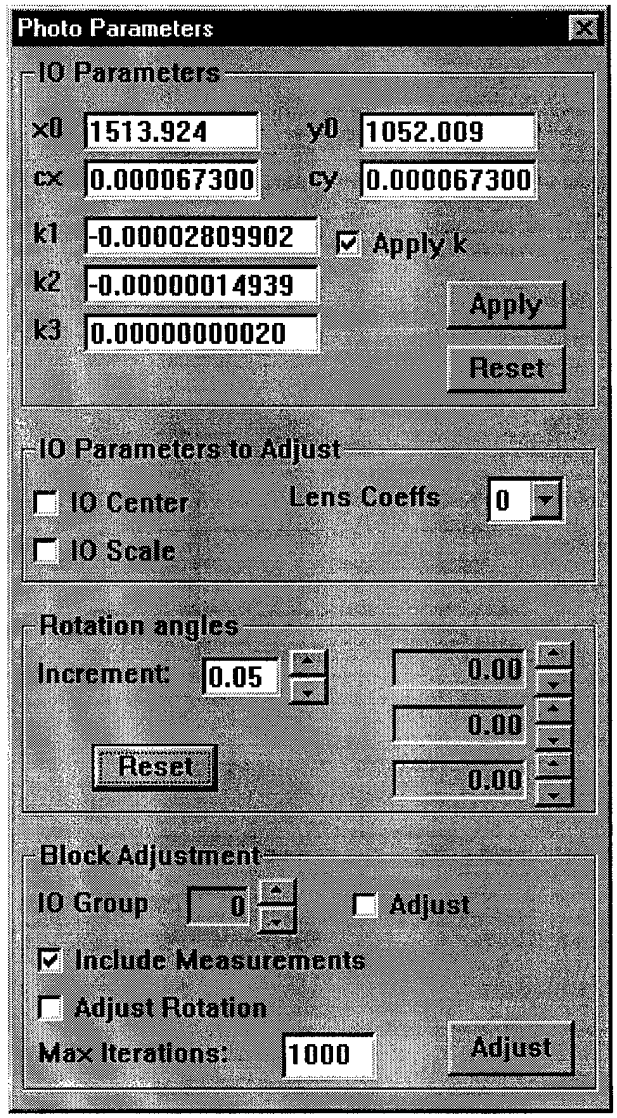
Block Photos

#	name	# meas	c.s.	rx	ry	rz	adj	io
0	cp4-L03	70	0	0.000	0.000	0.000	I	0
1	cp4-L04	76	0	0.065	-8.599	0.482	I	0
2	cp4-L05	90	0	0.227	-17.194	1.041	I	0
3	cp4-L06	101	0	0.454	-25.737	1.620	I	0
4	cp4-L07	71	0	0.804	-34.305	2.271	I	0
5	cp4-R03	60	1	0.152	1.476	1.212	I	0
6	cp4-R04	58	1	0.157	-8.125	1.248	I	0
7	cp4-R05	55	1	0.106	-16.624	1.273	I	0
8	cp4-R06	66	1	0.153	-25.120	1.479	I	0
9	cp4-R07	62	1	0.261	-34.177	1.693	I	0

The first of these dialogs, with the caption "Block Photos", is a list of the photos in the block in the same order they appear in the block file discussed in section 2. The columns of the list are the photo number and name, the number of point measurements recorded on the photo, the camera station (index into the camera station list), the three rotation (Euler) angles, an adjustment code, and an IO group number. All except the last two columns' values have been discussed earlier. The adjustment code is either a blank, the letter I, or the letter A, according to how the photo is treated in the Estimate Rotations and Adjust Block commands discussed in subsequent sections. An A indicates that the photo's rotation is to be adjusted by those commands, and also that its measurements are included in the adjustment; an I indicates that the photo's rotation will not be adjusted, but that its measurements will be included in adjusting other photos. A blank indicates the photo is left out of the command processing altogether. The user may change the adjustment codes of multiple photos at once, by selecting (highlighting) the photos on the list, and then checking/unchecking the corresponding boxes on the Photo Parameters dialog box.

The IO group numbers indicate which photos are grouped together and deemed to have the same IO parameters; i.e. if two photos have the same IO group number, then changing an IO parameter for one changes it for the other as well. The IO group number of multiple photos may be changed together by selecting (highlighting) the photos on the photo list, and then clicking the up/down arrow buttons beside the IO group number on the Photo Parameters dialog. The values of the IO parameters for the group may be edited at the top of the Photo Parameters dialog, by typing in new values and hitting the "Apply" button. The "Reset" button returns the values to initial estimates, setting (x_0, y_0) to the image center, and the lens distortion coefficients to zero.

The rotation angles of individual photos may also be manually controlled on the Photo Parameters dialog; in the case the edits apply to only one photo at a time.



5. Initial Rotation Estimates

The block adjustment in section 6 depends upon initial estimates of all the parameters (IO, camera stations, rotations). In setting up a new block the user enters initial estimates for the IO parameters and camera stations, as discussed earlier. To obtain initial estimates for the rotations, Range Finder provides two commands, "Estimate Rotations" and "Compute RO". The first of these computes rotations of successive photos at a single camera station; the second computes the rotation (and updates the camera station estimate) of the model photo that is not at the primary camera station. These commands may be used in conjunction as follows:

1) Measure stitch points on successive photos at the primary camera station. Mark those photos for adjustment by checking the "Adjust Rotations" option on the Edit Photo dialog box, and then select "Estimate Rotations" from the Adjust menu. The command performs pairwise rotation adjustments sequentially; for each successive pair, it leaves the rotation of the first photo fixed and updates the rotation of the second photo, based on point measurements on the two photos. Thus if applied to photos 0, 1, 2, and 3, it will in turn update the rotations first for photo 1 (using photos 0 and 1), then photo 2 (using photos 1 and 2), and then photo3 (using photos 2 and 3).

2) For each secondary camera station, begin by activating the model consisting of a photo at the primary station and the first photo at the secondary station. (Those two photos should be displayed in the main program window.) Now select the "Compute RO" command; this will update the coordinates of the secondary camera station, and compute a rotation for the first photo at that station. Now mark all the photos at the secondary station, and no others, with the "Adjust Rotations" option, and apply the "Estimate Rotations" command to estimate rotations for the remaining photos at that station.

6. Block Adjustment

Range Finder computes the final IO parameters, camera station coordinates, and rotation angles for all photos in the block using a numerical optimization procedure we call <u>block adjustment</u>. The adjustment attempts to find those values of the various parameters which minimize a sum of squared residuals. The user can control which parameters are included each time the adjustment is performed:

1) Each IO group may be included or excluded via a checkbox on the Photo Parameters dialog. In addition, individual IO parameters may be included or excluded: dialog checkbox options control whether (a) the IO center (x_0, y_0) is included, (b) the IO scale (c_x, c_y, which are assumed to be equal) is included, and (3) the number of lens distortion coefficients to include, 0 for no lens distortion, 1 to include just the first coefficient k_1, 2 to include k_1 and k_2, or 3 to include all three coefficients. The lens distortion coefficients not included are set equal to 0. The IO center and/or scale, when not included in the adjustment, are left at their current values. The choice of which IO parameters to include applies to all IO groups included in the adjustment.

2) For each photo whose adjustment flag is set (appears as an A on the photo list), three rotation angles are included in the adjustment. Also any such photo's camera station is adjusted, if it is not the primary camera station (which is fixed as the origin of the reference coordinate system).

The residuals whose sum of squares is to be minimized arise from measurements of an object point on multiple photos. Suppose a given object point has been measured on at least two photos of the block. For each pair of measurements, we convert the pixel coordinates of the measurements to photo coordinates (p_x, p_y) and (q_x, q_y) using the current values of the IO parameters for each photo. As in section 4 we form the rays

$$\mathbf{r}_0 = [\, p_x \;\; p_y \;\; 1 \,]\, R_0$$

$$\mathbf{r}_1 = [\, q_x \;\; q_y \;\; 1 \,]\, R_1$$

where we have rotated both vectors into the reference coordinate system using the rotation matrices R_0 and R_1 of the two photos. If the two photos have the same camera station, then $\mathbf{r}_0$ and $\mathbf{r}_1$ should be collinear (both on the line from the common camera station to the object point). Thus $\mathbf{r}_0/\|\mathbf{r}_0\|$ and $\mathbf{r}_1/\|\mathbf{r}_1\|$ should be equal, and we take as residual $\|\mathbf{d}\|$, where $\mathbf{d} = \mathbf{r}_0/\|\mathbf{r}_0\| - \mathbf{r}_1/\|\mathbf{r}_1\|$. If the two photos have different camera stations, then we perform the stereo intersection computations as in section 4, using $\mathbf{r}_0$ and $\mathbf{r}_1$ above and $\mathbf{b} = \mathbf{p}_1 - \mathbf{p}_0$, the vector joining the two camera stations, which are already in reference coordinates:

$$\gamma = (\mathbf{b} \cdot (\mathbf{r}_0 \times \mathbf{r}_1))/\|\mathbf{r}_0 \times \mathbf{r}_1\|^2$$

$$\alpha = ((\mathbf{b} \times \mathbf{r}_1) \cdot (\mathbf{r}_0 \times \mathbf{r}_1))/\|\mathbf{r}_0 \times \mathbf{r}_1\|^2$$

The object point and ray miss vector are given by

$$\mathbf{p}_{obj} = \alpha\mathbf{r}_0 + (\gamma/2)(\mathbf{r}_0 \times \mathbf{r}_1)$$

$$\mathbf{r}_{miss} = \gamma(\mathbf{r}_0 \times \mathbf{r}_1)$$

and for this case we take the residual to be the length of the miss vector divided by the distance to the object point, $\|\mathbf{r}_{miss}\|/\|\mathbf{p}_{obj}\|$. In addition to these residual terms, we include residuals representing relative differences in computed and given ranges of object points. When an object point has more than two measurements, including measurements involving multiple camera stations, we compute the minimum, maximum, and mean of the value $\|\mathbf{p}_{obj}\|$ as computed above, over all pairs of measurements from distinct camera stations. We then include a residual term of the form $(d_{max} - d_{min})/d_{mean}$. In addition, for any range point (object point with user-specified range) with at least one measurement pair from distinct camera stations, we include a residual term of the form $(d_{mean} - d_{given})/d_{given}$.

Because the residual terms are somewhat complicated, we are using a minimization procedure which does not require derivatives, basically following the direction set method found in [4]. The process requires a bit of trial and error on the users part, in varying the sequence in which photos and parameters are included in the adjustment. Generally we have found it best to adjust the values of the rotation angles from their initial estimates, before including the IO parameters, and then to include the IO scale before either the IO center or lens distortion coefficients.

References

[1] Stephen J. Dow, *Basic Skills Trainer Software Development*, Final Report, Contract DAAH01-97-D-R005 D.O. 3, June, 1998.

[2] Stephen J. Dow, *EPBST Workstation Improvement: Automated Point Matching*, Final Report, Contract DAAH01-97-D-R005 D.O. 22, November 2001.

[3] Berthold K.P. Horn, *Relative Orientation*, International Journal of Computer Vision, 4, 59-78. 1990.

[4] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes in C*, 2nd Ed., Cambridge Univ. Press, 1992.