A Method to Solve Interior and Exterior Camera Calibration Parameters for Image Resection

Ravi Samtaney[1]
NAS-99-003
April 1999

Abstract

An iterative method is presented to solve the internal and external camera calibration parameters, given model target points and their images from one or more camera locations. The direct linear transform formulation was used to obtain a guess for the iterative method, and herein lies one of the strengths of the present method. In all test cases, the method converged to the correct solution. In general, an over-determined system of nonlinear equations is solved in the least-squares sense. The iterative method presented is based on Newton-Raphson for solving systems of nonlinear algebraic equations. The Jacobian is analytically derived and the pseudo-inverse of the Jacobian is obtained by singular value decomposition.

[1]MRJ Technology Solutions, Inc., NASA Ames Research Center, Moffett Field, CA 94035-1000. email: samtaney@nas.nasa.gov

1 Introduction

Certain experimental flow visualization techniques such as the Pressure Sensitive Paint (PSP), Temperature Sensitive Paint and others involve taking images of aircraft models in a wind tunnel. Typically, these images are taken with one or more CCD cameras from different positions and orientations. These images are then subjected to image registration algorithms and mapped onto the model geometry. Details about registration of PSP images are discussed by Bell [2]. The mapping of the images to the model geometry, essentially a mapping relationship between model and image coordinate systems, is expressed by the equation of photogrammetry (See Marzan and Karara [6]). The relationship between the model and image coordinates requires the determination of the camera calibration parameters. The camera calibration paramters are: (a) external or extrinsic parameters which depend upon the location and orientation of the camera, and (b) internal or intrinsic parameters which are the camera focal length, other lens parameters, distortion parameters, etc. The camera parameters which are relevant to this report are discussed in more detail in Section 2. Typically, certain control points on the model and their projected images are known a priori. Given the coordinates of the control points, the equations of projective photogrammetry, which are nonlinear equations, are solved to yield the camera orientation. The most common method to solve the equations with an iterative solution technique is termed *bundle adjustment* ([5, 4]). Another technique to obtain the mapping between the image and model space is the Direct Linear Transform (DLT) method [1].

In this report, a method to solve for the camera calibration parameters is described. The solution method uses the DLT to obtain a guess for the camera parameters, and then iteratively solves the nonlinear equations using a Newton method. The Jacobian for the Newton iterative technique is obtained analytically, and the inverse (or rather pseudo-inverse) Jacobian is obtained by singular value decomposition.

2 Solution Method

2.1 Equations of Photogrammetry

Let the model and image coordinate systems be denoted by (X, Y, Z) and (x, y, z), respectively. The origin of the image coordinate system is located at (X_C, Y_C, Z_C) in model coordinate space. The orientation of the image coordinate system with respect to the model system is given by a 3×3 rotation matrix R. The nine terms $r_{ij}, i = 1, 2, 3 \quad j = 1, 2, 3$ of R are not independent and can be reduced to three parameters by taking the Euler angle approach, i.e., R is a combination of three separate rotations about the three principal axes. Let the rotation about the x, y and z axes be denoted by ω, ϕ, and κ, respectively. Then

$$R = R_z(\kappa) R_y(\phi) R_x(\omega) \tag{1}$$

where each individual rotation matrix is given by

$$R_x(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\omega & -\sin\omega \\ 0 & \sin\omega & \cos\omega \end{bmatrix}, \; R_y(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix},$$

$$R_z(\kappa) = \begin{bmatrix} \cos\kappa & -\sin\kappa & 0 \\ \sin\kappa & \cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

One must be careful and note that the rotations do not commute. After matrix multiplication, R is given by

$$R = \begin{bmatrix} \cos\phi\cos\kappa & -\cos\omega\sin\kappa + \sin\omega\sin\phi\cos\kappa & \sin\omega\sin\kappa + \cos\omega\sin\phi\cos\kappa \\ \cos\phi\sin\kappa & \cos\omega\cos\kappa + \sin\omega\sin\phi\sin\kappa & -\sin\omega\cos\kappa + \cos\omega\sin\phi\sin\kappa \\ -\sin\phi & \sin\omega\cos\phi & \cos\omega\cos\phi \end{bmatrix}.$$

Note that the rows (and columns) of the above matrix form orthogonal bases vectors, a property which will be exploited later.

In actual practice, several target locations are identified on the model and their coordinates are measured. The model is imaged by a CCD camera. For an ideal camera, the projection of a model point (X, Y, Z) is (x, y) in image coordinates. However, due to image distortion, the image of the model point is measured at (x', y').

The relationship between model and image coordinates is expressed by the projective equation of photogrammetry[6]

$$\begin{aligned} x - x_p &= x' - x_p + \Delta x \\ &= -f\frac{r_{11}(X - X_C) + r_{12}(Y - Y_C) + r_{13}(Z - Z_C)}{r_{31}(X - X_C) + r_{32}(Y - Y_C) + r_{33}(Z - Z_C)}, \\ y - y_p &= y' - y_p + \Delta y \\ &= -f\frac{r_{21}(X - X_C) + r_{22}(Y - Y_C) + r_{23}(Z - Z_C)}{r_{31}(X - X_C) + r_{32}(Y - Y_C) + r_{33}(Z - Z_C)}. \end{aligned} \tag{2}$$

In equation (2) $\Delta x, \Delta y$ are the terms which model the effects of symmetrical and asymmetrical lens distortion; x_p, y_p are the image coordinates of the principal point; and r_{ij} is the $i\ th$ row and $j\ th$ column element of R; and f is the focal length of the camera. A commonly used set of functions for Δx and Δy is

$$\begin{aligned} \Delta x &= \bar{x}(k_1 r^2 + k_2 r^4 + k_3 r^6) + p_1(r^2 + 2\bar{x}^2) + 2p_2\bar{x}\bar{y}, \\ \Delta y &= \bar{y}(k_1 r^2 + k_2 r^4 + k_3 r^6) + 2p_1\bar{x}\bar{y} + p_2(r^2 + 2\bar{y}^2) + a_1\bar{x} + a_2\bar{y}, \end{aligned} \tag{3}$$

where $\bar{x} \equiv x' - x_p$, $\bar{y} \equiv y' - y_p$, $r^2 \equiv \bar{x}^2 + \bar{y}^2$; and k_i, p_i and a_i represent radial, lens decentering, and affinity distortion terms, respectively. The six-tuple $(X_C, Y_C, Z_C, \omega, \phi, \kappa)$ which gives the position and orientation of the camera contains the *external calibration parameters*; while the ten-tuple $(k_1, k_2, k_3, p_1, p_2, a_1, a_2, x_p, y_p, f)$ representing the radial, lens decentering, affinity distortion terms, the principal point, and the focal length contains the *internal calibration parameters.*

Let the number of model target points be m. A given camera produces images of the model from n different locations. For each camera location, the image coordinates of a subset of the m model points may be measured. Let the number of measured image points for the $i\ th$ camera location be $m_i \leq m$. The total number of unknowns are $6n + 10$ where $6n$ are the six external calibration parameters for n camera locations and 10 internal calibration parameters. The total number of equations available is $2\sum_{i=1}^{n} m_i \leq 2nm$. For all the test cases discussed later in this report, we encounter an over-determined system, i.e., $2\sum_{i=1}^{n} m_i > 6n + 10$.

2.2 Solution Method

The over-determined system of equations governing the internal and external calibration parameters is solved in a least-squares sense as follows. We first

recast the equations in a convenient form. Define a vector of unknowns as

$$\xi \equiv \xi(\mathcal{X}_{C,1}, \Phi_1, \mathcal{X}_{C,2}, \Phi_2, \cdots \mathcal{X}_{C,n}, \Psi)^T \tag{4}$$

where $\mathcal{X}_{C,i} \equiv (X_{C,i}, Y_{C,i}, Z_{C,i})$ and $\Phi_i \equiv (\omega_i, \phi_i, \kappa_i)$ are the camera external calibration parameters for the $i\ th$ camera location; and
$\Psi \equiv (k_1, k_2, k_3, p_1, p_2, a_1, a_2, x_p, y_p, f)$ is the vector of the camera internal calibration parameters. Then rewrite the equations as

$$\mathcal{F}(\xi) = \left[\begin{array}{l} F_l(\xi) = (x'_j + \Delta x_j - x_p)(\mathcal{X}_j - \mathcal{X}_{C,i}) \cdot \hat{r}_{3,i} + f(\mathcal{X}_j - \mathcal{X}_{C,i}) \cdot \hat{r}_{1,i} = 0 \\ G_l(\xi) = (y'_j + \Delta y_j - x_p)(\mathcal{X}_j - \mathcal{X}_{C,i}) \cdot \hat{r}_{3,i} + f(\mathcal{X}_j - \mathcal{X}_{C,i}) \cdot \hat{r}_{1,i} = 0 \end{array} \right]. \tag{5}$$

In equation (5), $l = m_{i-1} + j$, $(m_0 = 0)$; $j = 1, 2, \cdots m_i$; and $i = 1, 2, \cdots n$, with the index j varying faster than index i. Furthermore, $\mathcal{X}_j$ is the position vector of the $j\ th$ target point, i.e., $\mathcal{X}_j \equiv (X_j, Y_j, Z_j)$; and $\mathcal{X}_{C,i}$ is the position vector of the $i\ th$ camera, i.e., $\mathcal{X}_{C,i} \equiv (X_{C,i}, Y_{C,i}, Z_{C,i})$. The unit vectors $\hat{r}_{k,i}$ correspond to the $k\ th$ column vector of the rotation matrix R and are functions of the $i\ th$ camera Euler angles $(\omega_i, \phi_i, \kappa_i)$.

The above system of nonlinear equations is solved in an iterative fashion as described below. Assume the solution vector ξ is known at iteration 'k'. The left-hand side of equation (5) may be expressed as a truncated Taylor series at iteration $k+1$.

$$\mathcal{F}^{k+1}(\xi) = \mathcal{F}^k(\xi) + \left(\frac{\partial \mathcal{F}}{\partial \xi}\right)^k \Delta\xi^k, \tag{6}$$

where $\Delta\xi^k \equiv \xi^{k+1} - \xi^k$. In the above equation, $\mathcal{F}$ is a column vector with $2\sum_i^n m_i$ elements, the Jacobian $\partial\mathcal{F}/\partial\xi$ is a matrix with $2\sum_i^n m_i$ rows and $(6n+10)$ columns, and $\Delta\xi$ is a column vector with $(6n+10)$ elements. Following the Newton-Raphson iterative technique, we set $\mathcal{F}^{k+1} = 0$ and solve for ξ^{k+1} as

$$\xi^{k+1} = \xi^k - \left[\left(\frac{\partial \mathcal{F}}{\partial \xi}\right)^k\right]^{-1} \mathcal{F}^k(\xi), \tag{7}$$

where the term $[]^{-1}$ is the inverse (pseudo-inverse) of the Jacobian matrix if $2\sum_i^n m_i = 6n+10$, $(2\sum_i^n m_i > 6n+10)$. The elements of the Jacobian matrix are analytically determined (see Appendix A for details). At each

iteration, we solve a system of linear equations. For over-determined systems ($2\sum_i^n m_i > 6n+10$) the linear system is solved in the least-squares sense. The above iterative procedure is terminated upon convergence. It is well known that the standard Newton-Raphson method converges if the initial guess is sufficiently close to the solution. The initial guess for the above procedure is discussed in detail in section 2.3.

2.2.1 Pseudo-inverse and Singular Value Decomposition

For an over-determined system ($2\sum_i^n m_i > 6n + 10$), the Jacobian matrix is not square. A preferred method for solving linear least-squares problems is singular value decomposition (SVD)[9]. In general, any $M \times N$ matrix A may be written as the product of an $M \times N$ column-orthogonal matrix U, an $N \times N$ diagonal matrix W, and the transpose of an $N \times N$ orthogonal matrix V. The system of equations $A.x = b$ is solved in the least-squares sense by finding x which minimizes $\chi^2 = ||A \cdot x - b||$. The solution x is given by

$$x = V \cdot [diag(1/w_j)] \cdot U^T \cdot b, \tag{8}$$

where w_j is the jth diagonal element of the diagonal matrix W and are called the singular values of A. The product of the three matrices,
$V \cdot [diag(1/w_j)] \cdot U^T$, in the equation above is the pseudo-inverse of A. In our case, the pseudo-inverse of the Jacobian is calculated using SVD, and the solution vector at ξ^{k+1} is calculated in the least-squares sense as outlined above. An added benefit of using singular value decomposition is that the above procedure works even when the Jacobian is singular or ill-conditioned. We compute the condition number (defined as the ratio of the largest to the smallest singular value) of the Jacobian matrix. If the inverse of the condition number is smaller than 10^{-p}, all singular values which are smaller than 10^{-p} times the maximum singular value are set to zero. This prevents round-off error from contaminating the solution. In our case, we choose $p = 12$ because our computations are done using double precision arithmetic.

2.2.2 Convergence Criteria

The convergence criteria to terminate the iterative solution is

$$|\mathcal{F}^{k+1} - \mathcal{F}^k| < \epsilon, \tag{9}$$

where ϵ is a "small" number determined by the user. Another alternative is to examine the change in the L_∞ norm of the solution vector $\Delta\xi^k$. For both criteria, convergence is achieved in $10-20$ iterations for $\epsilon = 10^{-8}$. In no test case did we need to iterate more than 20 times. Convergence was not quadratic, as one would expect for a standard Newton-Raphson technique. Perhaps this is not surprising because we are solving the equations in a least-squares sense, and in some cases the Jacobian matrix may have large condition numbers.

2.2.3 Gimbal Lock

We now focus attention on the case when $\phi = \pi/2$. The rotation matrix R for $\phi = \pi/2$ becomes

$$R = \begin{bmatrix} 0 & \sin(\omega-\kappa) & \cos(\omega-\kappa) \\ 0 & \cos(\omega-\kappa) & -\sin(\omega-\kappa) \\ -1 & 0 & 0 \end{bmatrix}. \tag{10}$$

This shows that the solution only depends upon the difference of ω and κ, i.e., the number of independent unknowns reduces by one. This is called "Gimbal Lock". Further discussion of this issue is given in Section 2.4.

2.3 Initial Guess

For a good optical system that does not suffer from optical distortion, the error terms in equation (2) may be neglected. Some algebraic manipulation leads to the following form, called the direct linear transform (DLT) formulation [1],

$$x = \frac{L_1X + L_2Y + L_3Z + L_4}{L_9X + L_{10}Y + L_{11}Z}, \quad y = \frac{L_5X + L_6Y + L_7Z + L_8}{L_9X + L_{10}Y + L_{11}Z}, \tag{11}$$

where $L_1 \cdots L_{11}$ are called the DLT coefficients. Given the model target points and their images, one solves for the DLT coefficients using the solution procedure given in Reference [8]. The DLT coefficients for n camera locations are obtained.

2.3.1 Estimate of Camera Position

In equation (2) the numerator and denominator are zero when $X = X_C, Y = Y_C, Z = Z_C$. This implies that the camera position vector $\mathcal{X}_C$ may be deter-

mined by solving the following linear system of equations:

$$\begin{aligned} L_1X_C + L_2Y_C + L_3Z_C + L_4 &= 0, \\ L_5X_C + L_6Y_C + L_7Z_C + L_8 &= 0, \\ L_9X_C + L_{10}Y_C + L_{11}Z_C + 1 &= 0. \end{aligned} \tag{12}$$

2.3.2 Estimate of Camera Focal Length

Recall that one of the internal calibration parameters to be determined is the camera focal length f. For convenience we define some new variables as follows:

$$\begin{aligned} \xi_1 \equiv \frac{L_1}{L_9} = -f\frac{r_{11}}{r_{31}}, \quad \xi_2 \equiv \frac{L_2}{L_9} = -f\frac{r_{12}}{r_{31}}, \quad \xi_3 \equiv \frac{L_3}{L_9} = -f\frac{r_{13}}{r_{31}}, \\ \eta_1 \equiv \frac{L_1}{L_{10}} = -f\frac{r_{11}}{r_{32}}, \quad \eta_2 \equiv \frac{L_2}{L_{10}} = -f\frac{r_{12}}{r_{32}}, \quad \eta_3 \equiv \frac{L_3}{L_{10}} = -f\frac{r_{13}}{r_{32}}, \\ \zeta_1 \equiv \frac{L_1}{L_{11}} = -f\frac{r_{11}}{r_{33}}, \quad \zeta_2 \equiv \frac{L_2}{L_{11}} = -f\frac{r_{12}}{r_{33}}, \quad \zeta_3 \equiv \frac{L_3}{L_{11}} = -f\frac{r_{13}}{r_{33}}. \end{aligned} \tag{13}$$

Now we use the fact that the row and column vectors of R are orthogonal unit vectors. It can then be trivially shown that

$$\frac{1}{f^2} = \frac{1}{\xi_i^2} + \frac{1}{\eta_i^2} + \frac{1}{\zeta_i^2}, \tag{14}$$

where $\xi_i^2 = \xi_1^2 + \xi_2^2 + \xi_3^2$ and so on. The focal length is then calculated by taking the positive square root of f^2. In the estimation of the focal length, an alternative is to use L_4, L_5, L_6 and r_{21}, r_{22}, r_{33} instead of L_1, L_2, L_3, r_{11}, r_{12}, r_{13}, respectively, in equation (13). Finally, because we have n camera locations, we calculate n sets of DLT coefficients and n focal lengths. We choose the starting point of the focal length of the camera as the average of the n estimated focal lengths.

2.3.3 Estimate of Camera Orientation

We begin by using the last row of R, the calculated DLT coefficients, and the estimated focal length to estimate the orientation Euler angles (ω, ϕ, κ) for each camera location. We first estimate ϕ by the following relation

$$\sin^2\phi = r_{31}^2 = \frac{f^2}{\xi_i^2} \tag{15}$$

We take the positive square root and take the principal value for ϕ. The other two angles are estimated by using the following equations:

$$\sin^2\omega = \frac{f^2}{\eta_i^2\cos^2\phi}, \quad \cos\kappa = \frac{L_1\sin\phi}{fL_9\cos\phi}. \tag{16}$$

For the rare case of $\phi = \pi/2$, we simply set $\phi = 0.95\pi/2$. This is justified because we are simply interested in an initial guess for ϕ. This eliminates the possibility that the denominator in equation (16) is zero. Note that as calculated above $\phi \in [0, \pi/2]$, $\omega \in [0, \pi/2]$ and $\kappa \in [0, \pi]$. Using these calculated angles, one may recalculate the focal length using the following relations:

$$\begin{aligned} f &= \frac{L_1\gamma}{r_{11}}, \quad f = \frac{L_2\gamma}{r_{12}}, \quad f = \frac{L_3\gamma}{r_{13}}, \quad f = \frac{-L_4\gamma}{r_{11}X_C + r_{12}Y_C + r_{13}Z_C}, \\ f &= \frac{L_5\gamma}{r_{21}}, \quad f = \frac{L_6\gamma}{r_{22}}, \quad f = \frac{L_7\gamma}{r_{23}}, \quad f = \frac{-L_8\gamma}{r_{21}X_C + r_{22}Y_C + r_{23}Z_C}, \end{aligned} \tag{17}$$

where $\gamma \equiv r_{31}X_C + r_{32}Y_C + r_{33}Z_C$. The estimate of the angles are refined by using $-\theta, \pi \pm \theta, \theta - \pi$, $(\theta = \omega, \phi, \kappa)$, to calculate R which is then used in equation (17). That combination of angles which gives a positive focal length for all the above eight calculations in equation (17) is taken as the initial guess of the orientation.

2.4 Test Cases and Uniqueness of Solution

For nonzero Δx and Δy in equation (2), there are a multiplicity of solutions. It is unclear how many solutions are actually present. Contrived tests were carried out with known camera calibration parameters. When the above procedure to obtain an initial guess of the solution is adopted, the solution obtained was always the correct one. For some contrived tests, if the solution vector was initialized to arbitrary quantities, the converged solution was not the correct one. In some cases, the solution converged to a very small focal length with large distortion parameters. It is recommended that the image points be recomputed once the camera calibration parameters are determined. Then a calculation of the root-mean-square error between the computed and the given image point coordinates is performed. For model and target points provided by Bell [3] for the B-70 and Boeing airplane test data, the root-mean-square error was always less than one pixel.

If $\Delta x = \Delta y = 0$ and if the focal length is provided, then the above iterative procedure converges to a unique solution. Of course, the solution for the orientation angles is unique up to an additive constant. It is trivial to ensure that the angles always lie between -2π and 2π.

For cases of Gimbal lock, tests indicated that the solution converges to arbitrary values of ω and κ, while the correct difference between these two angles is always attained. Thus, nothing special is done for the case when $\phi = \pi/2$. The reader is simply cautioned that the difference $\omega - \kappa$ is important and not the individual values. Because we are eventually interested in mapping between the image and model spaces, the mapping obtained even in the case when we have gimbal lock is correct.

Finally, we note that the above solution procedure was coded into a C-program library which may be obtained by sending electronic mail to the author. The library was made flexible enough so that, in the event the code converges to an undesirable solution (which is discerned quickly by an examination of the root-mean-square error between measured and calculated image point locations), the user may specify an initial guess for the solution. We now present two specific sample calculations. The data used in these sample calculations is presented in tabular form, and the interested reader may obtain the numerical values by sending electronic mail to the author.

2.4.1 Example 1: Joukowski Wing Dataset

A dataset was artificially generated to mimic the situation encountered in pressure sensitive paint experiments. Basically, we created a wing using several two-dimensional cross sections, each of which was a Joukowski airfoil [7]. Then we sprinkled 46 target control points in a random fashion on this wing. A camera was placed at four different stations. The resulting four images, each of size 512×512 pixels, are shown in Figure 1. The target points, and their locations in the images are given in Table (1), and Tables (2)-(5), respectively. Note that the image coordinates, in pixels, are usually scaled by a certain factor in order to operate in the same units as the model. In this example, the scale factor was $1/512$. The above method was used to determine the camera calibration parameters and the results are given in Table (6). In this contrived example, in 7 iterations $|\mathcal{F}^{k+1} - \mathcal{F}^k| = 5.7 \times 10^{-8}$ and in 8 iterations $|\mathcal{F}^{k+1} - \mathcal{F}^k| = 1.3 \times 10^{-15}$. Furthermore, in this example in 8 iterations we also achieved $|\mathcal{F}^{k+1}| < 10^{-15}$. Another point to note in this example, is that, for station numbered '1', the orientation angle $\phi = 90^o$ and

therefore we have a gimbal lock. The images were generated using $\omega = 0^o$ and $\kappa = 50^o$. The solution obtained shows $\omega = 1.381^o$ and $\kappa = 51.381^o$ so that the difference between these angles is still 50^o as it should be for the gimbal lock case. The internal parameters are very small as they should be in this example.

2.4.2 Example 2: Boeing Airplane Dataset

Data in the form of target point locations and their images was provided by Bell [3] for a Boeing airplane. The target points are given in Table (7). Of course, the units are normalized so that they do not correspond to the real dimensions of the actual model in the experiment. Three images, each of size 512×1024 pixels, were provided. The coordinates of the target points for each of the three images are given in Tables (8)-(10). The image coordinates were scaled by a factor 9.4488×10^{-4}. The above method was used to determine the camera calibration parameters, and the results are given in Table (11). In this example, convergence is achieved in 10 iterations when we obtain $|\mathcal{F}^{k+1} - \mathcal{F}^k| = 3.66 \times 10^{-9}$. The L_∞ norm of $\mathcal{F}$ settled down to 0.051818 after 6 iterations. Also shown in Tables (8)-(10) are the calculated coordinates of the target points in the images using equation (2). The root-mean-square error between the calculated and the measured locations of the image points is less than 0.2 pixels.

3 Conclusion

In this report, we have presented an iterative method to solve the internal and external camera calibration parameters given model target points and their images from one or more camera locations. The direct linear transform formulation was used to obtain a guess for the iterative method, and herein lies one of the strengths of the present method. In all test cases, the method gave the correct solution without user intervention in selecting a good initial guess. Sufficient details are given in the report, including detailed analytical expressions for the Jacobian, to enable one to write a computer program. Alternatively, a C-library which uses the present method to determine the camera calibration parameters may be obtained by sending electronic mail to the author.

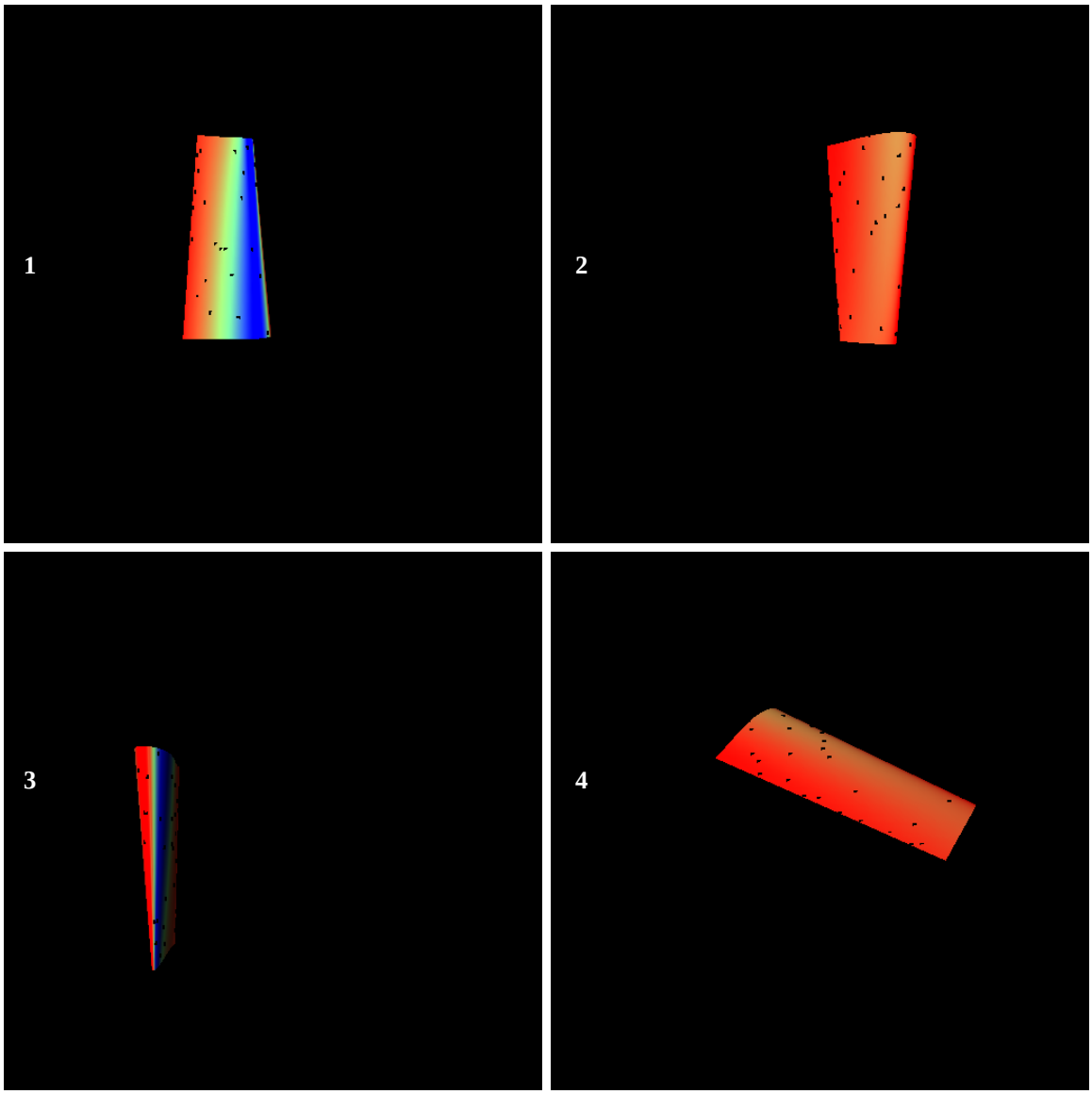


Figure 1: Images corresponding to the four different camera locations. The black dots on the images correspond to the target points. The color corresponds to the coefficient of pressure for the wing with red (blue) corresponding to high (low) values.

Acknowledgments

This work was supported by NASA Contract NAS2-14303. The author wishes to acknowledge Dr. James H. Bell for interesting discussions and for providing the data sets. The author is also thankful to Randy L. Kaemmerer for his meticulous proof-reading of the manuscript.

4 Appendix A: Derivation of the Jacobian

The Jacobian matrix in equation (6) is given by

$$\frac{\partial \mathcal{F}}{\partial \xi} = \begin{bmatrix} \frac{\partial F}{\partial \mathcal{X}_{C,1}} & \frac{\partial F}{\partial \Phi_1} & 0 & 0 & \cdots & 0 & 0 & \frac{\partial F}{\partial \Psi} \\ \frac{\partial G}{\partial \mathcal{X}_{C,1}} & \frac{\partial G}{\partial \Phi_1} & 0 & 0 & \cdots & 0 & 0 & \frac{\partial G}{\partial \Psi} \\ 0 & 0 & \frac{\partial F}{\partial \mathcal{X}_{C,2}} & \frac{\partial F}{\partial \Phi_2} & \cdots & 0 & 0 & \frac{\partial F}{\partial \Psi} \\ 0 & 0 & \frac{\partial G}{\partial \mathcal{X}_{C,2}} & \frac{\partial G}{\partial \Phi_2} & \cdots & 0 & 0 & \frac{\partial G}{\partial \Psi} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & 0 & \cdots & \frac{\partial F}{\partial \mathcal{X}_{C,n}} & \frac{\partial F}{\partial \Phi_n} & \frac{\partial F}{\partial \Psi} \\ 0 & 0 & 0 & 0 & \cdots & \frac{\partial G}{\partial \mathcal{X}_{C,n}} & \frac{\partial G}{\partial \Phi_n} & \frac{\partial G}{\partial \Psi} \end{bmatrix}, \qquad (18)$$

Each row of the Jacobian matrix consists of $6n+10$ entries, of which $6n-6$ entries are zero. The derivatives of F (and G) with respect to $\mathcal{X}_{C,i}$, Φ_i and Ψ are row vectors with 3, 3, and 10 elements, respectively. These derivatives are given below.

$$\begin{aligned} \frac{\partial F}{\partial \mathcal{X}_C} &= -(x - x_p)\hat{r}_3 - f\hat{r}_1, \\ \frac{\partial G}{\partial \mathcal{X}_C} &= -(y - y_p)\hat{r}_3 - f\hat{r}_2, \end{aligned} \qquad (19)$$

$$\begin{aligned} \frac{\partial F}{\partial \Phi} &= (x - x_p)(\mathcal{X} - \mathcal{X}_C) \cdot \frac{\partial \hat{r}_3}{\partial \Phi} + f(\mathcal{X} - \mathcal{X}_C) \cdot \frac{\partial \hat{r}_1}{\partial \Phi}, \\ \frac{\partial G}{\partial \Phi} &= (y - y_p)(\mathcal{X} - \mathcal{X}_C) \cdot \frac{\partial \hat{r}_3}{\partial \Phi} + f(\mathcal{X} - \mathcal{X}_C) \cdot \frac{\partial \hat{r}_2}{\partial \Phi}, \end{aligned} \qquad (20)$$

where

$$\frac{\partial \hat{r}_1}{\partial \Phi} = \begin{bmatrix} 0 & -\sin\phi\cos\kappa & -\cos\phi\sin\kappa \\ \sin\omega\sin\kappa + \cos\omega\sin\phi\cos\kappa & \sin\omega\cos\phi\cos\kappa & -\cos\omega\cos\kappa - \sin\omega\sin\phi\sin\kappa \\ \cos\omega\sin\kappa - \sin\omega\sin\phi\cos\kappa & \cos\omega\cos\phi\cos\kappa & \sin\omega\cos\kappa - \cos\omega\sin\phi\sin\kappa \end{bmatrix},$$

$$\frac{\partial \hat{r}_2}{\partial \Phi} = \begin{bmatrix} 0 & -\sin\phi\sin\kappa & \cos\phi\cos\kappa \\ -\sin\omega\cos\kappa + \cos\omega\sin\phi\sin\kappa & \sin\omega\cos\phi\sin\kappa & -\cos\omega\sin\kappa + \sin\omega\sin\phi\cos\kappa \\ -\cos\omega\cos\kappa - \sin\omega\sin\phi\sin\kappa & \cos\omega\cos\phi\sin\kappa & \sin\omega\sin\kappa + \cos\omega\sin\phi\cos\kappa \end{bmatrix},$$

$$\frac{\partial \hat{r}_3}{\partial \Phi} = \begin{bmatrix} 0 & -\cos\phi & 0 \\ \cos\omega\cos\phi & -\sin\omega\sin\phi & 0 \\ \cos\omega\cos\phi\sin\kappa & -\cos\omega\sin\phi & 0 \end{bmatrix}.$$

The derivatives with respect to the internal calibration parameters are given below:

$$\begin{aligned}
\frac{\partial F}{\partial k_1} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{x}r^2, \quad \frac{\partial F}{\partial k_2} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{x}r^4, \quad \frac{\partial F}{\partial k_3} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{x}r^6, \\
\frac{\partial F}{\partial p_1} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ (r^2 + 2\bar{x}^2), \quad \frac{\partial F}{\partial p_2} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ 2\bar{x}\bar{y}, \\
\frac{\partial F}{\partial a_1} &= 0, \quad \frac{\partial F}{\partial a_2} = 0, \\
\frac{\partial F}{\partial x_p} &= -(\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \left(1 - \frac{\partial \Delta x}{\partial x_p}\right), \\
\frac{\partial F}{\partial y_p} &= -(\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \frac{\partial \Delta x}{\partial y_p}, \\
\frac{\partial F}{\partial f} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_1,
\end{aligned} \tag{21}$$

and

$$\begin{aligned}
\frac{\partial G}{\partial k_1} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{y}r^2, \quad \frac{\partial G}{\partial k_2} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{y}r^4, \quad \frac{\partial G}{\partial k_3} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{y}r^6, \\
\frac{\partial G}{\partial p_2} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ (r^2 + 2\bar{y}^2), \quad \frac{\partial G}{\partial p_1} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ 2\bar{x}\bar{y}, \\
\frac{\partial G}{\partial a_1} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{x}, \quad \frac{\partial G}{\partial a_2} = (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \bar{y}, \\
\frac{\partial G}{\partial x_p} &= -(\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \frac{\partial \Delta y}{\partial x_p}, \\
\frac{\partial G}{\partial y_p} &= -(\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_3\ \left(1 - \frac{\partial \Delta y}{\partial y_p}\right), \\
\frac{\partial G}{\partial f} &= (\mathcal{X} - \mathcal{X}_C)\cdot \hat{r}_2
\end{aligned} \tag{22}$$

where

$$
\begin{aligned}
\frac{\partial \Delta x}{\partial x_p} &= -(k_1 + k_2 r^2 + k_3 r^4) r^2 - 2\bar{x}^2 (k_1 + 2k_2 r^2 + 3k_3 r^4) - 6p_1 \bar{x} - 2p_2 \bar{y}, \\
\frac{\partial \Delta x}{\partial y_p} &= -2\bar{x}\bar{y}(k_1 + 2k_2 r^2 + 3k_3 r^4) - 2p_1 \bar{y} - 2p_2 \bar{x}, \\
\frac{\partial \Delta y}{\partial x_p} &= -2\bar{x}\bar{y}(k_1 + 2k_2 r^2 + 3k_3 r^4) - 2p_1 \bar{y} - 2p_2 \bar{x} - a_1, \\
\frac{\partial \Delta y}{\partial y_p} &= -(k_1 + k_2 r^2 + k_3 r^4) r^2 - 2\bar{y}^2 (k_1 + 2k_2 r^2 + 3k_3 r^4) - 2p_1 \bar{x} - 6p_2 \bar{y} - a_2.
\end{aligned}
\tag{23}
$$

References

[1] Abdel-Aziz, Y., and Karara, H. Direct linear transformation from comparator coordinates into object space coordinates. Proc. ASP/UI Symp. on Close-Range Photogrammetric Systems, Urbana, IL., pp. 1-48, 1971.

[2] Bell, J. H., and McLachlan, B. G. Image registration for pressure-sensitive paint applications. Experiments in Fluids, Vol. 22, pp. 78-86, 1996.

[3] Bell, J. H. Private Communication. 1998.

[4] Fraser, C. On the use of non-metric cameras in analytical close-range photogrammetry. Canadian Survey, Vol. 36, pp. 259-279, 1982.

[5] Granshaw, S. Bundle adjustment methods in engineering photogrammetry. Photogrammetry Record, Vol. 10, pp. 181-207, 1980.

[6] Marzan, G., and Karara, H. A computer program for direct linear transformation solution of the collinearity condition and some applications of it. Proc. APS Symp. on Close-Range Photogrammetric Systems, Urbana, IL., pp. 420-476, 1975.

[7] Milne-Thomson, L. M. Theoretical Aerodynamics. Dover Publications. New York, 1973.

[8] Naftel, A. J. and Boot, J. C. An iterative linear transform algorithm for solution of the collinearity equations. Photogrammetric Engineering and Remote Sensing, Vol. 57, No. 7, pp. 913-919, 1991.

[9] Press, W. H., Flannery, B. P., Teukolsky, S. A., Vetterling, W. T. Numerical Recipes in C. Cambridge University Press. Cambridge, 1988.

Table 1: Target locations for the Joukowski wing dataset (Example 1).

No.	Target Locations			No.	Target Locations		
	X	Y	Z		X	X	Z
1	3.205	-0.097	-12.750	24	-2.144	0.062	-6.750
2	2.289	-0.107	-3.375	25	-2.054	0.124	-9.000
3	2.099	-0.135	-1.875	26	-1.899	0.224	-11.625
4	2.134	-0.225	-3.375	27	-1.614	0.332	-12.375
5	2.925	-0.651	-14.250	28	-1.121	0.211	-1.875
6	2.397	-0.587	-10.125	29	-0.897	0.352	-5.250
7	1.505	-0.190	-0.750	30	-0.587	0.586	-10.125
8	1.829	-0.589	-8.250	31	-0.197	0.824	-13.875
9	1.225	-0.291	-2.625	32	0.192	0.951	-14.625
10	1.095	-0.386	-4.500	33	0.339	0.711	-7.875
11	0.648	-0.162	-1.125	34	0.674	0.789	-8.625
12	0.910	-0.781	-13.125	35	1.124	0.966	-11.625
13	-0.304	-0.334	-8.250	36	1.280	0.840	-9.000
14	-1.007	-0.315	-12.750	37	1.030	0.416	-1.125
15	-1.356	-0.170	-10.500	38	2.097	0.990	-13.125
16	-1.558	-0.022	-4.875	39	2.090	0.779	-9.750
17	-2.038	-0.062	-11.625	40	2.391	0.754	-10.875
18	-1.827	0.019	-1.125	41	2.839	0.766	-13.875
19	-2.035	0.011	-2.625	42	2.131	0.352	-4.125
20	-2.222	0.005	-4.125	43	1.939	0.185	-0.750
21	-2.098	0.001	-1.500	44	3.168	0.348	-13.125
22	-2.354	0.001	-5.250	45	2.795	0.162	-8.625
23	-2.454	0.009	-7.500	46	2.999	0.048	-10.500

Table 2: Coordinates of target points in the image collected by the camera at station 1 for the Joukowski wing dataset (Example 1). The image is of size 512×512, and the image coordinates are in pixels.

No.	Image Coordinates		No.	Image Coordinates	
	x	y		x	y
1	190.662	146.457	24	160.456	224.876
2	107.044	256.018	25	176.300	204.277
3	95.147	272.707	26	194.569	179.656
4	109.632	257.106	27	198.479	171.013
5	212.553	136.367	28	116.458	267.469
6	174.584	185.310	29	140.216	234.806
7	90.885	284.589	30	175.208	186.178
8	160.174	207.777	31	201.894	146.481
9	110.841	265.472	32	205.808	135.855
10	129.142	246.778	33	151.325	204.287
11	101.105	279.287	34	155.103	195.114
12	208.778	161.670	35	177.012	160.952
13	167.772	210.771	36	154.601	188.768
14	205.799	170.129	37	92.015	274.746
15	187.955	191.079	38	186.246	139.137
16	144.309	241.727	39	157.572	178.074
17	197.497	182.269	40	166.679	164.445
18	117.047	275.284	41	192.987	127.665
19	129.630	261.814	42	110.244	243.567
20	141.855	248.571	43	82.843	281.860
21	121.703	271.856	44	189.287	137.977
22	150.844	238.776	45	149.809	193.604
23	167.733	219.143	46	168.010	172.339

Table 3: Coordinates of target points in the image collected by the camera at station 2 for the Joukowski wing dataset (Example 1). The image is of size 512×512, and the image coordinates are in pixels.

No.	Image Coordinates		No.	Image Coordinates	
	x	y		x	y
1	273.318	266.992	24	196.820	341.550
2	253.257	393.849	25	201.420	310.587
3	249.113	416.495	26	207.185	276.344
4	251.268	394.358	27	211.936	267.293
5	272.096	246.787	28	204.016	412.877
6	261.764	299.804	29	212.029	362.831
7	239.547	434.061	30	222.415	297.045
8	252.447	324.952	31	231.513	251.418
9	237.834	405.564	32	236.943	243.310
10	238.230	377.750	33	231.682	327.042
11	227.792	427.650	34	236.750	317.417
12	245.354	258.497	35	245.249	280.090
13	223.538	322.788	36	244.725	313.051
14	219.798	262.095	37	233.126	423.883
15	212.420	291.232	38	258.276	263.039
16	201.994	369.081	39	255.563	304.222
17	205.038	275.874	40	260.278	290.383
18	192.432	425.380	41	268.050	254.517
19	191.853	402.181	42	251.325	380.749
20	191.578	379.572	43	245.391	431.519
21	189.092	419.533	44	272.227	263.058
22	191.506	363.001	45	264.379	319.792
23	193.621	330.905	46	268.712	295.485

Table 4: Coordinates of target points in the image collected by the camera at station 3 for the Joukowski wing dataset (Example 1). The image is of size 512×512, and the image coordinates are in pixels.

No.	Image Coordinates		No.	Image Coordinates	
	x	y		x	y
1	337.370	315.663	24	275.158	235.344
2	338.494	193.709	25	274.256	263.767
3	338.016	172.490	26	273.800	296.027
4	336.123	193.743	27	276.530	305.758
5	331.017	329.827	28	293.282	170.472
6	329.749	280.974	29	292.566	216.949
7	331.159	156.270	30	291.402	280.842
8	324.820	257.256	31	292.451	328.188
9	324.480	182.952	32	296.587	338.494
10	320.130	208.458	33	306.162	253.432
11	318.596	161.344	34	309.823	263.852
12	307.405	313.285	35	312.269	303.571
13	297.457	255.452	36	317.657	269.655
14	284.349	308.228	37	324.532	159.827
15	281.972	281.989	38	323.479	323.971
16	284.542	211.460	39	327.750	280.318
17	272.525	294.754	40	330.297	295.116
18	284.647	160.137	41	332.208	333.185
19	280.418	180.911	42	335.878	203.741
20	276.576	201.061	43	338.005	155.220
21	280.674	165.361	44	337.050	322.356
22	273.887	215.788	45	338.187	264.851
23	270.691	244.470	46	338.046	288.547

Table 5: Coordinates of target points in the image collected by the camera at station 4 for the Joukowski wing dataset (Example 1). The image is of size 512×512, and the image coordinates are in pixels.

No.	Image Coordinates		No.	Image Coordinates	
	x	y		x	y
1	337.370	315.663	24	275.158	235.344
2	338.494	193.709	25	274.256	263.767
3	338.016	172.490	26	273.800	296.027
4	336.123	193.743	27	276.530	305.758
5	331.017	329.827	28	293.282	170.472
6	329.749	280.974	29	292.566	216.949
7	331.159	156.270	30	291.402	280.842
8	324.820	257.256	31	292.451	328.188
9	324.480	182.952	32	296.587	338.494
10	320.130	208.458	33	306.162	253.432
11	318.596	161.344	34	309.823	263.852
12	307.405	313.285	35	312.269	303.571
13	297.457	255.452	36	317.657	269.655
14	284.349	308.228	37	324.532	159.827
15	281.972	281.989	38	323.479	323.971
16	284.542	211.460	39	327.750	280.318
17	272.525	294.754	40	330.297	295.116
18	284.647	160.137	41	332.208	333.185
19	280.418	180.911	42	335.878	203.741
20	276.576	201.061	43	338.005	155.220
21	280.674	165.361	44	337.050	322.356
22	273.887	215.788	45	338.187	264.851
23	270.691	244.470	46	338.046	288.547

Table 6: Camera parameters for the Joukowski wing dataset (Example 1). Note that the RMS error and the principal locations (x_p, y_p) are in pixels, while the orientation angles are in degrees.

Parameter	Station 1	Station 2	Station 3	Station 4
X_C	20.000	0.000	0.000	-10.000
Y_C	-8.000	-20.000	20.000	10.000
Z_C	-10.000	-10.000	-5.000	0.000
ω	1.381	-80.000	80.000	60.000
ϕ	90.000	5.000	-10.000	30.000
κ	51.381	5.000	5.000	45.000
k_1	-4.053E-7	-4.053E-7	-4.053E-7	-4.053E-7
k_2	-2.179E-6	-2.179E-6	-2.179E-6	-2.179E-6
k_3	4.107E-5	4.107E-5	4.107E-5	4.107E-5
p_1	1.052E-9	1.052E-9	1.052E-9	1.052E-9
p_2	8.859E-8	8.859E-8	8.859E-8	8.859E-8
a_1	-2.334E-9	-2.334E-9	-2.334E-9	-2.334E-9
a_2	-6.064E-9	-6.064E-9	-6.064E-9	-6.064E-9
x_p	1.000	1.000	1.000	1.000
y_p	1.000	1.000	1.000	1.000
f	0.500	0.500	0.500	0.500
x RMS Error	5.040E-6	3.470E-06	4.070E-06	3.535E-06
y RMS Error	3.866E-6	2.788E-06	2.388E-06	6.169E-06

Table 7: Target locations for a Boeing airplane dataset (example 2).

No.	Target locations		
	X	Y	Z
1	1607.182	115.070	227.882
2	2032.488	97.430	231.272
3	2034.920	262.636	221.354
4	2290.974	93.098	231.192
5	2289.736	336.418	219.346
6	2456.156	235.920	226.210
7	2573.540	92.632	224.990
8	2577.158	422.288	220.214
9	2820.132	208.338	215.554
10	2783.998	373.076	219.732
11	2717.762	551.874	221.234
12	2874.836	741.880	221.912
13	1903.644	245.944	213.790
14	2088.480	322.896	197.480
15	2178.304	350.492	198.340
16	2426.106	435.612	198.062
17	2645.676	604.052	202.698
18	2869.044	568.930	213.628

Table 8: Measured and calculated image locations, in pixels, for camera at station 1 for the Boeing dataset (Example 2).

No.	Measured		Calculated	
	x	y	x_{calc}	y_{calc}
1	462.362	103.700	462.369	103.701
2	472.600	367.667	472.791	367.531
3	373.792	370.304	373.621	370.230
4	473.892	521.270	473.837	521.459
5	331.696	519.208	331.713	519.089
6	389.500	614.500	389.474	614.347
7	471.167	680.571	470.999	680.585
8	285.167	677.500	285.035	677.350
9	403.333	807.333	403.382	807.255
10	313.056	786.139	313.192	786.158
11	216.000	748.083	216.262	748.666
12	120.611	822.389	120.641	822.225
13	383.222	291.444	383.156	291.502
14	339.267	401.867	323.947	454.114
15	323.783	454.317	339.303	402.114
16	277.933	592.467	277.922	592.604
17	189.500	707.500	189.416	707.300
26	210.500	822.500	210.319	822.526

Table 9: Measured and calculated image locations, in pixels, for camera at station 2 for the Boeing dataset (Example 2).

No.	Measured		Calculated	
	x	y	x_{calc}	y_{calc}
1	455.778	154.861	455.800	154.794
2	471.736	358.264	471.947	358.090
3	374.000	363.583	373.839	363.730
4	478.204	501.444	478.263	501.570
5	327.167	503.429	327.124	503.329
6	387.903	601.097	387.768	601.140
7	481.600	677.000	481.429	676.947
8	267.042	675.137	266.967	675.094
9	403.000	840.905	403.168	840.768
10	292.892	811.175	292.935	811.069
11	179.104	761.340	179.548	761.946
12	51.852	858.571	51.845	858.545
13	384.111	299.444	384.075	299.653
14	338.889	398.556	338.954	398.631
15	321.444	446.444	321.520	446.285
16	265.056	586.333	265.005	586.267
17	153.389	716.611	153.220	716.359
18	162.944	861.944	162.700	861.915

Table 10: Measured and calculated image locations, in pixels, for camera at station 3 for the Boeing dataset (Example 2).

No.	Measured		Calculated	
	x	y	x_{calc}	y_{calc}
1	456.611	144.778	456.536	144.796
2	471.905	359.000	472.068	358.931
3	374.000	364.345	373.749	364.338
4	477.750	506.250	477.665	506.364
5	327.833	507.571	327.716	507.587
6	388.000	606.738	387.969	606.672
7	479.958	682.204	479.854	682.244
8	269.861	680.139	269.660	680.114
9	403.000	842.329	403.149	842.443
10	296.000	813.905	296.116	813.729
11	185.056	765.151	185.312	765.781
12	63.179	860.000	63.179	859.858
13	383.867	297.233	383.903	297.258
14	338.917	399.250	338.986	399.362
15	321.611	448.611	321.843	448.401
16	266.833	590.445	266.906	590.426
17	158.933	720.467	158.887	720.118
18	170.681	862.755	170.500	862.758

Table 11: Camera parameters for the Boeing dataset (example 2). Note that the RMS errors are given in pixels, while the orientation angles are given in degrees.

Parameter	Station 1	Station 2	Station 3
X_C	2056.68	2881.36	2773.12
Y_C	209.00	201.18	197.04
Z_C	1961.96	1763.88	1814.32
ω	-8.821	-9.476	-9.434
ϕ	8.948	-18.402	-14.592
κ	88.806	93.089	-24.335
k_1	0.163	0.163	0.163
k_2	-0.435	-0.435	-0.435
k_3	0.901	0.901	0.901
p_1	9.063 E-3	9.063 E-3	9.063 E-3
p_2	0.003	0.003	0.003
a_1	-5.496 E-3	-5.496 E-3	-5.496 E-3
a_2	4.362 E-4	4.362 E-4	4.362 E-4
x_p	-3.814 E-3	-3.814 E-3	-3.814 E-3
y_p	0.031	0.031	0.031
f	0.955	0.955	0.955
x RMS Error	0.124	0.160	0.143
y RMS Error	0.191	0.187	0.192