constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 565-593, 1986.
[7] H. H. Nagel, "On the estimation of optical flow: Relations between different approaches and some new results," *Artificial Intell.*, vol. 33, pp. 299-324, 1987.
[8] J. M. Prager and M. A. Arbib, "Computing the optical flow: The MATCH algorithm and prediction," *Comput. Vision, Graphics, Image Processing*, vol. 24, pp. 271-304, 1983.
[9] K. Prazdny, "On the information in optical flows," *Comput. Vision, Graphics, Image Processing*, vol. 22, pp. 239-259, 1983.
[10] W. B. Thompson, "Combining motion and contrast for segmentation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 543-549, 1980.

A Unified Approach to the Linear Camera Calibration Problem

WILLIAM I. GROSKY AND LOUIS A. TAMBURINO

***Abstract*—The camera calibration process relates camera system measurements (pixels) to known reference points in a three-dimensional world coordinate system. In this correspondence, the calibration process is viewed as consisting of two independent phases: the first is removing geometrical camera distortion so that rectangular calibration grids are straightened in the image plane, while the second is using a linear affine transformation as a map between the rectified camera coordinates and the geometrically projected coordinates on the image plane of known reference points. Phase one is camera dependent, and in some systems may not be necessary. Phase two is concerned with a generic model that includes 12 extrinsic variables and up to 5 intrinsic parameters.**

The generic extrinsic variables include a rotation matrix describing the orientation of the optical axis and the displacements of the camera's focal point in the world coordinate system. The intrinsic variables correct for scale, displacement of the optical axis, and skewing of the coordinate axis in the camera coordinate system. Although there are three independent rotation angles, we treat the components of the rotation matrix as nine extrinsic parameters satisfying six constraint equations. We present general methods which handle additional constraints on the intrinsic variables in a manner consistent with explicit satisfaction of all six constraints on the orthogonal rotation matrix. We describe the use of both coplanar and noncoplanar calibration points. There are fewer equations in the coplanar case; therefore, it is necessary for the user to supply up to three additional constraint equations.

***Index Terms*—Affine transformations, extrinsic parameters, intrinsic parameters, least squares, linear camera calibration**

I. Introduction

The Numerical Stereo Camera System [1], [7] which resides in the Avionics Laboratory of Wright-Patterson Air Force Base utilizes both a passive as well as an active camera to recover 3-D scene information. This is accomplished by solving an overdetermined system of linear equations by the well-known method of least-squares [6]. Specifically, for each camera there are two linear equations in the parameters x_W, y_W, and z_W the world coordinates of a given scene point which is to be determined, where the coefficients are specific functions of x^* and y^*, the known image coordinates of the projection of the given scene point, as well as of the camera geometry (*extrinsic parameters*) and the camera optics (*intrinsic parameters*). The extrinsic parameters give information regarding the camera position and orientation with respect to the world coordinate system, while the intrinsic parameters include the focal length, scale factors which go from units of length to pixels in the image plane, the intersection point of the camera axis with the image plane expressed in pixels, as well as values expressing the different types of possible lens distortions. The term *camera calibration* refers to finding the values of these parameters for a given camera set-up so that the coefficients of x_W, y_W, and z_W in these linear equations can be calculated as functions of x^* and y^*.

There has been much previous work in this area. The techniques to solve this problem range from simple linear equation solving to complex nonlinear optimization approaches. The latter methods have been used by [4], [8], but are extremely inefficient and must be manually guided. The former methods, most notably used by [5], [9], [10], while efficient, tend to ignore constraints which the extrinsic and intrinsic parameters must obey. These latter shortcomings become worse when an increasing number of parameters are specified in advance.

In this correspondence, we present unified solutions for many interesting subcases of this problem. Most importantly, our solutions satisfy all the necessary constraints as well as being relatively simple to compute.

The organization of this correspondence is as follows. Section II derives the linear camera calibration equations. In Section III, we present our unified solution technique as a *1-step method* of solution, while Section IV illustrates various subcases of the problem which may be solved using this method. A companion *2-step method* is developed in Section V. Section VI presents some experiments we have conducted using our techniques. Finally, in Section VII we offer our conclusions.

II. The General Linear Camera Calibration Problem

We start with a world coordinate system as shown in Fig. 1. We would like to express points in this system with respect to a camera-centered system, x_C, y_C, and z_C, where the camera axis z_C points along the $-z_W$ direction and the (x_C, y_C)-plane is parallel to the image plane. To accomplish this, we first translate the origin of the world system to the focal point of our camera (x_F, y_F, z_F) and then apply a pan θ about the y-axis; a tilt ϕ about the x-axis; and finally, a roll ψ about the z-axis. See Fig. 1. A point $(x_W, y_W, z_W, 1)$ expressed in homogeneous coordinates in the world-frame is mapped into the camera-centered system $(x_C, y_C, z_C, 1)$ by the following transformation [3]:

$$(x_C, y_C, z_C, 1) = (x_W, y_W, z_W, 1) \begin{bmatrix} R_{11} & R_{12} & R_{13} & 0 \\ R_{21} & R_{22} & R_{23} & 0 \\ R_{31} & R_{32} & R_{33} & 0 \\ -D_1 & -D_2 & -D_3 & 1 \end{bmatrix} \tag{1}$$

where

$$R_{11} = \cos\theta \cos\psi + \sin\theta \sin\phi \sin\psi,$$
$$R_{12} = -\cos\theta \sin\psi + \sin\theta \sin\phi \cos\psi,$$
$$R_{13} = \sin\theta \cos\psi,$$
$$R_{21} = \cos\phi \sin\psi,$$
$$R_{22} = \cos\phi \cos\psi,$$
$$R_{23} = -\sin\phi,$$
$$R_{31} = -\sin\theta \cos\psi + \cos\theta \sin\phi \sin\psi,$$

Manuscript received March 9, 1987; revised November 10, 1989. Recommended for acceptance by O.D. Faugeras. This work was supported Universal Energy Systems under Subcontract S-760-6MG-070, AFOSR Contract F49620-85-C-0013 and by the Institute for Manufacturing Research at Wayne State University.

W. I. Grosky is with the Department of Computer Science, Wayne State University, Detroit, MI 48202.
L. A. Tamburino is with the Systems Avionics Division, Wright-Patterson Air Force Base, Dayton, OH 45433.

IEEE Log Number 9036117.

$$R_{32} = \sin\theta \sin\psi + \cos\theta \sin\phi \cos\psi,$$

$$R_{33} = \cos\theta \cos\phi,$$

$$D_1 = x_F R_{11} + y_F R_{21} + z_F R_{31},$$

$$D_2 = x_F R_{12} + y_F R_{22} + z_F R_{32},$$

$$D_3 = x_F R_{13} + y_F R_{23} + z_F R_{33}.$$

Letting $\boldsymbol{T}$ be the matrix of (1), $\boldsymbol{C} = (x_C, y_C, z_C, 1)$, and $\boldsymbol{W} = (x_W, y_W, z_W, 1)$, (1) becomes

$$\boldsymbol{C}_j = \boldsymbol{W}_i \boldsymbol{T}_{ij}, \tag{2}$$

for $1 \le j \le 4$, where we use the notation of repeating subscripts being summed. Using the standard projection equations [3], the screen coordinates (x', y') satisfy

$$x' = -F\frac{x_C}{z_C} \tag{3a}$$

$$y' = -F\frac{y_C}{z_C}, \tag{3b}$$

where F is the focal length of the camera and the image plane is defined by $z_C = -F$.

Measurements on the calibration target image are made in raster or pixel coordinates (x^*, y^*). We characterize our linear calibration model by the following affine transformation which relates the measurements in raster coordinates to those in screen coordinates as follows:

$$x^* = x_0 + c_{11}x' + c_{12}y' \tag{4a}$$

$$y^* = y_0 + c_{21}x' + c_{22}y', \tag{4b}$$

or, reformatted in a manner to highlight the combination of rotation, scaling, and translation,

$$x^* = x_0 + p_x(x' \cos\omega + y' \sin\omega) \tag{5a}$$

$$y^* = y_0 + p_y(x' \sin\upsilon + y' \cos\upsilon), \tag{5b}$$

where υ, ω are rotation angles, p_x, p_y are scaling parameters in units of pixels/unit length, and x_0, y_0 are the displacements of the optical axis in raster coordinates.

In this calibration formalism, we do not include nonlinear distortions or warping. In practice, that type of distortion is handled separately before applying the linear calibration model. In the Numerical Stereo Camera System, we use bivariate polynomials to find a mathematical transformation to take us from the measured and distorted image coordinates (x_d, y_d), in pixels, to a new pixel coordinate system (x^*, y^*), in which the camera distortion is removed. The form of these polynomials is

$$x^* = x^*(x_d, y_d) = p_0 + p_1x_d + p_2y_d + p_3x_dy_d + p_4x_d^2 + p_5y_d^2 + \cdots \tag{6a}$$

$$y^* = y^*(x_d, y_d) = q_0 + q_1x_d + q_2y_d + q_3x_dy_d + q_4y_d^2 + q_5y_d^2 + \cdots. \tag{6b}$$

Equations (6a) and (6b) are used with a special calibration target which has an accurate rectangular grid. This grid appears warped when viewed on the monitor. These polynomials are used to map the warped grid images into undistorted rectangular grids [2]. The warping correction straightens out the grid lines. We shall assume that this type of mapping has been done prior to the application of the affine transformations of (4a) and (4b), which enables one to complete the calibration process. These affine transformations address scaling, displacement of the optical center, and any skewing of the coordinate axis in the corrected camera coordinate system (x^*, y^*).

Subsequent analysis reveals that it is only possible to solve for the variable combinations p_xF, p_yF, and υ-ω, where F is the focal length defined in (3a) and (3b). It is convenient in subsequent computations to re-express the affine transformations in the following form:

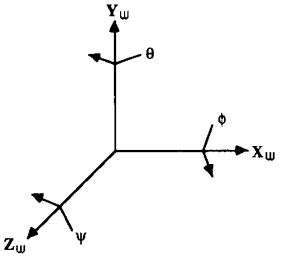

Fig. 1. World coordinate system.

$$x^* = x_0 + P_x\frac{a_{11}x' + a_{12}y'}{F} \tag{7a}$$

$$y^* = y_0 + P_y\frac{a_{21}x' + a_{22}y'}{F}, \tag{7b}$$

where $P_x = p_xF$, $P_y = p_yF$, $a_{11} = \cos\omega$, $a_{21} = \sin\omega$, $a_{21} = \sin\upsilon$, and $a_{22} = \cos\upsilon$. Using the standard projection equations (3a) and (3b), we get

$$x^* = x_0 - P_x\frac{a_{11}x_C + a_{12}y_C}{z_C} \tag{8a}$$

$$y^* = y_0 - P_y\frac{a_{21}x_C + a_{22}y_C}{z_C}. \tag{8b}$$

Substitution of (2) into (8a) and (8b) provides the following linear camera calibration equations, which relate the pixel and world coordinates:

$$\boldsymbol{W}_i[\boldsymbol{T}_{i3}(x^* - x_0) + P_x(a_{11}\boldsymbol{T}_{i1} + a_{12}\boldsymbol{T}_{i2})] = 0 \tag{9a}$$

$$\boldsymbol{W}_i[\boldsymbol{T}_{i3}(y^* - y_0) + P_y(a_{21}\boldsymbol{T}_{i1} + a_{22}\boldsymbol{T}_{i2})] = 0. \tag{9b}$$

The following abbreviated form for these basic equations highlights the 12 independent coefficients $\boldsymbol{T}_{i3}$, X_i, and Y_i:

$$x^*\boldsymbol{W}_{i3}\boldsymbol{T}_{i3} + \boldsymbol{W}_iX_i = 0 \tag{10a}$$

$$y^*\boldsymbol{W}_i\boldsymbol{T}_{i3} + \boldsymbol{W}_iY_i = 0, \tag{10b}$$

where three of the coefficients are the four extrinsic variables $T_{.3} = [R_{13}, R_{23}, R_{33}, -D_3]$, and the other eight are nonlinear functions of the camera model parameters $X_i = P_x(a_{11}T_{i1} + a_{12}T_{i2}) - x_0T_{i3}$, for $1 \le i \le 3$, $Y_i = P_y(a_{21}T_{i1} + a_{22}T_{i2}) - y_0T_{i3}$, for $1 \le i \le 3$, $X_4 = -P_x(a_{11}D_1 + a_{12}D_2) + x_0D_3$, and $Y_4 = -P_y(a_{21}D_1 + a_{22}D_2) - y_0D_3$. This form of (10a) and (10b) eliminates the need to write the cumbersome products p_xF and p_yF because it effectively absorbs the focal length into P_x and P_y.

Calibration consists of solving (10a) and (10b) for R_{ij}, $1 \le i, j \le 3$, P_x, P_y, x_0, y_0, D_1, D_2, D_3, and a_{ij}, $1 \le i, j \le 2$, so that one can finally find the 3-D coordinates of a scene point corresponding to a known image point, by substituting known values for x^* and y^* in (10a) and (10b), resulting in two linear equations for the unknowns x_W, y_W, and z_W. (Note that by utilizing a second camera, we will have four equations for these three unknowns.) We solve for the above unknowns by substituting known values for x^* and y^*, as well as known values for the corresponding x_W, y_W, and z_W. Note that x_F, y_F, and z_F can be easily calculated from values for the above variables, utilizing the definitions of D_1, D_2, and D_3.

The a_{ij} will mask some of the ambiguity of deciding what to do with the two angles υ and ω. The choice of $\upsilon = \omega$, for example, is superfluous because this represents a rotation about the optical axis that is included in the extrinsic parameters. Without loss of generality, we may choose to set either υ or ω equal to 0, with the

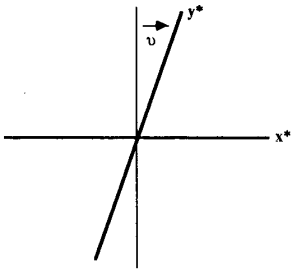

Fig. 2. Skew angle.

TABLE I
LINEAR CALIBRATION MODEL

Unknowns	Symbol	Quantity	Type
Rotation Matrix	R_{ab}	9	Extrinsic
Displacement	D_a	3	Extrinsic
Scaling	P_x	1	Intrinsic
Aspect Ratio	$\sigma = P_y/P_x$	1	Intrinsic
Displacement	x_O, y_O	2	Intrinsic
Skew Angle	υ	1	Intrinsic
	Total	17	

remaining angle representing a skew angle for a nonorthogonal coordinate axis. See Fig. 2.

Eliminating one angle also provides a proper match of 17 unknowns with the maximum number of available equations. See Table I. Hence, it provides the most general linear model. In order to remove the confusing two angle notation, we shall assume in the remainder of this correspondence that ω is identically equal to 0; that is, $a_{11} = 1$, $a_{12} = 0$, $a_{21} = \sin \upsilon$, and $a_{22} = \cos \upsilon$.

The object of this correspondence is to review methods of handling different numbers of intrinsic variables and constraints in a manner that is also consistent with the explicit satisfaction of all six constraints for the rotation matrix components. Although there are three independent rotation angles—pan, tilt, and roll—we treat the components of R_{ij} as 12 extrinsic parameters satisfying six constraint equations. This correspondence also distinguishes the use of both coplanar and noncoplanar calibration points. Because coplanar points involve fewer equations, it is necessary to supply at least three additional constraints. This reduces the number of unknowns from 17 to 14 in order to match the number of equations available in this case. Usually, we impose these additional constraints on the intrinsic variables.

III. A Unified Solution Technique and the One-Step Method

In this section, we show how to reduce the basic camera calibration equations (10a) and (10b), which form a set of homogeneous equations with 12 independent coefficients T_{i3}, X_i, and Y_i, for $1 \le i \le 4$, into sets of nonlinear equations which can easily be used to solve for the intrinsic and extrinsic model parameters. Since we have a homogeneous system, we put a proper subset of the 12 coefficients on the right side of the equations and solve for the remaining coefficients in terms of the coefficients in the given subset. This is done via the technique of least-squares [6]. The size of the subset put on the right side is proportional to the amount of degeneracy in the problem. The approach in which we use both (10a) and (10b) simultaneously, is called the *1-step method*.

Specifically, suppose we have N scene points (x_j, y_j, z_j), $1 \le j \le N$, in the world system, as well as their corresponding image coordinates

$$(x_j^*, y_j^*).$$

Our homogeneous system would then consist of $2N$ equations. Let $\{\lambda_1, \cdots \lambda_k\}$, $1 \le k < 12$, be a subset of the 12 coefficients which will be on the left-hand side of our equations, while $\{\rho_1, \cdots, \rho_{12-k}\}$ comprise the remaining coefficients. Our system then takes the form $\boldsymbol{JG} = \boldsymbol{KP}$, where $\boldsymbol{J}$ is a $2N \times k$ matrix of coefficients, $\boldsymbol{G}^T = (\lambda_1, \cdots, \lambda_k)$, $\boldsymbol{K}$ is a $2N \times (12 - k)$ matrix of coefficients, and $\boldsymbol{P}^T = (\rho_1, \cdots, \rho_{12-k})$. We must have $2N \ge k$, and in general, we will have $2N > k$, so that we have an overdetermined system of linear equations to be solved via least-squares. This is done as follows. We have $\boldsymbol{J}^T\boldsymbol{JG} = \boldsymbol{J}^T\boldsymbol{KP}$. Notice that $\boldsymbol{J}^T\boldsymbol{J}$ is a square matrix. Thus, $\boldsymbol{G} = (\boldsymbol{J}^T\boldsymbol{J})^{-1}\boldsymbol{J}^T\boldsymbol{KP}$, if $(\boldsymbol{J}^T\boldsymbol{J})^{-1}$ exists, which will be the case if the $2N$ points do not all lie on the same plane. The result of this calculation expresses k coefficients in terms of the remaining $12 - k$. In solving for our 20 original coefficients, we would then have k equations of the form $\lambda_j = \boldsymbol{MP}$, for $\boldsymbol{M}$ a row vector of coefficients and $1 \le j \le k$. Note that we do not advocate actually computing $\boldsymbol{M}$ as it is defined above. There are other, more numerically stable techniques, such as the singular value decomposition [6] which may be used. We also have six independent constraints on the quantities R_{ij}, $1 \le i, j, \le 3$, which express that these are elements of a rotation matrix. These constraints can be written in numerous ways: either that each row is a unit vector and is orthogonal to each of the other two rows, or that each column is a unit vector and is orthogonal to each of the other two columns, or that some two rows are unit vectors, orthogonal to each other, while the remaining row is the cross-product of the two given rows, or that some two columns are unit vectors, orthogonal to each other, while the remaining column is the cross-product of the two given columns.

We thus have $k + 6$ equations, $k \le 11$, which are sufficient to find the 17 extrinsic and intrinsic unknown variables. If there are any *a priori* constraints on these variables which reduce the number of unknowns, it is necessary to reduce accordingly the value of k from its maximum value of 11. When this is done, we say we have a *degeneracy*. Highly degenerate cases can result in more complicated computations involving systems of multivariate polynomials. This is more clearly indicated below.

For a particular value of k, using at least $\lceil k/2 \rceil$ noncoplanar 3-D points (x_i, y_i, z_i) as well as their known projections

$$(x_j^*, y_j^*).$$

We get, via the method of least-squares, equations of the form

$$P_x(a_{11}R_{i1} + a_{12}R_{i2}) - x_0R_{i3} = A_i, \quad \text{for } 1 \le i \le 3, \tag{11a-11c}$$

$$P_y(a_{21}R_{i1} + a_{22}R_{i2}) - y_0R_{i3} = B_i, \quad \text{for } 1 \le i \le 3, \tag{12a-12c}$$

$$R_{i3} = C_i, \quad \text{for } 1 \le i \le 3, \tag{13a-13c}$$

$$x_0D_3 - P_x(a_{11}D_1 + a_{12}D_2) = E_1, \tag{14a}$$

$$y_0D_3 - P_y(a_{21}D_1 + a_{22}D_2) = E_2, \tag{14b}$$

$$D_3 = G, \tag{14c}$$

where the right-hand side of each equation is a known linear combination of the unknowns in $\boldsymbol{P} = \{\rho_1, \cdots, \rho_{12-k}\}$. Some of these equations may be identities, however, depending on the nature of $\boldsymbol{P}$. For example, if $\boldsymbol{P} = \{D_3, R_{33}\}$, then (13c) and (14c) are not found via least-squares, but are the identities $R_{33} = 0 \cdot D_3 + 1 \cdot R_{33}$ and $D_3 = 1 \cdot D_3 + 0 \cdot R_{33}$, respectively.

Let us define $\boldsymbol{A} = (A_1, A_2, A_3)$, $\boldsymbol{B} = (B_1, B_2, B_3)$, $\boldsymbol{C} = (C_1, C_2, C_3)$, $\boldsymbol{E} = (E_1, E_2)$, $\Omega_1 = (R_{11}, R_{21}, R_{31})$, $\Omega_2 = (R_{12}, R_{22}, R_{32})$, $\Omega_3 = (R_{31}, R_{32}, R_{33})$, $\tau_1 = (a_{11}, a_{12}, 0) = (1, 0, 0)$, and $\tau_2 = (a_{21}, a_{22}, 0) = (\sin \upsilon, \cos \upsilon, 0)$. Note that $|\tau_1| = |\tau_2| = 1$, $\tau_1 \cdot \tau_2 = \sin \upsilon$ and $\tau_1 \times \tau_2 = \cos \upsilon$. Then (11a)-(13c) may be

expressed as the three vector equations

$$P_x(a_{11}\Omega_1 + a_{12}\Omega_2) - x_0\Omega_3 = \boldsymbol{A}, \tag{15a}$$

$$P_y(a_{21}\Omega_1 + a_{22}\Omega_2) - y_0\Omega_3 = \boldsymbol{B}, \tag{15b}$$

$$\Omega_3 = \boldsymbol{C}. \tag{15c}$$

The six necessary constraints can now be written as

$$\Omega_i \cdot \Omega_j = \delta_{ij}, \quad \text{for } 1 \le i, j \le 3, \tag{16}$$

where δ_{ij} is the Kronecker delta function.

Using the above equations and defining $\boldsymbol{M} = \boldsymbol{A} - (\boldsymbol{A} \cdot \boldsymbol{C})\boldsymbol{C} = \boldsymbol{A} + x_0\boldsymbol{C}$ and $\boldsymbol{N} = \boldsymbol{B} - (\boldsymbol{B} \cdot \boldsymbol{C})\boldsymbol{C} = \boldsymbol{B} + y_0\boldsymbol{C}$, we can easily derive that

$$x_0 = -\boldsymbol{A} \cdot \boldsymbol{C}, \tag{17a}$$

$$y_0 = -\boldsymbol{B} \cdot \boldsymbol{C}, \tag{17b}$$

$$P_x = |\boldsymbol{A} - (\boldsymbol{A} \cdot \boldsymbol{C})\boldsymbol{C}| = |\boldsymbol{A} \times \boldsymbol{C}|, \tag{18a}$$

$$P_y = |\boldsymbol{B} - (\boldsymbol{B} \cdot \boldsymbol{C})\boldsymbol{C}| = |\boldsymbol{B} \times \boldsymbol{C}|. \tag{18b}$$

Equations (15a)–(15b) then become two simultaneous linear equations for the two unknowns Ω_1 and Ω_2. Solving, we get that

$$\Omega_1 = \frac{\boldsymbol{M}}{|\boldsymbol{M}|}, \tag{19a}$$

$$\Omega_2 = \sec\upsilon \frac{\boldsymbol{N}}{|\boldsymbol{N}|} - \tan\upsilon \frac{\boldsymbol{M}}{|\boldsymbol{M}|}, \tag{19b}$$

$$\Omega_3 = \boldsymbol{C}. \tag{19c}$$

Using

$$\boldsymbol{C} \cdot \boldsymbol{C} = 1, \tag{20}$$

one can verify that these solutions satisfy all the constraints of (16).

Using (14a) and (14b), we see that

$$D_1 = -\frac{(\boldsymbol{A} \cdot \boldsymbol{C})\, G + E_1}{|\boldsymbol{M}|}, \tag{21a}$$

$$D_2 = \tan\upsilon \frac{(\boldsymbol{A} \cdot \boldsymbol{C})\, G + E_1}{|\boldsymbol{M}|} - \sec\upsilon \frac{(\boldsymbol{B} \cdot \boldsymbol{C})\, G + E_2}{|\boldsymbol{N}|}. \tag{21b}$$

Finally, from (15a) and (15b), we have the following equation which relates τ_1 and τ_2:

$$\tau_1 \cdot \tau_2 = \frac{\boldsymbol{A} \cdot \boldsymbol{B} - (\boldsymbol{A} \cdot \boldsymbol{C})(\boldsymbol{B} \cdot \boldsymbol{C})}{|\boldsymbol{A} - (\boldsymbol{A} \cdot \boldsymbol{C})\boldsymbol{C}|\,|\boldsymbol{B} - (\boldsymbol{B} \cdot \boldsymbol{C})\boldsymbol{C}|}, \tag{22a}$$

$$\sin\upsilon = \frac{\boldsymbol{M} \cdot \boldsymbol{N}}{|\boldsymbol{M}|\,|\boldsymbol{N}|}. \tag{22b}$$

Taking cross-products of (15a)–(15c) does not produce any new equations when one uses the identity $(\Sigma_1 \times \Sigma_2) \cdot (\Sigma_3 \times \Sigma_4) = (\Sigma_1 \cdot \Sigma_3)(\Sigma_2 \cdot \Sigma_4) - (\Sigma_1 \cdot \Sigma_4)(\Sigma_2 \cdot \Sigma_3)$, along with (20).

The above equations do not, as yet, provide numerical solutions for our parameters, as $\boldsymbol{A}$, $\boldsymbol{B}$, and $\boldsymbol{C}$ are linear combinations of $12 - k$ unknowns. We will next show how to use these equations to solve for these unknowns for various subcases of our problem.

IV. Some Illustrative Solved Subcases for the One-Step Method

We begin with the most general case, which has 17 model parameters, because it is both interesting and easy to follow. This is the case with no constraints on the skew angle. For this case, set $\boldsymbol{P} = \{D_3\}$. Thus, $\boldsymbol{A} = \alpha D_3$, $\boldsymbol{B} = \beta D_3$, $\boldsymbol{C} = \gamma D_3$, $\boldsymbol{E} = \epsilon D_3$, and $G = 1 \cdot D_3$, where α, β, γ, and ϵ are numerically known three-dimensional vectors. Using (20), we can then solve for D_3, after which, back-substituting this value of D_3 in $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{E}$, we can solve for υ using (22b), and x_0, y_0, p_xF, p_yF, Ω_1, Ω_2, Ω_3, D_1, and D_2 using (17a)–(21b). See [5] for a different interpretation of this situation.

Also of particular interest is the case in which the skew angle is known to be orthogonal, perhaps as the result of an initial phase that removed camera distortion. Here we take $\upsilon = 0$ so that the number of unknown variables is 16. One sees from (22b) that the degeneracy involves a zero dot product on the right side of the equation. In order to resolve this degeneracy, we let $\boldsymbol{P} = \{D_3, R_{33}\}$. Thus, $\boldsymbol{A} = \alpha D_3 + \alpha^* R_{33}$, $\boldsymbol{B} = \beta D_3 + \beta^* R_{33}$, $\boldsymbol{C} = \gamma D_3 + \gamma^* R_{33}$, $\boldsymbol{E} = \epsilon D_3 + \epsilon^* R_{33}$, and $G = 1 \cdot D_3 + 0 \cdot R_{33}$, where α, β, γ, ϵ, α^*, β^*, γ^*, and ϵ^* are numerically known three-dimensional vectors having $\gamma_3 = 0$ and $\gamma^*_3 = 1$. We then use (20) and (22b) to solve for D_3 and R_{33}, which involves solving a fourth order polynomial equation. This is followed by back-substituting, as before, to get our final results. Note that $D_3 > 0$, which narrows down the choices.

Taking the example one step further, suppose we also know that the aspect ratio $\sigma = P_y/P_x = 1$. This requires equality between (18a) and (18b), which ordinarily define P_x and P_y independently. We now put $\boldsymbol{P} = \{D_3, R_{33}, X_4\}$, and use this equality between (18a) and (18b), together with (20) and (22b), to solve for this subset of three coefficients followed by back-substitution as before. Actually, we could have used Y_4 instead of X_4 here. In practical applications, the particular geometry will dictate the preference. In this case, one starts out with (20) and solves for X_4 in terms of D_3 and R_{33}. When $X_4(D_3, R_{33})$ is substituted into the other equations, one obtains two eighth-order bivariate polynomials to solve for D_3 and R_{33}. One now sees a pattern of more constraints requiring more coefficients on the right side in order to maintain the independence of the unknowns in these equations.

In general, each additional constraint will introduce another eight-order multivariate polynomial equation to the set to be solved. The procedures are fairly straightforward for each new situation: first solve the basic linear equations and then solve a set of multivariate polynomial equations.

The above cases are solved utilizing known corresponding scene and image points such that not all the scene points lie on the same plane. If all the scene points are coplanar, the least-squares based technique previously discussed breaks down, as the matrix for which we need the inverse becomes singular. We can, however, proceed as follows. Assume, without loss of generality, that the plane in which all the scene points lie is $z_W = 0$. Equations (10a) and (10b) would then form a homogeneous system in nine linearly independent unknowns. The unknowns X_3, Y_3, and R_{33} would not appear. Thus, we have $k + 6$ equations, $k \le 8$, in 17 unknowns, which means that we must deliberately provide at least three additional constraints in order to reduce the number of variables to 14 or less.

As an example, suppose we know x_0, y_0, and υ. Taking $k = 8$, we let $\boldsymbol{P} = \{D_3\}$. Thus, $\boldsymbol{A} = \alpha D_3$, $\boldsymbol{B} = \beta D_3$, $\boldsymbol{C} = \gamma D_3$, $\boldsymbol{E} = \epsilon D_3$, and $G = 1 \cdot D_3$, where α_3, β_3, γ_3, as well as D_3 are to be determined. Using (17a), (17b), (20), and (22b), we have four equations for these four unknowns. When the skew angle υ is equal to 0, the computation is slightly different than when it is not equal to 0. In the former case, we must solve a quadratic equation for $(D_3)^2$.

Cases where other pairs of parameters are known are similarly solved. To solve cases where we know three parameters, we let $k = 7$ and $\boldsymbol{P} = \{D_3, R_{33}\}$.

See Table II for some resolved cases of the one-step method, where various combinations of the intrinsic parameters are known beforehand. Note that as more information is known, coplanar points allow calibration with less complexity than noncoplanar points.

V. The Two-Step Method

The one-step method solved (10a) and (10b) simultaneously. Now, we show how to solve them independently. This method will, in general, allow simpler solutions, but at the price of lower accuracy.

In solving (10a), we would have (14a), (14c), (15a), and (15c) to solve, while in solving (10b), we would have (14b), (14c), (15b), and (15c) to solve.

TABLE II
SOME RESOLVED CASES OF THE LINEAR CALIBRATION PROBLEM

Intrinsic Unknowns						Method
Scaling P_x, P_y	Center Displacement x_O, y_O	Aspect Ratio $\sigma = P_y/P_x$	Skew Angle υ	Number of Unknowns	Non-Coplanar or Coplanar	1-Step or 2-Step
×	×	×	×	17	N	1
×	×	×		16	N	1,2
×	×		×	16	N	1
	×		×	15	N	1
×		×	×	15	N	1
×	×			15	N	1,2
	×			14	N	1,2
×		×		14	N	1,2
×			×	14	N	1
			×	13	N	1
×				13	N	1,2
				12	N	1,2
	×			14	C	1,2
×		×		14	C	1,2
×			×	14	C	1
			×	13	C	1
×				13	C	1,2
				12	C	1,2

Letting $\boldsymbol{R} = \boldsymbol{A} \times \boldsymbol{C}$ and $\boldsymbol{S} = \boldsymbol{B} \times \boldsymbol{C}$, in the former case we would first have solutions to (15a) and (15c) consisting of

$$x_0 = -\boldsymbol{A} \cdot \boldsymbol{C}, \tag{23}$$

$$P_x = |\boldsymbol{A} - (\boldsymbol{A} \cdot \boldsymbol{C})\boldsymbol{C}| = |\boldsymbol{A} \times \boldsymbol{C}|, \tag{24}$$

$$\Omega_1 = \frac{\boldsymbol{M}}{|\boldsymbol{M}|}, \tag{25a}$$

$$\Omega_2 = -\frac{\boldsymbol{R}}{|\boldsymbol{R}|}, \tag{25b}$$

$$\Omega_3 = \boldsymbol{C}, \tag{25c}$$

while in the latter case, we would first have solutions to (15b) and (15c) consisting of

$$y_0 = -\boldsymbol{B} \cdot \boldsymbol{C}, \tag{26}$$

$$P_y = |\boldsymbol{B} - (\boldsymbol{B} \cdot \boldsymbol{C})\boldsymbol{C}| = |\boldsymbol{B} \times \boldsymbol{C}|. \tag{27}$$

$$\Omega_1 = \sin \upsilon \frac{\boldsymbol{N}}{|\boldsymbol{N}|} + \cos \upsilon \frac{\boldsymbol{S}}{|\boldsymbol{S}|}, \tag{28a}$$

$$\Omega_2 = \cos \upsilon \frac{\boldsymbol{N}}{|\boldsymbol{N}|} - \sin \upsilon \frac{\boldsymbol{S}}{|\boldsymbol{S}|}, \tag{28b}$$

$$\Omega_3 = \boldsymbol{C}. \tag{28c}$$

Assuming that (20) is satisfied, one can verify that both approaches produce solutions satisfying all constraint equations (16). Note, however, that each solution is based on only a single image coordinate x^* or y^*. Also note that we get two solutions for certain parameters.

We now illustrate this technique in the noncoplanar situation, when all we know is υ. Recall that in the one-step approach, we put $\boldsymbol{P} = \{D_3, R_{33}\}$. In this case we put $\boldsymbol{P} = \{D_3\}$, resulting in a simpler solution. Thus, $\boldsymbol{C} = \gamma D_3$. Solving (10a), we would use (20) to solve for D_3, then back-substitute this value in (23)–(25c) to get numerical values for x_0, P_x, Ω_1, Ω_2, and Ω_3. Note that in solving (10a), we would also have an equation in D_1 and D_2. Solving (10b), we would also use (20) to solve for D_3, then back-substitute this value in (26)–(28c) to get numerical values for y_0, P_y, Ω_1, Ω_2, and Ω_3. Note that in solving (10b), we would have yet another equation in D_1 and D_2. Using these two linear equations in D_1 and D_2, we can solve for these parameters individually. We thus have solved numerically for every unknown, with some unknowns, namely D_3, Ω_1, Ω_2, and Ω_3, having two possible values.

See Table II for some resolved cases of the two-step method, where various combinations of the intrinsic parameters are known beforehand.

VI. SOME EXPERIMENTS UTILIZING OUR TECHNIQUES

Using data supplied by the Systems Avionics Division of Wright-Patterson Air Force Base, and referring to Table II, we conducted experiments for the one-step method of Cases 1 and 2 and for the two-step method of Case 2. These are the cases where P_x, P_y, x_0, y_0, and σ are all unknown. The given data consisted of 20 points of $(x^*, y^*, x_W, y_W, z_W)$ values from a passive camera and 27 points of $(x^*, y^*, x_W, y_W, z_W)$ values from an active (ranging) camera for a standard calibration object. See Tables III(a) and (b) for a listing of this data.

To see how sensitive the various calibration techniques were to the quantity of points used, we conducted experiments using N points, $16 \le N \le 20$, of passive camera data.

For the one-step method of Case 1 of Table II, the results are exhibited in Tables IV(a)–(e). To determine the accuracy of the results, we used (10a) and (10b) to solve for x^* and y^*, respectively. Utilizing the given 20 passive calibration points, we then back-substituted the values of X_1, X_2, X_3, X_4, Y_1, Y_2, Y_3, Y_4, D_3, R_{13}, R_{23}, R_{33}, x_W, y_W, z_W into these equations to calculate the corresponding values for x^* and y^*, and compared them to the given values. We also utilized the active camera calibration along with the passive camera calibration and solved for the x_W, y_W, z_W values of the given 27 active calibration points, comparing them with their true values. This follows from the fact that by using a second camera, we have four linear equations for these three unknowns. In Tables IV(a) and (b) notice the relative stability of parameter values for the different values of N, except for the values of x_0 and y_0. These values seem to be particularly sensitive to the number of points chosen, and hence, to noise. As shown in Table IV note that each dot product of the form $\Omega_i \cdot \Omega_i$, for $1 \le i \le 3$, equals 1, while each dot product of the form $\Omega_i \cdot \Omega_j$, for $1 \le i \ne j \le 3$, is $O(10^{-17})$.

In Table IV(a) pay particular attention to the value of $\sin \upsilon$. This demonstrates that the image axes are not quite perpendicular. Since they are supposed to be perpendicular, we also implemented the one-step method of Case 2 of Table II and let $\upsilon = 0$. This technique is much more complicated than the previous one. Since $\upsilon = 0$, we had to solve a 4th degree equation for $(D_3)^2$. Even though $D_3 > 0$, we still had up to four values of D_3 to consider. We eliminated those values of D_3 which resulted in large errors from back-substituting and comparing the true values of (x^*, y^*) with the computed values of (x^*, y^*). These results are shown in Tables V(a)–(d). Note here that each value of N gives 2 sets of values for the intrinsic and extrinsic parameters. These two sets of values show up for each value of N. Also notice that the set of values exhibiting the smaller errors agrees with that found in Case 1. The errors in Case 2 are very compatible with those found in Case 1, sometimes better, sometimes worse. At the end of this section, we will present an error analysis of these and other cases which indicates that the Case 2 formulation is less sensitive to error than the Case 1 formulation. As shown in Table V(e), note that each dot product of the form $\Omega_1 \cdot \Omega_i$, for $1 \le i \le 3$, equals 1, while each dot product of the form $\Omega_i \cdot \Omega_j$, for $1 \le i \ne j \le 3$, is $O(10^{-16})$.

Similar results are exhibited in Tables VI(a)–(e) for the two-step method of Case 2 of Table II.

Lastly, we compared our techniques with those of [5]. As shown in Tables VII(a) and (b), the difference in values of the dot products $\Omega_1 \cdot \Omega_2$ and $\Omega_2 \cdot \Omega_3$ between his approach and our previous approaches are considerable, ranging from 1 to 15 orders of magnitude. His values for D_1, D_3, p_xF, p_yF, x_0, y_0, Ω_1, and Ω_3 are similar to ours. For D_2 and Ω_2, we have

$$\sin \upsilon\, \Omega_1^{(\text{OurCase 1})} + \cos \upsilon\, \Omega_2^{(\text{OurCase 1})} = \Omega_2^{(\text{Ganapathy})} \tag{29a}$$

TABLE III
(a) PASSIVE CAMERA DATA. (b) ACTIVE CAMERA DATA.

x^*	y^*	x_W	y_W	z_W
150.632	80.803	-2.500	0.500	3.000
97.458	145.261	-4.000	1.000	-3.000
149.290	208.382	-2.500	3.000	3.000
97.204	239.473	-4.000	3.000	-3.000
147.415	336.548	-2.500	5.500	3.000
95.390	334.558	-4.000	5.000	-3.000
252.988	337.916	0.000	5.500	3.000
93.701	428.709	-4.000	7.000	-3.000
358.680	340.171	2.500	5.500	3.000
171.781	430.308	-2.000	7.000	-3.000
359.739	210.775	2.500	3.000	3.000
249.319	430.779	0.000	7.000	-3.000
361.369	83.183	2.500	0.500	3.000
327.836	431.901	2.000	7.000	-3.000
255.629	81.922	0.000	0.500	3.000
406.361	433.175	4.000	7.000	-3.000
253.782	209.060	0.000	3.000	3.000
408.177	338.498	4.000	5.000	-3.000
409.838	243.217	4.000	3.000	-3.000
409.900	148.277	4.000	1.000	-3.000

(a)

x^*	y^*	x_W	y_W	z_W
146.14	79.99	-2.6069	0.4851	3.0
153.99	207.46	-2.3825	2.9751	3.0
147.27	344.90	-2.5010	5.6634	3.0
257.47	338.17	0.1046	5.4948	3.0
361.43	337.86	2.5646	5.4542	3.0
356.90	212.22	2.4261	3.0117	3.0
253.27	83.13	-0.0644	0.5228	3.0
258.52	206.52	0.0907	2.9220	3.0
100.36	152.74	-3.9232	1.1577	-3.0
96.48	240.81	-3.9927	3.0225	-3.0
93.32	336.84	-4.0409	5.0558	-3.0
94.25	420.71	-3.9886	6.8306	-3.0
168.59	428.64	-2.0841	6.9759	-3.0
253.66	427.11	0.0916	6.9183	-3.0
324.15	429.29	1.8956	6.9429	-3.0
401.58	429.04	3.8764	6.9144	-3.0
253.95	69.34	-0.0523	0.2531	3.0
184.24	175.82	-1.6743	2.3499	3.0
345.11	201.00	2.1435	2.7955	3.0
184.44	275.65	-1.6421	4.2972	3.0
321.22	328.46	-1.6106	5.2847	3.0
158.26	328.81	-2.2457	5.3452	3.0
95.65	249.92	-4.0107	3.2154	-3.0
101.32	410.34	-3.8112	6.6089	-3.0
271.16	391.07	0.5280	6.1541	-3.0
387.57	415.34	3.5135	6.6301	-3.0
254.46	403.96	0.1048	6.4305	-3.0

(b)

and

$$\sin \upsilon \, D_1^{(\text{Our Case 1})} + \cos \upsilon \, D_2^{(\text{Our Case 1})} = D_2^{(\text{Ganapathy})}. \quad (29b)$$

The (x^*, y^*) and (x_W, y_W, z_W) errors of his approach are otherwise identical to ours.

Let us now compare the tolerance of our techniques to noise. We use a result of [6] which states the following:

Theorem 6.1-3:

Suppose x, r, x', and r' satisfy

$$\|Ax - b\|_2 = \min,$$

$$r = b - Ax,$$

$$\|(A + \delta A)x' - (b + \delta b)\|_2 = \min,$$

$$r' = (b + \delta b) - (A + \delta A)x',$$

where A and δA are $m \times n$ real matrices with $m \geq n$, $b \neq 0$, and δb is an m element real vector. Assume that

$$\epsilon = \max\left\{\frac{\|\delta A\|_2}{\|A\|_2}, \frac{\|\delta b\|_2}{\|b\|_2}\right\} < \frac{\sigma_n(A)}{\sigma_1(A)},$$

and that

$$\sin \theta = \frac{\rho_{\text{least-squares}}}{\|b\|_2} \neq 1,$$

TABLE IV
(a) INTRINSIC PARAMETER VALUES FOR THE ONE-STEP METHOD OF CASE 1 OF TABLE II. (b) x^*, y^* ERRORS FOR THE N PASSIVE POINTS FOR THE ONE-STEP METHOD OF CASE 1 OF TABLE II. (c) x^*, y^* ERRORS FOR THE REMAINING $20 - N$ PASSIVE POINTS FOR THE ONE-STEP METHOD OF CASE 1 OF TABLE II. (d) x_W, y_W, z_W ERRORS FOR THE 27 ACTIVE POINTS FOR THE ONE-STEP METHOD OF CASE 1 OF TABLE II. (e) DOT PRODUCTS FOR THE ONE-STEP METHOD OF CASE 1 OF TABLE II.

N	P_x, P_y	x_O, y_O	sin υ
16	.299650E+4,.365984E+4	.120993E+3,.411295E+3	-.301645E-2
17	.298060E+4,.364273E+4	.115077E+3,.375328E+3	-.322721E-2
18	.306097E+4,.373413E+4	.726774E+2,.476317E+3	-.221632E-2
19	.313900E+4,.381954E+4	.415888E+2,.600368E+3	-.116770E-2
20	.311653E+4,.379532E+4	.504322E+2,.557981E+3	-.157175E-2

(a)

N	Maximum Absolute x^*,y^* Errors	Average Absolute x^*,y^* Errors
16	.643862,.684055	.197008,.215006
17	.623482,.749474	.221106,.231009
18	.698331,.799317	.265789,.255020
19	.709364,.872149	.333795,.272628
20	.867626,.854868	.327508,.251712

(b)

N	Maximum Absolute x^*,y^* Errors	Average Absolute x^*,y^* Errors
16	2.04595,.713251	1.28321,.377788
17	2.13823,.432688	1.60614,.272443
18	1.36062,.159048	.893255,.0950419
19	.466599,.126079	.466599,.126079

(c)

N	Maximum Absolute x_W,y_W,z_W Errors	Average Absolute x_W,y_W,z_W Errors
16	.0115942,.0157764,.0671509	.00498998,.00549650,.0282848
17	.0117823,.0176848,.0682295	.00576469,.00605661,.0281345
18	.0105744,.0159950,.0688043	.00508967,.00631082,.0280055
19	.0121288,.0131764,.0680794	.00558025,.00620280,.0282362
20	.0108416,.0147754,.0684651	.00541669,.00625791,.0280832

(d)

N	$\Omega_1 \bullet \Omega_1$,$\Omega_2 \bullet \Omega_2$,$\Omega_3 \bullet \Omega_3$	$\Omega_1 \bullet \Omega_2$,$\Omega_1 \bullet \Omega_3$,$\Omega_2 \bullet \Omega_3$
16	1,1,1	.542101E-19,.173472E-17,-.268882E-16
17	1,1,1	-.238524E-17,-.260209E-17,-.416334E-16
18	1,1,1	.119262E-17,-.867362E-18,-.693889E-17
19	1,1,1	.921572E-18,-.867362E-18,-.147451E-16
20	1,1,1	.379471E-18,.173472E-17,-.138778E-16

(e)

where $\sigma_1(A)$ and $\sigma_n(A)$ are the largest and the smallest singular values of A, respectively and $\rho_{\text{least-squares}}$ is the 2-norm of $Ax_{\text{least-squares}} - b$, the minimal 2-norm solution to the least-squares problem $Ax = b$.

TABLE V

(a) INTRINSIC PARAMETER VALUES FOR THE ONE-STEP METHOD OF CASE 2 OF TABLE II. (b) x^*, y^* ERRORS FOR THE N PASSIVE POINTS FOR THE ONE-STEP METHOD OF CASE 2, TABLE II. (c) x^*, y^* ERRORS FOR THE REMAINING $20 - N$ PASSIVE POINTS FOR THE ONE-STEP METHOD OF CASE 2 OF TABLE II. (d) x_W, y_W, z_W ERRORS FOR THE 27 ACTIVE POINTS FOR THE ONE-STEP METHOD OF CASE 2 OF TABLE II. (e) DOT PRODUCTS FOR THE ONE-STEP METHOD OF CASE 2 OF TABLE II. (NOTE: THE FIRST ENTRY FOR EACH VALUE OF N IS THE RESULT OF SOLVING (10a), WHILE THE SECOND ENTRY IS THE RESULT OF SOLVING (10b).)

N	P_x,P_y	x_O,y_O
16	.325259E+4,.396128E+4 .253521E+4,.309089E+4	.241776E+2,.664282E+3 .249703E+3,.776877E+2
17	.326647E+4,.398283E+4 .252207E+4,.307428E+4	.572971E+1,.643679E+3 .243891E+3,.640633E+2
18	.322213E+4,.392575E+4 .242598E+4,.294485E+4	.158851E+2,.628306E+3 .238120E+3,.469103E+2
19	.320801E+4,.390072E+4 .203676E+4,.244644E+4	.213323E+2,.663663E+3 .260398E+3,-.352754E+2
20	.324626E+4,.394914E+4 .737271E+3,.856160E+3	.326571E+2,.649784E+3 .239747E+3,.679661E+3

(a)

N	Maximum Absolute x^*,y^* Errors	Average Absolute x^*,y^* Errors
16	.816316,.980756 1.38975,1.03441	.145054,.308651 .507534,.368891
17	.813973,1.11053 1.36916,1.07960	.272702,.351476 .533907,.360294
18	.767269,1.01471 2.12539,1.49851	.269617,.331737 .722981,.449278
19	.716573,.971215 3.89493,1.83682	.314379,.305725 1.55232,1.07120
20	.789722,1.07578 30.7386,30.4420	.312550,.321217 12.2398,12.2825

(b)

N	Maximum Absolute x^*,y^* Errors	Average Absolute x^*,y^* Errors
16	.953593,.918077 4.35140,2.47148	.559621,.538866 3.04063,.987077
17	.935005,.766244 4.46457,2.46652	.546598,.521107 3.88574,1.22159
18	.852048,.456067 3.86486,3.08734	.574271,.319661 3.82977,2.07200
19	.678259,.337824 4.28133,4.62378	.678259,.337824 4.28133,4.62378

(c)

N	Maximum Absolute x_W,y_W,z_W Errors	Average Absolute x_W,y_W,z_W Errors
16	.0162692,.0172558,.0628429 .0333510,.0248235,.0804603	.00590280,.00637397,.0296902 .0132804,.00804624,.0283624
17	.0179098,.0178886,.0624437 .0328384,.0255327,.0826289	.00637518,.00722737,.0296873 .0139315,.00827182,.0281803
18	.0147181,.0162975,.0637086 .0470818,.0275160,.0933348	.00561600,.00706280,.0291384 .0161417,.00928292,.0315290
19	.0144712,.0154536,.0657424 .0943310,.0482593,.121068	.00551838,.00671034,.0287691 .0306460,.0209127,.0498809
20	.0157361,.0173190,.0638308 1.05930,.770761,1.20717	.00532212,.00727314,.0291270 .322620,.251545,.381684

(d)

N	$\Omega_1 \bullet \Omega_1$,$\Omega_2 \bullet \Omega_2$,$\Omega_3 \bullet \Omega_3$	$\Omega_1 \bullet \Omega_2$,$\Omega_1 \bullet \Omega_3$,$\Omega_2 \bullet \Omega_3$
16	1,1,1 1,1,1	.121702E-15,0,-.199493E-16 .113950E-15,-.433681E-17,0
17	1,1,1 1,1,1	.101288E-15,0,-.335561E-16 .933498E-16,-.780626E-17,.277556E-16
18	1,1,1 1,1,1	.121864E-15,.138778E-16,-.173472E-16 .973071E-16,-.346945E-17,.277556E-16
19	1,1,1 1,1,1	-.352366E-17,0,-.806646E-16 .715573E-17,.320924E-16,.277556E-16
20	1,1,1 1,1,1	-.902259E-17,.867362E-18,.290566E-16 -.693889E-17,.182416E-16,-.111022E-15

(e)

Then,

$$\frac{\|x' - x\|_2}{\|x\|_2} \leq \epsilon \left\{ \frac{2\kappa_2(A)}{\cos\theta} + \tan\theta\, \kappa_2(A)^2 \right\} + O(\epsilon^2), \quad (30)$$

and

$$\frac{\|r' - r\|_2}{\|b\|_2} \leq \epsilon \left(1 + 2\kappa_2(A)\right) \min(1, m - n) + O(\epsilon^2), \quad (31)$$

where $\kappa_2(A) = \sigma_1(A)/\sigma_n(A)$, for $\sigma_1(A)$ as above and $\sigma_n(A)$ the nth largest singular value of A, where $n = \text{rank}(A)$.

Note that (30) bounds the so-called *least-squares error*, while (31) bounds the so-called *residual error*.

Assuming that x^* and y^* are accurate to within $1/2$ of a pixel, our calculations show that the bounds on the residual errors from both the one-step methods of Case 1 and Case 2 of Table II are of the same order of magnitude, although the latter upper bound is smaller. However, for the least-square error bounds, the one from Case 2 is of a lower order of magnitude than the one from Case 1. Note that Case 1 covers the methodology used by the Numerical Stereo Camera Group and by Ganapathy. For the two-step method of Case 2 of Table II, the error bounds resulting from solving (10a) give comparable results to the one-step method of Case 2, while the error bounds resulting from solving (10b) give comparable results to the one-step method of Case 1. See Table VIII.

VII. CONCLUSIONS

We have developed mathematically elegant solutions for the general linear (affine) camera calibration problem which are relatively easy to compute. The procedures presented above are inherently similar for both the coplanar and noncoplanar cases and satisfy all necessary constraints.

In [10], all rotation constraints are satisfied for his coplanar solution, but one constraint, $\Omega_1 \cdot \Omega_2 = 0$, is not explicitly satisfied

TABLE VI

(a) INTRINSIC PARAMETER VALUES FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (b) x^*, y^* ERRORS FOR THE N PASSIVE POINTS FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (c) x^*, y^* ERRORS FOR THE REMAINING $20 - N$ PASSIVE POINTS FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (d) x_W, y_W, z_W ERRORS FOR THE 27 ACTIVE POINTS FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (e) DOT PRODUCTS FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (NOTE: THE FIRST ENTRY FOR EACH VALUE OF N IS THE RESULT OF SOLVING (10a), WHILE THE SECOND ENTRY IS THE RESULT OF SOLVING (10b).)

N	P_x, P_y	x_O, y_O
16	.302992E+4,.360846E+4	-.233419E+2,.417409E+3
17	.303090E+4,.354931E+4	-.577266E+2,.315235E+3
18	.310153E+4,.356448E+4	-.148480E+3,.372180E+3
19	.317242E+4,.358690E+4	-.204724E+3,.357617E+3
20	.313781E+4,.367812E+4	-.193214E+3,.438119E+3

(a)

N	Maximum Absolute x^*, y^* Errors	Average Absolute x^*, y^* Errors
16	.622056,.637643	.180635,.206637
17	.614327,.683898	.194378,.230014
18	.630910,.658218	.228660,.241280
19	.656891,.681818	.270772,.237603
20	.730650,.737427	.271578,.242142

(b)

N	Maximum Absolute x^*, y^* Errors	Average Absolute x^*, y^* Errors
16	1.75585,.804689	1.03413,.528106
17	1.69476,.881589	1.16251,.500922
18	1.05516,1.06503	.573862,.741914
19	.704355,.776075	.704355,.776075

(c)

N	Maximum Absolute x_W, y_W, z_W Errors	Average Absolute x_W, y_W, z_W Errors
16	.0144790,.0152544,.0694297	.00627972,.00509687,.0283752
17	.0157207,.0182907,.0701896	.00706684,.00611636,.0285385
18	.0143352,.0171194,.0664368	.00761140,.00582456,.0285222
19	.0148205,.0172846,.0652476	.00816411,.00592737,.0289918
20	.0144452,.0159917,.0651333	.00804846,.00580071,.0287193

(d)

N	$\Omega_1 \cdot \Omega_1, \Omega_2 \cdot \Omega_2, \Omega_3 \cdot \Omega_3$	$\Omega_1 \cdot \Omega_2, \Omega_1 \cdot \Omega_3, \Omega_2 \cdot \Omega_3$
16	1,1,1 1,1,1	-.867362E-18,0,0 .704731E-18,-.867362E-18,-.260209E-17
17	1,1,1 1,1,1	-.867362E-18,.416334E-16,.867362E-18 -.853152E-18,.867362E-18,.138778E-16
18	1,1,1 1,1,1	-.102999E-17,.277556E-16,0 .433681E-18,.173472E-17,-.277556E-16
19	1,1,1 1,1,1	-.596311E-18,.138778E-16,0 .867362E-18,0,-.416334E-16
20	1,1,1 1,1,1	-.140946E-17,.138778E-18,0 .379471E-18,0,-.225514E-16

(e)

TABLE VII

(a) INTRINSIC PARAMETER VALUES FOR GANAPATHY'S METHOD. (b) DOT PRODUCTS FOR GANAPATHY'S METHOD.

N	P_x, P_y	x_O, y_O
16	.299650E+4,.365984E+4	.120994E+3,.411295E+3
17	.298060E+4,.364273E+4	.115077E+3,.375328E+3
18	.306097E+4,.373413E+4	.726774E+2,.476317E+3
19	.313900E+4,.381954E+4	.415888E+2,.600368E+3
20	.311653E+4,.379532E+4	.504322E+2,.557980E+3

(a)

N	$\Omega_1 \cdot \Omega_1, \Omega_2 \cdot \Omega_2, \Omega_3 \cdot \Omega_3$	$\Omega_1 \cdot \Omega_2, \Omega_1 \cdot \Omega_3, \Omega_2 \cdot \Omega_3$
16	1,1,1	-.301645E-2,-.372098E-15,-.213371E-15
17	1,1,1	-.322721E-2,.630572E-15,.208167E-15
18	1,1,1	-.221632E-2,.700828E-15,-.372966E-16
19	1,1,1	-.116770E-2,.241127E-14,.876035E-16
20	1,1,1	-.157175E-2,-.162717E-14,-.119696E-15

(b)

in his noncoplanar solution. How well the constraint equation is satisfied by the solution is interpreted as a goodness measure of the results [5], [10]. Experiments have been performed which show the efficacy of our methodology.

In our approach, one can assume that known constraints are unknown and solve the less degenerate problem. We can use the differences with the known constraints as a similar measure of goodness. This is not possible for the nondegenerate case; however, one can easily solve for the maximum number of unknowns. The simplicity of the formalism will facilitate future error and sensitivity

TABLE VIII
(a) ERROR BOUNDS IN THE LEAST-SQUARES APPROXIMATION FOR THE ONE-STEP METHOD OF CASE 1 OF TABLE II. (b) ERROR BOUNDS IN THE LEAST-SQUARES APPROXIMATION FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (c) ERROR BOUNDS IN THE LEAST-SQUARES APPROXIMATION FOR THE TWO-STEP METHOD OF CASE 2 OF TABLE II. (NOTE: THE FIRST ENTRY FOR EACH VALUE OF N IS THE RESULT OF SOLVING (10a), WHILE THE SECOND ENTRY IS THE RESULT OF SOLVING (10b).)

N	Least-Squares Error Bound	Residual Error Bound
16	.357225 + O(1E-9)	.0657278 + O(1E-9)
17	.389803 + O(1E-9)	.0659954 + O(1E-9)
18	.395339 + O(1E-9)	.0628752 + O(1E-9)
19	.335127 + O(1E-9)	.0552581 + O(1E-9)
20	.227335 + O(1E-9)	.0449261 + O(1E-9)

(a)

N	Least-Squares Error Bounds	Residual Error Bounds
16	.152639 + O(1E-9)	.0485381 + O(1E-9)
17	.156002 + O(1E-9)	.0497836 + O(1E-9)
18	.150126 + O(1E-9)	.0492166 + O(1E-9)
19	.135654 + O(1E-9)	.0468126 + O(1E-9)
20	.110831 + O(1E-9)	.0418054 + O(1E-9)

(b)

N	Least-Squares Error Bounds	Residual Error Bounds
16	.143571 + O(1E-9) .352650 + O(1E-9)	.0411035 + O(1E-9) .0663711 + O(1E-9)
17	.145215 + O(1E-9) .371023 + O(1E-9)	.0405117 + O(1E-9) .0648716 + O(1E-9)
18	.170505 + O(1E-9) .383174 + O(1E-9)	.0420885 + O(1E-9) .0657937 + O(1E-9)
19	.173474 + O(1E-9) .373942 + O(1E-9)	.0402378 + O(1E-9) .0648470 + O(1E-9)
20	.156673 + O(1E-9) .257355 + O(1E-9)	.0374909 + O(1E-9) .0517886 + O(1E-9)

(c)

analysis. We see potential use for these techniques in interpreting calibration parameters and performing recalibrations with certain parameters changing in known ways.

REFERENCES

[1] J. T. Dijak, "Precise three-dimensional calibration of numerical stereo camera systems for fixed and rotatable scenes," AFWAL Tech. Rep. TR-84-1105, 1984.

[2] J. T. Dijak, "A method for correcting geometric distortion in video cameras," in *Proc. 1985 IEEE Nat. Aerospace and Electronics Conf.*, Dayton, OH, May 1985, pp. 1382-1388.

[3] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[4] W. Faig, "Calibration of close-range photogrammetry systems: Mathematical formulation," *Photogrammetric Eng. Remote Sensing*, vol. 41, pp. 1479-1486, 1975.

[5] S. Ganapathy, "Decomposition of transformation matrices for robot vision," in *Proc. IEEE Int. Conf. Robotics and Automation*, Atlanta, GA, Mar. 1984, pp. 130-139.

[6] G. H. Golub, Matrix Computations. Baltimore, MD: Johns Hopkins University Press, 1983.

[7] J. L. Posdamer and M. D. Altschuler, "Surface measurement by space-encoded projected beam systems," *Comput. Graphics Image Processing*, vol. 18, pp. 1-17, 1982.

[8] I. Sobel, "On calibrating computer controlled cameras for perceiving 3-D scenes," *Artificial Intell.*, vol. 5, pp. 185-198, 1974.

[9] T. M. Strat, "Recovering the camera parameters from a transformation matrix," in *Proc. Image Understanding Workshop*, New Orleans, LA, Oct. 1984, pp. 254-271.

[10] R. Tsai, "An efficient and accurate camera calibration technique for 3-D machine vision," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, pp. 364-374.

Splitting-Shooting Methods for Nonlinear Transformations of Digitized Patterns

Z. C. LI, T. D. BUI, C. Y. SUEN, AND Y. Y. TANG

Abstract—**In this correspondence, new splitting-shooting methods are presented for nonlinear transformations T: $(\xi, \eta) \rightarrow (x, y)$ where $x = x(\xi, \eta)$, $y = y(\xi, \eta)$. These transformations are important in computer vision, image processing, pattern recognition, and shape transformations in computer graphics. Our methods can eliminate superfluous holes or blanks thus leading to better images while requiring only modest computer storage and CPU time. The implementation of the proposed algorithms is simple and straightforward. Moreover, these methods can be extended to images with gray levels, to color images, and to three dimensions. They can also be implemented on parallel computers or VLSI circuits.**

Theoretical analysis is presented to prove the convergence of the algorithms and to provide error bounds for the resulting images. The complexity of the algorithms is linear. Graphical and numerical experiments are given to verify the analytical results and to demonstrate the effectiveness of our methods.

Index Terms—**Image smoothing, nonlinear transformations, shape transformations, splitting-shooting method.**

I. INTRODUCTION

In [4], we provide variant shape transformations

$$T:(\xi, \eta) \rightarrow (x, y),$$
$$x = x(\xi, \eta), y = y(\xi, \eta), \tag{1.1}$$

for images and patterns. All those transformation models are *continuous*, but the image pixels are *discrete*. Here the question is: how can we properly apply the continuous transformations (1.1) to image transformations? One trouble is that some superfluous holes and blanks occur in the transformed images even for simple cases such as dilations and rotations. The innovative techniques in Lee *et al.* [3] are presented to handle this problem, but they are, unfortunately, confined to linear transformations only. In this correspondence we study nonlinear techniques defined by (1.1). Our ap-

Manuscript received May 23, 1988; revised December 13, 1989. Recommended for acceptance by O. D. Faugeras. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada, by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche of Quebec, and by the Ministère de l'Enseignement Supérieur et de la Science (Action Structurante).

The authors are with the Centre de Recherche Informatique de Montreal and the Department of Computer Science, Centre for Pattern Recognition and Machine Intelligence, Concordia University, Montreal, P.Q. H3G 1M8, Canada.

IEEE Log Number 9034374.