NASA CR 192005

IN-61-CR
142870
P.17

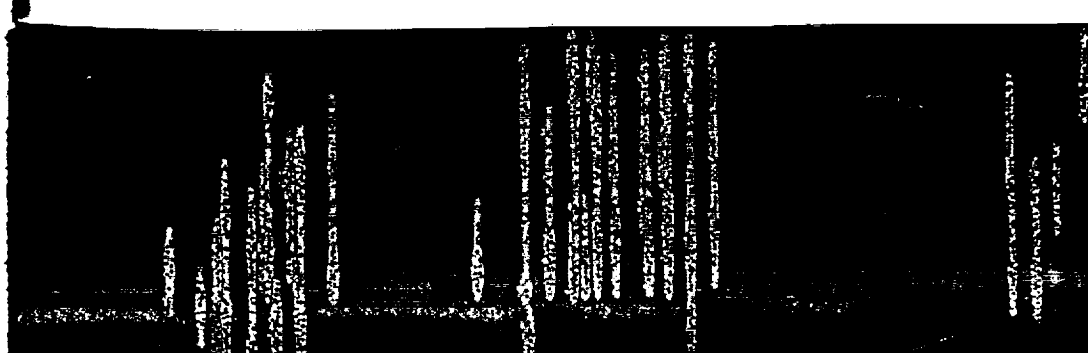
COMPUTER VISION and IMAGE PROCESSING

(NASA-CR-192005) FORMULATION AND ERROR ANALYSIS FOR A GENERALIZED IMAGE POINT CORRESPONDENCE ALGORITHM (Washington Univ.) 17 p
N93-18570
Unclas
G3/61 0142870

Edited by
Linda Shapiro
Department of Computer Science
University of Washington
Seattle, Washington

Azriel Rosenfeld
Center for Automation Research
University of Maryland
College Park, Maryland

ACADEMIC PRESS, INC.
Harcourt Brace Jovanovich, Publishers
Boston San Diego New York
London Sydney Tokyo Toronto



This book is printed on acid-free paper. ♾

ACADEMIC PRESS, INC.
1250 Sixth Avenue, San Diego, CA 92101-4311

United Kingdom Edition published by
ACADEMIC PRESS LIMITED
24-28 Oval Road, London NW1 7DX

Library of Congress Cataloging-in-Publication Data:

Computer vision and image processing / [edited by] Linda Shapiro, Azriel Rosenfeld.
p. cm.
Includes bibliographical references and index.
ISBN 0-12-638660-9 (alk. paper)
1. Computer vision. 2. Image processing—Digital techniques.
I. Shapiro, Linda G. II. Rosenfeld, Azriel, date.
TA1632.C6592 1992 91-29657
621.39'9—dc20 CIP

PRINTED IN THE UNITED STATES OF AMERICA

92 93 94 95 9 8 7 6 5 4 3 2 1

*FORMULATION AND ERROR ANALYSIS FOR A GENERALIZED IMAGE POINT CORRESPONDENCE ALGORITHM**

SUNIL FOTEDAR

Advanced Automation Technology Group, McDonnell Douglas Corporation, Space Station Division, Houston, Texas 77062

RUI J. P. DEFIGUEIREDO

Department of Electrical and Computer Engineering, University of California, Irvine, California 92717

and

KUMAR KRISHEN

Johnson Space Center, NASA, Houston, Texas 77058

A *Generalized Image Point Correspondence* (GIPC) algorithm, which enables the determination of 3-D motion parameters of an object in a configuration where both the object and the camera are moving, is discussed. A detailed error analysis of this algorithm has been carried out. Furthermore, the algorithm was tested on both simulated and video-acquired data, and its accuracy was determined.

I. Introduction

Motion analysis, based on robotic vision, has widely been discussed in the literature [1-5] in developing the *Image Point Correspondence* (IPC) algorithm. Methods used are *Two-view motion analysis* or *monocular vision*, *stereo* or *binocular vision*, and *stereo motion*. However, the IPC algorithm has not been applied to the more general problem of motion analysis involving a situation where both the object and the camera are moving [4]. Industrial

*This research was supported by NASA Contract NAS 9-17145 and NASA/JSC (RICIS) Grant NCC 9-16. Thanks are due to Mr. Olin Graham. Special thanks to Mrs. Lovely K. Fotedar for technical assistance.

ISBN 0-12-638660-9

and space robots face this situation in locating and tracking various objects/scenes when both the camera/video system and the object move asynchronously. In the GIPC, a generalization of the IPC algorithm, this problem of motion analysis is discussed. The three methods of motion analysis mentioned before become special cases of the GIPC presented in this paper.

The accuracy of the IPC/GIPC algorithms depends on the availability and errors in the measured input parameters. Input errors also include the deformation of the object over time. The error propagation for various stages of the IPC will, therefore, be described in terms of the error bounds.

II. The Generalized Image Point Correspondence Algorithm

2.1. The Algorithm

The general case of motion analysis is illustrated in Fig. 1. For simplicity in presentation, the equations that track a single point P on a moving object viewed by a moving camera are considered. F_i and F_j are the two frames with which the camera coordinate system coincides at two different instants of time t_j and t_j $(t_j > t_i)$, respectively. Point P moves from one position P_i to another position P_j due to the *rigid-body* motion of the object. We assume $(R_i, \mathbf{T}_i)$ and $(R_j, \mathbf{T}_j)$ to be the transformation parameters (rotation and translation) that link the frames F_i and F_j respectively with the standard frame S. Also, let $(R_{ij}, \mathbf{T}_{ij})$ be the transformation parameters that link the frame F_i with the frame F_j. The object moves with the unknown motion parameters $(R, \mathbf{T})$. The image plane is assumed to be at the focal point of the camera with its X- and Y-axes parallel to those of the camera coordinate system, where Z-axis is the line of sight.

The desired relationship between the coordinates of the initial and the final positions of point P (P_i and P_j, respectively) recorded by the camera, with respect to the frames F_i and F_j, respectively, is given by the equation [4]

$$\mathbf{p}_{jj} = R'_{ij}\, \mathbf{p}_{ii} + \mathbf{T}'_{ij}, \tag{2.1a}$$

where $\mathbf{p}_{\alpha\beta} = (x_{\alpha\beta}, y_{\alpha\beta}, z_{\alpha\beta})^{\mathrm{T}}$ is the vector of 3-D coordinates of P_β relative to F_α at instant t_β $(\alpha, \beta = i, j)$, and

$$R'_{ij} = R_{ij}^{\mathrm{T}} R_i R R_i^{\mathrm{T}} = R_j R R_i^{\mathrm{T}} \quad \text{and} \quad \mathbf{T}'_{ij} = -R_j R R_i^{\mathrm{T}} \mathbf{T}_i + R_j \mathbf{T} + \mathbf{T}_j. \tag{2.1b,c}$$

Equation (2.1a) gives the expression for the *generalized version of the motion-analysis equation*. Clearly, it does not matter whether the object or the

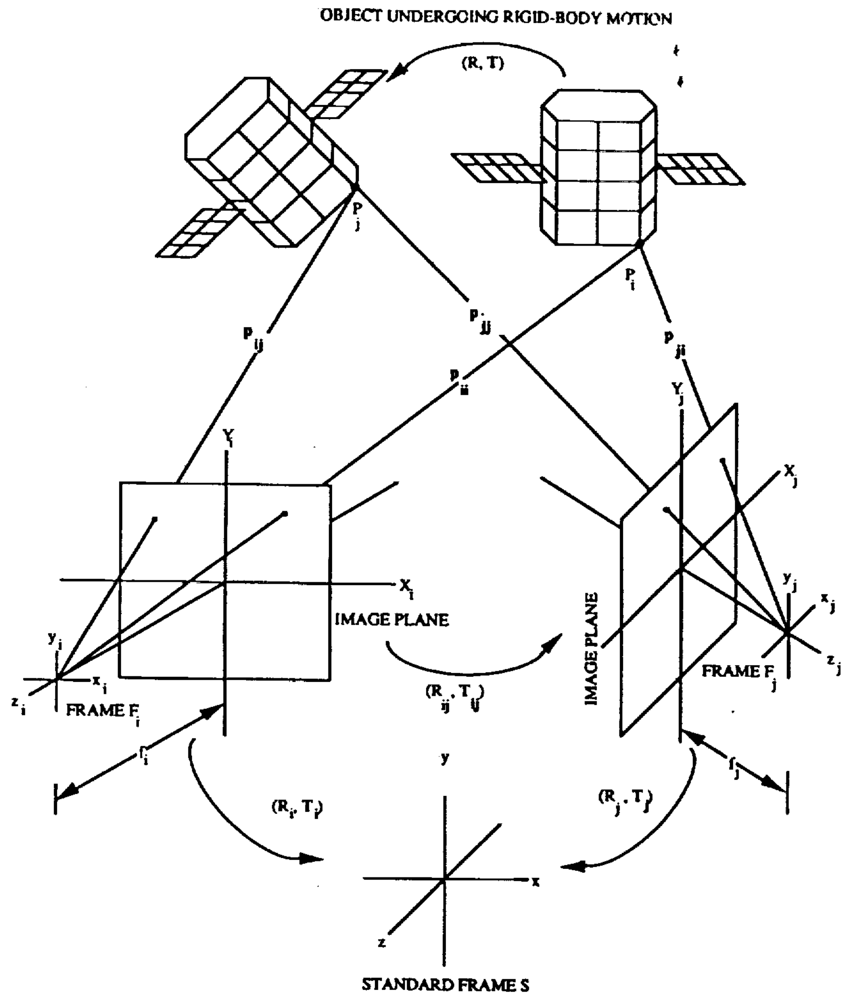


FIG. 1. Geometry illustrating GIPC algorithm.

camera is moved first. R and $\mathbf{T}$, the desired parameters to be estimated, are defined as

$$R \equiv \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad \text{and} \quad \mathbf{T} \equiv \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix}, \tag{2.1d,e}$$

where $r_{\alpha\beta}$ $(\alpha, \beta = 1, 2, 3)$ are the rotational elements and t_α $(\alpha = 1, 2, 3)$ the translations along x-, y-, and z-axes, respectively.

Special Cases. If F_i is standard frame S, the motion equation can be written as

$$R'_{ij} = R^{\mathrm{T}}_{ij} R = R_j R \text{ and } \mathbf{T}'_{ij} = R_j \mathbf{T} + \mathbf{T}_j \qquad (2.2\mathrm{a,b})$$

The IPC algorithm can be used to estimate the motion parameters $(R'_{ij}, \mathbf{T}'_{ij})$ and hence $(R, \mathbf{T})$ of the moving object, assuming R_{ij} and $\mathbf{T}_{ij}$ *are known.*

2.1.1. Monocular Vision

In the case of the two-view motion-analysis equation, the location of the camera taking the pictures of the moving object, is fixed. In that case,

$$R_i = R_j = R_{ij} = I \text{ and } \mathbf{T}_{ij} = \mathbf{T}_j = \mathbf{O}. \qquad (2.2\mathrm{c,d})$$

The generalized motion equation, using Eqs. (2.2a,b) and (2.2c,d), reduces to

$$\mathbf{p}_{jj} = R\mathbf{p}_{ii} + \mathbf{T}$$

or to the more familiar two-view motion equation

$$\mathbf{p}' = R\mathbf{p} + \mathbf{T}$$

as the frames F_i and F_j coincide with the frame S, such that $\mathbf{p}_{ii} = \mathbf{p}_i = \mathbf{p}$ and $\mathbf{p}_{jj} = \mathbf{p}_j = \mathbf{p}'$, where $\mathbf{p}_i = (x_i, y_i, z_i)^{\mathrm{T}}$ and $\mathbf{p}_j = (x_j, y_j, z_j)^{\mathrm{T}}$ are 3-D coordinates of P_i and P_j relative to S, respectively.

2.1.2. Stereo Vision/Stereo Motion

For a stereo vision/stereo motion case, the object is assumed to be stationary. In that case,

$$R = I;\ R_{ij} = R_j \text{ and } \mathbf{T} = \mathbf{O} \qquad (2.2\mathrm{e,f})$$

and the generalized motion-analysis equation, using Eqs. (2.2a,b) and (2.2e,f), reduces to

$$\mathbf{p}' = R_j\mathbf{p} + \mathbf{T}_j.$$

These cases of motion analysis have been found to be equivalent [1]. The motion of point P_i to point P_j with respect to a fixed frame F_j is similar to the motion of frame F_j to frame F_i with respect to a fixed point P.

III. Error Analysis

Since the IPC/GIPC algorithms can be implemented using a sequence of object images, any error in the input data and sensor parameters becomes a

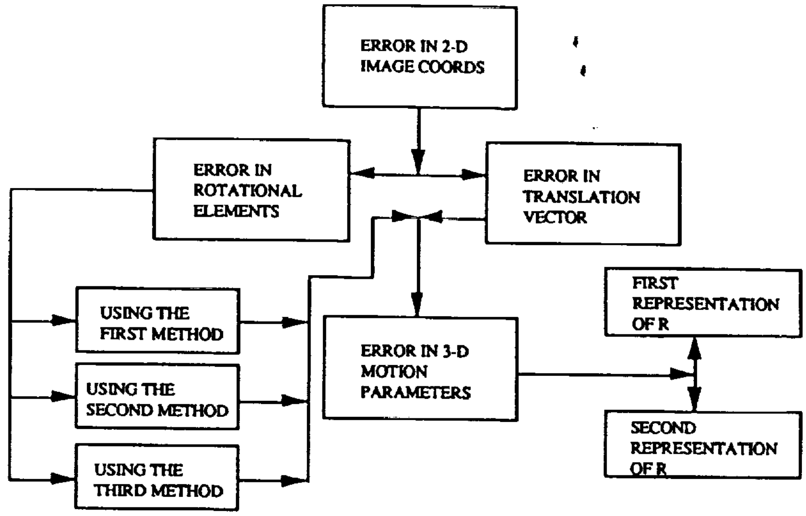


FIG. 2. Error analysis for various stages of the IPC algorithm.

source of inaccuracy in the output data. The input data are the set of feature coordinates of the object, and the output data are the 3-D motion parameters. The sources of perturbation errors have been identified in [6, 7].

In this section, we analyze the effect of changes in the input parameters on the output parameters for the *three* different methods for the IPC/GIPC algorithm [1-4]. The *two representations* of the rotation matrix will also be considered [2, 4, 8]. Since various steps are involved in this algorithm, the propagation of the error will be studied and the error bounds for each stage of the algorithm (Fig. 2) [7-9].

3.1. Error in Essential Elements

The sensitivity of the *essential elements* (elements of a matrix Q) to the error in input set of 2-D image coordinates of the features, common to the three methods, will be studied in this section. By definition, matrix Q is represented in terms of rotational and translational elements as [2, 4]

$$Q \equiv \begin{bmatrix} q_{11} & q_{12} & q_{13} \\ q_{21} & q_{22} & q_{23} \\ q_{31} & q_{32} & q_{33} \end{bmatrix} = \pm \begin{bmatrix} t_3r_{21} - t_2r_{31} & t_3r_{21} - t_2r_{23} & t_3r_{21} - t_2r_{33} \\ t_1r_{31} - t_2r_{11} & t_1r_{32} - t_3r_{12} & t_1r_{33} - t_3r_{13} \\ t_2r_{11} - t_1r_{21} & t_2r_{12} - t_1r_{22} & t_2r_{13} - t_1r_{23} \end{bmatrix} \tag{3.1a}$$

In our case, the equation solved is

$$\tilde{N}\tilde{Q} = (-1, -1, \ldots, -1)^{\dagger}$$

and the true equation to be solved is

$$NQ = (-, -1, \ldots, -1)^{\mathrm{T}},$$

where $\tilde{N} = N + \delta N$; $\tilde{Q} = Q + \delta Q$;

$$N = (N_1, N_2, N_3, \ldots,)^{\mathrm{T}};$$

$$N_\alpha = (X_\alpha X'_\alpha, Y_\alpha, X'_\alpha, X'_\alpha, X_\alpha, Y'_\alpha, Y_\alpha, Y'_\alpha, Y'_\alpha, X_\alpha, Y_\alpha) \qquad (\alpha = 1, 2 \ldots, n; n \geq 8);$$

and each element of Q is divided by q_{33}. Let

$$\delta N_\alpha = (\delta a_{\alpha 1}, \delta a_{\alpha 2}, \delta a_{\alpha 3}, \ldots, \delta a_{\alpha 8}),$$

where

$$\delta a_{\alpha 1} = \delta X'_\alpha X_\alpha + \delta X_\alpha X'_\alpha; \qquad \delta a_{\alpha 2} = \delta X'_\alpha Y_\alpha + \delta Y_\alpha X'_\alpha; \qquad \delta a_{\alpha 3} = \delta X'_\alpha;$$

$$\delta a_{\alpha 4} = \delta Y'_\alpha X_\alpha + \delta X_{\alpha 1} Y'_\alpha; \qquad \delta a_{\alpha 5} = \delta Y'_\alpha Y_\alpha + \delta Y_\alpha Y'_\alpha; \qquad \delta a_{\alpha 6} = \delta Y'_\alpha;$$

$$\delta a_{\alpha 7} = \delta X_\alpha \quad \text{and} \quad \delta a_{\alpha 8} = \delta Y_\alpha.$$

The products of errors are neglected and $(X_{\alpha, Y\alpha})$ are 2-D image coordinates for the αth data point. It follows from [9] that

$$N\,\delta Q = -\,\delta N\,Q = -\begin{bmatrix} \eta_1 \\ * \\ * \\ \eta_n \end{bmatrix} = -\begin{bmatrix} \delta a_{11} & * & * & \delta a_{18} \\ * & * & * & * \\ * & * & * & * \\ \delta a_{n1} & * & * & \delta a_{n8} \end{bmatrix} \begin{bmatrix} q_{11} \\ * \\ * \\ q_{32} \end{bmatrix}. \tag{3.1b}$$

Thus, if the inherent errors $\delta a_{\alpha\beta}$ were known, the corresponding solution errors $\delta q_{\alpha\beta}$ would be obtained by solving Eq. (3.1b). The degree of accuracy is consistent with the assumption of neglecting product of errors. Based on dimensions of N, we have the following cases:

Case I: If N is square (i.e., $n = 8$) and non-singular,

$$|\eta_\alpha| \leq E_\alpha \qquad (\alpha = 1, 2, \ldots, 8), \tag{3.1c}$$

where

$$E_\alpha = \varepsilon_\alpha \sum_{\alpha=1}^{3} \sum_{\beta=1}^{3} |q_{\alpha\beta}| \quad (\alpha, \beta = 1, 2, 3; \alpha \text{ and } \beta \neq 3 \Rightarrow q_{33} \text{ not included}) \tag{3.1d}$$

for $-\varepsilon_\alpha \leq \delta a_{\alpha\beta} \leq \varepsilon_\alpha$ $(\alpha, \beta = 1, 2, \ldots, 8)$. Therefore,

$$|\delta q_{\alpha\beta}| \leq E_\alpha \sum_{\gamma=1}^{8} |\bar{A}_{3(\alpha-1)+\beta,\gamma}| \qquad (\text{for } \alpha, \beta = 1, 2, 3; \alpha \text{ and } \beta \neq 3), \tag{3.1e}$$

where $\tilde{A}_{\alpha\beta}$ ($\alpha, \beta = 1, 2, \ldots, 8$) are the elements of the βth row of $N - 1$. Thus, if the elements of N^{-1} are calculated, approximate upper bounds on the effects of inherent errors can be obtained from Eq. (3.1e). Since they were derived under the assumption that the products of errors are negligible relative to E_α, they are not *strictly* upper bounds. However, they are acceptable as close approximations to the true upper bounds.

Case II. If N is not a square matrix, we define a residual matrix by

$$\delta N^+ = \tilde{N}^+ - N^+,$$

where N^+ and $\tilde{N}^+$ are the pseudo-inverses of N and $\tilde{N}$ respectively. Then from [10]

$$\|\delta N^+\| \leq \|\delta N_1^+\| + \|\delta N_2^+\| + \|\delta N_3^+\|, \tag{3.1f}$$

where

$$\|\delta N_1^+\| \leq \|\delta N\| \cdot \|N^+\| \cdot \|\tilde{N}^+\|, \ \|\delta N_2^+\| \leq \|\delta N\| \cdot \|\tilde{N}^+\|, \quad \text{and}$$
$$\|\delta N_3^+\| \leq \|\delta N\| N^+\|^2.$$

3.2. Error in Rotational Elements

The matrix Q is found in all the three methods in the same manner. But the rotation matrix R is computed from Q in three different ways. In this section, we shall treat these methods separately in order to investigate the propagation of error due to error in essential elements.

3.2.1. *Error Analysis using the First Method*

The propagation of error in the rotational elements, when the *first method* for the IPC algorithm is used [2, 4]. is investigated in this section. If the singular value decomposition (SVD) of $\tilde{Q}$, defined as

$$\tilde{Q} = Q + \delta Q = \tilde{U}\,\tilde{\Lambda}\,\tilde{V}^{\mathrm{T}} = (U + \delta U)(\Lambda + \delta\Lambda)(V + \delta V)^{\mathrm{T}},$$

is an approximation to the true value

$$Q = U\,\Lambda\,V^{\mathrm{T}},$$

where δU, $\delta\Lambda$, δV are the error matrics, then the αth singular vector $\tilde{\mathbf{u}}_\alpha$ of the perturbed matrix $\tilde{Q}$ in terms of the αth singular vector $\mathbf{u}_\alpha$ of the matrix Q can be approximated by the first-order Taylor Series Expansion and is given by [11]

$$\tilde{\mathbf{u}}_\alpha = \mathbf{u}_\alpha + \left.\frac{\partial \tilde{\mathbf{u}}_\alpha}{\partial \tilde{\mathbf{q}}_{\beta\gamma}}\right|_{\tilde{q}_{\beta\gamma} = q_{\beta\gamma}} (\tilde{q}_{\beta\gamma} - q_{\beta\gamma}) \qquad (\alpha, \beta, \gamma = 1, 2, 3). \tag{3.2a}$$

Therefore,

$$|\delta u_\alpha| \le \left| \frac{\partial \tilde{\mathbf{u}}_\alpha}{\partial \tilde{\mathbf{q}}_{\beta\gamma}} \Big|_{\tilde{q}_{\beta\gamma} = q_{\beta\gamma}} \right| \left| \delta q_{\beta\gamma} \right|, \tag{3.2b}$$

where $q_{\beta\gamma}$, $\tilde{q}_{\beta\gamma}$ are the entries of the matrices Q and $\tilde{Q}$, respectively. The derivatives of the singular vectors (for $k = 1, 2, \ldots, 9$), assuming the singular vector **V** is known, are

$$\begin{aligned} \frac{\partial \tilde{\mathbf{u}}_\alpha}{\partial \tilde{\mathbf{q}}_{\beta\gamma}} \Big|_{\tilde{q}_{\beta\gamma} = q_{\beta\gamma}} & \\ &= \frac{1}{\sigma_{\alpha\alpha}} \mathbf{B}_\kappa \mathbf{v}_\alpha - \frac{1}{\sigma_{\alpha\alpha}} (\mathbf{u}_\alpha^{\mathrm{T}} \mathbf{B}_\kappa \mathbf{v}_\alpha) \mathbf{u}_\alpha \\ &\quad + \sum_{\substack{\gamma \ne \alpha \\ \gamma = 1}}^{3} \left[\frac{\sigma_{\gamma\gamma}}{\sigma_{\alpha\alpha}^2 - \sigma_{\gamma\gamma}^2} (\mathbf{u}_\alpha^{\mathrm{T}} \mathbf{B}_\kappa \mathbf{v}_\gamma) + \frac{\sigma_{\gamma\gamma}^2}{\sigma_{\alpha\alpha}(\sigma_{\alpha\alpha}^2 - \sigma_{\gamma\gamma}^2)} (\mathbf{u}_\gamma^{\mathrm{T}} \mathbf{B}_\kappa \mathbf{v}_\alpha) \right] \mathbf{u}_\gamma \end{aligned} \tag{3.2c}$$

$$[\mathbf{B}_\kappa]_{\alpha,\mu} = \begin{cases} 1, & \text{if } \alpha + \mu - 1 = \kappa = 3(\beta - 1) + \gamma \\ 0, & \text{otherwise} \end{cases} \tag{3.2d}$$

where $\sigma_{\alpha\alpha}$ ($\alpha = 1, 2, 3$) are the singular values of Q. The advantage of using Taylor Series Expansion is that it provides the first-order derivative of the principal singular subspace with respect to the essential elements, and it can be combined with other derivatives via the chain rule to obtain first-order perturbations in essential parameters. In this approach, the singular vectors are considered to be vector-valued functions of essential elements. This approach, however, works only if we have distinct singular values. The expressions become ill-conditioned as singular values get close together.

After finding error bounds for the singular vectors, we are in a position to find the same for the rotational elements, which are given by the equations

$$|\delta r_{\alpha\beta}| \le \varepsilon_{u\sigma v} \sum_{\gamma=1}^{3} (|u_{\alpha\gamma}| + |v_{\beta\gamma}|) \qquad (\alpha, \beta = 1, 2, 3), \tag{3.2e}$$

where $u_{\alpha\beta}$, $\sigma_{\alpha\beta}$, and $v_{\alpha\beta}$ are the entries of U, Λ, and V, respectively, and $\varepsilon_{u\sigma v} = \varepsilon_u = \varepsilon_\sigma = \varepsilon_v$ for $-\varepsilon_u \le \delta u_{\alpha\beta} \le \varepsilon_u$, $-\varepsilon_\sigma \le \delta\sigma_{\alpha\beta} \le \varepsilon_\sigma$, and $-\varepsilon_v \le \delta v_{\alpha\beta} \le \varepsilon_v$ has been assumed.

3.2.2. Error Analysis using the "Second Method"

The rotational elements in terms of essential elements, using the ***second*** *method* [1, 4], are given as

$$\begin{aligned} r_{\alpha,\beta} = \pm\, [q_{\alpha,\beta+2}(q_{\alpha+1,\beta+2}q_{\alpha+2,\beta} - q_{\alpha+1,\beta}q_{\alpha+2,\beta+2}) \\ - q_{\alpha,\beta+1}(q_{\alpha+1,\beta}q_{\alpha+2,\beta+1} - q_{\alpha+1,\beta+1}q_{\alpha+2,\beta}) \\ \pm\, t_\alpha(q_{\alpha+1,\beta+1}q_{\alpha+2,\beta+2} - q_{\alpha+1,\beta+2}q_{\alpha+2,\beta+1})]/t_\alpha \sum_{\alpha=1}^{3}\sum_{\beta=1}^{3} q_{\alpha\beta}^2, \end{aligned} \tag{3.3a}$$

where α, $\beta = 1, 2, 3$, and are cyclic (for instance, for $\alpha = 2$, $\beta = 3$, $q_{\alpha+2,\beta+1} = q_{11}$). (t_1, t_2, t_3) are the translations along the x-, y-, and z-axes respectively. Therefore, the error bounds for the rotational elements are

$$|\delta r_{\alpha\beta}| \le \varepsilon_q \sum_{\gamma=1}^{3}\sum_{\kappa=1}^{3} \left|\frac{\partial r_{\alpha\beta}}{\partial q_{\gamma\kappa}}\right| \tag{3.3b}$$

for $-\varepsilon_q \le \delta q_{\gamma\kappa} \le \varepsilon_{q1}$ $(\gamma, \kappa = 1, 2, 3)$. The partial derivatives in Eq. (3.3b) have not been computed in this paper.

3.2.3. *Error Analysis using the "Third Method"*

In this section, the error propagation using the *third method* for computation of R from Q [3] is presented. Defining $H = Q^{\mathrm{T}}$ and

$$\mathbf{W}_\alpha = \mathbf{h}_\alpha \times \mathbf{T} = \mathbf{W}_{\alpha 1}\hat{\imath} + \mathbf{W}_{c2}\hat{\jmath} + \mathbf{W}_{\alpha 3}\hat{k}, \tag{3.4a}$$

where $\mathbf{h}_\alpha$ is the αth row of H, $(\hat{\imath}, \hat{\jmath}, \hat{k})$ are the unit vectors, and

$$\mathbf{W}_{\alpha\beta} = (q_{\beta+1,\alpha}t_{\beta+2} - q_{\beta+2,\alpha}t_{\beta+1}), \tag{3.4b}$$

where $\alpha, \beta = 1, 2, 3$, and are in cyclic order. Therefore, rotational elements are given by

$$\begin{aligned} r_{\alpha\beta} = [(q_{\alpha+2,\beta+1}t_\alpha - q_{\alpha,\beta+1}t_{\alpha+2})(q_{\alpha,\beta+2}t_{\alpha+1} - q_{\alpha+1,\beta+2}t_\alpha) \\ - (q_{\alpha,\beta+1}t_{\alpha+1} - q_{\alpha+1,\beta+1}t_\alpha)(q_{\alpha+2,\beta+2}t_\alpha - q_{\alpha,\beta+2}t_{\alpha+2})] \\ + [q_{\alpha+1,\beta}t_{\alpha+2} - q_{\alpha+2,\beta}t_{\alpha+1}]. \end{aligned} \tag{3.4c}$$

Hence, the bounds are defined as

$$|\delta r_{\alpha\beta}| = \varepsilon_q \sum_{\gamma,\kappa=1}^{3} \left|\frac{\partial r_{\alpha\beta}}{\partial q_{\gamma\kappa}}\right| \quad \text{for } -\varepsilon_q \le \delta q_{\gamma\mu} \le \varepsilon_q\ (\alpha, \beta, \gamma, \kappa = 1, 2, 3). \tag{3.4d}$$

The partial derivatives in Eq. (3.4d) are not computed in this paper.

3.3. Error in Translational Vector

The translational elements t_α ($\alpha = 1, 2, 3$), in terms of essential elements, are given as [2]

$$t_\alpha = \sqrt{\sum_{\beta=1}^{3} \left(-q_{\alpha\beta}^2 + q_{\alpha+1,\beta}^2 + q_{\alpha+2,\beta}^2\right)} \quad (\alpha \text{ is cyclic}). \tag{3.5a}$$

The error bounds for these elements are given as

$$|\delta t_\alpha| \le \varepsilon_q \sum_{\beta=1}^{3} \sum_{\gamma=1}^{3} \left|\frac{q_{\beta\gamma}}{t_\alpha}\right| \tag{3.5b}$$

for $-\varepsilon_q \le \delta_{q\gamma\kappa} \le \varepsilon_q$ ($\gamma, \kappa = 1, 2, 3$) because

$$\frac{\partial t_\alpha}{\partial q_{\beta\gamma}} = \begin{cases} +\dfrac{q_{\beta\gamma}}{t_\alpha} & \text{for } \beta \ne \alpha \\[2ex] -\dfrac{q_{\beta\gamma}}{t_\alpha} & \text{for } \beta = \alpha. \end{cases} \tag{3.5c}$$

3.4. Error in 3-D Motion Parameters

In this section, the errors in the motion parameters due to errors in rotational elements, are studied separately for two representations of the rotation matrix R [2, 7, 8].

3.4.1. Using the First Representation of the Rotation Matrix

From the definitions of the directional cosines of an arbitrary axis, and the angle of rotation around this axis using the *first representation* of R [2, 7], the error bounds for these motion parameters are found to be

$$|\delta v_\alpha| \le \varepsilon_r \sum_{\substack{\beta,\gamma=1 \\ \beta \ne \gamma}}^{3} \left|\frac{\partial v_\alpha}{\partial r_{\beta\gamma}}\right| \quad \text{and} \quad |\delta\theta| \le \varepsilon_r \sum_{\alpha,\beta=1}^{3} \left|\frac{\partial \theta}{\partial r_{\alpha\beta}}\right| \tag{3.6a,b}$$

for $-\varepsilon_q \le \delta_{r\beta\gamma} \le \varepsilon_r$ ($\alpha, \beta, \gamma = 1, 2, 3$). General expressions for the partial derivatives are

$$\frac{\partial v_\alpha}{\partial r_{\beta\gamma}} = \begin{cases} \pm \dfrac{(r_{\alpha+2,\alpha+1} - r_{\alpha+1,\alpha+2})(r_{\gamma\beta} - r_{\beta\gamma})}{d^3} & \text{for } \beta, \gamma = 1, 2, 3;\ \beta \ne \gamma \text{ and } \alpha \text{ is cyclic} \\[2ex] \pm \dfrac{d^2 - (r_{\gamma\beta} - r_{\beta\gamma})^2}{d^3} & \text{for } \beta, \gamma \ne \alpha;\ \beta, \gamma \text{ in cyclic order} \end{cases} \tag{3.6c}$$

and

$$\frac{\partial \theta}{\partial r_{\alpha\beta}} = \pm \frac{(r_{\alpha\beta} - r_{\beta\alpha})}{d\sqrt{4 - d^2}} \qquad \text{(for } \alpha = 1, 2, 3;\ \beta = \alpha + 1;\ \alpha, \beta \text{ are cyclic).} \tag{3.6d}$$

3.4.2. Using the Second Representation of the Rotation Matrix

The *second representation* of the rotation matrix R is used in this section [6, 8]. The error bounds for the motion parameters in this case are:

$$|\delta\theta| \le \left|\frac{d\theta}{dr_{23}}\right| |\delta r_{23}| \tag{3.7a}$$

$$|\delta\phi| \le \left|\frac{\partial\phi}{\partial r_{13}}\right| |\delta r_{13}| + \left|\frac{\partial\phi}{\partial r_{33}}\right| |\delta r_{33}| \tag{3.7b}$$

$$|\delta\psi| \le \left|\frac{\partial\psi}{\partial r_{21}}\right| |\delta r_{21}| + \left|\frac{\partial\psi}{\partial r_{22}}\right| |\delta r_{22}| \tag{3.7c}$$

where

$$\frac{d\theta}{dr_{23}} = \frac{1}{\sqrt{1 - r_{23}^2}} \tag{3.7d}$$

$$\frac{\partial\phi}{\partial r_{13}} = -\frac{r_{33}}{1 - r_{23}^2}; \qquad \frac{\partial\phi}{\partial r_{33}} = \frac{r_{13}}{1 - r_{23}^2}; \tag{3.7e}$$

$$\frac{\partial\psi}{\partial r_{21}} = -\frac{r_{22}}{1 - r_{23}^2}; \qquad \frac{\partial\psi}{\partial r_{22}} = \frac{r_{21}}{1 - r_{23}^2}. \tag{3.7f}$$

IV. Experimental Results

In this section, we present experimental results for the IPC and the GIPC algorithm tested successfully on real data. The various error plots will also be discussed. These plots indicate the relationship between the errors in the input set of data (coordinates of the features) and the errors in the output data (motion parameters).

4.1. Using Real Data

Separate experiments, corresponding to 2-D images of the three positions of an *octbox* in Fig. 3(a), were conducted. The octbox has two parallel octagonal faces opposite to each other, and eight rectangular faces. In the first

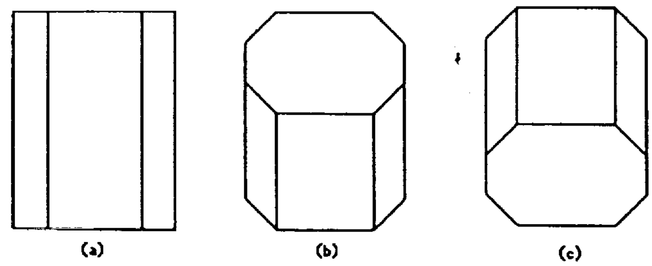


FIG. 3. Experiments with real data. (a) *Octbox* in its initial position; (b) first case of motion where *Octbox* is rotated through 15° around x-axis; and (c) second case of motion where *Octbox* is rotated through 105° around x-axis.

experiment (Fig. 3(b)), the octbox was rotated around the x-axis by $-15°$. In the second experiment (Fig. 3(c)), the octbox was rotated around the x-axis by 105°. In these experiments, rotation aroudn the x-axis means that the angle of rotation, by definition, is *roll*, or equivalently, the direction cosines of the arbitrary axis, around which the octbox rotates, are given by $v_1 = 1.0$; $v_2 = 0.0$; $v_3 = 0.0$. The translation along the three axes is 1 unit each. The data for these three cases of octbox rotations are shown in Figs. 4(a) and 4(b), respectively, where coordinates of the vertices of the octbox before and after the motion are given.

4.1.1. Data Acquisition and Digitization

A solid octbox with side dimensions 1.5" × 4.5" was constructed and placed on a mount capable of motion in all three axes. The mount was flat black to minimize reflection from it. A video camera was placed at the same height above the floor as the octbox and focused on it. The illumination was provided by a television floodlight, also at the same height as the octbox. The octbos was videotaped in 30-s intervals with the illumination source moved from being placed along the axis of the camera to a 45° angle from the camera and finally to a right angle from the camera. These scenes were recorded on a VHS videotape recorder on standard tape at the fastest tape speed allowed. The tape was taken to the Image Processing/Computer Vision Laboratory at Rice University for digitizing and processing. The tape was digitized using a Chorus Data Systems PC-EYE digitizer. Mounted in an IBM-XT, this digitizer is capable of capturing a 640 (H) × 400 (V)pixel image with 6 bits per pixel quantization. The images were then transferred to the main image processing computer system for enhancement and analysis.

x	y	z	X'	Y'
0	-1	-1	3.414214	7.595754
-1.5	-1	-0.5	-3.058409	10.654163
-2	1	1	-0.585786	-0.131652
-1.5	1	-0.5	-0.238625	0.584223
-1.5	1	2.5	-0.379110	-1.268983
0	1	-1	0.449490	0.767327
1.5	1	-0.5	1.193126	0.584223
2	1	1	1.757359	-0.131652

(a)

x	y	z	X'	Y'
0	-1	-1	-4.44949	-1.303225
-1.5	-1	-0.5	-1.936348	0.633123
-2	1	1	-0.449490	0.767327
-1.5	1	-0.5	-0.644449	2.700675
-1.5	1	2.5	-0.136105	0.359012
0	1	-1	3.414214	7.595754
1.5	1	-0.5	3.222247	2.700675
2	1	1	1.348469	0.767327

(b)

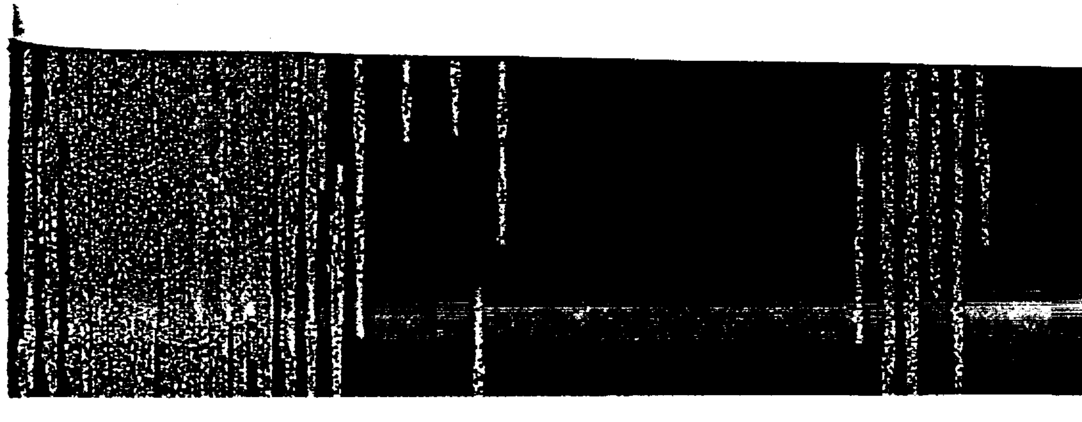
FIG. 4. Real data for motion of *Octbox*. (a) Data for the first case of motion, and (b) data for the second case of motion.

4.1.2. Wireframe Extraction

Our procedure for wireframe extraction consisted of processing the digitized picture for noise removal and edge detection. In both the raw pictures, noise was removed by Wiener filtering. The transfer function is of the form

$$W(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + S_{nn}(u, v)/S_{ff}(u, v)}. \tag{4.1}$$

In our experiment, we assumed that the degradation process $H(u, v)$ was a low-pass filter and the noise-to-signal ratio $S_{nn}(u, v)/S_{ff}(u, v)$ was equal to



$2\sigma^2$. The procedure for removing the noise consisted of the following steps:

Step 1. Separate the processed picture into two regions according to its intensity level.

Step 2. Compute the variance of each region.

Step 3. Process each region using Eq. (4.1) with these variances.

For edge detection, the Sobel edge detector was used. From edges, information about the coordinates of the corners was obtained by comparing the magnitudes of the gradients with a preset threshold value.

The GIPC algorithm has been applied to the first and second experiments, and the results are shown in Figs. 5(a) and 5(b). In both the experiments, the camera was rotated through 10° around its x-axis. With the same set of data

Estimated Translational Vector (up to a scale factor) is:

$$T = [\ 3.863706,\ 3.863707,\ 3.863717\ ]^T$$

Two possible solutions of Rotation Matrix are:

$$R = \begin{bmatrix} -0.333334 & 0.816496 & 0.471405 \\ 0.772304 & -0.050319 & 0.633257 \\ 0.540773 & 0.575154 & -0.613810 \end{bmatrix} \quad \text{and} \quad R' = \begin{bmatrix} 1.000000 & 0.000000 & -0.000000 \\ -0.000000 & 0.996195 & -0.087156 \\ 0.000000 & 0.087156 & 0.996195 \end{bmatrix}$$

The directional cosines of the axis and the angle of rotation about the axis (corresponding to R and R') are respectively:

$v_1 = 0.576984;\ v_2 = 0.688845;\ v_3 = 0.438843;\ \theta = 182.886128$

$v_1' = 1.000000;\ v_2' = 0.000006;\ v_3' = 0.000001;\ \theta' = 354.999983$

Conclusion: Choose R' and its associated parameters as the final solution.

(a)

Estimated Translational Vector (up to a scale factor) is:

$$T = [\ 1.035277,\ 1.035273,\ 1.035283\ ]^T$$

Two possible solutions of Rotation Matrix are:

$$R = \begin{bmatrix} 1.000000 & -0.000002 & 0.000003 \\ 0.000003 & -0.087156 & -0.996195 \\ 0.000002 & 0.996195 & -0.087156 \end{bmatrix} \quad \text{and} \quad R' = \begin{bmatrix} -0.333332 & 0.471405 & -0.816496 \\ 0.772304 & 0.633257 & 0.050321 \\ 0.540773 & -0.613810 & -0.575153 \end{bmatrix}$$

The directional cosines of the axis and the angle of rotation about the axis (corresponding to R and R') are respectively:

$v_1 = 1.000000;\ v_2 = 0.000000;\ v_3 = 0.000002;\ \theta = 95.000000$

$v_1' = 0.431055;\ v_2' = 0.860937;\ v_3' = -0.195299;\ \theta' = 230.385837$

Conclusion: Choose R and its associated parameters as the final solution.

(b)

FIG. 5. Demonstration of GIPC algorithm using real data. (a) For the first case of *Octbox* rotation, and (b) for the second case of *Octbox* rotation.

for these experiments for the octbox rotation (Figs. 3(b), 3(c)), the directional cosines are found to be same. The angles of rotation are $-5°$ and $95°$, respectively, which means that the angles of rotation of the octbox are added to by the amount of rotation by the camera, and that indeed should be the case. These two experiments show the success of GIPC algorithm with real data.

V. Concluding Remarks

In this paper, a generalized expression for motion-analysis equation was derived, and the other three cases of motion-analysis were found to be special cases of this case. In addition, the expression for error bounds were dervied for various stages of the algorithm.

REFERENCES

1. X. Zhuang and R. M. Haralick. Two-view motion analysis, stereo vision and a moving camera's positioning: Their equivalence and new solution procedure, in *IEEE International Conference on Robotics and Automation, March, 25–28, 1985*, pp. 6-10.
2. R. Y. Tsai and T. S. Huang, Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces, *IEEE Trans. Pattern Anal. Mach. Intelligence* PAMI-6, 1984, 13-27.
3. H. C. Longuiet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature* **293**, 1981, 133-135.
4. S. Fotedar, R. J. P. deFigueiredo, and K. Krishen, Determination of motion parameters of a moving object from moving camera data, in *Proceedings of Robexs'89, the Fourth Annual Workshop on Robotics and Expert Systems, Palo Alto, California, August, 2–4, 1989*, Vol. 4, pp. 143-154.
5. A. N. Netravali, T. S. Huang, A. S. Krishnakumar, and R. J. Holt, Algebraic methods in 3-D motion estimation from two-view point correspondences, *Int. J. Imaging Systems Technol.* **1**, 1989, 78-99.
6. R. Ray, J. Birk, and R. B. Kelley, Error analysis of surface normals determined by radiometry, *IEEE Trans. Pattern Anal. Mach. Intelligence* PAMI-5, 1983, 631-645.
7. S. Fotedar, R. J. P. deFigueiredo, and K. Krishen, Error analysis for an image point correspondence algorithm in, *Proceedings of Robexs'89, the Fourth Annual Workshop on Robotics and Expert Systems, Palo Alto, California, Aug. 2–4, 1989*, Vol. 4, pp. 131-141.
8. S. Ganapathy, Decomposition of transformation matrices for robot vision, in *Proc. Int. Conf. on Robotics, Mar. 13–15, 1984.*
9. F. B. Hildebrand, *Introduction to Numerical Analysis*, McGraw-Hill, New York, 1956.
10. C. L. Lawson and R. J. Hanson, *Solving Least Squares Problems*, Prentice-Hall, Englewood Cliffs, MA, 1974.
11. R. J. Vaccaro and A. C. Kot, A perturbation theory for the analysis of SVD-based algorithms, in *IEEE International Conference on Acoustics, Speecgh, and Signal Processing, Apr. 6–9, 1987*, Vol. 3, pp. 1613-1616..