COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 62, No. 3, November, pp. 279–297, 1995

# Integrated Matching and Segmentation of Multiple Features in Two Views[1]

SANGHOON SULL[2] AND NARENDRA AHUJA

*Beckman Institute and Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 405 North Mathews Avenue, Urbana, Illinois 61801*

Received March 17, 1993; accepted September 15, 1994

**We present an integrated method to match multiple features including points, regions, and lines in two perspective images, and simultaneously segment them such that all features in each segment have the same 3D motion. The method uses local affine (first-order) approximation of the displacement field under the assumption of locally rigid motion. Each distinct motion is represented in the image plane by a distinct set of values for six displacement parameters. To compute the values of these parameters, the 6D space is split into two 3D spaces, and each is exhaustively searched coarse-to-fine. This yields two results simultaneously, correspondences between features and segmentation of features into subsets corresponding to locally rigid patches of moving objects. Since matching is based on the 2D approximation of 3D motion, problems due to motion or object boundaries and occlusion can be avoided. Large motion is also handled in a manner unlike the methods based on flow field. Integrated use of the multiple features not only gives a larger number of features (overconstrained system) but also reduces the number of candidate matches for the features, thus making matching less ambiguous. Experimental results are presented for four pairs of real images.** © 1995 Academic Press, Inc.

## 1. INTRODUCTION

The problem we address in this paper is to match and segment features such that all features in each segment have the same three-dimensional (3D) motion from a pair of two-dimensional (2D) images. The difficulty of establishing feature correspondences comes from the 3D structural discontinuities and occlusions, as well as from independently moving objects. One goal of this paper is to show that integration of matching and segmentation and the use of multiple features simultaneously makes it easier to find correspondences.

The existing methods for matching can be largely divided into two categories which are based on the computation of flow field [1–3] or the establishment of discrete feature correspondences [4–11], respectively. Even though the flow-based approaches do not require feature extraction and matching, problem occurs when the motion is large, as in the case for stereo. Medioni and Nevatia [4] establish line correspondences based on relaxation. McIntosh and Mutch [5] and Liu and Huang [6] propose algorithms for finding line correspondences. Sethi and Jain [7] present an algorithm for finding smooth point trajectories over an image sequence. Crowley *et al.* [8] and Deriche and Faugeraus [9] present algorithms for tracking lines by using local affine models using four parameters for each line. Venkateswar and Chellappa [10] describe a hierarchical matching method in which the matching starts at the surface level and ends at the line level. Weng *et al.* [11] present a point-matching method which yields dense flow. Most of these methods for discrete feature matching [4, 5, 7–11] are based on maximization of compatibility of 2D attributes and relative 2D locations. They may fail, for example, near the image boundary where relative locations are not preserved, or for motion along the optical axis which involves significant change in the relative 2D locations of features. Sawhney and Hanson [12] describe a method for tracking a set of three lines based on an affine transformation using four parameters but do not discuss segmentation and matching of lines having different motions. In fact, all of the existing tracking methods need an initial matching in the first two frames as a bootstrapping step. Sull and Ahuja [13] present an algorithm for matching and segmentation of regions in two views using affine transformations. In this paper, the algorithm is extended to utilize multiple features. It is our intention to show that the matching problem becomes easier by matching and segmenting multiple features simultaneously.

Adiv [14] presents a method for the segmentation of optic flow based on the Hough transform. Black and Anandan [2] describe a model for incremental estimation of

[1] The support of the Defense Advanced Research Projects Agency and the National Science Foundation under Grant IRI-89-02728 is gratefully aknowledged.

[2] Current address: Mail Stop 210-1, NASA Ames Research Center, Moffett Field, CA 94035-1000.

1077-3142/95 $12.00

flow field over an image sequence based on a line process [15]. But, as indicated in [16], a line process cannot integrate information from nearby but disconnected regions even if they belong to the same object. Darrell and Pentland [16] present a method for scene segmentation based on a simple direct motion model of translating objects. All these methods are based on flow and therefore problems occur for large motion.

In this paper we present an integrated method which matches and simultaneously segments multiple features such as points, regions, and lines in two perspective images. Our method is based on local affine approximation of the displacement field which is derived under the assumption of locally rigid motion (see Section 2). Thus, the displacement vector $[d_x, d_y]'$ at $t_1$, which represents the displacement caused by motion of a point located at $(x, y)$, is locally approximated by

$$d_x = c_0 + c_1 x + c_2 y$$
$$d_y = c_3 + c_4 x + c_5 y.$$

Each distinct motion is represented in the image plane by a distinct set of values of six parameters $\mathbf{c} =^{\text{def}} \{c_0, c_1, c_2, c_3, c_4, c_5\}$. All sets of values supported by feature locations in two adjacent frames are identified by exhaustive coarse-to-fine search. However, to reduce computational complexity the 6D parameter space is decomposed into two disjoint 3D spaces. The support for any set is computed from the image plane distances between the observed feature locations and those predicted by the parameter values. The well-supported sets of values thus found yield two results simultaneously; first, they establish correspondences between features, and second, they segment the features into subsets corresponding to locally rigid patches of the moving objects.

Since features are matched based on 3D motion constraints, problems due to motion or object boundaries and occlusion can be avoided. Further, our method can handle large motion as well as small motion. The integrated use of multiple features not only gives a larger number of features (overconstrained system) but also reduces the number of candidate matches for features, thus making matching less ambiguous.

In this paper, the term *point feature* denotes both a distinguished image point as well as the intersection of lines (line points). The former is defined as a point whose location corresponds to the local maxima or minima of the intensity values.

Section 2 discusses local affine modeling of the displacement field. Section 3 describes affine models for individual features and the integration of multiple features. Section 4 describes the different steps of the algorithm. Section 5 presents the details of implementation and the results obtained from four experiments. Section 6 presents conclusions and extensions.

## 2. AFFINE MODEL FOR DISPLACEMENT FIELD

This section presents an affine displacement model to locally describe the image plane motion of an object undergoing a general 3D rigid motion. The subscript $i$ for the $i$th feature and the $i$th segment is dropped when there is no confusion. The $X$ and $X'$ denote variables or labels at $t_1$ and $t_2$, respectively.

Consider a point $\vec{X}_0 = [X_0, Y_0, Z_0]^T$ on an object in 3 D at time $t_1$. Let $\vec{X}_0'$ be the corresponding point at $t_2$. Denoting the rotation matrix and translation vector as $\mathbf{R}$ and $\vec{T}$, respectively, the general 3D rigid motion is expressed by

$$\vec{X}_0' = \mathbf{R}\vec{X}_0 + \vec{T}. \tag{1}$$

Assuming the perspective projection with the focal length equal to one, the image coordinates $(x_0', y_0')$ of $\vec{X}_0'$ at $t_2$ can be expressed as

$$x_0' = \frac{r_{11}x_0 + r_{12}y_0 + r_{13} + T_X/Z_0}{r_{31}x_0 + r_{32}y_0 + r_{33} + T_Z/Z_0} \tag{2}$$

$$y_0' = \frac{r_{21}x_0 + r_{22}y_0 + r_{23} + T_Y/Z_0}{r_{31}x_0 + r_{32}y_0 + r_{33} + T_Z/Z_0}, \tag{3}$$

where $(x_0, y_0)$ are the image coordinates at $t_1$, and $r_{11}, \ldots, r_{33}$ are the nine elements of $\mathbf{R}$.

Consider a point $\vec{X} = [X, Y, Z]^T$ on the same object in a neighborhood of $\vec{X}_0$. If we assume that the depth difference is small compared to $Z_0$ (i.e., $|Z - Z_0|/Z_0 \ll 1$), we have

$$Z = Z_0\left(1 + \frac{Z - Z_0}{Z_0}\right) \approx Z_0. \tag{4}$$

Let $(x, y)$ and $(x', y')$ be the image coordinates of $\vec{X}$ at $t_1$ and $\vec{X}'$ at $t_2$, respectively. Then, the displacement field of the point $\vec{X}$ in the neighborhood of $\vec{X}_0$ is represented by

$$d_x(x, y) \overset{\text{def}}{=} x' - x = \frac{r_{11}x + r_{12}y + r_{13} + T_X/Z_0}{r_{31}x + r_{32}y + r_{33} + T_Z/Z_0} - x \tag{5}$$

$$d_y(x, y) \overset{\text{def}}{=} y' - y = \frac{r_{21}x + r_{22}y + r_{23} + T_Y/Z_0}{r_{31}x + r_{32}y + r_{33} + T_Z/Z_0} - y. \tag{6}$$

Since we are considering the neighbor point of $\vec{X}_0$, which has a small depth difference as stated above, we can assume that the first partial derivatives of $d_x(x, y)$ and $d_y(x, y)$ are continuous in a closed neighborhood of $(x_0, y_0)$ and the second partial derivatives exist in the open neighborhood.

Therefore, we can derive expressions for the values of the displacement field near a point $(x_0, y_0)$ by using Taylor's series [17]:

$$d_x(x,y) = d_x(x_0,y_0) + \left((x-x_0)\frac{\partial}{\partial x} + (y-y_0)\frac{\partial}{\partial y}\right) d_x(x_0,y_0) + R_x^{(2)} \qquad (7)$$

$$d_y(x,y) = d_y(x_0,y_0) + \left((x-x_0)\frac{\partial}{\partial x} + (y-y_0)\frac{\partial}{\partial y}\right) d_y(x_0,y_0) + R_y^{(2)}, \qquad (8)$$

where

$$R_x^{(2)} \overset{\text{def}}{=} \frac{1}{2}\left(((x-x_0)\frac{\partial}{\partial x} + (y-y_0)\frac{\partial}{\partial y}\right)^2 d_x(x_0+\alpha(x-x_0), y_0+\alpha(y-y_0)), \quad 0<\alpha<1 \qquad (9)$$

$$R_y^{(2)} \overset{\text{def}}{=} \frac{1}{2}\left(((x-x_0)\frac{\partial}{\partial x} + (y-y_0)\frac{\partial}{\partial y}\right)^2 d_y(x_0+\beta(x-x_0), y_0+\beta(y-y_0)), \quad 0<\beta<1. \qquad (10)$$

If we assume that the values of the second partial derivatives in Eqs. (9) and (10) are not large, we can ignore the remainders $R_x^{(2)}$ and $R_y^{(2)}$ since the values of $|x - x_0|$ and $|y - y_0|$ are practically very small in the neighborhood of $(x_0, y_0)$. Therefore, if we assume that (1) $|Z - Z_0|/Z_0 \ll 1$, and (2) the values of the second partial derivatives in the neighborhood of $(x_0, y_0)$ are not large, we can locally approximate the displacement field using affine transformations:

$$d_x = x' - x = c_0 + c_1x + c_2y \qquad (11)$$

$$d_y = y' - y = c_3 + c_4x + c_5y. \qquad (12)$$

To derive more specific assumptions for the validity of the first-order approximations, we rewrite the denominator $h(x, y)$ in Eqs. (5) and (6):

$$\begin{aligned} h(x,y) &= r_{31}(x_0 + (x-x_0)) + r_{32}(y_0 + (y-y_0)) \\ &\quad + r_{33} + \frac{T_Z}{Z_0} \qquad (13) \\ &= h(x_0,y_0) + r_{31}(x-x_0) + r_{32}(y-y_0). \end{aligned}$$

Since $|r_{31}| \leq 1$, $|r_{32}| \leq 1$ and $|x - x_0| \ll 1$ and $|y - y_0| \ll 1$ for the CCD cameras available currently, we can approximate $h(x, y)$ as a constant near $(x_0, y_0)$ if $|h(x_0, y_0)| > 1 - \varepsilon$ ($\varepsilon$: a small positive number). Therefore, if we assume that (i) $|Z - Z_0|/Z_0 \ll 1$ and (ii) $|h(x_0, y_0)| > 1 - \varepsilon$ ($\varepsilon$: a small positive number), we can locally approximate the displacement field using the affine transformations. We note here that Assumption (ii) is a special case of Assumption (2) stated above. Further, the assumption that (a) the rotation angle is small and (b) $T_Z/Z_0 \ll 1$ is a special case of Assumption (ii), which was used in [11, 14]. In summary, we can model the displacement field by using the affine transformations if Assumptions (1) and (2) are satisfied. We note here that Assumptions (1) and (2) are general enough to be satisfied in most cases except at the object or occlusion boundary.

## 3. AFFINE MODELS FOR FEATURES

This section presents affine displacement models for points, regions, and lines, which are used later to integrate segmentation and matching of image features of points, regions, and lines.

### 3.1. *Point*

The displacement is modeled locally using $\mathbf{c} =^{\text{def}} \mathbf{c}_x \cup \mathbf{c}_y$, where $\mathbf{c}_x =^{\text{def}} \{c_0, c_1, c_2\}$ and $\mathbf{c}_y =^{\text{def}} \{c_3, c_4, c_5\}$ as in Eqs. (11) and (12). Then, given $\mathbf{c}$, we define the following error measures for a pair of point features $(P_i, P'_j)$ at $t_1$ and $t_2$:

$$\delta_{x,P,ij}(\mathbf{c}_x) \overset{\text{def}}{=} x'_j - (c_0 + (1+c_1)x_i + c_2y_i) \qquad (14)$$

$$\delta_{y,P,ij}(\mathbf{c}_y) \overset{\text{def}}{=} y'_j - (c_3 + c_4x_i + (1+c_5)y_i). \qquad (15)$$

Here $P$ in the subscript denotes the fact that the features under consideration are point features. Then, the image error for the point features $(P_i, P'_j)$ is defined as

$$\delta_{P,ij}(\mathbf{c}) \overset{\text{def}}{=} \sqrt{\delta_{x,P,ij}(\mathbf{c}_x)^2 + \delta_{y,P,ij}(\mathbf{c}_y)^2}. \qquad (16)$$

### 3.2. *Region*

Let $R$ and $R'$ be the corresponding regions at $t_1$ and $t_2$, respectively. For a smooth function $f$, we know

$$\int\int_{R'} f(x,y)\,dx\,dy = \int\int_R f(x+d_x, y+d_y)J\,dx\,dy, \qquad (17)$$

where $J$ is the Jacobian

$$J = \left|\left(1 + \frac{\partial d_x}{\partial x}\right)\left(1 + \frac{\partial d_y}{\partial y}\right) - \frac{\partial d_x}{\partial y}\frac{\partial d_y}{\partial x}\right|. \qquad (18)$$

Using Eqs. (11), (12), (17), and (18), we have

$$J = |(c_1 + 1)(c_5 + 1) - c_2c_4|. \tag{19}$$

If $f = 1$, then the equation simply represents the area relationship

$$J = \frac{M'_{00}}{M_{00}}, \tag{20}$$

where

$$\begin{aligned} M_{ij} &\overset{\text{def}}{=} \int\int_R x^i y^j \, dx \, dy \\ M'_{ij} &\overset{\text{def}}{=} \int\int_{R'} x^i y^j \, dx \, dy. \end{aligned} \tag{21}$$

For $f = x$ and $f = y$, we obtain two equations for each region correspondence from Eqs. (11), (12), and (17):

$$\frac{M'_{10}}{J} - M_{10} = c_0M_{00} + c_1M_{10} + c_2M_{01}$$

$$\frac{M'_{01}}{J} - M_{01} = c_3M_{00} + c_4M_{10} + c_5M_{01}.$$

These can be rewritten as

$$C'_x = c_0 + (1 + c_1)C_x + c_2C_y \tag{22}$$

$$C'_y = c_3 + c_4C_x + (1 + c_5)C_y, \tag{23}$$

where

$$C'_x = \frac{M'_{10}}{M'_{00}} \quad C'_y = \frac{M'_{01}}{M'_{00}}$$

$$C_x = \frac{M_{10}}{M_{00}} \quad C_y = \frac{M_{01}}{M_{00}}.$$

These are the relationships of the 2D coordinates of centroids of the corresponding regions.

In general, given **c**, a set of six parameters, we define the following error measures for a pair of regions $(R_i, R'_j)$ at $t_1$ and $t_2$:

$$\delta_{x,R,ij}(\mathbf{c}_x) \overset{\text{def}}{=} C'_{x,j} - (c_0 + (1 + c_1)C_{x,i} + c_2C_{y,i}) \tag{24}$$

$$\delta_{y,R,ij}(\mathbf{c}_y) \overset{\text{def}}{=} C'_{y,j} - (c_3 + c_4C_{x,i} + (1 + c_5)C_{y,i}). \tag{25}$$

The image error for the pair of regions $(R_i, R'_j)$ is defined as

$$\delta_{R,ij}(\mathbf{c}) \overset{\text{def}}{=} \sqrt{\delta_{x,R,ij}(\mathbf{c}_x)^2 + \delta_{y,R,ij}(\mathbf{c}_y)^2}. \tag{26}$$

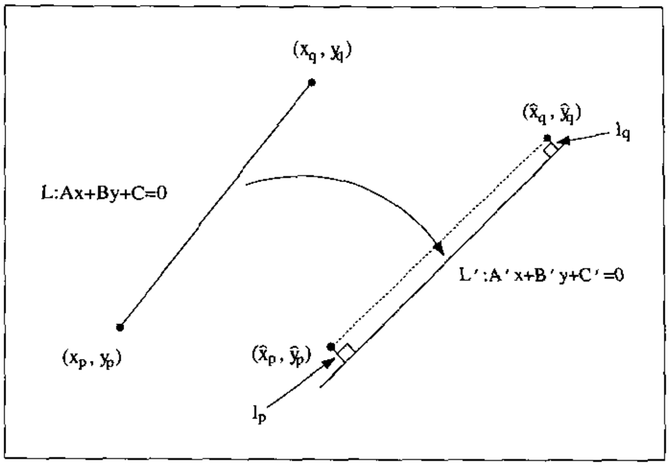

FIG. 1. Illustration of constraint for lines.

### 3.3. *Line*

Let $L$ and $L'$ be the corresponding lines at $t_1$ and $t_2$, respectively as shown in Fig. 1. Consider an end point $(x_p, y_p)$ of $L$ at $t_1$ in the image plane. Then, the image coordinates $(\hat{x}_p, \hat{y}_p)$ predicted from the affine transformations are expressed by using Eqs. (11) and (12).

Let $l_p$ be the perpendicular distance from the predicted image coordinates to the corresponding line in the image plane at time $t_2$. Then, referring to Fig. 1, we have

$$l_p \overset{\text{def}}{=} \frac{|A'\hat{x}_p + B'\hat{y}_p + C'|}{\sqrt{A'^2 + B'^2}}. \tag{27}$$

Since $l_p$ should be zero, we remove the absolute sign in the above equation. For the other end point $(x_q, y_q)$, $l_q$ should be also zero. Most of the existing line tracking work [8, 9, 12] used locations of end points of lines in their computation although they are known to be very unstable. In this paper, the longitudinal information of a line is only utilized in such a way that $\delta_{P,ij}$ defined in Eq. (16) for two center points of $L$ and $L'$ should be less than a fraction of the length of $L$.

Therefore, given **c**, a set of six parameters, the following error measures for $l_p$ and $l_q$ are derived for a pair of lines $(L_i, L'_j)$ at $t_1$ and $t_2$, respectively:

$$\delta_{p,L,ij}(\mathbf{c}) \overset{\text{def}}{=} \frac{\begin{array}{c} A'(c_0 + (1 + c_1)x_{p,i} + c_2y_{p,i}) \\ + B'(c_3 + c_4x_{p,i} + (1 + c_5)y_{p,i}) + C' \end{array}}{\sqrt{A'^2 + B'^2}} \tag{28}$$

$$\delta_{q,L,ij}(\mathbf{c}) \overset{\text{def}}{=} \frac{\begin{array}{c} A'(c_0 + (1 + c_1)x_{q,i} + c_2y_{q,i}) \\ + B'(c_3 + c_4x_{q,i} + (1 + c_5)y_{q,i}) + C' \end{array}}{\sqrt{A'^2 + B'^2}}. \tag{29}$$

Then, the image error for the pair of lines $(L_i, L'_j)$ is defined as

$$\delta_{L,ij}(\mathbf{c}) \stackrel{\text{def}}{=} \sqrt{\frac{\delta_{p,L,ij}(\mathbf{c})^2 + \delta_{q,L,ij}(\mathbf{c})^2}{2}}. \tag{30}$$

We note here that Eqs. (28) and (29) involve both sets of parameters $\mathbf{c}_x$ and $\mathbf{c}_y$ while only one set appears in Eqs. (14) and (15), and Eqs. (24) and (25).

### 3.4. *Integration of Multiple Features*

The affine transformations for points, regions, and lines are given by Eqs. (14) and (15), Eqs. (24) and (25), and Eqs. (28) and (29), respectively. Note that the error measure of each feature ($\delta_{P,ij}$, $\delta_{R,ij}$, $\delta_{L,ij}$) is in terms of the same unit, i.e., the image error, and we can treat them equally when we construct a support function to be maximized. Consider a group of *local* features at $t_1$ which are close together (within a predetermined distance) since the affine transformations are valid locally in the image plane. Let $n_P$, $n_L$ and $n_R$ be the numbers of points, regions, and lines under consideration at $t_1$, respectively. For each pair of the same type of features at $t_1$ and $t_2$, we define a support function $F$ of $\mathbf{c} = \{c_0, c_1, c_2, c_3, c_4, c_5\}$ as

$$F(\mathbf{c}) \stackrel{\text{def}}{=} \sum_{i=1}^{n_P}\sum_{j=1}^{n_{P,i}} w_{P,ij}\Psi_{\varepsilon_p}(\delta_{P,ij}(\mathbf{c})) + \sum_{i=1}^{n_R}\sum_{j=1}^{n_{R,i}} w_{R,ij}\Psi_{\varepsilon_p}(\delta_{R,ij}(\mathbf{c})) + \sum_{i=1}^{n_L}\sum_{j=1}^{n_{L,i}} w_{L,ij}\Psi_{\varepsilon_p}(\delta_{L,ij}(\mathbf{c})), \tag{31}$$

where

$$\Psi_{\varepsilon_p}(x) \stackrel{\text{def}}{=} \begin{cases} 1 - \dfrac{|x|}{\varepsilon_p} & \text{if } -\varepsilon_p < x < \varepsilon_p \\ 0 & \text{otherwise;} \end{cases} \tag{32}$$

$\varepsilon_p$ is a predetermined number; $w_{P,ij}$, $w_{R,ij}$, and $w_{L,ij}$ represent weights; and $n_{P,i}$ represents the number of the neighboring point features at $t_2$ with 2D attributes similar to those of the $i$th point at $t_1$ ($n_{R,i}$ and $n_{L,i}$ are defined in the same way). The choice of $\Psi_{\varepsilon_p}(x)$ does not matter as long as it is a decreasing function of $x$ since it is used only to find the location of the maximum of $F$ by exhaustive search (empirical results support this observation). Note that only those pairs of features having similar attributes at two time instants are considered to reduce the computation as well as to increase confidence in the solution.

Our goal is to find all sets of the six parameters $\hat{\mathbf{c}}$ corresponding to the dominant local maxima of $F$. The support function $F$ is likely to have many small local maxima since the extracted features contain unknown subsets, each having a different, unknown motion. Therefore, exhaustive search is one way to obtain all dominant local maxima. However, since the 6D space is too large to search, we decompose this into two disjoint 3D spaces. A similar decomposition of 6D space was also used by Adiv [14] for the Hough transform. We define the decomposed support functions $F_x$ and $F_y$ as

$$F_x(\mathbf{c}_x) \stackrel{\text{def}}{=} \sum_{i=1}^{n_P}\sum_{j=1}^{n_{P,i}} w_{P,ij}\Psi_{\varepsilon_p/\sqrt{2}}(\delta_{x,P,ij}(\mathbf{c}_x)) + \sum_{i=1}^{n_R}\sum_{j=1}^{n_{R,i}} w_{R,ij}\Psi_{\varepsilon_p/\sqrt{2}}(\delta_{x,R,ij}(\mathbf{c}_x)) \tag{33}$$

$$F_y(\mathbf{c}_y) \stackrel{\text{def}}{=} \sum_{i=1}^{n_P}\sum_{j=1}^{n_{P,i}} w_{P,ij}\Psi_{\varepsilon_p/\sqrt{2}}(\delta_{y,P,ij}(\mathbf{c}_y)) + \sum_{i=1}^{n_R}\sum_{j=1}^{n_{R,i}} w_{R,ij}\Psi_{\varepsilon_p/\sqrt{2}}(\delta_{y,R,ij}(\mathbf{c}_y)), \tag{34}$$

where the terms representing line pairs are not included since the expressions for $\delta_{p,L,ij}$ and $\delta_{q,L,ij}$ in Eqs. (28) and (29) involve all six parameters.

To obtain all the dominant local maxima of $F$, we find the global maximum of $F$ one at a time for the largest remaining features as follows: For the largest group of local features, we first find $N_3$ of the candidate sets of $\mathbf{c}_x$ and $\mathbf{c}_y$ corresponding to the peak values of $F_x$ and $F_y$ by searching each quantized 3D space. Then, for $N_3^2$ combinations of $\{\mathbf{c}_x, \mathbf{c}_y\}$, we select $\hat{\mathbf{c}}$ corresponding to the maximum of $F$ defined in Eq. (31). These two steps are performed at coarse-to-fine resolution to reduce the computation. Note that $\varepsilon_p$ in Eq. (31) is a function of resolution in the search space. The search is separately performed in the 3D parameter spaces for feature points, line points, and regions. Then, the combination of solution triples which corresponds to the maximum value of $F$ is obtained. Feature correspondences are established by using the set of the six parameters yielding the maximum value of $F$ and the corresponding matched features comprise a segment. After removing these matched features from further consideration, the above process continues until there is no dominant peak in the search space.

Although lines are used only in the combination step (since the six parameters are not separable into two disjoint sets), line points are utilized in the separate spaces. The integrated use of the multiple features not only gives a larger number of features (overconstrained system) but also reduces the number of candidate matches for feature points. Also, note that features such as regions and lines (therefore, line points) have a small number of neighbors (only one in many cases) which have similar 2D attributes since those features have more measurable 2D attributes than points. As a result, when all of these features are conservatively used together in the search, the dominant peaks of $F$ values are obtained.

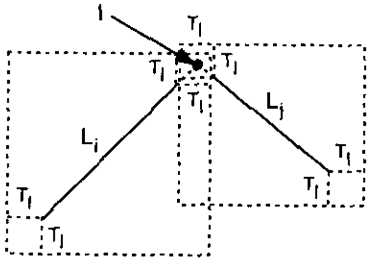

FIG. 2. An intersection point I is said to be *near* two lines $L_i$ and $L_j$, if I is inside the upright rectangle containing $L_i$ as well as that containing $L_j$, after each rectangle has been expanded by $T_l$ pixels.

The affine modeling of the displacement field using six parameters is able to describe more general motion than using four parameters as in [12]. If four parameters are used to describe the displacement field, then both scale and rotation parameters appear in expressions for $d_x$ and $d_y$ and therefore the support function $F$ cannot be decomposed into two parts. Consequently, the 4D space needs to be searched instead of 3D space, as is the case in our method.

### 3.5. *Linear Estimation of Six Affine Parameters*

Given a group of feature correspondences, denoted by $S_l$, a set $\mathbf{c}_l$ of the six affine parameters of $S_l$ is linearly computed, thus yielding more accurate values than $\hat{\mathbf{c}}_l$ obtained by searching the quantized space. This linear computation is useful especially for the step of the algorithm presented later to merge segments. (A merging step is sometimes desirable since matched features corresponding to different segments are oversegmented by the local affine model. If any two segments satisfy one affine transformation, they are merged into one segment.) Our goal is to linearly compute $\mathbf{c}_l = \{c_{0l}, c_{1l}, c_{2l}, c_{3l}, c_{4l}, c_{5l}\}$, which best describes the given correspondences in $S_l$. Let $m_{P,l}$, $m_{L,l}$, and $m_{R,l}$ be the numbers of matched pairs $(P_i, P_i')$, $(R_i, R_i')$, and $(L_i, L_i')$ of points, regions, and lines in $S_l$, respectively. (Here, we assume that features are relabeled in such a way that matched features have the same subscripts.)

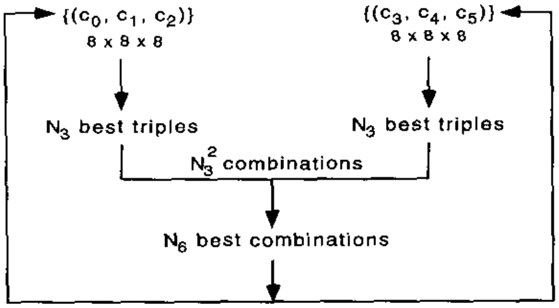

FIG. 3. Multiresolution search with decomposition.

Then, we can define an objective function to be minimized with respect to $\mathbf{c}_l$ as

$$\begin{aligned}\delta_{S_l}^2(\mathbf{c}_l) \overset{\text{def}}{=} \sum_{i=1}^{m_P} w_{P,ii}\delta_{P,ii}^2(\mathbf{c}_l) + \sum_{i=1}^{m_R} w_{R,ii}\delta_{R,ii}^2(\mathbf{c}_l)\\ + \sum_{i=1}^{m_L} w_{L,ii}\delta_{L,ii}^2(\mathbf{c}_l),\end{aligned} \tag{35}$$

where $w_{P,ii}$, $w_{R,ii}$, and $w_{L,ii}$ represent weights and $\delta_{P,ij}$, $\delta_{R,ij}$, and $\delta_{L,ij}$ are defined in Eqs. (16), (26), and (30), respectively. Note that each term in the above objective function has the same unit. This minimization is a standard linear least-squares problem which can be easily solved.

To measure the goodness of the segment $S_l$ and the estimated $\mathbf{c}_l$, we define the average image error for matched features in $S_l$ as follows:

$$\bar{\delta}_{S_l}(\mathbf{c}_l) \overset{\text{def}}{=} \sqrt{\frac{\delta_{S_l}^2(\mathbf{c}_l)}{(m_P + m_L + m_R)}}. \tag{36}$$

### 3.6. *Correlation Error for Individual Features*

It is desirable to verify the matched features in a segment $S_l$ by checking the values of correlation errors defined below since the error measures previously defined by $\delta_{P,ij}$, $\delta_{R,ij}$, and $\delta_{L,ij}$ consider only the centers of regions and perpendicular distances for lines, for example.

Let $I_1$ and $I_2$ be image intensity functions at $t_1$ and $t_2$, respectively. Given $S_l$ and $\mathbf{c}_l$, we can define the following errors for a pair of corresponding regions $(R_i, R_j')$ in $S_l$ at $t_1$ and $t_2$, respectively,

$$\varepsilon_{R_i}(\mathbf{c}_l) \overset{\text{def}}{=} \frac{1}{A} \sum_{(x,y)\in R_i} |I_1(x, y) - I_2(x + d_x(\mathbf{c}_l), y + d_y(\mathbf{c}_l))| \tag{37}$$

$$\varepsilon_{R_j'}(\mathbf{c}_l) \overset{\text{def}}{=} \frac{1}{A'} \sum_{(x,y)\in R_j'} |I_1(x - d_x'(\mathbf{c}_l), y - d_y'(\mathbf{c}_l)) - I_2(x, y)|, \tag{38}$$

where $[d_x', d_y']'$ is the displacement vector defined from $I_2$ to $I_1$ (which is easily computed given $\mathbf{c}_l$), $A$ and $A'$ are the areas (number of pixels) of $R_i$ and $R_j'$ at $t_1$ and $t_2$, respectively. Then, we can define a simple correlation error measure for the matched $R_i$ and $R_j'$ as follows:

$$\varepsilon_{R,ij}(\mathbf{c}_l) \overset{\text{def}}{=} \max(\varepsilon_{R_i}(\mathbf{c}_l), \varepsilon_{R_j'}(\mathbf{c}_l)). \tag{39}$$

For corresponding points $(P_i, P_j')$ in $S_l$, $\varepsilon_{P_i}(\mathbf{c}_l)$ and $\varepsilon_{P_j'}(\mathbf{c}_l)$ are defined in the same way as in Eqs. (37) and (38), while the summations are done over $N_{P_i}$ and $N_{P_j'}$ instead

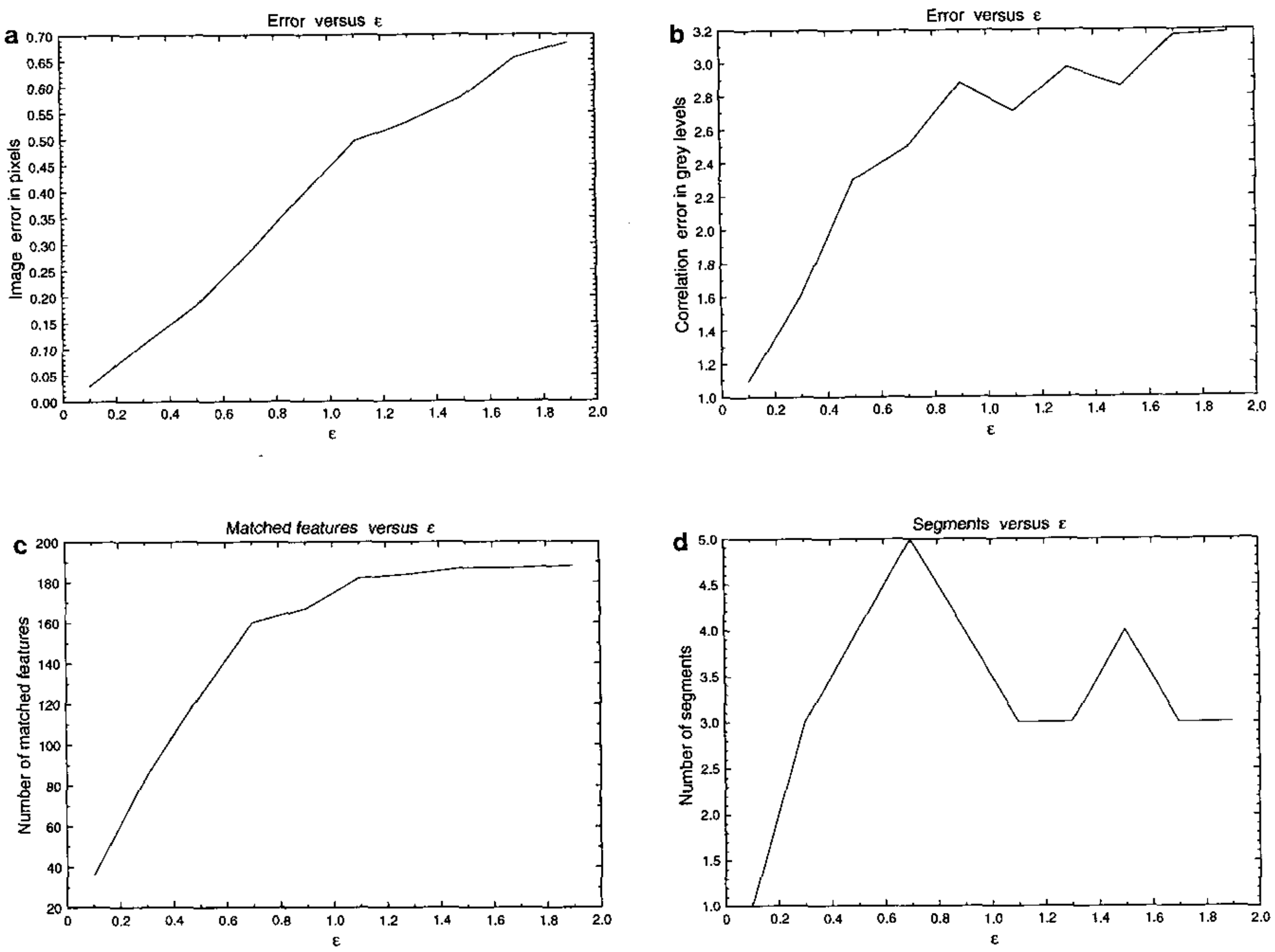

FIG. 4. Sensitivity of results to varying $\varepsilon_p^{(3)}$ for two images in Experiment 4. (a) Average image error. (b) Average correlation error. (c) Number of matched features. (d) Number of segments.

of $R_i$ and $R'_j$, respectively, where $N_{P_i}$ and $N_{P'_j}$ represent windows (of size 7 × 7) centered at the locations of $P_i$ and $P'_j$, respectively. Then, we have a correlation error for $(P_i, P'_j)$ defined as

$$\varepsilon_{P,ij}(\mathbf{c}_l) \overset{\text{def}}{=} \max(\varepsilon_{P_i}(\mathbf{c}_l), \varepsilon_{P'_j}(\mathbf{c}_l)). \tag{40}$$

For a corresponding pair $(L_i, L'_j)$ of lines in $S_l$, the opposite sides along the lines are considered separately since it is possible for them to belong to two differently moving objects. The $\varepsilon_{l,L_i}(\mathbf{c}_l)$ and $\varepsilon_{r,L_i}(\mathbf{c}_l)$ are defined in the same way as in Eq. (37), while the summations are done over $N_{l,L_i}$ and $N_{r,L_i}$, respectively, instead of $R_i$, where $N_{l,L_i}$ and $N_{r,L_i}$ are defined by two parallelograms (of size 3 × *length of* $L_i$) located in opposite sides along the line $L_i$. The $\varepsilon_{l,L'_j}(\mathbf{c}_l)$ and $\varepsilon_{r,L'_j}(\mathbf{c}_l)$ are defined similarly for $L'_j$. Then, a simple correlation error is defined as

$$\begin{aligned}\varepsilon_{L,ij}(\mathbf{c}_l) \overset{\text{def}}{=} \max(&\min(\varepsilon_{l,L_i}(\mathbf{c}_l), \varepsilon_{r,L_i}(\mathbf{c}_l)),\\ &\min(\varepsilon_{l,L'_j}(\mathbf{c}_l), \varepsilon_{r,L'_j}(\mathbf{c}_l))).\end{aligned} \tag{41}$$

For the noninteger values of $(d_x, d_y)$ in the computation of correlation errors defined above, the bilinear interpolation is used. Note that $\varepsilon_{R,ij}(\mathbf{c}_l) = 0$ is a necessary condition for two regions to be matched. The same argument holds for points and lines. This verification is necessary to remove false matches even though its number was very small in our experiments.

## 4. ALGORITHM

This section describes our algorithm. The algorithm uses points, regions, and lines which are extracted independently in each frame. It groups features based on the similarity of six affine parameters by exhaustive search. It also

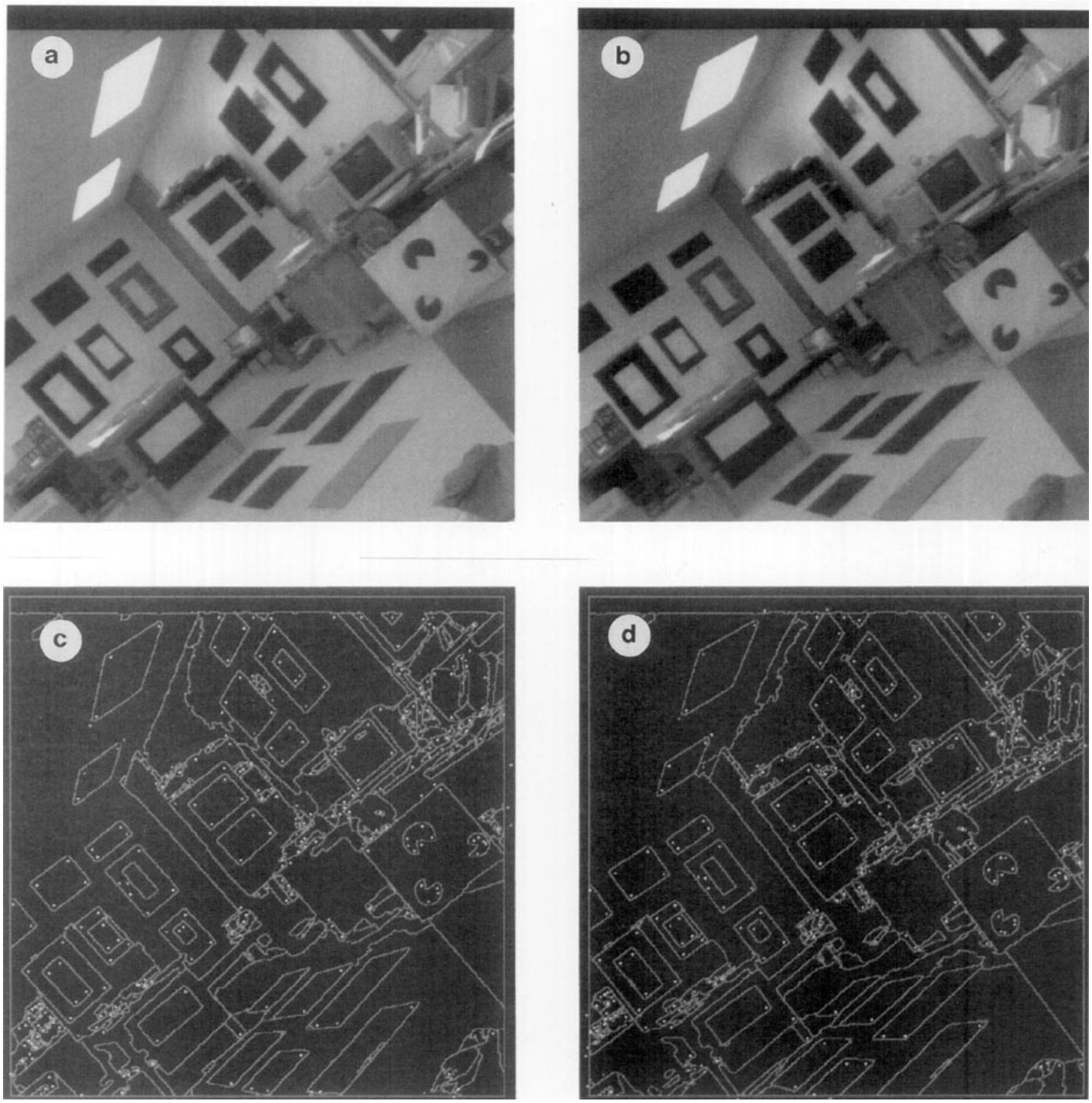

FIG. 5. Experiment 1, indoor images (a) First image $I_1$. (b) Second image $I_2$. (c) Extracted points and regions in $I_1$. (d) Extracted points and regions in $I_2$. (e) Extracted lines and line points in $I_1$. (f) Extracted lines and line points in $I_2$. (g) Matched features in $I_1$. (h) Matched features in $I_2$.

establishes correspondences between points, regions, and lines in two images.

If two lines are not on the same plane in 3D, the intersection of their projected 2D lines can generate a spurious line point in the image plane. However, the spurious point is unlikely to have a corresponding point which satisfies the local affine approximation. Line points are redundant since they are determined by the detected lines, however, they are very useful due to the following two advantages. Since many man-made objects consist of planar surfaces, the intersection of two lines often provides a good feature (for example, see Experiment 1). Another advantage of having line points is that it allows us to use the information from lines in the decomposed 3D parameter spaces.

The algorithm consists of seven steps in which connected groups (clusters) are obtained by grouping image features (points, regions, and lines) in each frame. The largest connected group is defined as a connected group which has

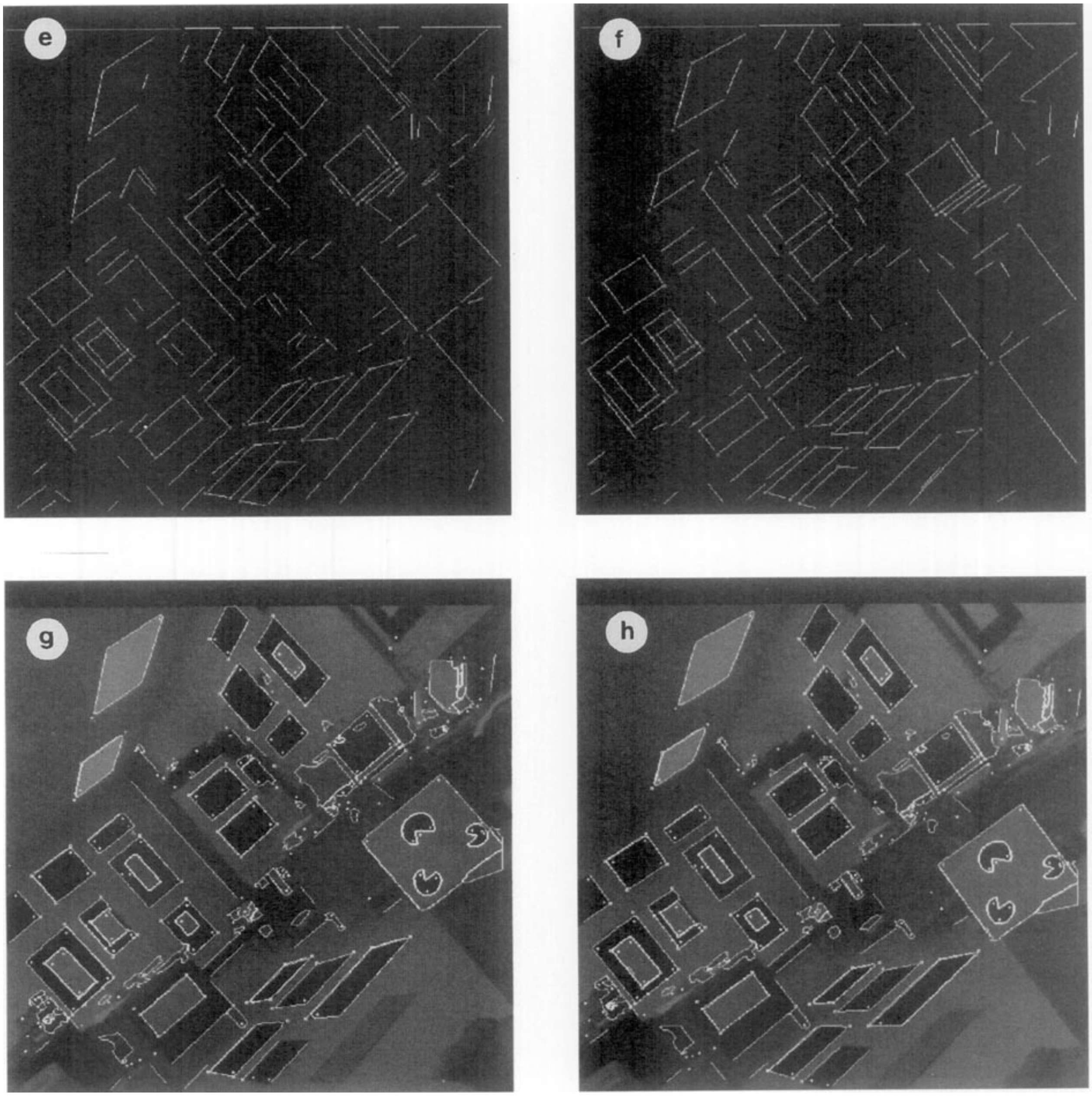

FIG. 5—*Continued*

the largest number of features. Each connected group at $t_1$ has the property that the minimum distance between any feature and the others in the group is not larger than a given threshold $T_{sd}$.

The distance between a point and a line is defined as the perpendicular distance if the perpendicular projection of the point onto the line is on the line. Otherwise, it is defined as the smaller of the distances between the point and the two end points of the line. The distance between one line and another line is defined as the minimum distance an end point is from the other line. If two lines intersect, the distance between them is defined as zero. For simplicity, the location of a region is represented as its centroid, defined in Eqs. (22) and (23).

We now describe these steps.

### Matching and Segmentation Algorithm

*Step* 1: *Initialization*

1.1. *Make Lists of Neighboring Points, Regions, and Lines.* For each region at $t_1$, make a list of the neighboring regions at $t_2$ within a distance $T_{td}$ and having similar values

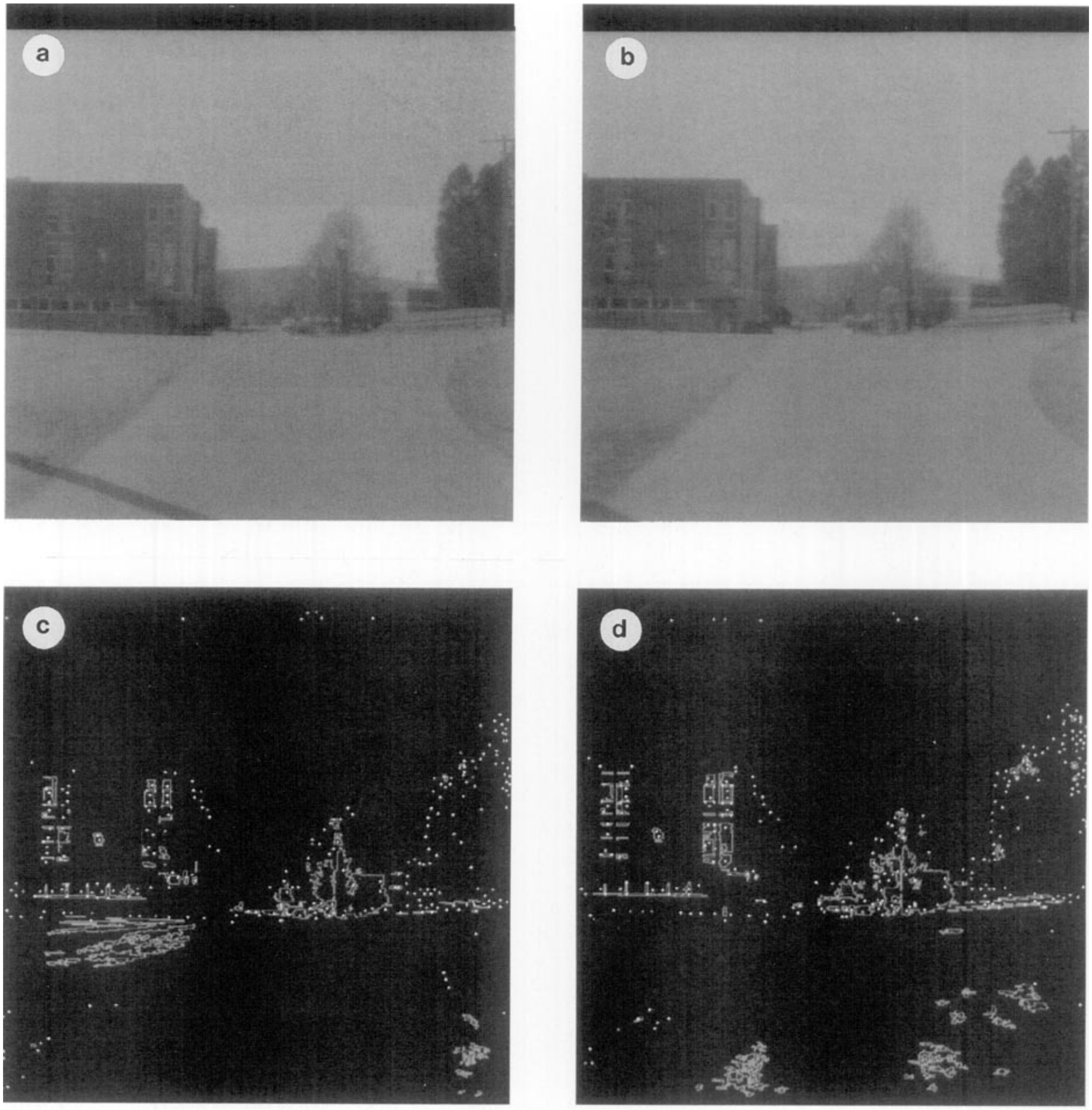

FIG. 6. Experiment 2, outdoor images. (a) First image $I_1$. (b) Second image $I_2$. (c) Extracted points and regions in $I_1$. (d) Extracted points and regions in $I_2$. (e) Extracted lines and line points in $I_1$. (f) Extracted lines and line points in $I_2$. (g) Matched features in $I_1$. (h) Matched features in $I_2$.

of 2D attributes, such as average intensity value, area, and aspect ratio.

For each feature point $P$ at $t_1$, make a list of the neighboring points at $t_2$ within a distance $T_{td}$ and having similar values of 2D attributes such as correlation values. If a point $P$ is inside a region $R$ at $t_1$ and a neighboring point $P'$ is inside a region $R'$ at $t_2$, then $P'$ is marked as a neighbor of $P$ only if $R'$ is a neighbor of $R$.

For each line at $t_1$, make a list of the neighboring lines at $t_2$ within a distance $T_{td}$ and having similar values of 2D attributes, orientation, length, and average intensity values, for example.

1.2. *Make Line Points.* For each pair $(L_i, L_j)$ of lines at $t_1$ which have at least one neighbor at $t_2$, generate a line point if the intersection of $L_i$ and $L_j$ is near $L_i$ and $L_j$ (see Fig. 2 for a definition of *nearness*).

For each pair $(L_i', L_j')$ of lines at $t_2$ which have at least one neighbor at $t_1$, generate a line point if the intersection of $L_i'$ and $L_j'$ is near $L_i'$ and $L_j'$.

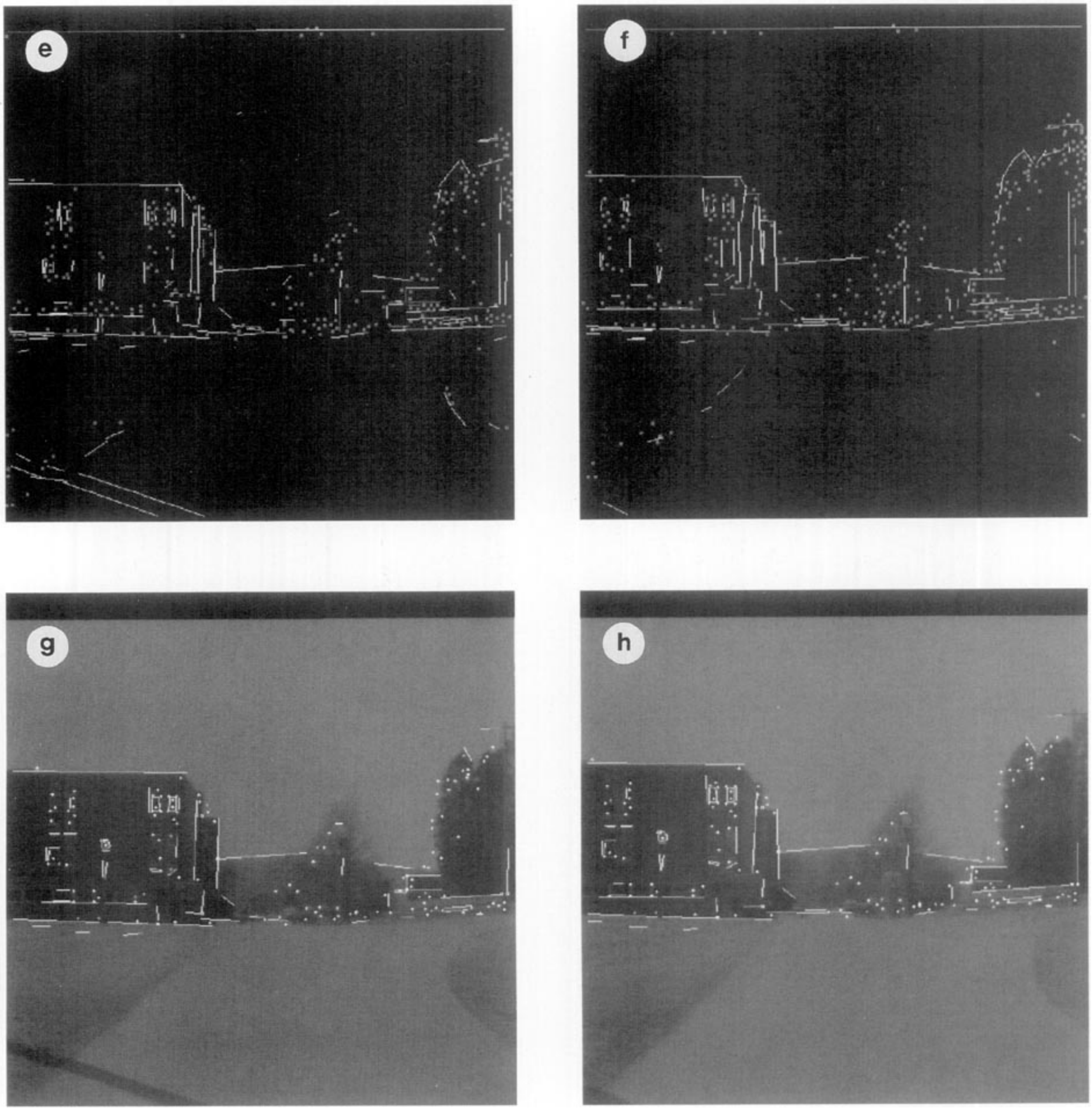

FIG. 6—*Continued*

For each line point I at $t_1$, make a list of the neighboring line points at $t_2$ within a distance $T_{ld}$ if the two lines that generated a line point at $t_2$ are neighbors of the two lines that generated the line point I at $t_1$.

*Step* 2: *Make an Initial Largest Connected Group S*

Let $T$ be the set which consists of those points, regions, and lines at $t_1$ which have at least one neighbor obtained in Step 1. Form the largest connected group $S$ from $T$.

*Step* 3: *For the Largest Remaining Connected Group S, Search for the Set of Six Parameters $\hat{c}$ Yielding the Maximum Value of F*

Multiple resolutions are used for coarse-to-fine search of each of the two quantized spaces $(c_0, c_1, c_2)$ and $(c_3, c_4, c_5)$, as shown in Fig. 3. Start with a 3D voxel in each space the size of which is determined by the initial range of values for $(c_0, c_1, c_2)$ and $(c_3, c_4, c_5)$, respectively. We call this initial voxel in each parameter space a *good* voxel.

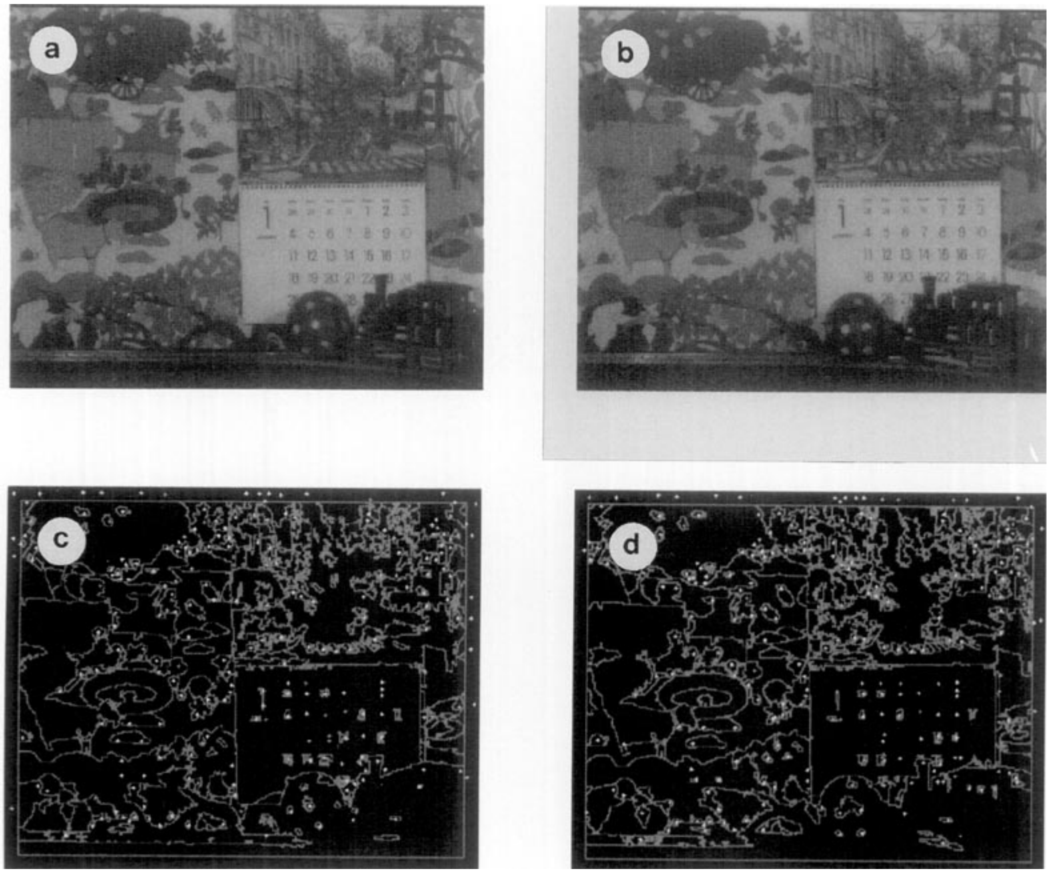

FIG. 7. Experiment 3, mobile images. (a) First image $I_1$. (b) Second image $I_2$. (c) Extracted points and regions in $I_1$. (d) Extracted points and regions in $I_2$. (e) Extracted lines and line points in $I_1$. (f) Extracted lines and line points in $I_2$. (g) Matched features in $I_1$. (h) Matched features in $I_2$.

3.1. Divide each good voxel in the space of $(c_0, c_1, c_2)$ by $8 \times 8 \times 8$. Each good voxel in the space of $(c_3, c_4, c_5)$ is also quantized by $8 \times 8 \times 8$. Among the parameter triples corresponding to centroids of the finer quantized voxels in the space of $(c_0, c_1, c_2)$ (or $(c_3, c_4, c_5)$), find a set of the parameter triples $\mathbf{C}_x =^{\text{def}} \{\mathbf{c}_{x,i} =^{\text{def}} \{c_{0i}, c_{1i}, c_{2i}\}$: $i = 1, \ldots, N_3\}$ (or $\mathbf{C}_y =^{\text{def}} \{\mathbf{c}_{y,i} =^{\text{def}} \{c_{3j}, c_{4j}, c_{5j}\}$: $j = 1, \ldots, N_3\}$) which yield the largest $N_3$ support values of $F_x(\mathbf{c}_{x,i})$ in Eq. (33) or $F_y(\mathbf{c}_{y,j})$ in Eq. (34)). (Remember that all pairs of each feature in $S$ and its neighbors at $t_2$ are used when the values of $F_x$ and $F_y$ are computed.)

3.2. If the resolution is the finest, go to Step 3.4. Otherwise, select good $\mathbf{c}_{x,i}$ and $\mathbf{c}_{y,j}$ from $\mathbf{C}_x$ and $\mathbf{C}_y$, respectively, which give the best $N_6$ combinations of solution triples corresponding to the largest $\text{N}_6$ values of $F(\mathbf{c}_{x,i}, \mathbf{c}_{y,j})$ defined in Eq. (31) from the set of the total $N_3^2$ combinations, $\mathbf{C}_{xy} = \mathbf{C}_x \times \mathbf{C}_y = \{(c_{0i}, c_{1i}, c_{2i}, c_{3j}, c_{4j}, c_{5j})$: $i = 1, \ldots, N_3, j = 1, \ldots N_3\}$.

3.3. A voxel corresponding to each good candidate triple is called a good voxel. Go to Step 3.1.

3.4. Choose $\hat{\mathbf{c}} = \{c_{0i}, c_{1i}, c_{2i}, c_{3j}, c_{4j}, c_{5j}\}$, the combination of the solution triples, which yields the maximum of $F(\mathbf{c}_{x,i}, \mathbf{c}_{y,j})$ $(i = 1, \ldots, N_3, j = 1, \ldots, N_3)$. If there is no dominant peak in the search space (i.e., the maximum value of $F$ is less than $\varepsilon_F$), go to Step 6.

*Step 4: Find a New Segment $S_l$ which Consists of All the Largest Connected Groups from $S$ and Establish Correspondences Based on $\hat{c}$*

Let $\tilde{S}$ to be an empty set. The values of $(\hat{c}_0, \ldots, \hat{c}_5)$ are given by $\hat{\mathbf{c}}$ obtained in Step 3.

For each $P_i$ in $S$, select its neighboring point $P_j'$ at $t_2$, which yields the minimum value of $\delta_{P,ij}(\hat{\mathbf{c}})$ in Eq. (16) among its neighboring points. If $\delta_{P,ij}(\hat{\mathbf{c}}) < \varepsilon_p$, we call this pair $(P_i, P_j')$ as matched points and include $P_i$ in $\tilde{S}$.

For each $R_i$ in $S$, consider only those neighboring regions $R_j'$ at $t_2$ which satisfy the following constraint on area:

$$\left| \frac{M_{00,j}'}{M_{00,i}} - |(1 + \widehat{c_1})(1 + \widehat{c_5}) - \widehat{c_2}\widehat{c_4}| \right| < \varepsilon_J. \qquad (42)$$

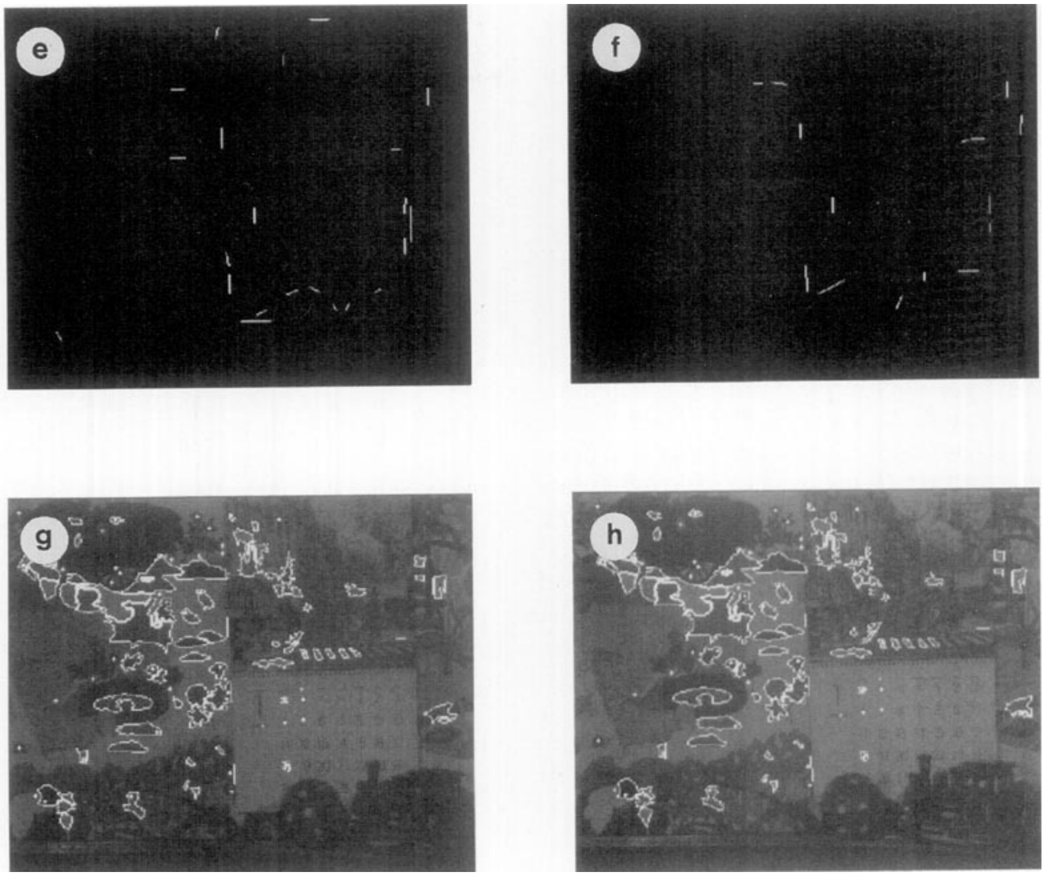

FIG. 7—*Continued*

Then, among those neighbors, select the $R_j'$ which yields the minimum value of $\delta_{R,ij}(\hat{\mathbf{c}})$ in Eq. (26). If $\delta_{R,i}(\hat{\mathbf{c}}) < \varepsilon_p$, we call this pair $(R_i, R_j')$ matched regions and include $R_i$ in $\tilde{S}$.

For each $L_i$ in $S$, consider only those neighboring lines $L_j'$ at $t_2$ which satisfy the following constraint on the longitudinal location: $\delta_{P,ij}$ in Eq. (16) of two center points of $L_i$ and $L_j'$ should be less than 0.4 × length of $L_i$. Then, among those neighbors, select the $L_j'$ which yields the minimum value of $\delta_{L,ij}(\hat{\mathbf{c}})$ in Eq. (30) among its neighboring lines. If $\delta_{L,ij}(\hat{\mathbf{c}}) < \varepsilon_p$, we call this pair $(L_i, L_j')$ matched lines and include $L_i$ in $\tilde{S}$.

Find all connected groups, each of which contains at least three features from $\tilde{S}$. A set of these groups defines one resulting segment $S_l$.

*Step 5: Find the New Largest Connected Group S from T*

Delete $S_l$ from $T$ and find the new largest connected group $S$ from $T$. Go to Step 3.

*Step 6: Verify the Matching and Segmentation*

Let $S_l$ ($l = 1, \ldots, N_S$) be segments of matched features obtained in the previous steps. For each segment $S_l$, compute $\mathbf{c}_l$ linearly (using the method presented in Section 3.5). Then, remove each point pair $(P_i, P_j')$ from $S_l$ if $\varepsilon_{P,ij}(\mathbf{c}_l) > \varepsilon_{\text{cor}}$. Similarly, remove each region pair $(R_i, R_j')$ from $S_l$ if $\varepsilon_{R,ij}(\mathbf{c}_l) > \varepsilon_{\text{cor}}$, and remove each line pair $(L_i, L_j')$ if $\varepsilon_{L,ij}(\mathbf{c}_l) > \varepsilon_{\text{cor}}$.

*Step 7: Merge the Segments*

Let $\mathbf{c}_{ij}$ be the linearly computed affine parameters corresponding to the segment $S_i \cup S_j$. Merge each pair of segments $S_i$ and $S_j$ recursively into one if $\overline{\delta}_{S_i}(\mathbf{c}_{ij}) < \varepsilon_{mg}$, $\overline{\delta}_{S_j}(\mathbf{c}_{ij}) < \varepsilon_{mg}$, and $\overline{\delta}_{S_i \cup S_j}(\mathbf{c}_{ij}) < \varepsilon_{mg}$.

## 5. IMPLEMENTATION DETAILS AND EXPERIMENTAL RESULTS

We have applied our algorithm to a large variety of pairs of images with satisfactory results. Four experimental results are presented here. We visually check the experimental results by watching alternating frames of matched features, which is an effective test.

### 5.1. *Implementation*

We use a feature point detector which locates local maxima and minima of intensity values described in [18]. Regions are extracted using a method described in [19]. (A region is defined as a connected set of pixels having intensity values which are similar to those of its surroundings.) Those regions intersecting the boundary of the image plane are removed. We detect lines using a modified version of the method described in [6].

The values of $T_{sd}$ and $T_{td}$ in Steps 1 and 2 of our algorithm are set to 50 and 64 pixels, respectively. This implies that for each feature at $t_1$ we consider only those features at $t_1$ and $t_2$ which are within 50 and 64 pixels, respectively. The value of $T_l$ used in generating the line points is 10 pixels. In Step 1.1 of our algorithm to make lists of neighboring features within a distance $T_{td}$, two regions are said to have similar values of 2D attributes if (i) the difference of average grey values is less than 15, (ii) the area ratio is between 0.7 and 1/0.7, and (iii) the aspect ratio is between 0.49 and 1/0.49. Two points are said to have similar values of 2D attributes if the average difference of the grey values of the corresponding pixels in 7 × 7 windows centered at the two points is less than 15. Two lines are said to have similar values of 2D attributes if (i) the difference in orientation is less than 30 degrees, (ii) the difference in length divided by length of the longer line is less than 0.4, and (iii) the difference in average intensity around two lines (see [6] for detailed definition) is less than 20.

The values of row and column are used to represent the image coordinates since an affine transformation remains affine after a linear transformation. The initial range of each of the two parameters $c_0$ and $c_3$ is taken to be from $-T_{td}$ pixels to $T_{td}$ pixels. The initial value of each of the four parameters $c_1$, $c_2$, $c_4$, and $c_5$ ranges from $-1$ to 1. Three resolutions are used for the coarse-to-fine search of each of the two quantized spaces $(c_0, c_1, c_2)$ and $(c_3, c_4, c_5)$. At each resolution, a voxel in each space is quantized by $8 \times 8 \times 8$. Let $\Delta c_i^{(k)}$ denote the quantization error of $c_i$ at the $k$th resolution where the first and third resolutions represent the coarsest and finest resolutions, respectively. Then, the values of $\Delta c_0^{(k)}$ and $\Delta c_3^{(k)}$ are less than $1/2(2T_{td}/8^k)$, which is equal to 8, 1, and 0.125 at $k = 1$, 2, and 3, respectively. The values of $\Delta c_1^{(k)}$, $\Delta c_2^{(k)}$, $\Delta c_4^{(k)}$, and $\Delta c_5^{(k)}$ are less than $1/2(2 \times 1/8^k)$, which equals 0.125, 0.016, and 0.002 at $k = 1$, 2, and 3, respectively. Uncertainty of $d_x$ and $d_y$ at the $k$th resolution due to quantization can be estimated by

$$\Delta d_x^{(k)} = |\Delta c_0^{(k)}| + |\Delta c_1^{(k)} x| + |\Delta c_2^{(k)} y| \tag{43}$$

$$\Delta d_y^{(k)} = |\Delta c_3^{(k)}| + |\Delta c_4^{(k)} x| + |\Delta c_5^{(k)} y|. \tag{44}$$

For example, for a pixel at $x = y = 100$, $d_x^{(1)}$ (or $d_y^{(1)}$), $d_x^{(2)}$ (or $d_y^{(2)}$), and $d_x^{(3)}$ (or $d_y^{(3)}$) are equal to 33, 4.125, and 0.516 pixels. Therefore, to lower the uncertainty, it is important to reduce the range of values for $x$ and $y$. In our implementation of Step 3 in the algorithm, a center of mass for the largest remaining component $S$ is computed and it is used as a new origin of the image coordinates. We tried our algorithm with or without using the dynamic origin and obtained better results with the dynamic origin. Let $\varepsilon_p^{(k)}$ represent the threshold value in Eq. (32) at the $k$th resolution. The value of $\varepsilon_p^{(3)}$ is set to 0.75 pixels, implying that the image errors up to $0.5 \times \sqrt{2}$ pixels are tolerated. At the first and second resolutions, they are set to the values $\Delta c_0^{(1)}$ (or $\Delta c_3^{(1)}$) and $\Delta c_0^{(2)}$ (or $\Delta c_3^{(2)}$), respectively. Considering the fact that the affine transformations are an approximation to the displacement field, the value 0.75 is conservative.

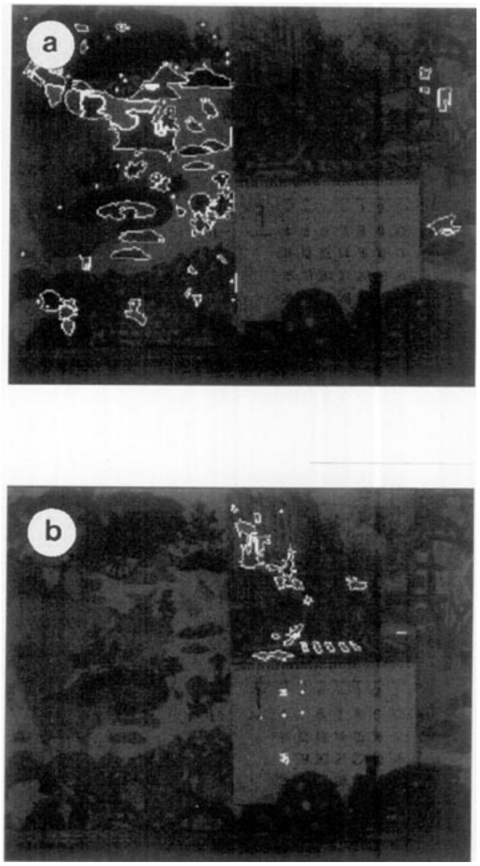

FIG. 8. Two segments found in Experiment 3. (a) Segment 1 on the wall in $I_1$. (b) Segment 2 on the calendar in $I_1$.

The values of $N_3$ and $N_6$ are experimentally set to 15 and 30, respectively, which are adequate to give all dominant local maxima. All the values of $w_{P,ij}$, $w_{R,ij}$, and $w_{L,ij}$, are set to 1, for simplicity.

The value of $\varepsilon_F$ in Step 3.4 is 4.5, making each segment contain more than three features. In Step 4, the value of $\varepsilon_J$ is 0.2. The value of $\varepsilon_{\rm cor}$ for Step 6 is 5. The maximum tolerable average image error $\varepsilon_{mg}$ for merging in Step 7 is set to be equal to $\varepsilon_p^{(3)}$. The merged segments which contain less than five features are removed. Note that the distance

TABLE 1
Number of Correspondences Found for Each Feature Type in Experiment 1

| Feature type | $N_1$ | $N_2$ | $M$ |
|---|---|---|---|
| Region | 320 | 310 | 74 |
| Feature point | 302 | 302 | 147 |
| Line | 183 | 178 | 85 |
| Line point | 115 | 124 | 79 |

*Note.* $N_1$, number of extracted features at $t_1$; $N_2$, number of extracted features at $t_2$; $M$, number of matched features.

TABLE 3
Number of Correspondences Found for Each Feature Type in Experiment 3

| Feature type | $N_1$ | $N_2$ | $M$ |
|---|---|---|---|
| Region | 369 | 357 | 71 |
| Feature point | 244 | 255 | 56 |
| Line | 23 | 15 | 5 |
| Line point | 0 | 0 | 0 |

*Note.* $N_1$, number of extracted features at $t_1$; $N_2$, number of extracted features at $t_2$; $M$, number of matched features.

between two segments is not considered when we merge them.

The $\varepsilon_p^{(k)}$ is varied to evaluate the sensitivity of results. Results are more sensitive to $\varepsilon_p^{(3)}$ than to $\varepsilon_p^{(1)}$ and $\varepsilon_p^{(2)}$. Figure 4 shows the average image error (Section 3.5), the average correlation error (Section 3.6), and numbers of matched features and segments for the two images in Experiment 4. As the value of $\varepsilon_p^{(3)}$ is increased, the relative number of false matches increases, thus increasing the average image error and the average correlation error (Fig. 4), particularly for points. Stationary feature points in the background in the first image of Experiment 4 begin to be matched with an error of one pixel, for example. Regions and lines are less sensitive to this mismatching since they are more global than points. Figure 4d shows that the segmentation capability also degrades as the value of $\varepsilon_p^{(3)}$ is increased. For the lower values of $\varepsilon_p^{(3)}$, the numbers of matched features and segments are too small. Therefore, the value of $\varepsilon_p^{(3)}$ which is equal to 0.75 is a reasonable choice.

### 5.2. *Experiment* 1

Two image frames of indoor scenes, of size 512 by 512, are used. These are the second and third frames obtained from the image database of the 1991 IEEE Workshop on Visual Motion.

Table 1 shows the number of extracted features and correspondences found for each feature type. We see that a fairly good number of correspondences are obtained. Figure 5 shows the images, detected points, regions, lines, and line points. The matching results are also presented and appear to be good. Note that line points serve as useful features. We obtain six segments, each of which corresponds to a part of the scene having similar depth.

### 5.3. *Experiment* 2

Two image frames of outdoor scenes, of size 512 by 512, are used. These images are the fourth and fifth frames, which were also obtained from the database of the 1991 IEEE Workshop on Visual Motion.

Table 2 shows the number of extracted features and the correspondences found for each feature type. Figure 6 shows the images, detected points, regions, lines, and line points. The part of the image corresponding to the road does not yield good features and therefore no correspondence is obtained there. Overall, the matching results are satisfactory. We obtain seven segments, each of which contains features, the motion of which is described by a distinct set of six parameters.

### 5.4. *Experiment* 3

Two frames of size 288 by 360 are used. These are the first and fifth frames obtained from a standard image sequence for testing MPEG performance.

Table 3 shows the number of extracted features and correspondences found for each feature type. Figure 7 shows the images, detected points, regions, lines, line

TABLE 2
Number of Correspondences Found for Each Feature Type in Experiment 2

| Feature type | $N_1$ | $N_2$ | $M$ |
|---|---|---|---|
| Region | 88 | 123 | 13 |
| Feature point | 301 | 302 | 128 |
| Line | 106 | 91 | 45 |
| Line point | 15 | 16 | 9 |

*Note.* $N_1$, number of extracted features at $t_1$; $N_2$, number of extracted features at $t_2$; $M$, number of matched features.

TABLE 4
Number of Correspondences Found for Each Feature Type in Experiment 4

| Feature type | $N_1$ | $N_2$ | $M$ |
|---|---|---|---|
| Region | 182 | 195 | 47 |
| Feature point | 195 | 186 | 77 |
| Line | 59 | 70 | 30 |
| Line point | 16 | 16 | 8 |

*Note.* $N_1$, number of extracted features at $t_1$; $N_2$, number of extracted features at $t_2$; $M$, number of matched features.

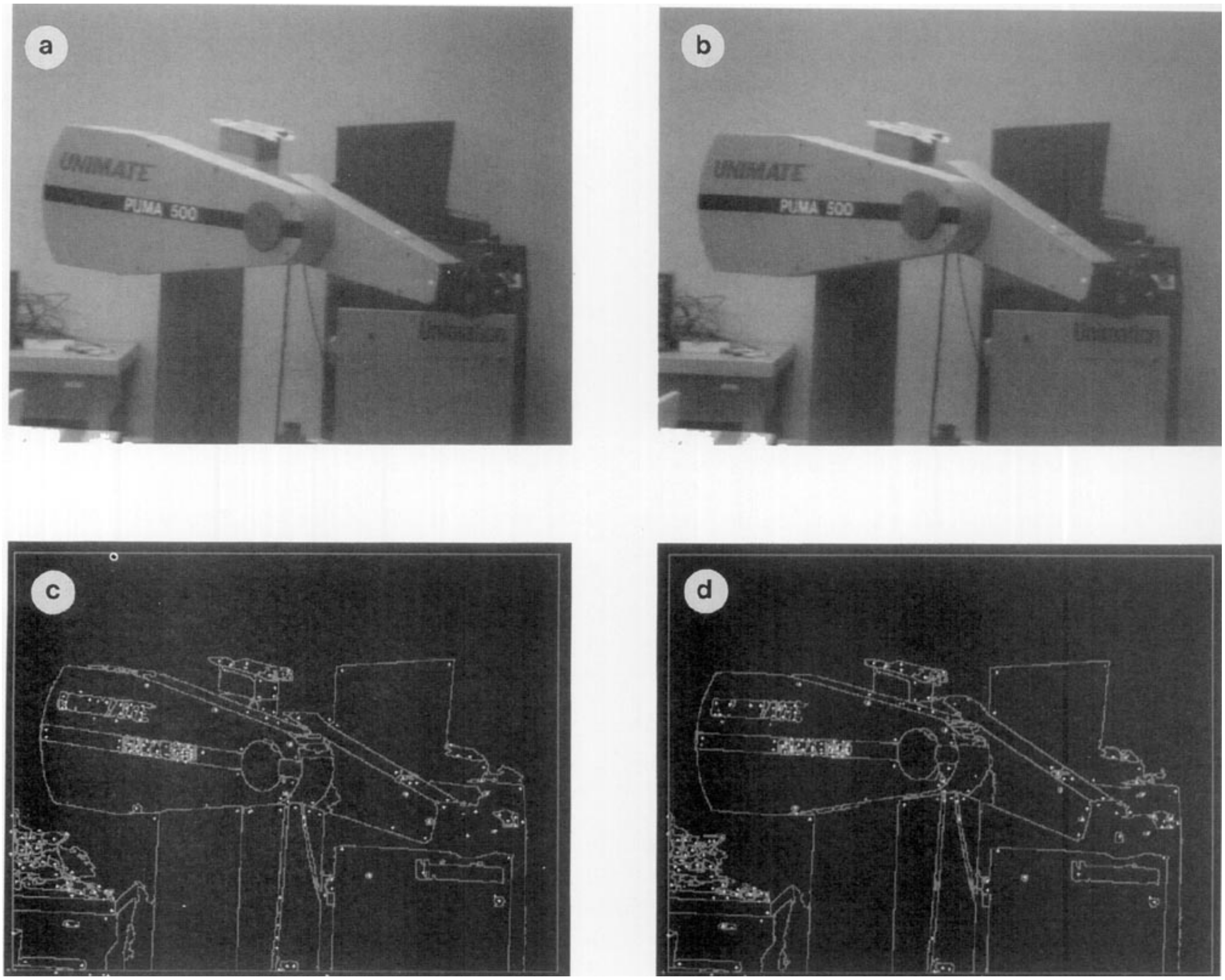

FIG. 9. Experiment 4, robot images. (a) First image $I_1$. (b) Second image $I_2$. (c) Extracted points and regions in $I_1$. (d) Extracted points and regions in $I_2$. (e) Extracted lines and line points in $I_1$. (f) Extracted lines and line points in $I_2$. (g) Matched features in $I_1$. (h) Matched features in $I_2$.

points, and matching results. Figure 8 shows two resulting segments corresponding to the wall and the calendar. The features corresponding to the moving train are not segmented out since feature detectors could not extract good features from the parts of images corresponding to the moving train. The matching and segmentation results are satisfactory, and this example demonstrates the feasibility of our approach for the segmentation of a scene into independently moving objects.

### 5.5. *Experiment* 4

Two real images of a PUMA 500 are used. The field of view of the camera is approximately 13°. The image size is 384 by 500.

Table 4 shows the number of extracted features and correspondences found for each feature type. Figure 9 shows the images, detected points, regions, lines, line points, and matching results. The line detector used could be improved to get better matching results. Figure 10 shows four resulting segments. Segments 1 and 3 correspond to the stationary background and the small arm, respectively. Segments 2 and 4 are the part of the large arm, although these two segments are not merged into one segment using the value of merging threshold $\varepsilon_{mg}$ equal to 0.75 (the value of $\overline{\delta}_{S_4}(\mathbf{c}_{ij})$ for $i = 2$ and $j = 4$ was 1.32). Segment 3 contains a small region which should be segmented into the background. This is because extracted region boundaries are not perfect and the coefficients of affine transformation obtained from coarse-to-fine search processes are quantized although the uncertainty at the finest resolution is small. This example also demonstrates the feasibility of our approach for the segmentation of a scene.

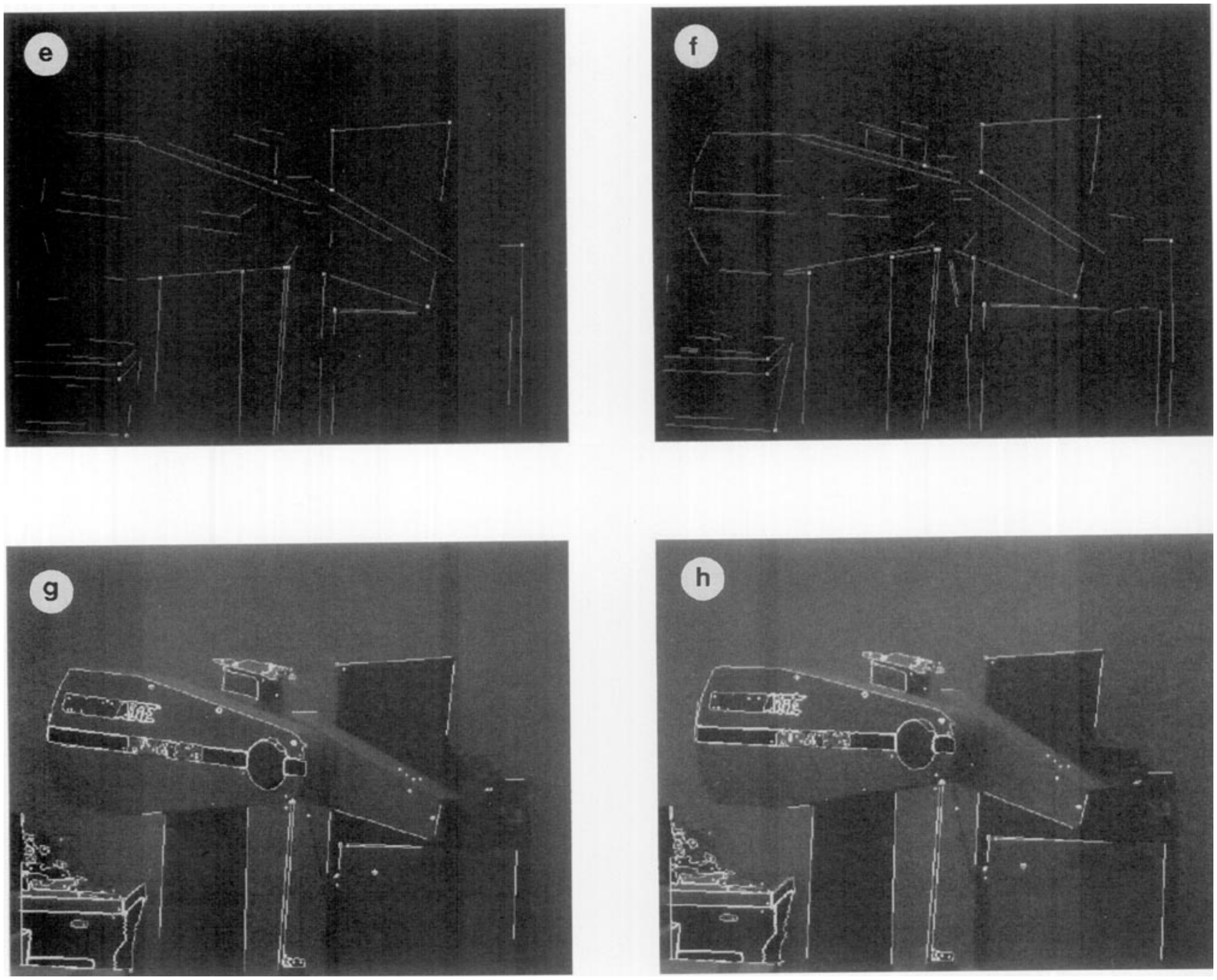

FIG. 9—*Continued*

## 6. CONCLUSIONS AND EXTENSIONS

We have described an integrated method which matches and simultaneously segments multiple features such as points, regions, and lines from two perspective images. We have also presented the results of four experiments to demonstrate our algorithm.

It is well known that region boundary and line end points are unstable. This problem can be partly solved by considering only centroids for regions and perpendicular distance for lines, as seen from the experimental results. If regions and lines fragment or merge due to noise in the projection process, occlusion, etc., they will not match in our current implementation.

In our experiments, the algorithm presented gives robust results for matching and segmentation although segmentation is harder than matching.

Although feature points were defined as local maxima and minima of intensity values in this paper, other types of point features could be used. If more than one attribute (for example, intensity maximum and optical flow vector) is available at the same pixel, the displacement vector at that pixel is determined by selecting the one which best satisfies the affine transformations. By considering more than one type of feature point, we can use more information from images, resulting in better performance at the expense of increased computation.

We plan to extend our method to a sequence of images. Parallel implementation of the algorithm is also planned.

## ACKNOWLEDGMENT

We thank the anonymous reviewers for their valuable suggestions. Also, we thank Yuncai Liu, Srikanth Thirumalai, and Mark Tabb for letting us use their programs for feature detection.

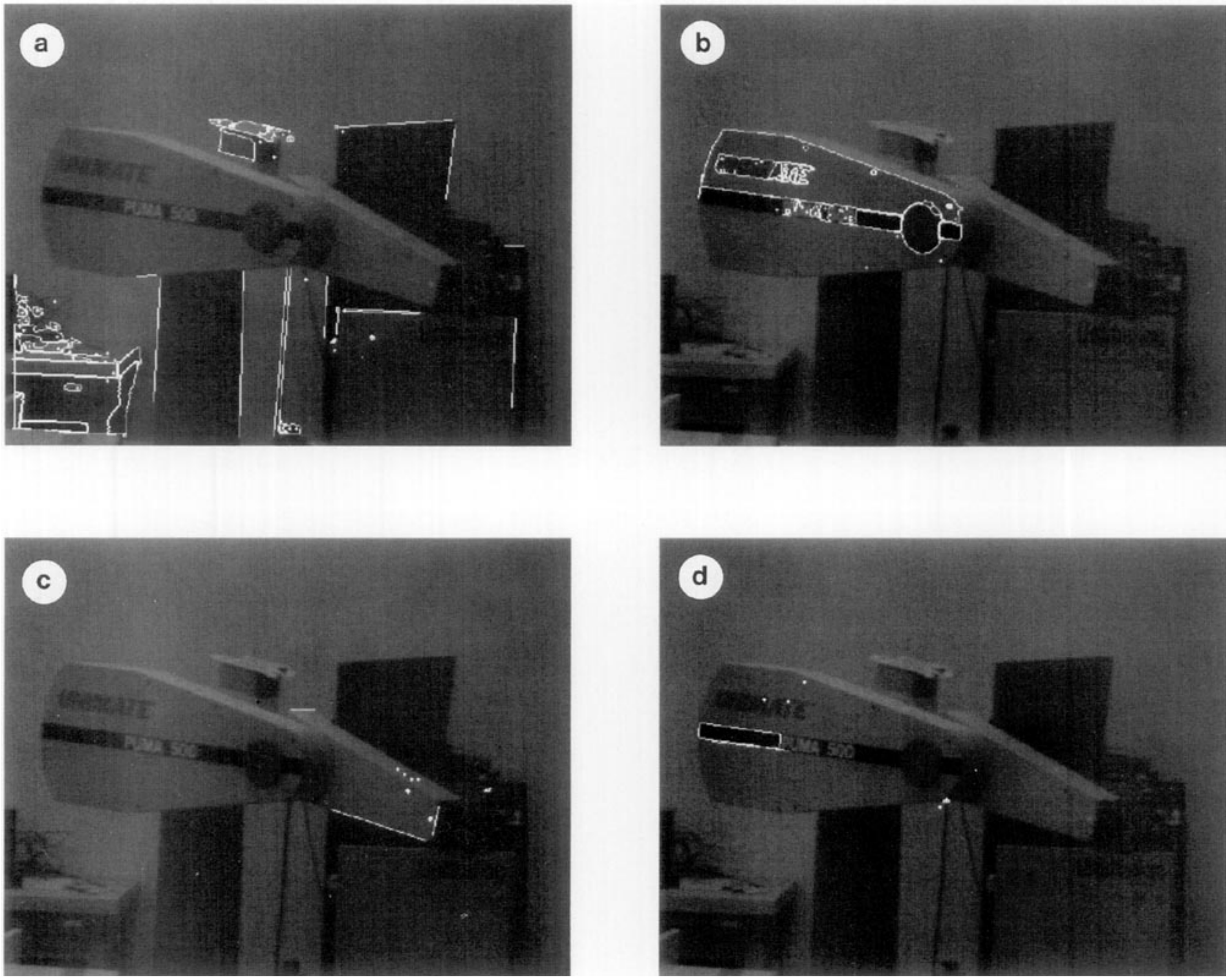

FIG. 10. Four segments found in Experiment 4 (a) Segment 1 in $I_1$. (b) Segment 2 in $I_1$. (c) Segment 3 in $I_1$. (d) Segment 4 in $I_1$.

## REFERENCES

1. B. Horn and B. Schunck, Determining optical flow, *Artif. Intell.* **17,** 1981, 185–203.
2. M. Black and P. Anandan, A model for the detection of motion over time, in *Proceedings International Conference on Computer Vision, Osaka, Japan, 1990,* pp. 33–37.
3. D. F. J. L. Baron, Performance of optical flow technique, in *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition, Champaign, IL, June 1992,* pp. 236–242.
4. G. Medioni and R. Nevatia, Matching images using linear features, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-6,** Nov. 1984, 675–686.
5. J. McIntosh and K. Mutch, Matching straight lines, *Comput. Vision Graphics Image Process.* **43,** 1988, 386–408.
6. Y. Liu and T. Huang, Determining straight line correspondences from intensity images, *Pattern Recognit.,* **24,** 1991, 489–504.
7. I. Sethi and R. Jain, Finding trajectories of feature points in a monocular image sequence, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-9,** 1987, 56–73.
8. P. S. J. Crowley and C. Discours, Measuring image flow by tracking edge-lines, in *Proc. Int. Conf. Comput. Vision,* 1988, 658–664.
9. R. Deriche and O. Faugeras, Tracking line segments, in *Proc. European Conf. Comput. Vision,* 1990, 259–268.
10. V. Venkateswar and R. Chellappa, Hierarchical feature based matching for motion correspondence, in *Proc. IEEE Workshop on Visual Motion,* Oct. 1991, 280–285.
11. N. A. J. Weng and T. S. Huang, Matching two perspective views, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-14,** Aug. 1992, 806–825.
12. H. Sawhney and A. Hanson, Identification and 3d description of shallow environmental structure in a sequence of images, in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.* 1991, 179–185.
13. S. Sull and N. Ahuja, Segmentation, matching and estimation of structure and motion of textured piecewise planar surfaces, in *Proceedings, IEEE Workshop on Visual Motion, Princeton, NJ, Oct. 1991,* pp. 274–279.
14. G. Adiv, Determining three-dimensional motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-7,** July 1985.
15. S. Geman and D. Geman, Stochastic relaxation, Gibbs distribution,

and the Bayesian restoration of images, *IEEE Trans. Pattern Anal. Machine Intell.* **PAMI-6,** Nov. 1984, 721–741.

16. T. Darrell and A. Pentland, Robust estimation of a multi-layered motion representation, *Proc. IEEE Workshop on Visual Motion,* Oct. 1991, 173–178.

17. M. R. Spiegel, *Advanced Calculus,* McGraw-Hill, New York, 1963.

18. C. Debrunner and N. Ahuja, A hankel matrix based on motion estimation algorithm, in *Proceedings, International Conference on Pattern Recognition, Atlantic City, June 1990,* pp. 384–389.

19. M. Tabb and N. Ahuja, Detection and representation of multiscale low-level image structure using a new transform, in *Proceedings, Asian Conference on Compuer Vision, Osaka, Japan, 1993,* pp. 155–159.