COMPUTER VISION AND IMAGE UNDERSTANDING
Vol. 66, No. 3, June, pp. 271–285, 1997
ARTICLE NO. IV960512

Matching Feature Points in Image Sequences through a Region-Based Method

Filiberto Pla

Department of Computer Science, Universitat Jaume I, Campus Penyeta Roja, 12071 Castello, Spain

and

John A. Marchant

Silsoe Research Institute, Wrest Park, Silsoe, Bedfordshire MK45 4HS, United Kingdom

Received October 6, 1994; revised February 5, 1996

In order to achieve reliable matching in image sequences, a hierarchical approach is proposed. First, matching is established between regions of consecutive segmented images. In a second step, feature point matching between features of matched regions is performed. Regions provide a structural representation for features in the same region. To establish matching between features of two corresponding regions, a relational graph is built. Matching features is based on three principles: exclusion, proximity, and rigidity. The use of subgraph matching techniques through maximal clique detection provides a method to cope with partial occlusion and missing features between frames. © 1997 Academic Press

INTRODUCTION

Computing correspondences between image sequences is a key step in several computer vision problems, such as detecting moving objects in a scene, or calculating motion parameters of the camera with respect to the objects in the scene, and thus recovering structure of the objects. An important application of the structure from motion problem is its use in mobile robots or autonomous vehicles. The work reported here has been motivated by calculation of camera motion to guide an autonomous vehicle to perform crop protection tasks.

Most of the work in calculating motion assumes that a set of corresponding points between images in a sequence have been found. The problem arises when the correspondence computation step does not provide reliable matches, causing a biased estimation of motion parameters with respect to the real parameters. This mainly occurs, as in our case, when the set of corresponding points is rather small. Therefore, first of all, we have focused our work in solving the correspondence problem, in order to look for a satisfactory solution in our particular problem domain, or other similar situations.

Depending on whether the motion of objects in the scene is small or large, two main approaches for measuring visual motion can be differentiated. One is called the flow-based method, which uses local changes in light intensity to compute image flow at each image point, and then to compute the motion parameters based on the flow calculated. The other approach is the feature-based method, which consists of establishing the correspondence between features extracted from images of a sequence. After features have been matched, an algorithm computes the motion parameters using landmark points associated with each feature in each image. The method we are presenting in this work is an example of the feature-based approach.

Image features commonly used in the feature-based approach are corners [1], straight lines [2, 3], arcs [4], or free form curves [5]. One of the most exploited idea to solve the feature correspondence problem is that image appearance of an object point in successive frames should be similar. Thus, several matching algorithms use similarities between features to match them, for instance, sign of change and orientation in zero crossings [6], or orientation, contrast, and length in straight line segments [3], or similarity in first and second derivatives at corners [7].

To establish correspondence between features, the most important principle used is uniqueness or the principle of exclusion; that is, one feature can only be matched with a single feature in the other frame. In order to satisfy this constraint, several strategies have been used to look for the corresponding features, for instance, using a distance function to evaluate how similar two features are [3, 7, 8]. In most cases, after an initial matching, some procedure is used to delete inconsistent or ambiguous matches, and to propagate or extend other matches to nearby features starting from strong candidates to final matching [2, 8], or using a beam search strategy supporting several possible matches per feature, solving them in subsequent observations [9]. Other authors use registration techniques [5] to find the best and the most overall coherent matches, helped by a feedback from the calculation of the motion parameters using the matches provided at each iteration. Relaxation techniques have also been used to solve matching, updating iteratively probabilities assigned to possible matches [10]. Other methods, although using feature points, use only geometric properties to solve the matching [11].

Feature matching between frames can be seen as a matching between a model, represented by the features of one frame, and a pattern, represented by the other frame. Therefore, model-matching-based techniques could be used in this sense, like the local-feature-focus method [12] or local search matching techniques using a steepest descent strategy to search for correspondences based on the Hamming distance defined in the space of model-data correspondence [13]. In fact, the concept of a model matching technique is used in this work, using an approach similar to the local-feature-focus method, to try to cope with inexact matching, since local search matching techniques [13] have more difficulty in dealing with such a problem.

Scott and Longuet-Higgins [14] proposed an algorithm for associating features between two images based on a combination of the principle of proximity and the principle of exclusion. Shapiro and Brady [15] modified Scott and Longuet-Higgins method to cope with large rotations in the image plane. To do so, distances between features in the same image were taken instead of interimage distances to build the proximity matrix, introducing a rigidity constraint. In other words, the structural information is more important to solve large displacements and rotations. One drawback of this method is that it cannot deal with partial occlusion, or if features in the image belong to several objects, because the algorithm is based on a global shape representation of objects, providing a closed and global solution.

Our approach is based on the ideas of the two methods mentioned above. The work presented here has been aimed at developing a method to match features between two successive images in a sequence, exploiting the idea of a structural representation, using distances between features in the same image, and incorporating the principle of proximity through interimage distances.

In order to achieve a reliable matching a hierarchical approach is proposed. Instead of trying to match directly features between frames, correspondence is first established between structural representations of features, that is, between regions from frame to frame. The idea of a hierarchical approach, by first matching regions, has been used in other works [16]. However, it has not been treated as a technique to provide reliable enough matches on its own, thus avoiding further refinement techniques to overcome noisy correspondences.

Matching regions produced by segmenting two consecutive frames is more robust and reliable than trying to match feature points individually, since regions, as more structured and rich representations, are less likely to be assigned to falsely corresponding regions. However, motion from region correspondence does not provide accurate information to calculate motion between frames, since a region is a sparse set of points, and they do not provide an accurate disparity map between frames. So, after matching of regions is performed, feature point matching between features of matched regions is established, providing more accurate matched locations for feature points.

The rest of the paper is organized as follows: after a statement of the problem, the region matching method used is described. In the subsequent section, the association graph technique used to match features is described, and how this graph is built. Experimental results are presented using real image sequences. Finally, some conclusions on the present work are drawn.

PROBLEM STATEMENT

As it has been mentioned, the purpose of this work is to look for correspondence features between images of sequences taken from a moving vehicle. Once correspondences are established, we can calculate the motion that the vehicle undergoes with respect to a fixed world, and use this information to guide a crop protection vehicle along the rows of a crop, that is, along a predefined straight line trajectory in the scene.

The camera model used is the pin-hole model. The relation between 3D coordinates of a point (x, y, z), and its corresponding image coordinates (x', y') in this model is

$$x' = xf/z$$
$$y' = yf/z$$

with f the focal length of the lens.

Let X^k be the 3D coordinates of a point (x, y, z) at the instant when frame number $k, k = 1, \dots, N$, of a sequence of N images is taken, and let x^k the corresponding image plane coordinates (x', y') of that point in frame k. The motion of a point from position X^k to position X^{k+1} can be expressed in general as a rotation R^k followed by a translation t^k, that is,

$$X^{k+1} = R^k X^k + t^k,$$

where R^k is an orthogonal rotation matrix and t^k the translation vector.

We shall denote x^{k+1} as the projected point in frame $k + 1$, of the point x^k in frame k, due to motion undergone between frames.

The technique we are describing here is based on the following assumptions:

—Objects present in the scene undergo a rigid motion with respect to the camera.
—The motion changes smoothly across frames.
—Only small changes of image appearance have occurred from image to image.

The interpretation and effects of the above mentioned assumptions on our approach are explained in subsequent sections.

MATCHING REGIONS

Shapiro and Brady's work [15] showed that relations between features in a single object provide a robust way to establish correspondences between two images of the same object. These relations were distances between features in the same image, that is, some sort of rigidity constraint as a structural representation of features in a single object, but allowing small changes between these distances to cope with small variations in the global shape of objects, due to possible small image shape deformations from image to image.

Following this idea, and considering real scenes with multiple objects, an obvious way of partitioning images in different objects is using some image segmentation technique. If we segment an image according to certain criterion or method, regions resulting from segmentation can be taken as "objects." In fact, what we need is to group a set of features in a coherent framework or structure, without being too concerned whether this structure represents a real object in the world or part of it. The problem to be avoided is mixing features from different objects in a single structure.

Regions from segmented images provide this coherent framework to group features in a same relational structure. Although we do not establish a correspondence between regions and objects in the scene, we assume that each region in a segmented image is a patch of an object in the scene, because all points in a single region have common characteristics; thus, all points in a region are very likely to belong to the same object in the scene.

Therefore, let us assume that two successive images in a sequence have been segmented by a given method. If we extract some set of feature points in each region, features belonging to the same region can be related among them. The first step in solving matching is to associate each region in one image to the corresponding region in the next image of the sequence. Once regions are matched, we will match features between regions.

Matching regions in a first step is part of a hierarchical strategy to derive final matching between feature points. Regions from two consecutive frames can be matched in a more reliable and exact way than single points, because they contain more information that characterize individually each region, and because regions are represented by a sparse set of connected points in the image, allowing an easy tracking between images if image motion between frames is small.

Motion across images in a sequence is assumed to vary smoothly between frames. Therefore, to look for corresponding regions between two frames, k and $k + 1$, motion parameters from previous frames, $k - 1$ and k, are used as a first estimate of the new motion between frames being analyzed. Each region in frame k is then projected into frame $k + 1$ with these motion parameters. Once a region from frame k is projected into frame $k + 1$, its corresponding region is taken as the nearest region in frame $k + 1$ to the projected region.

Let us define the nearest region in frame $k + 1$ to a region projected from frame k. Let $x^k_{i,j}$, $i = 1, \ldots, N_j$ be the N_j points of region L^k_j in frame k, and let $x^{k+1}_{i,j}$ be the corresponding projected points in frame $k + 1$. Coordinates of point $x^{k+1}_{l,m}$ in frame $k + 1$ are associated with some region L^{k+1}_m in frame $k + 1$. Let $N_{j,m}$ be the number of points associated to region m in frame $k + 1$ projected from region j into frame k. The distance d between region L^k_j and L^{k+1}_m is defined as

$$d(L^k_j, L^{k+1}_m) = 1 - N_{jm}/\min(N_j, N_m)$$

with N_j being the area of region L^k_j, and N_m the area of region L^{k+1}_m. This measure indicates the proportion of points of region L^k_j which falls on some point of region L^{k+1}_m after their projection in frame $k + 1$ by motion parameters from the previous frames. The normalization factor $\min(N_j, N_m)$ means that this proportion is taken with respect to the region that has the minimum area from L^k_j and L^{k+1}_m; thus, distances are always between 0 and 1, and distance 0 means maximum proximity.

Therefore, the corresponding region in frame $k + 1$ of a region L^k_j in frame k is the region L^{k+1}_p in frame $k + 1$, whose distance to region L^k_j is minimum, that is,

$$L^{k+1}_p; d(L^k_j, L^{k+1}_p) = \min(L^k_j, L^{k+1}_m); m = 1, \ldots, L^{k+1}$$

with L^{k+1} the number of regions in frame $k + 1$.

A threshold D in this minimum distance is used to avoid assignments to regions in frame k which have passed out of the field of view frame $k + 1$; that is, in the above expression, if $d(L^k_j, L^{k+1}_p) > D$, it is considered that region L^k_j in frame k does not have any match in frame $k + 1$.

Note that the formulation described to assign matchings between regions considers the possibility of many to one matches; that is, a region in frame k can have several corresponding regions in frame $k + 1$, and conversely, several regions in frame k can match the same region in frame $k + 1$. This is because, although the same segmentation criterion is used in both images, a region may be split into several parts from frame to frame, or two or more regions in frame k may be merged into a single one in frame $k + 1$. To avoid these over segmentation problems in the matching process, the above formulation does not support the principle of exclusion.

The algorithm to match regions, after segmentation of frames k and $k + 1$ has been performed, assigns each region L_j^k in frame k to its corresponding region L_p^{k+1} of minimum distance in frame $k + 1$, unless it has been considered that region L_j^k has passed out of the field of view in frame $k + 1$. If several regions in frame k have been merged into a single region in frame $k + 1$, the algorithm will assign these regions to the same region in frame $k + 1$. On the other hand, if a region in frame k has been split in several regions in frame $k + 1$, only one region in frame $k + 1$ will be assigned as corresponding, that with minimum distance. To assure the possibility of many to one matches in that direction too, a second part of the procedure looks for regions in frame $k + 1$ which have not been matched to any region in frame k, and repeats the procedure in the direction from frame $k + 1$ to frame k, but only for nonmatched regions.

As a global representation of a set of points, regions can be matched even if small changes from frame to frame occur. Problems could arise when regions resulting from segmentation are small. Resulting small regions can be avoided using a region growing criterion, where regions under a threshold area are forced to be merged.

The algorithm described could be used to decide if oversegmentation is present between frames, and to identify such regions, but for our purpose this is not necessary. We are only interested in finding corresponding regions, to provide a framework to look for corresponding feature points in both regions, and thus to obtain accurate matched locations.

MATCHING FEATURE POINTS BETWEEN REGIONS

Once regions from two images have been matched, they do not provide enough information to calculate motion undergone from frame to frame. After regions have been matched, the next step is to extract some feature points in each region, and to match features between matched regions.

Let L_j^k be the region in frame k which has been matched to region L_m^{k+1} in frame $k + 1$. Let $x_{i,j}^k$, $i = 1, \ldots, F_j^k$, be the F_j^k feature points extracted from region L_j^k in frame k, and $x_{l,m}^{k+1}$, $l = 1, \ldots, F_m^{k+1}$, be the F_m^{k+1} feature points extracted from region L_m^{k+1} in frame $k + 1$. We are looking for a mapping between a subset of the $x_{i,j}^k$ feature points into a subset of the $x_{l,m}^{k+1}$ feature points, since, in general $F_j^k \neq F_m^{k+1}$, and some features may not have correspondences in the other set, due to partial occlusion or missing features between frames.

Our approach to solve this problem is based on the following three principles:

—Rigidity. As we have already pointed out, it is assumed objects undergo a rigid motion, and that an object's appearance from image to image cannot change substantially. Thus, although the image projection of an object in the scene can vary slightly from frame to frame, feature points in a single object in two consecutive images maintain their rigidity relations with possible small changes on distances among features in the image. These 2D rigidity relations in the image plane are used to establish the matching.

—Proximity. Small motion between frames leads to the principle of proximity, since features do not move far between frames. This proximity principle also holds when motion is not so small but varies smoothly. Thus, if we project a feature point from one frame to its subsequent frame using as an estimate the calculated motion of the two previous frames, its corresponding feature will be nearby. This principle establishes a relation between features across frames, and the principle of rigidity establishes a relation between features within the same frame.

—Exclusion. Because feature points are assumed as indivisible units of representation located at a point in the image, only one feature point in the next image can be assigned as a corresponding feature, establishing a one-to-one correspondence criterion.

By the principle of rigidity, distances among feature points in the image plane can be used as structural relation within the same frame. If we construct a graph for each region, where nodes represent feature points of the region, and links represent distances between features, the feature matching problem can be seen as a graph matching problem between the graph representing features of a region and the graph representing features of the corresponding region.

Association Graphs

Using graphs as relational structure to solve matching problems is a concept widely reported in the literature [17]. Graph matching techniques have been used as global structural representation in stereo images [18], object recognition and location [12], and image sequence analysis [19]. Graph matching techniques have proved to be a robust method for matching partial representation of a pattern and its model.

In our approach, graph matching techniques are used to match features from a region in an image, and features from the corresponding region in the next image of a sequence, providing a method for inexact feature matching to solve the problems of partial occlusion between the pattern and the model. In our approach, the relational structure of features from a region in the first frame is considered the pattern, and the relational structure of features from the corresponding region in the next frame is considered the model.

To establish the matching between both structures, an association graph is used. Nodes of the association graph represent potential matches between two features, and links connecting nodes represent compatibilities between pairs of potential matches. A maximal clique detection technique can be used to find possible assignments between features in the pattern and the model, and to achieve inexact matching between features in both regions, providing a way to tolerate partial occlusions and unexpected variations of structural representations (regions) from image to image.

Given a graph, any fully connected subgraph is called a clique. The maximal clique of a graph is the largest subgraph in which all its nodes are linked among them. Thus, the maximal clique represents the largest number of nodes in the graph which are all compatible among them. Maximal clique detection is known to be an NP-complete problem [20], and this means the problem has exponential cost with respect to the number of nodes in the graph.

Building the Association Graph between Regions

Given the $x^k_{i,j}$, $i = 1, \ldots, F^k_j$, feature points extracted from region L^k_j in frame k, and the $x^{k+1}_{l,m}$, $l = 1, \ldots, F^{k+1}_m$, feature points extracted from region L^{k+1}_m in frame $k + 1$, nodes of the association graph between features of these two regions represent pairs of possible matches between features of both regions; that is, each node is represented by a pair $(x^k_{i,j}, x^{k+1}_{l,m})$ that represents a possible match between feature $x^k_{i,j}$ and feature $x^{k+1}_{l,m}$. Remember that $x^k_{i,j}$ are considered features from the pattern and $x^{k+1}_{l,m}$ features from the model.

In general, the association graph would have $F^k_j * F^{k+1}_m/2$ nodes as possible matches, that is, approximately the number of features squared. Moreover, we can apply the principle of proximity to reduce significantly the number of nodes in the association graph. Thus, if we project a feature from frame k to frame $k + 1$ using as estimated motion parameters the motion between the two previous frames, the corresponding feature in frame $k + 1$ must be very close to the projected feature from frame k, due to the fact that motion changes smoothly.

Therefore, if R^{k-1} and t^{k-1} are the motion parameters found for the two previous frames $k - 1$ and k, we can establish the following condition to consider a possible match between feature $x^k_{i,j}$ in frame k and feature $x^{k+1}_{l,m}$ in frame $k + 1$,

$$\|p^k_{i,j} - x^{k+1}_{l,m}\| < D_f$$

with $p^k_{i,j}$ the image coordinates in frame $k + 1$ after point $x^k_{i,j}$ has been projected from frame k into frame $k + 1$ with estimated motion parameters R^{k-1} and t^{k-1}.

Therefore, $\|p^k_{ij} - x^{k+1}_{l,m}\| < D_f$, the Euclidian distance between a projected feature from frame k and a feature from frame $k + 1$, must be less than a certain margin of tolerance D_f. Using this constraint, the number of nodes in the association graph can now be considered as proportional to the number of features in the pattern, since if each feature in the pattern is now associated with an average of m features in the model, for instance, two or three. The number of nodes in the association graph now is reduced to $F^k_j * m$, with m being a constant for every region in frame k.

Links between nodes denote compatibility between the matches represented by the two nodes linked. In order to establish these compatibility links, the principles of rigidity and exclusion are applied. Therefore, to establish a link between two nodes, $(x^k_{i,j}, x^{k+1}_{l,m})$ and $(x^k_{h,j}, x^{k+1}_{n,m})$, the following two conditions must be satisfied:

—Exclusion. Since only one-to-one matches are allowed, links between nodes which could establish a many to one match are discarded. Therefore, pairs of nodes of either the form $(x^k_{i,j}, x^{k+1}_{l,m})$ and $(x^k_{i,j}, x^{k+1}_{n,m})$ or $(x^k_{i,j}, x^{k+1}_{l,m})$ and $(x^k_{h,j}, x^{k+1}_{l,m})$ cannot be linked, since they would allow, for instance in the first case, that feature $x^k_{i,j}$ from region L^k_j in frame k could be matched with feature $x^{k+1}_{l,m}$ and feature $x^{k+1}_{n,m}$ from region L^{k+1}_m in frame $k + 1$ at the same time, violating the principle of exclusion.

—Rigidity. As it has been pointed out before, distances between features in the same region are approximately constant between frames. Therefore, to establish a compatibility link between node $(x^k_{i,j}, x^{k+1}_{l,m})$ and node $(x^k_{h,j}, x^{k+1}_{n,m})$ requires that if these nodes represent matched features, the distance between feature $x^k_{i,j}$ and $x^k_{h,j}$ must be approximately the same as distance between features $x^{k+1}_{l,m}$ and $x^k_{n,m}$. This constraint can be expressed by the condition

$$C((x^k_{i,j}, x^{k+1}_{l,m}), (x^k_{h,j}, x^{k+1}_{n,m})) = \| \|x^k_{i,j} - x^k_{h,j}\| - \|x^{k+1}_{l,m} - x^k_{n,m}\| \| \leq D_r,$$

where D_r is a small tolerance allowing possible differences between frames. The quantity C is associated to each link, and it is used later to resolve possible ambiguities.

After the association graph has been built, a clique detection algorithm is used to find the maximal clique, whose nodes correspond to the largest number of compatible matching pairs. Among a wide variety of clique detection algorithms, that reported by Yang *et al.* [21] was used in this work. This algorithm uses a depth-first-search strategy with NP-complete performance; that is, it has, in the worst case, an exponential cost with respect to the number of nodes in the graph. The algorithm can be summarized as follows:

```
Algorithm: Find_Maximal_Clique $C_m$
        For each node $i$: = 1 to $N$, $label(i)$: = $i$;
        For each clique $i$: = 1 to $N$
                If $label(i)$ = = $i$, $C$: = {$i$}; /* new root clique */
                Fill_Clique($C$, node $i$, $N$);
                If $C$ is maximal, $C_m$: = $C$.
end Find_Maximal_Clique.

Procedure: Fill_Clique($C$, node $k$, $N$)
        for each node $j$: = $k + 1$ to $N$
                if node $j$ has links to all nodes in $C$
                        $C$: = $C$ + {$j$};
                        $label(j)$: = $label(k)$;
                        Fill_Clique($C$, node $j$, $N$).
                        end Fill_Clique.
```

If there is more than one maximal clique with the same number of nodes, a criterion based on a function to assess the rigidity constraint is used. From the several possible maximal cliques, if C_l is the number of links present in a maximal clique, we choose the clique which minimizes the sum of all quantities C_i associated to each link $i = 1, \ldots, C_l$ in the clique, that is,

$$\sum_{i=1}^{C_l} C_i = \text{minimum},$$

favoring matches which minimize the difference between distances from pairs of features in a frame, and pairs of corresponding features in the other frame.

The procedure described is repeated independently for each pair of matched regions between frames, thus matching features between matched regions as part of a hierarchical matching strategy. After feature points from each region have been matched, they provide adequate information for calculating the motion parameters between these frames.

Matching problems due to symmetries of feature distribution of the same object in two successive frames is solved by the principle of proximity, which only allows for matches of nearby features between frames, discarding all other possible symmetry transformations of feature distributions. Partial occlusion, which leads to missing features between frames, is solved through the properties of partial matching of clique detection techniques, allowing matching of a subset of features in the pattern with a subset of features in the model.

An Example

In order to understand how the described method works, refer to the little example of Fig. 1. Features in frame 1 (Fig. 1a) and frame 2 (Fig. 1b) are considered part of the same region which has already been matched. The first step is to project each feature from frame 1 (circles) to frame 2 (solid circles), using motion parameters from the previous two frames. To define the nodes in the association graph (Fig. 1c), the proximity constraint is used. Thus, allowing for a margin of error, in the case of features labeled as 1 and 2 in the first frame (Fig. 1a) and features labeled as *a* and *b* in second frame (Fig. 1b), nodes (1,a), (1,b), (2,a), and (2,b) are defined in the association graph. Features 3 and 4 in the first frame only define nodes (3,c) and (4,d) because of the previously mentioned constraint.

To build links between nodes, the principle of exclusion does not allow us to establish links, for example, between nodes (1,a) and (1,b), since this would mean that feature 1 could be matched at the same time with feature a and b, violating the assumption of a one-to-one match. The same applies between nodes (1,a) and (2,a), or between (1,b) and (2,b), and so on.

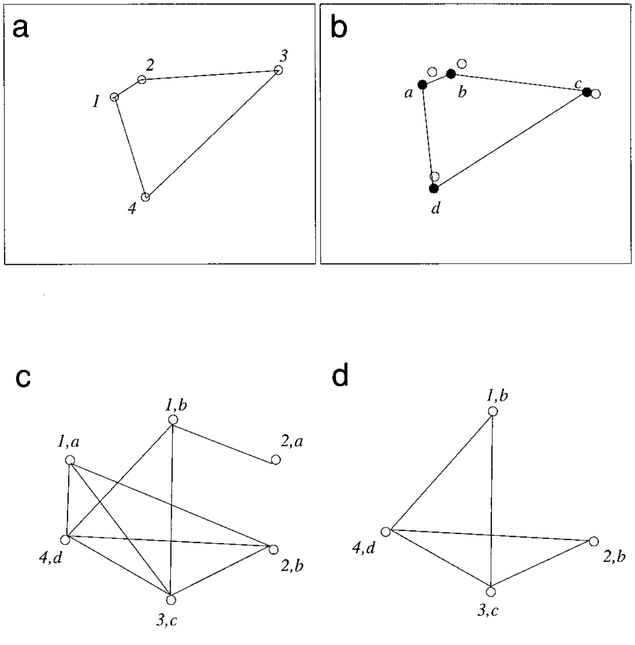


FIG. 1. Example of a set of features in two consecutive frames. Circles in the second frame (b) represent projected features from the first frame (a) using an estimation of motion parameters. Solid circles in the second frame (b) denote real location of features after real motion undergone from the first frame. (c) represents the association graph between features from frame (a) and (b). (d) represents the same association graph but supposing that feature *a* in the second frame has been lost.

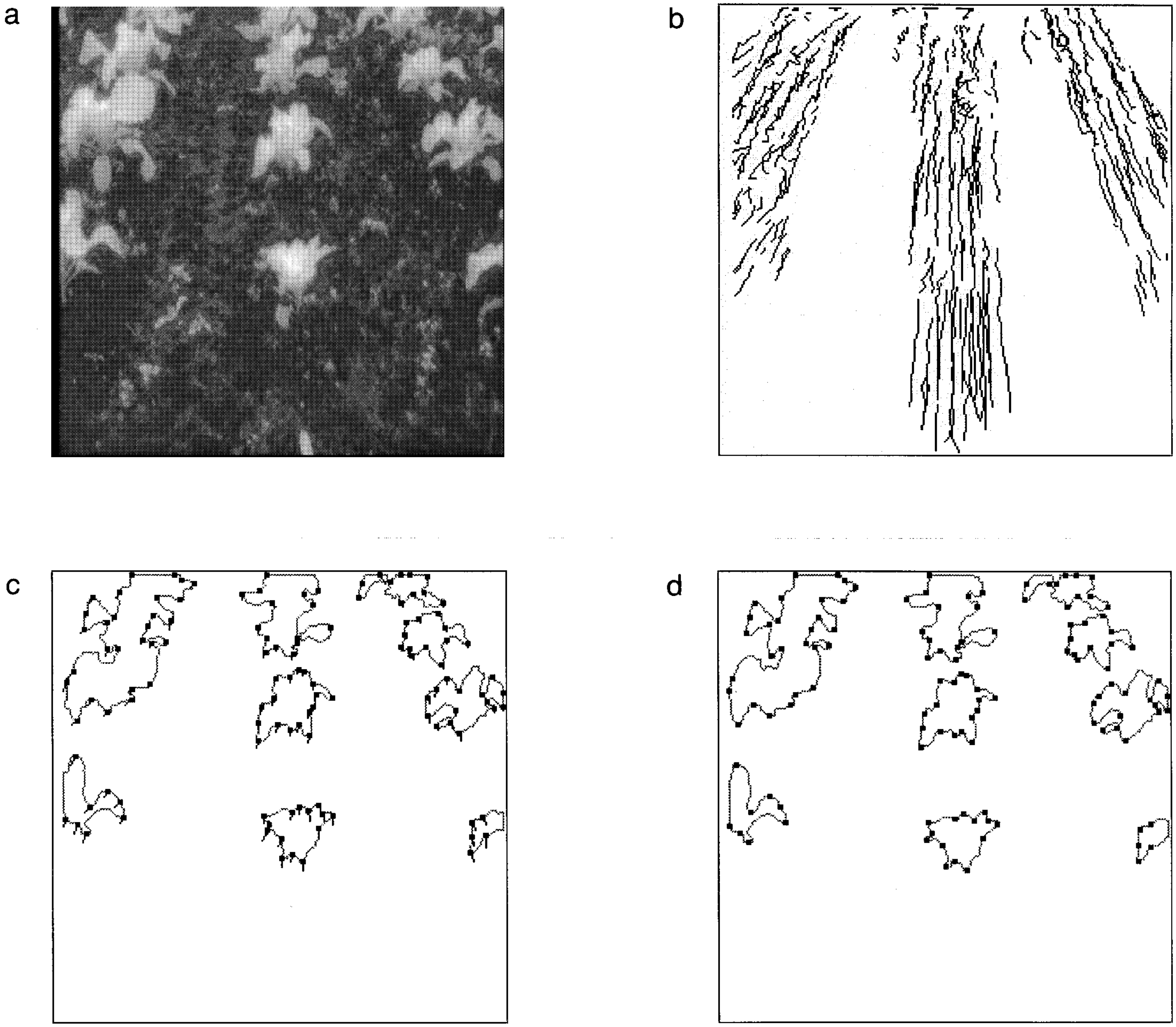


FIG. 2. (a) Infrared gray level image from a sequence taken with the front camera. (b) Trajectories of the features found across the first 10 frames of the sequence. (c) and (d) are a pair of contiguous images in a sequence. Solid points on region contours denote matched feature points between the pair of frames, and the image displacement vectors of each feature are shown with respect to its corresponding feature in subsequent frames.

Therefore, taking into account this principle of exclusion, a link between nodes (1,a) and (3,c), for instance, is established since the distance between features 1 and 3 is approximately the same as distance between features a and c; that is, they satisfy the principle of rigidity between features in the same frame. The same applies for the rest of the links established.

Once all links have been calculated, the maximal clique in the association graph of Fig. 1c is detected. In this case, the maximal clique is composed of nodes (1,a), (2,b), (3,c) and (4,d); thus the features matched correspond to the nodes in the clique. That is, feature 1 in the first frame has been matched with feature a in the second frame, and so on.

Let us suppose now that by effect of partial occlusion, or an error in the feature extraction step, feature a in the second frame is missed, thus leading to the association graph represented in Fig. 1d. In this case there are two possible sets of matches, since there are two maximal cliques. Therefore, to choose the clique which establishes the matching, the criterion function described in the previous section is used, that is, the clique which minimizes the sum of the quantities assigned to each link in the clique. The clique which minimizes this sum better preserves the rigidity constraint. In the case of the example, the right clique should be chosen using this criterion, the clique represented by nodes (2,b), (3,c), and (4,d) in Fig. 1d.

EXPERIMENTS AND DISCUSSION

As pointed out in the Introduction, the motivation of this work was to match feature points in image sequences, in order to extract motion information to guide a crop protection vehicle. Therefore, real images were used to test the described method. Image sequences were taken from a camera mounted on a moving vehicle during its movement along a crop grown in rows. Three types of images were used, images taken from a front camera, looking ahead with a broad view of the crop, images taken from the front camera but with a narrower view, and images taken from a side camera, with a close-up view of the plants.

Image sequences were taken at a rate of 4.17 images per second, and the vehicle was moving at 0.5 m per second, approximately. Each image sequence recorded had 30 images. Image acquisition was carried out under natural illumination condition. Scenes are usually composed of plants, weeds, soil, and sky.

To segment images in a sequence, two segmentation methods were used. The original images of the sequence in Figs. 2a and 3a were gray level images acquired with an infrared filter, and segmented using a hysteresis thresholding algorithm to differentiate soil (background) and plants (foreground). The two thresholds of the hysteresis algorithm were extracted automatically from the first image of the sequence, using a histogram-based method from the image points with high value of the gradient magnitude [22]. The same thresholds were used for the remaining images of the sequence.

The rest of the image sequences were segmented with a color segmentation algorithm [23], using the first image of the sequence to extract the color samples to train the color classifier. Each image was segmented into two types of region, plants and soil. After the initial segmentation result, plant regions with small areas were removed, and relabeled as soil regions (considered as background). A set of feature points from each region was extracted. These feature points were chosen as points from region contours. Particularly, feature points in the contour were placed at zero crossings of the derivative of the contour curvature [24].

Although in these experiments feature points have been chosen from the region contours, the method proposed could deal in general with any set of feature points extracted from the regions, for example, local maxima in gray level. Therefore, this matching approach could be used in a broader sense, rather than matching only region contour points.

From every set of correspondences between two frames, motion parameters were calculated using the method described in [25]. Better motion estimates from motion parameters calculated in previous frames could be done by a Kalman filter approach [26]. The motion calculated from previous frames was used to project regions and feature points from frame to frame, in order to perform the algorithm described in previous sections.

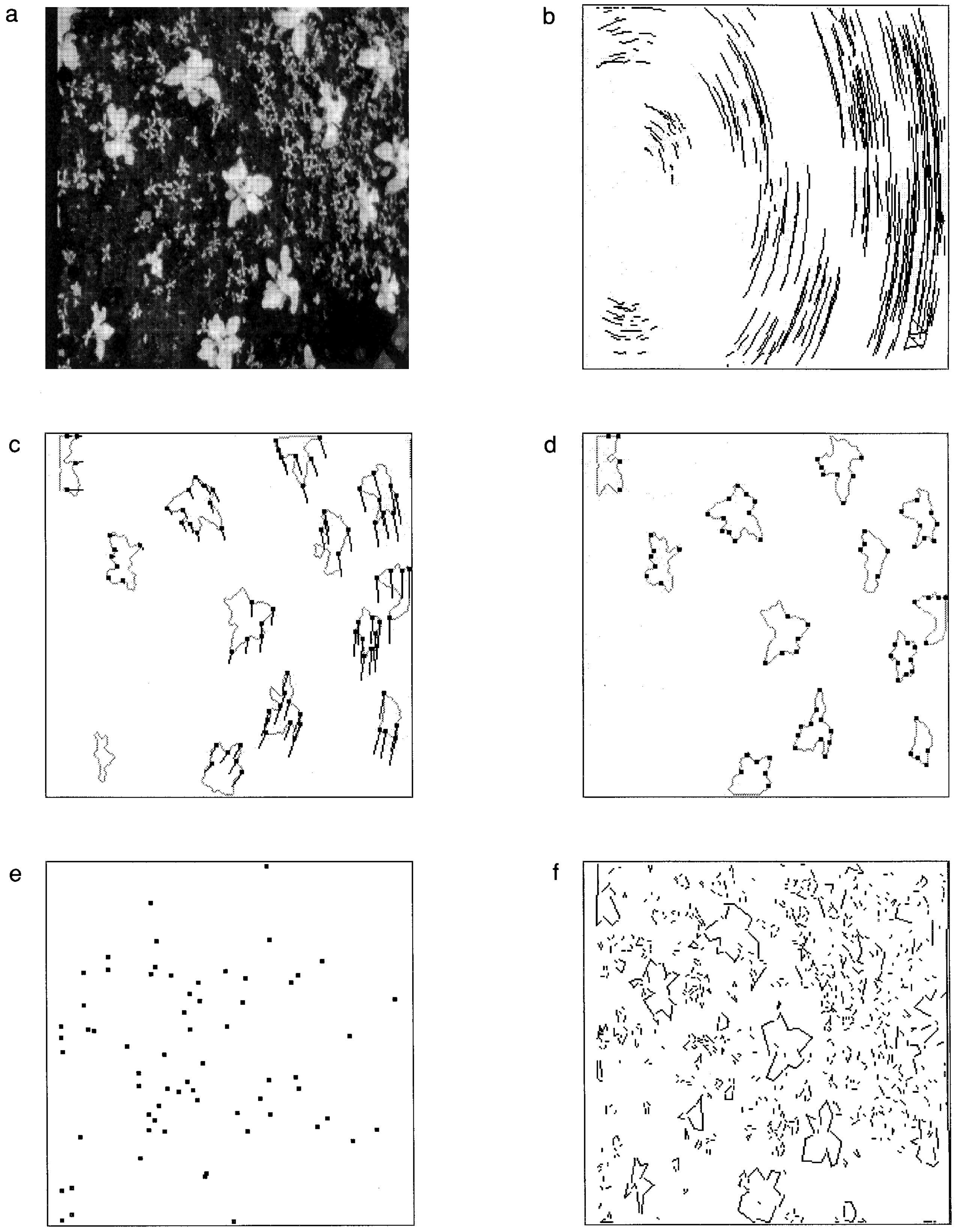
Figure 2a represents an original infrared image of a sequence taken by a front camera and with a middle range view. Figures 2c and 2d show the region contours of a pair of contiguous images from that sequence. Solid points on region contours denote matched feature points between pairs of frames, and image displacement vectors are shown for each feature point with respect to its corresponding matched feature in the subsequent frame. Note the reasonable goodness of matches, taking into account that they have been calculated in a closed form, without further refinement to filter possible noisy correspondences.

Region tracking by the approach described did not give any false matched regions. Although the region matching method is simple, it is efficient and provides very satisfactory results with the assumptions made for the problem. In problems where this region matching method would not give satisfactory results, other region matching methods could be applied instead [21, 27], and the rest of the method could be used as described here.

Figure 2b shows the trajectories in the image plane found for all the features matched across the first 10 images of the sequence. The trajectories show that the movement was toward the top of the image. These trajectories represent the tracking of features across several frames. The trajectories are sometimes split, due to the fact that their corresponding features were not successfully matched at some instant, or that they were missed in some frames. Small missing parts of trajectories could be recovered by techniques to interpolate missed points [28]; however, the purpose of this work is to establish correspondences between frames, rather than finding trajectories of features across consecutive frames.

Figure 3 shows the result of the algorithm applied to the same type of images as in Fig. 2 but with a rotational movement, in order to test the method in these situations. Note how the trajectories found in Fig. 3b denote a rotation around the rotational axis placed on the left of the image.

FIG. 3. (a) Infrared gray level image from a sequence taken with a rotational movement. (b) Trajectories of the features found across the first 10 frames of the sequence. (c) and (d) are a pair of contiguous images in a sequence. (e) are the corners detected from image (a). (f) is the polygonal approximation of contours extracted from image (a).


a
b
c
d
e
f

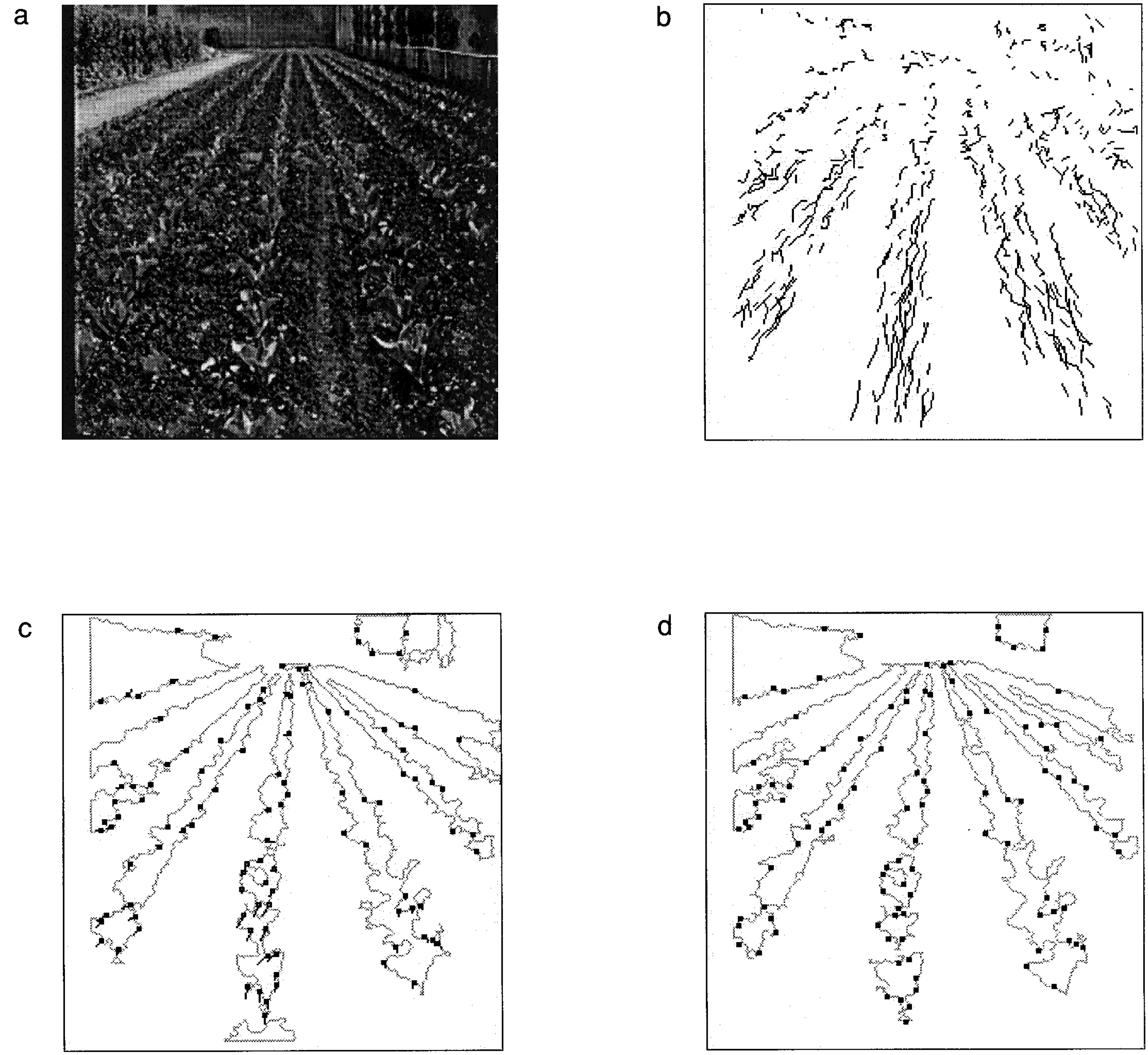
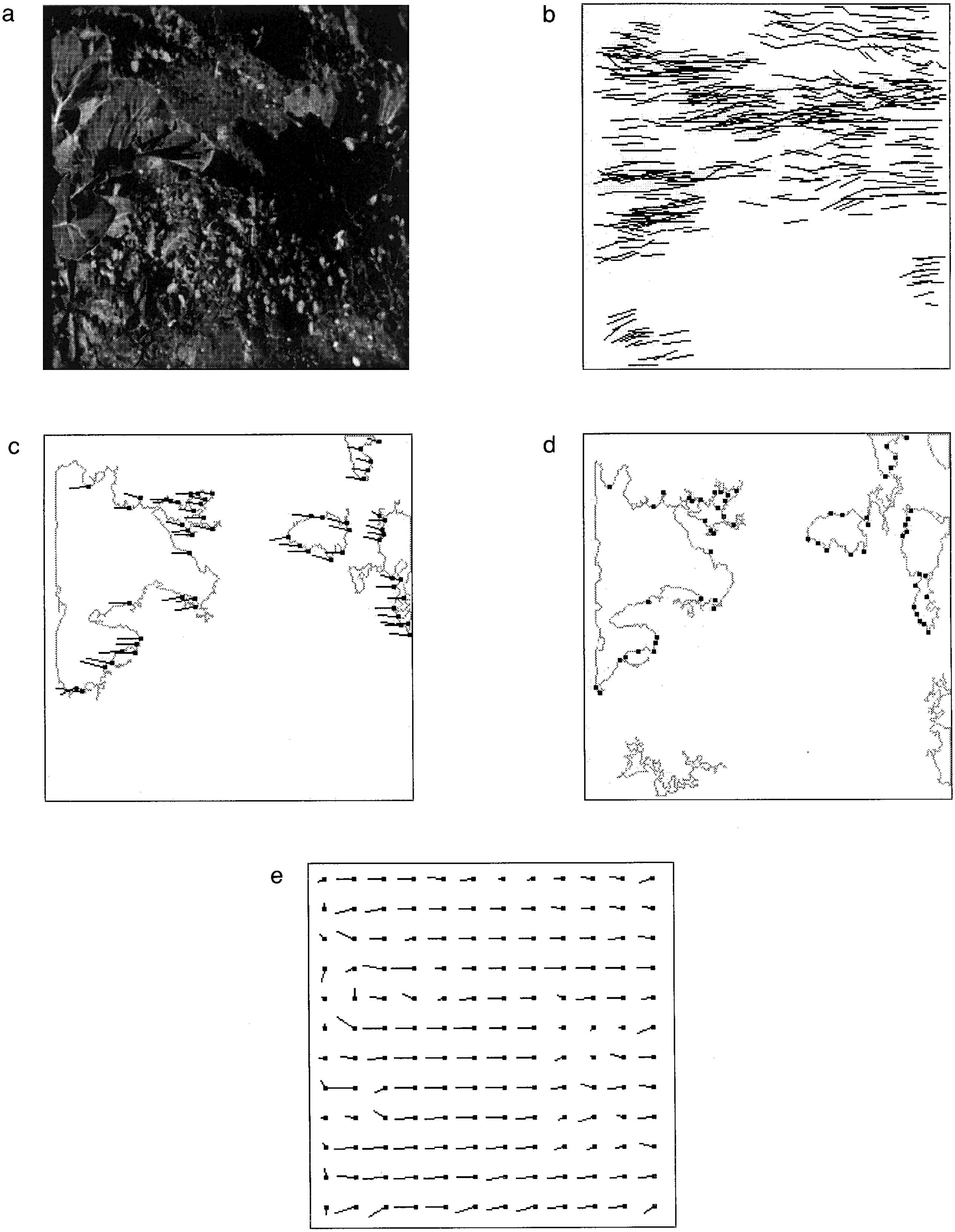


FIG. 4. (a) Red band of a color image from a sequence taken with the front camera (broad view). (b) Trajectories of the features found across the first 20 frames of the sequence. (c) and (d) are a pair of contiguous images in a sequence.

When regions represent parts of more than one object in the scene, significant region deformations from frame to frame may be produced, due to the fact that different parts of the region would undergo different motion with respect to the image plane. Figure 4 shows an example of this problem. Figure 4a represents a red band of a color image of the crop when the camera is looking ahead while the vehicle is moving. Because of the different image plane motion for features at the bottom of the image and features at the top of the image in the same region, the rigidity

FIG. 5. (a) Red band of a color image from a sequence taken with the side camera (close-up view). (b) Trajectories of the features found across the 10 frames of the sequence. (c) and (d) are a pair of contiguous images in a sequence. (e) Gray level image representing the magnitude of the velocity vectors found a standard optical flow algorithm [29].


a
b
c
d
e

constraint fails if this deformation effect in the image plane is significant.

In such cases, where regions span a large portion of the image, a splitting strategy is used to divide the problem; so instead of matching two sets of features, either of them is divided into subsets and then the matching is performed between each subset with the other set of features. Thus, given the set of features of a large region in the image, its features are grouped into different clusters according to their distance, using a sequential clustering algorithm. Therefore, each subset satisfies the rigidity constraint locally in the part of the region that each subset represents. The result of this approach is shown in Fig. 4. Image plane motion in this example is quite small, since the camera is looking ahead, and with virtually no image plane motion at the top of the image across the whole sequence.

Figure 5a shows the red band of a color image from a sequence taken by the side camera. Figures 5c and 5d are a pair of images from the sequence, presenting a case of oversegmentation (part of a region in Fig. 5d is missed in Fig. 5c), partial occlusion (the region on the right in Fig. 5d is partly gone out of view in Fig. 5c), and small deformation effects in contour regions from frame to frame. Figure 5b shows the trajectories of the features across the first 10 images of the sequence, showing a rightward movement.

Discussion

One advantage of the method is that it produces one shot matches, with unlikely false matches. This is a different approach from methods which, like relaxation-based methods or methods which perform iterative algorithms, solve possible ambiguities from an initial matching map using consistency rules. On the other hand, a drawback of the method is its dependence on the result of the segmentation procedure used, since an inadequate segmentation could lead to false matches.

The method is particularly suitable in applications where the attention is focused on some reference objects in the scene. Figure 6a represents the red band of a color image from a sequence taken in the laboratory. After segmenting these images, we focused the attention on certain objects, and our algorithm to extract matched feature points was applied, giving the results shown in Fig. 6.

The method presented here could be an alternative method to other feature-based methods, like point- or line-based methods, in some situations. Figure 3e represents the corners extracted from the image in Fig. 3a using the Plessey corner detector [1], and Fig. 3f the straight line approximation of edges found by applying the Canny operator to the original image in Fig. 3a. In this type of image, i.e., natural environments with strong presences of textured objects, corners and line segments are not very reliable, since textured patches may produce false corners (Fig. 3e), and polygonal approximations of contours in free form objects lead to high numbers of small linear segments (Fig. 3f). In these situations, matching from more integrated information, as it has been proposed, may overcome these problems.

On the other hand, images from man-made environments (Fig. 6) are more appropriate for corner- and line-based methods, since in these situations corners (Fig. 6e) and line segments (Fig. 6f) are extracted more reliably, and they are stable across images in the sequence. Moreover, the hierarchical method proposed here can also be applied to these situations (Figs. 6c and 6d).

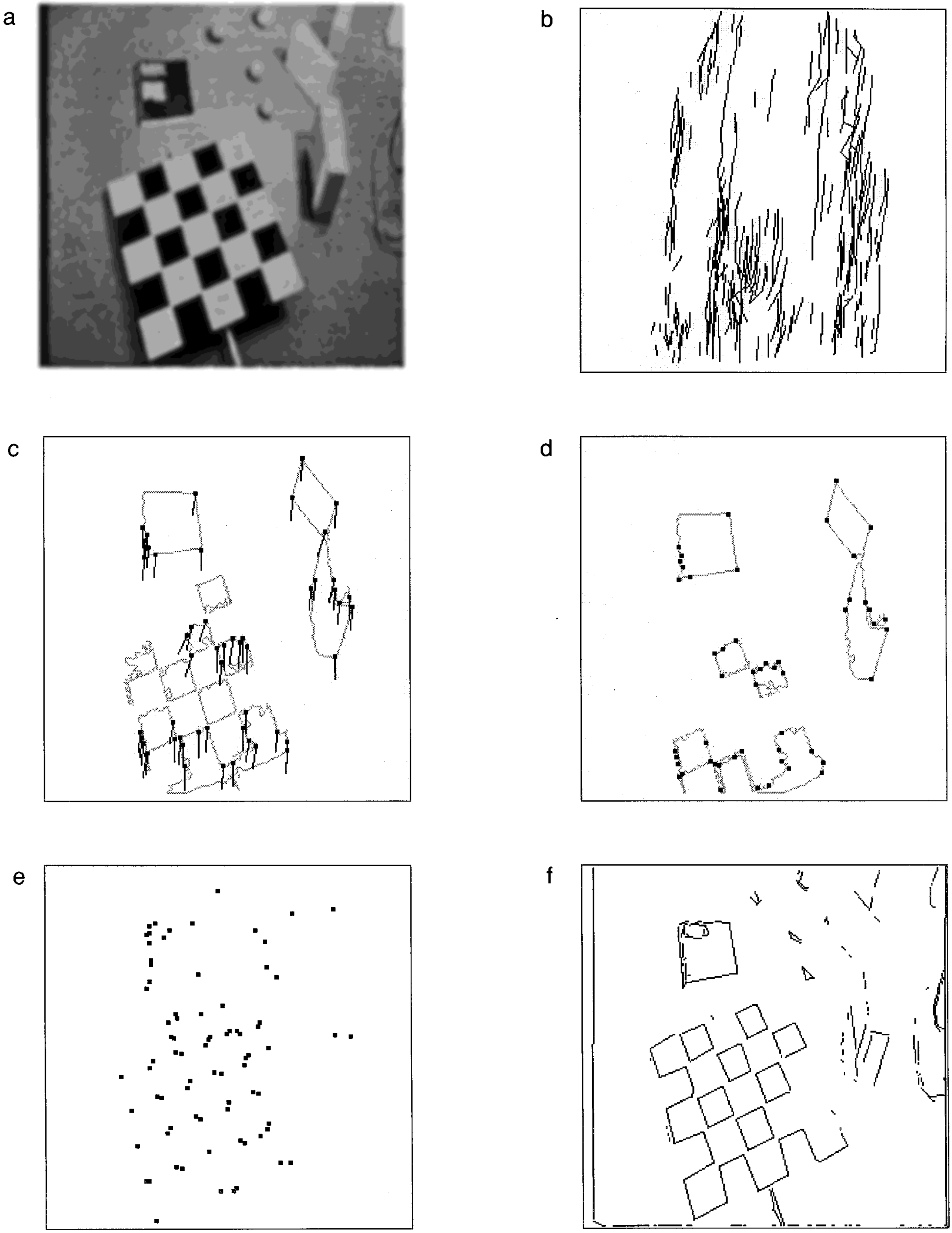
Another difference between our method and the usual techniques of matching corners and lines across frames is that they usually establish matches based on feature properties themselves (image intensity, contrast, gradient, length of segments, orientation, etc.). In contrast, the method proposed does not use specific properties of features; it is based on the principles of rigidity, proximity, and exclusion.

It is worth pointing out that, like any feature-based method, motion between frames does not have to be very small, as long as motion changes smoothly and image appearance does not significantly differ from frame to frame.

With respect to optical flow-based methods, Fig. 5e shows the result of a standard optical flow algorithm [29]. Optical flow methods rely on local data from the surroundings of a point, and some patches in the image may not provide good information. Thus, image patches which may introduce noise should be identified and suppressed to extract the motion information. On the other hand, the hierarchical approach proposed uses the information integrated over regions, and the attention is focused on the patches of the image these regions represent. Therefore the information is somehow selected.

Concerning computational time, the main drawback is the maximal clique search. This complexity has been reduced by splitting the problem by using association graphs between regions instead of the overall image. However, if the feature set of a region is large, computational costs can be significantly increased with respect to regions with

FIG. 6. (a) Red band of a color image from a sequence taken in a laboratory scene. (b) Trajectories of the features found across the first 10 frames of the sequence. (c) and (d) are a pair of contiguous images in a sequence. (e) are the corners detected from image (a). (f) is the polygonal approximation of contours extracted from image (a).


a
b
c
d
e
f

smaller feature sets, due to the exponential complexity of the clique detection algorithms.

The time spent to perform matching on a typical pair of images from examples shown, after segmentation, is about 0.9 s in a HP-9000/730 workstation. Image segmentation has a negligible cost in the case of thresholding for the infrared images, 0.04 s. Color segmentation can be implemented by input lookup tables after the training step, although in these experiments the full process was applied spending about 0.5 per image.

CONCLUSIONS

A method has been presented to establish feature matches between images in sequences. The algorithm is based on a hierarchical approach, first matching regions between images previously segmented, and then matching feature points between pairs of matched regions. Regions are matched according to a region distance criterion, projecting region points from frame to frame, using an estimation of motion parameters between frames being analyzed. Feature points between regions are matched using a subgraph matching technique, to deal with the problem of partial occlusion and missing features.

Matching features is based on three principles: exclusion, proximity, and rigidity. The use of subgraph matching techniques through maximal clique detection allows the method to tackle the problem of partial occlusion and missing features between frames with satisfactory results. The hierarchical scheme allows the method to provide one shot matching, that is, matched features in a single step, without further resolution of ambiguities, with unlikely false matches.

The method is particularly suitable for image sequences where attention can be focused on some objects in the scene, tracking some type of objects to match feature points of those objects.

Because the algorithm requires images to be segmented, the choice of an unsuitable image segmentation criterion could lead to a poor matching performance, for instance, if regions resulting from segmentation are too small.

Although the region matching algorithm proposed cannot know if regions have been split or merged from frame to frame, it provides pairs of whole or partially matched regions. This information is enough to reach the objective of finding matched feature points.

Further work is required to achieve a real time performance, since the maximal clique search must be optimized. We will consider either possible implementation of present algorithms for clique detection in parallel systems, or to search for an alternative clique detection algorithm adapted to the type of graphs generated by this method, in order to try to speed up the clique detection, although it is known to be a NP-complete problem.

APPENDIX: NOMENCLATURE

(x, y, z)	Coordinates of a point in 3D
(x', y')	Coordinates of a point in the image plane
f	Focal length of the camera
X^k	3D coordinates (x, y, z) of a point at the instant when frame k is acquired
x^k	Image plane coordinates (x', y') of a 3D point X^k
L_j^k	Region number j in frame k
$x_{i,j}^k$	Feature number i in region L_j^k.
F_j^k	Feature points extracted from region L_j^k in frame k
N_m	Area of region m
$N_{j,m}$	Number of points associated to region m in frame $k + 1$ projected from region j into frame k
$d(L_j^k, L_m^{k+1})$	Distance between region L_j^k and region L_m^{k+1}
L^k	Number of regions in frame k
D	Threshold for the minimum distance between regions to be considered
D_f	Margin of tolerance in distance between features of different frames
D_r	Tolerance value for the difference of distances between pairs of matched features
C_l	Number of links present in a maximal clique
C_i	Score associated to link i of a maximal clique

ACKNOWLEDGMENTS

This work has been partially supported by the BBSRC, and Grant PF92-52660540 from the Spanish Ministry of Science and Education.

REFERENCES

1. C. Harris and M. A. Stephens, A combined corner and edge detector, in *Proceedings of Alvey Vision Conference, 1988,* pp. 147–151.
2. Z. Zhang and O. Faugeras, Estimation of displacements from two 3-D frames obtained from stereo, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(12), 1992, 1141–1156.
3. Y. Liu and T. S. Huang, Three-dimensional motion determination from real scene images using straight line correspondences, *Pattern Recognit.* **25**(6), 1992, 617–639.
4. J. Porrill and S. Pollard, Curve matching and stereo calibration, *Image Vision Comput.* **9,** 1991, pp. 45–50.
5. Z. Zhang, On local matching of free-from curves, in *Proceedings of British Machine Vision Conference, 1992,* pp. 345–356.
6. N. M. Nasrabadi, A stereo vision technique using curve-segments and relaxation matching, *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(5), 1992, 566–572.
7. N. A. Thacker, Y. Zheng, and R. Blackbourn, Using a combined stereo/temporal matcher to determine ego-motion, in *Proceedings of British Machine Vision Conference, 1990,* pp. 121–126.

8. M. J. Stephens, Matching features from edge-processed image sequences, in *Proceedings Alvey Vision Conference, 1987,* pp. 185–188.

9. Z. Zhang, Token tracking in a cluttered scene, *Image Vision Comput.* **12**(2), 1994, pp. 110–120.

10. R. N. Strickland and Z. Mao, Computing correspondences in a sequence of non-rigid objects, *Pattern Recognition,* **25**(9), 1992, pp. 901–912.

11. C. H. Lee and A. Joshi, Correspondence problem in image sequence analysis, *Pattern Recognition,* **26**(1), 1993, pp. 47–61.

12. R. C. Bolles and R. A. Cain, Recognizing and locating partially visible objects: The local-feature-focus method, *Int. J. Robot. Res.* **1**(3), 1982, pp. 57–82.

13. J. R. Beveridge, R. Weiss, and E. M. Risseman, Combinatorial optimization applied to variable scale 2D model matching, in *Proceedings of International Conference on Pattern Recognition, 1990,* pp. 18–23.

14. G. L. Scott and H. C. Longuet-Higgins, An algorithm for associating the features of two images, *Proc. R. Soc. London B* **244,** 1991, pp. 21–26.

15. L. S. Shapiro and J. M. Brady, Feature-Based Correspondence: An eigenvector approach, *Image Vision Comput.* **10**(5), 1992, pp. 283–288.

16. Y. C. Kim and K. Price, Refinement of noisy correspondence using feedback from 3-D motion, in *Proceedings of International Conference on Computer Vision and Pattern Recognition, 1991,* pp. 836–838.

17. N. Kehtarnavaz and S. Mohan, A framework for estimation of motion parameters from range images, *Compt. Vision Graphics Image Process.,* **45,** 1989, pp. 88–105.

18. R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques, *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(11), 1989, pp. 1168–1180.

19. B. Radig, Image sequence analysis using relational structures, *Pattern Recognition,* **17**(1), 1984, pp. 161–167.

20. K. Mehlhorn, *Data Structure and Algorithms 2: Graphs Algorithms and NP-Completeness,* Springer-Verlag, Berlin/New York, 1984.

21. B. Yang, W. E. Snyder and G. L. Bilbro, Matching oversegmented 2D images to models using association graphs, *Image Vision Comput.* **7**(2), 1989, pp. 135–143.

22. R. Brivot and J. A. Marchant, Segmentation of plants and weeds for a precision crop protection robot using infrared images, submitted to *Proceedings IEE Vision, Image and Signal Processing.*

23. F. Pla, F. Juste, F. Ferri, and M. Vicens, Colour segmentation based on a light reflection model to locate citrus fruits for robotic harvesting, *Comput. Electronics Agriculture,* **9,** 1993, 53–70.

24. F. Mokhtarian and A. Mackworth, Scale-based description and recognition of planar curves and two-dimensional shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* **8**(1), 1993, pp. 34–43.

25. K. Kanatani, Unbiased estimation and statistical analysis of 3D rigid motion from two views, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(1), 1993, pp. 37–50.

26. L. Matthies and T. Kanade, Kalman filter-based algorithms for estimating depth from image sequences, *Int. J. Comput. Vision,* **3,** 1989, pp. 209–236.

27. H. S. Ranganath and L. J. Chipman, Fuzzy relaxation approach for inexact scene matching, *Image Vision Comput.* **10**(9), 1992, pp. 631–640.

28. V. Salari and I. K. Sethi, Feature point correspondence in the presence of occlusion, *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(1), 1990, pp. 87–91.

29. E. P. Simoncelli, *Distributed Representation of Image Velocity,* Vision and Modeling Technical Report 202, MIT Media Laboratory, November 1990.