Pattern Recognition, Vol. 26, No. 4, pp. 595-610, 1993
Printed in Great Britain

0031-3203/93 $6.00+.00
Pergamon Press Ltd
Pattern Recognition Society

MATCHING MOTION TRAJECTORIES USING SCALE-SPACE

KRISHNAN RANGARAJAN, WILLIAM ALLEN and MUBARAK SHAH†
Computer Science Department, University of Central Florida, Orlando, FL 32816, U.S.A.

(*Received* 22 *January* 1992; *in revised form* 8 *July* 1992; *received for publication* 18 *August* 1992)

Abstract—The goal is to design a recognition system which can distinguish between two objects with the same shape but different motion, or between two objects with the same motion but a different shape. The input to the system is a set of two-dimensional (2D) trajectories from an object tracked through a sequence of *n* frames. The structure and three-dimensional (3D) trajectories of each object in the domain are stored in the model. The problem is to match the information in the model with the input set of 2D trajectories and determine if they represent the same object. The simplest way to perform these steps is to match the input 2D trajectories with the 2D projections of the 3D model trajectories. First, a simple algorithm is presented which matches two single trajectories using only motion information. The 2D motion trajectories are converted into two one-dimensional (1D) signals based on their speed and direction components. The signals are then represented by scale-space images, both to simplify matching and because the scale-space representations are translation and rotation invariant. The matching algorithm is extended to include spatial information and a second algorithm is proposed which matches multiple trajectories by combining motion and spatial match scores. Both algorithms are tested with real and synthetic data.

Motion analysis   Motion representation   Recognition   Scale-space   Trajectory matching

1. INTRODUCTION

One of the goals of computer vision is to design object recognition systems that can identify a specific object of interest. Previous research in object recognition has relied exclusively on shape information. Shape is a very important attribute which defines the form and spatial arrangement of an object, and is invariant to certain transformations. Other information about an object exists, however, such as motion, specularity and texture, which may prove useful in recognition.

Our research has been directed toward the use of motion information in object recognition. We believe that in many cases, where an object has a fixed and predefined motion, the trajectories of several points on the object may serve to uniquely identify the object. Therefore, it should be possible to recognize certain objects based on motion information obtained from the trajectories of representative points. We have developed a method for matching sets of trajectories which supplements motion information with knowledge about the spatial relationships between certain points on the object.

The input to our system is a set of two-dimensional (2D) trajectories from an object tracked through a sequence of *n* frames. These trajectories are segmented by identifying each trajectory with a single object in the image.[1] The structure and three-dimensional (3D) trajectories of each object in the domain are stored in the model. The problem is to match the information in the model with the input set of 2D trajectories. The matching process has two main steps: (1) establish a one-to-one, onto correspondence between the object model 3D trajectories and the input set of 2D trajectories, and (2) determine the position of the object in the model coordinate system such that the input 2D trajectories closely match the corresponding 2D projections of the model's 3D trajectories. The simplest way to perform these steps is to match the input 2D trajectories with the 2D projections of the 3D model trajectories. In this paper, we propose a multi-scale approach for matching 2D trajectories. First, we present Algorithm A, which matches two single trajectories using only motion information. We then extend that algorithm to include spatial information and propose Algorithm B, which matches multiple trajectories.

2. RELATED WORK

We will briefly survey recent research in motion and recognition. A great deal of work has been done in the field of psychology to show that people can recognize objects from their trajectories. It has been theorized that humans can recognize an object based on the motion of several points on that object by inferring the 3D structure of the object from the transformations the 2D image undergoes. Todd[2] is interested in distinguishing between rigid and several types of non-rigid motion such as bending, stretching, twisting and flowing. By displaying the trajectories of either rigid or non-rigid objects, Todd shows that human observers are able to distinguish between the two. Cutting[3] and Johansson[4] discuss the relative motion of individual parts of an object and the common

† Author to whom correspondence should be addressed.

motion of the object as a whole. Cutting defines absolute motion as the path of a particular point on an object. The absolute motion of a point is also its trajectory.

In the computer vision field, several researchers have reported work related to detection of motion before recognition. Allmen and Dyer[5] describe a method for detecting cyclic motion using spatiotemporal surfaces and curves. They define cyclic motion to be repeating curvature values along a path of motion. They argue that the detection and description of cyclic motion does not depend on prior recognition of objects or on absolute position information and must make use of long-range temporal sequences and be sensitive to multiple scales. They use the curvature scale-space of the spatiotemporal curve to detect intervals of repeating curvature values. Koller *et al.*[6] use a set of German motion verbs to characterize vehicle trajectory segments in terms of natural language concepts.

We have found only a few papers related to the use of motion in object recognition systems. Murray *et al.*[7] have developed a system for recognizing moving polyhedral objects. They recover 3D structure from visual motion and use that structure information to match a CAD-based 3D wireframe model. In recovering structure they assume uniform motion, therefore, no significant motion cues are employed in the matching process. Hogg[8] uses a generate and test strategy to find a known object in an image. He models humans with generalized cylinders, predicts the occluding edges and compares the predicted edge points with the edge points in the image. A difference picture is used to identify the approximate position of moving objects in the first frame of a sequence of images. The search-space in subsequent frames is reduced through the use of kinematic constraints.

3. MOTION TRAJECTORIES

A trajectory is defined as a sequence of points $((x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots, (x_n, y_n))$, ordered by an implicit time dimension. We represent a 2D trajectory as two one-dimensional (1D) functions: speed and direction, which are functions of time. The speed function s_i and direction function d_i are defined as follows:

$$s_i = \sqrt{((x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2)}$$

$$d_i = \arctan\frac{y_{i+1} - y_1}{x_{i+1} - x_1}.$$

Both the speed and direction functions are translation and rotation invariant. When a trajectory is scaled by a factor k, the direction function is scale invariant, but the speed function is scaled by the same factor k. The proposed matching algorithm has been designed to give a good match score between a function and its scaled version. Motion trajectories are similar to 2D shape contours but with two notable differences: (1) two trajectories, which could be considered identical by shape-matching methods, may prove to be very different when they are considered as trajectories, and (2) shape outlines are generally considered to be closed curves, however, motion trajectories may not always produce closed curves.

4. MATCHING SINGLE TRAJECTORIES

Algorithm A is intended to match pairs of single trajectories using a scale-space representation as the basis for matching. To produce the 2D scale-space image, we convolve the 1D speed and direction signals with the second derivative of the Gaussian over a range of σ values. We then locate the speed and direction zero-crossings by scanning the scale-space images and testing the values in a neighborhood around each point. If we let t_0 be the point that we are testing, then $t_0 + \Delta t$ and $t_0 - \Delta t$ will be points at a distance Δt on either side of t_0. We will let $f(t_0 + \Delta t)$ and $f(t_0 - \Delta t)$ be the functions at those points. When we convolve those functions with the second derivative of the Gaussian, if the results have opposite signs, then a zero-crossing occurs at t_0. Therefore, if $f''(t_0 + \Delta t) * \exp[-(t_0 + \Delta t)^2/2\pi\sigma^2] > 0$ and $f''(t_0 - \Delta t) * \exp[-(t_0 - \Delta t)^2/2\pi\sigma^2] < 0$ or vice versa, then a zero-crossing exists at t_0. The strength and polarity of each zero-crossing is referred to as the zero-crossing potential. The strength and polarity of the zero-crossing are defined, respectively, as the magnitude and sign of the function $f(t_0)$ convolved with the first derivative of the Gaussian, $f'(t_0) * \exp(-t_0^2/2\pi\sigma^2)$.

It is well known that the zero-crossings of a distorted signal are delocalized and as a result, the zero-crossing potentials from a distorted signal are not the same as those from an undistorted signal. However, a smoothed zero-crossing potential is less sensitive to noise and our algorithm matches two trajectories by computing the difference between their smoothed zero-crossing potentials.

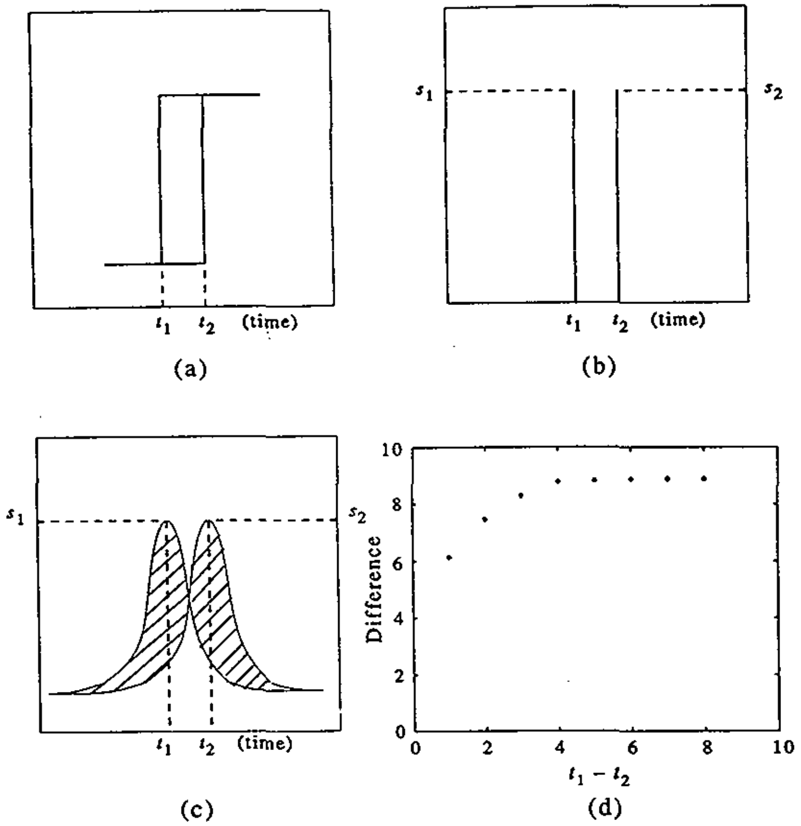
Assume that the velocity and speed changes can be modeled as step functions. Let $u(t)$ be a step function and let $s_1 u(t - t_1)$ be a step function of size s_1, which occurs at time t_1. When we convolve $s_1 u(t - t_1)$ with the first derivative of the Gaussian, we get

$$s_1 u(t - t_1) * \frac{\mathrm{d}}{\mathrm{d}t} \exp\left(\frac{-t^2}{2\pi\sigma^2}\right).$$

This is equal to

$$\frac{\mathrm{d}}{\mathrm{d}t}[s_1 u(t - t_1)] * \exp\left(\frac{-t^2}{2\pi\sigma^2}\right)$$

which gives us $s_1 \exp[-(t - t_1)^2/2\pi\sigma^2]$. Its value at $t = t_1$ can be approximated by $s_1 \delta(t - t_1)$, where $\delta(t)$ is an impulse function. Applying the diffusion step by convolving $s_1 \delta(t - t_1)$ with the Gaussian we obtain $s_1 \exp[-(t - t_1)^2/2\pi\sigma^2]$. The mismatch ξ, between a step of size s_1 at t_1 and a step of size s_2 at t_2 is computed by integrating the absolute difference of diffused po-



$t_1 - t_2$	ξ
8	$\sqrt{2\pi}(1 + erf(2\sqrt{2})) = 8.89$
7	$\sqrt{2\pi}(1 + erf(1.75\sqrt{2})) = 8.88$
6	$\sqrt{2\pi}(1 + erf(1.5\sqrt{2})) = 8.87$
5	$\sqrt{2\pi}(1 + erf(1.25\sqrt{2})) = 8.83$
4	$\sqrt{2\pi}(1 + erf(\sqrt{2})) = 8.78$
3	$\sqrt{2\pi}(1 + erf(0.75\sqrt{2})) = 8.29$
2	$\sqrt{2\pi}(1 + erf(0.5\sqrt{2})) = 7.47$
1	$\sqrt{2\pi}(1 + erf(0.25\sqrt{2})) = 6.14$

(e)

Fig. 1. (a) Two steps of the same size at different locations, t_1 and t_2; (b) detected zero-crossings of strength s_1 and s_2; (c) the diffused zero-crossings (their difference is shown by the shaded area and is denoted by ξ); (d) ξ in a graph form; (e) ξ in a table form.

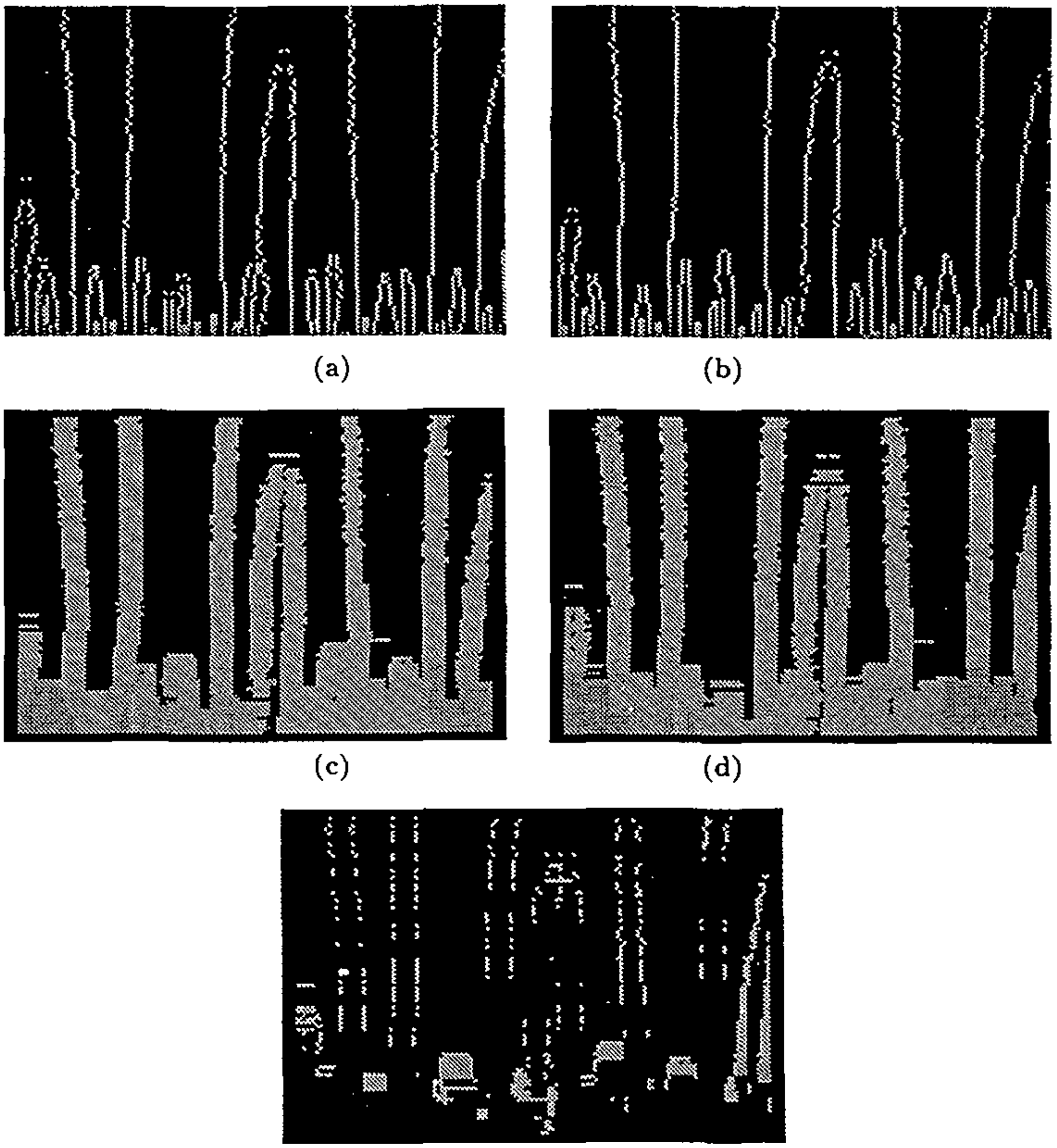
tentials over the whole image:

$$\xi = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left| s_1 \exp\left[\frac{-(t-t_1)^2}{2\pi\sigma^2}\right] - s_2 \exp\left[\frac{-(t-t_2)^2}{2\pi\sigma^2}\right]\right| \mathrm{d}t\,\mathrm{d}\sigma. \qquad (1)$$

This mismatch captures the differences in strength, polarity and location of the zero-crossings.

For the case when $s_1 = s_2 = 1$, equation (1) was solved symbolically; the results are shown in graph form in Fig. 1(d) and in table form in Fig. 1(e). It is clear that for a fixed σ the mismatch ξ decreases with the decrease in the distance between the two zero-crossings. When σ gets large enough, the distance between the two zero-crossings becomes insignificant, and the response becomes 0.

The effect of diffusing the zero-crossing potential in two dimensions is shown in Figs 2 and 3. The two *scale-space images to be compared are shown in parts* (a) and (b) of these figures, while parts (c) and (d) show the zero-crossing potentials scaled between 0 and 100. In order to make a comparison of these contours, the same scaling function was used for (a) and (b). The diffused versions of the zero-crossing potentials are shown in (c) and (d), and the difference picture between



(a)

(b)

(c)

(d)

(e)

Fig. 2. (a) The zero-crossing potential of the speed scale-space of trajectory A1 (shown in Fig. 8(a)); (b) the zero-crossing potential of the speed scale-space of trajectory B1 (a rotated version of A1, shown in Fig. 8(b)); (c) diffused version of (a); (d) diffused version of (b); (e) difference picture between (c) and (d).

(c) and (d) is shown in (e). The scale-space images in Fig. 2 correspond to the speed functions of trajectories A1 and B1 (a rotated version of A1). Since the speed function is rotation invariant, A1 and B1 match well, and as a result the difference picture in Fig. 2(e) is dark. Figure 3 shows two scale-space images that correspond to the speed functions of trajectories A1 and B3 (which is distinct from A1). Since these scale-spaces are different, their diffused zero-crossing potentials also differ and the difference picture in Fig. 3(e) shows more bright portions.

Algorithm A, for matching single trajectories, is given in Fig. 4. The model base contains the diffused zero-crossing potentials of the speed (α_s) and direction (α_d) functions. The input is a trajectory which will be compared with the information in the model. A match score is produced which indicates how closely the two trajectories match.

The first step in the algorithm is to decompose the trajectory into two 1D signals: speed and direction. The second step constructs the scale-space images of the speed and direction signals by convolving each

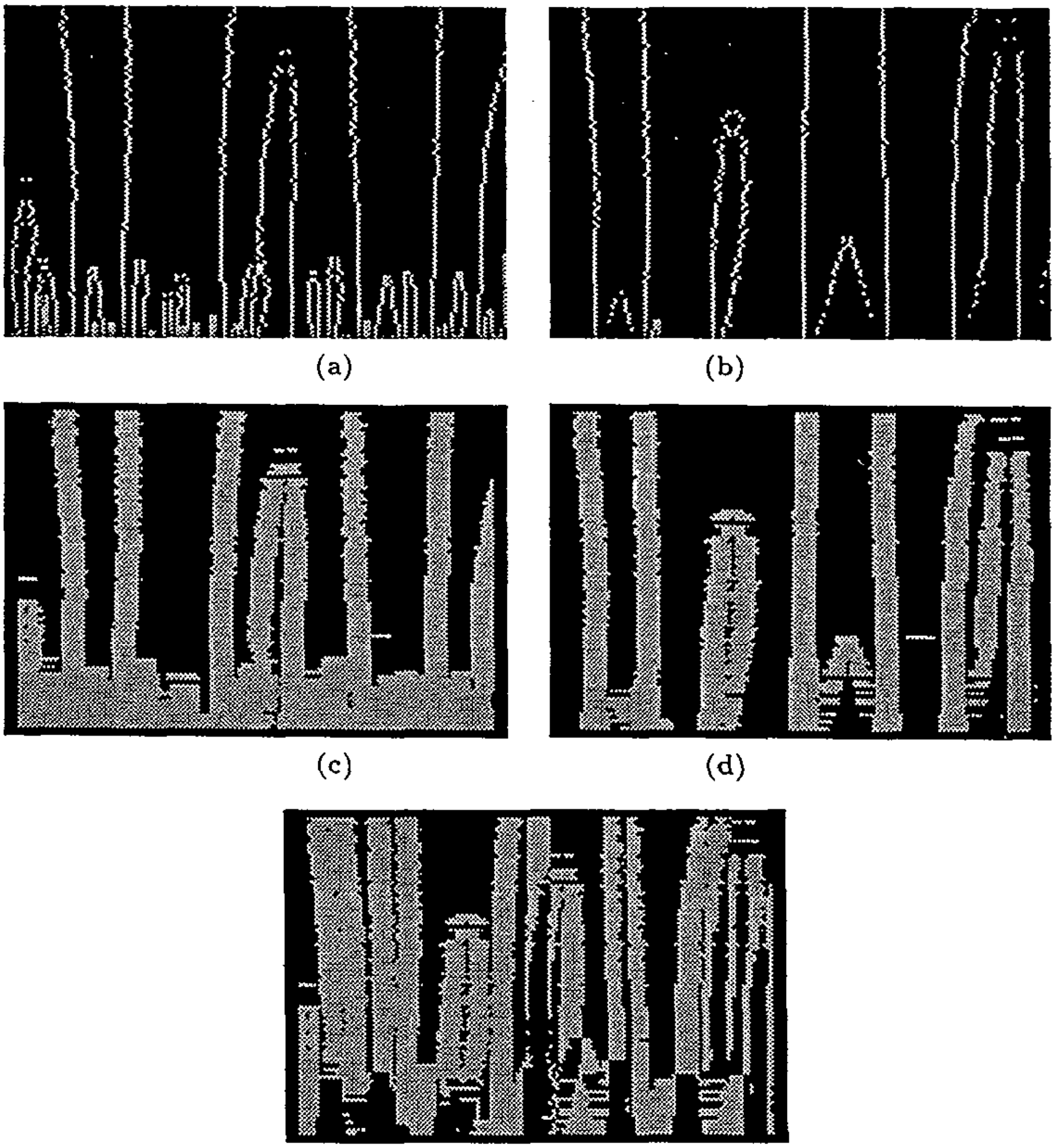


Fig. 3. (a) The zero-crossing potential of the speed scale-space of trajectory A1 (shown in Fig. 8(a)); (b) the zero-crossing potential of the speed scale-space of trajectory B3, which is distinct from trajectory A1; (c) diffused version of (a); (d) diffused version of (b); (e) difference picture between (c) and (d).

signal with the second derivative of the Gaussian over a range of σ values. This produces two 2D arrays, which we call β_s and β_d, where one dimension is the frame number and the other dimension is the σ value used in the Gaussian mask. In our implementation, we used σ values from 1 to 10 in steps of 0.1. The arrays are the same size as the original image and each location in the array holds the zero-crossing potential of the corresponding location in the image. In the third step of the algorithm, the zero-crossing potentials in β_s and β_d are diffused and the results stored in arrays γ_s and γ_d. The diffusion is achieved by distributing the zero-crossing potential at each point into a small neighborhood using a 2D Gaussian mask with $\sigma = 1$. The fourth step is performed to take care of scaling, and is based on the fact that the zero-crossing potential is proportional to the step size. The terms, $(\sum\sum\alpha_s[n,\sigma])/(\sum\sum\gamma_s[n,\sigma])$ and $(\sum\sum\alpha_d[n,\sigma])/(\sum\sum\gamma_d[n,\sigma])$, are used as the scaling factors for speed and direction. The entries of γ_s and γ_d are multiplied by the scaling factors and the results are stored back in those arrays. Step five finds the difference between

TRAJECTORY MATCHING ALGORITHM

Model for matching: The model trajectory speed signal α_s and direction signal α_d are stored as diffused zero-crossing potentials in arrays $\alpha_s[n,\sigma]$ and $\alpha_d[n,\sigma]$, respectively.

Input: A trajectory from the input image sequence.

1. Compare the speed and direction *signals from the input trajectory.*
2. Generate the speed and direction scale-space images by convolving the speed and direction signals with the second derivative of the Gaussian over a range of σ values. This produces two 2D arrays, $\beta_s[n,\sigma]$ and $\beta_d[n,\sigma]$, containing the zero-crossing potentials at each point. In these arrays, points which are not zero-crossings will hold a zero value.
3. Diffuse the zero-crossing potentials β_s and β_d using a 2D Gaussian mask with $\sigma = 1$, and store the result in γ_s and γ_d.
4. Scale the values in γ_s and γ_d, respectively, by the scaling factors $(\sum\sum\alpha_s[n,\sigma])/(\sum\sum\gamma_s[n,\sigma])$ and $(\sum\sum\alpha_d[n,\sigma])/(\sum\sum\gamma_d[n,\sigma])$.
5. Perform an element by element subtraction of the α and γ arrays and store the absolute value in a pair of arrays, ϵ_s for speed and ϵ_d for direction.
6. Compute the match scores for speed and direction as $1-(\sum\sum|\epsilon_s(n,\sigma)|)/(2*|\sum\sum\alpha_s(n,\sigma)|)$ and $1-(\sum\sum|\epsilon_d(n,\sigma)|)/(2*|\sum\sum\alpha_d(n,\sigma)|)$, respectively.
7. Average the speed and direction match scores to generate the overall match score.

A perfect match between trajectories will produce a match score of 1.

Fig. 4. Algorithm A.

the input zero-crossing potentials in γ_s and γ_d and the model zero-crossing potentials in α_s and α_d. Two new arrays, ϵ_s and ϵ_d, are produced by a simple element-by-element subtraction of the α and γ arrays. Step six computes the speed match score and the direction match score. The match function is based on the observation that the maximum possible value of $\sum\sum|\epsilon_s(n,\sigma)|$ is $2*\sum\sum|\alpha_s(n,\sigma)|$, and the maximum possible value of $\sum\sum|\epsilon_d(n,\sigma)|$ is $2*\sum\sum|\alpha_d(n,\sigma)|$. Finally, in step seven, the overall match score is computed from the average of the speed and direction scores.

5. MATCHING MULTIPLE TRAJECTORIES

In the previous section we proposed a method for matching pairs of single trajectories that used only motion information. A single motion trajectory does not carry any information about the shape† of the moving object. When a number of points belonging to a single object are tracked, the trajectories of those points define the motion of the object as well as its shape.

In this section we propose a scheme for matching sets of trajectories that uses both motion and spatial information and can be employed in developing an object recognition system. The method discussed in the previous section will be used to match single trajectories using motion information only. Since our model does not contain information about the shape outline of the object, we will use the Euclidean distances between selected points in the object's trajectories for our spatial information and include that data in the model. In our current implementation, we manually select those points that will provide the best spatial representation for matching purposes. We have left the design of an automatic system for determining these points to a future project.

We compare the spatial information in the model with the Euclidean distances between corresponding points in the trajectories that we are matching. If four points in a trajectory are tracked, we will store

$$\binom{4}{2} = 6$$

Euclidean distances in the model for each frame in the sequence. Figure 5 shows an object with four selected points P_1, P_2, P_3 and P_4. The Euclidean distances between those points (a, b, c, d, e and f) are stored in the model. If the points on an object are rigid with respect to each other in all frames, the Euclidean distance

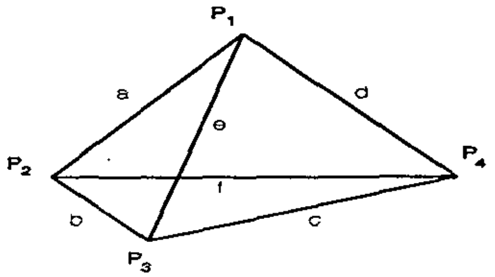


Fig. 5. An object with four points P_1, P_2, P_3 and P_4. The Euclidean distances a, b, c, d, e and f will be stored in the model.

† By shape, we mean the relative spatial orientation of two or more points. The points may or may not have physical edges between them.

TRAJECTORY SET MATCHING ALGORITHM

Input: Trajectory sets S_1 and S_2. Let $\{S_1^1, S_1^2, S_1^3, \ldots, S_1^n\}$ be the component trajectories of S_1 and let $\{S_2^1, S_2^2, S_2^3, \ldots, S_2^n\}$ be the component trajectories of S_2.

Assumptions:

1. S_1 and S_2 both have n component trajectories.
2. $M: S_1 \rightarrow S_2$, a one-to-one, onto function that maps a trajectory from set S_1 to S_2 is known.
3. The weights, $W(S_1^i)$ for motion match and $W(S_1^i, S_1^j)$ for spatial match, are stored in the model.

Algorithm:

1. Using Algorithm A, compute the motion match score $\xi(S_1^i, S_2^{M(S_1^i)})$ between trajectories S_1^i and $S_2^{M(S_1^i)}$, $n \geq i \geq 1$.
2. Compute the spatial match score $\epsilon(\{S_1^i, S_1^j\}, \{S_2^{M(S_1^i)}, S_2^{M(S_1^j)}\})$ by using spatial information to match each pair of trajectories $\{S_1^i, S_1^j\}$ and $\{S_2^{M(S_1^i)}, S_2^{M(S_1^j)}\}$, $n \geq i \geq 1$, $n \geq j \geq 1$, and $i \neq j$.
3. Compute the overall match score $\zeta(S_1, S_2)$ of trajectory sets S_1 and S_2.

$$\zeta(S_1, S_2) = \sum_{i=1}^{i=n} W(S_1^i)\xi(S_1^i, S_2^{M(S_1^i)}) + \sum_{i=1, j=1, i \neq j}^{i=n, j=n} W(S_1^i, S_1^j)\varepsilon(\{S_1^i, S_1^j\}, \{S_2^{M(S_1^i)}, S_2^{M(S_1^j)}\}).$$

Fig. 6. Algorithm B.

between the points will be the same and we need only consider the distances in one frame. However, if the points are non-rigid, we will have to determine the distances in all frames. Let $\Xi(X, Y)$ be the average Euclidean distance between corresponding points in the two trajectories X and Y. We define ε below as the normalized difference between these average Euclidean distances and consider it to be a measure of the spatial match between the trajectory pairs $\{T1, T2\}$ and $\{T3, T4\}$.

$$\varepsilon(\{T1, T2\}, \{T3, T4\}) = 1 - \min\left(1, \left|\frac{\Xi(T3, T4) - \Xi(T1, T2)}{\Xi(T1, T2)}\right|\right).$$

Let S_1 and S_2 be sets of n trajectories and let $\{S_1^1, S_1^2, S_1^3, \ldots, S_1^n\}$ be the component trajectories of S_1 and $\{S_2^1, S_2^2, S_2^3, \ldots, S_2^n\}$ be the component trajectories of S_2. Assume that we know $M: S_1 \rightarrow S_2$, a one-to-one, onto function that maps a trajectory from set S_1 to S_2. We define $\xi(S_1^i, S_2^{M(S_1^i)})$ as the motion match score between trajectories S_1^i and $S_2^{M(S_1^i)}$ and $\varepsilon(\{S_1^i, S_1^j\}, \{S_2^{M(S_1^i)}, S_2^{M(S_1^j)}\})$ as the spatial match score.

We will store constant values in the model to represent weight factors to be applied to the motion and spatial match terms. These weights will allow adjusting the proportions of the motion and spatial match scores that are included in the overall score. We define $W(S_1^i)$ as the set of weights for the motion match

	A1	B1	B2	B3	B4
A1	(1.00,1.00,1.00)	(0.77,0.58,0.68)	(0.40,0.38,0.39)	(0.07,0.42,0.25)	(0.99,1.00,1.00)
B1		(1.00,1.00,1.00)	(0.36,0.37,0.37)	(0.05,0.43,0.24)	(0.76, 0.58, 0.67)
B2			(1.00,1.00,1.00)	(0.38,0.23,0.31)	(0.38,0.22,0.3)
B3				(1.00,1.00,1.00	(0.06, 0.42, 0.24)

(a)

Trajectory	Ordered Match
A1	A1, B4, B1, B2, B3
B1	B1, A1, B4, B2, B3
B2	B2, A1, B1, B3, B4
B3	B3, B2, A1, B4, B1

(b)

Fig. 7. (a) Match scores of comparing trajectories A1, B1 (a rotated version of A1), B2 (a noisy version of A1), B3 (a distinct trajectory from A1), and B4 (a magnified version of A1 by a factor of 2). Each entry is a triple (a, b, c), "a" the match score of speed scale-spaces, "b" the match score of direction scale-spaces, and "c" the overall match score ((a + b)/2); (b) the matches for a given trajectory ordered on the match score.

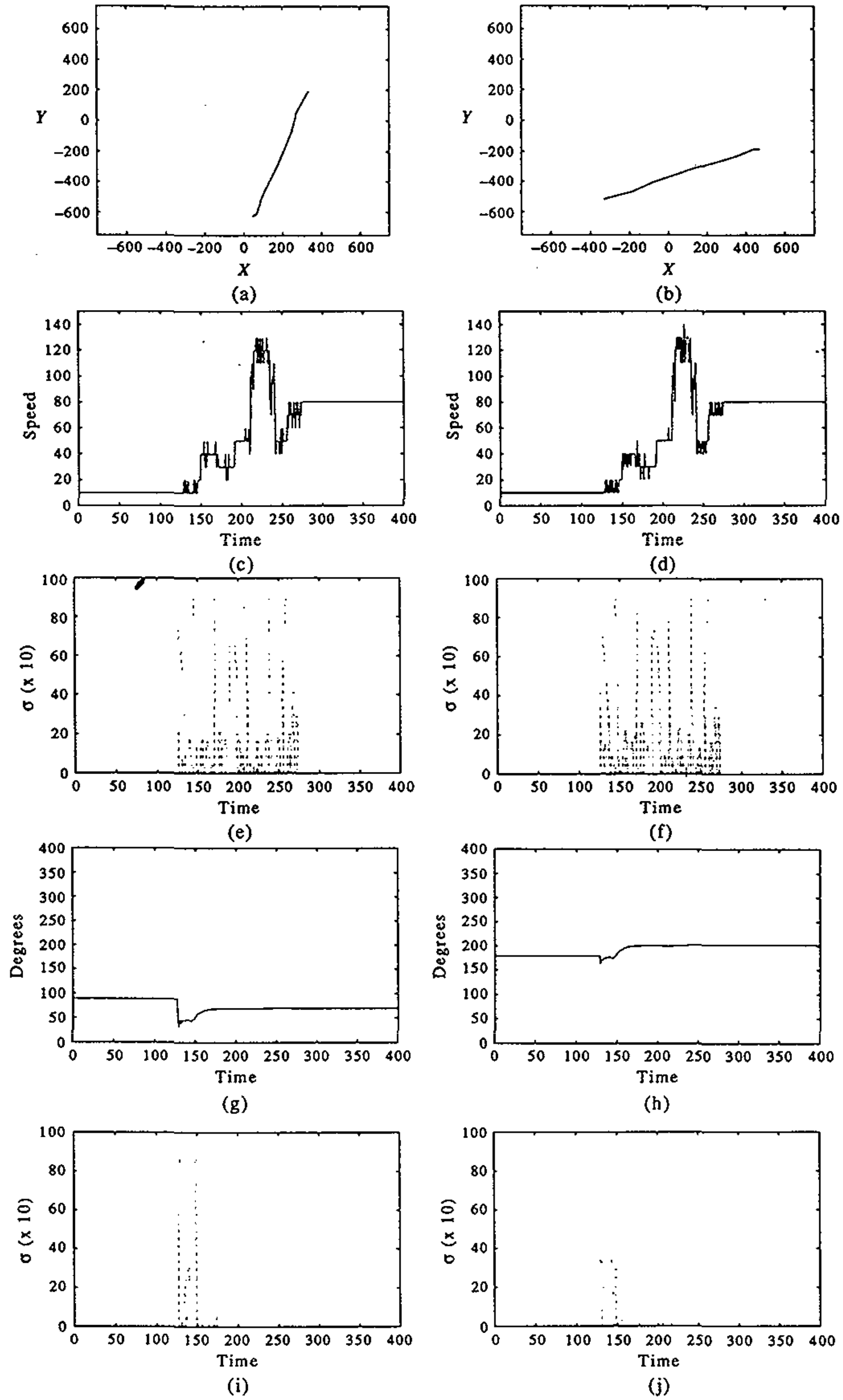


Fig. 8. (a) Trajectory A1; (b) trajectory B1 (a rotated version of A1); (c) speed signal of trajectory A1; (d) speed signal of trajectory B1; (e) scale-space of speed signal of trajectory A1; (f) scale-space of speed signal of trajectory B1; (g) direction signal of trajectory A1; (h) direction signal of trajectory B1; (i) scale-space of direction signal of trajectory A1; (j) scale-space of direction signal of trajectory B1.

terms and $W(S_1^i, S_1^j)$ as the set of weights for the spatial match terms.

The overall match score $\xi(S_1, S_2)$ between trajectory sets, S_1 and S_2, will be computed as

$$\zeta(S_1,S_2) = \sum_{i=1}^{i=n} W(S_1^i)\xi(S_1^i, S_2^{M(S_1^i)}) + \sum_{i=1,j=1,i\neq j}^{i=n,j=n} W(S_1^i, S_1^j) \times \varepsilon(\{S_1^i, S_1^j\}, \{S_2^{M(S_1^i)}, S_2^{M(S_1^j)}\}).$$

Figure 6 shows Algorithm B for matching multiple trajectories.

6. RESULTS

In this section we present the results of applying both Algorithm A and Algorithm B on synthetic and real sequences. We show that both algorithms performed quite well on these test sequences.

6.1. *Algorithm A*

Figure 7(a) shows the results of comparing trajectories A1, B1, B2, B3 and B4. B1 is a rotated version of A1, B2 is a noisy and rotated version of A1, B3 is a trajectory distinct from A1, and B4 is a version of A1 magnified by a factor of two. Each entry is a triple (a, b, c), where "a" is the match score of speed scale-spaces, "b" is the match score of direction scale-spaces, and "c" is the overall match score given by (a + b)/2. If the two scale-spaces are identical, the algorithm gives a perfect match score of 1. Figure 7(b) shows the match scores ordered from best to worst. Figure 8(a) shows trajectory A1 and Fig. 8(b) shows trajectory B1 (a rotated version of A1). Figures 8(c)–(f) show the speed signals of trajectories A1 and B1 and their respective scale-space images. The direction signals of trajectories A1 and B1 and their scale-space images are shown in Figs 8(g)–(j). Since B1 differs from A1 only in orientation, its speed and direction signals should be identical to those of A1. There are slight differences in the signals caused by truncation errors since the signals are scaled by a constant factor of ten upon input to the program. The truncation error in angle computation is greater, therefore the errors between the direction signals of B1 and A1 are much more than the errors between their respective speed signals. Consequently, the scale-space images of speed (shown in Figs 8(e) and (f)) resemble each other more closely than the scale spaces of direction (shown in Figs 8(i) and (j)). The match score of speed signals was 0.77 and the match score of direction signals was 0.58.

It is clear from Fig. 7 that the rotated (B1) and magnified (B4) versions of A1 both produced good match scores. As expected, the distinct trajectory (B3) generated a low match score while the rotated and noisy version (B2) did not match as well as either B1 or B4, but much better than B3.

To produce real data to test our algorithms, we used a video camera to record scenes of people walking. We videotaped a person, K, walking at two different times, and generated two distinct image sequences. We also videotaped a person, W, and generated a single image sequence. From each image sequence, we produced a set of trajectories by manually tracking nine body points. Figure 9 shows the points tracked and our labels for them.

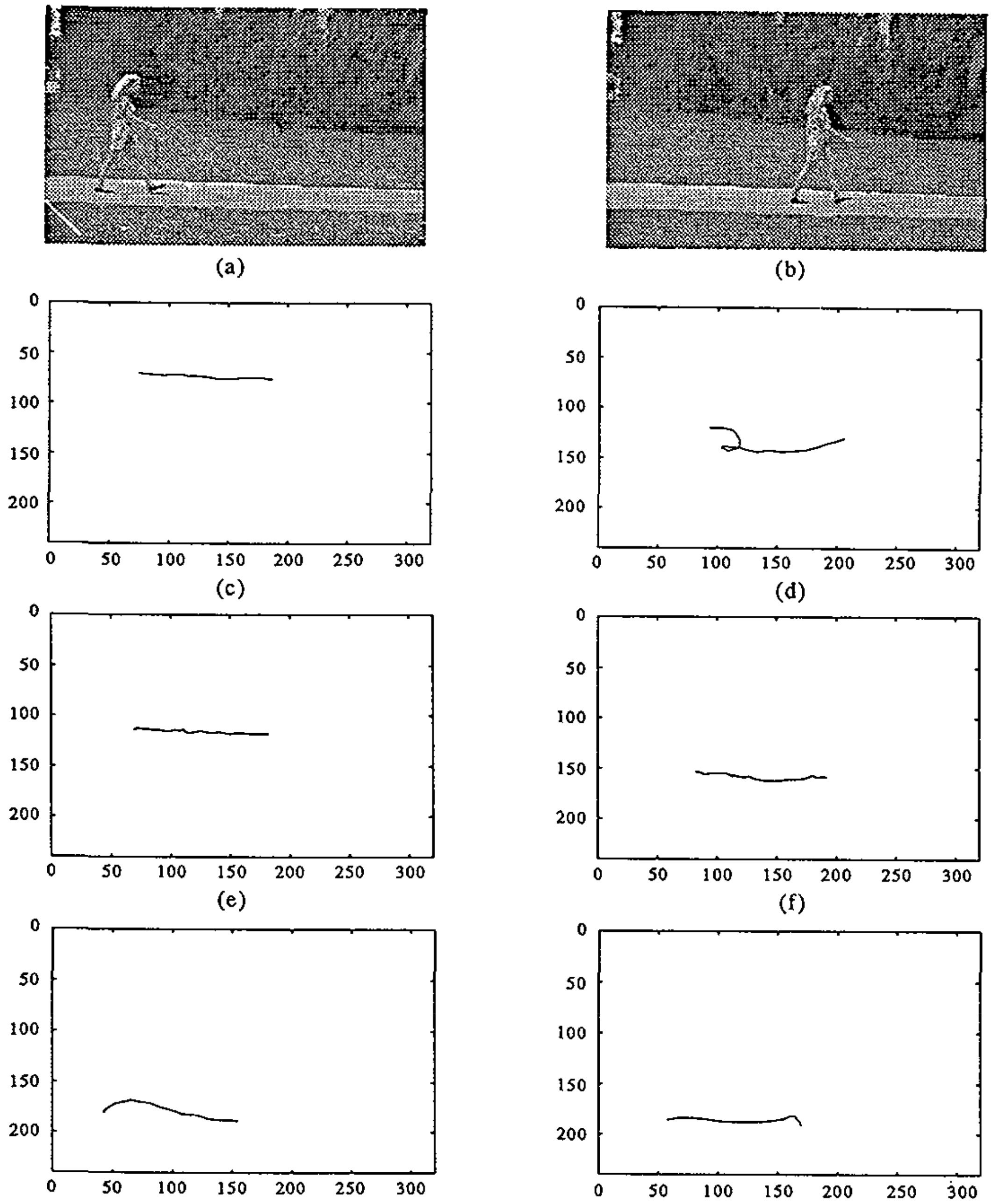
Figure 13 shows the match scores between sequences K^1 and K^2 and Fig. 14 shows the scores between sequences K^1 and W^1. We used sequence K^1 as the model, therefore the rows of each table show the results of comparing selected body points from sequence K^1 with corresponding body points from sequences K^2 and W^1. The *lknee* and *rknee*, *lheel* and *rheel*, and *ltoe* and *rtoe* produced similar results, so we included only one of each in this report. As expected, body points from sequence K^1 produced better match scores to the corresponding body points in sequence K^2 than to dissimilar body points from either sequence. As the graphs in Figs 10–12 indicate, the *head*, *elbow* and *knee* have relatively flat trajectories and, as expected, produce higher match scores to each other than to more active body points such as *hand*, *heel* and *toe*.

6.2. *Algorithm B*

To test the algorithm for matching multiple trajectories, the walking sequences of persons K and W

Body Part	sequence K^1	sequence K^2	sequence W^1
top of head	K^1_{head}	K^2_{head}	W^1_{head}
right elbow	K^1_{elbow}	K^2_{elbow}	W^1_{elbow}
right hand	K^1_{hand}	K^2_{hand}	W^1_{hand}
right knee	K^1_{rknee}	K^2_{rknee}	W^1_{rknee}
right heel	K^1_{rheel}	K^2_{rheel}	W^1_{rheel}
right toe	K^1_{rtoe}	K^2_{rtoe}	W^1_{rtoe}
left knee	K^1_{lknee}	K^2_{lknee}	W^1_{lknee}
left heel	K^1_{lheel}	K^2_{lheel}	W^1_{lheel}
left toe	K^1_{ltoe}	K^2_{ltoe}	W^1_{ltoe}

Fig. 9. Body points tracked in sequences K^1, K^2 and W^1.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 10. Image sequence K^1 in which person K is walking. There are 30 frames in this sequence. (a), (b) First and last frames of the sequence K^1; (c) trajectory of K^1_{head}; (d) trajectory of K^1_{hand}; (e) trajectory of K^1_{elbow}; (f) trajectory of K^1_{lknee}; (g) trajectory of K^1_{rheel}; (h) trajectory of K^1_{rtoe}.

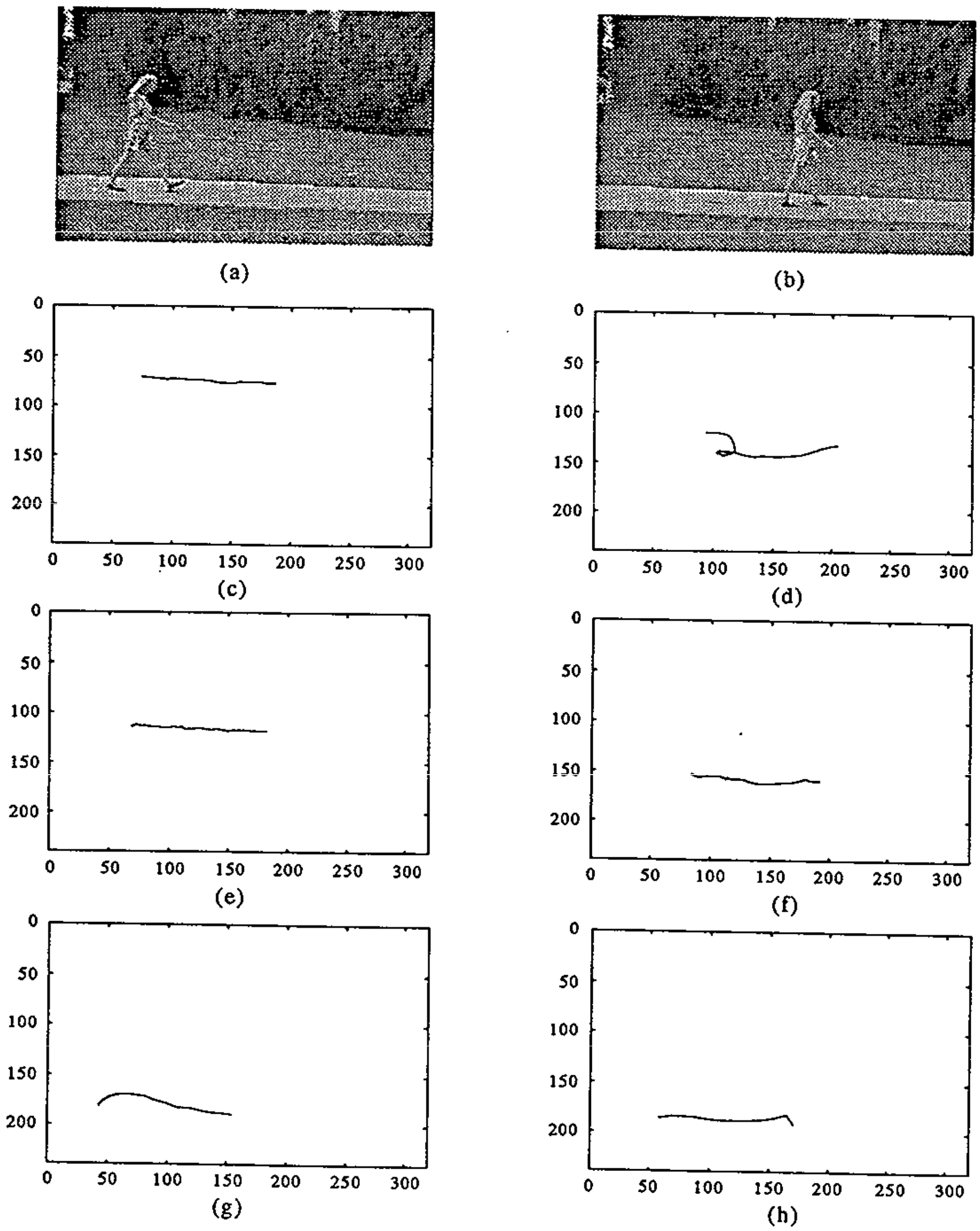
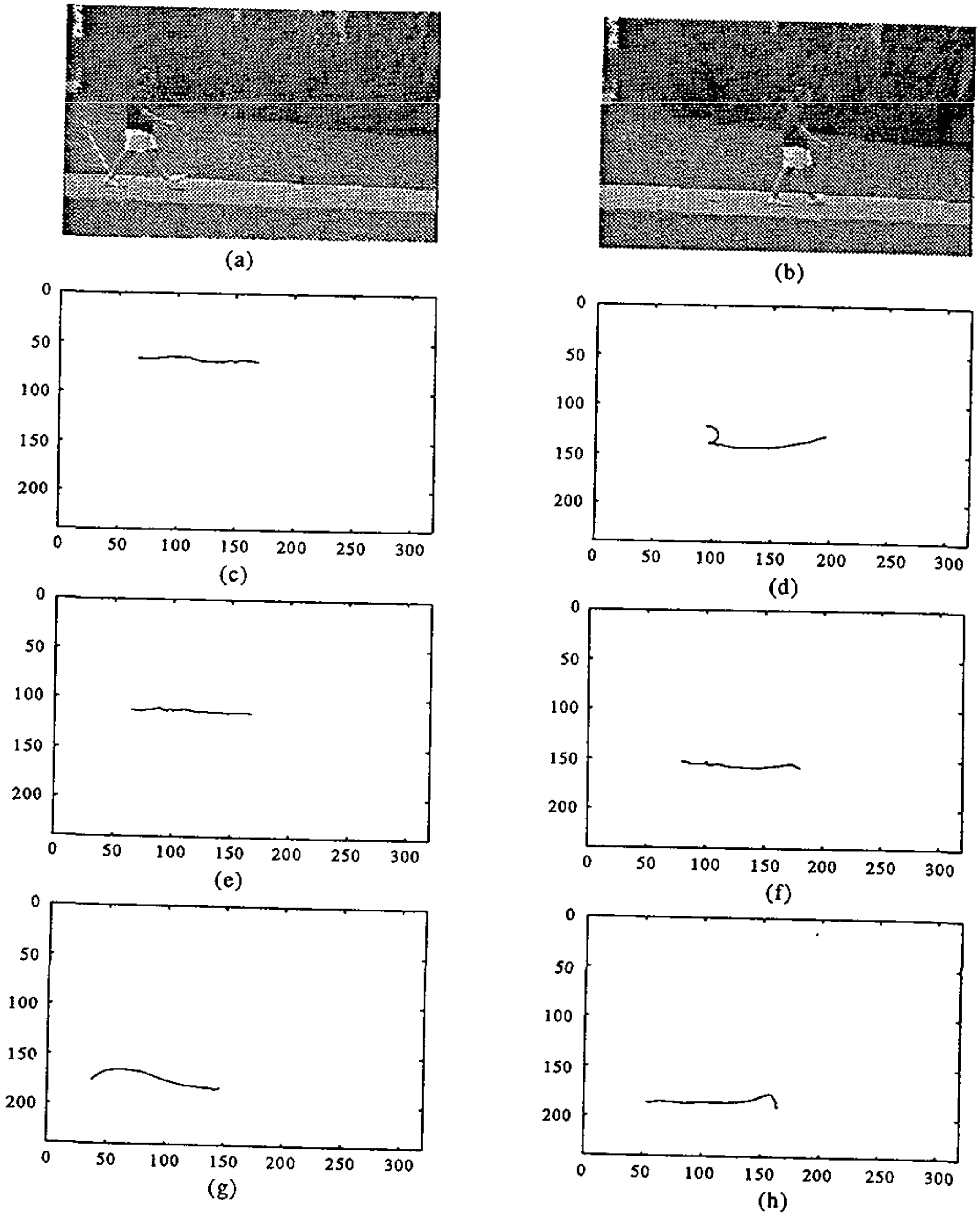


Fig. 11. Image sequence K^2 in which person K is walking. There are 30 frames in this sequence. (a), (b) First and last frames of the sequence K^2; (c) trajectory of K^2_{head}; (d) trajectory of K^2_{hand}; (e) trajectory of K^2_{elbow}; (f) trajectory of K^2_{lknee}; (g) trajectory of K^2_{rheel}; (h) trajectory of K^2_{rtoe}.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 12. Image sequence W^1 in which person W is walking. There are 30 frames in this sequence. (a), (b) First and last frames of the sequence W^1; (c) trajectory of W^1_{head}; (d) trajectory of W^1_{hand}; (e) trajectory of W^1_{elbow}; (f) trajectory of W^1_{lknee}; (g) trajectory of W^1_{rheel}; (h) trajectory of W^1_{rtoe}.

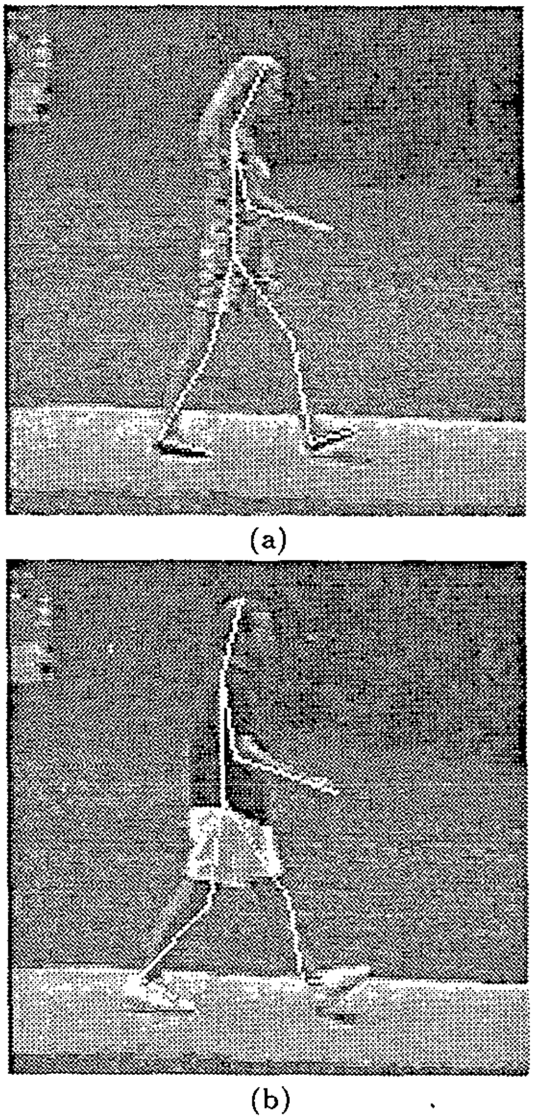
	K^2_{head}	K^2_{hand}	K^2_{elbow}	K^2_{lknee}	K^2_{rheel}	K^2_{rtoe}
K^1_{head}	(0.60,0.73,0.67)	(0.28,0.13,0.21)	(0.35,0.23,0.29)	(0.32,0.15,0.24)	(0.16,0.15,0.16)	(0.13,0.27,0.20)
K^1_{hand}	(0.32,0.16,0.24)	(0.91,0.91,0.91)	(0.23,0.47,0.35)	(0.25,0.39,0.32)	(0.10,0.03,0.07)	(0.16,0.08,0.12)
K^1_{elbow}	(0.35,0.24,0.30)	(0.20,0.51,0.36)	(0.72,0.70,0.70)	(0.51,0.45,0.48)	(0.13,0.07,0.10)	(0.10,0.09,0.10)
K^1_{lknee}	(0.39,0.15,0.27)	(0.26,0.39,0.33)	(0.42,0.41,0.42)	(0.80,0.86,0.83)	(0.14,0.04,0.09)	(0.09,0.15,0.12)
K^1_{rheel}	(0.15,0.19,0.17)	(0.11,0.05,0.08)	(0.11,0.08,0.10)	(0.16,0.04,0.10)	(0.91,0.89,0.90)	(0.58,0.50,0.54)
K^1_{rtoe}	(0.10,0.26,0.18)	(0.17,0.04,0.11)	(0.08,0.21,0.15)	(0.11,0.16,0.14)	(0.56,0.41,0.49)	(0.73,0.82,0.78)

Fig. 13. Table showing the match scores produced by comparing trajectories from sequence K^1 with trajectories from sequence K^2. As expected, there is generally a higher match score between the same body points, i.e. K^1_{hand} has a better match with K^2_{hand} than with K^2_{rtoe}. Note that since similar body parts (K^1_{lknee} and K^1_{rknee}) produced similar scores, we omitted them from this table.

	W^1_{head}	W^1_{hand}	W^1_{elbow}	W^1_{lknee}	W^1_{rheel}	W^1_{rtoe}
K^1_{head}	(0.48,0.24,0.36)	(0.45,0.31,0.38)	(0.19,0.17,0.18)	(0.23,0.25,0.24)	(0.16,0.14,0.15)	(0.17,0.37,0.27)
K^1_{hand}	(0.32,0.16,0.24)	(0.28,0.20,0.24)	(0.20,0.15,0.18)	(0.18,0.19,0.19)	(0.20,0.04,0.12)	(0.18,0.16,0.17)
K^1_{elbow}	(0.28,0.16,0.22)	(0.27,0.37,0.32)	(0.17,0.19,0.18)	(0.25,0.24,0.25)	(0.17,0.07,0.12)	(0.12,0.11,0.12)
K^1_{lknee}	(0.43,0.15,0.29)	(0.40,0.42,0.41)	(0.33,0.25,0.29)	(0.13,0.29,0.21)	(0.25,0.07,0.17)	(0.11,0.16,0.14)
K^1_{rheel}	(0.20,0.13,0.17)	(0.27,0.05,0.16)	(0.11,0.06,0.09)	(0.12,0.06,0.09)	(0.48,0.48,0.48)	(0.41,0.31,0.36)
K^1_{rtoe}	(0.11,0.28,0.20)	(0.17,0.06,0.12)	(0.08,0.17,0.13)	(0.12,0.18,0.15)	(0.30,0.48,0.39)	(0.46,0.50,0.48)

Fig. 14. Table showing the match scores produced by comparing trajectories from sequence K^1 with the trajectories from sequence W^1. Although some scores indicate similar trajectories by similar body points on different people, for example K^1_{rtoe} and W^1_{rtoe}, the match scores are much lower than between the same body points in the sequences K^1 and K^2 which are from the same person.

shown in Figs 10–12 were used. We tracked the trajectories of nine distinct body points (see Fig. 9) from each sequence. Some body points, such as *hand*, *heel* and *toe* are rich in motion information and are useful in producing a unique model of the motion of an individual. The trajectories of other points, such as *head*, *elbow* and *knee*, are relatively flat and less important to matching motion, but provide useful spatial information. We include the *head* trajectory in our model since it is the most valuable in terms of spatial information. However, due to the limited range of motion that the *head* exhibits, the match scores are low, even between sequences from the same person. We assume the trajectory set $\{K^1_{head}, K^1_{hand}, K^1_{rheel}\}$ to be the model. Our goal is to compare this set with another set from the same person (K^2), and with a set from a different person (W^1). Figure 16 shows the spatial configuration of the nine body points that are tracked in each frame of the sequences. The points are connected by lines to generate stick figures that model the human form. The shoulder and hip points in each figure were approximated and are included to produce a more natural looking figure. The spatial relationship of the body points can be seen in these drawings as well as the relative motion of those points. As this time, we weight the match score terms equally. We did not address the issue of adjusting the relative proportion of motion and spatial scores in the overall score, but intend to pursue this area in the future.


(a)

(b)

Fig. 15. Images with stick figure drawing superimposed to indicate spatial relationship between body points: (a) image from sequence K^1; (b) image from sequence W^1. Note: hip and shoulder points approximated and added for clarity, also note that the *heel* points were taken from the upper heel of the shoe to improve the accuracy of tracking those points.

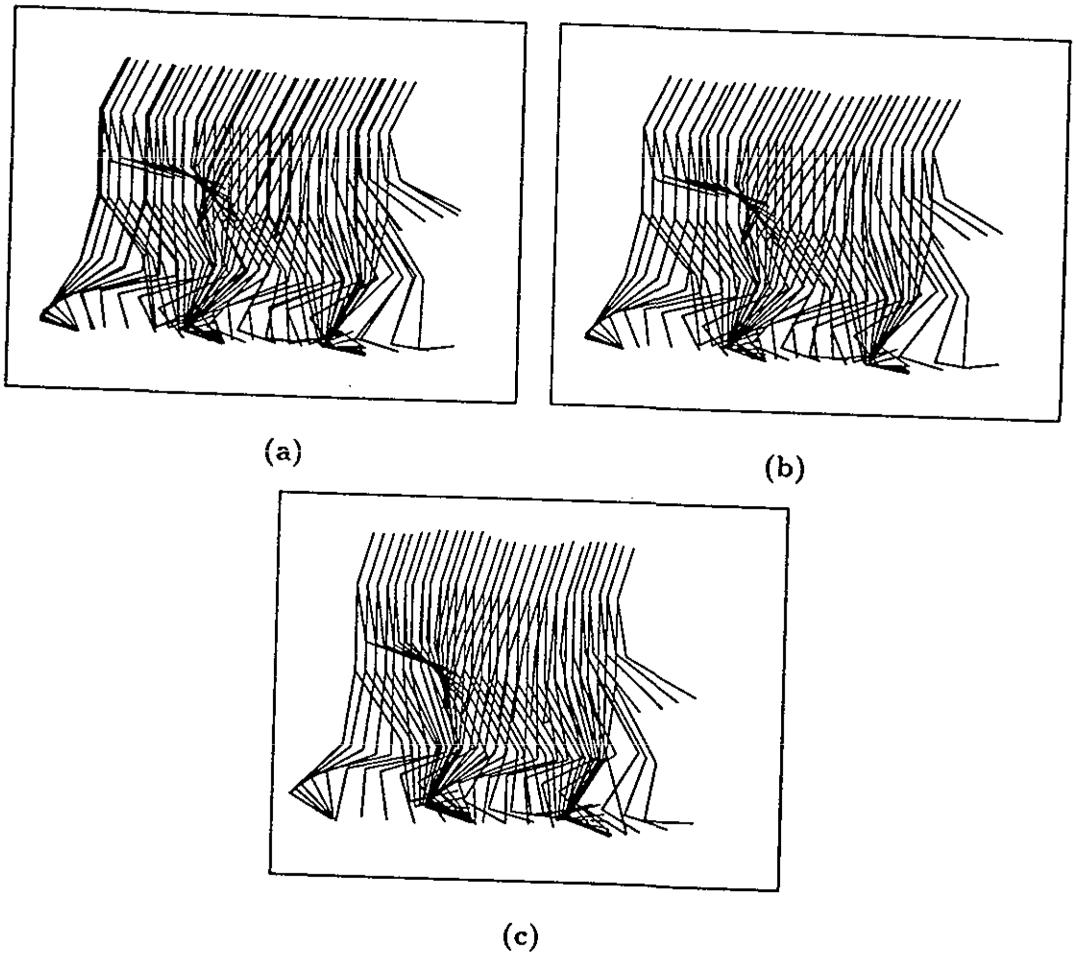


Fig. 16. Stick figure drawings showing the nine body points tracked for all 30 frames in: (a) sequence K^1; (b) sequence K^2; (c) sequence W^1. Note: hip and shoulder points approximated and added for clarity.

Motion Match Score

	(K^2_{head})
(K^1_{head})	0.67

	K^2_{hand}
K^1_{hand}	0.91

	K^2_{rheel}
K^1_{rheel}	0.90

Spatial Match Score

	(K^2_{head}, K^2_{hand})
(K^1_{head}, K^1_{hand})	0.99

	$(K^2_{head}, K^2_{rheel})$
$(K^1_{head}, K^1_{rheel})$	0.99

	$(K^2_{hand}, K^2_{rheel})$
$(K^1_{hand}, K^1_{rheel})$	0.99

$$\zeta(K^1, K^2) = (0.67 + 0.91 + 0.90 + 0.99 + 0.99 + 0.99) \ / \ 6 = 0.91$$

Fig. 17. Match score of trajectory sets $\{K^1_{head}, K^1_{hand}, K^1_{rheel}\}$ and $\{K^2_{head}, K^2_{hand}, K^2_{rheel}\}$. The six terms are given equal weight by simple averaging.

Motion Match Score

	(W^1_{head})
(K^1_{head})	0.36

	W^1_{hand}
K^1_{hand}	0.24

	W^1_{rheel}
K^1_{rheel}	0.48

Spatial Match Score

	(W^1_{head}, W^1_{hand})
(K^1_{head}, K^1_{hand})	0.88

	$(W^1_{head}, W^1_{rheel})$
$(K^1_{head}, K^1_{rheel})$	0.95

	$(W^1_{hand}, W^1_{rheel})$
$(K^1_{hand}, K^1_{rheel})$	0.95

$$\zeta(K^1, W^1) = (0.36 + 0.24 + 0.48 + 0.88 + 0.95 + 0.95) \;/\; 6 = 0.64$$

Fig. 18. Match scores of trajectory sets $\{K^1_{head}, K^1_{hand}, K^1_{rheel}\}$ and $\{W^1_{head}, W^1_{hand}, W^1_{rheel}\}$. The six terms are given equal weight by simple averaging.

Algorithm B produced a match score of 0.91 for trajectory set $\{K^1_{head}, K^1_{hand}, K^1_{rheel}\}$ and trajectory set $\{K^2_{head}, K^2_{hand}, K^2_{rheel}\}$ of the same person, and a match score of 0.64 for trajectory sets $\{K^1_{head}, K^1_{hand}, K^1_{rheel}\}$ and $\{W^1_{head}, W^1_{hand}, W^1_{rheel}\}$ corresponding to two different persons. In both cases the mapping from one set of trajectories to the other was assumed to be known. The match scores are tabulated in Figs 17 and 18 and show that K1 matches K2 significantly better than it does W1.

7. CONCLUSIONS

In this paper, we presented a method for matching motion trajectories that can be used to incorporate the direct use of motion in an object recognition system. We utilize motion information from extended trajectories generated by points on a moving object. A simple and efficient algorithm for trajectory matching was described which uses a scale-space representation of trajectory speed and direction. The algorithm was then extended to include spatial information from points on the object to assist in matching multiple trajectories. We demonstrated the performance of both algorithms on real and synthetic cases.

Acknowledgement—The research reported here was supported by the Florida HiTech Council under grant 65 02 731, and the National Science Foundation under grants IRI 8900798 and CDA 9100898.

REFERENCES

1. M. Shah, K. Rangarajan and P.-S. Tsai, Generation and segmentation of motion trajectories, *Proc. Int. Conf. on Pattern Recognition*, pp. 74–77, August (1992).
2. J. T. Todd, Visual information about rigid and nonrigid motion: a geometric analysis, *J. Exp. Psychol.* **8**, 238–252 (1982).
3. J. E. Cutting, *Motion Representation and Perception*, pp. 264–270. North-Holland, New York (1986).
4. G. Johansson, Visual perception of biological motion, *Scient. Am.* **232**, 76–89 (1975).
5. M. Allmen and C. Dyer, Cyclic motion detection using spatiotemporal surfaces and curves, *Proc. 10th Int. Conf. on Pattern Recognition*, IEEE Computer Society, pp. 365–370, June (1990).
6. D. Koller, N. Heinze and H. Nagel, Algorithmic characterization of vehicle trajectories from image sequences by motion verbs, *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 90–95, June (1991).
7. D. W. Murray, D. A. Castelow and B. F. Buxton, From image sequences to recognized moving polyhedral objects, *Int. J. Comput. Vision* **3**, 181–208 (1989).
8. D. Hogg, Finding a known object using a generate and test strategy, *Parallel Architectures and Computer Vision*, J. Page, ed., pp. 119–132. Oxford Science Publications, Oxford (1988).

About the Author—KRISHNAN RANGARAJAN received the M.Tech. degree in computer science from IIT, Delhi, India, in 1985 and the Ph.D. degree in computer science from University of Central Florida, Orlando, in 1990. Research interests include computer vision, databases and software engineering. Currently he is on contract with Bell Labs. He has also held positions at the Centre for AI & Robotics, Bangalore, India and the Computer Maintenance Corporation, India.

About the Author—WILLIAM ALLEN received the B.S. in computer science from the University of Central Florida in December 1992 and began graduate studies in January 1993. During the 1991–92 academic year, he participated in an Undergraduate Research Experience in Computer Vision sponsored by the National Science Foundation. His research interests include motion analysis and object recognition.

About the Author—MUBARAK SHAH received his Ph.D. from Wayne State University in 1986. Since 1986 he has been with the University of Central Florida, where he is currently an Associate Professor of Computer Science. Professor Shah has done research in dynamic scene analysis, scale space, multiresolution edge detection, and multisensor fusion. He has served on the program committees of the Machine Vision and Robotics Conference, and the ACM Computer Science Conference. He has chaired sessions at several conferences, including the International Conference on Pattern Recognition, and the SPIE Applications of AI Conference. He is a regular reviewer for leading journals.