Pergamon

Pattern Recognition, Vol. 31, No. 11, pp. 1589–1600, 1998

0031-3203/98 $19.00 + 0.00

PII: S0031-3203(98)00018-1

ON-LINE HANDWRITING RECOGNITION USING PHYSICS-BASED SHAPE METAMORPHOSIS

IOANNIS PAVLIDIS,[†] RAHUL SINGH[‡,§] and NIKOLAOS P. PAPANIKOLOPOULOS[‡,1]

[†]Honeywell Technology Center, Honeywell Inc., 3660 Technology Drive, MN65-2500, Minneapolis, MN 55418, U.S.A.
[‡]Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building, 200 Union Street SE, Minneapolis, MN 55455, U.S.A.

(*Received* 24 *November* 1996; *in revised form* 6 *February* 1998)

Abstract—We present a new technique based on shape metamorphosis for on-line recognition of handwritten words and simple shapes in a user-dependent setting. The approach includes a segmentation method that does not try to locate letters but instead performs the significantly easier task of locating corners and some key low curvature points. This is part of the method's strategy to see the word as a generic on-line shape. The segmentation points are used to model a cursive word or a hand-drawn line figure by pieces of wire. Shape metamorphosis occurs through stretching and bending of the artificial wire. The amount of energy spent in morphing one shape to another is used as a dissimilarity measure. For any two given shapes an optimal morph can be computed in quadratic time by constraining the metamorphosis to the segmentation points of these shapes. Experiments with multiple subjects indicate that the method can handle collectively cursive words and hand-drawn line figures, both useful forms of communication in pen-based computing. © 1998 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Handwriting recognition   On-line   Shape metamorphosis   Cursive words
Line figures   Pen-based computing

1. INTRODUCTION

The recognition of handwritten patterns like cursive script and line drawings is an open research area with far-reaching applications.[1,2] A particular application area of relevance that has a high technological impact nowadays is the on-line recognition of handwritten words and line drawings.[3] This is due to the ever increasing spread of pen-based computing and its gradual establishment as an alternative or as a complement to the classical keyboard. An effective use of pen-based computing calls for robust and accurate recognition of cursive script and hand-drawn line figures that is challenging to the capabilities of current algorithms and technologies.

The method described in this paper introduces a framework for handling collectively on-line handwritten patterns in a user-dependent setting. A preliminary version of this work was reported in Pavlidis *et al.*[4] It proposes the use of physics-based shape metamorphosis as a potentially powerful way of dealing with the recognition of difficult on-line patterns. Shape metamorphosis is a well-established graphics technique[5,6] that refers to the problem of computing a continuous shape transformation from an initial shape to a target shape. It has widespread applications in computer animation, but it is the first time (to the best of our knowledge) that it is used for recognition purposes. The proposed system uses to its advantage the intuitive fact that two shapes that are quite similar do not have to go through an extensive metamorphosis process in order for one to assume the shape of the other. Thus, the *degree of morphing* between a test pattern and a reference pattern may serve as the matching criterion. The *degree of morphing* is an abstract quality that in our system is substantiated through a physics-based approach to shape metamorphosis first proposed by Sederberg *et al.*[6] Sederberg's approach was developed for animation purposes and has been suitably modified to deal with the pattern recognition task at hand.

Each shape, either this is a cursive word, or a hand-drawn line figure, is considered to be made of a piece of wire. Shape metamorphosis takes place through appropriate stretching and bending of the artificial wire out of which the initial shape is made of. This allows for the problem to be formulated as an energy minimization problem. The energy expended for stretching and bending a wire shape to some other wire shape is the entity that quantifies the *degree of morphing* concept. The metamorphosis is guided by a few key points which are the result of a segmentation process. A novel segmentation algorithm with

[1]Author to whom all correspondence should be addressed. E-mail: npa.pas@cs.umn.edu.

appropriate qualities has been developed to locate the key shape points.

This is one of the first attempts to attack the problem of difficult on-line handwritten patterns in a comprehensive way. Researchers so far attacked the various areas (handled collectively in the present approach), separately. Thus, there are separate methods for on-line cursive script recognition[7–15] and on-line hand-drawn line figure recognition.[16,17] Broadly speaking, however, these methods work either at the global or the local level in the shape. The proposed approach is a hybrid method that combines the advantages of both the global and the local methods.

In this paper, we first present some relevant work conducted in the area of on-line handwriting recognition and compare it with the proposed method (Section 2). Then, in Section 3 an outline of the system is presented. The segmentation algorithm is described in Section 4. Section 5 unveils the shape metamorphosis method and its use in the pattern recognition context. In Section 6, the results from experimental tests are presented and discussed. Finally, in Section 7 the paper is summarized, conclusions are drawn, and future work is outlined.

2. RELEVANT WORK

On-line handwriting recognition is difficult in general due to the highly variable nature of the handwritten patterns. On-line cursive script recognition is particularly difficult because several characters can be written with a single stroke. Most methods for on-line cursive script recognition operate on word units. Most of these break a word into subparts. Such methods could be characterized as local methods. In contrast, a few methods follow the whole-word approach that leaves the words intact and avoids the segmentation problem entirely. Such methods could be characterized as global methods.

There is a large variety of local methods. A common approach is to analyze a word by stroke segments. Then, sequences of stroke segments are used to identify letters.[9] Words have also been analyzed on a letter-by-letter basis.[10] A particularly interesting approach for letter-level word recognition is elastic matching.[15] Elastic matching evaluates recognition at all possible segmentations. Thus, it alleviates the adverse effect that a poor rigid segmentation can have on the recognition process. Stroke-level segmentation is more susceptible to global blindness than letter-level segmentation. Schomaker[12] has made an interesting study comparing the effectiveness of letter level segmentation vs. stroke-level segmentation. The results of this study showed that letter-level segmentation performed better than stroke-level segmentation in the framework of Kohonen maps. In general, local methods achieve high recognition rates because they involve a detailed shape analysis. Their major shortcoming is that they usually lack a global shape perspective. For example, local methods fail to account for handwriting variations due to the influence of one letter on another (co-articulation) because of their stroke- or letter-level vision only. Another shortcoming of the local methods is that letter-level segmentation and stroke-level fusion can be made perfect only if the word is known. Finally, local methods sometimes suffer from combinatorial complexity,[14] something that precludes them from operating in real-time.

Global methods[7,8] achieve lower recognition rates than local methods because they lack a detailed shape analysis component. On the other hand, global methods have the advantage that they maintain a global perspective. They also avoid segmentation entirely. By this, they avoid the problems accompanying segmentation, but at the same time they miss some crucial local information.

A distinct recent trend, in the on-line cursive word recognition research, is the introduction of statistical and neural network techniques from the speech processing domain.[18,11,19,13,14] These techniques give promising results at the expense of extensive training.

The proposed metamorphosis-based recognition approach is a hybrid approach that combines different characteristics from local and global methods. The main idea of the method is that it treats the cursive word as a curve and analyzes it at the local level through an appropriate segmentation algorithm. There is, however, a significant departure from the mainstream segmentation philosophy followed in such cases. The segmentation algorithm does not attempt to locate letters or parts of letters. In contrast, it performs robustly the significantly easier task of locating corners and some key low curvature points. Due to this approach, the method can successfully and uniformly treat not only cursive words, but also hand-drawn line figures. The other important innovation of the method is that it uses the local information gathered from the segmentation process not to recognize individual characters but to produce an energy measure that is of a cumulative global character. This way, the method can account for co-articulation. Furthermore, local variability of detrimental nature is smoothed out because its contribution to the global measure is not of sufficient strength. An example of such detrimental variability is the case of a retrograde stroke that appears sometimes in the writing of a letter. This sort of variability is very difficult to be handled at the local level alone where its effect may look dominant. Finally, the proposed method operates in real-time and can perform well with minimal training in a user-dependent setting.

3. OVERVIEW OF THE ON-LINE HANDWRITING RECOGNITION SYSTEM

The proposed method consists of three parts (see Fig. 1):

1. *Shape sampling and preprocessing*: The handwritten shape is sampled as it is input through

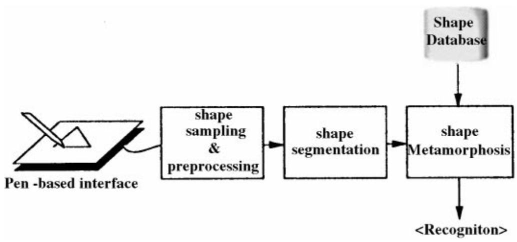

Fig. 1. Metamorphosis-based shape recognition system.

a pen-based interface. The sampled shape undergoes a simple smoothing and normalization operation. The smoothing technique adopted is that suggested by Arakawa.[20] The normalization includes orientation, slant, and size normalization. Orientation and slant normalization are based on the work of Guerfali *et al.*[21] Size normalization is achieved through the mapping of each shape to a unit rectangle. Size normalization is considered necessary because the computation of the stretching energy is size dependent (see Section 5).

2. *Shape segmentation*: After preprocessing, the shape is segmented by identifying points of local curvature extrema. From this point on, the shape is represented by these segmentation points. If it is a reference shape, the set of segmentation points becomes part of the shape database maintained by the system. If it is a test shape, the segmentation points are used to implement the metamorphosis.
3. *Shape metamorphosis*: The test shape is morphed to each and every shape kept in the database. The energy expended for each metamorphosis is used as the matching criterion.

The experimental setup, in which the above method has been implemented, consists of a graphics workstation (SGI Indigo$^{TM}R4000$) and a graphics tablet (WACOM $UD-0608R$) that features an inking stylus. The inking stylus creates an environment for the user that is very much like the traditional pen and paper handwriting environment.

4. SHAPE SEGMENTATION

The result of any segmentation process is highly influenced by the segmentation criteria employed.[22] The two criteria, which are of interest in on-line handwriting recognition are:

- Reduction of the reconstruction error.
- Minimizing cost of recognition (as a function of the number of segmentation points).

Representing a curve by its dominant points (maxima of local curvature) is a popular compromise between these two conflicting criteria. This is based on Attneave's observation that shape information of a curve is concentrated at points of high curvature.[23] Unlike polygonal shapes, where dominant points provide adequate representation, cursive handwriting often consists of strokes with a slowly varying curvature. Segmentation of such shapes based on dominant points alone, leads to poor reconstruction. Our segmentation strategy is based on identifying points on the curve corresponding to *both* local curvature maxima and to local curvature minima. A primary consideration in the detection of local curvature extrema, is the automatic definition of the region of support for each segmentation point. Our approach is based on an algorithm proposed by Brault and Plamondon,[24] for the detection of corner points. This algorithm, as well as the one proposed by us in this paper, allow for automatic definition of region of support for each curvature extrema.

The determination of points representing curvature maxima is done in a way very similar to the method proposed by Brault and Plamondon.[24] In short, the angles $\omega(c+i)$ and $\omega(c-i)$ (see Fig. 2) are computed for each pair of neighbors $c \pm i$ $(i = 1, 2 \ldots)$. In order for a pair $c \pm i$ to belong to the region of support of point c, the following inequalities must be satisfied:

$$\omega(c+i) < \frac{\pi}{2} \quad \text{and} \quad \omega(c-i) < \frac{\pi}{2}. \tag{1}$$

The contribution CF (Cornerness Factor) of each pair $c \pm i$ to the potential corner c is computed by the formula

$$CF(c, i) = \cos(\omega(c+i)) * \cos(\omega(c-i)). \tag{2}$$

The k points that satisfy the inequalities ((1)) constitute the corner domain of point c and their total contribution to the cornerness of point c is computed by

$$TCF(c) = \sum_{i=1}^{k} CF(c, i). \tag{3}$$

The TCF values of the curve points present a very consistent pattern: strings of nonzero values spaced by strings of zero values. Each of the nonzero strings corresponds to a high curvature segment, and the maximum value contained in each such string corresponds to a corner segmentation point.

The points representing curvature minima are computed in a manner conjugate to corner detection. The

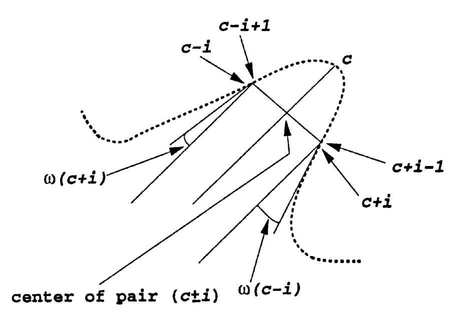

Fig. 2. Geometric model for corner determination as proposed by Brault.[24]

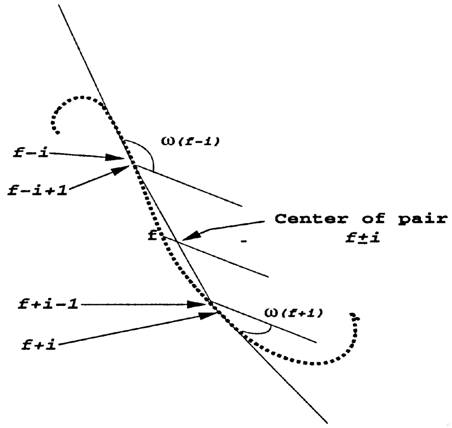

Fig. 3. Geometric model for key low curvature point determination.

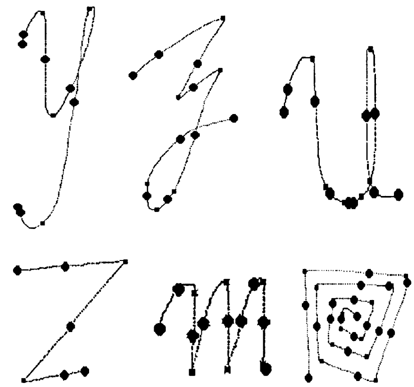

Fig. 4. Segmentation results on some hand-drawn shapes.

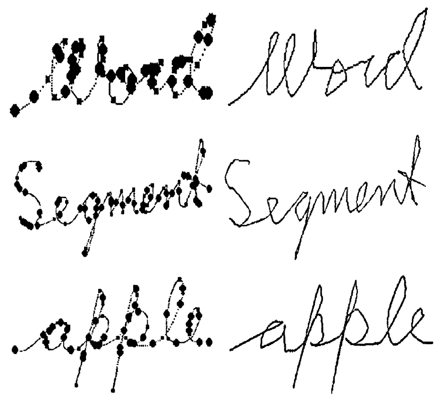

Fig. 5. Cursive word segmentation and reconstruction using the proposed method.

geometric parameters shown in Fig. 3 are the same with those in Fig. 2 and are computed for each pair of neighbors $f \pm i$ $(i = 1, 2 \ldots)$ of every point f of the curve. This time, however, the larger the angles $\omega(f+i)$, and $\omega(f-i)$ are than $\pi/2$, the more the corresponding pair of neighboring points contributes to point f being a curvature minima. As a result, by a suitable analysis of the angles $\omega(f+i)$ and $\omega(f-i)$, one can determine whether or not the pair of points $f \pm i$ is a part of the low curvature domain of f and, in addition, can estimate the importance of the contribution of these points to the low curvatureness of point f. The angles $\omega(f+i)$ and $\omega(f-i)$ must satisfy the following inequalities:

$$\omega(f+i) > \frac{\pi}{2} \quad \text{or} \quad \omega(f-i) > \frac{\pi}{2}. \tag{4}$$

The contribution FF (low curvatureness Factor) of each pair $f \pm i$ to the making of the candidate key low curvature point i is computed by the formula

$$FF(f, i) = |\cos(\omega(f+i))| * |\cos(\omega(f-i))|. \tag{5}$$

In contrast to equation (3), equation (5) uses the absolute value of the trigonometric function *cos* since the range of the angles $\omega(f+i)$ and/or $\omega(f-i)$ features now $\pi/2$ as a lower and not as an upper limit. The total contribution of the first $M(f)$ points belonging to the low curvature domain of f (the ones that satisfy the inequalities (4)) is computed by

$$TFF(f) = \sum_{i=1}^{M(f)} FF(f, i). \tag{6}$$

The identification of the key low curvature segmentation points from the function $TFF(f)$ is done in a way analogous to the determination of corner points from the function $TCF(c)$.

Figure 4 gives some examples of the algorithm's performance on hand-drawn shapes. Figure 5 shows the segmentation of some cursive words and their B-spline reconstruction using the segmentation points provided by the proposed algorithm. In both figures, points of curvature maxima are indicated by small squares and points of curvature minima are shown as small discs. Small discs are also used to indicated the beginning and ending points of strokes.

Modeling of pen-tip movement has made a lot of progress in recent years. In the handwriting generation model of Plamondon,[25] the velocity of pen-tip movement is considered as a resultant of curvilinear and angular velocity components. Segmentation results, similar to those obtained by us, have been reported by researchers by obtaining the maximum of angular velocity or minimum of curvilinear velocity of pen-tip movement.[26]

5. SHAPE METAMORPHOSIS

Shape metamorphosis is defined as the transformation of one shape (initial) to another (target). In our approach, a shape is represented by the segmentation points of its contour. If S_i $(i = 0, 1, \ldots, n)$ denote the segmentation points of the shape $\mathbf{S}$, then the shape $\mathbf{S}$ is represented in vector form as

$$\mathbf{S} = [S_0, S_1, \ldots, S_n]. \tag{7}$$

Metamorphosis from an initial shape $\mathbf{S}^I$ to a target shape $\mathbf{S}^T$ is accomplished by performing interpolation between the corresponding segmentation points of the two shapes. We employ linear interpolation for reasons that will be explained shortly. At any time t the shape transformation process is described by the following equation:

$$\mathbf{S}(t) = u\mathbf{S}^I + t\mathbf{S}^T$$
$$= [uS_0^I + tS_0^T, uS_1^I + tS_1^T, \ldots, uS_n^I + tS_n^T]$$
$$= [S_0(t), S_1(t), \ldots, S_n(t)], \quad (8)$$

where t is the time variable normalized to the interval $[0, 1]$ and $u = 1 - t$.

Typically, the initial and the target shapes do not have the same number of segmentation points and even if they do, a point correspondence will produce, in general, intermediate *physically invalid shapes*. By *physically invalid shapes*, we mean shapes that do not maintain the similar parts between the initial and the target shape. Since we are trying to relate the *degree of morphing* with the dissimilarity between the initial shape and the target shape, intermediate physically invalid shapes are unacceptable. In fact, a method should be found so that similar parts of the shapes are maintained throughout the metamorphosis. This way, an initial shape which is identical with a target shape will remain completely unchanged throughout the metamorphosis process, signifying a perfect match. In other words, zero *degree of morphing* will correspond to zero dissimilarity. If the initial shape is similar but not identical to the target shape, then intermediate shapes should be almost imperceptibly different, thus signifying a highly likely match. In other words, small *degree of morphing* will correspond to small dissimilarity. Figure 6 shows the proposed system exhibiting the desired metamorphosis performance in the case of two similar hand-drawn triangles. Figure 7 shows the same hand-drawn triangles as the two ends of a physically invalid metamorphosis. This latter metamorphosis behavior has been produced because a random correspondence of the triangles' segmentation points has been utilized instead of the correspondence suggested by the proposed method. It is evident from Fig. 7 that the production of significantly deformed intermediate shapes in a case like this, leaves no room for intuitive connection between *degree of morphing* and shape dissimilarity.

Thus, point correspondence is the central issue in metamorphosis-based shape recognition. All the possible point correspondences between an initial shape of n points and a target shape of m points can be represented by a graph in the form of an $m \times n$ matrix. The determination of the preferred point correspondence set is considered as a graph optimization problem and is solved by employing a dynamic programming technique. For this to happen, each candidate point correspondence set is associated with a value (cost of the point correspondence set). The computation of the cost of the point correspondence set is based on the following physics paradigm. The shapes are considered to be made of virtual wires. In this context, metamorphosis takes place through stretching and bending of the initial wire (shape) to the target wire (shape). Such a formulation allows each point correspondence set to be associated with a deformation measure that represents the stretching and bending energy that is consumed during the metamorphosis. The optimal point correspondence set is the point correspondence set that consumes the least metamorphosis energy.

5.1. Design choices for metamorphosis-based shape matching

To apply shape metamorphosis for shape matching purposes we faced certain practical problems and made specific design choices. In particular:

1. Animators in the computer graphics world manually select the features (segmentation points) in the initial and target shapes. Experience is the primary factor that guides the selection process. This is not a viable practice in the pattern recognition domain where systems are expected to perform segmentation automatically. We solved this problem by introducing the segmentation algorithm[27] described in Section 4.

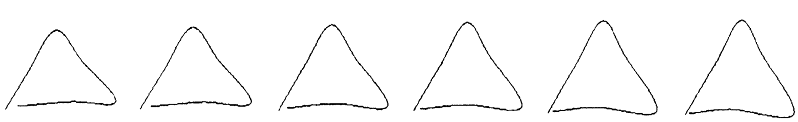

Fig. 6. Physically valid metamorphosis of a triangle to a similar triangle.

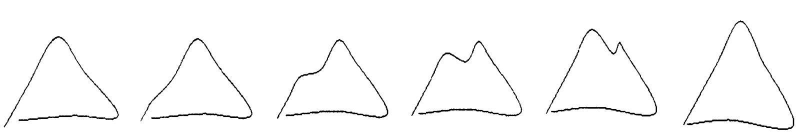

Fig. 7. Physically invalid metamorphosis of a triangle to a similar triangle.

2. The initial and target shapes will not usually have the same number of segmentation points. Therefore, steps should be taken to balance out the segmentation points in order to achieve correspondence and perform the shape metamorphosis. As a preprocessing step, additional segmentation points can be added to the shape with the smaller segmentation point count. This typically involves further segmentation of the relatively longer curve segments of the shape. Alternatively, excess segmentation points can be removed from the shape with the bigger segmentation point count. The discrete solution we adopted here can be solved in $O(mn)$ time (n: initial shape segmentation point count, m: target shape segmentation point count).
3. The choice of the interpolation technique presented an additional design dilemma. We chose linear interpolation not only for efficiency reasons but also for reasons that involve the computability of the cost function.
4. The choice of the cost function that would be used in the dynamic programming paradigm presented the most challenging problem. We chose the physics-based objective function proposed by Sederberg *et al.* in their seminal computer animation paper.[6] This functional measures the stretching and bending energy of the curve that undergoes metamorphosis and has a direct intuitive appeal. We have expanded the original formula with an additional factor that brings to bear knowledge specific to the handwriting domain.

The last two points deserve further analysis and will be elaborated in the following subsection.

5.2. Linear interpolation and cost function

We define a *point correspondence set* to be a specific correspondence scheme between the points of the initial shape and the points of the target shape. The total cost (deformation energy) E_{total} of a point correspondence set is the sum of the stretching and bending energy that a metamorphosis driven by this set will incur on the curve. The minimum total cost (optimal) of all possible point correspondences serves as the matching criterion. The physics-based metamorphosis method responds predictably as the shape dissimilarity increases by producing proportionally greater energy deformation values. The technique works best when the number of segmentation points for the initial and the target shapes are the same. In this respect, we have found, that the response of the system becomes more robust by augmenting the cost (deformation energy E^{opt}_{total}) of the optimal correspondence set in the following way:

$$E_a = E^{opt}_{total} + 5 * |n - m|, \tag{9}$$

where n is the number of segmentation points of the test (initial) shape, m is the number of segmentation points of the reference (target) shape, and E_a is the augmented energy requirement. The weight factor 5 in equation (9) has been chosen for its adequate experimental behavior. The additional term added to the original energy requirement emphasizes the fact that different versions of the same shape (either cursive word or line figure) do not have radically different numbers of segmentation points when they are written by the same person. The energy augmentation factor in equation (9) is domain-specific knowledge that strengthens the desired system behavior.

The point correspondence cost is materialized in terms of stretching and bending energy ($E_{total} = E_s + E_b$). Stretching (and complementary compressing) energy, is computed for each adjacent pair of segmentation points. Such a pair is considered as the ends of a virtual segment of wire that is stretched to another shape. The sum of the stretching energies expended for the deformation of every segment of the initial shape yields the total stretching energy requirement of the metamorphosis process. The stretching energy for a piece of virtual wire is computed from the equation:

$$E_s = f_s \frac{|L_T - L_I|^2}{(1 - c_s)\min(L_I, L_T) + c_s \max(L_I, L_T)}, \tag{10}$$

where L_I is the initial length of the wire and L_T is its length after the deformation. The term c_s (set to 0.5) is a user-definable parameter which controls the penalty for segments of the wire that collapse to points during the metamorphosis. Finally, f_s (set to 100.0) is the user-definable stretching stiffness parameter. The exponent of 2 in the numerator of equation (10) signifies that we are considering only elastic deformations. Of course, this equation is physically accurate only for small displacements. Beyond that, the stress–strain relationship becomes non-linear. Furthermore, we use these equations for wires under compression as well as under tension, which clearly is invalid since a wire will buckle under very little force. Our goal, however, is not to accurately model wire shapes, but to look for clues on how to measure shape metamorphosis. On that basis, experience justifies equation (10). Experiments with exponents higher than 2 in the stretching formula gave increased computational complexity and diminishing returns.

As it can be inferred from equation (10) energy expended for stretching varies with the scale of the shapes (see the variables L_I and L_T in equation (10)). This fact necessitated the inclusion of a size normalization step during preprocessing.

While stretching changes the local size of the curve, bending changes the local shape of the curve. The bending energy equation is formulated analogously to the stretching energy equation with two notable differences. Firstly, bending deforms angles and not wire segments and hence applies to consecutive triples of segmentation points instead of pairs of segmentation points. Secondly, the bending energy formula is affected by the consideration to ensure robustness to local variations and accommodate a physically valid metamorphosis. To this end, the bending angle ϕ of a segment should subscribe to the following two conditions:

1. $\phi(t) \neq 0,\ 0 \leq t \leq 1$, that is the shape should not self-intersect during metamorphosis;
2. $\phi(t)$ should change monotonically from $\phi(0)$ to $\phi(1)$.

The above conditions can be checked in a computationally tractable way if we interpret the itinerary of the bending angle $\phi(t)$ during metamorphosis as a degree two Bézier curve:[6]

$$Q(t) = Q_0(1-t)^2 + Q_1 2t(1-t) + Q_2 t^2, \qquad (11)$$

where $\phi(t) = \angle((1,0),(0,0),Q(t))$. $Q(t)$ is the trace of the bending angle as it deforms over time during the metamorphosis. It is trivial to check that self-intersection occurs if $Q(t)$ intersects the positive x axis and non-monotonicity occurs if a line through the origin intersects $Q(t)$ more than once. Equation (11) is based on the assumption of linear interpolation.[6] Figures 8–10 show graphically the relationship between the bending angle $\phi(t)$ and the Bézier curve traced by its side during three different metamorphosis scenarios.

Based on the above two conditions and the physics bending energy formula for an elastic wire, the bending energy for a physically valid shape metamorphosis is given by the equation

$$E_b = \begin{cases} f_b(\Delta\phi + m_b\Delta\phi^*)^2 + p_b & \text{if } \phi(t) \text{ does go to zero,} \\ f_b(\Delta\phi + m_b\Delta\phi^*)^2 & \text{if } \phi(t) \text{ never goes to zero} \end{cases} \qquad (12)$$

where f_b indicates bending stiffness (set to 0.02), m_b penalizes bending angles which deform non-monotonically (set to 100.0), and p_b penalizes bending angles from going to zero (set to 1000.0). The term $\Delta\phi$ represents the bending angle change in radians due to the metamorphosis of the particular triplet of segmentation points. $\Delta\phi^*$ is a non-negative quantity that expresses in radians how much the bending angle change deviates from monotonicity and it is computed from equation (11).

5.3. Practical Example

The proposed metamorphosis-based matching works as follows: It employs dynamic programming to find the optimal point correspondence set. It is the by-product of this process (i.e., the cost of the optimal correspondence) which is really used for the shape matching task. Figure 11 shows partially the optimal point correspondence set between a test sample and the prototype of the handwritten word *hi*. The beginning of the reference word differs from the beginning of the test word in that it features an extra stroke. Figure 11 shows how the dynamic programming algorithm optimally solves the local imbalance

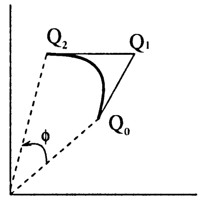

Fig. 8. The bending angle changes monotonically and the shape does not self-intersect.[6]

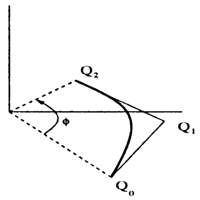

Fig. 9. The shape self-intersects—the bending angle becomes 0 at a specific instance.[6]

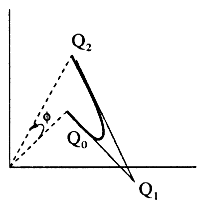

Fig. 10. The bending angle changes non-monotonically.[6]

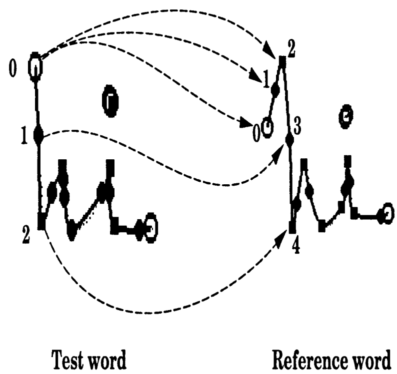

Fig. 11. Optimal point correspondence for the handwritten word *hi*.

in points by corresponding point 0 from the test word to points 0, 1, and 2 in the reference word. Figure 12 shows partly the optimal point correspondence set in matrix representation for the words in Figure 11.

For cursive words, the beginning and the end of the test and reference words are known (since we are dealing with on-line handwriting) and the cost of the dynamic programming computation is $O(mn)$ (m reference shape points, n test shape points). For line figures, however, the beginning and the end of the trace

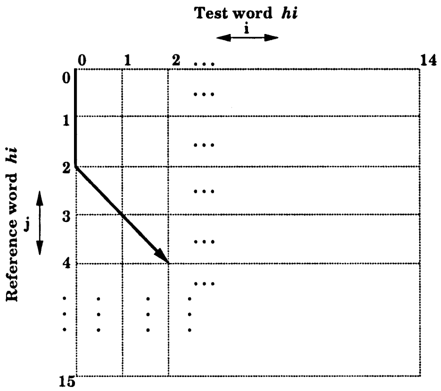

Fig. 12. Matrix representation of all possible point correspondences. The bold arrow path marks the optimal point correspondence set.

are not standard (e.g., you do not start drawing a triangle always from the same vertex). The problem in these cases is solved by considering all possible starting correspondences. This can be done in $O(mn \ln n)$.[28]

Figure 11 shows how the segmentation algorithm supports the metamorphosis-based matching by providing consistent segmentations for similar shapes. That leaves a small discrepancy in terms of the number of segmentation points and their arrangement that reflects the inherent variation of handwriting. The discrepancy is taken into account by the dynamic programming technique and the energy minimization approach that supports it.

Since we are interested in linking minimum energy metamorphosis to shape dissimilarity, a good way to test our minimum energy model is to see how it behaves in the case of two very similar shapes. The correct model should give almost imperceptibly different in-between shapes and it should yield almost zero energy expenditure. An example of the proposed method's behavior in this respect is given in Fig. 13.

5.4. *Multi-segment fusion*

The method as it is described applies to singly connected shapes only (no pen-up movements). Cursive words and simple line figures do not have many segments, but they rarely have just one. In this case, metamorphosis may take place on a segment by segment basis and the total energy (*degree of morphing*) is defined as the sum of all the segment deformation energies. Thus, for handwritten shapes we have a segment correspondence problem on top of the point correspondence problem. A writer can break a word

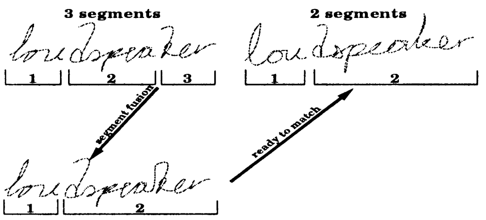

Fig. 14. Segment fusion for multi-segment shape recognition. An instance of the word *loudspeaker* with 3 segments is matched with an instance with 2 segments. The figure shows the correct segment fusion combination.

or a figure in variable points from writing to writing. The problem is, to have the test and reference shapes have the same number and arrangement of segments. Only then, similar shapes will appear similar to the metamorphosis method too.

Because for cursive words and typical line figures, the number of segments is very small (usually not more than 6–8 segments per shape), we opted to solve the problem in the following way: we compute the degree of morphing as the minimum energy value over all possible fusion combinations of consecutive segments. A characteristic example of the way the combinatorial algorithm works is shown in Fig. 14. The fusion of consecutive segments is carried out on the shape which has the larger number of segments, down to the number of segments in the other shape.

Delayed strokes are handled in a special manner. They are connected from left to right and they constitute always a single segment labeled as *delayed*. The reason for this is that many writers are not consistent in the way they write delayed strokes. For example, sometimes they put the dots on the "*i*"s before they finish the word, and sometimes after. Consequently, left-to-right direction is a far more appropriate way to describe these strokes than time sequence. If only one of the matching words has a *delayed* segment and the other has not, then the *delayed* segment is left out of consideration.

6. EXPERIMENTAL RESULTS

Twelve users have participated in the experiments. A reference database of one hundred cursive words, and seven hand-drawn line figures has been established from each user. For each shape in the reference databases, four test shapes have been collected at different days and times over a period of four months from the corresponding users. Figure 15 shows handwritten samples from a single user and Fig. 16 shows

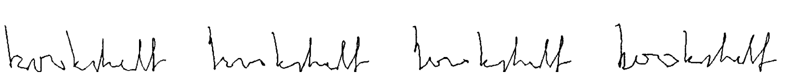

Fig. 13. Metamorphosis of a test sample of the word *bookshelf* to its prototype sample kept in the database (user 1). The metamorphosis is an example of a minimal deformation between similar handwritten words and produces the correct matching.

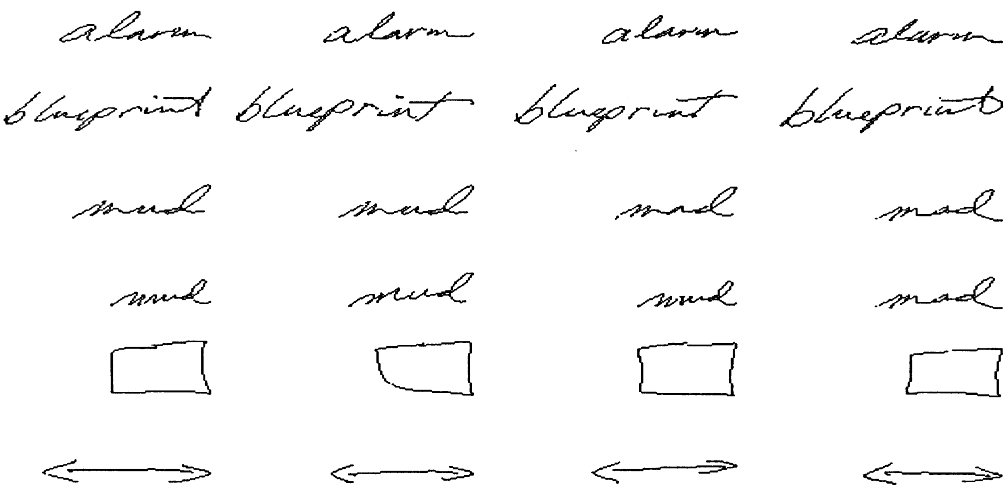

Fig. 15. Handwritten samples from one of the participants (user 9).

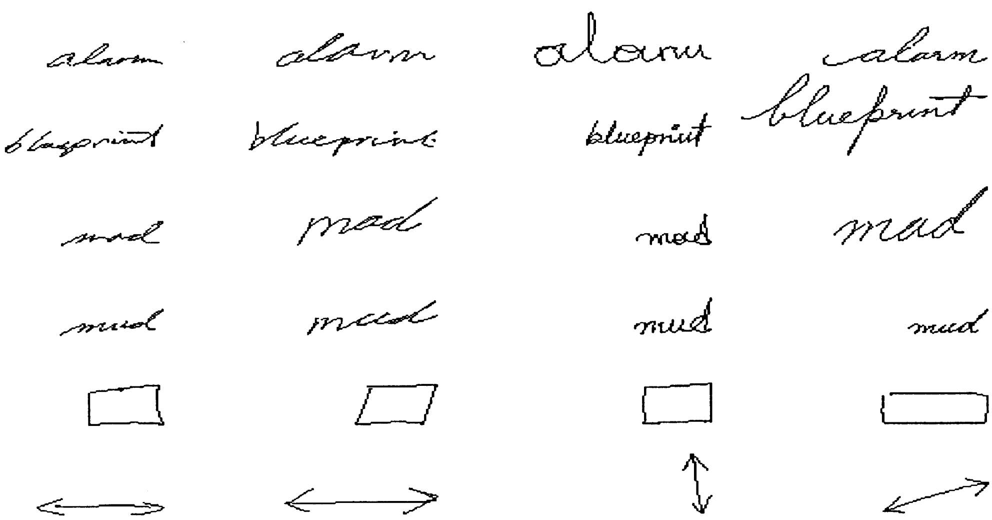

Fig. 16. Handwritten samples from users 9, 12, 1, and 5.

Table 1. Test results

Users	Reference Shapes	Test Shapes	Recognized correctly	Success rate (%)
1	107	428	424	99.0
2	107	428	386	90.1
3	107	428	393	91.8
4	107	428	391	91.3
5	107	428	400	93.4
6	107	428	386	90.2
7	107	428	385	89.9
8	107	428	394	92.0
9	107	428	369	86.2
10	107	428	392	91.6
11	107	428	384	89.7
12	107	428	399	93.2

handwritten samples from four different users. The test shapes from each user were matched against the corresponding reference database only (user-dependent system). The results of the experiment are presented in Table 1.

Most of the words have been chosen randomly from a 60,000 word lexicon. Some of the words, as well as the line figures have been chosen with the purpose to highlight the potential application of the method in electronic note pads. The method appears to be ideal for interpreting rough design sketches that include handwritten words, simple line figures, and short notes. This is the favorite mode of communication between engineers and scientists during the first stages of design. Fig. 17(a) for example, shows the handwritten block diagram of the proposed metamorphosis-based recognition system as it was produced by user 1. Fig. 17(b) shows the interpretation our system gave to Fig. 17(a) after having successfully matched each word and line figure in the sketch.

The success rates ranged all the way from 86.2 to 99.0% depending on the handwriting of the user. Two

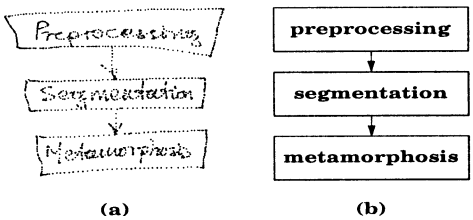

Fig. 17. (a) Handwritten block-diagram from user 1. (b) Interpretation by the metamorphosis-based system.

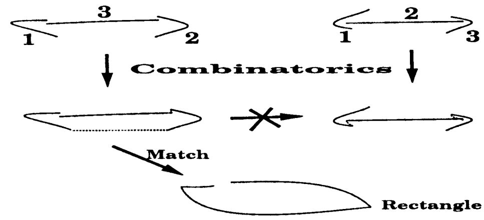

Fig. 20. Arrows drawn with different segment sequence by the same user. The numbers indicate the time sequence of drawing. The segment fusion produces for the leftmost arrow a shape instance close to a prototype handwritten rectangle. The result is an erroneous match.

major sources of failure were identified: One source of failure was poor preprocessing performance, especially in words which look alike. For example, Figs 18 and 19 shows what happened in the case of the words *mad* and *mud*, both included in the experimental sample of user 9. Sloppy handwriting, coupled with information loss during smoothing and the inherent resemblance of the involved words fooled the system.

The other source of failure was poor postprocessing performance of the segment fusion algorithm in the case of line figures. The time sequence of the various segments in certain line figures can vary from drawing to drawing. For example, in Fig. 20, the double arrow is drawn one time by the user as *arrow tip–line–arrow tip* and another time as *arrow tip–arrow tip–line*. The current segment fusion algorithm always preserves the time sequence of the segments (except of delayed strokes). Consequently, it is unable to handle the change in the time sequence observed sometimes in the case of the double arrow. If the drawing sequence were *arrow tip–arrow tip–line*, by fusing together the two arrow tips and the line we may get something close to an elongated rectangle, and as a result, an erroneous match.

The dynamic programming technique that is used to determine the *degree of morphing* is very efficient and the system performs in real-time. Nevertheless, the real-time performance of the system degrades as we move from cursive to run-on to print handwriting because the segment fusion algorithm starts incurring heavy computational expense.

7. CONCLUSIONS AND FUTURE WORK

A novel method for the recognition of difficult on-line patterns such as cursive words and hand-drawn line figures has been described. The on-line shapes are segmented into corner and key low curvature points. These points guide the metamorphosis of a test shape to each and every reference shape through a physics-based minimum energy formulation. The specific formulation links the *degree of morphing* to shape dissimilarity and allows the best match to be identified as the match that expended the least metamorphosis energy. The strong points of the method are:

1. robust behavior verified experimentally,
2. real-time performance,
3. good performance in a user-dependent setting,
4. collective handling of cursive words and figures.

The method has been designed with user-dependent systems in mind and has been tested only in such systems. In user-independent systems, its performance is expected to deteriorate since it is considered very difficult to handle the tremendous increase in variability without any statistical aid. The proposed method would ideally fit in a pen-based computing environment where the user establishes his/her personal reference database through two means: (a) Through an appropriately designed interactive tour that will instruct the user to write some sentences containing

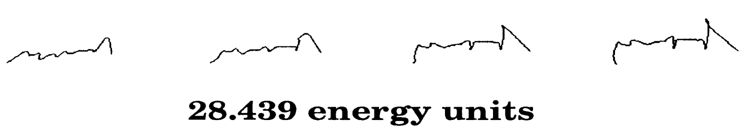

Fig. 18. Metamorphosis of a test sample of the word *mud* with the reference template of the word *mad* for user 9 produces erroneously the best match.

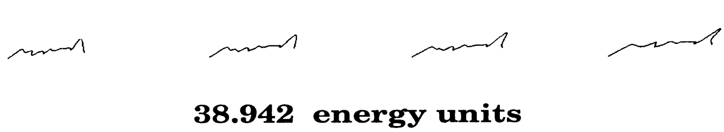

Fig. 19. Metamorphosis of the test sample of the word *mud* with its reference template for user 9 produces the second best match.

high frequency words and draw the most common line figures and (b) as a by-product of daily use (unforeseen shapes). This personal database can be stored in a disk medium and carried along by the user the same way his/her id is carried. The need by the system of a small number of reference samples per shape reduces significantly the training burden. There should be a way, however, for the system to automatically update the personal reference database when significant, although tolerable changes are detected. That will account for the long-term drift in the writing habits of the average user.

Currently, the weakest point of the method appears to be the unidirectional flow of information from the preprocessing stage towards the metamorphosis stage. For example, the presence of multiple close matching scores (Figs. 18 and 19) may indicate the need to adjust the preprocessing to capture more discriminant features. The metamorphosis-based matching proved to be as good as the quality of information that is provided. Unfortunately, there may be cases where the handwriting is sloppy enough to be impossible even for a human being to interpret. In this case, the linguistic context may be of some help. In addition, in the context of rough sketches, like that of Fig. 17, the linguistic context may be minimal or non-existent. The only solution in such cases is to enlist the help of the user through an appropriately designed interface.

The system at present cannot handle instances involving crossing out words, or rewriting part of a word to make it more visible, or better looking. Future research will also focus towards enhancing the system with the appropriate tools to deal with these situations. The segment fusion algorithm currently in use, is better suited for cursive words than 2-D on-line figures. Research also needs to be directed towards more effective ways of handling the segment correspondence problem in the case of line figures. This is part of our longer term objective to extend the applicability of the metamorphosis-based matching technique in the more general area of object recognition.

Acknowledgements—This work has been supported by the National Science Foundation through Contracts #IRI-9410003 and #IRI-9502245, the Center for Transportation Studies through Contract #USDOT/DTRS 93-G-0017-01, the Minnesota Department of Transportation through Contracts #71789-72983-169 and #71789-72447-159, the Department of Energy (Sandia National Laboratories) through Contracts #AC-3752D and #AL-3021, the McKnight Land-Grant Professorship Program, and the Department of Computer Science of the University of Minnesota. Finally, we would like to extend our sincere appreciation to the anonymous reviewer for his/her thoughtful comments and suggestions.

REFERENCES

1. S. Mori, C. Y. Suen and K. Yamamoto, Historical review of OCR research and development, *Proc. IEEE* **80**(7), 1029–1058 (1992)
2. C. Y. Suen, M. Berthod and S. Mori, Automatic recognition of handprinted characters—The state of the art, *Proc. IEEE* **68**(4), 469–487 (1980)
3. C. C. Tappert, C. Y. Suen and T. Wakahara, The state of the art in on-line handwriting recognition, *IEEE Trans. Pattern Anal. Mach. Intelligence* **12**(8), 787–808 (1990).
4. I. Pavlidis, R. Singh and N. Papanikolopoulos. Recognition of on-line hand-written patterns through shape metamorphosis. *Proc. 13th Int. Conf. on Pattern Recognition*, pp. 18–22 (1996).
5. J. R. Kent, Shape transformation for polyhedral objects, *Comput. Graphics* **26**(2), 47–54 (1992).
6. T. W. Sederberg and E. Greenwood, A physically based approach to 2-D shape blending, *Comput. Graphics* **26**(2), 25–34 (1992).
7. M. K. Brown and S. Ganapathy, Cursive script recognition, *Proc. Int. Conf on Cybernetics and Society*, pp. 47–51 (1980).
8. R. F. Farag, Word-level recognition of cursive script, *IEEE Trans. Comput.* **28**(2), 172–175 (1979).
9. L. S. Frishkopf and L. D. Harmon, Machine reading of cursive script, C. Cherry, ed. *Information Theory* (*4th London Symp.*), pp. 300–316. Butterworths, London (1961).
10. N. M. Herbst and J. H. Morissey, Segmentation mechanism for cursive script character recognition system, Patent 4 024 500 (May 1977).
11. K. S. Nathan, H. S. M. Beigi, S. Jayashree, G. J. Clary and H. Maruyama, Real-time on-line unconstrained handwriting recognition using statistical methods, *Proc. 1995 Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2619–2622 (1995).
12. L. Schomaker, Using stroke- or character-based self-organizing maps in the recognition of on-line, connected cursive script, *Pattern Recognition*, **26**(3), 443–450 (1993).
13. G. Seni, N. Nasrabadi and R. Srihari, An on-line cursive word recognition system. *Proc. 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 404–410 (1994).
14. G. Seni, R. K. Srihari and N. Nasrabadi, Large vocabulary recognition of on-line handwritten cursive words, *IEEE Trans. Pattern Anal. Mach. Intelligence* **18**(7), 757–762 (1996).
15. C. C. Tappert, Cursive script recognition by elastic matching, *IBM J. Res. Dev.* **26**(6), 765–771 (1982).
16. W. C. Lin and J. H. Pun, Machine recognition and plotting of hand-sketched line figures, *IEEE Trans. on Systems, Man Cybernet.* pp. 52–57 (1978).
17. H. Murase and T. Wakahara, On-line hand-sketched figure recognition, *Pattern Recognition* **19,** 147–160 (1986).
18. J. Hu and M. K. Brown, On-line handwriting recognition with constrained N-best decoding, *Proc. Int. Conf. on Pattern Recognition*, Vol. 3, pp. 23–27 (1996).
19. K. S. Nathan, J. Subrahmonia and M.P. Perrone, Parameter typing in writen-dependent recognition of on-line handwriting, *Proc. 13th Int. Conf. on Pattern Recognition*, Vol. 3, pp. 28–32 (1995).
20. H. Arakawa, On-line recognition of handwritten characters—Alphanumerics, Hiragana, Katakana, Kanji, *Pattern Recognition* **16**(1), 9–16 (1983).
21. W. Guerfali and R. Plamondon, Normalizing and restoring on-line handwriting, *Pattern Recognition* **26**(3), 419–431, 1993.
22. M. A. Fischler and R. C. Bolles, Perceptual organization and curve partitioning. *IEEE Trans. Pattern Anal. and Mach. Intell.* **8**(1), 100–105, 1986.
23. F. Attneave, Some information aspects of visual perception, *Psychological Rev.* **61**(3), 183–193 (1954).
24. J. J. Brault and R. Plamondon. Segmenting handwritten signatures at their perceptually important points, *IEEE Trans. Pattern Anal. Mach. Intell.* **15**(9), 953–957, (1993).

25. R. Plamondon, A model-based segmentation framework for computer processing of handwriting, *Proc. 11th Int. Conf. on Pattern Recognition* pp. 303–307 (1992).
26. X. Li and D. Yeung. On-line handwritten alphanumeric character recognition using dominant points in strokes, *Pattern Recognition* **30**(1), 31–44 (1997).
27. I. Pavlidis and N. P. Papanikolopoulos, A curve segmentation algorithm that automates deformable-model-based target tracking, Technical Report TR 96–041, Computer Science Department, University of Minnesota (1996).
28. H. Fuchs, Z. M. Kedem and S. P. Uselton, Optimal surface reconstruction from planar contours, *Commun. ACM* **20**(10), 693–702 (1977).

About the Author—IOANNIS PAVLIDIS was born in Komotini, Greece in 1963. He received the Diploma Degree in Electrical Engineering (with excellence) from the Democritus University of Thrace, Xanthi, Greece in 1987. In 1989 he was awarded the M.S. Degree in Robotics from the University of London, London, U.K. In 1995 he received the M.S. Degree and in 1996 the Ph.D. Degree in Computer Science from the University of Minnesota, Minneapolis, U.S.A. Currently, he works as a Senior Research Scientist at the Honeywell Technology Center. His main research interests are in the areas of Pattern Recognition, Computer Vision, and Computer Graphics. He is a member of the IEEE and the ACM.

About the Author—RAHUL SINGH received his Master of Science in Engineering Degree in Computer Science (with excellence) from the Moscow Power Engineering Institute in 1993. He is currently working on his doctorate in the Computer Science and engineering Department of the University of Minnesota. His main research interests are in the areas of Computer Vision and Pattern Recognition.

About the Author—NIKOLAOS P. PAPANIKOLOPOULOS (S’88-M’93) was born in Piraeus, Greece, in 1964. He received the Diploma Degree in Electrical and Computer Engineering from the National Technical University of Athens, Athens, Greece, in 1987, the M.S.E.E. in Electrical Engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, in 1988, and the Ph.D. in Electrical and Computer Engineering from Carnegie Mellon University, Pittsburgh, PA, in 1992. Currently, he is an Associate Professor in the Department of Computer Science and Engineering at the university of Minnesota. His research interests include Computer Vision, Pattern Recognition, and Robotics. He has authored or coauthored more than 70 journal and conference papers in the above areas. He was finalist for the Anton Philips Award for Best Student Paper in the 1991 IEEE Robotics and Automation Conference. Furthermore, he was recipient of the Kritski fellowship in 1986 and 1987. He is a McKnight Land-Grant Professor at the University of Minnesota for the period 1995–1997 and has received the NSF Research Initiation and Early Career Development Awards.