



Planar shape recognition by shape morphing

Rahul Singh, Nikolaos P. Papanikolopoulos*

*Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building, 200 Union Street SE,
Minneapolis, MN 55455, USA*

Received 16 December 1998; received in revised form 18 June 1999; accepted 18 June 1999

Abstract

A novel method based on shape morphing is proposed for 2D shape recognition. In this framework, the shape of objects is described by using their contour. Shape recognition involves a morph between the contours of the objects being compared. The morph is quantified by using a physics-based formulation. This quantification is used as a dissimilarity measure to find the reference shape most similar to the input. The dissimilarity measure is shown to have the properties of a metric as well as invariance to Euclidean transformations. The recognition paradigm is applicable to both convex and non-convex shapes. Moreover, the applicability of the method is not constrained to closed shapes. Based on the metric properties of the dissimilarity method, a search strategy is described that obviates an exhaustive search of the template database during recognition experiments. Experimental results on the recognition of various types of shapes are presented. © 2000 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Shape recognition; Shape morphing; Content-based retrieval; Pen-based computing

1. Introduction

Shape recognition is a fundamental problem of pattern recognition and machine vision. This problem arises in a variety of contexts, examples of which include retrieval by content from image databases, document image analysis, automated industrial inspection, analysis of medical imagery as well as vision-based robotics and visual tracking. A large number of shape recognition techniques have been proposed in the literature. Broadly speaking, these may be classified by the shape representation framework and the dissimilarity measure used. Various schemes proposed for shape representation include representation using global features like moments [1], Fourier descriptors [2], autoregressive coefficients [3], texture [4], and color [5,6]. Other representational techniques have employed local features [7], eigenmode representations [8], subspace representations [9], skeletons [10], part-

based descriptions [11], and boundary-based (contour) representations [12–14]. Some examples of various dissimilarity measures used in shape recognition include deformation-based measures like modal deformation energy [8], applications of standard metrics either directly on geometric shape descriptors [15] or on transformed shape representation [9], and measures defined on non-geometric shape attributes like color [5].

The problem of recognition when shapes are encoded by their contours is interesting for a multitude of reasons. First, contours are easy to obtain and can be used for shape representation regardless of shape convexity or the lack thereof. Second, recognition techniques based on contour analysis have broad applicability since contours are one of the most commonly used shape descriptors [16]. Finally, due to the fact that contours may be considered as high order curves, algorithms for contour-based recognition are potentially extendible to more generic shape classes including cursive words and hand-drawn sketches. In the context of contour-based shape representation, some of the dissimilarity measures that have been used include the sum of squares of the Euclidean distances from each vertex of a polygon to the convex hull of the other polygon [15], the L_2 distance between

*Corresponding author. Tel.: +001-612-625-0163; fax: +001-612-625-0572.

E-mail addresses: sing@cs.umn.edu (R. Singh), npapas@cs.umn.edu (N.P. Papanikolopoulos).

the turning functions of two polygons [17], the Hausdorff metric [18], and elastic deformation energy [19,20].

Generally speaking, in order to be effective, a recognition measure should satisfy the following properties proposed by Arkin et al. [17]:

- The measure should be a metric.
- It should be invariant under translation, rotation, and scale change.
- It should be easy to compute.
- It should match intuitive notions of shape resemblance.

From an applied perspective, certain additional properties are desirable in a generic shape recognition technique. These may include the applicability of a method regardless of shape convexity as well as its ability to deal with closed as well as open shapes. The latter requirement is of primary importance in applications like OCR and recognition problems in pen-based computing. Additionally, it contributes to the robustness of a recognition system since in real images, noise and errors in edge linking may lead to non-closure of object contours. Another important property is the ability of a recognition system to handle shape deformations. Many contemporary applications like content-based retrieval by matching image contours with hand-drawn sketches, recognition of articulate shapes, and handwriting recognition require a recognition methodology to capture the perceptual similarity of two shapes. Often this means that two shapes have to be placed in the same similarity class even if they are deformed versions of each other. In such cases, many conventional distance measures perform poorly since the mathematical description of the deformed shapes may not exactly match each other [21].

In recent literature, different solutions have been proposed to address the above issues. Bookstein [22] proposed the use of thin-plate splines to model shape deformations. Sclaroff et al. [8] describe a closed shape in terms of the eigenvectors of its stiffness matrix. Shape similarity is defined as the amount of modal deformation energy required to align two shapes. Yuille et al. [23] have used deformable templates to identify and track facial features. In the area of image retrieval, the QVE system [24] involves computing the correlation (with limited horizontal and vertical shifts) between a user sketch and an edge image in the database. Bimbo et al. [20] propose an elastic matching technique where the degree of matching along with the deformation energy spent is used to rank the similarity of hand-drawn sketches with database images. A similar idea has been used by Azencott et al. [19] for recognition of generic planar shapes.

The method described in this paper attempts to address this problem while conforming to the theoretical and applied criteria mentioned above. It differs from the

techniques mentioned in the previous paragraph in that the identity of a shape is established by morphing its contour to templates stored in a database and using a quantification of the morph as a dissimilarity measure. The quantification is formulated in terms of the stretching and bending of the contours and is invariant to similarity transformations. Unlike modal matching [8], this formulation is not restricted to closed contours. Neither does it require extensive a priori shape modeling. Furthermore, the approach does not seek to model deformations based on simple horizontal and vertical shifts during the convolution of the template with the image. While the underlying idea of the proposed method is conceptually similar to the work of Azencott et al. [19] and Bimbo et al. [20] in that it uses deformations for matching shapes, it is (unlike the aforementioned works) invariant to rotations. Additionally, the morph provides, via the synthetic images, an image plane representation of the shape and pose transformation between the input and the template. In Refs. [25,26] we have shown that these synthetic images can be interleaved with real images of an object and used as visual feedback for an eye-in-hand manipulator. Based on the apparent motion described by the virtual images, the trajectory of the manipulator is controlled to perform positioning [26] and grasping tasks [25]. Thus, the approach described can form the basis of a unified framework for addressing the problem of shape recognition as well as that of using recognition to control purposive robotic actions.

We organize this paper by looking at shape representation issues in Section 2. The recognition method is described in Section 3. In Section 4, we propose a method to reduce the number of direct shape comparisons required for recognition. The experimental results are presented in Section 5. Finally, in Section 6, the paper is summarized, conclusions are drawn, and future work is outlined.

2. Shape representation and modeling

Shapes are represented in our approach by their contours. This choice is based on the ability of shape contours to effectively capture the *visual form*, as well as the applicability of contours for the representation of different types of shapes. The contour of each shape is modeled piecewise by virtual wires. Shape morphing occurs through deformation (stretching and bending) of the artificial wires. In this formulation, the shape recognition problem can be treated as an energy minimization one, where shape similarity is quantified by the energy consumed for stretching and bending one wire-form contour model to another. Shape morphing is guided by a few key points, which are determined by segmenting the object contour.

2.1. Contour segmentation

Contour segmentation is a well-established area in computer vision and many segmentation algorithms have been proposed. A taxonomy of these techniques can be provided by broadly dividing them into two classes; methods [27–29] that place segmentation points by minimizing some error norm and methods [27,30–33] based on the identification of perceptually important points (corners). Contour segmentation can, in general, significantly reduce the number of contour points while maintaining a sufficiently accurate shape description. In the current context, the two issues that are of importance are *representational accuracy* and *representational consistency*. For rigid polygonal shapes both error-based segmentation as well as dominant point detection techniques can provide adequate representational accuracy. The problem lies in that the results of polygonal approximation may be different (in terms of the number of segmentation points and their placement), especially in the presence of noise and orientation changes. On the other hand, in deformable shapes like hand-drawn line figures, sections of the contour are often characterized by slowly varying curvature. Segmentation of such contours using dominant point detection techniques leads to poor reconstruction [34]. Moreover, small deformations can significantly alter the number of segmentation points obtained with an error-based method.

In this work, depending on the problem context, we have used two different segmentation techniques. For problems like recognition of hand-drawn figures, where the shapes from the same class may vary due to deformations, we use the segmentation algorithm proposed by Pavlidis et al. [35]. For the recognition of rigid shapes, we use an algorithm that is based on modification to the error-based segmentation technique of Ray et al. [29]. The modifications are primarily designed to obtain consistent segmentation results. In the following, we provide a brief description of the above techniques.

The basic idea of the algorithm proposed by Pavlidis et al. [35] is to represent a contour as a succession of high curvature points (corners) and relatively low curvature regions, that are represented by a single point called a *key low curvature point* (see Fig. 1). The actual algorithm consists of two parts. In the first part, corner points are detected by using the algorithm proposed by Brault and Plamondon [30]. These are then interleaved with low curvature points which are computed by using a criteria that is conjugate to the one used for computing the corners. The primary advantages of this algorithm lie in the consistent segmentation pattern ... *corner – key low curvature point – corner* ... and the automatic identification of the region of support for computing both types of curvature extrema. Fig. 1 shows the segmentation and reconstruction of some hand-drawn shapes by using this technique.

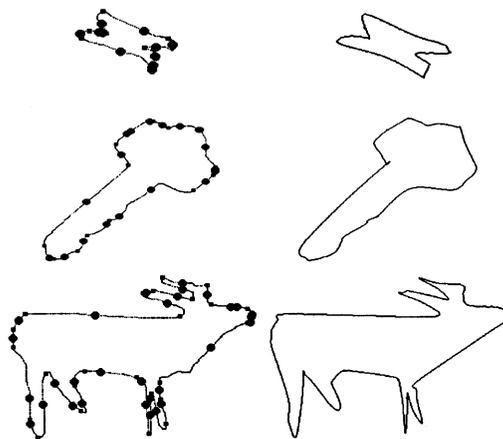


Fig. 1. Segmentation-point placement and B-spline reconstruction using the Pavlidis' algorithm [35]. Curvature maxima are indicated by small squares and curvature minima are shown as small discs.

The segmentation of rigid objects is done by using a modified version of the error-based algorithm of Ray et al. [29]. This algorithm produces a piecewise linear approximation of a contour by determining the longest possible line segment that can fit a set of contour points with the smallest possible error.

In our modifications, this algorithm is extended by computing the curvature at each segmentation point. Points having extremely low curvature are then suppressed. As a result of this operation, redundant points are removed while preserving the significant corner points. The segmentation list may however contain points, that are due to noise or quantization errors. An additional merging procedure, similar to that suggested by Huang and Wang [13], is then applied to remove any such points. The strategy is based on computing the deviation of each segmentation point from the chord joining its neighbors. If this deviation is less than a predefined threshold, the corresponding point is removed. This merging procedure is repeated till no segmentation point having a deviation less than the threshold remains. In Fig. 2 we provide some examples of the algorithm's performance on rigid shapes at different orientations and positions.

3. Shape morphing and recognition

Let S^I and S^T be the input and the target shapes, respectively. The morph of S^I to S^T is a transformation of the shape, pose and other available image attributes (like color or texture) of S^I to those of S^T . The morph is characterized by a sequence of intermediate images that depict this transformation. In the current

formulation, the shape of an object is described by its contour. The morph between two objects is therefore defined as the morph between their respective contours.

In Fig. 3, two morphs are shown. In the first case (top row), the morph occurs between two instances of the same object differing in terms of their pose with respect to the camera. The intermediate images synthesized during the morph show, predominantly, the progressive transformation of the input pose to that of the target. In the lower row, we present an example where the morph occurs between two different shapes.

Let the input and the target shapes, as represented by their segmentation points be denoted as $S^I = [S_0^I, \dots, S_n^I]$ and $S^T = [S_0^T, \dots, S_n^T]$, respectively. One possible way to morph S^I and S^T is through a cross-dissolve operation on the corresponding segmentation points of the

two contours [36]:

$$\begin{aligned}
 S(t) &= uS^I + tS^T \\
 &= [uS_0^I + tS_0^T, uS_1^I + tS_1^T, \dots, uS_n^I + tS_n^T] \\
 &= [S_0(t), S_1(t), \dots, S_n(t)],
 \end{aligned}
 \tag{1}$$

where $u = 1 - t$ and $S_i(t)$ is the i th contour point in the intermediate shape formed at time t . The time parameter t is normalized to the interval $[0, 1]$.

The contours S^I and S^T will, in general, have a different number of segmentation points. Therefore, for a morph as defined by the cross-dissolve operation of Eq. (1) to occur, a point correspondence between the segmentation points in the input and the target is needed, wherein every segmentation point on the input contour corresponds to at least one segmentation point on the target contour and vice versa.

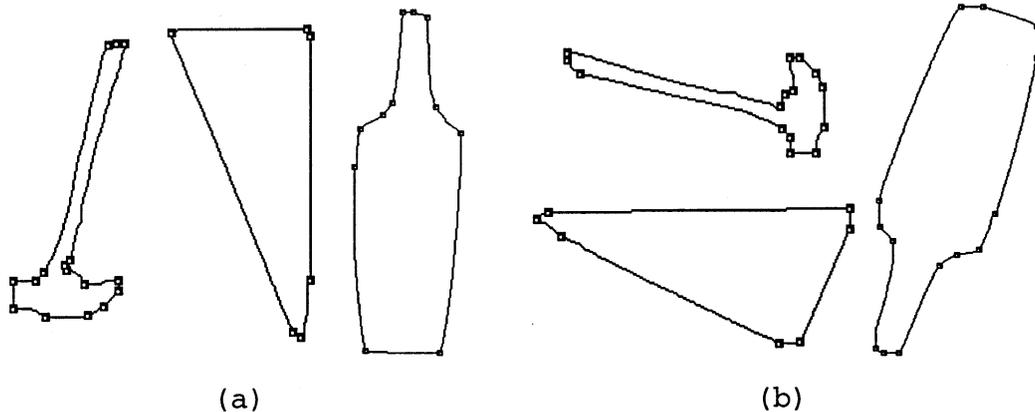


Fig. 2. Segmentation results for some shapes.

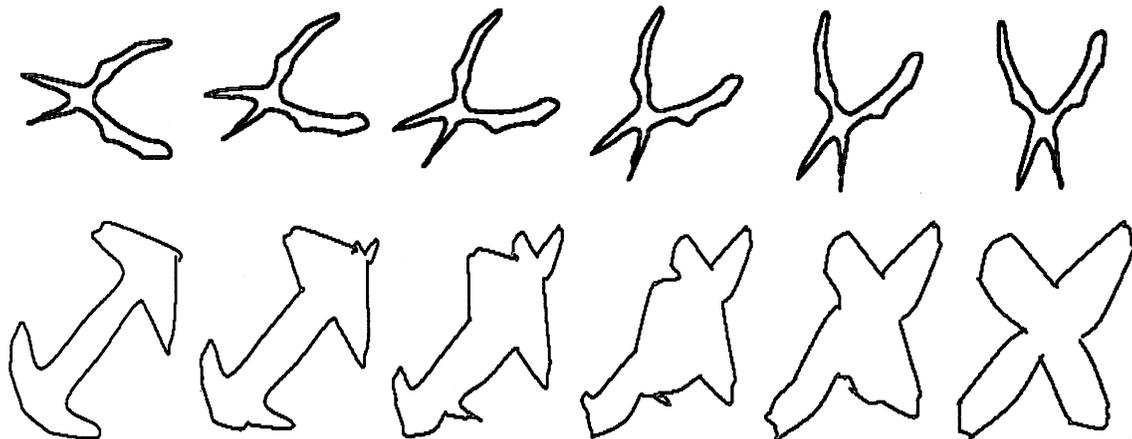


Fig. 3. Shape morphing.

The following formulation for the computation of point correspondences is motivated by the physics-based approach of Sederberg et al. [36]. The cost of a point correspondence is defined as the sum of stretching and bending required to deform the wireform contour so as to bring about the required correspondence. The stretching energy is computed for every segment (pair of points) and is defined as

$$E_s = k_s \frac{|(L_T - L_O)^2 - (L_I - L_O)^2|}{(1 - c_s)L_{\min} + c_s L_{\max}}, \quad (2)$$

where

$$L_{\min} = \min(L_O, \dots, L_I, L_T)$$

and

$$L_{\max} = \max(L_O, \dots, L_I, L_T).$$

In Eq. (2), E_s denotes the stretching energy spent in the current deformation, L_O , L_I , and L_T denote the segment lengths at the beginning, before the current deformation, and after the current deformation, respectively. The term c_s corresponds to the penalty for segments collapsing to points and k_s is the stretching stiffness parameter. The bending energy E_b is computed for point triplets and denotes the cost of angular deformation. It is defined as

$$E_b = k_b |[(\phi_T - \phi_O)^2 - (\phi_I - \phi_O)^2]|. \quad (3)$$

In the above equation, k_b indicates bending stiffness, ϕ_O represents the original angle, and ϕ_I and ϕ_T denote the angle before the current deformation and the angle after the current deformation, respectively.

The optimal morph between two contours is determined by the correspondence requiring the least stretching and bending energy. By constraining the deformations at the segmentation points, the following optimal substructure property may be observed: The optimum cost of the point correspondence (S_i^I, S_j^T) equals the optimum cost of the prior point correspondence (S_{i-1}^I, S_j^T) or (S_{i-1}^I, S_{j-1}^T) or (S_i^I, S_{j-1}^T) and the cost of establishing the correspondence (S_i^I, S_j^T). Based on the above, an efficient ($O(mn)$) dynamic programming scheme can be constructed for

morphing a contour C_A with m points to another C_B having n points. Since the energy computation described above, requires a starting point correspondence, we define the optimal morphing between two contours C_A and C_B as

$$D_{\text{morph}}(C_A, C_B) = \min_{\Omega} E(C_A, C_B). \quad (4)$$

Here, Ω denotes the set of all starting point correspondences between the contours C_A and C_B . $D_{\text{morph}}(C_A, C_B)$ (hereafter called the *degree of morphing*), denotes the cost of the optimal morph between the contours C_A and C_B .

Since the formulation of the morph is based on a linear cross-dissolve operation (see Eq. (1)), *physically valid* intermediate shapes in the morph, even between instances of the same object, are not guaranteed unless the input and the target shapes are rotationally aligned. Such lack of physical validity is expressed by a crossover of the object contours during the intermediate stages of the morph. An example of a morph with physically invalid intermediate images is presented in Fig. 4.

Physically invalid morphing is inconsistent in the sense of our formulation, because the deformations caused by the crossovers are due to alignment differences and not shape differences. A solution to this problem can be obtained by warping the contours prior to the application of the cross-dissolve operation. This warp can be defined on the basis of the observation that the point correspondence obtained during the computation of the optimal morph is invariant to translation and rotation of the objects. Based on this correspondence, elongation axes can be computed for each shape. The rotation transformation between the two shapes can be estimated by computing the angle between the elongation axes. Similarly, the translational discrepancy between the shapes may be obtained by computing the vector joining the centroids of the two shapes. The shape morphing process is thus divided into two stages. The sequence of intermediate images generated during the first stage (warping) exhibit a progressive rectification of the rigid transformations (translation and rotation) between the input and the target shapes. This rectification involves an update of the coordinates describing the input contour. The

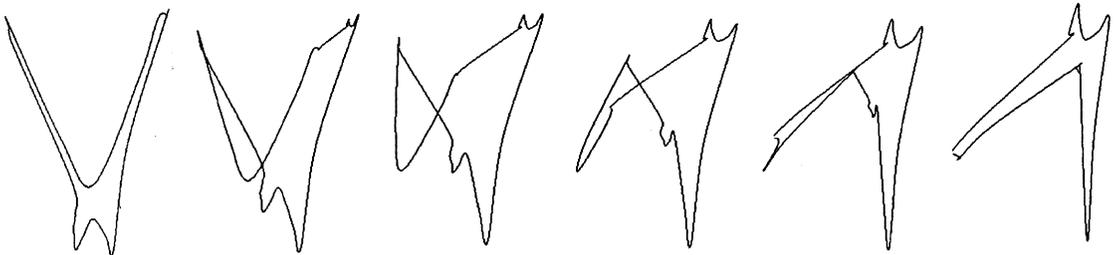


Fig. 4. A physically invalid morph.

correspondences between the input and the target contours, however, remain unaffected. These correspondences along with the updated values of the contour points are then used to rectify the shape deformations between the input and the target using the cross-dissolve operation. We would like to point out to the reader that in our formulation, the cross-dissolve operation is only defined on the geometric description of the shapes (coordinates of the segmentation points).

The recognition process thus consists of the following three steps:

1. *Shape recognition*: The template closest to the input shape, in terms of the stretching and bending energies is identified. The correspondences between the segmentation points of the input and the target are determined by using the optimal substructure property described above.
2. *Rotational and translational alignment*: The elongation axis and the centroids are computed for the input and its corresponding template. Based on the translation vector between the shape centroids and the angle between the elongation vectors, the input coordinates are updated to align the shapes.
3. *Rectification of deformations*: The updated input coordinates along with the correspondences computed during shape recognition are used to deform the input contour (by stretching and bending) till it becomes identical to the template.

3.1. Analysis of the recognition technique

Invariance of the dissimilarity measure to translation and rotation follows from the fact that the contours are described in object-centered coordinate systems and not absolute coordinate systems. Invariance to scale changes is obtained by normalizing the contour length. Furthermore, the formulation as well as the computation of the dissimilarity measure does not involve any assumptions about closure or convexity of contours. The intuition behind the proof for the metric properties of the measure comes from the fact that we model the shape changes as a conservative system. A formal proof of these properties is presented in Appendix A.

4. Reducing shape comparisons by using the triangle inequality and the primordial shape

The shape recognition framework described thus far assumes an exhaustive search of the database involving the comparison of each template to the input shape in order to find the closest template to the input. Efforts to avoid an exhaustive search of the database so as to answer a similarity query are needed because of the

following two reasons:

1. Relative high costs of computing *on-line* the dissimilarity function.
2. Potential increase in the size of the image database.

In this section, we examine the use of reference shapes in conjunction with the metric properties of the proposed dissimilarity measure to avoid an exhaustive search of the image database. The basic idea [37–39] lies in determining how shapes in the database are related to a pre-determined reference shape. In addition to this, if the similarity between the input (query) shape and this reference shape can be computed, then the templates in the database that are highly dissimilar from the query can be excluded from contention, without resorting to costly on-line comparisons.

Let S^I be the input shape (query). Further, let $D = \{S_1^T, S_2^T, \dots, S_n^T\}$ be the image database. Denote by C_A , C_B , and C_C three arbitrary shapes (contours). Let $D_{morph}(C_A, C_B)$ be the dissimilarity measure between the shapes C_A and C_B and let S_R^T be the reference shape, $S_R^T \in D$.

Since $D_{morph}(C_A, C_B)$ is a metric, it has the following properties:

$$D_{morph}(C_A, C_B) \geq 0, \quad (5)$$

$$D_{morph}(C_A, C_B) = 0 \Leftrightarrow C_A \equiv C_B, \quad (6)$$

$$D_{morph}(C_A, C_B) = D_{morph}(C_B, C_A), \quad (7)$$

$$D_{morph}(C_A, C_B) + D_{morph}(C_B, C_C) \geq D_{morph}(C_A, C_C), \quad \forall C_C. \quad (8)$$

Given an arbitrary shape C_x in the image database, it follows from the triangle inequality (8) that

$$D_{morph}(S^I, C_x) \geq D_{morph}(S^I, S_R^T) - D_{morph}(C_x, S_R^T). \quad (9)$$

Let ε be some threshold on shape similarity. For instance, ε may be selected as the distance between the input shape S^I and a template shape that is closest to the input after a *partial* search through the database. It follows from Eq. (9), that if

$$D_{morph}(S^I, S_R^T) - D_{morph}(C_x, S_R^T) \geq \varepsilon, \quad (10)$$

then the comparison between S^I and C_x does not need to be considered. This is because the definition of ε guarantees the existence of at least one shape in the database that is closer to S^I than C_x with respect to the dissimilarity measure $D_{morph}(\cdot, \cdot)$. Similarly, we also have from the triangle inequality

$$D_{morph}(C_x, S^I) + D_{morph}(S^I, S_R^T) \geq D_{morph}(C_x, S_R^T). \quad (11)$$

But from the symmetry of the dissimilarity measure, we have

$$D_{morph}(C_x, S^I) = D_{morph}(S^I, C_x). \quad (12)$$

Substituting the above in Eq. (11) and simplifying we get

$$D_{\text{morph}}(\mathbf{S}^I, \mathbf{C}_x) \geq D_{\text{morph}}(\mathbf{C}_x, \mathbf{S}_R^T) - D_{\text{morph}}(\mathbf{S}^I, \mathbf{S}_R^T). \quad (13)$$

Once again if

$$D_{\text{morph}}(\mathbf{C}_x, \mathbf{S}_R^T) - D_{\text{morph}}(\mathbf{S}^I, \mathbf{S}_R^T) \geq \varepsilon, \quad (14)$$

then the comparison between \mathbf{S}^I and \mathbf{C}_x does not need to be carried out.

The constraints in Eqs. (10) and (14) describe the criteria that can be used to exclude shapes in the database without influencing the correctness of the recognition process [37]. The comparison of the database shapes with the reference shape, $D_{\text{morph}}(\mathbf{C}_x, \mathbf{S}_R^T)$, is computed off-line. The comparison between the input shape and the reference shape, $D_{\text{morph}}(\mathbf{S}^I, \mathbf{S}_R^T)$, is done on-line for every new input. If during the comparisons, a template shape is found that is closer to the query shape than the best match obtained till that point in the search, then the identity of the best match is updated to this template.

The speedup obtained by using the above idea depends on the computation of the threshold ε . Computing a *good* value of ε can substantially ameliorate the search complexity. However, the difficulty lies in that computing a *good* value of ε itself involves searching large parts of the image database. A possible solution to these mutually conflicting goals can be obtained by using the idea of a *primordial shape*. Every shape \mathbf{C}_y in the database is morphed to a *primordial shape* \mathbf{S}_P^T . The primordial shape can be, for instance, a point or a line. The value of the morph $D_{\text{morph}}(\mathbf{C}_y, \mathbf{S}_P^T)$, then becomes an indicator of the complexity of the shape \mathbf{C}_y . An unknown input \mathbf{S}^I is first morphed to the primordial shape \mathbf{S}_P^T . The value of this morph is used to identify a subset \mathbf{D}_S of the image database \mathbf{D} . The subset \mathbf{D}_S consists of shapes having the same order of similarity as \mathbf{S}^I with respect to \mathbf{S}_P^T . Formally,

$$\mathbf{D}_S = \{ \mathbf{C}_y: \mathbf{C}_y \in \mathbf{D}, |D_{\text{morph}}(\mathbf{C}_y, \mathbf{S}_P^T) - D_{\text{morph}}(\mathbf{S}^I, \mathbf{S}_P^T)| < \delta \}. \quad (15)$$

In the above equation, δ is a parameter the value for which is provided. The reference shape can then be selected either as the primordial shape or as the shape closest to the input \mathbf{S}^I in terms of the cost of the morph from \mathbf{S}_P^T . The search space is pruned by computing the dissimilarity measure over the shapes in \mathbf{D}_S . An alternative to providing the parameter δ in Eq. (15), is to use a *K-nearest neighbor* rule to select the templates for which the shape comparison is carried out.

5. Experimental results

Three different sets of experiments were used to test the performance of the proposed system. In each of the experiments, a different data set was employed to test

the applicability of the method in different domains. In the first experiment the method was used for the recognition of rigid objects. In this case the test images differed from the templates in terms of Euclidean transformations as well as due to poor thresholding and/or partial occlusions. The second experiment was designed to test the recognition performance on shapes which differ from their corresponding templates both by Euclidean transformations and deformations. The applicability of the method for the recognition of open shapes (handwritten cursive words in an on-line setting), was studied in the final experiment.

5.1. Recognition of rigid objects

The effectiveness of the proposed method was validated in an experiment with 16 objects. An image of each object, taken at an arbitrary location and orientation with respect to the camera was stored as a template. The imaging plane was assumed to be perpendicular to the optical axis of the camera. In Fig. 5, the objects used in the experiment are shown. For each object, five test images were captured by arbitrarily varying the object pose in terms of translation, rotation and scale. The object contours were extracted after automatic moment-preserving thresholding. The recognition results are summarized in Table 1. In Fig. 6, examples of some noisy shapes from the test set are shown (bottom row), along with their corresponding templates (top row). For the cases shown in columns (a) and (b), the translation of the camera lead the object to partially move out of the imaging region. In the first instance (column (a)), the recognition was not affected. For the instance shown in column (b), a misrecognition occurred. The shape variations shown in columns (c) and (d) were due to poor thresholding. Both the instances were identified correctly.

The average values of the dissimilarity measure over the test cases for each shape are presented in Tables 2 and 3. The lowest value for the morph for each test set is underlined. In Table 4, we present the results for pruning the database search based on the use of the triangle inequality and the primordial shape. The test set used in this experiment is the same as the one for which the recognition values are presented in Tables 2 and 3. In this experiment, the value of the morph $D_{\text{morph}}(\mathbf{S}_i^T, \mathbf{S}_P^T)$, between each template shape \mathbf{S}_i^T and the primordial shape \mathbf{S}_P^T was computed offline along with the morph values between the templates. In the on-line recognition phase, each input was initially morphed to the primordial shape. The value of this morph $D_{\text{morph}}(\mathbf{S}^I, \mathbf{S}_P^T)$ was then compared with the value of the morphs $D_{\text{morph}}(\mathbf{S}_i^T, \mathbf{S}_P^T)$. A selection criteria (*K*-nearest neighbors, denoted henceforth as *k*-NN) was used to define the subset of template shapes from which the identity of the input was established using the constraints defined in Eq. (10) and Eq. (14). The initial reference shape \mathbf{S}_R^T for each subset of templates was

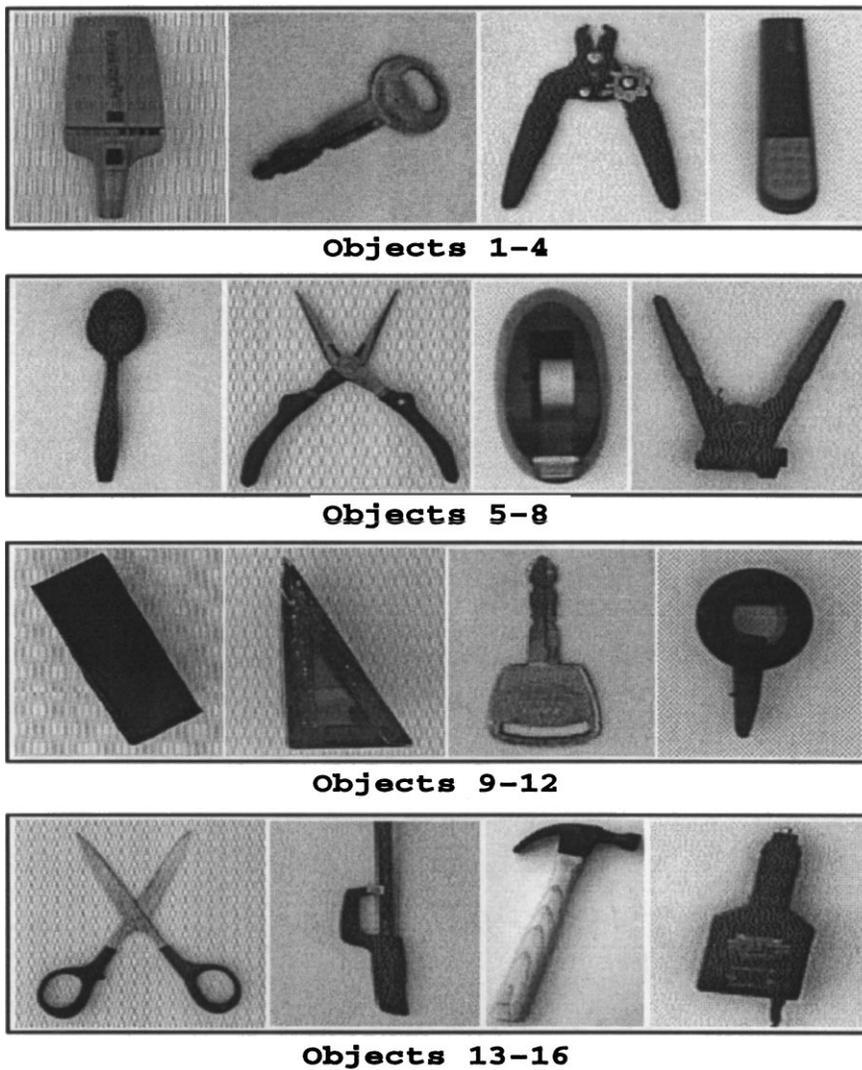


Fig. 5. Examples of rigid shapes.

Table 1
Recognition results for rigid objects

Templates	Test shapes	Correctly recognized	Misrecognized	Recognition rate
16	80	79	1	98.75%

defined to be the template which was closest to the input in terms of the distance from the primordial shape.

For the test set considered in Table 4, an average number of 2.50 shape comparison operations of complexity $O(mn)$ were required for a recognition rate of 100%. In comparison, the general case where the query is compared to each of the templates requires 15 such compari-

sons. In the given experiment, the correct template for each test set was in the list of templates obtained by using the 3-NN rule. This along with the use of Eqs. (10) and (14) lead to the reduction in the number of comparisons. With increase in the size of the image database, a rigid clustering criteria like k -NN may not guarantee the inclusion of the correct template in the subset of shapes on which the search is performed. The use of flexible clustering criteria is likely to provide more robust results than by using fixed criteria or thresholds.

5.2. Recognition of deformable shapes

A collection of 15 shapes was used in this experiment. Amongst these shapes were hand-drawn outlines of industrial tools and objects, sketches of natural objects,

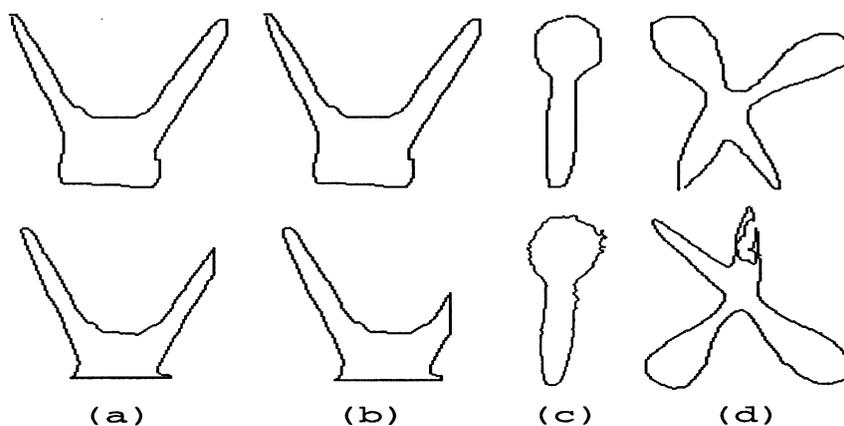


Fig. 6. Examples of noisy shapes used in the recognition experiment.

Table 2
Average *degree of morphing* values (templates 1–8)

Templates	1	2	3	4	5	6	7	8
Test shapes								
1	<u>257.87</u>	748.86	598.12	672.65	710.90	862.61	648.74	510.66
2	955.50	<u>123.77</u>	413.30	705.64	311.31	789.61	576.60	487.28
3	705.86	<u>458.35</u>	<u>218.21</u>	691.19	418.22	810.12	535.62	457.82
4	512.91	561.29	<u>656.64</u>	<u>64.45</u>	302.67	1296.70	125.87	605.22
5	612.82	272.64	414.44	<u>341.25</u>	<u>91.00</u>	1038.10	360.85	448.81
6	997.98	788.57	651.96	1269.90	<u>966.67</u>	<u>249.26</u>	1079.30	804.86
7	579.06	432.03	513.92	207.46	325.58	<u>1044.50</u>	<u>90.73</u>	536.59
8	560.71	392.93	446.17	725.53	411.54	717.71	<u>624.52</u>	<u>146.35</u>
9	454.38	517.94	556.84	119.63	353.85	1169.20	155.61	<u>523.22</u>
10	637.74	622.64	658.76	193.19	474.10	1282.70	199.74	558.12
11	705.45	233.01	325.90	639.79	210.34	767.89	478.65	472.54
12	919.27	265.60	331.16	813.49	390.15	710.22	641.96	543.37
13	761.71	473.08	439.39	1034.40	592.99	534.53	838.42	495.18
14	728.36	347.26	444.69	394.56	377.66	986.59	323.42	412.02
15	815.73	355.02	602.02	404.27	581.37	918.63	278.25	380.51
16	771.66	210.32	324.55	689.60	270.57	864.95	530.77	396.49

and contours from medical images. A database of 60 samples sketches (four per template) was collected from a group of users. The aim of the experiment was to test the ability of the system to retrieve objects having the same visual form as the sketch irrespective of the shape variations introduced during the drawing. The users were shown each template prior to the collection of the respective test samples. However, during the actual drawing phase no references were allowed to the template. This step curtailed excessive drawing variations while ensuring that the recognition problem was not trivialized. No other constraints were imposed on the drawings. Some examples of hand-drawn sketches used as tem-

plates (in black borders), and corresponding sample sketches used as test shapes are shown in Fig. 7. The recognition results for the shapes used in this experiment are reported in Table 5. For most of the test shapes, the proposed system exhibited good tolerance to shape variations. In Fig. 8, we present an example of content-based retrieval based on user sketches. The template is an industrial object and the query image is its hand-drawn sketch. In this particular example, the query image differed from the template in terms of translations, rotations and deformations. In the figure, the intermediate images of the morph show the rectification of rotations and deformations.

Table 3
Average degree of morphing values (templates 9–16)

Templates	9	10	11	12	13	14	15	16
Test shapes								
1	585.11	686.62	671.31	791.66	744.59	687.49	872.89	753.69
2	677.40	745.02	247.05	231.22	517.49	428.31	443.33	208.07
3	621.80	663.35	301.18	290.54	395.78	485.72	617.66	331.91
4	148.10	215.85	531.18	755.18	988.87	342.45	378.89	609.38
5	375.89	475.28	199.65	381.70	586.18	338.07	568.63	275.92
6	1137.70	1178.50	771.63	628.56	467.02	938.76	924.05	776.21
7	205.02	253.35	422.09	604.21	768.88	288.72	341.82	415.04
8	636.38	636.61	344.73	437.92	399.27	477.11	442.34	353.25
9	39.08	101.14	467.97	761.95	892.65	281.38	410.54	546.85
10	129.43	64.38	589.40	852.05	952.88	257.22	358.76	645.72
11	573.85	631.73	137.77	258.50	405.93	407.27	527.82	197.93
12	815.06	868.29	282.65	92.01	373.43	549.18	569.08	247.04
13	908.83	969.32	420.80	364.95	180.63	687.73	614.99	504.04
14	331.88	240.05	413.58	566.94	674.91	70.94	280.58	420.11
15	425.85	407.73	501.38	553.58	592.91	331.76	72.19	413.18
16	642.01	697.38	223.61	244.67	438.63	379.38	502.87	109.32

Table 4
Performance of the search pruning strategy

Test shape	$D_{morph}(S^t, S_p^t)$	Clustering rule	Template subset	Shape identity	Number of comparisons	Correct recognition
1	1465.28	3-NN	{1, 6, 13}	1	2	Yes
2	499.55	3-NN	{5, 2, 16}	2	3	Yes
3	927.31	3-NN	{3, 8, 10}	3	3	Yes
4	781.96	3-NN	{4, 7, 9}	4	3	Yes
5	485.80	3-NN	{5, 2, 16}	5	2	Yes
6	1608.85	3-NN	{6, 1, 13}	6	2	Yes
7	785.70	3-NN	{4, 9, 7}	7	3	Yes
8	956.54	3-NN	{3, 8, 10}	8	3	Yes
9	800.26	3-NN	{9, 4, 7}	9	3	Yes
10	850.76	3-NN	{10, 9, 4}	10	1	Yes
11	616.07	3-NN	{11, 16, 14}	11	3	Yes
12	730.78	3-NN	{15, 12, 7}	12	3	Yes
13	1106.57	3-NN	{13, 8, 3}	13	3	Yes
14	680.74	3-NN	{14, 15, 12}	14	1	Yes
15	731.17	3-NN	{15, 12, 7}	15	3	Yes
16	566.52	3-NN	{16, 11, 5}	16	2	Yes
Average number of comparisons					2.50	

5.3. Recognition of open shapes

In this experiment, the method was used for the recognition of unconstrained cursive handwriting in a user-dependent, on-line setting. Four users participated in the experiment. A vocabulary of ten randomly selected words was used (see Table 6). Each user provided one template sample and four test samples for each word. The data was collected using a WACOM UD-0608R tablet

with an inking stylus. Table 7 contains the recognition results for each of the users.

Examples of correct morphing for the words *adventure* and *guard* both taken from user 4 are shown in Fig. 9. In case of the word *guard*, the reader may note the poor segmentation of the descender stroke of the letter *g*. The resultant loss of information in this case is similar to instances where parts of a shape get occluded. Correct recognition in the given example was facilitated by the

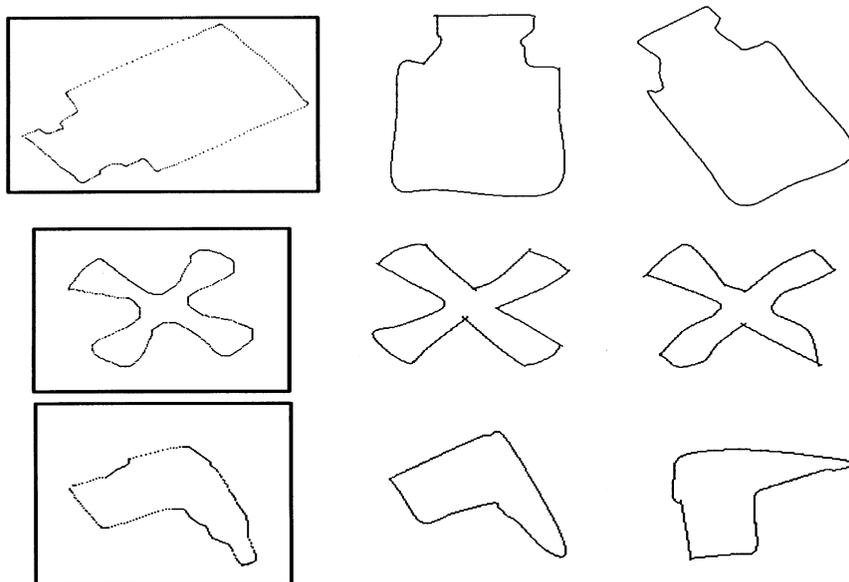


Fig. 7. Examples of deformable shapes.

Table 5
Experimental results

Templates	Test shapes	Correct classifications	Misclassifications	Success rate
15	60	56	4	93.33%

Table 6
List of words used in the handwriting recognition experiment

adventure	banana	bookshelf	cannon	flywheel
guard	landmark	mad	peach	tooth

features detected in the rest of the word as well as the ability of the recognition method to handle deformations. It is worthwhile to contrast this case with the one in Fig. 10, where a misrecognition occurred. The test sample in this case was written at a high speed and with insufficient pressure on the tablet. The curve describing this word, therefore, consisted of a small number of unevenly distributed points. This caused the segmentation algorithm to perform poorly in terms of capturing important feature points. Consequently, the cost of the morph to the template *cannon* was lower than that to the correct template (*banana*), which was the second lowest.

The results presented in this section are conceptually similar to the more detailed experiments described in a previous paper [34]. The basic differences lie in the use of a different dissimilarity measure, a different formulation

Table 7
Test results for cursive words

Users	Reference words	Test words	Correctly recognized	Recognition rate
1	10	40	40	100.0%
2	10	40	36	90.0%
3	10	40	37	92.5%
4	10	40	39	97.5%

of the shape-morphing process, and in that the preprocessing steps of rotation and slant normalization were omitted in the present case. While the proposed method is invariant to rotations, changes in the slant of letters within a word were treated as deformations during the recognition process.

6. Conclusions and future work

In this paper, we have described a technique for comparing shape similarity based on quantifying the morph of one shape to another. Each shape is represented by a polygonal approximation of its contour. Shape morphing occurs by the stretching and bending of the contours. Quantification of the morph is obtained by computing the incremental energy spent in deforming one shape to another as given by a physics-based model. The recognition methodology uses a pruning scheme, based on the mathematical properties of the morph, to significantly

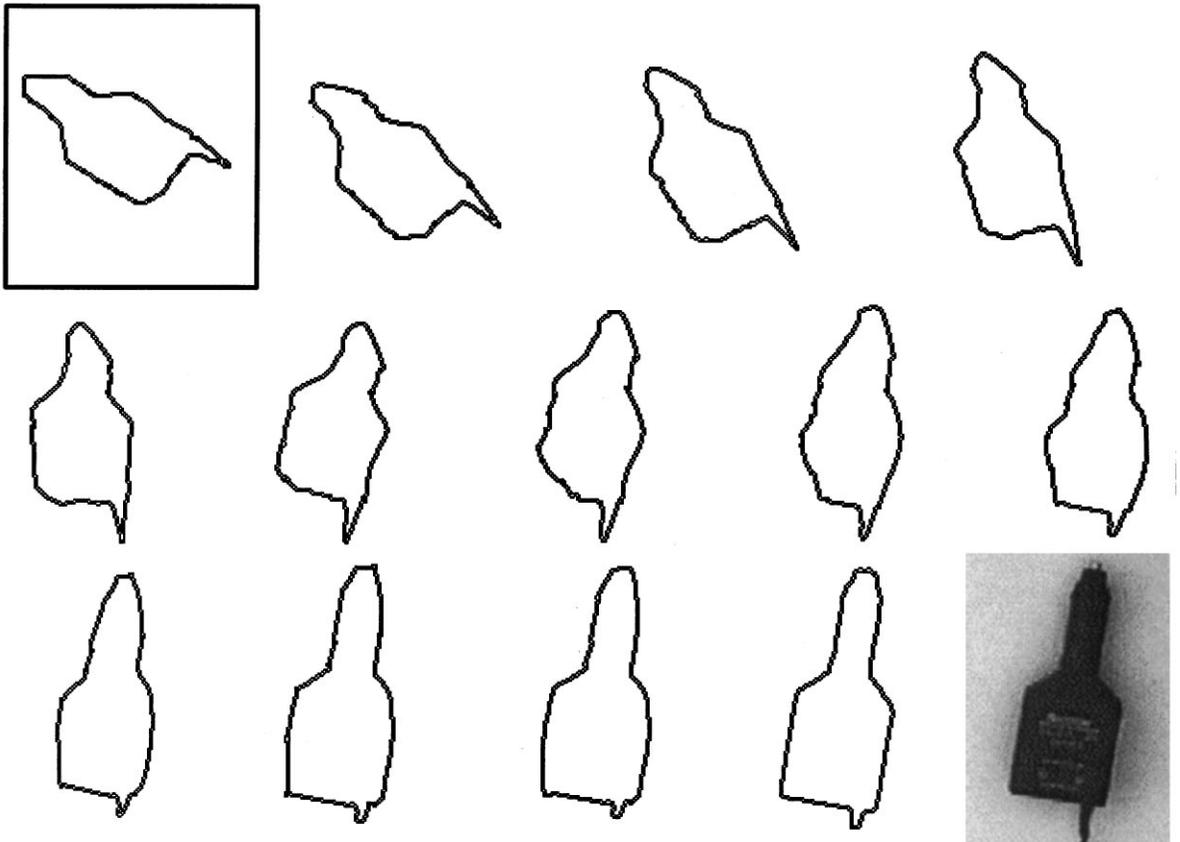


Fig. 8. Content-based retrieval by morphing a user sketch to the correct database shape.

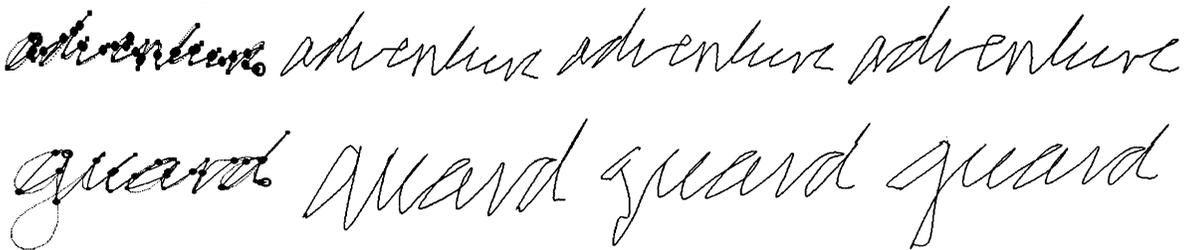


Fig. 9. Morphs of the words *adventure* and *guard* to their respective templates kept in the database (user 4). The input words are shown with the segmentation points superimposed.

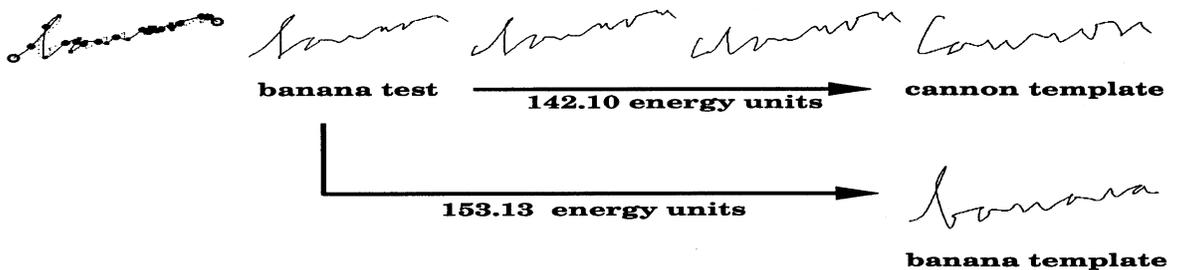


Fig. 10. Morph of the word *banana* to the template for the word *cannon* (user 3) produces erroneously the best match. The input word is shown with the segmentation points superimposed.

reduce the number of on-line shape comparisons during the recognition phase.

The proposed approach has been applied for the recognition of both rigid and deformable shapes in different application domains. The salient properties and advantages of the proposed method include:

1. Invariance of the method to translations, rotations and scale changes.
2. The method has the properties of a metric.
3. Its applicability to convex as well as non-convex shapes.
4. It can be used for the recognition of open shapes like letters and cursive words.
5. The method has low computational complexity and it is intuitive.

Due to its ability to handle deformations, the proposed recognition paradigm can be used in retrieval-by-content systems or in applications like pen-based computing where on-line recognition of deformable shapes is required. Another significant attribute of the proposed method is that the intermediate images in the morph describe the shape and pose transformation needed to align the input and the target images. The transformation in the morph plane is relative to a static virtual camera. They can however be interpreted in terms of a mobile real camera. In such a case the pose transformation described by the virtual images can be used to control the camera motion so that the real views of the object being served correspond to the virtual views generated by the morph. Based on the above idea, we have obtained promising results in controlling robotic interactions like positioning and grasping by using image morphing [25,26].

An important constraint on the performance of the system is its dependence on the segmentation algorithm. In particular, we have observed that segmentation algorithms that lead to inconsistent point placement may lead to degradation of the recognition performance even though the error during shape segmentation is small. We have looked at two segmentation strategies that have reasonable performance in terms of consistent point placement. Better segmentation strategies, in the above sense, will improve the recognition performance. Since the proposed approach is based on matching contour descriptions, it performs poorly for shapes where the primary difference in appearance is due to internal topological features or other attributes like texture or color. Furthermore, since contour description is inherently sensitive to illumination and/or shadowing, the performance of the proposed technique will degrade in the presence of shadows and poor control of illumination. In the current experimentations, a single arbitrarily selected image of each object was used as a template. Improvements in the recognition performance can be expected either by optimizing the choice of the template

or by using multiple templates for each shape class. Extending the present framework to incorporate image attributes like color and texture as well as recognition of 3D objects by morphing 3D shapes are other possible directions of future work.

7. Summary

A novel method based on shape morphing is proposed for 2D shape recognition. In this framework, the shape of objects is described by using their contours. Shape recognition involves a morph between the contours of the objects being compared. The morph is quantified by using a physics-based formulation. This quantification serves as a dissimilarity measure to find the reference shape most similar to the input. The proposed dissimilarity measure is shown to have the properties of a metric as well as invariance to Euclidean transformations. The recognition paradigm is applicable to both convex and non-convex shapes. Moreover, the applicability of the method is not constrained to closed shapes. Based on the metric properties of the dissimilarity method, a search strategy is described that obviates an exhaustive search of the template database during recognition experiments. Experimental results on the recognition of various types of shapes are presented.

Acknowledgements

Ioannis Pavlidis had participated in the initial phase of this research. The proof of the metric properties has benefited from numerous discussions with Soumyendu Raha. The critique of Richard Voyles was instrumental in developing the segmentation algorithm used for rigid shapes. The presentation of this paper has also improved due to the comments provided by the anonymous reviewer. The authors wish to express their gratitude to each of the aforementioned. The research of Rahul Singh on this project was supported by the National Science Foundation through grants #IRI-9410003 and #IRI-9502245.

Appendix A. Proof of the metric property

To prove the metric properties of the dissimilarity measure we consider the stretching energy and the bending energy separately. We have from the definition of the stretching energy (see Eq. (2)),

$$E_s = |W_s| = k_s \times \frac{|(L_T - L_O)^2 - (L_I - L_O)^2|}{(1 - c_s) \min(L_O, \dots, L_I, L_T) + c_s \max(L_O, \dots, L_I, L_T)} \quad (\text{A.1})$$

By assuming a hierarchy of shape complexity, as discussed in this paper and the related references, the shape recognition problem can be solved without an exhaustive search of the shape database. The basic idea lies in considering shapes in the database to be derived, by stretching and bending, from a *primordial shape*, like a point or a line. Given a query shape, the measure of its dissimilarity from the primordial shape can be used to identify a subset of templates similar to the input. Since the original length L_0 and the original angle ϕ_0 are defined to be the length and angle prior to any deformations they essentially are parameters of the primordial shape. Conceptually, the primordial shape should lie at the origin of the shape hierarchy. Its parameters L_0 and ϕ_0 are therefore defined as

$$L_0 \equiv \min(L_0, L_1, \dots, L_k) \quad (\text{A.2})$$

$$\phi_0 \equiv \min(\phi_0, \phi_1, \dots, \phi_k) \quad (\text{A.3})$$

where L_i and ϕ_i refer to the length and the angle after the i -th deformation respectively.

We base the proof of the metric property on the above construction. Consider the case of the stretching energy; Let the original length be denoted by L_0 . The lengths after deformation (stretching) at the stages i, j , and k are correspondingly denoted by L_i, L_j , and L_k respectively. The stretching energy spent in compressing or expanding a wire of length L_i to the length L_j is represented as $E_s(i, j)$.

From the formula of the stretching energy (see Eq. (2)), it may be verified trivially that

$$E_s(i, j) = |W_s(i, j)| \geq 0 \quad (\text{A.4})$$

and

$$E_s(i, j) = E_s(j, i). \quad (\text{A.5})$$

To prove the triangle inequality, we consider the following three types of length changes: *monotonic decrease*, *monotonic increase*, and *non-monotonic length change*.

1. *Monotonic decrease*: Let the lengths of the virtual wire at stages i, j , and k be L_i, L_j , and L_k respectively. Furthermore, $L_i \geq L_j \geq L_k$ (monotone decrease in length). To prove the triangle inequality we need to show that

$$E_s(i, j) + E_s(j, k) \geq E_s(i, k) \quad (\text{A.6})$$

Expanding each of the above terms by using the formula for stretching energy, the inequality to be proved becomes

$$\begin{aligned} & k_s \frac{|(L_j - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_s L_i} \\ & + k_s \frac{|(L_k - L_0)^2 - (L_j - L_0)^2|}{(1 - c_s)L_0 + c_s L_j} \\ & \geq k_s \frac{|(L_k - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_s L_i}. \end{aligned}$$

The above inequality can be simplified to the following form:

$$\begin{aligned} & \frac{(L_i - L_0)^2 - (L_j - L_0)^2}{(1 - c_s)L_0 + c_s L_i} + \frac{(L_j - L_0)^2 - (L_k - L_0)^2}{(1 - c_s)L_0 + c_s L_j} \\ & \geq \frac{(L_i - L_0)^2 - (L_k - L_0)^2}{(1 - c_s)L_0 + c_s L_i}. \end{aligned} \quad (\text{A.7})$$

The right-hand side of the above inequality may be rewritten as:

$$\begin{aligned} & \frac{(L_i - L_0)^2 - (L_k - L_0)^2}{(1 - c_s)L_0 + c_s L_i} \equiv \frac{(L_i - L_0)^2 - (L_j - L_0)^2}{(1 - c_s)L_0 + c_s L_i} \\ & + \frac{(L_j - L_0)^2 - (L_k - L_0)^2}{(1 - c_s)L_0 + c_s L_j}. \end{aligned} \quad (\text{A.8})$$

By substituting the above in the RHS of (A.7), we obtain an equality and thus the proof.

2. *Monotonic increase*: If the lengths of the virtual wire at stages i, j , and k are L_i, L_j , and L_k respectively. Then in this case, $L_i \leq L_j \leq L_k$. To prove the triangle inequality we need to show that

$$E_s(i, j) + E_s(j, k) \leq E_s(i, k). \quad (\text{A.9})$$

Expanding the above by using the formula for stretching energy and simplifying for the absolute values, we get

$$\begin{aligned} & k_s \frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_s L_j} + k_s \frac{(L_k - L_0)^2 - (L_j - L_0)^2}{(1 - c_s)L_0 + c_s L_k} \\ & \geq k_s \frac{(L_k - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_s L_k}. \end{aligned} \quad (\text{A.10})$$

Rewriting the right-hand side of the above inequality, we get

$$\begin{aligned} & k_s \frac{(L_k - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_s L_k} \equiv k_s \frac{(L_k - L_0)^2 - (L_j - L_0)^2}{(1 - c_s)L_0 + c_s L_k} \\ & + k_s \frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_s L_j}. \end{aligned} \quad (\text{A.11})$$

Substituting the above on the right-hand side of inequality (A.10), we have

$$\begin{aligned} & k_s \frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_s L_j} + k_s \frac{(L_k - L_0)^2 - (L_j - L_0)^2}{(1 - c_s)L_0 + c_s L_k} \\ & \geq k_s \frac{(L_k - L_0)^2 - (L_j - L_0)^2}{(1 - c_s)L_0 + c_s L_k} \\ & + k_s \frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_s L_j}. \end{aligned} \quad (\text{A.12})$$

The above inequality is valid iff

$$\frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_sL_j} \geq \frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_sL_k} \tag{A.13}$$

Clearly the above is true because $L_j < L_k$. \square

3. *Non-monotonic length changes*: Consider the case where the lengths of the virtual wire at stages i, j , and k are L_i, L_j , and L_k respectively. Furthermore, let $L_k \leq L_i \leq L_j$. Rewriting the triangle inequality in terms of the stretching energy, we obtain the following inequality which needs to be proved:

$$k_s \frac{|(L_j - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_sL_j} + k_s \frac{|(L_k - L_0)^2 - (L_j - L_0)^2|}{(1 - c_s)L_0 + c_sL_j} \geq k_s \frac{|(L_k - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_sL_i} \tag{A.14}$$

For the term on the RHS of the above inequality, we have

$$k_s \frac{|(L_k - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_sL_i} \leq k_s \frac{|(L_k - L_0)^2 - (L_j - L_0)^2|}{(1 - c_s)L_0 + c_sL_i} + k_s \frac{|(L_j - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_sL_i} \tag{A.15}$$

Substituting the above in the RHS of inequality (A.14), we obtain the following inequality, the validity of which we need to prove:

$$k_s \frac{|(L_j - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_sL_j} + k_s \frac{|(L_k - L_0)^2 - (L_j - L_0)^2|}{(1 - c_s)L_0 + c_sL_j} \geq k_s \frac{|(L_k - L_0)^2 - (L_j - L_0)^2|}{(1 - c_s)L_0 + c_sL_i} + k_s \frac{|(L_j - L_0)^2 - (L_i - L_0)^2|}{(1 - c_s)L_0 + c_sL_i} \tag{A.16}$$

The above inequality, after simplification for the absolute values and regrouping of terms reduces to

$$\frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_sL_j} - \frac{(L_j - L_0)^2 - (L_i - L_0)^2}{(1 - c_s)L_0 + c_sL_i} \geq \frac{(L_j - L_0)^2 - (L_k - L_0)^2}{(1 - c_s)L_0 + c_sL_i} - \frac{|(L_j - L_0)^2 - (L_k - L_0)^2|}{(1 - c_s)L_0 + c_sL_j} \tag{A.17}$$

The above may be further simplified as

$$[(L_j - L_0)^2 - (L_i - L_0)^2] \times \left[\frac{(1 - c_s)L_0 + c_sL_i - (1 - c_s)L_0 - c_sL_j}{((1 - c_s)L_0 + c_sL_j)((1 - c_s)L_0 + c_sL_i)} \right] \geq [(L_j - L_0)^2 - (L_k - L_0)^2] \times \left[\frac{(1 - c_s)L_0 + c_sL_j - (1 - c_s)L_0 - c_sL_i}{((1 - c_s)L_0 + c_sL_j)((1 - c_s)L_0 + c_sL_i)} \right] \tag{A.18}$$

from where it follows that

$$\frac{c_s(L_i - L_j)[(L_j - L_0)^2 - (L_i - L_0)^2]}{((1 - c_s)L_0 + c_sL_j)((1 - c_s)L_0 + c_sL_i)} \geq \frac{c_s(L_j - L_i)[(L_j - L_0)^2 - (L_k - L_0)^2]}{((1 - c_s)L_0 + c_sL_j)((1 - c_s)L_0 + c_sL_i)}$$

For the above to hold true, we must have

$$(L_j - L_0)^2 - (L_i - L_0)^2 \geq (L_k - L_0)^2 - (L_j - L_0)^2 \tag{A.19}$$

or, equivalently,

$$(L_j - L_0)^2 \geq \frac{(L_k - L_0)^2 + (L_i - L_0)^2}{2} \tag{A.20}$$

To prove the above, we note that from the initial conditions of the non-monotonic length changes we have $L_i \geq L_k$, therefore

$$(L_k - L_0)^2 \leq (L_i - L_0)^2 \tag{A.21}$$

Substituting $(L_i - L_0)^2$ for $(L_k - L_0)^2$ on the RHS of inequality (A.20), we get

$$(L_j - L_0)^2 \geq \frac{2(L_i - L_0)^2}{2} \tag{A.22}$$

Clearly, the above inequality is true because $L_i \leq L_j$, hence the proof. For other cases of non-monotonic changes, the triangle inequality may be proved likewise.

The proof of the metric property for the bending energy is based on similar ideas. In the following, we denote the angle before any deformations by ϕ_0 and the angles after bending at the i, j , and k instances by ϕ_i, ϕ_j , and ϕ_k respectively. The bending energy spent in changing the angle ϕ_i to ϕ_j is denoted by $E_b(\phi_i, \phi_j)$.

From the formulation of the bending energy in Eq. (3), it is straightforward to note that

1. $E_b(\phi_i, \phi_j) \geq 0$.
2. $E_b(\phi_i, \phi_j) = E_b(\phi_j, \phi_i)$.

For the proof of the triangle inequality, we proceed on lines similar to the one followed for the stretching energy by considering angle changes that are *monotonic* and

non-monotonic. For monotonic changes, we present the proofs for both decreasing and increasing angular changes. For non-monotonic angular changes, we present the proofs for two cases. The other cases may be proved analogously.

1. *Monotonically decreasing angle changes*: Let the angles at the instances i , j , and k be ϕ_i , ϕ_j , and ϕ_k , respectively. Further let $\phi_i \geq \phi_j \geq \phi_k$. For the triangle inequality we need to prove that

$$E_b(\phi_i, \phi_j) + E_b(\phi_j, \phi_k) \geq E_b(\phi_i, \phi_k) \quad (\text{A.23})$$

$$\begin{aligned} &\Rightarrow k_b |(\phi_j - \phi_o)^2 - (\phi_i - \phi_o)^2| \\ &\quad + k_b |(\phi_k - \phi_o)^2 - (\phi_j - \phi_o)^2| \\ &\geq k_b |(\phi_k - \phi_o)^2 - (\phi_i - \phi_o)^2|. \end{aligned} \quad (\text{A.24})$$

Simplifying, we have

$$\begin{aligned} &(\phi_i - \phi_o)^2 - (\phi_j - \phi_o)^2 + (\phi_j - \phi_o)^2 - (\phi_k - \phi_o)^2 \\ &\geq (\phi_i - \phi_o)^2 - (\phi_k - \phi_o)^2. \end{aligned} \quad (\text{A.25})$$

The above gives us an equality, thus proving the validity of the triangle inequality.

2. *Monotonically increasing angle changes*: For this case we consider the angle changes to have the following relationship: $\phi_i \leq \phi_j \leq \phi_k$. We thus have, for the triangle inequality,

$$\begin{aligned} &k_b |(\phi_j - \phi_o)^2 - (\phi_i - \phi_o)^2| \\ &\quad + k_b |(\phi_k - \phi_o)^2 - (\phi_j - \phi_o)^2| \\ &\geq k_b |(\phi_k - \phi_o)^2 - (\phi_i - \phi_o)^2|. \end{aligned} \quad (\text{A.26})$$

Taking into consideration the relationships between the angles and accounting for the absolute values, we get

$$\begin{aligned} &(\phi_j - \phi_o)^2 - (\phi_i - \phi_o)^2 + (\phi_k - \phi_o)^2 - (\phi_j - \phi_o)^2 \\ &\geq (\phi_k - \phi_o)^2 - (\phi_i - \phi_o)^2. \end{aligned} \quad (\text{A.27})$$

On simplification, we get an equality. \square

3. *Non-monotonically angle changes*

\circ Let $\phi_i \leq \phi_k \leq \phi_j$, The triangle inequality

$$E_b(\phi_i, \phi_j) + E_b(\phi_j, \phi_k) \geq E_b(\phi_i, \phi_k) \quad (\text{A.28})$$

may then be expanded as

$$\begin{aligned} &(\phi_j - \phi_o)^2 - (\phi_i - \phi_o)^2 + (\phi_j - \phi_o)^2 - (\phi_k - \phi_o)^2 \\ &\geq (\phi_k - \phi_o)^2 - (\phi_i - \phi_o)^2. \end{aligned} \quad (\text{A.29})$$

This inequality then simplifies to the inequality

$$(\phi_j - \phi_o)^2 \geq (\phi_k - \phi_o)^2. \quad (\text{A.30})$$

Since $\phi_k \leq \phi_j$, the above inequality holds.

\circ Consider the case where $\phi_k \leq \phi_i \leq \phi_j$. The triangle inequality then takes the following form:

$$\begin{aligned} &(\phi_j - \phi_o)^2 - (\phi_i - \phi_o)^2 + (\phi_j - \phi_o)^2 - (\phi_k - \phi_o)^2 \\ &\geq (\phi_i - \phi_o)^2 - (\phi_k - \phi_o)^2. \end{aligned} \quad (\text{A.31})$$

Simplifying, we have

$$(\phi_j - \phi_o)^2 \geq (\phi_i - \phi_o)^2, \quad (\text{A.32})$$

which is true, since $\phi_j \geq \phi_i$. \square

References

- [1] R.L. Anderson, Real-time gray-scale video processing using a moment generating chip, IEEE J. Robotics Automat. 1 (1985) 70–85.
- [2] C.C. Lin, R. Chellapa, Classification of partial 2D shapes using Fourier descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 9 (1987) 686–690.
- [3] R.L. Kashyap, R. Chellapa, Stochastic models for closed boundary analysis: representation and reconstruction, IEEE Trans. Inform. Theory 27 (5) (1981) 627–637.
- [4] A. Pentland, R.W. Picard, S. Sclaroff, Photobook: content-based manipulation of image databases, Int. J. Comput. Vision 18 (3) (1996) 233–254.
- [5] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vision 7 (1) (1991) 11–32.
- [6] T.F. Syeda-Mahmood, Data and model-driven selection using color regions, Int. J. Comput. Vision 21 (1/2) (1997) 9–36.
- [7] R.C. Bolles, R.A. Cain, Recognizing and locating partially visible objects: the local-feature focus method, Int. J. Robotics Res. 1 (1982) 57–82.
- [8] S. Sclaroff, A.P. Pentland, Modal matching for correspondence and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 17 (6) (1995) 545–561.
- [9] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, Int. J. Comput. Vision 14 (1995) 5–24.
- [10] M. Brady, H. Asada, Smoothed local symmetries and their implementation, Int. J. Robotics Res. 3 (1984) 36–61.
- [11] T. Phillips, A shrinking technique for complex object decomposition, Pattern Recognition Lett. 3 (1985) 271–277.
- [12] J. Chen, J.A. Ventura, Optimization models for shape matching of nonconvex polygons, Pattern Recognition 28 (6) (1995) 863–877.
- [13] L. Huang, M.J. Wang, Efficient shape matching through model-based shape recognition, Pattern Recognition 29 (2) (1996) 207–215.
- [14] I. Tchoukanov, R. Safaee-Rad, B. Benhabib, K.C. Smith, A new boundary-based shape recognition technique, in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 1992, pp. 1030–1037.
- [15] P. Cox, H. Maitre, M. Minoux, C. Ribeiro, Optimal matching of convex polygons, Pattern Recognition Lett. 9 (1989) 327–334.
- [16] P.J. van Otterloo, A Contour-Oriented Approach to Shape Analysis, Prentice-Hall, Hemel Hempstead, 1991.

- [17] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, J.S.B. Mitchell, An efficiently computable metric for comparing polygonal shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (3) (1991) 209–216.
- [18] W. Rucklidge, *Efficient Visual Recognition Using the Hausdorff Distance*, Springer, Berlin, 1996.
- [19] R. Azencott, F. Coldefy, L. Younes, A distance for elastic matching in object recognition, in: *Proceedings of the 13th International Conference on Pattern Recognition*, Vol. 1, 1996, pp. 687–691.
- [20] A.D. Bimbo, P. Pala, Visual image retrieval by elastic matching of user sketches, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (2) (1997) 121–132.
- [21] S.E. Sclaroff, *Modal matching: a method for describing, comparing, and manipulating digital signals*, Ph.D. Thesis, School of Architecture and Planning, Massachusetts Institute of Technology, 1995.
- [22] F.L. Bookstein, Principal warps: thin-plate splines and the decomposition of deformations, *IEEE Trans. Pattern Anal. Mach. Intell.* 11 (6) (1989) 567–585.
- [23] A. Yuille, P. Hallinan, Deformable templates, in: A. Blake, A. Yuille (Eds.), *Active Vision*, MIT Press, Cambridge, MA, 1992, pp. 21–38.
- [24] K. Hirata, T. Kato, Query by visual example, content-based image retrieval, in: A. Pirotte, C. Delobel, G. Gottlob (Eds.), *Advances in Database Technology-EDBT92*, Springer, Berlin, 1992.
- [25] R. Singh, R.M. Voyles, D. Littau, N.P. Papanikolopoulos, Grasping real objects using virtual images, *Proceedings of the IEEE Conference on Decision and Control*, 1998.
- [26] R. Singh, R.M. Voyles, D. Littau, N.P. Papanikolopoulos, Pose alignment of an eye-in-hand system using image morphing, *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.
- [27] H. Freeman, L. Davis, A corner-finding algorithm for chain-coded curves, *IEEE Trans. Comput.* 26 (1977) 297–303.
- [28] T. Pavlidis, S.T. Horowitz, Segmentation of plane curves, *IEEE Trans. Comput.* 23 (1974) 860–870.
- [29] B.K. Ray, K.S. Ray, Determination of optimal polygon from digital curve using L_1 norm, *Pattern Recognition* 26 (4) (1993) 505–509.
- [30] J.J. Brault, R. Plamondon, Segmenting handwritten signatures at their perceptually important points, *IEEE Trans. Pattern Anal. Mach. Intell.* 15 (9) (1993) 953–957.
- [31] M.A. Fischler, R.C. Bolles, Perceptual organization and curve partitioning, *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (1) (1986) 100–105.
- [32] B.K. Ray, K.S. Ray, An algorithm for detection of dominant points and polygonal approximation of digitized curves, *Pattern Recognition Lett.* 13 (12) (1992) 849–856.
- [33] P. Zhu, P.M. Chirlian, On critical point detection of digital shapes, *IEEE Trans. Pattern Anal. Mach. Intell.* 17 (8) (1995) 737–748.
- [34] I. Pavlidis, R. Singh, N.P. Papanikolopoulos, On-line handwriting recognition using physics-based shape metamorphosis, *Pattern Recognition* 31 (11) (1998) 1589–1600.
- [35] I. Pavlidis, N.P. Papanikolopoulos, A curve segmentation algorithm that automates deformable-model based target tracking, Technical Report TR 96-041, University of Minnesota, 1996.
- [36] T.W. Sederberg, E. Greenwood, A physically based approach to 2D shape blending, *Comput. Graphics* 26 (2) (1992) 25–34.
- [37] J. Barros, J. French, W. Martin, P. Kelly, M. Cannon, Using the triangle inequality to reduce the number of comparisons required for similarity-based retrieval, in: *SPIE, Storage and Retrieval for Still Images and Video Databases*, Vol. 2670, 1996, pp. 392–403.
- [38] W.A. Burkhard, R.M. Keller, Some approaches to best-match file searching, *Commun. ACM* 16 (4) (1973) 230–236.
- [39] M. Shapiro, The choice of reference points in best-match file searching, *Commun. ACM* 20 (5) (1977) 339–343.

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, the M.S. in Computer Science from the University of Minnesota in 1997 and the Ph.D. in Computer Science from the University of Minnesota in 1999. Currently he is a scientist in Exelixis Inc., in San-Francisco where he is working on the area of molecular shape recognition and its applications in computational prediction of pharmacologically relevant molecular properties. In addition to the above areas, his research interests include computer vision, image morphing, document image analysis, and applications of virtual reality in vision-based robotics.

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 at the University of Minnesota. His research interests include Computer Vision, Pattern Recognition, and Robotics. He has authored or coauthored more than 90 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.