Pattern Recognition, Vol. 25, No. 11, pp. 1277–1288, 1992
Printed in Great Britain

0031-3203/92 $5.00 + .00
Pergamon Press Ltd

BINARY SHAPE RECOGNITION USING THE MORPHOLOGICAL SKELETON TRANSFORM

P. E. TRAHANIAS
Department of Electrical Engineering, University of Toronto, Toronto, Ontario, Canada M5S 1A4

(*Received* 1 *July* 1991; *in revised form* 5 *March* 1992; *received for publication* 16 *March* 1992)

Abstract—Binary image representation by its morphological skeleton transform has been proposed in the past as an information preserving representation. In this paper this skeleton representation is adopted as the start point for the development of a binary shape recognition scheme. A skeleton matching algorithm is presented that efficiently characterizes the similarity between two skeletons as a distance measure. This skeleton matching algorithm resembles the well known from speech processing elastic matching technique.

Morphological skeleton transform Binary images Shape representation Shape recognition
Skeleton matching

1. INTRODUCTION

The ultimate goal in many computer vision systems is the recognition of the shapes in the image and possibly the description (understanding) of the image.[1] Shape recognition is usually accomplished by adopting a representation scheme that reduces a shape to a (small) number of shape descriptors and then classifying the shape according to the values of the descriptors. Alternatively, a shape can be represented by a set of shape primitives and possibly a set of primitive connection operators and then a structural recognition scheme can be used to perform the shape recognition task.[2,3]

A large number of shape descriptors has been reported so far.[1] Pavlidis[4] characterizes the shape descriptors as external or internal to distinguish between those which examine the boundary and those which examine the whole area of a shape. He also classifies them as information preserving and information nonpreserving depending on whether it is possible to reconstruct the original shape from the shape descriptor. From the pattern recognition point of view the above classifications are irrelevant as far as correct recognition is achieved. A representation scheme can potentially realize this goal if it carries all the information that is important for the class of shapes considered. In practice, it turns out that a representation scheme may be application dependent and this can be decided only through experimentation.

The skeleton of a shape has been proposed as an internal and information preserving descriptor which reduces a shape to an axial representation. The skeleton is actually a distance transform since it can be defined as the set of points whose distance from the nearest boundary is locally maximum.[5] Among other techniques, the skeleton can be obtained by morphological set operations; this is usually referred to as the morphological skeleton transform (MST) to distinguish from the skeletons obtained using other approaches.[6,7] It was pointed out by Maragos and Schafer[6] that the MST unifies and generalizes previous approaches to skeletonization. A fast algorithm for skeleton decomposition has also been provided by the authors.

In this paper a scheme is presented for binary shape recognition which uses the MST representation. This scheme is based on a skeleton matching algorithm (SMA) which renders the similarity between two MSTs as a distance measure. The SMA resembles the well-known elastic matching technique that is widely used in the context of speech processing.[8,9] Based on the distance measure, a given shape is classified as that shape from which its distance is minimum. Approaches to shape recognition based on mathematical morphology have also been proposed recently for binary[3,10] and range images.[11] These approaches are based mainly on shape decomposition into primitive components.

In what follows a brief introduction to morphological operations and the MST is first presented in Section 2 for the self completeness of the paper. The SMA is then described in Section 3 and the recognition scheme is presented in Section 4. Results of the application of the proposed SMA are described in Section 5 and the paper concludes with a brief discussion in Section 6.

2. THE MORPHOLOGICAL SKELETON TRANSFORM

2.1. *Morphological operations*

The fundamental morphological operations, which form the basis of most morphological transformations, are erosion and dilation.[7,12] Erosion and dilation are initially defined for sets; however, their definitions can be extended to functions via the Umbra transformation.[13] Since this paper deals with binary images (sets) only, the definitions of these operations will be given for this case only. The reader interested in function morphological operations may see references (7, 13).

Erosion $\ominus$ is a shrinking operation and dilation $\oplus$ is an expanding operation. For any binary shape X

$$X \ominus B = \bigcap_{b \in B} X - b = \{z : (B + z) \subseteq X\} \quad (1)$$

$$X \oplus B = \bigcup_{b \in B} X + b = \{x + b : x \in X \text{ and } b \in B\} \quad (2)$$

where B represents a simple shape acting as a probe. It is called structuring element in morphological terminology. The output of an erosion is the set of translation points such that the translated structuring element is contained in the input set X. Similarly, the output of a dilation is the set of translated points such that the translate of the reflected structuring element $\check{B} = \{-b : b \in B\}$ has a non-empty intersection with X.

Two additional fundamental operations can be defined as combinations of erosion and dilation. Opening $\circ$ is an erosion followed by a dilation and closing $\bullet$ is a dilation followed by an erosion

$$X \circ B = (X \ominus B) \oplus B \quad (3)$$

$$X \bullet B = (X \oplus B) \ominus B. \quad (4)$$

2.2. *MST representation*

The MST of a binary shape can be obtained by successive erosions and openings of the shape by a structuring element B. The basic algorithm has continuous and discrete versions. In the continuous case the MST of a binary shape X is calculated by

$$SK(X) = \bigcup_{r > 0} S_r(X) = \bigcup_{r > 0} [(X \ominus rB) \setminus (X \ominus rB)_{drB}] \quad (5)$$

where "\" denotes set difference, rB an open disk of radius r and drB a closed disk of infinitesimally small radius dr.

For the discrete case the MST of a digital binary shape X can be obtained by a similar algorithm

$$SK(X) = \bigcup_{n=0}^{N} S_n(X) = \bigcup_{n=0}^{N} [(X \ominus nB) \setminus [(X \ominus nB) \circ B]] \quad (6)$$

$$N = \max\{n : (X \ominus nB) \neq \varnothing\}.$$

The sets S_n, $n = 0, 1, \ldots, N$, are referred to as the skeleton components of X with respect to B. From the sets S_n, X can be reconstructed exactly as

$$X \circ kB = \bigcup_{k \leq n \leq N}^{N} S_n \oplus nB, \quad 0 \leq k \leq N. \quad (7)$$

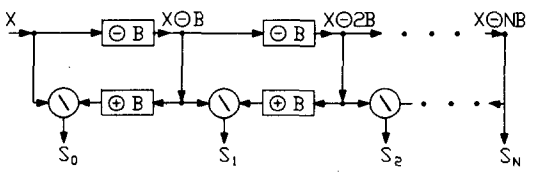

Fig. 1. Skeleton decomposition algorithm.

If we let $k = 0$ then all the skeleton components are used and $X \circ kB = X$ (exact reconstruction). If $k = m$, $m > 0$, then we obtain the opening (smoothed version) of X by mB.

Maragos and Schafer[6] have given two algorithms for skeleton decomposition and reconstruction which operate in linear time, as opposed to the initial algorithm described by Serra[12] which operates in quadratic time. The decomposition algorithm (Fig. 1) takes into account the relations

$$nB = \underbrace{B \oplus B \oplus \cdots \oplus B}_{n \text{ times}}, \quad n = 1, 2, \ldots \quad (8)$$

and

$$X \ominus nB = X \ominus (B \oplus B \oplus \cdots \oplus B)$$

$$= \underbrace{(((X \ominus B) \ominus B) \ominus \cdots \ominus B)}_{n \text{ times}}, \quad n = 1, 2, \ldots \quad (9)$$

i.e. the erosion of X by nB is performed by n successive erosions.

The MST is translation invariant, antiextensive, and idempotent. Illustrative examples of the MST of some simple geometrical shapes are shown in Fig. 2.

The information contained in all the skeleton subsets S_n can be compactly represented by adopting the skeleton function (SKF) which is defined as follows:

$$[SKF(X)](x, y) = \begin{cases} n + 1, & (x, y) \in S_n(X) \\ 0, & (x, y) \notin S_n(X) \end{cases} \quad (10)$$

Since the skeleton subsets are disjoint sets, the SKF is a single valued function. MST pixels that are associated with large SKF values represent the main body of the shape whereas those associated with small values represent boundary details.

3. THE SKELETON MATCHING ALGORITHM

If one is interested in exact shape matching the skeleton matching would be a trivial task since two shapes X_1 and X_2 could be considered the same iff $SK(X_1) = SK(X_2)$. If we consider the translation invariancy of the MST, either X_1 or X_2 in this equality test can be replaced by a translated version of it so that the two shapes are located in the same position in the image plane. However, in practice, the above equality test would (almost) always fail since shape variations result in shape representations that are different even for similar shapes. Consequently, what is needed is a skeleton matching scheme that would compare two skeletons and provide an estimate of how similar they are.

Before proceeding to the formulation of the algorithm, which constitutes the main contribution of our work, the objectives that this algorithm should fulfill will be outlined. In the rest of this section it will be assumed that the shapes to be matched have been normalized with respect to scale and rotation since the MST is not invariant under scaling and rotation. For the sake of simplicity and in order to avoid unnecessary translations in the matching process, it is also assumed that a

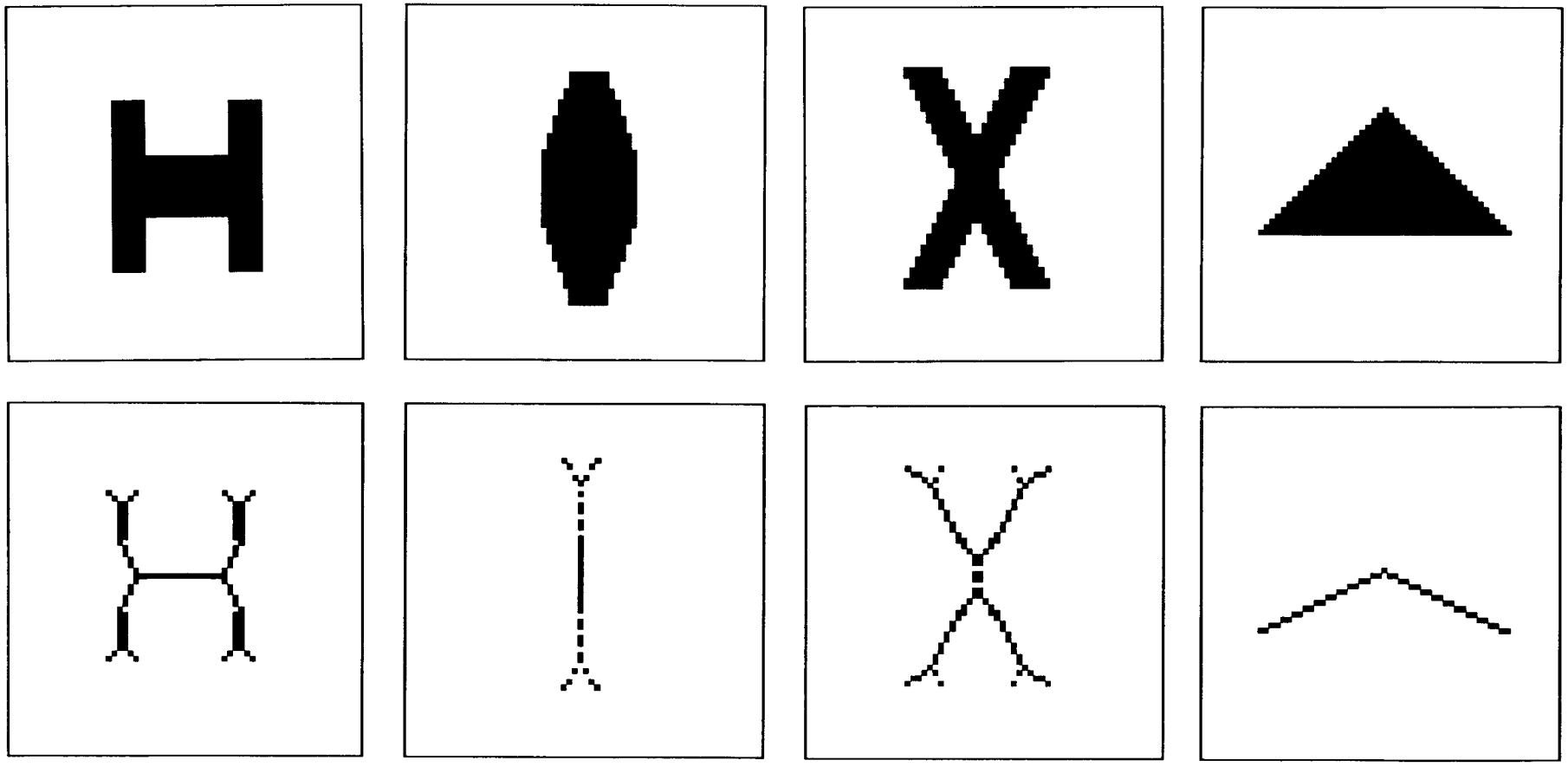

Fig. 2. MST examples of binary shapes with respect to the "RHOMBUS" structuring element.

translation normalization has also taken place. These tasks are not explained further here in order to concentrate on the SMA itself; they are discussed in more detail in the next section.

The SMA should at first produce results that conform with our intuition about skeletal matching. Consequently, it should be able to compare MSTs of different lengths and geometrical distributions in a flexible manner. This implies that an alignment of the MSTs under consideration should be performed. A kind of local alignment should be allowed in order to take care of local matching in the whole process of skeleton matching. This idea is similar to the idea behind the elastic matching technique that is used in speech processing[8,9] and also in on-line character recognition.[14] However, in the elastic matching technique the situation is much simpler since it is dealing with one-dimensional signals only which are characterized by a start and end point. One is thus able to match the start and end points of the two signals to be aligned and then allow for a non-linear alignment of the time axes so that a predefined distance measure is minimized. An application of this technique to the recognition of shapes having a closed boundary and no holes has been presented in reference (15). In this approach, a boundary representation scheme is adopted which encodes a shape to an ordered sequence that represents the Euclidean distance between the center of mass and all contour pixels in the shape. The start point for the matching process is selected as the contour pixel that lies on the major axis of the shape.

In the skeleton matching process, which is applicable to any binary shape, the added difficulty is encountered that there are no definitive start and end points to be matched nor an apparent sequence of the pixels that belong to an MST can be defined. Consequently, it should proceed by searching for the local similarities between the two MSTs or equivalently by matching parts that result in a small "cost". In the matching process the significance of each part that is examined in the MST representation should also be taken into account. In other words, when matching parts that represent similar shape components (i.e. the main bodies or the boundaries) a weighting factor that strengthens their similarity should be used. On the contrary, when matching parts that represent different shape components the weighting factor should weaken their similarity. Computationally, the SMA should be efficient in order to allow for implementation in image analysis systems. Therefore, exhaustive searches that may break down its performance for large image arrays should be avoided. This situation can arise in the elastic matching technique when a large time warping range is allowed.

Based on the above observations, a matching algorithm has been designed which compares the MSTs of two shapes and renders a distance measure that expresses their similarity (small distance measures indicate similar shapes whereas large distance measures indicate dissimilar shapes). This algorithm tries to align the two MSTs under the constraint that the "cost" associated with the alignment of each MST component (pixel) is small. This is achieved by matching each pixel in the first MST with its nearest in the second MST. Pixels in the second MST not considered in this process are subsequently matched with their nearest ones in the first MST. The cost associated with each match is computed as the distance between the two pixels weighted by a coefficient that is a function of their SKFs. This weighting is important in order to ensure that the matching of pixels representing different parts of the two shapes is expensive whereas the matching of parts representing similar parts is cheap. The sum of all the costs is the final distance measure computed by the algorithm. A formal presentation of it follows in a Pascal like notation.

Algorithm. SMA
Input. S_1, S_2/* S_1 and S_2 stand for the two MSTs to be matched, $S_1 = SK(X_1)$, $S_2 = SK(X_2)$ */
Output. D /* A distance measure that expresses the similarity between S_1 and S_2*/
Method.
Step 1: (*Initialization*)
Set $D = 0$;
n_1 = number of pixels in S_1;
n_2 = number of pixels in S_2;
let $n_1 \geq n_2$, otherwise change the roles of S_1 and S_2
Step 2: **for** each pixel $(x, y) \in S_1$ **do**
begin
find its nearest pixel $(x', y') \in S_2$;
/* nearest is meant in the Euclidean sense */
compute $d = W[(x, y), (x', y')] \cdot R[(x, y), (x', y')]$;
/* d represents the cost for matching pixel (x, y) with pixel (x', y')
W stands for weighting factor and is computed as
$W[(x, y), (x', y')] = |SKF(x, y) - SKF(x', y')| + 1$
R stands for length of the straight line that connects the grid locations (x, y) and (x', y') */
increase D by d;
mark (x', y') as visited;
end;
Step 3: **for** each unvisited pixel $(x'', y'') \in S_2$ **do**
begin
find its nearest pixel $(x, y) \in S_1$;
compute $d = W[(x'', y''), (x, y)] \cdot R[(x'', y''), (x, y)]$;
increase D by d;
end;
Step 4: **stop**.

A few comments on this algorithm follow. The

computed distance between two MSTs by the SMA is not the minimum one since, although the expression $R[(x,y),(x',y')]$ in the computation of d is guaranteed to be minimum, the product $W[(x,y),(x',y')]\cdot R[(x,y),(x',y')]$ may or may not be minimum and hence the algorithm may reach a suboptimal solution. However, the choice of this "local" cost function is justified since (a) it yields a computationally cheap algorithm since the search for (x',y') stops after locating the nearest pixel (usually after a few iterations). In the case that the global minimum would be searched for, an exhaustive search of all the pixels in S_2 is required and the risk exists in matching pixels that are located "far away" (b) It resembles human perception of shape similarity which usually starts from a similarity of parts examination and then based on the overall geometry (spatial relations) the final decision is made.[16]

By matching each pixel in the first MST with its closest in the second MST and accumulating the costs associated with each match into the global distance measure, the SMA actually performs a local non-linear alignment of the two MSTs in both directions in the image plane as opposed to the elastic matching which considers only one direction. The cost accumulation in the global distance measure provides a means for determining the overall similarity or dissimilarity of two shapes. In other words, the distance measure is the final criterion for judging two shapes as similar or not. The use of the weighting factor $W(\cdot)$ guarantees that each pair of pixels being matched has a large contribution to the whole distance measure if they represent different parts of the two shapes whereas it has a small contribution if they represent similar parts of the two shapes.

The computational complexity of this algorithm is difficult to compute analytically since the number of iterations needed to find the pixel $(x',y')\in S_2$ that is the nearest to $(x,y)\in S_1$ is not known in advance. Similarly, not known in advance is the number of unvisited pixels from S_2 and hence the number of iterations in step 3. In the worst case the whole $N\times N$ image grid would have to be searched in order to find the nearest pixel to a given pixel, and thus the worst case computational complexity of step 2 is $O(n_1N^2)$ and if we consider that n_v pixels from S_2 have been visited in step 2 (in the worst case $n_v=1$) then the computational complexity of step 3 is $O((n_2-n_v)N^2)$. Since n_1 and n_2 (number of pixels in S_1 and S_2, respectively) are usually in the order of N, the above analysis results in a worst case complexity $O(N^3)$. In practice, however, only a small number of iterations is needed in the search for a nearest pixel which, for all practical purposes, can be taken as a constant k. Thus steps 2 and 3 have complexities $O(n_1k)$ and $O((n_2-n_v)k)$, respectively, and taking as $n=\max\{n_1,n_2-n_v\}$ we conclude that in practice the complexity of the whole algorithm is $O(n)$ which is roughly equal to $O(N)$. The assumption of very small k was tested in practice and was found to be valid.

4. SHAPE RECOGNITION BASED ON SKELETON MATCHING

A shape recognition scheme is presented in Fig. 3 which is based on the SMA described in the previous section. The distance of the shape being examined from each of a set of reference shapes is computed by the SMA and the shape is classified as that reference shape from which its distance is a minimum. The reference shapes are not stored in their initial form; instead their MSTs are stored and this provides for large computational savings.

Since the MST representation is not invariant under scale and rotation, a normalization process is applied before the MST computation concerning scale and rotation. Translation normalization is also applied so that the shapes are placed in a predetermined position in the image plane. The shape normalization ensures representation invariance (and consequently recognition invariance) under rotations and scalings. These tasks are performed as follows.

Translation normalization. The axial system is moved so that the origin coincides with the center of mass of the shape. That is, if $(\bar{x},\bar{y})$ are the coordinates of the center of mass and $f(x,y)$ represents the initial binary shape, the translated function $f_t(x,y)$ is given as $f_t(x,y)=f(x+\bar{x},y+\bar{y})$.

Scale normalization. The shape is enlarged or reduced so that the number of pixels n_p in it becomes a predefined constant β. This can be accomplished by transforming the function $f(x,y)$ to $f_s(x,y)=f(x/\alpha,y/\alpha)$, where $\alpha=\sqrt{(\beta/n_p)}$.[17] Both translation and scale normalization can be performed in one step by transforming $f(x,y)$ to $f_{ts}(x,y)$, where $f_{ts}(x,y)=f(\bar{x}+x/\alpha,\bar{y}+y/\alpha)$.

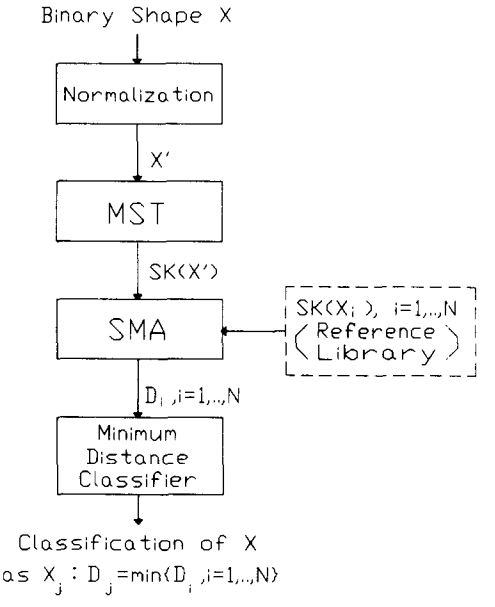

Fig. 3. Binary shape recognition scheme.

Rotation normalization. The shape is rotated by an angle ϕ so that the angle between the major axis of the shape and the horizontal direction takes a predefined value ψ. Rotation is performed by computing the new location (x', y') of each pixel (x, y) as follows:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}. \tag{11}$$

The initial angle θ between the major axis and the horizontal direction is computed as

$$\theta = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{12}$$

where μ_{pq} denotes the $(p+q)$th order central moment and is computed as

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y). \tag{13}$$

5. EXPERIMENTAL RESULTS

The proposed SMA has been implemented in the C language and some results concerning its application are presented here. For illustrative purposes, the detailed matching process is presented in Fig. 4 for two simple binary shapes X_1 = "+" and X_2 = "H". In Fig. 4(a) the two binary shapes are shown and their MSTs, with respect to the "RHOMBUS" structuring element, are presented in Fig. 4(b). These shapes have intentionally been scaled using a small value for the constant β ($\beta = 125$) in order to keep the number of pixels in the MSTs small and simplify the presentation of the matching process. The information contained in the skeleton functions is shown in Fig. 4(c) where for each pixel $(x, y) \in \mathrm{MST}_i$, $i = 1, 2$, the value of the skeleton function $[SKF(X_i)](x, y)$, $i = 1, 2$, is plotted. In Fig. 4(d) the matching of the two skeletons is shown. In this figure the two MSTs are superimposed. For purposes

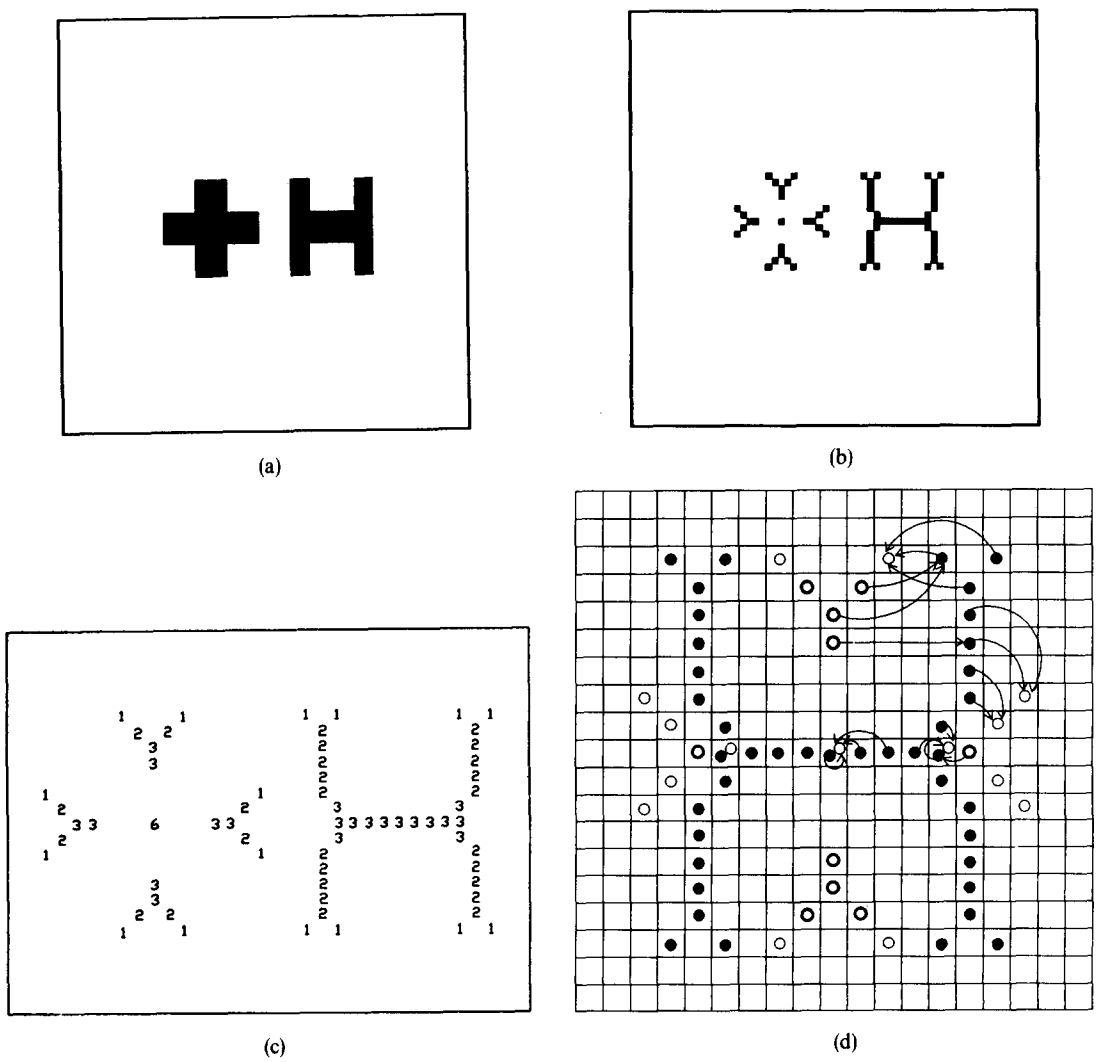

Fig. 4. Illustration of the skeleton matching process: (a) binary shapes; (b) their MSTs; (c) their skeleton functions; (d) skeleton matching process.

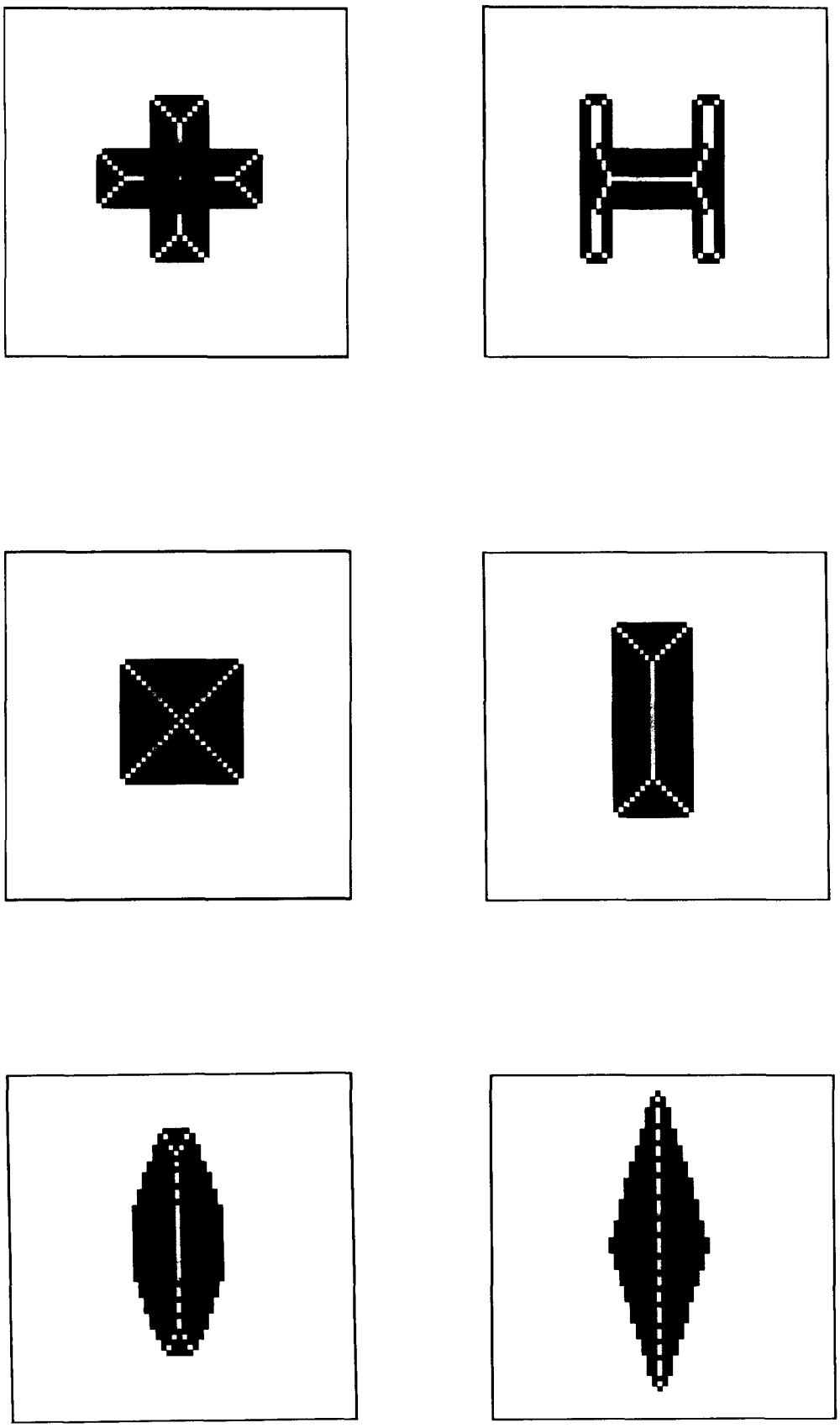

Fig. 5. Set of binary images used as test set. The MSTs are superimposed on the shapes. From left to right and top to bottom: "+", "H", "■", "rectangle", "ellipse", "rhombus", "X", "T", "triangle", "⊓", "⊟", "□".

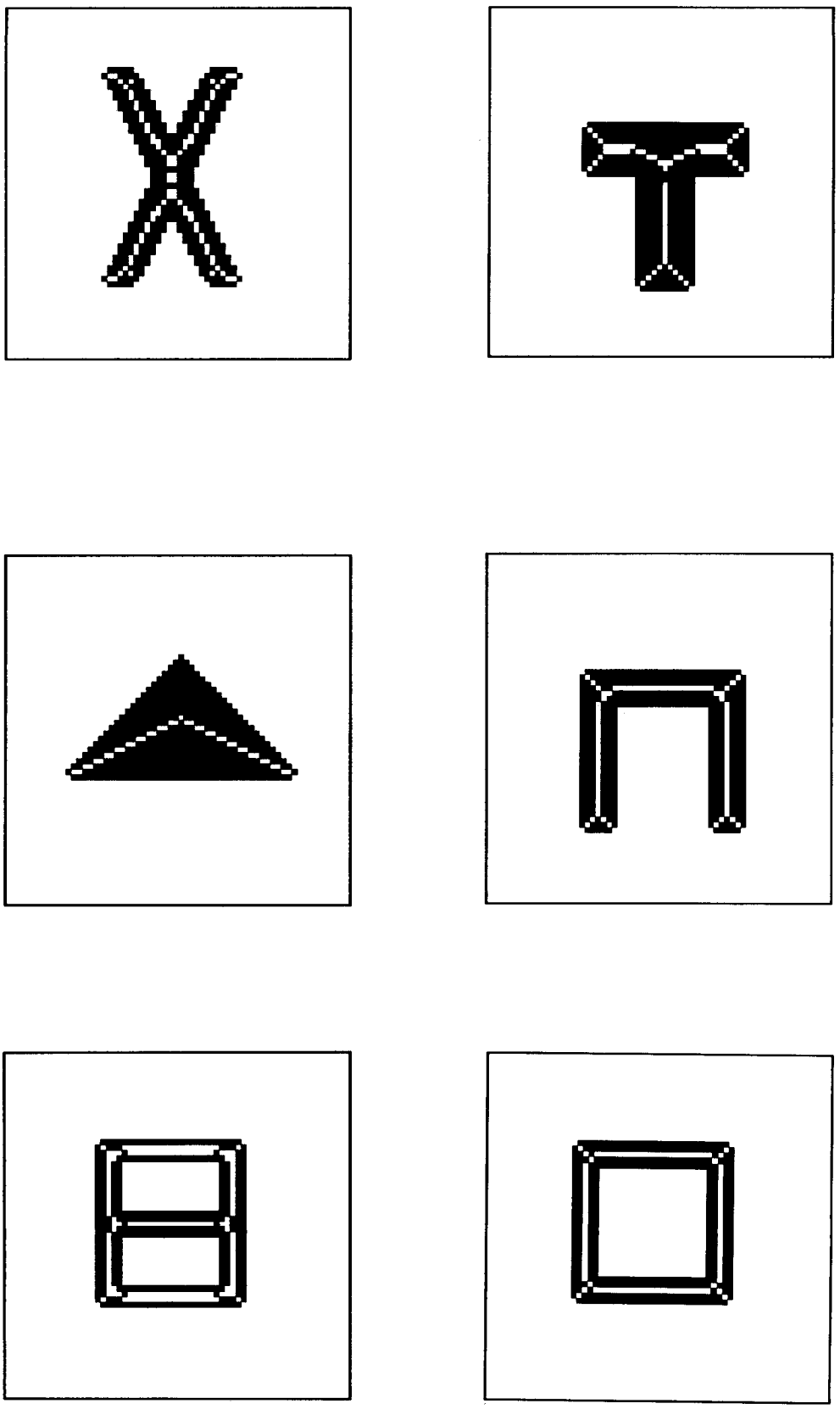

Fig. 5. (*Continued.*)

Table 1. Confusion matrix for the 12 shapes of Fig. 5

	Distances ($\times 10^{-1}$) computed by the SMA											
Shape	+	H	■	rectangle	ellipse	rhombus	X	T	triangle	⊓	⊟	□
+	0	175	107	137	265	301	184	126	265	191	168	146
H	175	0	87	289	540	639	148	187	502	107	90	151
■	107	87	0	80	185	192	135	90	175	187	256	276
rectangle	137	289	80	0	26	70	98	139	345	282	369	294
ellipse	265	540	185	26	0	22	170	268	513	433	693	559
rhombus	301	639	192	70	22	0	235	353	583	509	782	622
X	184	148	135	98	170	235	0	151	619	130	202	190
T	126	187	90	139	268	353	151	0	361	226	301	219
triangle	265	502	135	345	513	583	619	361	0	303	1039	569
⊓	191	107	187	282	433	509	130	226	303	0	153	69
⊟	168	91	256	369	693	782	202	301	1039	153	0	50
□	146	151	276	294	559	622	190	219	569	69	50	0

of clarity the pixels of MST_1 are plotted using the symbol "o", whereas the pixels of MST_2 are plotted using the symbol "•". It is noted that the number of pixels in MST_2 is greater than the number of pixels in MST_1. A graphic notation is used to denote the matching process. In this notation, each arrow that connects two pixels from the two MSTs indicates that these pixels are matched, i.e. $p \rightarrow p'$, $p \in MST_i$, $p' \in MST_j$, $i, j = 1, 2$, $i \neq j$, means that when p is considered, p' is found as its nearest pixel. The pixels of MST_1 which are marked with a bold borderline are not visited in step 2 of the SMA and thus they are matched in step 3. This can also be verified from the direction of the arrows. For the sake of simplicity only a part of the skeleton matching process is illustrated (upper right corner of the two MSTs) since the rest is symmetrical to this. The distance measure D computed for these two shapes is $D = 220$.

The set of geometrical shapes shown in Fig. 5 has been used as a test set for the SMA. This set consists of 12 shapes which are shown after being normalized. For the scale normalization a value of $\beta = 500$ has been selected. In this figure, the MSTs of the shapes are also shown superimposed on the actual shapes. The distance measures computed by the SMA for the binary shapes of Fig. 5 are given in the form of a confusion matrix in Table 1. As can be verified, similar shapes result in much smaller distance measures than other shapes, which coincides with our perception of shape similarity. Shape "⊓" for example, appears to be very similar to shape "H" and also similar to shape "□". Similarly, shapes "rectangle", "ellipse" and "rhombus" appear as similar.

A second experiment has been conducted concerning the recognition of the shapes shown in Fig. 5 after being corrupted by a "boundary deformation" procedure. This procedure produces a noisy version of a binary shape in two steps: (a) for each contour pixel of the initial shape a pixel is randomly selected from the set consisting of the contour pixel and the background pixels neighboring it, and (b) the value of the selected pixel is changed with a probability γ. Higher deformations can be produced by increasing γ or by repeating this process δ times for a given γ. This process affects the shape representation by skewing the MST and by introducing spurious pixels in the MST at or near the shape contour. Sample noisy shapes, along with their MSTs, are shown in Fig. 6 for $\delta = 3$ and 4, and $\gamma = 10\%$. This value of γ has been kept constant in this experiment whereas δ has been varied from 1 to 10. For each value of δ ten deformed versions of each shape have been produced. In summary, a set of 1200

Table 2. Recognition errors for the shapes of Fig. 5 deformed at various levels of δ

	δ									
Shape	1	2	3	4	5	6	7	8	9	10
+	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
■	0	0	0	0	0	0	0	0	0	0
rectangle	0	0	0	0	0	0	0	0	0	3
ellipse	0	0	1	3	3	5	4	6	6	7
rhombus	0	0	0	0	0	0	4	2	4	6
X	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0
triangle	0	0	0	0	0	0	3	2	4	5
⊓	0	0	0	0	0	0	0	0	0	0
⊟	0	0	0	0	0	0	0	0	0	0
□	0	0	0	0	0	0	0	0	0	0

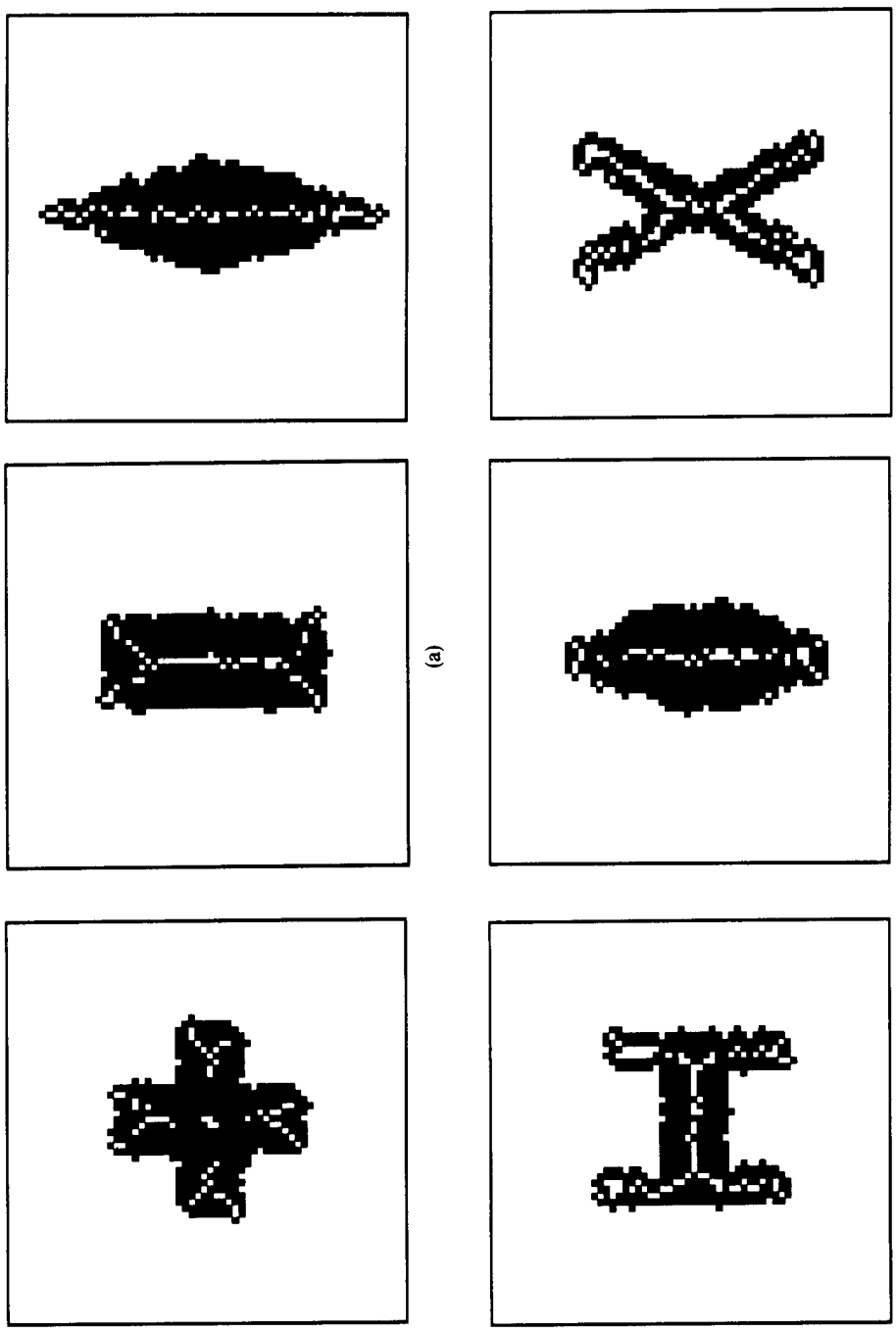

Fig. 6. Noisy shapes produced with the "boundary deformation" procedure: (a) $\delta = 3$; (b) $\delta = 4$.

deformed shapes has been produced (100 for each shape, or 120 for each value of δ). This set has been used as a test set, and the set of initial shapes has been used as the reference library in the experiment (12-class problem). The recognition errors for the various values of δ and for each shape are given in Table 2. As can be verified from this table, no errors were made for (a) $\delta = 1, 2, 3$ and all the classes of shapes, and (b) all δ values and eight out of the twelve shape classes. The majority of errors were observed for the "ellipse" that was misrecognized mostly as "rectangle". The total number of errors was 68 which accounts for a recognition rate of 94.33% for this specific set of noisy shapes. It should be noted at this point that with the application of an "opening–closing" filter to the deformed shapes all but a very few recognition errors were corrected.

6. DISCUSSION

A binary shape recognition scheme has been presented in this paper. This scheme adopts the MST as the shape representation formalism. In the heart of this shape recognition scheme lies the proposed SMA which is responsible for the decision regarding the similarity between two shapes. The SMA attempts to perform a local skeleton matching that is weighted according to the significance (skeletal function) of the skeleton points in each representation. The SMA is based on similar concepts as the elastic matching technique but in order to achieve computational efficiency it reaches a suboptimal solution instead of the globally optimal one.

Shape normalization is performed before the application of the SMA in order to ensure invariance in shape representation (and recognition), under rotations and scalings. This process is not necessarily optimal with respect to the matching performed and cost minimization with respect to normalization could be applied. Such a solution has not been employed, however, due to the increase in the computational cost associated with it. This is due to the fact that cost minimization would require examination of many possible normalizations and internal skeleton point correspondences that would make the algorithm prohibitively slow.

The SMA has been tested in practice using a set of geometrical shapes and its performance has been found to be very promising. Its robustness when applied to noisy data has been demonstrated using a set of deformed shapes at various levels. Besides being accurate, the proposed approach possesses some very important characteristics:

- It is simple and intuitively appealing since it expresses the similarity between two shapes based on the alignment of their MSTs which can be thought of as the thinned sketches of the shapes.
- It is computationally very efficient since both the MST and the SMA operate in $O(N)$ time.
- It can be applied to any type of binary shape without making any assumptions about its outer form or geometrical distribution.

Extensive testing of the SMA is however still needed and is planned for real shapes, e.g. machine parts, characters, etc. Other future work on this subject includes the extension of the SMA to graytone images using the morphological skeleton for graytone images[6] which is defined as the pointwise sum of the skeleton components S_n:

$$S_n = (f \ominus ng) - [(f \ominus ng) \circ g], \quad 0 \leq n \leq N \qquad (14)$$

where f represents the graytone image function, $N = \max\{n: f \ominus ng \neq -\infty\}$ and g the structuring element which can be a binary or a graytone function.

Acknowledgements—The reviewers' suggestions and comments were very helpful and greatly appreciated by the author.

REFERENCES

1. D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice-Hall, Englewood Cliffs, New Jersey (1982).
2. P. M. Griffin and B. L. Deuermeyer, A methodology for pattern matching of complex objects, *Pattern Recognition* **23**, 245–254 (1990).
3. L. G. Shapiro, R. S. MacDonald and S. R. Sternberg, Ordered structural shape matching with primitive extraction by mathematical morphology, *Pattern Recognition* **20**, 75–90 (1987).
4. T. Pavlidis, Algorithms for shape analysis of contours and waveforms, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-2**, 301–312 (1980).
5. A. K. Jain, *Fundamentals of Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey (1989).
6. P. A. Maragos and R. W. Schafer, Morphological skeleton representation and coding of binary images, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-34**, 1228–1244 (1986).
7. P. A. Maragos and R. W. Schafer, Morphological systems for multidimensional signal processing, *Proc. IEEE*, Vol. 78, pp. 690–710 (1990).
8. F. Itakura, Minimum prediction residual principle applied to speech recognition, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-23**(1), 67–72 (1975).
9. C. Myers *et al.*, Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-28**, 623–635 (1980).
10. I. Pitas and A. N. Venetsanopoulos, Morphological shape decomposition, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI-12**, 38–45 (1990).
11. I. Pitas and A. Maglara, Range image analysis by using morphological signal decomposition, *Pattern Recognition* **24**, 165–181 (1991).
12. J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, New York (1982).
13. S. R. Sternberg, Grayscale morphology, *Comput. Vision Graphics Image Process.* **35**, 333–355 (1986).
14. C. C. Tappert, Cursive script recognition by elastic matching, *IBM J. Res. Dev.* **26**, 765–771 (1982).
15. L. Gupta and M. D. Srinath, Invariant planar shape recognition using dynamic time alignment, *Pattern Recognition* **21**, 235–239 (1988).
16. I. Rock, *An Introduction to Perception*. Macmillan, New York (1975).
17. A. Khotanzad and J. H. Lu, Classification of invariant image presentations using a neural network, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-38**, 1028–1038 (1990).

About the Author—PANAGIOTIS TRAHANIAS received the B.S. in physics from the University of Athens, Greece, in 1985 and the Ph.D. in computer science from the National Technical University of Athens, Greece, in 1988. From 1985 to 1989 he served as a research assistant at the Institute of Informatics & Telecommunications, National Center for Scientific Research "Demokritos", Greece, and from 1990 to 1991 he worked at the same institute as a postdoctoral research associate. Since October 1991 he has been with the Department of Electrical Engineering, University of Toronto, Toronto, Ontario, where he holds a postdoctoral research position. His research interests include pattern recognition, signal/image processing and computer vision and image analysis. Dr Trahanias is a member of the IEEE Society.