Euclidean skeleton via centre-of-maximal-disc extraction

Carlo Arcelli and Gabriella Sanniti di Baja

A skeletonization algorithm is presented, characterized by two main features: invariance under isometric transformations of the pattern, and recoverability. The algorithm is driven by the Euclidean distance map of the pattern. Invariance under isometric transformations is guaranteed due to the use of the Euclidean distance to compute the distance map; recoverability is guaranteed by the inclusion in the set of the skeletal pixels of the centres of the maximal discs. The skeletonization algorithm includes a beautifying step and a pruning step, which favour the use of the skeleton for shape analysis tasks.

Keywords: Euclidean distance map, centre-of-maximal-disc, skeletonization

The skeletonization of a digital pattern is a process leading to the extraction of a linear subset, the skeleton, which is spatially placed along the medial region of the pattern. The skeleton is a stick-like representation of the pattern and, depending on the problem domain, is required to account for different shape properties, such as symmetry, elongation, width and contour curvature. The recent literature on skeletonization focuses on the development of more efficient algorithms, favouring the use of the skeleton for investigating the shape of complex patterns.

We present a sequential skeletonization algorithm requiring a reduced computational effort, and producing a skeleton which is a suitable representation of the pattern. Skeletonization is driven by a distance map, so that the computational cost is independent of pattern thickness. The use of the distance map also favours the representative power of the skeleton, as it allows ascribing the correct distance label to the skeletal pixels. Moreover, it facilitates the detection of the centres of the maximal discs which, when included in the skeleton, ensure complete recovery of the input pattern. Skeleton stability under pattern rotation is guaranteed by the use of the Euclidean metric to build the distance map. Inclusion in the skeleton of the centres of the maximal discs distinguishes the algorithm proposed in this paper from algorithms, recently proposed, to compute the Euclidean skeleton[1–3].

We do not elaborate on how to get the Euclidean distance map (EDM), and how to find therein the centres of the maximal discs (cmd), as these results are already available in the literature[4–8]. Since the set of the cmd is generally not connected, further pixels are identified to guarantee skeleton connectedness. The operations to obtain a skeleton which is unit wide, well shaped, and rid of branches meaningless in the problem domain, are discussed.

PRELIMINARIES

We refer to a binary picture, rid of salt-and-pepper noise, where $F=\{1\}$ and $F=\{0\}$ are the foreground and the background, respectively. The pixels of F and $\bar{F}$ are also referred to as black and white pixels. Without loss of generality, we suppose that F is a connected set, regardless of its connectivity order, and does not touch the frame of the picture. The 8-metric and the 4-metric are respectively adopted for F and $\bar{F}$. Since the skeleton S of F is a subset of F, the 8-metric is chosen also for S.

The neighbours of p are the eight pixels n_i ($i = 1, 8$) found when going clockwise around p, starting from the pixel placed at West of p. The n_i, i odd, and the n_i, i even, are also referred to as horizontal/vertical neighbours and diagonal neighbours, respectively. The set $N(p) = \{n_i, i = 1,8\}$ is called the neighbourhood of p.

A black pixel for which all the n_i, i odd, are black, is termed a 4-internal pixel. A black pixel for which at least one n_i, i odd, is white, is termed a contour pixel.

The connectivity number $C_8(p)$ and the crossing number $X_4(p)$, respectively count the 8-connected components and the 4-connected components of black pixels in the neighbourhood of p. $C_8(p)$ and $X_4(p)$ are

Istituto di Cibernetica, CNR, Via Toiano 6, 80072 Arco Felice, Naples, Italy (Email: imag@arco.na.cnr.it)

Paper received: 3 June 1992; revised paper received: 1 September 1992

computed[9,10] by the following expressions, where $n_9 = n_1$, and $\overline{n_k} = 1 - n_k$:

$$C_8(p) = \sum_{k=1}^{4} (\overline{n}_{2k-1} - \overline{n}_{2k-1} \times \overline{n}_{2k} \times \overline{n}_{2k+1})$$

$$X_4(p) = 1/2 \times \sum_{k=1}^{8} |n_{k+1} - n_k|$$

Through this paper, $C_8(p)$ and $X_4(p)$ are computed for pixels belonging to the EDM. Hence, a suitable binary version of $N(p)$ has to be taken into account.

The Euclidean distance map of F with respect to $\overline{F}$ is a replica of the picture, where the pixels are labelled with their distance from $\overline{F}$. Since the operators to be used within skeletonization mostly apply to an equal degree on distances and on the square of distances, we

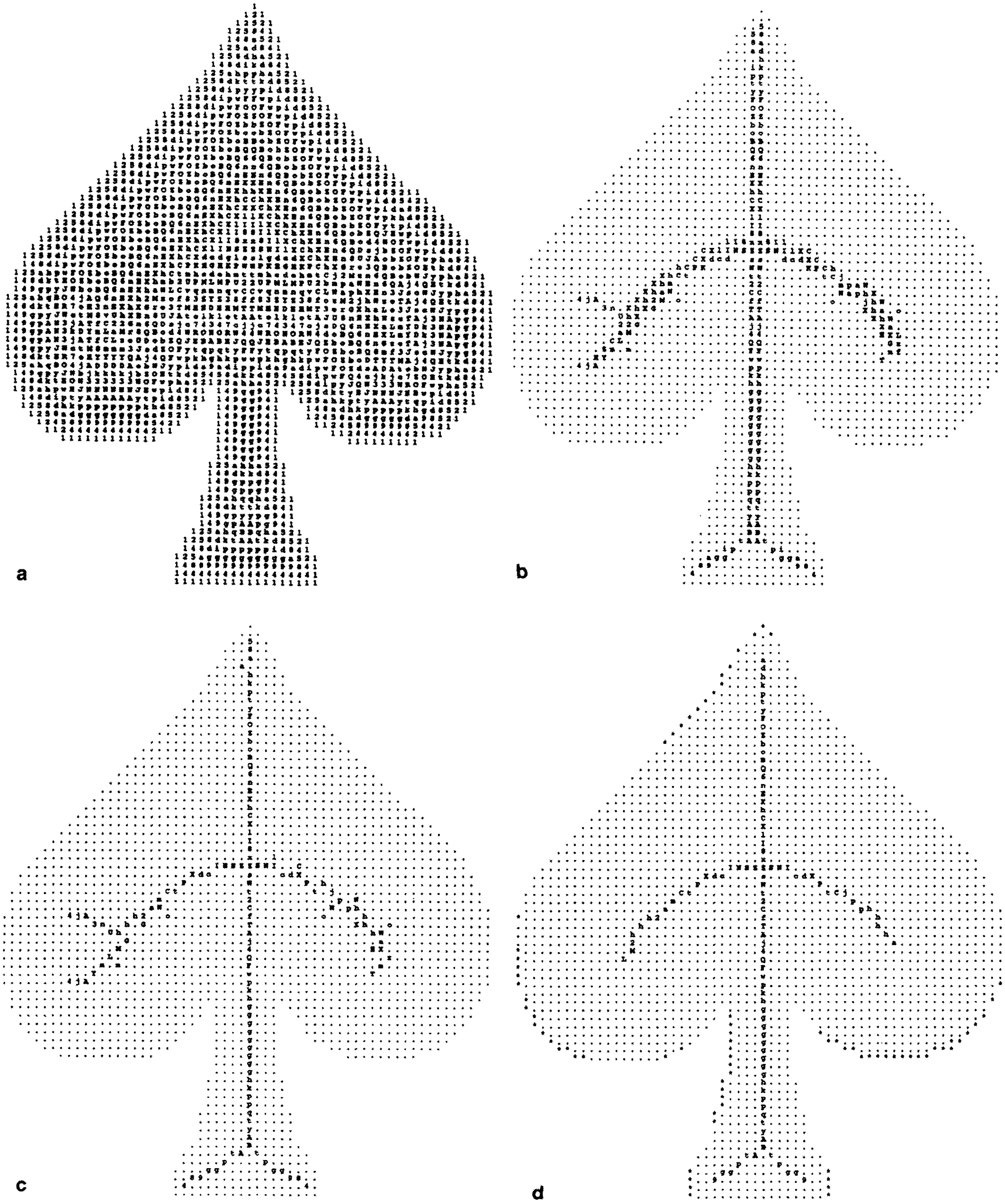

Figure 1. (a) The EDM. Numbers lower-case and capital letters cyclically indicate the square of increasing Euclidean distances; (b) set of the intrinsic and induced skeletal pixels; (c) thinned set of skeletal pixels; (d) skeleton

consider the pixels of the EDM labelled with the square value of the Euclidean distance (see Figure 1a). We compute the square root only when this result is necessary within specific sub-tasks, so that most of the computation on the EDM is performed by using integers. Unless differently specified, the letter used to indicate a pixel is also used for its associated label.

Any black pixel p of the EDM can be interpreted as the centre of a digital disc D_p, which includes each pixel q such that $d_e(p, q) < \sqrt{p}$, where $d_e(p, q)$ is the Euclidean distance between p and q. Obviously, D_p does not include pixels of $\bar{F}$.

D_p is a maximal disc if it is not completely overlapped by the disc centred on any other pixel of the EDM. The union of the maximal discs has the same size and shape as F. If the centres of the maximal discs are detected and ascribed to the skeleton, F can be recovered from S by associating the corresponding disc to each cmd, and the skeletonization process is called *reversible*. Recovery can be achieved by computing the reverse Euclidean distance transformation[8] from the skeleton.

In the city-block and chessboard distance map, called DM in the following, the centres of the maximal discs coincide with the pixels labelled not less than their neighbours (only the n_i with i odd should be understood as neighbours in case of the city-block DM). On the EDM this coincidence does not occur, and a comparison between the label of p and the label of each of its neighbours is not sufficient to establish whether D_p is a maximal disc. Two suitable lookup tables, Hlut and Dlut, have been introduced[8] which allow us to identify the cmd. The minimal label that a horizontal or a vertical neighbour q should have, so that D_q completely overlaps D_p, is stored in Hlut(p). In turn, Dlut(p) contains the minimal value for a diagonal neighbour q, such that D_q completely overlaps D_p. Then, the pixel p is a cmd if the label of each horizontal/vertical neighbour is less than Hlut(p) and the label of each diagonal neighbour is less than Dlut(p).

The skeleton S consists of simple arcs and curves[11]. Any arc is located in the middle of a region of F, and any curve (loop of S) corresponds to a hole of F.

After a suitable set of operations has been applied to get S, its pixels can be classified as end points, branch points and normal points. The end points are generally defined as pixels having just one neighbour in S, since they are expected to be extremes of skeleton arcs found in correspondence with pattern protrusions. However, in case of pattern protrusions whose elongation is rather small, the previous definition does not identify the extremes of the corresponding skeleton arcs. For instance, see Figure 2, where the starred pixel is extreme of the shortest arc in the skeleton. If the adopted set of operations has not been able to remove that pixel, the only way to get rid of it and of the associated meaningless unitary skeleton arc is to resort to a pruning step. To this purpose, the starred pixel has to be recognized as an end point, notwithstanding the number of neighbours in S is greater than one. Thus, an alternative definition is preferable, which classifies as end points the pixels having one 4-component of neighbours not in S. Any arc including an end point is called a peripheral skeleton branch. Pixels having more than two neighbours in S, without being end points, are called branch points. Branch points indicate crossings among arcs of S. The remaining pixels of S are the normal points.

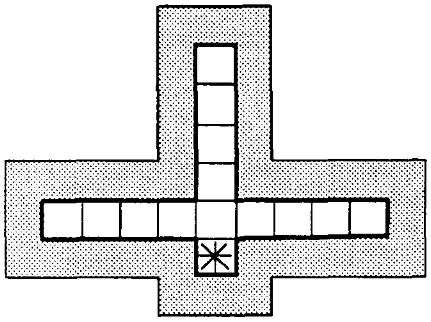
Figure 2. Classical definition of an end point, as a pixel having just one neighbour in the skeleton, does not allow classification of the starred pixel as an end point

Except for the end points, S is rid of pixels superfluous to preserve 8-connectedness. In this sense, S is termed a *unit wide 8-connected set*.

To guarantee complete recovery of F starting from its skeleton, S should include all the centres-of-maximal-discs. However, inclusion of all the centres is not compatible with unit thickness of S. Lack of complete recovery due to reduction to unit width of the set of the centres-of-maximal-discs does not meaningfully affect the possibility of using S to represent F, since only a few pixels, generally spread along the contour of F, are not recovered. In fact, the presence of a centre-of-maximal-disc in the skeleton is not always essential for recovery purposes, because its associated disc may be completely overlapped by the union of the discs associated to two or more centres-of-maximal-discs, still remaining in the skeleton.

EXTRACTING THE SKELETON

DM case

Skeletonization can be accomplished by inspecting the distance map to identify and mark therein the candidates to belong to the skeleton. Let us call these pixels skeletal pixels. On the DM, the constraints on the labels in the neighbourhood of a pixel p allow one to find all the possible configurations, where a conspicuous number of skeletal pixels are embedded. These pixels, called *intrinsic skeletal pixels*, can be identified by means of parallel 3 × 3 operations. Their set includes, but does not coincide with, the set of the centres-of-maximal-discs. The set of the intrinsic skeletal pixels is not guaranteed to be connected. Components of intrinsic skeletal pixels have to be suitably linked to each other[12,13] by means of further skeletal pixels, called *induced skeletal pixels*, which constitute connecting paths.

Two components of intrinsic skeletal pixels are expected to be linked if they include two pixels p and q such that, in the region of the pattern in between p and q, the labels of the DM have a strictly monotonical behaviour. Suppose $p > q$. The path may be built either from p to q, or from q to p. The procedure to build the path should accept as induced skeletal pixels the pixels which are neighbours of each other and have labels

					1	1	1					
					1	②	1					
					1	②	1					
1	1	1	1	1	1	②	1	1	1	1	1	1
1	2	2	2	2	2	②	2	2	2	2	2	1
1	2	3	3	3	3	3	3	3	3	3	2	1
1	2	3	4	4	4	4	4	4	4	3	2	1
1	2	3	4	5	5	5	5	5	4	3	2	1
1	2	3	4	5	6	6	6	5	4	3	2	1
1	2	3	4	5	6	⑦	$\boxed{6}$	$\boxed{5}$	$\boxed{4}$	$\boxed{3}$	$\boxed{2}$	$\boxed{1}$
1	2	3	4	5	6	6	6	5	4	3	2	1
1	2	3	4	5	5	5	5	5	4	3	2	1
1	2	3	4	4	4	4	4	4	4	3	2	1
1	2	3	3	3	3	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3. Two components of intrinsic skeletal pixels (encircled labels). The decreasing path (square-framed labels) starting from the encircled 7 does not link that pixel to the component of pixels labelled 2

step-by-step increasing (decreasing). Moreover, to avoid thickening of the path, the procedure should identify only one pixel in each connected component of neighbours, candidates to be the next element in the path. Selection of a decreasing path cannot be done, since it does not guarantee skeleton connectedness. In fact, several decreasing paths can be started from p, so that the path really constructed may not link p to q. As an example referring to the chessboard DM case, see Figure 3. On the contrary, if the path is built through increasing labels from q, the structure of the distance map ensures that the path converges towards the component of skeletal pixels including p.

The intrinsic skeletal pixels include the centres of the maximal discs in the DM, as well as other pixels referred to as saddle points. The saddle points have in their neighbourhood two components of pixels with a smaller label, and one or two components of pixels with a higher label. Having one or two components of pixels with higher labels depends on whether the pixel delimits an elongated saddle configuration or is centred in a sand-glass configuration. In the first case, the saddle point is a neighbour of an equally labelled pixel, which is the centre of a maximal disc. Figure 4 refers to chessboard distance maps. In the presence of pattern subsets having even thickness, an extended 4×4 neighbourhood should be used to identify saddle points. However, it is possible to avoid the use of extended neighbourhoods if one accepts as saddle point any pixel p for which at least one of the following conditions is satisfied in $N(p)$:

1. there is more than one 8-connected component of pixels labelled more than p
2. there is more than one 4-connected component of pixels labelled less than p
3. there exists a triple of neighbours n_i, n_{i+1}, n_{i+2} (i odd) which are all labelled p.

1	1	1	1	1				1	1	1	1	1
1	2	2	2	1				1	2	2	2	1
1	2	3	2	①	1	1	1	①	2	3	2	1
1	2	2	2	1				1	2	2	2	1
1	1	1	1	1				1	1	1	1	1

a

1	1	1	1	1				
1	2	2	2	1				
1	2	3	2	1				
1	2	2	2	1				
1	1	1	1	①	1	1	1	1
				1	2	2	2	1
				1	2	3	2	1
				1	2	2	2	1
				1	1	1	1	1

b

Figure 4. On the chessboard distance map, the encircled pixels are saddle points in an elongated saddle configuration (a) and in a sand-glass configuration (b)

The above components are counted by the connectivity number $C_8(p)$ and the crossing number $X_4(p)$, respectively. To check Condition (1) (Condition (2)), $N(p)$ is binarized by setting to 1 the neighbours labelled more (less) than p, and to 0 the remaining neighbours. Note that Conditions (2) and (3) may also be satisfied by some cmd. Thus, the set of pixels accepted due to any of the previous conditions includes, but does not coincide with, the set of the saddle points.

The saddle points are the only intrinsic skeletal pixels from which paths can be grown, and are centred within pattern necks. Thus, in correspondence with convex patterns, saddle points should not exist, and the set of centres-of-maximal-discs should be connected. However, due to the shape of the adopted disc, necks also exist in correspondence of regions perceived as bounded by convex contour arcs. As an example, refer to the chessboard distance map of the pattern shown in Figure 5a.

The set of the skeletal pixels is seldom unit wide everywhere, so that not all the pixels marked on the distance map can be confirmed in the skeleton. Further skeletal pixels may not be confirmed in the skeleton when performing pruning aimed at removing meaningless peripheral branches.

Accepting as skeletal pixels the centres-of-maximal-discs is useful to have a reversible skeletonization, and to provide a reliable criterion to identify the pixels from which peripheral skeleton branches originate. These pixels will be classified as end points in the resulting skeleton. Instead of using a blind sequential removal process, the pixels from which peripheral skeleton branches originate are found in parallel according to a geometric criterion. They are the centres of the maximal discs sharing with the contour of F a (relevant) connected percentage of their contour.

When the disc is a polygon with a small number of sides oriented along fixed directions, inclusion of the

				①				
			1	$\boxed{1}$	1			
		1	1	②	1	1		
	1	1	2	$\boxed{2}$	2	1	1	
①	$\boxed{1}$	②	$\boxed{2}$	③	$\boxed{2}$	②	$\boxed{1}$	①
	1	1	2	$\boxed{2}$	2	1	1	
		1	1	②	1	1		
			1	$\boxed{1}$	1			
				①				

a

1	1	1	1	1	1	1
1	2	2	2	2	2	1
1	2	3	3	3	2	1
1	2	3	④	3	2	1
1	2	3	3	3	2	1
1	2	2	2	2	2	1
1	1	1	1	1	1	1

b

Figure 5. Different number and location of the centres-of-maximal-discs (encircled labels) found in the chessboard DM of a differently oriented square. Square framed labels indicate saddle points

centres of the maximal discs in the set of the skeletal pixels is not sufficient to guarantee skeleton stability under rotation. A classical example is given in Figure 5, with reference to the chessboard metric. For certain pattern orientations, a single disc is enough to fit a large portion of the contour of F. When the pattern is rotated, a number of discs, aligned along the bisector of a contour subset and characterized by increasing size, are necessary to fit the subset. Skeleton stability is greatly favoured by employing the EDM. In fact, when the Euclidean distance is considered, the shape of the disc is no longer that of a polygon with few sides, having fixed orientation, but is circular (as soon as the label of the centre is large enough). Accordingly, nearly the same number of discs fits the shape of the contour, whichever the orientation of the pattern.

EDM case

On the DM, the constraints on label distribution in the neighbourhood of any pixel p are quite strong, since the label of n_i can only assume one of three possible values, namely $p-1, p, p+1$. Moreover, when components of pixels having both smaller and larger labels exist in $N(p)$, these components are divided by pixels labelled as p. On the contrary, on the EDM any value ranging between $\sqrt{p}-1$ and $\sqrt{p}+1$ is possible for a neighbour n_i (i odd), while any value ranging between $\sqrt{p}-\sqrt{2}$ and $\sqrt{p}+\sqrt{2}$ is possible for a neighbour n_i (i even). Moreover, pixels respectively labelled less and more than p are allowed to be neighbours of each other. Thus, 3×3 operations to identify all the intrinsic skeletal pixels cannot be devised.

Intrinsic and induced skeletal pixels on the EDM

The cmd can be detected by referring to the lookup tables Hlut and Dlut, previously introduced. Their set almost exhausts the set of the intrinsic skeletal pixels, since cmd are often also found in pattern necks, where saddle points would be expected. Moreover, the existence of saddle points in correspondence of pattern subsets perceived as bounded by convex contour arcs is strongly reduced, due to the rounded shape of the Euclidean digital disc. The cmd which are also saddle points are not necessary for recovery purposes. Their presence in the set of the skeletal pixels is anyway indispensable. In fact, as already pointed out in the DM case, connecting paths have to be grown starting from the saddle points to guarantee skeleton connectedness. To identify the saddle points which are not cmd, Conditions (1)–(3) are parallelwise checked on the EDM. In Figure 6, examples of detected saddle points are shown.

The detection of the induced skeletal pixels is accomplished by growing paths along the direction of the steepest gradient in the EDM, starting from any intrinsic skeletal pixel. The path is built by accepting as skeletal pixels, the pixels for which the maximal value of the gradient is positive. Accordingly, the first pixel in a path is the neighbour q of an intrinsic skeletal pixel p, such that $q > p$, and the gradient of q with respect to p is maximum in the neighbourhood of p. The gradient is computed as follows:

$$\text{grad}\,(q) = w_i\,(q-p)$$

where $w_i = 1$ or $w_i = 1/\sqrt{2}$, depending on whether q is a horizontal/vertical neighbour or a diagonal neighbour of p, and where p and q should be understood as the square root of the corresponding labels in the EDM. The next pixel, if any, in the path is similarly found, by analysing a suitable subset of the neighbours of q, selected depending on the direction $p \rightarrow q$.

Generally, two paths are expected to be started from each saddle configuration. The two paths should originate from the same pixel, only when this pixel is a saddle point centred in a sand-glass configuration. All the remaining saddle configurations are elongated, and include more than one saddle point, so that the two paths, possibly originating from the configuration, start from different pixels. Indeed, the growth of two paths is tried from any pixel found to be an intrinsic skeletal pixel, since we do not use separate detectors to distinguish the different types of intrinsic skeletal pixels. Note that once a path has been grown up starting from a pixel p; a second path can be originated from p only if its first pixel is not a neighbour of the first pixel in the path already built. Otherwise, the second path would only thicken the first path.

A list L with the coordinates of the marked intrinsic and induced skeletal pixels is built, while the EDM is inspected.

The set of skeletal pixels obtained after the path growing process is a set of pixels marked on the EDM. In Figure 1b, intrinsic and induced skeletal pixels are shown superimposed over the input pattern.

Hole filling

The set of the skeletal pixels may include one-pixel-sized spurious holes. These are created in the presence of paths converging towards common positions. A spurious hole is a black pixel which is not a skeletal pixel, while all its four horizontal/vertical neighbours are skeletal pixels (see Figure 7). Due to the existence of one-pixel-sized spurious holes, the number of loops in the skeleton would be larger than the number of holes in the input pattern. Thus, spurious holes have to be filled in to guarantee topology preservation. Filling

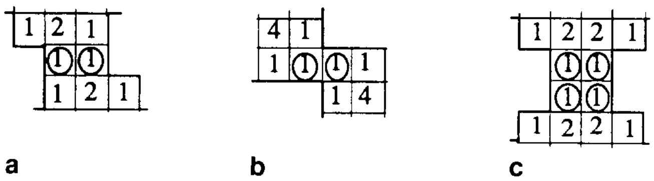

Figure 6. On the Euclidean distance map, the encircled pixels satisfy one of the Conditions (1)–(3) for saddle point detection

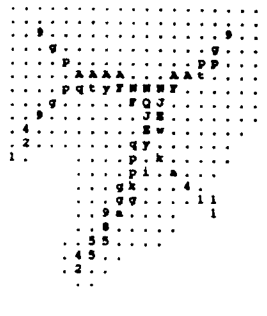

Figure 7. A one-pixel-sized spurious hole created by the convergence of connecting paths

does not create an excessive increase in the thickness of the set of the skeletal pixels, since only few sparse spurious holes are likely to exist.

To perform hole filling, the list L of the skeletal pixels is exhausted. The horizontal/vertical neighbours of any skeletal pixel p are inspected on the array. Any not yet marked neighbour q is marked provided that all its neighbours n_i, i odd, are marked. The coordinates of any pixel filling a spurious hole are stored in L.

We take advantage of the inspection of L, done to accomplish hole filling, to also process the list, so as to properly organize data for the successive thinning step. Each pixel in L is accessed and, if found to be 4-internal in the set of the skeletal pixels, a flag H is set in the list. The flag H is also set for the pixels filling the spurious holes.

Final thinning

Final thinning is performed to get a unit wide 8-connected skeleton, and is accomplished by removing the marker from suitable skeletal pixels. We refer to this operation as *unmarking*. The skeletal pixels are directly accessed through the list L. Two inspections of L are used to have a better result for skeleton centrality within thick regions of skeletal pixels, as those found in correspondence with the lobes of the pattern in Figure 1b. The flag H, introduced during the hole filling step, is used to skip, within the first inspection of L, the pixels which are 4-internal in the initial set of the skeletal pixels. During the first inspection of L, unmarking is attempted only for unflagged pixels. If unmarking is not accomplished, the flag H is set on the list. During the second inspection of L, only the pixels with flag H are accessed. The flag H is removed for any pixel in L in correspondence with unmarking accomplished on the array. At the end of the second inspection of L, the pixels with flag H constitute the unit-wide skeleton. These pixels are the only ones still marked on the array (see Figure 1c).

In correspondence with every skeletal pixel p accessed through L, the set $M(p)$, binary version of $N(p)$, is built as follows: $m_i = 1$ if n_i is marked, $m_i = 0$ otherwise. During the first and second inspections of the list, the unmarking condition is, respectively:

$$y \geqslant 1 \quad \text{(during the 1st inspection)} \qquad (4)$$

$$(1-z) \times y \geqslant 1 \quad \text{(during the 2nd inspection)} \qquad (5)$$

$$\text{where} \quad z = m_1 \times m_3 \times m_5 \times m_7,$$

$$y = (m_1 \times m_3 \times \bar{m}_6 + m_3 \times m_5 \times \bar{m}_8 + m_5 \times m_7 \times \bar{m}_2 + m_7 \times m_1 \times \bar{m}_4),$$

$$\bar{m}_i = 1 - m_i.$$

Conditions (4)–(5) preserve topology and prevent shortening of branches when sequentially checked on the EDM. Requiring that the two terms $(1-z)$ and y be different from 0 prevents, respectively, hole creation (during the second inspection of L) and disconnections. Creation of holes is prevented by requiring that at least one m_i, i odd, is different from 1. As for the first inspection of L, holes are not created because 4-internal pixels are not accessed at all.

In order that one of the products in y, say $m_1 \times m_3 \times \bar{m}_6$, be equal to 1, n_1 and n_3 must be marked pixels, while n_6 must not be marked. Since unmarking is attempted only if p is not 4-internal, at most, one further n_i, i odd, can be a marked pixel. Let this neighbour be n_5. Even if the product $m_3 \times m_5 \times \bar{m}_8$ is equal to 0, no disconnection occurs when unmarking p, since the diagonal marked neighbour n_8 remains linked to the component of skeletal pixels, including n_3 and n_5, through n_1. Thus, Condition (4) guarantees connectedness preservation when unmarking any pixel p (not 4-internal). The term y is also tailored to correctly thin down two-pixel-wide columns and rows of skeletal pixels.

Shortening is prevented because the unmarking condition is false when applied to pixels having just one neighbour in the set of the skeletal pixels. Two-pixel-wide columns (rows) of skeletal pixels are reduced to unit width without any shortening, since as soon as the marker is removed from one of the two pixels on the tip of the column (row), the condition becomes false for the second pixel on the tip, which remains in the skeleton as an end point.

Beautifying step

The aim of this step is to improve the aesthetics of the skeleton. A more pleasing appearance is obtained by straightening the zigzags of the skeleton branches, mostly caused by the operations employed to accomplish unmarking during final thinning. Zigzags are eliminated by shifting the marker from a skeletal pixel to one of its (unmarked) neighbours.

The skeletal pixels with flag H are directly accessed by using the list L. For any pixel p, the crossing number is computed to count the number of components of neighbours of p not in the skeleton. If $X_4(p) = 1$, p is classified as an end point, and a flag E is set in the list to record such an event and to drive the following pruning step. Otherwise, the connectivity number is computed to count the number of components of skeletal pixels in $N(p)$. If $C_8(p) = 2$, the possibility of shifting the marker from p to any of its neighbours is checked. If p has two marked neighbours n_i and n_{i+2} (i even), and n_{i+1} is an unmarked pixel with a positive label (i.e. n_{i+1} does not belong to the background), the marker is shifted from p to n_{i+1}. Shifting does not destroy skeleton loops. This would happen only if n_{i+1} is the centre of a one-pixel-sized hole, whose existence is prevented either by the cleaning process to which the input pattern is subjected, or by the hole filling process. Also, skeleton connectedness is not altered. In fact, the connection between the two components of skeletal pixels in $N(p)$ is equally guaranteed by p as well as by n_{i+1}.

Shifting the marker from p to n_{i+1} may cause either

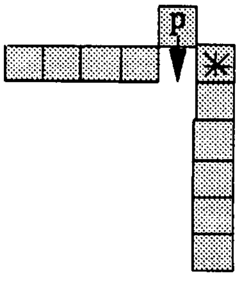

Figure 8. Shifting the pixel p *in the position pointed to by the arrow, to reduce skeleton jaggedness, changes the starred pixel into a superfluous pixel*

of n_i and n_{i+2} to become a superfluous skeletal pixel (see Figure 8). Thus, when shifting of p is performed, the pixels n_i and n_{i+2} are checked, and their connectivity number computed. Unmarking is accomplished for any of these pixels if the connectivity number is equal to 1 and the number of marked neighbours is greater than one.

Pruning step

If the reverse distance transformation[8] is applied to the skeleton, the recovered pattern differs from the input pattern by the absence of a thin set of peripheral pixels. The difference, due to the requirement of having a unit wide 8-connected skeleton, is generally disregardable, and the skeleton can be considered as a representation of the input pattern. In particular, a correspondence exists between peripheral skeleton branches and pattern protrusions. Skeleton branches which do not allow recovery of significant portions of the input pattern should be pruned. Keeping them in the skeleton may result in a structure which is difficult to investigate, and which is not adequate for a shape analysis task. This is particularly true when the pattern to be skeletonized may appear in different orientations. As an example, see Figure 9, showing the results of applying skeletonization to the pattern of Figure 1, digitized in different orientations. The resulting sets are dramatically different from each other, and their use in a recognition process is inhibited.

We describe in the following a pruning criterion exclusively based on low-level operations, but clever enough to produce a skeleton free of noisy branches, and to significantly reduce skeleton sensitivitiy to pattern translation, rotation and scaling. This has been shown by the experimentation we have performed on a conspicuous number of test patterns. A more sophisticated technique, able, for instance, to treat skeletons of patterns which have undergone non-isometric transformations, would require high-level operations, and is beyond the scope of this paper.

The modified skeleton represents a pattern which can be understood as obtained by flattening some protrusions of the input pattern. Thus, the pruning criterion can be based on the geometric properties which allow discrimination between the protrusions regarded as significant and those to be flattened during a smoothing process. Generally, tapering protrusions which, sticking out from the pattern, become narrower and narrower, are candidates to be flattened. Accordingly, pruning should be accomplished only if the sequence of distance labels found while tracing a peripheral branch is not decreasing. Only protrusions moderately jutting are expected to be flattened.

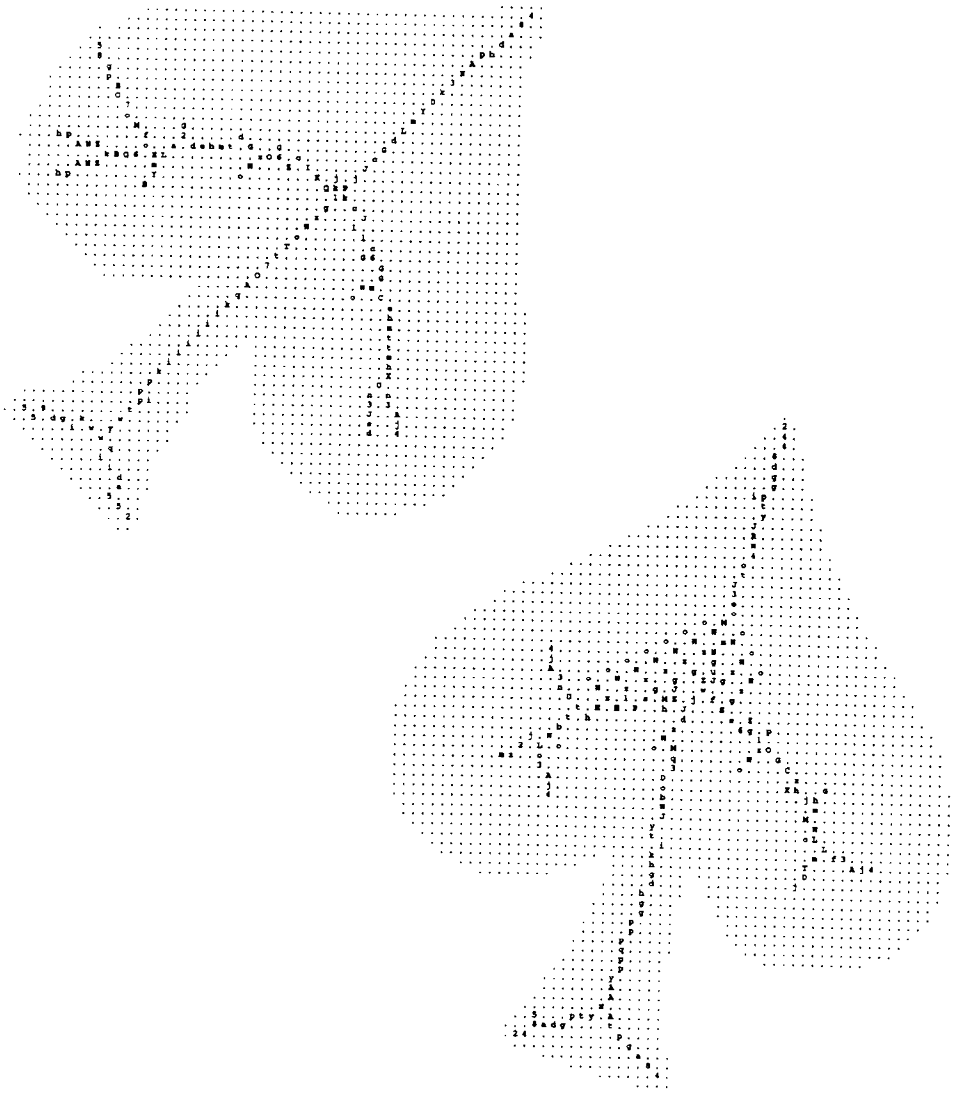

Figure 9. Sets of skeletal pixels of differently oriented patterns before pruning

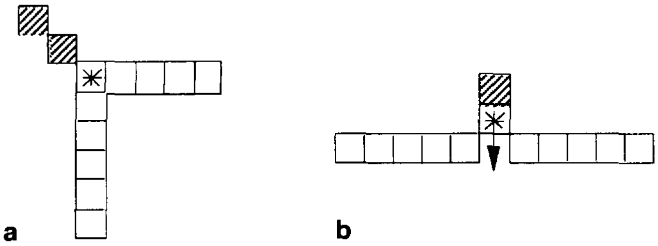

Figure 10. After pruning the dashed pixels, the starred pixel may not be left in the skeleton (a) or may be shifted (b)

Correspondingly, pruning should be accomplished only if the jut of the protrusion mapped in the skeleton peripheral branch is below a given threshold.

In this paper we dynamically compute the difference in jut between the protrusion mapped into a peripheral skeleton branch, and the protrusion mapped in what would remain after pruning (a portion of) that branch. The aim is to allow pruning from the end point p up to the most internal pixel q (belonging to the same peripheral branch as p), in correspondence with which the difference in jut is not above the threshold. The difference $J_p(q)$ is computed by:

$$J_p(q) = \sqrt{p} - [\sqrt{q} - d_e(p, q)] \qquad (6)$$

Pruning is accomplished by unmarking the pixels on the skeleton branch. The list L is inspected, to identify the end points. Each pixel p with flag E is accessed, and tracing is started from it. Tracing is continued until $J_p(q)$ is not above the threshold, and q is a normal point. As soon as a pixel r is identified, such that $J_p(r)$ has reached the threshold, the skeleton branch is pruned from p to r, r excluded. When r is a branch point, or is a neighbour of a branch point, keeping it could leave an undesired very short branch in the skeleton. Then, $C_8(r)$ is computed, and r is unmarked if $C_8(r) = 1$. The same is done for the neighbours of r (see Figure 10a). If $C_8(r) > 1$, the possibility of shifting the marker from r to one of its neighbours n_{i+1}, i even, is checked by using the operation employed during the beautifying step. This improves skeleton shape (see Figure 10b).

The behaviour of the pruning criterion with respect to pattern rotation can be appreciated in Figure 11. The same pruning threshold (equal to 1.4 in this example) produces skeletons having almost the same shape, whichever pattern orientation. As in the case of Figure 1d, stars indicate pattern pixels not recovered by the reverse Euclidean distance transformation. The negligible significance of the pruned branches is witnessed by the marginal difference between the input patterns and those recovered by their skeletons. Invariance under scale change is illustrated in Figure 12, where a pruning threshold equal to 1.4 has also been used. In the case of scaled patterns, the skeleton of the low

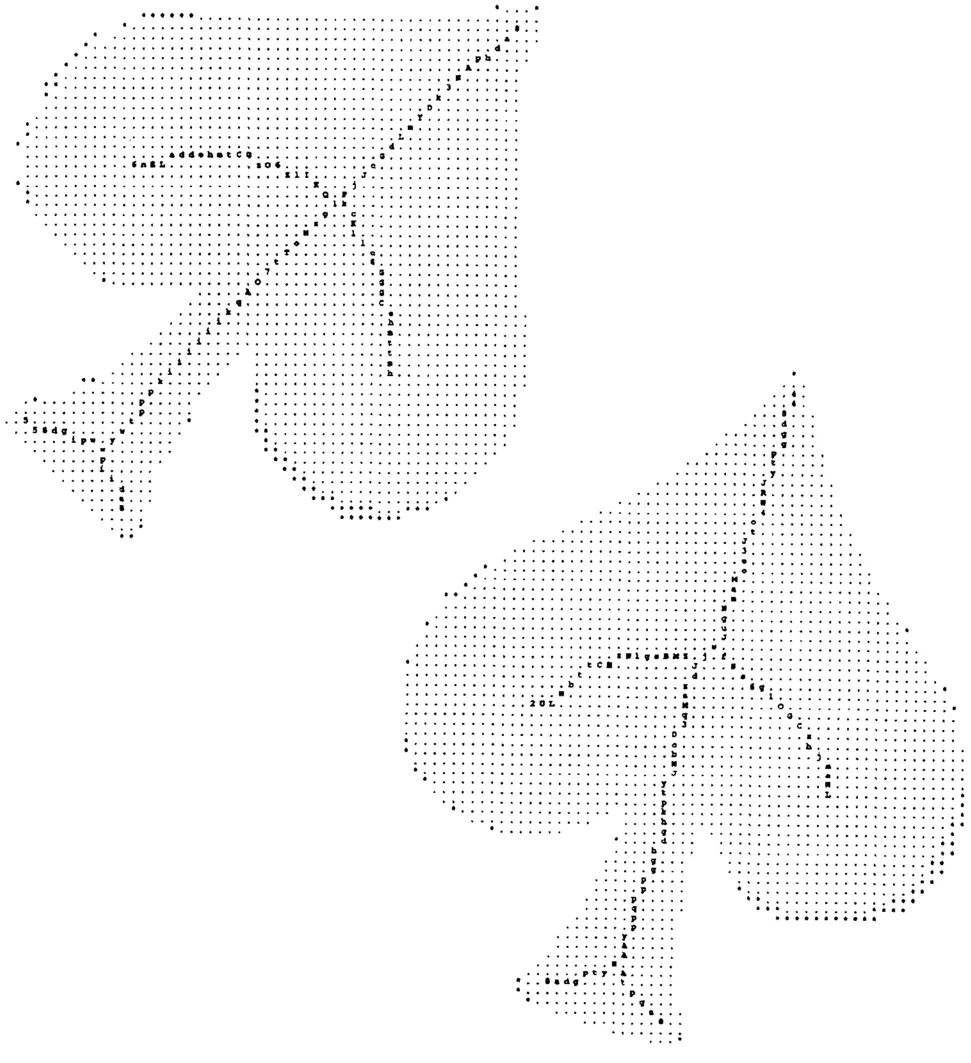

Figure 11. Skeletons of the differently oriented patterns of Figure 9, after pruning

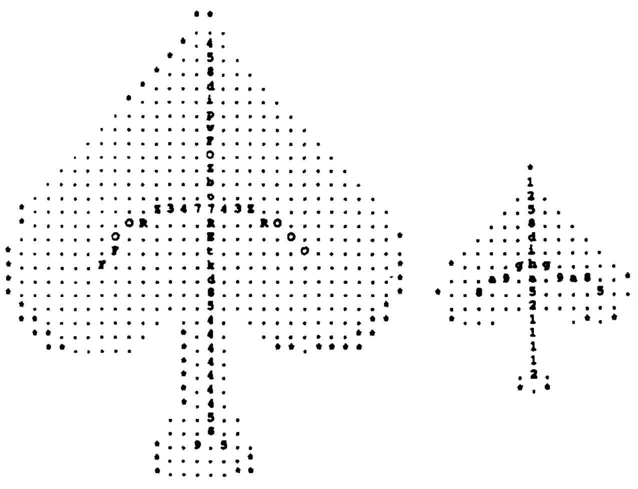

Figure 12. Skeletons of scaled patterns

resolution pattern may have a smaller number of peripheral branches, but still generally possesses the same main structure as the skeleton of the high resolution pattern. Finally, pruning is effective also when noise biasing the pattern is not simply due to the digitization process. In this case, the pruning threshold has to be assigned a value adequate for the removal of branches corresponding to pattern protrusions caused by noise. This value is usually larger than 1.4, which is adequate for the removal of mere digitization noise. As an example refer to Figure 13, where the threshold has been set to 2.

THE ALGORITHM

Cleaning of the input pattern is a standard process which can be accomplished by employing well known techniques[14], or by directly updating the pattern while computing the EDM. In this respect, one could refer to a previous paper[12], where cleaning is accomplished while computing the chessboard distance map. The EDM can be obtained by a simple three raster scan algorithm[6], or by a more sophisticated and less time-consuming contour processing based method[7].

A forward raster scan inspection of the EDM is done, during which each pixel p with a positive label is marked as a cmd if the label of each horizontal/vertical neighbour is less than Hlut(p), and the label of each diagonal neighbour is less than Dlut(p). Otherwise, p is checked against Conditions (1)–(3), and is possibly marked as a saddle point. Once p is marked, the raster scan is temporarily interrupted. The neighbours of p are visited and path growing is started, coherently with the direction of the maximal gradient. Path growing continues while neighbours with increasing labels are found. Path growing towards regions already inspected by the raster scan terminates as soon as the neighbour with the maximal gradient is marked. Before resuming the raster scan, the second path, possibly originating from p, is grown up. When the raster is resumed, pixels already marked can be encountered, since paths also proceed towards directions not yet visited by the scan. For these pixels no check has to be done. The inspection is continued until the whole EDM has been examined.

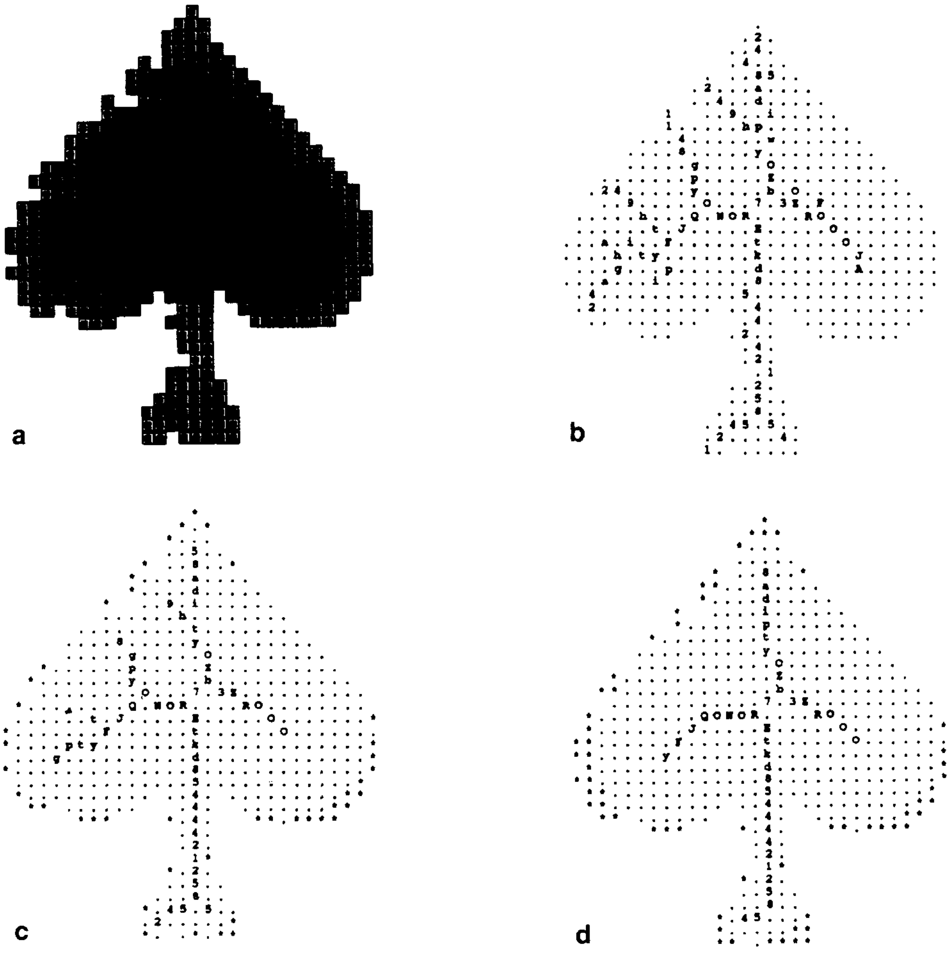

Figure 13. (a) A pattern whose contour is biased by noise; (b) skeleton before pruning; (c) skeleton obtained after pruning with threshold 1.4; (d) and with threshold 2

The remaining steps of the process do not require lengthy raster scan inspections of the array where the EDM is recorded, since the skeletal pixels are directly accessed by employing the list L of their coordinates.

The steps of the algorithm, performed on the EDM, can be summarized as follows:

1. *Marking the skeletal pixels*. The intrinsic skeletal pixels are the centres-of-maximal-discs detected by using the tables Hlut and Dlut, and the saddle points, detected using Conditions (1)–(3). The induced skeletal pixels are the neighbours of skeletal pixels maximizing the gradient.
2. *Spurious-hole filling*. Pixels are marked as skeletal pixels if all their horizontal/vertical neighbours are skeletal pixels.
3. *Final thinning*. The marker is removed from any skeletal pixel satisfying Conditions (4) and (5), respectively during the first and the second inspection of the list L.
4. *Beautifying*. The marker is shifted from a skeletal pixel to one of its unmarked neighbours.
5. *Pruning*. The pixels of a skeleton branch are unmarked until a pixel is met, such that the jut evaluated by expression (6) reaches a given threshold.

We do not include the proof that the proposed skeletonizing algorithm is topologically correct, as it is lengthy and does not significantly differ from the proof given elsewhere[12], relative to the case of the skeleton found on the chessboard distance map. The correctness of a skeletonizing algorithm driven by a distance map is guaranteed by the detection of all the centres-of-maximal-discs and all the saddle points. In fact, these pixels include all the necessary seeds from which connecting paths are grown up, and the structure of the distance map guarantees correct construction and termination of the paths.

One further example showing the performance of the algorithm is illustrated in Figure 14, where stars still denote pixels of the input pattern which are not recovered by applying the reverse Euclidean distance transformation to the skeleton. We may note that the skeleton is well centred within the pattern, and that good recoverability is achieved.

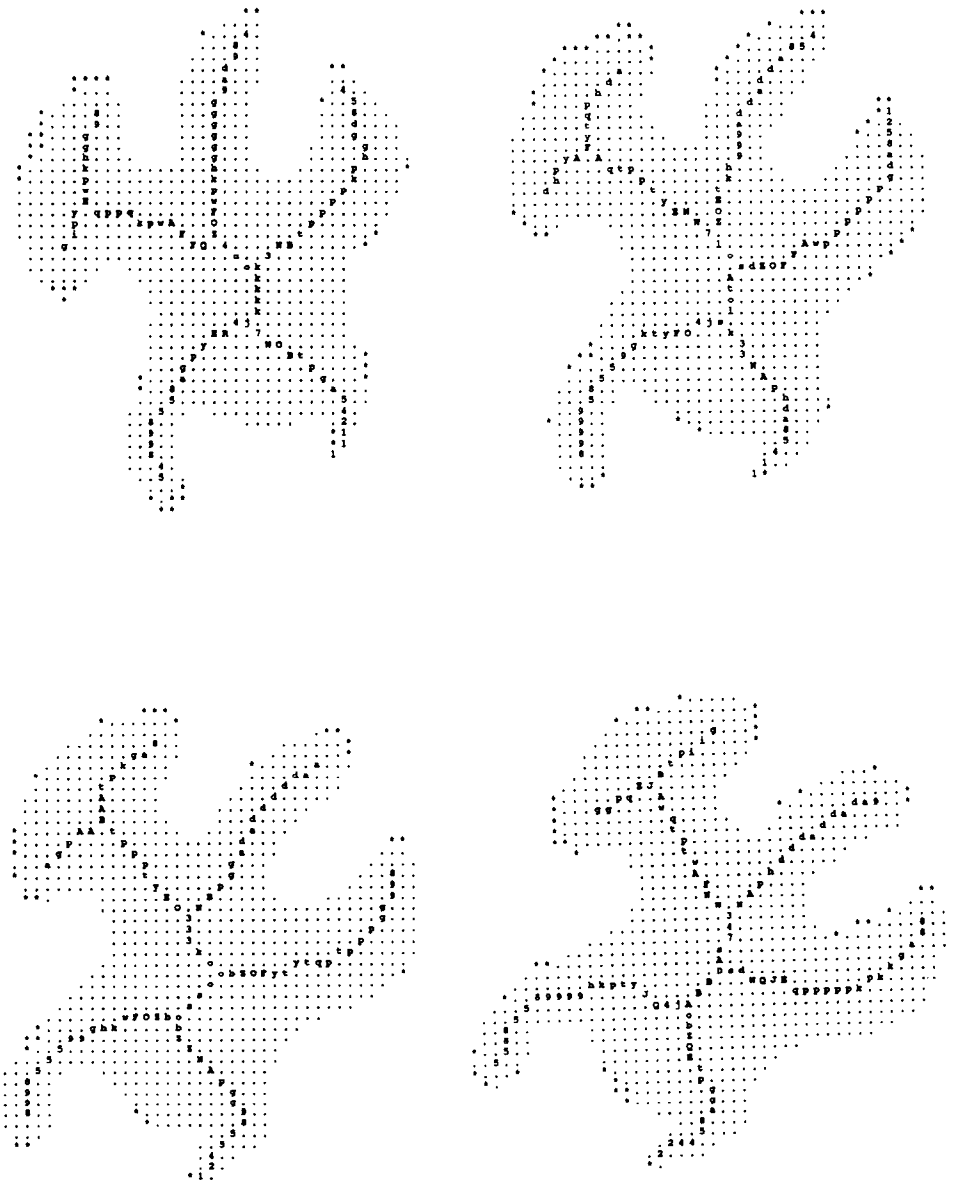

Figure 14. Stability of the skeleton under pattern rotation

CONCLUSION

We have described a skeletonization algorithm driven by the Euclidean distance map. The main feature of the algorithm is the inclusion in the skeleton of all the centres-of-maximal-discs, except for those whose presence prevents unit width of the skeleton. This feature characterizes our algorithm with respect to other algorithms designed to obtain a Euclidean skeleton[1-3], and is relevant when using the skeleton in shape analysis tasks, since it guarantees pattern recoverability. Moreover, detection of the centres of the maximal discs allows us to identify directly the pixels which will constitute the end points in the resulting skeleton.

ACKNOWLEDGEMENTS

The help of Mr Salvatore Piantedosi in preparing the illustrations is gratefully acknowledged.

REFERENCES

1 **Klein, F and Kubler, O** 'Euclidean distance transformations and model-guided image interpretation', *Patt. Recogn. Lett.*, Vol 5 (1987) pp 19–29

2 **Kruse, B** 'An exact sequential Euclidean distance algorithm with application to skeletonizing', *Proc. 7th Scand. Conf. on Image Analysis*, Aalborg, Denmark (1991) pp 982–992

3 **Arcelli, C and Sanniti di Baja, G** 'Ridge points in Euclidean distance maps', *Patt. Recogn. Lett.*, Vol 13 (1992) pp 237–243

4 **Danielsson, P E** 'Euclidean distance mapping', *Computer. Graph. Image Process.*, Vol 14 (1980) pp 227–248

5 **Forchhammer, S** 'Euclidean distances from chamfer distances for limited distances', *Proc. 6th Scand. Conf. on Image Analysis*, Oulu, Finland (1989) pp 393–400

6 **Ragnemalm, I** *Generation of Euclidean distance maps*, Thesis No 206 Department of Electrical Engineering, Linkoping University, Sweden (1990)

7 **Ragnemalm, I** 'Contour processing distance transforms', in **V Cantoni, L P Cordella, S Levialdi and G Sanniti di Baja (eds)**, *Progress in Image Analysis and Processing*, World Scientific, Singapore (1990) pp 204–212

8 **Borgefors, G, Ragnemalm, I and Sanniti di Baja, G** 'The Euclidean distance transform: finding the local maxima and reconstructing the shape', *Proc. 7th Scand. Conf. on Image Analysis*, Aalborg, Denmark (1991) pp 974–981

9 **Rutovitz, D** 'Pattern recognition', *J. Royal Statist. Soc.*, Vol 129 (1966) pp 504–530

10 **Yokoi, S, Toriwaki, J I and Fukumura, R** 'An analysis of topological properties of digitized binary pictures using local features', *Computer. Graph. Image Process.*, Vol 4 (1975) pp 63–73

11 **Rosenfeld, A** 'Arcs and curves in digital pictures' *J. ACM*, Vol 20 (1973) pp 81–87

12 **Arcelli, C and Sanniti di Baja, G** 'A width-independent fast thinning algorithm', *IEEE Trans. PAMI*, Vol 7 (1985) pp 463–474

13 **Arcelli, C and Sanniti di Baja, G** 'A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform', *IEEE Trans. PAMI*, Vol 11 (1989) pp 411–414

14 **Rosenfeld, A and Kak, A C** *Digital Picture Processing*, Academic Press, New York (1982)