COMPUTING AND MANIPULATING SKELETONS OF PLANAR PATTERNS FOR SHAPE REPRESENTATION AND DESCRIPTION

Gabriella Sanniti di Baja* and Edouard Thiel+

*Istituto di Cibernetica, CNR, Via Toiano 6, 80072 Arco Felice, Naples, Italy
+Equipe TIMC-IMAG, IAB-Domaine de la Merci, 38706 La Tronche cedex, France

Abstract: The path-based distance skeleton of a digital pattern, perceived as the superposition of elongated regions, is computed. The skeleton is then used to analyse the shape of the pattern. To this purpose, it is suitably decomposed and the resulting components are used to represent and describe the corresponding planar regions.

Keywords: Distance map, skeleton, polygonal approximation, shape decomposition, shape description.

I. INTRODUCTION

Shape representation and shape description are important steps in pattern recognition. For binary digital images, i.e., for images where two values are enough to characterise the pattern and its complement, commonly employed pattern representations are based on contour extraction and skeletonization. In fact, both the contour and the (labelled) skeleton of a planar pattern have a small cardinality, can be stored in vector form and allow pattern recovery. The skeleton is particularly convenient when working with elongated patterns, as its shape strongly resembles the shape of the represented pattern. In turn, the labels attached to the skeletal pixels can be used to measure local pattern thickness, and allow one to use the skeleton to analyse patterns whose thickness is not constant.

The structural approach to shape description can be conveniently followed to analyse complex patterns, perceived as the superposition of simpler regions. The pattern (or a suitable representation of the pattern) is first decomposed into the constituting components, whose description can be easily achieved. Then, the description of the pattern is provided in terms of the description of the individual components and of their interrelations.

The skeleton is a good tool to analyse, via the structural approach, the shape of patterns perceived as the superposition of elongated regions. In fact, an immediate correspondence exists between the skeleton branches and the elongated regions constituting the pattern. Thus, decomposing the skeleton into its constituting branches is equivalent to decomposing the pattern into the corresponding elongated regions. If the shape of the obtained regions is simple enough, the decomposition process terminates. Otherwise, the skeleton branches can be furthermore decomposed, in such a way that the regions corresponding to the so obtained skeleton components are simple.

This paper is mainly divided into two parts, respectively dealing with skeleton extraction and skeleton decomposition. The skeleton is extracted from the distance map of the pattern by means of a skeletonization algorithm equally running, whichever distance is used to build the distance map, among four well known path-based distance functions. The skeleton is then interpreted as a curve in the 3D space (where the co-ordinates of any pixel are the planar co-ordinates and the distance label); there, the skeleton is suitably decomposed, so as to produce a decomposition of the pattern into simple regions.

II. BASIC NOTIONS

Let F={1} and B={0} be the two sets constituting the pattern and its complement in a binary picture P digitised on the square grid. We suppose that no pixel of F is placed on the first two and the last two rows and columns of P, so that 5×5 operations can be safely applied on every pixel of F. To avoid topological paradoxes, we select the 8-metric for F and the 4-metric for B. Without loss of generality, we suppose F constituted by a single 8-connected component, while no assumption is done on the number of 4-connected components constituting B.

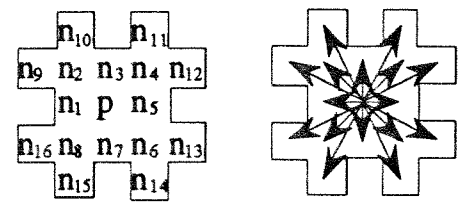


Fig. 1. The set N(p) and the directions between p and the $n_i(p)$.

The neighbourhood N(p) of a pixel p includes the 16 pixels surrounding p, as shown in Fig. 1. The $n_i(p)$ are also called horizontal/vertical neighbours, for i={1, 3, 5, 7}; diagonal neighbours, for i={2, 4, 6, 8}; and knight neighbours, for i={9, 10,..., 16}. The neighbours of $n_i(p)$ are indicated as $n_k(n_i)$ (k=1, 2,...,16). In the following, p will be used to indicate both the pixel and its associated label.

The distance map DM of F with respect to B is a replica of F, where the pixels are labelled with their distance from B, computed according to a given distance function. On the discrete plane, path-based distances are commonly used, where the distance between two pixels is defined as the length of a shortest path linking them. The degree of approximation to the Euclidean distance depends on the number of different unit moves permitted along the path, and on the weights used to measure them. The city-block distance d_1 (one unit move, unitary weight) and the chessboard distance $d_{1,1}$ (two unit moves, both with unitary weight) [1] are a natural choice on the square grid, but roughly approximate the Euclidean distance. Better approximations are obtained by using weighted distance functions allowing two (or three) differently weighted unit moves. Instances of well known weighted distance functions are the two-weight distance $d_{3,4}$ and the three-weight distance $d_{5,7,11}$ [2].

Any pixel of the DM can be interpreted as the centre of a disc fitting F: the label of the pixel is related to the radius length; the disc is a polygon with a number of sides depending on the adopted distance function (d_1 and $d_{1,1}$ provide a 4-side polygon, while $d_{3,4}$ and $d_{5,7,11}$ provide discs having eight and sixteen sides, respectively). A pixel p of the DM is centre of a maximal disc if the associated disc is maximal, i.e. is not included by a single disc centred on any $n_i(p)$. A maximal disc can be included in the union of two or more other maximal discs.

The skeleton S of F is a subset of F characterised by the following properties: 1) S has the same number of 8-connected components as F, and each component of S has the same number of 4-connected holes as the corresponding component of F. 2) S is centred within F. 3) S is the unit-wide union of simple 8-arcs and 8-curves. 4) The pixels of S are labelled with their distance from B. 5) S includes almost all the centres of the maximal discs of F (complete inclusion is not compatible with fulfilment of property 3).

The end points, the normal points and the branch points of the skeleton are the pixels of S having in S one, two and more than two neighbours $n_i(p)$ (i=1,2,..,8), respectively.

A skeleton branch is an arc of S, consisting of normal points except for the extremes which are end points or branch points. It can be interpreted as the spine of the region obtained from it by means of a sweeping disc. The disc sweeps out the shape by moving along the spine and changing size as it moves. When the spine is a straight line segment and the sweeping disc change its size in a monotonic and linear way, the region is said to be a simple region. A simple region is characterised by linearly changing width and orientation.

III. SKELETON EXTRACTION

The skeletonization algorithm includes three main phases: computation of the distance map DM, according to any distance among d_1, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$; identification of the set SK of the skeletal pixels; extraction of the skeleton S from SK.

The three phases of the process are individually described in the following.

A. Distance Map Computation

The distance map can be interpreted as the result of a propagation process. A wave front originating on the contour of F at a given instant of time, propagates towards the inside of F. Every pixel p of F receives distance information from more external pixels, which are at unit distance from it; in turn, the pixel p propagates distance information to more internal pixels, which are at unit distance from it.

The sequential algorithm [1] to compute the DM is suitably modified so as to be used whichever distance (among d_1, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$) is selected. In detail, the two following sequential local operations $f_1(p)$ and $f_2(p)$ are respectively applied to every p in F, during a forward and a backward raster inspection of P.

$$f_1(p)= \min_{i=\{1,3\}}\{\min [n_i(p)+w_1], \min_{i=\{2,4\}} [n_i(p)+w_2], \min_{i=\{9,10,11,12\}} [n_i(p)+w_3]\}$$

$$f_2(p)=\min_{i=\{5,7\}}\{p,\min_{i=\{6,8\}}[n_i(p)+w_1], \min_{i=\{13,14,15,16\}}[n_i(p)+w_2], \min [n_i(p)+w_3]\}$$

The values to be used for w_1, w_2, and w_3, so as to cause correct propagation of local distance information between p and the pixels $n_i(p)$ which are at unit distance from p according to the selected distance function, are given in Table 1, where ∞ stands for a sufficiently high value.

Table 1

	w_1	w_2	w_3
d_1-DM	1	∞	∞
$d_{1,1}$-DM	1	1	∞
$d_{3,4}$-DM	3	4	∞
$d_{5,7,11}$-DM	5	7	11

B. Identification of the Skeletal Pixels

The set SK includes the centres of the maximal discs, the

saddle points, the linking pixels and the pixels filling the "spurious holes". The centres of the maximal discs are the pixels responsible for pattern recovery; the saddle points and the linking pixels guarantee skeleton connectedness; the pixels filling the spurious holes guarantee that the complement of S has the same number of components as B. The centres of the maximal discs and the saddle points are identified and marked on the DM within one raster scan of P. Whenever a pixel is marked, the raster scan is temporarily interrupted and a path growing process is accomplished to identify and mark the linking pixels. The spurious holes are identified and filled in during a further scan of P. The skeletal pixels are marked in such a way to distinguish them from the remaining pixels in the DM without losing distance information (our choice is to change the sign of the label pertaining the pixels detected as skeletal pixels from positive to negative).

Marking the centres of the maximal discs

The centres of the maximal discs are identified by means of look-up tables recording, for any possible radius p, which are the minimal radii of the discs that, centred on every $n_i(p)$, would prevent the disc of p to be maximal [3].

Except for a few irregularities, the fields of every record in the look-up table can be filled in by following the rule shown in Table 2, where, h/v, d and k respectively refer to the horizontal/vertical, diagonal and knight neighbours of p, and the w_i (i=1,3) are as in Table 1.

Table 2

label	h/v	d	k
p	$p+w_1$	$p+w_2$	$p+w_3$

Only a few records pertaining the $d_{3,4}$-DM and $d_{5,7,11}$-DM cases, cannot be filled in by the rule of Table 2. These records are given in Table 3 and Table 4. Blank cells in these Tables correspond to fields where the filling rule correctly applies.

Table 3 ($d_{3,4}$-DM case)

label	h/v	d	k
3	4	6	
6	8	9	

A pixel p of the DM is marked as a centre of a maximal disc if all the $n_i(p)$ are labelled less than the values stored in the corresponding fields of the record associated to p.

Table 4 ($d_{5,7,11}$-DM case)

label	h/v	d	k
5	7	10	14
7	11		
10	14	15	20
14	18	20	
16		22	
18	22		28
20		26	30
21		27	
25	28	30	35
27		33	
29	33		
31		37	
32		38	
35	39	41	45
38		44	
39		45	
40	44		
42		48	
46		52	
49		55	
53		59	
60		66	

Marking the saddle points

Saddle points constitute a ridge between two subsets of the DM consisting of pixels with higher label. Most of the saddle points are also centres of maximal discs; the remaining ones can be identified by means of local conditions which reflect the definition of saddle point. In detail, a pixel p of the DM is marked as a saddle point if at least one of the following conditions is satisfied:

(1) The set $\{n_1(p), n_2(p), ..., n_8(p)\}$ includes two (4-connected) components of pixels having label smaller than p;
(2) The set $\{n_1(p), n_2(p), ..., n_8(p)\}$ includes two (8-connected) components of pixels having label larger than p;
(3) The pixels of one of the triples $(n_1(p), n_2(p), n_3(p))$, $(n_3(p), n_4(p), n_5(p))$, $(n_5(p), n_6(p), n_7(p))$, $(n_7(p), n_8(p), n_1(p))$ are all labelled as p;

Condition (3) is checked for all the pixels of the DM when $w_1=1$, only for the pixels labelled w_1, otherwise.

Marking the linking pixels

Path growing through the ascending gradient is attempted in correspondence of every pixel p, found to be centre of

maximal disc or saddle point. Since N(p) includes at most two 8-connected components of pixels labelled more than p, at most two ascending paths can be started from p. For each component of pixels labelled more than p, the neighbour $n_i(p)$ in correspondence of which the gradient assumes the maximal value is marked as the first linking pixel in that path. Then, the $n_k(n_i)$ are inspected to recursively identify the next linking pixel. Path growing proceeds, through the ascending gradient, as far as pixels providing a positive gradient are found. For any $n_i(p)$ labelled more than p, the gradient is computed as:

$$grad_i = [n_i(p) - p] / v_k$$

where the v_k assume the values indicated in Table 5.

Table 5

	v_1	v_2	v_3
d_1-DM	2	3	-1
$d_{1,1}$-DM	2	3	-1
$d_{3,4}$-DM	3	4	-1
$d_{5,7,11}$-DM	5	7	11

Negative values are assigned to the v_k corresponding to the unit moves which are not permitted in the selected DM. In the d_1-DM and $d_{1,1}$-DM cases, the values of the positive v_k are different from the values of the w_k used to compute the DM; this is done both to force the creation of 8-connected paths, and to avoid path thickening and fan creation. In the $d_{3,4}$-DM, at most a pair of neighbours $n_i(p)$ (for i=1, 2,..., 8), 4-adjacent to each other, equally maximise the gradient. When this is the case, only the $n_i(p)$ with i odd is accepted in the path, to avoid path thickening. In the $d_{5,7,11}$-DM case, three $n_i(p)$ could equally maximise the gradient: a horizontal/vertical neighbour, a diagonal neighbour and a knight neighbour, such that the directions between p and each of these three $n_i(p)$ be consecutive (with reference to Fig. 1, the three neighbours could be, for instance, n_1, n_2 and n_9). To have an 8-connected path and to avoid path thickening, we accept as linking pixels both the knight neighbour and the horizontal/vertical neighbour. If only the knight neighbour maximises the gradient, the horizontal/vertical neighbour and the diagonal neighbour on its sides are checked, and the pixel, out of these two, providing the largest gradient is also accepted in the path. Whenever the knight neighbour is accepted as a linking pixel, path growing continues only from it.

Marking the pixels inside the spurious holes

Spurious holes are created in the set SK when two paths of linking pixels, aligned along parallel directions, are so close to each other that pixels in a path are diagonal neighbours of pixels in the other path. In the d_1-DM, $d_{1,1}$-DM and $d_{3,4}$-DM, the only possibility to create spurious holes occurs in presence of pairs of diagonally oriented paths. These spurious holes are one-pixel in size. In the $d_{5,7,11}$-DM, directions constituted by runs of two horizontal/vertical pixels are created during path growing, when the knight neighbour is accepted as a linking pixel. If two paths of this type are very close to each other, one more possibility exists to create spurious holes. These spurious holes are two-pixel in size. Hole filling does not create excessive thickening of the set of the skeletal pixels, since only a few sparse spurious holes generally exist.

An unmarked pixel p is marked as a spurious hole, if any of the following conditions is satisfied, during a forward scan of P:

(1) all the horizontal/vertical $n_i(p)$ are marked as skeletal pixels;
(2) $n_5(p)$ is not marked, while $n_1(p)$, $n_3(p)$, $n_4(p)$, $n_6(p)$, $n_7(p)$ and $n_5(n_5)$ are marked;
(3) $n_7(p)$ is not marked, while $n_1(p)$, $n_3(p)$, $n_5(p)$, $n_6(p)$, $n_8(p)$ and $n_7(n_7)$ are marked;

C. Skeleton Extraction

The set SK is likely to be more than one pixel wide, so that suitable topology preserving removal operations are necessary to extract from it the desired skeleton. Reduction to unit width is here referred to as an "unmarking" process since we remove from any deletable pixel the marker, previously ascribed to distinguish it from the non-skeletal pixels of the DM. This process can be accomplished by directly accessing the skeletal pixels, provided that their co-ordinates have been stored during the scan necessary to mark the spurious holes. After reduction to unit width, two further steps are taken into account, respectively devoted to pruning and beautifying the skeleton. Also these steps can be accomplished without resorting to raster scan inspections of P.

Reduction to unit width

Reduction of SK to unit width is obtained by applying topology preserving unmarking operations, designed in such a way to avoid excessive shortening of the skeleton branches. To favour skeleton centrality within the pattern, unmarking is accomplished within two inspections of the set SK. The pixels which are 4-internal in the initial SK are prevented from unmarking during the first inspection. All the marked pixels undergo the unmarking process, during the second inspection.

A marked pixel p is unmarked during the first (the second) inspection of the set SK, if N(p) satisfies the following Condition (1) (Conditions (1) - (2)):

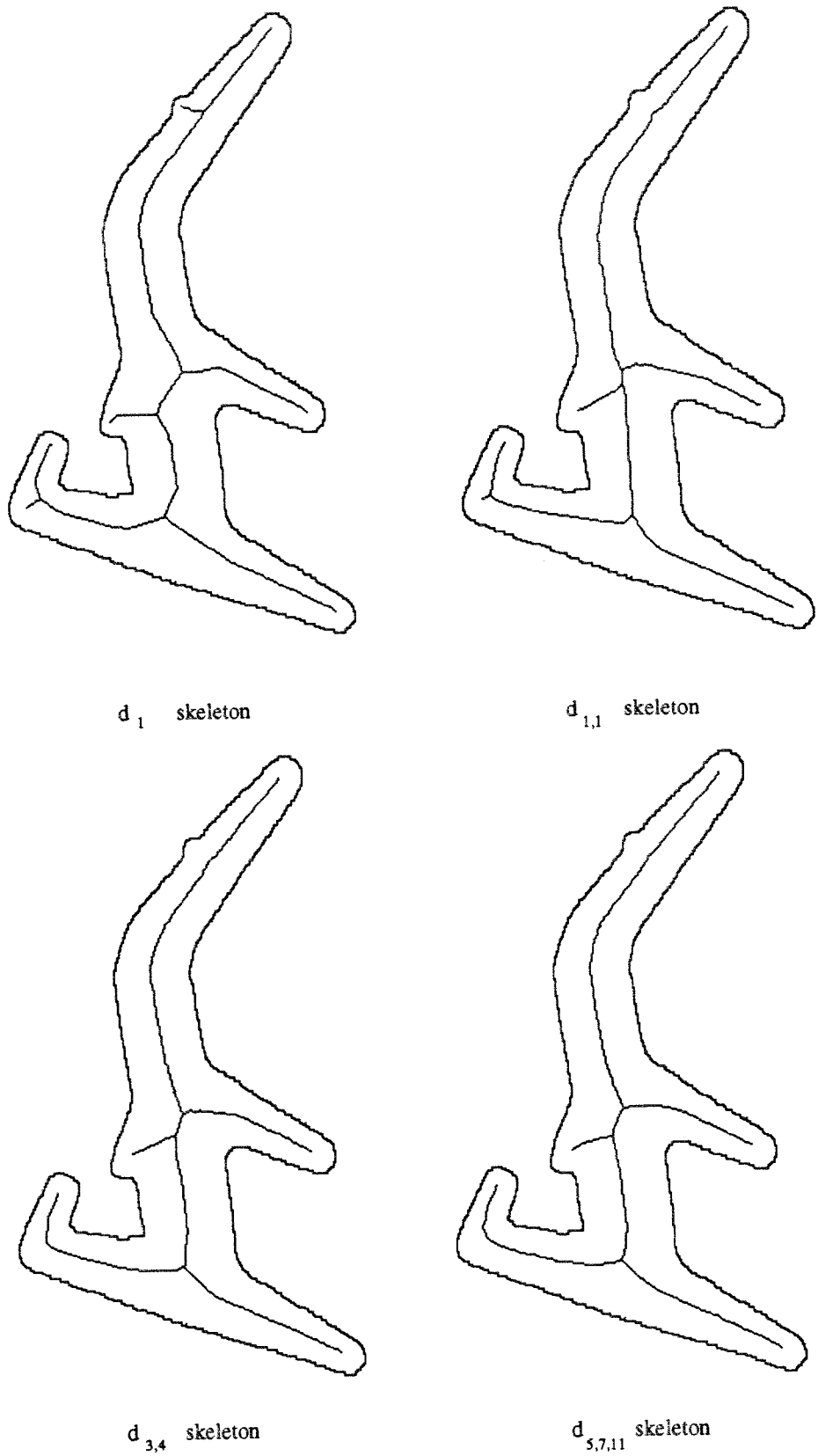


Fig.2. The resulting skeleton.

(1) At least a triple of neighbours $n_i(p)$, $n_{i+2}(p)$, $n_{i+5}(p)$ exists (i=1, 3, 5, 7, addition modulo 8), such that $n_i(p)$ and $n_{i+2}(p)$ are marked, while $n_{i+5}(p)$ is not marked;
(2) At least one horizontal/vertical neighbour of p is not marked;

Pruning

Pruning is accomplished to simplify the skeleton structure by deleting peripheral branches (i.e. branches delimited by end points) that do not correspond to pattern protrusions, significant in the problem domain. To keep under control the loss of information caused by branch deletion, a criterion based on the relevance of the protrusion associated with the skeleton branch is used. A branch can be safely pruned if a negligible difference exists between the two sets recovered by applying the reverse distance transformation to the skeleton and to the pruned skeleton, respectively. Such a difference can be evaluated in terms of labels and relative distance of the extremes of the (portion of) skeleton branch to be pruned. The difference can then be compared with a suitable threshold to decide about pruning [4].

Let K be the maximum number of peripheral rows and/or columns one accepts to lose in recovery due to branch deletion. Let p be the end point of a peripheral skeleton branch, and q be any other pixel on that skeleton branch. Pruning from p to q, q excluded, can be safely accomplished in the limits of the adopted tolerance if:

$$p - q + d \leq K \times w_1$$

where d is the adopted distance.

The most internal pixel up to which pruning can be accomplished is identified by checking the above condition on every pixel q, found while tracing the peripheral skeleton branch originating from p.

In [5], it has been shown that whichever is the selected distance among d_1, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$, the following expression can be used to compute d:

$$d = w_1 H + w_2 D + Flag \times \min(H,D)$$

where H and D are the horizontal/vertical and diagonal unit moves, along a minimal length 8-connected path linking the two pixels p and q, and it is Flag= 0 in the d_1-DM, $d_{1,1}$-DM, and $d_{3,4}$-DM cases, while Flag= -1 in the $d_{5,7,11}$-DM case.

Beautifying

To have a skeleton whose shape is more appealing, the zigzags due to noise on the contour of F and/or to the process done to obtain a unit wide skeleton should be straightened. The beautifying process is accomplished by shifting the marker from a pixel p having just two marked neighbours $n_i(p)$ and $n_{i+2}(p)$ (i=2, 4, 6, 8; addition modulo 8) to its neighbour $n_{i+1}(p)$. Shifting the marker from p to $n_{i+1}(p)$ may cause either of $n_i(p)$ and $n_{i+2}(p)$ to become deletable. After shifting, these neighbours of p are checked and possibly removed.

An example of the performance of the skeletonization algorithm is given in Fig. 2.

IV. SKELETON DECOMPOSITION

The decomposition algorithm requires two phases, respectively aimed at dividing the skeleton into a number of significant components, and merging contiguous components that satisfy suitable criteria.

A. Skeleton Decomposition: the Partition Step

The skeleton is interpreted as a concatenation of skeleton branches. This decomposition is equivalent to the decomposition of the pattern into a number of elongated regions. Generally, these regions are not simple: their local thickness may change in a non linear way or their spine may be not a straight line segment. Thus, the skeleton branches have to be furthermore decomposed in such a way that the obtained skeleton components be the spines of simple regions.

Spine identification

Every skeleton branch is interpreted as a 3D arc, the three co-ordinates of any skeletal pixel being the planar co-ordinates and the (normalised) label, and is polygonally approximated by using a split type algorithm [6]. The only skeletal pixels to be recorded after the polygonal approximation of the skeleton are the vertices, found wherever non linear changes (in curvature or label) occurred along the 3D arcs.

Pairs of successive vertices identify the spines of the simple regions constituting the decomposition of F. The value of the threshold θ used during the polygonal approximation, is fixed depending on the tolerance regarded as acceptable for the specific task. It should be rather small to favour a quite faithful representation of F by means of the found regions. In our experiments, the value θ =1.5 has been revealed as adequate.

Spine annihilation

Some simple regions may be (almost completely) overlapped by adjacent regions and should be disregarded while describing the shape of F. To this purpose, the corresponding spines should be identified and ignored. Among these spines, those delimited by branch points cannot be completely ignored, though having no region

representation power. These spines, play the role of linking elements among significant spines and are necessary to illustrate the spatial relations among the significant regions of the decomposition of F.

Short spines (i.e., spines with length less than 4, in pixel unit) are associated with generally disregardable regions. These short spines are ignored, if they are delimited by end points and/or branch points. In turn, isolated short spines (i.e., short spines which are embedded in between two non short spines) are annihilated. Annihilation of an isolated short spine s_i is done by moving its extremes (shared by s_i with the contiguous spines s_{i-1} and s_{i+1}) towards a common point in the 3D space. This is either the centre of gravity of s_i or the intersection between s_{i-1} and s_{i+1}, depending on the angle between s_{i-1} and s_{i+1}. See Fig. 3.

Annihilation causes s_i to disappear, and s_{i-1} and s_{i+1} to be modified and possibly merged into a unique spine. Merging is done, if the two modified spines result aligned, in the limits of the tolerance adopted when performing the polygonal approximation.

Also a non short spine may correspond to a simple region almost completely overlapped by the two adjacent regions, and be superfluous for pattern representation and description. This is the case, when the discs centred on the extremes of a spine partially overlap, so that the envelope of the two discs does not significantly differ from their union. The distance d between the extremes of the spine and the value of their labels, say l_1 and l_2, can be used to evaluate the overlapping degree. A spine is regarded as superfluous if the overlapping condition is satisfied, i.e. if it results $l_1 + l_2 \geq d$. A superfluous spine is annihilated if i) its extremes are normal points, and ii) the contiguous spines are not superfluous.

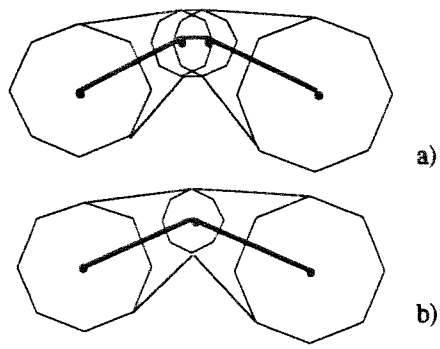


Fig. 3. After the intermediate spine is annihilated, a), a more significant decomposition is obtained, b).

Superfluous spines delimited by branch points play the role of linking elements. Let b_1, b_2,..., b_n be the skeleton branches sharing the branch point v_b, and let v_i (i=1, n) be the first vertex along b_i. The subset of b_i, delimited by v_b and v_i, is regarded as rid of region representation power if any of the following cases occurs:

i) a vertex v_j, located on b_j ($j \neq i$), exists such that the overlapping condition is satisfied by the discs centred on v_j and v_i.
ii) the overlapping condition is satisfied by the discs centred on v_b and v_i

As an example, refer to Fig. 4. There, the above cases i) and ii) respectively occur in Fig. 4a and Fig. 4c. Fig.4b and Fig. 4d show the simplified skeletons.

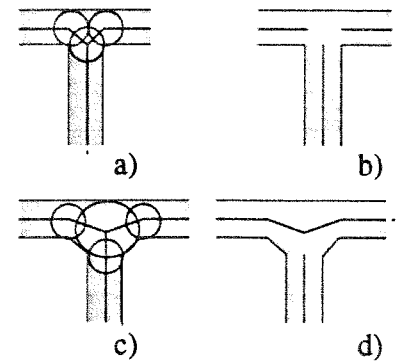


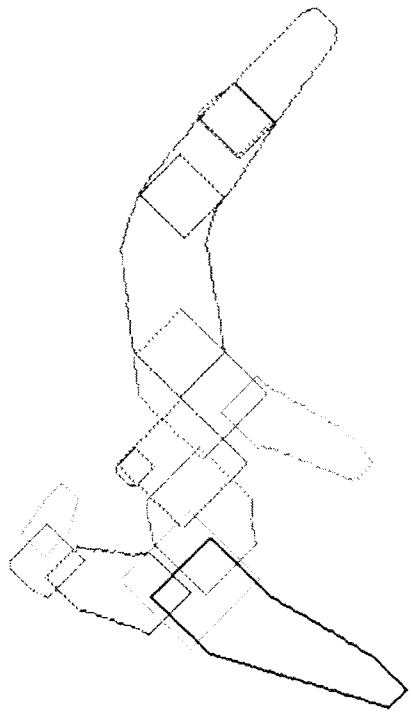
Fig. 4. Suppression of skeleton segments rid of region representation power.

B. Skeleton Decomposition: the Merging Step

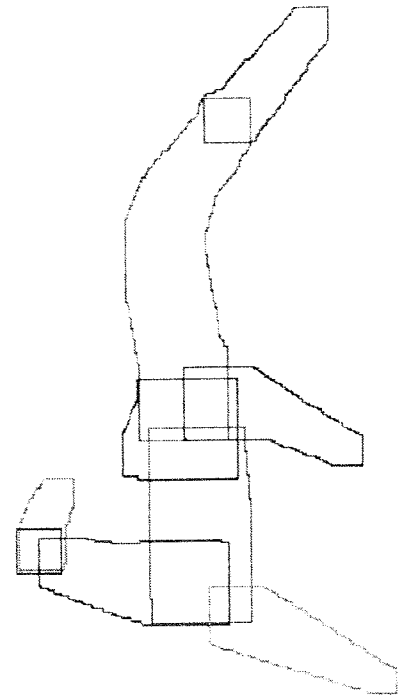
Although the spines remaining at this stage of the process all significantly contribute to pattern recovery, merging some of them could be useful to reduce the number of regions into which the pattern is interpreted as decomposed. Reducing the number of regions generally increases the stability of the decomposition and produces results more in accordance with human intuition. Simple regions having sufficiently similar width and orientation could be merged by concatenating the corresponding spines. A merged region, though no longer a simple region, can still be simply represented by the 3D co-ordinates of the vertices in the corresponding concatenation.

Two contiguous regions are merged if their spines, represented in the 3D space, are aligned in the limits of the adopted tolerance, τ. By employing different values for the merging threshold τ, different concatenations of spines are possible, which produce different pattern decompositions.

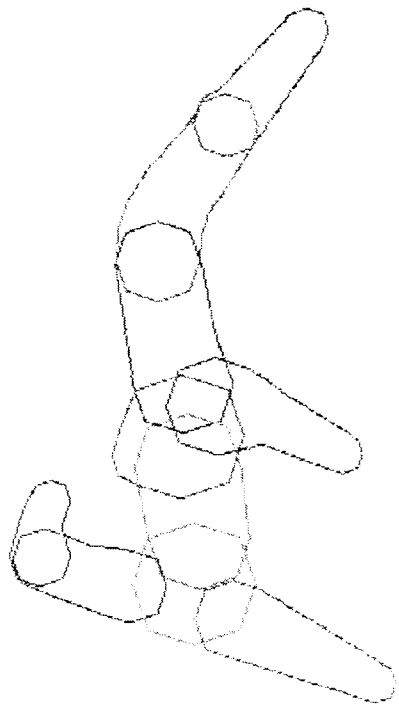
To avoid a propagation effect that could cause excessive spine merging, we proceed as follows. Each pair of successive spines, on the same skeleton branch, is examined. Let (v_{i-1}, v_i), and (v_i, v_{i+1}) be the vertices delimiting the



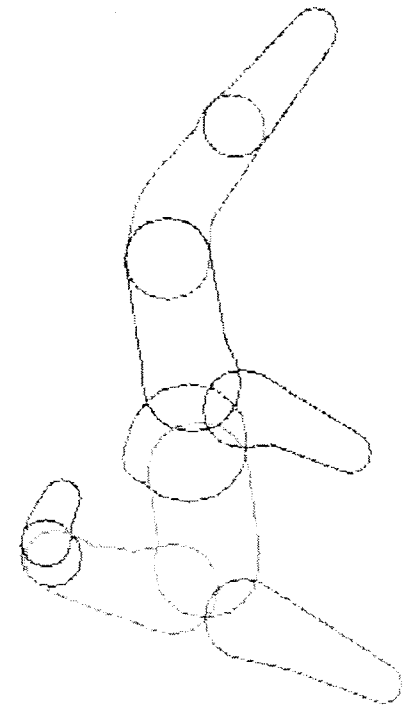
d_1 skeleton



$d_{1,1}$ skeleton



$d_{3,4}$ skeleton



$d_{5,7,11}$ skeleton

Fig. 5. Decompositions obtained starting from the d_1, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$ skeleton, for τ=0.15.

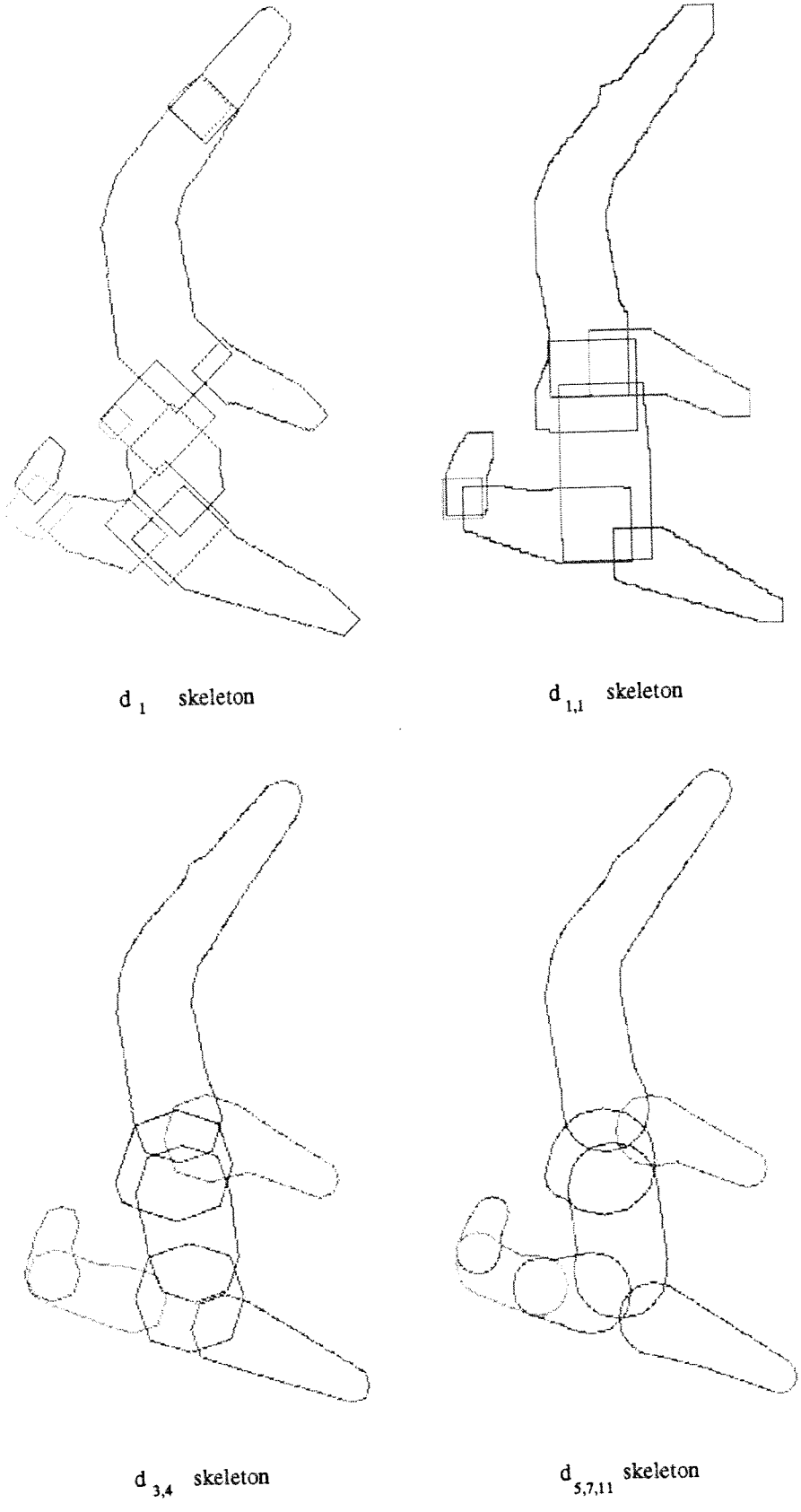


Fig. 5. Decompositions obtained starting from the d_1, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$ skeleton, for τ=0.15.

current pair of spines. Let L_i and D_i be respectively the Euclidean length of the segment joining v_{i-1} and v_{i+1}, and the 3D Euclidean distance of v_i from the segment. A flag $\mathcal{F}$, initially equal to "0", is set to "1" in correspondence of each vertex v_i, such that D_i/L_i is less than the merging threshold τ. Let v_1, v_2,..., v_n be a sequence of successive vertices in correspondence of which it is $\mathcal{F}=1$. Moreover, let v_0 and v_{n+1} be the two vertices immediately preceding v_1 and immediately following v_n.

-If n=1, the two spines (v_0, v_1) and (v_1, v_{n+1}) are merged.

-If n>1, for every i={1,2,...,n} the distance D_i of v_i from the straight line segment joining v_0 with v_{n+1} is divided by the length L of the segment. If for every vertex, it is $D_i/L < \tau$, all the spines are merged. Otherwise, the sequence v_2, v_3, ..., v_{n-1} is considered and for each of its vertices the merging ratio D_i/L is checked with reference to the straight line segment joining v_1 and v_n. The process is repeated until for the sequence v_k, v_{k+1}, ...,v_j (k=1+i, j=n-i, i>0) the merging condition is verified by all vertices. The corresponding spines are merged. Then, the merging condition is recursively checked on the two sub-sequences v_1, v_2,...,v_{k-1} and v_{j+1}, v_{j+2},...,v_n.

The vertices delimiting the set of the concatenation of spines are taken as the extremes of the resulting complex spine. Note that the remaining vertices still maintain their region representation power, since the region associated with a complex spine is the union of the simple regions associated with the merged spines.

The value of the merging threshold τ depends on the desired merging tolerance. Large values favour merging. An example is shown in Figs. 5,6 where two different values (τ=0.15 and τ=0.25) have been used for the merging threshold. The decompositions are obtained starting from the polygonal approximation of the skeleton, performed with θ=1.5.

The possibility of merging spines sharing a branch point as a common vertex could also be taken into account, in order the final pattern decomposition be not conditioned by the preliminary decomposition of the skeleton into its constituting branches.

V. CONCLUSION

We have illustrated a method to extract the path-based distance skeleton of a digital pattern and use it as a tool for shape analysis. The method is adequate for patterns that can be perceived as constituted by the union of elongated (ribbon-like) regions.

Using different distance functions provides differently structured distance maps and, accordingly, different skeletons. Thus, one can select the distance function more suited to the specific needs. In particular, skeletons computed according to the 2-weight or 3-weight distance functions are more suited to the applications due to their stability under pattern rotation. In turn, stability of the decomposition is favoured by the annihilation and merging steps, which reduce the skeleton decomposition components to the most significant ones. A relevant feature of the proposed decomposition method is the possibility to obtain decompositions at different resolution levels, by changing the threshold used during the merging step. The decompositions differ from each other for number and shape of the constituting regions, but all have the same global representative power.

The computational burden of the entire process is rather modest. In fact, only four raster scan inspections of the picture P are necessary to compute the skeleton, whichever is the thickness of the input pattern. Then, the decomposition process is accomplished working on a small amount of data (the skeletal pixels and, afterwards, the vertices of the polygonal approximation), which are stored in vector form.

ACKNOWLEDGEMENTS

This work has been partially supported by the "Galileo Program for the co-operation between France and Italy".

REFERENCES

1 A. Rosenfeld and J.L. Pfaltz, Distance functions on digital pictures, *Pattern Recognition*, **1**, pp. 33-61, 1968.

2 G. Borgefors, Distance transformation in digital images, *Comput. Vision Graphics Image Process.*, **34**, pp. 344-371, 1986.

3 G. Borgefors, Centres of maximal discs in the 5-7-11 distance transform, *Proc. 8th Scandinavian Conf. on Image Analysis*, pp. 105-111, 1993.

4 G. Sanniti di Baja, Well-shaped, stable and reversible skeletons from the (3,4)-distance transform, *Journal of Visual Communication and Image Representation*, **5**, pp. 107-115, 1994.

5 G. Sanniti di Baja, E. Thiel, Computing and comparing distance-driven skeletons", in *Aspects of Visual Form Processing*, C. Arcelli, et al. Eds., World Scientific, Singapore, pp. 475-486, 1994.

6 T. Pavlidis, Structural pattern recognition, Springer Verlag, New York, chapter 7, (1977).