COMPUTING AND COMPARING DISTANCE-DRIVEN SKELETONS

GABRIELLA SANNITI DI BAJA
Istituto di Cibernetica, CNR,
80072 Arco Felice, Naples, Italy

and

EDOUARD THIEL
Equipe TIMC-IMAG, CERMO BP 53
38041 Grenoble Cedex 9, France

ABSTRACT

A unified distance-driven algorithm is presented to extract the skeleton of a digital pattern from its distance map. The algorithm equally runs whichever distance is selected to compute the distance map, among four commonly used path-based distance functions. The resulting skeletons are compared on a large set of 512×512 input pictures, in terms of reversibility and computation time.

1. Introduction

Distance maps, computed according to different distance functions, can be adopted to guide pattern skeletonization. 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. City-block distance (one unit move, unitary weight) and chessboard distance (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 [2-5].

Skeletonization algorithms driven by the city-block distance d_1, the chessboard distance $d_{1,1}$, the two-weight distance $d_{3,4}$ and the three-weight distance $d_{5,7,11}$ can be respectively found in [6-9].

In this paper we provide a unique skeletonization algorithm equally running, whichever among the previous four distance functions is used to build the distance map DM of the pattern to be skeletonized. After computing the DM, the skeletal pixels (i.e. the centres of the maximal discs, the saddle points and the linking pixels) are identified and

marked on it, during one sequential raster scan inspection of the DM. An unmarking process is then accomplished on the DM, so as to reduce to unit width the set of the skeletal pixels; finally, an unmarking-and-shifting process is done to prune the skeleton and to improve its aesthetics, during which the marker is removed from a skeletal pixel or is shifted to some of its non marked neighbors. Unmarking and unmarking-and-shifting can be accomplished by directly accessing on the DM the skeletal pixels, whose coordinates can be recorded at a limited memory expense. The performance of the algorithm when using each of the four distances is evaluated by comparing the results obtained on a large set of 512×512 input pictures, in terms of reversibility and computation time.

2. The distance driven skeletonization algorithm

Let F={1} and B={0} be the two sets constituting a binary picture digitized on the square grid. We select the 8-metric for F and the 4-metric for B. We suppose F constituted by a single 8-connected component, while no assumption is done on the number of 4-connected components of B.

The distance map DM of F with respect to B is a replica of F, where the pixels are labeled with their distance from B. Each pixel of DM can be interpreted as the centre of a disc fitting F, the label of the pixel being related to the radius length. A pixel is centre of a maximal disc if the associated disc is maximal, i.e. is not included by any other single disc of F.

The skeleton S of F is a subset of F characterized 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 labeled 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).

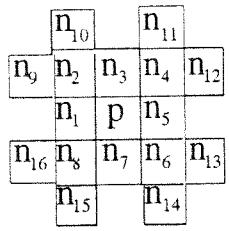

Figure 1. The set N(p).

Generally, the neighborhood N(p) of a pixel p includes the neighbors $n_i(p)$ of p, i.e. the pixels that can be reached with a unit move from p. Accordingly, N(p) should include from a minimum of 4 pixels, when d_1 is used, up to 16 pixels, when $d_{5,7,11}$ is

used. In this paper, N(p) includes all the 16 pixels surrounding p, as shown in Figure 1. The $n_i(p)$ are also called horizontal/vertical neighbors, for i={1, 3, 5, 7}; diagonal neighbors, for i={2, 4, 6, 8}; and knight neighbors, for i={9, 10,..., 16}.The neighbors 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.

2.1 Distance Map Computation

To compute the DM, the 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 the picture.

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

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

The values to be used for w_1, w_2, and w_3, so as to cause correct propagation from p to the pixels $n_i(p)$ which are neighbors of 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

Table 2

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

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

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

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

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	

2.2 Centre of maximal disc detection

With reference to the four distance functions adopted in this paper, centres of maximal discs can be detected by using local criteria involving suitable label comparisons [9,10], or by resorting to the use of look-up tables [11]. We prefer the second possibility, which makes the algorithm more homogeneous. Given a distance function, the corresponding look-up table has a number of entries, representing the labels occurring in the selected DM. Each entry p is associated a record containing as many fields as many are the different unit moves, which orderly identify the minimal label that the neighbors of p should have to prevent p to be centre of a maximal disc. A pixel p of the DM is marked as a centre of a maximal disc if its neighbors are labeled less than the values stored in the corresponding fields of the record associated to p.

The look-up table for the selected DM is built in during the distance map computation. Generally, the fields of a record are 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 neighbors of p. A few exceptions occur in the $d_{3,4}$-DM and $d_{5,7,11}$-DM case, and the records for which the filling rule of Table 2 does not apply have to be memorized . They are given in Table 3 and Table 4, respectively. Blank cells in these Tables correspond to fields where the filling rule correctly applies.

2.3 Saddle point detection

Saddle points constitute a ridge between two subsets of the DM, including pixels with higher label. Most of the saddle points are centres of maximal discs; the remaining ones are the pixels p for which 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 labeled as p;

To count the 4-connected and the 8-connected components in Conditions (1) and (2), the crossing number [12] and connectivity number [13] are respectively used. Condition (3) is checked for all the pixels of the DM when $w_1=1$, only for the pixels labeled w_1, otherwise.

2.4 Linking pixel detection

Generally, the set including the centres of the maximal discs and the saddle points is not connected. The linking pixels, necessary to gain skeleton connectedness, are identified by growing paths through the ascending gradient in the DM. Path growing is attempted in correspondence of every marked pixel p of the DM (centre of maximal disc

or saddle point). N(p) includes at most two 8-connected components of pixels labeled more than p, from which an ascending path can be started. For each component, the neighbor $n_i(p)$ in correspondence of which the gradient assumes the maximal value is marked as the first linking pixel in the path. Then, the $n_k(n_i)$ are inspected to recursively identify the next linking pixel. (Note that only one component of pixels labeled more than $n_i(p)$ is likely to exist.) Path growing proceeds, through the ascending gradient, as far as pixels providing a positive gradient are found. For any $n_i(p)$ labeled more than p, the gradient is: $grad_i = [n_i(p) - p] / v_k$, where the v_k assume the values indicated in Table 5.

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 neighbors $n_i(p)$ (for i=1, 2,..., 8), 4-adjacent to each other, equally maximize 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 pixels could equally maximize the gradient: a horizontal/vertical neighbor, its 4-adjacent diagonal neighbor, and the intermediate knight neighbor (e.g., 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 neighbor and the horizontal/vertical neighbor. If only the knight neighbor maximizes the gradient, the horizontal/vertical neighbor and diagonal neighbor 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 neighbor is accepted as a linking pixel, path growing continues only from it.

2.5 Spurious hole filling

Spurious loops are created in the set of the skeletal pixels when two paths, aligned along parallel directions, are so close to each other that pixels in a path are diagonal neighbors 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 loops occurs in presence of pairs of diagonally oriented paths. The enclosed 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 neighbor 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 loops. The enclosed spurious holes are two-pixel in size.

Spurious holes have to be identified and filled, i.e., their pixels have to be marked as skeletal pixels. 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 belongs to a spurious hole and is marked as skeletal pixel, if any of the following conditions is satisfied during a forward raster scan inspection of the DM:

(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;

2.6 Reduction to Unit Width

Reduction to unit width of the set of the skeletal pixels can be obtained by applying topology preserving removal operations, designed in such a way to avoid excessive shortening of the skeleton branches. 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.

To favor skeleton centrality within the figure, unmarking is accomplished within two inspections of the set of the skeletal pixels. The pixels which are 4-internal in the set of the skeletal pixels are preliminarily identified. They 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 of the skeletal pixels, if it satisfies the following Condition (1) (Conditions (1) and (2)):

(1) At least a triple of neighbors $n_i(p)$, $n_{i+2}(p)$, $n_{i+5}(p)$ 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 neighbor of p is not marked;

Condition (1) prevents both altering the connectedness, and shortening skeleton branches, while Condition (2) prevents creation of holes in the set of the skeletal pixels.

2 .7 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 figure 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 [8]. We generalize the expression used in [8], in such a way that it can be used whichever of the four distance functions is considered. Moreover, we do not limit pruning only to the branches which are peripheral in the initial skeleton.

Let H and D be the horizontal/vertical and diagonal unit moves, along a minimal length 8-connected path linking two pixels p and q. It is $H=(2d_{1,1}-d_1)$ and $D=(d_1-d_{1,1})$.

By taking into account that when $d_{5,7,11}$ is used any pair of (horizontal/vertical, diagonal) moves is substituted by a knight move, it is immediate to verify that the distance between p and q can be computed as follows:

$d_1 = H + 2D$ $\qquad$ $d_{1,1} = H + D$

$d_{3,4} = 3H + 4D$ $\qquad$ $d_{5,7,11} = 5H + 7D + (11\text{-}7\text{-}5) \times \min(H,D)$

A unique expression can be used to compute any of the four above distances:

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

where 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.

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 and 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:

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

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.

When a branch point is transformed into an end point, due to deletion of all the peripheral branches sharing it, a new peripheral branch is originated. This branch can be pruned without causing a summation effect in the loss of figure recovery. To this purpose, the information relative to the starting point(s) of the branch(es) is propagated through the branch(es) while performing pruning. In this way, the relevance of the entire protrusion mapped in the union of the current peripheral skeleton branch with the neighboring, already pruned, skeleton branches can be evaluated.

2.8 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 neighbors $n_i(p)$ and $n_{i+2}(p)$ (i=2, 4, 6, 8; addition modulo 8) to its neighbor $n_{i+1}(p)$. 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. These neighbors of p are checked and possibly removed.

3. Discussion and Conclusion

In this paper, we have introduced a unified scheme to extract the labeled skeleton from the distance map, computed according to four commonly used path-based distance

functions. The algorithm can be extended to treat DM computed by using other path-based distance functions, where the weights are different from the ones we have selected. 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. Moreover, the comparison among the different skeletons associated to the same figure by different distance functions is facilitated.

The skeletonization algorithm can be summarized as follows:

Step 1. Compute the distance map DM, and build the look-up table. (Two raster scans)

Step 2. Mark on the DM the skeletal pixels, i.e., the centres of the maximal discs, the saddle points and the linking pixels. (One raster scan interleaved with a path growing process)

Step 3. Mark on the DM the pixels inside the spurious loops of the set of the skeletal pixels and record the coordinates of all the marked pixels. (One raster scan)

Step 4. Unmark on the DM suitable skeletal pixels, so as to obtain the unit wide skeleton (The skeletal pixels are directly accessed)

Step 5. Unmark on the DM suitable skeletal pixels, so as to perform pruning and beautifying (The skeleton is traced starting from the end points)

At the end of the process, the skeleton is the set of pixels marked on the DM.

The computational cost of the algorithm is modest since only four raster scan inspections are necessary whichever is the thickness of the pattern to be skeletonized. However, the computation time depends on the ratio between the pixels of F and B, and on the adopted distance function.

The algorithm has been checked on a large number of patterns in 512×512 binary pictures. A set of 20 differently structured sample patterns is shown in Figure 2. From top left to bottom right the set includes: elongated rubber joints for car windows, rather free from digitization noise (p01-p07); complex clusters of differently sized blobs mutually overlapping (p08-p11); artificially drawn patterns characterized by remarkable thickness and significant contour curvature variations (p12-p15); and thin, but intertwined in a rather complex way, wood fibres in paper pulp (p16-p20).

Some results, pertaining the previous sample patterns are shown in Table 6. The last row refers to the mean value computed on the sample set. The second column (F/B) gives the percentage ratio between the number of pixels in the foreground F and in the background B. The successive 16 columns are grouped in quadruplets, respectively pertaining the d_1-DM, $d_{1,1}$-DM, $d_{3,4}$-DM, and $d_{5,7,11}$-DM cases. For each quadruplet, the first column (F/S) gives the ratio between the pixels of the foreground to the pixels of the skeleton and indicates the compression ratio measured in number of pixels; the second column (t) shows the computation time measured in seconds on a SUN Sparcstation 2; the third column (bp) and the fourth column (ap) indicate the percentage ratio to the pixels of F of the pixels which are not recovered starting from the skeleton before and after the

Table 6

		d_1				$d_{1,1}$				$d_{3,4}$				$d_{5,7,11}$			
	F/B	F/S	t	bp	ap	F/S	t	bp	ap	F/S	t	bp	ap	F/S	t	bp	ap
p01	12.13	41.72	6.46	0.52	1.09	41.34	6.28	1.31	2.30	45.35	6.17	0.24	1.44	45.35	6.92	0.15	1.13
p02	9.70	39.14	7.17	0.50	1.35	39.20	5.71	1.06	2.85	40.57	5.81	0.26	1.20	41.43	6.38	0.12	1.38
p03	8.90	28.12	6.17	0.86	1.85	27.46	5.55	1.26	2.71	28.68	5.65	0.45	1.43	29.00	6.25	0.18	1.36
p04	7.46	35.35	5.05	0.60	1.74	34.24	5.33	1.24	3.42	35.48	5.40	0.21	1.95	35.74	5.94	0.10	1.64
p05	20.54	43.18	8.36	0.61	1.06	44.32	8.14	0.85	2.16	44.43	7.54	0.33	1.15	44.32	8.69	0.25	0.82
p06	16.42	45.41	8.14	0.72	1.36	46.49	7.17	0.86	2.03	47.88	6.97	0.44	1.10	47.99	7.90	0.27	0.81
p07	11.65	43.37	6.15	0.53	1.42	43.37	6.17	1.07	2.82	45.03	6.07	0.41	1.48	45.85	6.84	0.21	1.38
p08	55.95	18.18	15.28	0.57	11.33	20.98	17.12	1.94	18.09	22.58	15.31	0.73	12.72	22.79	17.73	1.06	11.91
p09	58.89	16.05	15.84	0.58	10.25	17.84	17.69	1.85	15.11	18.71	15.73	0.68	10.34	18.79	18.58	0.97	9.78
p10	56.37	16.18	15.40	0.65	11.72	18.14	17.23	2.07	17.12	19.13	15.30	0.77	11.66	19.33	18.07	1.04	11.03
p11	59.25	13.24	17.69	0.91	10.76	14.36	21.08	2.37	14.93	14.56	16.01	1.04	10.33	14.73	19.13	1.25	10.08
p12	59.92	76.65	16.91	0.07	0.20	75.80	16.54	0.22	0.65	74.05	14.08	0.05	0.35	76.13	19.82	0.03	0.30
p13	63.86	73.58	17.50	0.00	0.67	73.90	17.57	0.06	0.96	78.04	14.71	0.00	1.21	105.54	18.30	0.00	1.40
p14	48.05	72.60	12.40	0.19	0.34	80.23	13.99	0.48	0.65	77.08	12.04	0.14	0.40	78.23	16.58	0.08	0.28
p15	30.24	47.67	11.86	0.33	0.77	42.32	10.23	0.76	1.31	43.65	9.22	0.27	0.70	43.65	11.04	0.12	0.49
p16	10.36	5.18	10.66	7.34	13.34	5.30	7.24	12.81	20.12	5.21	8.27	6.53	11.58	5.22	7.29	6.59	11.62
p17	10.21	4.29	6.45	8.93	15.21	4.41	6.44	16.27	25.33	4.30	6.99	7.48	13.17	4.31	7.45	7.58	13.22
p18	9.23	3.83	6.77	10.65	19.73	3.97	6.28	17.16	28.75	3.85	6.62	8.99	17.54	3.85	7.33	9.05	17.57
p19	10.32	9.58	5.81	3.57	6.38	9.78	6.08	5.58	10.48	9.65	6.29	2.42	5.60	9.69	7.01	2.59	5.37
p20	9.48	4.04	5.78	9.02	11.67	4.08	6.16	15.24	19.72	4.04	6.47	7.43	9.84	4.05	7.17	7.48	9.89
avg.	28.45	31.86	10.29	2.36	6.11	32.37	10.40	4.22	9.58	33.11	9.53	1.94	5.76	34.79	11.22	1.96	5.57

pruning step, respectively. The values in (bp) and (ap) give a measure of skeleton reversibility.

By looking at the mean values, we note that the computation time is a bit larger in the $d_{5,7,11}$-DM case, as expected due to the need of performing operations on a larger number of arguments. The larger time required in the $d_{1,1}$-DM case with respect to the remaining two cases, is mostly due to the generally larger thickness of the set of the skeletal pixels. (This can be appreciated by noting that the ratio bp is higher in the $d_{1,1}$-DM case, where a remarkable number of centres of maximal discs have to be canceled to reduce to unit width the set of the skeletal pixels.)

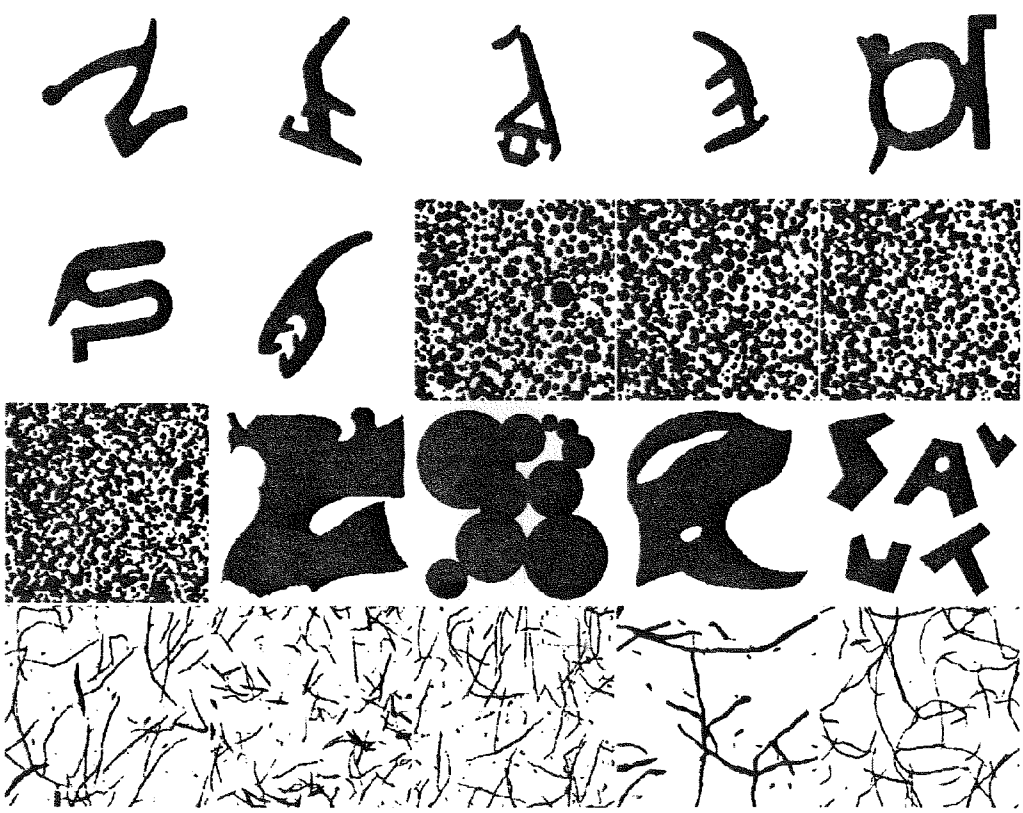

Figure 2. The set of sample patterns in 512× 512 binary pictures.

A reasonable compromise between computation time and skeleton reversibility is obtained in the $d_{3,4}$-DM case. This skeleton also shows a good stability under pattern rotation, translation or scaling. Thus, the $d_{3,4}$-skeleton can be generally employed for applications. The $d_{5,7,11}$-skeleton has to be preferred when a considerable accuracy is required, since its geometrical features very closely reflect pattern features; the slightly

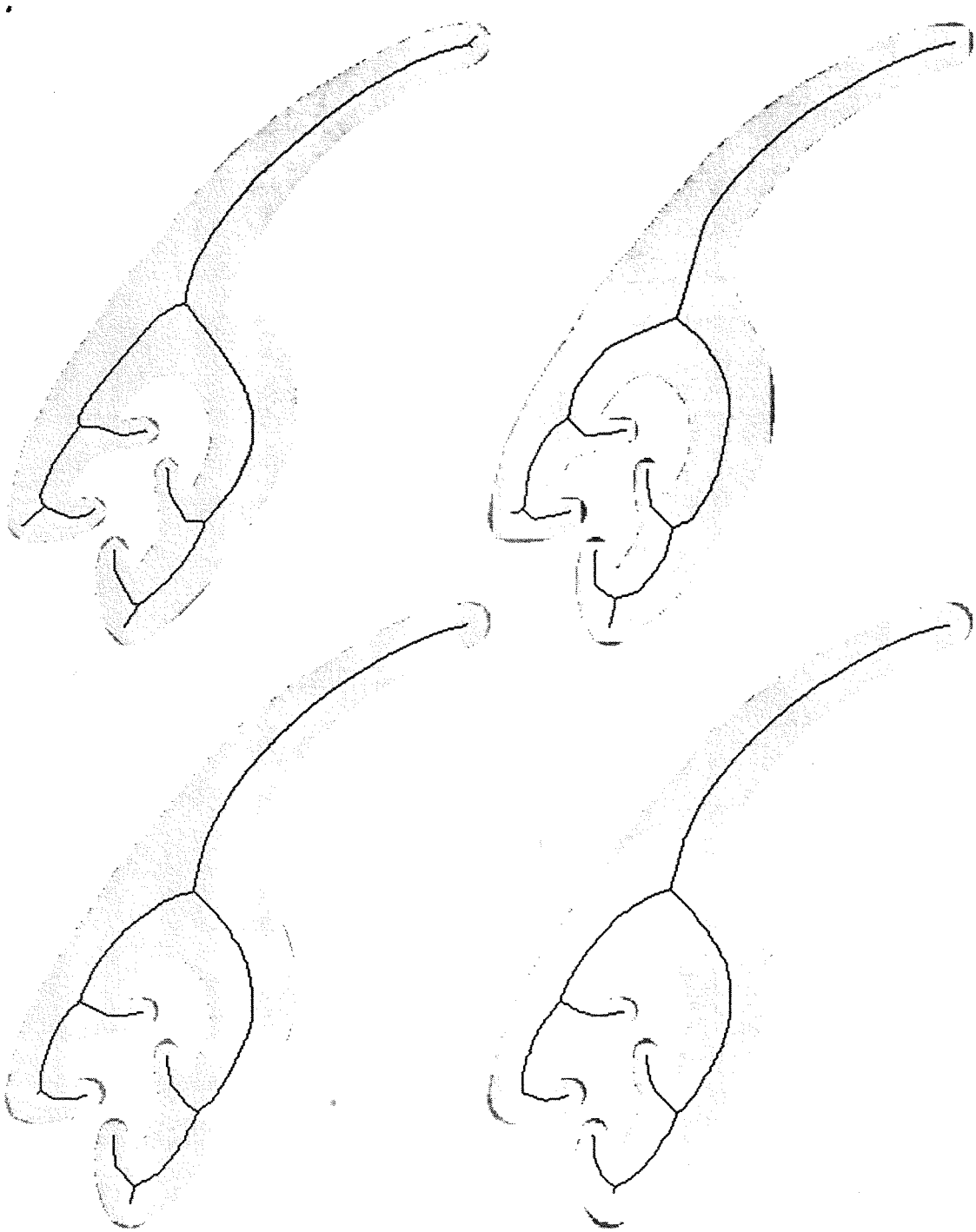

Figure 3. Skeletons driven by d_1, $d_{1,1}$, $d_{3,4}$, and $d_{5,7,11}$ (from top left to bottom right).

higher cost necessary for its computation is repaid by the higher stability degree under isometric transformations of the pattern.

In Figure 3, the performance of the skeletonization algorithm driven by the four different distance functions is shown on the test pattern p07. The four frames refer, from top left to bottom right, respectively to the d_1, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$ case. The threshold adopted during the pruning step is $2 \times w_1$. The skeleton (black pixels) is shown superimposed on the input pattern (grey pixels). To point out that the representative power of the skeleton is not remarkably biased by the unmarking process, the pixels of the input pattern which are not recovered starting from the pruned skeleton are shown in dark grey.

References

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

2 J. Hilditch and D. Rutovitz, Chromosome recognition, *Ann. New York Acad. Sci.*, 157, pp. 339-364, 1969.

3 G. Borgefors, Distance transformations in arbitrary dimensions, *Comput. Vision Graphics Image Process.*, 27, pp. 321-345, 1984.

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

5 E. Thiel and A.Montanvert, Chamfer masks: discrete distance functions, geometrical properties and optimization, *Proc. 11th Int. Conf. on Pattern Recognition*, The Hague, pp. 244-247, 1992.

6 C. Arcelli and G. Sanniti di Baja, A one-pass two-operations process to detect the skeletal pixels on the 4-distance transform, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11, pp. 411-414, 1989.

7 C. Arcelli and G. Sanniti di Baja, A width-independent fast thinning algorithm, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 7, pp. 463-474, 1985.

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

9 C. Arcelli and M. Frucci, Reversible skeletonization by (5,7,11)-erosion, in *Visual Form Analysis and Recognition*, C.Arcelli, L.P.Cordella and G. Sanniti di Baja, Eds., Plenum, New York, pp. 21-28, 1992.

10 C. Arcelli and G. Sanniti di Baja, Weighted distance transforms: a characterization, in *Image Analysis and Processing II*, V. Cantoni et al. Eds., Plenum Press, pp. 205-211, 1988.

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

12 D. Rutovitz, "Pattern recognition", *Journal of Royal Statist. Soc.*, 129, Series A, pp. 504-530, 1966.

13 S.Yokoi, J.I. Toriwaki, T. Fukumura, An analysis of topological properties of digitized binary pictures using local features, *Comput. Graph. Image Processing*, 4, pp. 63-73, 1975.