Representing Shape by Line Patterns

Gabriella Sanniti di Baja

Istituto di Cibernetica, National Research Council of Italy,
Via Toiano 6, I-80072 Arco Felice (Napoli), Italy
gsdb@imagm.na.cnr.it

Abstract. A digital pattern can be represented by a stick-like subset, centred within the pattern. This subset can be obtained by thinning or skeletonizing the pattern. Although thinning and skeletonization are terms often used interchangeably in the literature, they are different processes and identify slightly different pattern subsets. Some of the numerous thinning and skeletonization algorithms available in the literature are briefly discussed.

1 Introduction

Picture processing necessarily involves the design of methods for both representing and manipulating images inside an electronic device. In particular, representation is a crucial task because the ability to efficiently process pictorial data greatly depends upon the suitability of the encoding technique used to represent the input image. Depending on the specific problem, different techniques can be adopted for pictorial data representation. When data compression is mainly motivated by the need of reducing memory occupation, the coding technique should be information preserving, in the sense that the input data should be perfectly reconstructed starting from the adopted representation scheme. When data compression is required in the framework of feature extraction applications, the coding technique should reduce memory occupation in such a way that enough information is preserved to automatically distinguish patterns belonging to different classes. In this case, the coding technique produces an approximate representation.

Thinning is useful for both data reduction and pattern representation. Generally, it is performed by iteratively removing suitable contour pixels from the pattern, while retaining the topological properties of the original image. The terms thinning and skeletonization have often been used interchangeably in the literature, while a distinction is here made between the two. Thinning is a process that applied to elongated patterns characterised by nearly constant thickness (e.g., printed or handwritten characters, line drawings, or some biological specimens) leads to a set of lines centred within the pattern and retaining the relevant structural and shape information of the pattern. This set of lines is called here the medial line (*ML*, for short). Skeletonization is suited to patterns that are not elongated or have variable thickness, and originates a stick-like representation (called the skeleton) including also branches originating from contour convexities.

2. Preliminary Notions

Let *B* and *W* be a pattern and its complement in a binary image *I*, digitised on the square grid. The sets *B* and *W* are also referred to as the sets of the black pixels and the white pixels. The 8-metric and the 4-metric are used to define connectedness on B and W, respectively. Without loss of generality, we suppose that B is constituted by a single component; no assumption is done on the number of components constituting W. The frame of I is the set of all pixels in the first or last row of I, or the first or last column of I. We assume that all the pixels of the frame are white.

For every black pixel p, let $N(p) = \{n_k \mid 1 \leq k \leq 8\}$, where $n_1 ..., n_8$ are the pixels successively encountered while going clockwise around p starting from the pixel to the left of p. The n_k are also called the neighbours of p.

The contour C of B includes every black pixel p for which at least one n_k, k odd, is white. Accordingly, C is an 8-connected curve, when B is simply connected, while is union of 8-connected curves, when B is multiply connected.

A 3x3 topology preserving removal operation can be designed by taking into account the notion of simple point [1], or by using the crossing number [2] or the connectivity number [3]. Alternatively, since the number of configurations of black and white pixels in a 3x3 neighbourhood is limited to 256, one can exhaustively identify all the templates allowing safe removal of the central pixel.

The distance transform DT of B is a multivalued replica of the pattern, where each pixel is labelled with its distance from W.

In the DT, the label of a pixel measures the radius of the disc that can be centred on the pixel and fits the shape of the pattern. The shape of the disc depends on the distance function used to compute the DT. The discs are diamond shaped when using the city-block distance $d_{1,2}$, and square-shaped with the chessboard distance $d_{1,1}$. The non circular shape of these discs is due to the fact that both the city-block and the chessboard distance functions provide a quite rough approximation to the Euclidean distance. To get a closer approximation, one should suitably weigh the unit moves from a pixel p to any of its neighbours, so as to take into account that horizontal/vertical moves and diagonal moves have different Euclidean length. The resulting distance function is called a weighted distance function. To avoid resorting to real numbers, suitable integer weights are used. A thorough investigation of the weighted distance functions and weight selection can be found in [4-5]. A good approximation is obtained by using the weights 3 and 4, for the horizontal/vertical and diagonal moves, respectively. The corresponding distance function is denoted by $d_{3,4}$. If this function is used to compute the DT, the disc associated with any pixel is octagon-shaped. An even better approximation of the Euclidean distance can be obtained by enlarging the set of neighbours of p, so as to include also the pixels that could be reached by the knight's move (in the game of chess). In this case, a third weight has to be introduced. The weights 5, 7 and 11 can satisfactorily be employed for the horizontal/vertical, diagonal and knight moves, respectively and the corresponding distance is denoted by $d_{5,7,11}$.

A pixel is a centre of maximal disc, if the associated disc is maximal, i.e. is not included by any other single disc of B. Centres of maximal disc will be denoted by *cmd*'s in the following. The pattern can be recovered by the union of its maximal discs; the union of the maximal discs can be conveniently obtained by applying the reverse distance transformation to the *cmd*'s.

3 Thinning

Ideally, thinning is an isotropic compression process. Since compression takes place from all directions at the same rate, its implementation by means of a parallel algorithm is a natural choice. Indeed, both parallel and sequential algorithms have been developed. In parallel algorithms, e.g. [6-13], the processing in an iteration is a function of the pattern resulted from the previous iteration only. In sequential algorithms, e.g. [14-17], the pixels are processed one after another and are updated in terms both of the pattern resulting from the previous iteration, and of the modifications produced so far in the current iteration. Thus, the medial line *ML* obtained from a sequential algorithm depends on the order in which pixels are processed. Sometimes, spurious branches appear in a particular order of processing, but do not appear in a different order. In the following we focus on parallel thinning algorithms, because potentially the resulting medial lines have always or never branches originating from equally shaped, differently oriented, pattern protrusions.

It is known that a fully parallel thinning cannot guarantee global topology preservation, if 3×3 operations are used [18]. Pattern regions that are initially an even number of pixels in width are first thinned to two pixels wide sets and then disappear completely, respectively causing excessive shortening of the *ML* branches and disconnections. In fact, one side of a two pixels wide set is removed on the assumption that connectedness is provided by the presence of the other side. However, since the operations take place in parallel, the other side is removed on a similar assumption.

Some authors (see, for instance, [9]) have overcome this trouble by examining a larger neighbourhood. A more commonly followed alternative is to perform thinning by dividing each iteration into a suitable sequence of subiterations, during each of which thinning is performed from given directions (e.g., [1,8]). Selection of the thinning operation should be anyway accomplished by taking into account the adopted sequence. In fact, the sequence of subiterations generally influences the isotropy of the transformation [15], and may bias the medial line structure. Moreover, it affects also the type of layer, whose pixels are the candidates for removal in an iteration.

The algorithms considered in this section represent various ways in which compression takes place depending on the adopted sequence of subiterations. Each scheme corresponds to a different type of propagation of the background over the pattern. Some algorithms remove a layer that is at most an 8-connected curve per iteration. Others remove at most a 4-connected layer. Yet, others remove a subset of an 8- or of a 4-connected layer, depending on pattern orientation.

In the following, thinning algorithms are analysed in terms of the size of the neighbourhood, the number of subiterations, the type of layers removed per iteration, and the quality of the resulting *ML*. All the algorithms described below adopt the 8-connectedness for *B* and the *ML*.

In [6] Stefanelli and Rosenfeld proposed two algorithms for parallel thinning based on 3×3 operations. In the first algorithm, every iteration is split into four subiterations, respectively removing the South, North, West and East contour pixels, and two templates are used in each subiteration. Removal of a contour pixel during any subiteration may expose to the background pixels which were not originally contour pixels. These extra pixels are checked against removal and are possibly removed in a subsequent subiteration. The effect is that a 4-connected layer is removed in every iteration. In many cases, this algorithm gives an *ML* with more pixels than is necessary to maintain connectedness. The second algorithm in [6] is derived from the first algorithm. It has two subiterations, obtained by combining the first and fourth subiterations to remove the South or East contour pixels, and the second and third subiterations to remove the North or West contour pixels. In any iteration, a 4-connected layer is removed from the Southeast and Northwest facing edges while an 8-connected layer is removed from the Southwest and Northeast facing edges. As a consequence, thinning is twice as fast in the Southeast and Northwest directions than in the Southwest and Northeast directions. This non isotropic compression causes an unwanted bias in the resulting medial line. A further drawback of this algorithm is complete vanishing or disconnections may occur during the process. See Figure 1a. The thinning algorithm proposed by Zhang and Suen [7] is a 2-subiteration algorithm based on 3x3 operations. South or East contour pixels are removed in the first subiteration, North or West contour pixels in the second subiteration. Removal is done safely, as the notion of crossing number and the cardinality of the set of the black 8-neighbours are used. A drawback is that the resulting medial line has often more pixels than necessary to maintain 8-connectedness. Asymmetric compression, which generally affects 2-subiteration thinning algorithms is prevented by checking also pixels that are both North and West contour pixels in the first subiteration, and pixels that are both South and East contour pixels in the second subiteration. The result is that a 4-connected layer is removed per iteration. Complete vanishing is likely to occur in critical patterns and, since the end point detection criterion is not adequate for a correct mapping of the pattern protrusions into *ML* branches, excessive branch shortening is likely to happen. See Figure 1b. In [8], two algorithms based on 3x3 operations are proposed. We refer here to the first algorithm, which employs two subiterations. In the first subiteration, East and some of the North contour pixels are checked, while in the second subiteration, West and some of the South contour pixels are checked. In this way, differently from many algorithms with two or four subiterations, an 8-connected layer is removed at each iteration, instead of a 4-connected layer. A side effect is a reduction in speed because there are less pixels in an 8-connected layer than in a 4-connected layer and hence less pixels are removed per iteration. This algorithm always produces a one pixel thin *ML*, with the possible exception of the central pixel in a T-junction. See Figure 1c. A one-pass algorithm is reported in [9]. As a counterpart to the fact that there are no subiterations, a 4x4 neighbourhood has to be used to detect and safely thin down two pixel wide regions. If a vertical two pixel wide feature is found, the East side is retained and the West side is removed. If a horizontal two pixel wide feature is found, the North side is retained and the South side is removed. This algorithm removes less pixels than there are in an 8-connected layer, per iteration. In fact, pixels placed in some locally concave configurations are not removed, even though their removal would not disconnect the pattern. They may be removed in a subsequent iteration if the neighbourhood configurations have suitably changed by then. The resulting medial line may contain more pixels than necessary to maintain connectedness. Moreover, there are certain configurations which cannot be thinned down completely. See Figure 1d. Finally, in [10] a 4-subiteration parallel thinning algorithm has been proposed, which is based on the use of 3x3 templates. It compresses the pattern in an octagonal manner, by removing a pair of successive layers (a 4-connected layer and an 8-connected layer) during each iteration. This aim is reached by compressing the pattern, at each subiteration, from a pair of directions forming a 90 degree angle. The use of the octagonal metric enables the placement of the medial line in the region of the pattern, which can be seen as central in a quasi-Euclidean sense, and renders the *ML* less sensitive to pattern orientation. Another feature of the algorithm is a carefully chosen end point detection process, which allows one to have consistent results with respect to regular patterns such as disks and squares at various orientations. Although the compression process is isotropic, thinning is not completely isotropic, because of the articulation of any iteration into subiterations. The effect of the sequentiality, introduced in the parallel process, could be that some branch of the *ML* is shorter than the others, as the end points are not detected in all the homologous positions of the same contour configuration, when this is differently oriented. This degrades the performance of the algorithm, but is anyhow preferable to the complete absence of one or more branches. See Figure 1e.

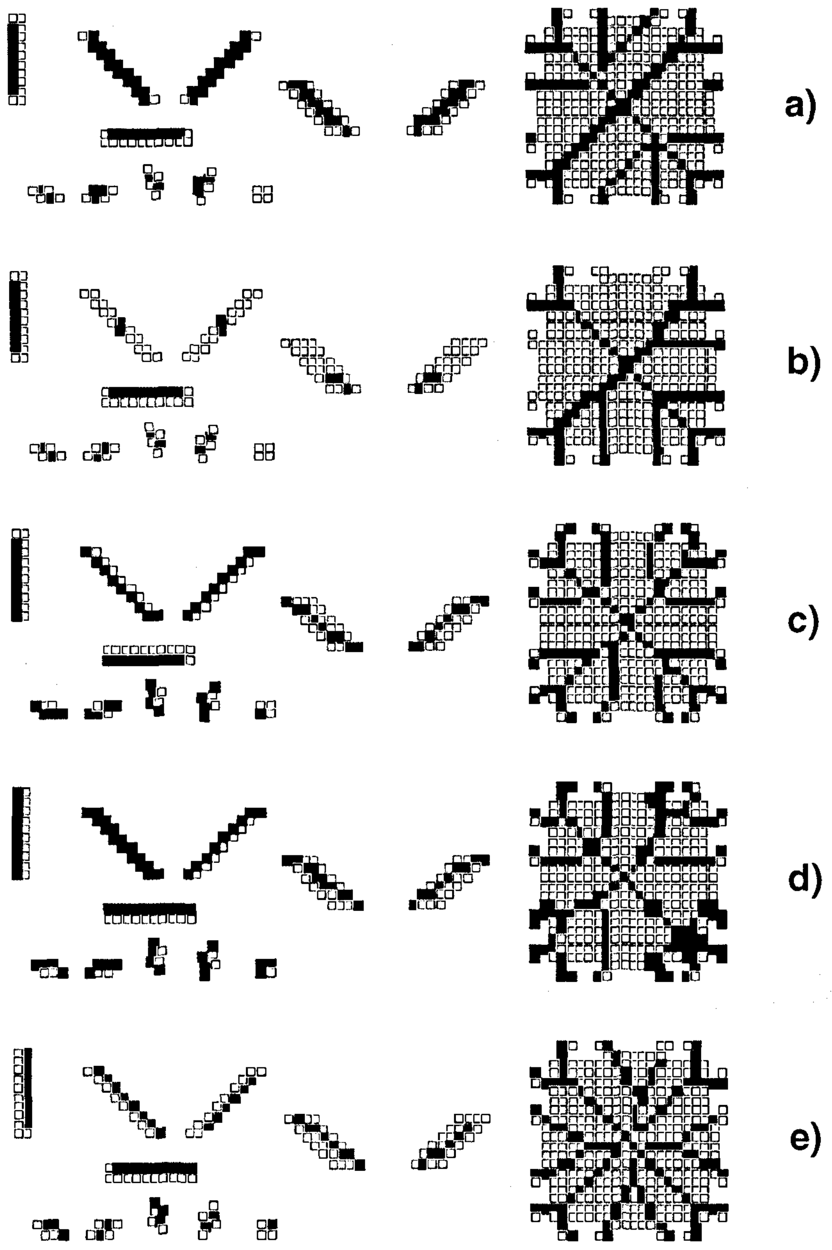

Figure 1. Critical patterns for thinning. Performance of five parallel algorithms. See text.

4 Skeletonization

Skeletonization of a digital pattern is a process leading to the extraction of a piecewise curvilinear subset, the skeleton, which is spatially placed along the medial region of the pattern. A skeleton is a stick-like representation of the pattern and, depending on the problem domain, is expected to account for different shape properties, such as symmetry, elongation, width, and contour curvature. Differently from thinning, skeletonization is generally a reversible transformation.

Most of the research on skeletonization has been influenced, at least implicitly, by the work of Blum dealing with a geometry based on the primitive notions of a symmetric point and a growth process [19-21]. In a continuous pattern, a point *p* is called symmetric if at least two points of the boundary exist, such that their distances from *p* are equal to the distance from *p* to the boundary. For every symmetric point, the associated maximal disc is the largest disc, obtained by growth of the symmetric point, which is contained in the pattern. The set of symmetric points, each labelled with the radius of the associated maximal disc, constitutes a skeleton of the pattern. The pattern can be exactly reconstructed as the union of the maximal discs, the envelope of the discs being the pattern boundary.

Skeletonization can also be understood by referring to the propagation of fire fronts [22]. Consider a flat uniform field of dry grass embedded in an environment of wet grass, and imagine that, at a certain instant of time, a fire is lit simultaneously at all points on the boundary of the field. As time passes, the fire front propagates towards the inside with uniform velocity and disappears wherever it intersects with itself. This occurs at those internal points of the pattern, called extinction points, which are reached by the fire at the same instant of time from different boundary points. Due to the constant velocity of propagation of the fire front, any extinction point is a symmetric point as defined above. The extinction points form a connected piecewise curvilinear set, centred within the pattern. If each extinction point is labelled with the value of the instant of time at which it is reached by the fire front, the set of labelled extinction points constitutes a skeleton of the pattern.

In the digital case, unit velocity is usually assumed for the fire front. After one unit of time, all the pixels at unit distance from the fire front are reached. The label of any pixel is then equal to the distance of the pixel from the complement of the pattern. The set of pixels where the fire extinguishes includes all the centres of the maximal discs, i.e., the discs not properly included in any other disc, and the union of the maximal discs coincides with the pattern.

Although the structure of the skeleton S depends on the algorithm employed for its computation, S always has one or more of the following properties: 1) S has the same number of components as B, and each component of S has the same number of holes as the corresponding component of B. 2) S is centred within B. 3) S is a unit-wide union of simple arcs and curves. 4) The pixels of S are labelled with their distances from W. 5) S includes the centres of the maximal discs of B. 6) S has arcs that correspond to regions of B bounded by contour subsets with sufficiently high curvatures.

Skeletonization can be achieved by repeatedly applying a contour peeling process, or by using a distance transform based approach. The latter method is more directly related to the Blum's notions of a symmetric point and a growth process. In fact, in the distance transform the pixels are labelled with their distance from the complement of the pattern, computed according to a given distance function. Thus, the pixels symmetrically placed within a digital pattern, as well as their associated radii, can be easily found in the *DT*.

The pattern can be almost completely recovered by applying to its skeleton the reverse distance transformation. Complete recovery is not compatible with the requirement that the skeleton be one pixel wide. In fact, this requirement forces removal of a number of centres of maximal discs from the set of the skeletal pixels.

Roughly speaking, a distance-driven skeletonization algorithm includes three steps:

- distance transform computation
- identification of a nearly thin set of the skeletal pixels (centres of the maximal discs - necessary to guarantee skeletonization reversibility-, saddle pixels and linking pixels - necessary to guarantee topology preservation)
- reduction of the set of the skeletal pixels to unit width

Detection of the centres of the maximal discs in the *DT* can be done by suitably comparing the label of any pixel (i.e., the radius of the associated disc) with the labels of its neighbours (i.e., the radii of the associated discs). Generally, the set of the centres of the maximal discs is not connected, even for a connected pattern, and is more than one pixel wide, wherever the thickness of the pattern is given by an even number of pixels. To gain skeleton connectedness, further skeletal pixels (the saddle pixels, and the linking pixels) have to be found on the distance transform. Detection of the saddle pixels can be done by analysing the neighbourhood of any pixel, so as to count the number of components of neighbours with smaller label and with larger label. Detection of the linking pixels can be done by growing paths along the direction of the steepest gradient in the distance transform, starting from any already found centre of maximal disc or saddle pixel. Finally, the set of the skeletal pixels can be reduced to the unit wide skeleton, by employing topology preserving removal operations, designed in such a way to prevent excessive shortening of the skeleton branches.

A number of algorithms can be found in the literature, each of which tailored to a specific distance function [23-27]. Generally, although all these algorithms follow more or less the above scheme, ad hoc rules are often used (for instance to identify the centres of the maximal discs, or to obtain skeleton connectedness through the linking pixels) which apply only to the specific distance case. A few algorithms can be applied to more than one distance transform [28,29]. Most of the available skeletonization algorithms do not include an important step, devoted to skeleton pruning and beautifying. This step should not be simply regarded as an optional postprocessing for a skeletonization algorithm. In fact, pruning is useful to get rid of superfluous noisy branches and is indispensable to make the skeleton stable under pattern rotation, by eliminating those branches whose presence in the skeleton depends on pattern orientation. In turn, beautifying can improve skeleton aesthetics and favour its use for shape analysis. In this paper we briefly illustrate the algorithm introduced in [29]. A peculiarity of this algorithm is the inclusion of the pruning and beautifying step. The algorithm equally runs whichever path based distance is used. Indeed, it refers to four possible distances only (the city-block distance $d_{1,2}$, the chessboard distance $d_{1,1}$ and the weighted distances $d_{3,4}$ and $d_{5,7,11}$), but could apply also in case of other distances, if suitably slightly modified. These four distances have been chosen for the following reasons: the city block and chessboard distance functions are the most commonly used and, due to the shape of the disc associated to any skeletal pixel by them, are particularly suited to the square tessellation of the discrete plane; the $d_{3,4}$ and the $d_{5,7,11}$ weighted distances represent very good approximations to the Euclidean distance by using respectively only two and three weights.

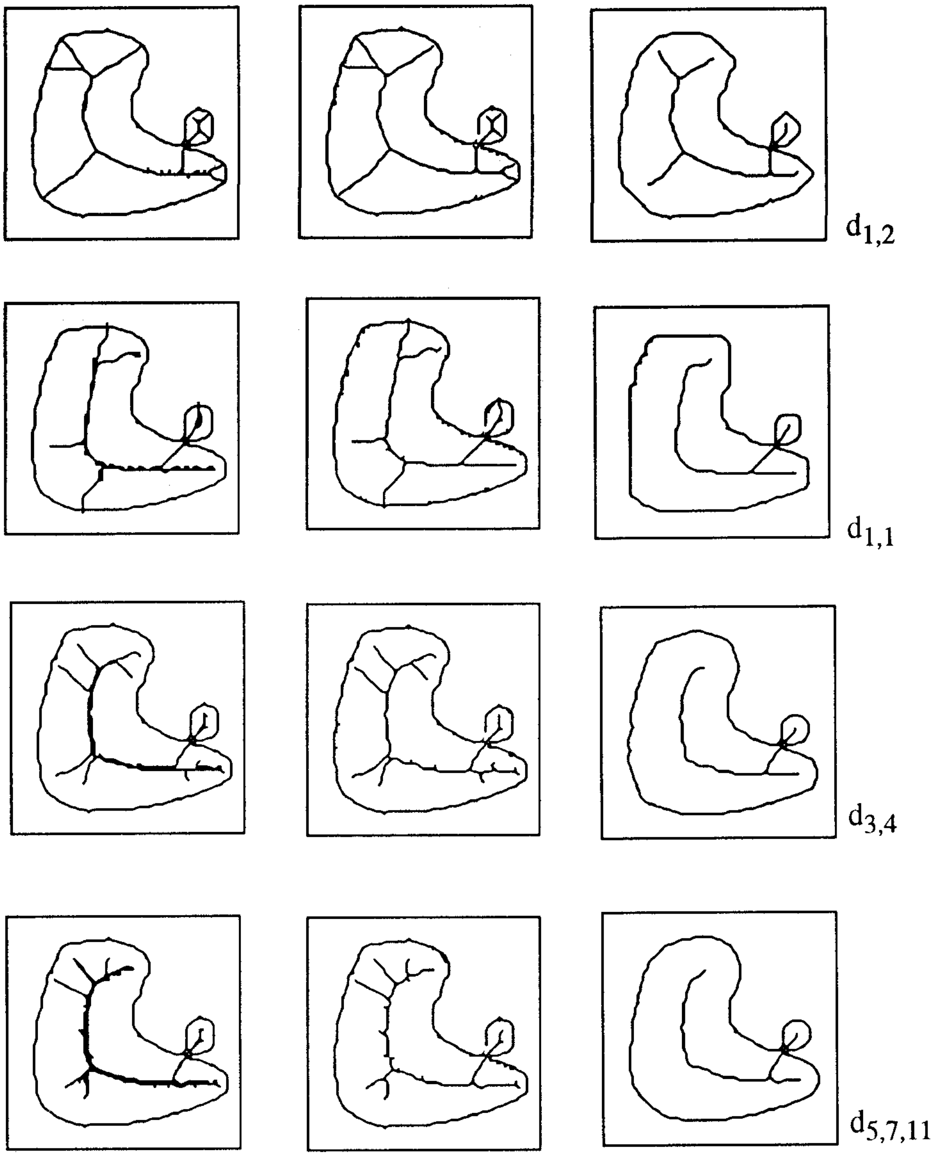

Figure 2. Skeletons computed on different distance transforms. See text.

The centres of the maximal discs are identified by resorting to the use of look-up tables, while the saddle pixels are detected by means of 3×3 local operations. The linking pixels are identified by growing paths through the ascending gradient in the *DT*. Path growing is tempted each time a centre of maximal disc or a saddle pixel is identified and marked. The neighbour having label larger than that of the marked pixel and maximising the gradient is marked as the first linking pixel in the path. Its neighbours are then inspected to identify the one maximising the gradient. Path growing terminates when the maximal gradient is no longer positive. An unmarking process is then accomplished on the *DT*, so as to reduce to unit width the set of the skeletal pixels; to this purpose, standard topology preserving removal operations are employed to remove from any deletable pixel the marker, previously ascribed to distinguish it from the non-skeletal pixels of the *DT*. Since the set of the skeletal pixels can include internal pixels, to favour skeleton centrality within the pattern, 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. Finally, an unmarking-and-shifting process is done to prune the skeleton and to improve its aesthetics. During this step the marker is removed from a skeletal pixel or is shifted to some of its non marked neighbours. The main effect of beautifying consists in the reduction of the zigzags due to noise on the contour of *B* and/or to the process done to obtain a unit wide skeleton. Pruning is accomplished to simplify the skeleton structure by deleting peripheral branches 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. A smoothed version of the input pattern is recovered starting from the pruned skeleton.

Different distance transforms originate different skeletons for the same pattern. As an example, refer to Figure 2 where the skeletons computed according to $d_{1,2}$, $d_{1,1}$, $d_{3,4}$ and $d_{5,7,11}$ are shown from top to bottom. For each set, the nearly thin set of the skeletal pixels (left) and the unit wide skeleton (centre) are shown superimposed on the original pattern, while the pruned skeleton (right) is shown superimposed on the pattern recovered by the pruned skeleton itself.

References

1 A. Rosenfeld, "A characterization of parallel thinning algorithms", Inf. Control, **29**, (1975), pp. 286-291.

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

3 S.Yokoi, J.I. Toriwaki, T. Fukumura, "An analysis of topological properties of digitized binary pictures using local features", *Comput. Graphics Image Process.*, **4**, (1975), pp. 63-73.

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

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

6 R. Stefanelli, A. Rosenfeld, "Some parallel thinning algorithms for digital pictures", *J. ACM*, **18**, (1971), pp. 255-264.

7 T.Y. Zhang, C.Y. Suen, "A fast parallel algorithm for thinning digital patterns", *Communications ACM*, **27**, (1984), pp.236-239.

8 Z. Guo, R.W. Hall, "Parallel thinning with two subiteration algorithms", *Communications ACM*, **32**, (1989), pp. 359-373.

9 R. T. Chin, H.K. Wan, D. L. Stover, R. D. Iverson, "A one-pass thinning algorithm and its parallel implementation", *Computer Vision, Graphics and Image Processing*, **40**, (1987), pp. 30-40.

10 C. Arcelli, P.C.K. Kwok, G. Sanniti di Baja, "Parallel pattern compression by octagonal propagation", *Int. Journal of Pattern Recognition and Artificial Intelligence*, **7**, (1993), pp. 1077-1102.

11 C. J. Hilditch, "Comparison of thinning algorithms on a parallel processor", *Image Vision Comput.* , **1**, (1983), pp. 115-132.

12 Y.S. Chen, W.H. Hsu, "A systematic approach for designing 2-subcycle and pseudo 1-subcycle parallel thinning algorithms", *Pattern Recognition*, **22**, (1989), pp. 267-282.

13 Y.Y. Zhang, P.S.P. Wang, "A modified parallel thinning algorithm", *Proc.9th Int. Conf. on Pattern Recognition*, Rome (1988), pp. 1023-1025.

14 C. Arcelli, G. Sanniti di Baja, "On the sequential approach to medial line transformation", *IEEE Trans. Systems, Man, and Cybernetics*, **8**, (1978), pp.139-144.

15 C. Arcelli, G. Sanniti di Baja, "A thinning algorithm based on prominence detection", *Pattern Recognition*, **13**, (1981), pp. 225-235.

16 V. K Govindan, A. P. Shivaprasad, "A pattern adaptive thinnin algorithm", *Pattern Recognition*, **20**, (1987), pp. 623-637.

17 P.C.K. Kwok, "A thinning algorithm by contour generation", *Communications ACM* , **31**, (1988), pp. 1314-1324.

18 A. Rosenfeld, "Connectivity in digital pictures", *J. ACM*, **17**, (1970), pp. 146-160.

19 H. Blum, A transformation for extracting new descriptors of shape, in *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, ed., M.I.T. Press, Cambridge, MA, (1967), pp. 362-380.

20 H. Blum, Biological shape and visual science, *J. Theor. Biol.*, **38**, (1973), pp. 205-287.

21 H. Blum and R.N. Nagel, Shape description using weighted symmetric axis features, *Pattern Recognition*, **10**, (1978), pp. 167-180.

22 Y. Xia, Skeletonization via the realization of the fire front propagation and extinction in digital binary shapes, *IEEE Trans. Patt. Anal. Mach. Intell.*, **11**, (1989), pp. 1076-1086.

23 B.J.H. Verwer, Improved metrics in image processing applied to the Hilditch skeleton, *Proc.9th Int. Conf. on Pattern Recognition*, Rome, (1988), pp. 137-142.

24 C. Arcelli and G. Sanniti di Baja, A one-pass two-operation process to detect the skeletal pixels on the 4-distance transform, *IEEE Trans. Patt. Anal. Mach. Intell.*, **11**, (1989), 411-414.

25 C. Arcelli and G. Sanniti di Baja, Distance driven skeletonization, *Proc. IEEE Conf. on Computer and Communication Systems*, Hong Kong, (1990), pp. 304-308.

26 C. Arcelli and M. Frucci, Reversible skeletonization by (5,7,11)-erosion, in*Visual Form Analysis and Recognition*, C. Arcelli et Al. eds., Plenum, New York, (1992), pp. 21-28.

27 G. Sanniti di Baja, Well-shaped, stable and reversible skeletons from the (3,4)-distance transform, *J. Visual Comm. Image Repres.*, **5**, (1994), pp. 107-115.

28 L. Dorst, Pseudo-Euclidean skeletons, *Proc. 8th Int. Conf. on Pattern Recognition*, Paris, (1986), pp. 286-288.

29 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, (1994), pp. 475-486.