Width-Independent Fast Skeletonization Algorithm for Binary Pictures

V. M. Kiyko and M. I. Schlesinger

Department 420, V. M. Glushkov Institute of Cybernetics, Academy of Sciences of the Ukrainian SSR,
Prospekt Akademika Glushkova, 20/22, Kiev 207, 252207, USSR

ABSTRACT

An effective algorithm of skeletonization for digital binary pictures based on input-time-tracing principle and sequential decomposition of input picture into simple parts is described. The proposed algorithm is fast and requires no image memory. Its computing complexity does not depend on the thickness of picture patterns and is determined not by the number of picture pixels but by the essentially small number of horizontal and vertical lines constituting the contour of the picture.

I. INTRODUCTION

The well known skeletonization procedure for binary pictures introduced by Blum [1] consists of determining the maximal circular disks in the picture; disks that fit inside the picture objects but in no other disk within picture ones. The set of centers and radii of maximal disks in the picture is termed its skeleton, representing a convenient description of the form and thickness of the elongated objects in the picture. The skeleton preserves all the details of any object in the picture and permits its reconstruction; thus there is no loss of information. The skeletonization problem is equivalent also to determining for a black object in a two-dimensional Euclidean plane the set of points $\{x\}$ such that there are at least two points of the contour that are equidistant from $\{x\}$ and are closest to $\{x\}$.

There are many algorithms defining skeletons in a number of ways in the continuous as well as in the discrete plane, but the majority of implementations use discrete space concepts. Those algorithms that use Euclidean [2, 3] or quasi-Euclidean [4] metrics on quantized forms are time-consuming and exhibit severe drawbacks. The algorithm [5] defines a skeleton of a ribbon or tree-like object only using sequential data of its boundary. The algorithms [6–12] determine not maximal disks but maximal squares in the picture. Thinning algorithms, such as those described in Refs. 6 and 7, use representation of the picture as an array of white and black pixels and delete iteratively black border pixels if their removal does not affect the connectivity of the objects. These algorithms are generally expensive both in terms of the time and in terms of data required for skeleton generation. The distance transform (DT) technique developed by Rosenfeld and Phaltz [8] determines at the first step the distances of black pixels to the object boundary by twofold scanning of the picture. After that skeletal pixels having local maximum of this distance are defined. Unfortunately, the DT technique requires image memory also and does not preserve the connectivity of the skeleton derived.

There are many algorithms, such as those described in Refs. 10 and 11, having much in common with the algorithms described in Refs. 6 and 8. The algorithm developed by Wakayama [12] has more essential distinctive features. This algorithm is based on an input-time-tracing principle and defines core lines of objects in a digitized binary picture by introduction, enlarging, deriving, and meeting operations on squares in this picture. The computing complexity of this algorithm and memory area required for the input data depend on the length of the largest maximal square in the picture.

The algorithm below preserves some attractive features of the Wakayama algorithm and defines a skeleton of the objects having a quantized form in the continuous plane. This algorithm is based on the successive decomposition of the input picture into parts, producing such elementary parts that its skeletonization is evident. The set of defined maximal square centers represents, unlike Refs. 6–11, not isolated points but thin lines. Thus the proposed algorithm, like the Wakayama algorithm, is oriented towards a structure-descriptive core line representation. The connectivity of the derived skeleton is preserved as far as its original object is connected. This algorithm is fast because it uses economical representation of the input picture and its principle is based, like Wakayama's, on the input-time-tracing concept. The computational complexity of the algorithm does not depend on the size of largest maximal square in the picture. It is of order $O(N_{\text{cor}})$, where N_{cor} is number of horizontal and vertical lines constituting the contour of the input picture. Memory area required for the input data also does not depend on side length of the largest maximal square and is of order $O(n)$, where n is the number of rows in an $n \times n$ picture array. The algorithm has also other distinctive features described below.

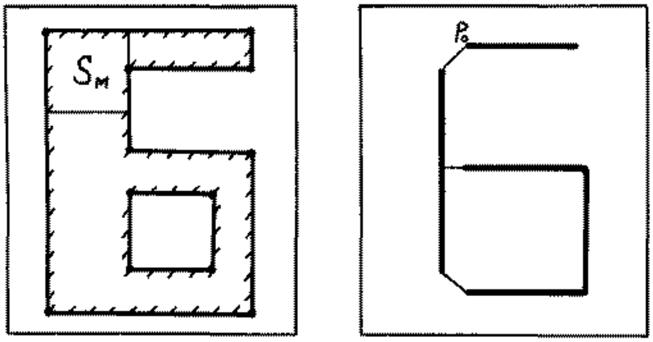
II. SKELETON DEFINITION

Let the picture $V \subset R^2$ be the subset of black points on a Euclidean plane, every connected component of the contour (boundary) of this picture being a closed polygonal line without self-crossings, composed of a finite number of horizontal and vertical line segments. A certain square $S_m \subset$

Received 4 April 1991

International Journal of Imaging Systems and Technology, Vol. 3, 222–226 (1991)

CCC 0899–9457/91/030222–05$04.00



(a) (b)

Figure 1. The basic and auxiliary skeletons (b) of the picture (a).

R^2 is called maximal square if (a) the sides of this square are the segments of vertical and horizontal lines, (b) $S_m \subset V$, and (c) there is no other square $S \neq S_m$ for which conditions (a), (b), and $S_m \subset S$ are satisfied simultaneously (Fig. 1).

The basic skeleton of the picture V is defined as the set of central points of the maximal squares. Every connected component of the basic skeleton is either an isolated point, or consists of one or more horizontal and vertical line segments. There are two types of the ends of the basic skeleton segments: the close end if it is the central point of the maximal square and the open end otherwise. Let p_o be the open end of the segment of the basic skeleton. The maximal square S_m is defined as corresponding to point p_o if S_m contains all squares $S \subset V$ with the central point p_o. The auxiliary skeleton is the set of additional line segments, each of them linking the open end of the basic skeleton segment with the central point of the corresponding maximal square. The basic and auxiliary skeletons are shown in Fig. 1 by thick and thin lines, accordingly.

Our goal is to elaborate the effective algorithm for detecting the basic and auxiliary skeletons in any binary picture V.

III. SKELETONIZATION VIA DECOMPOSITION OF THE PICTURE INTO SIMPLE PARTS

Let vertices of those polygons that constitute the contour of the picture be called corners. All eight types of corners are shown in Fig. 2, corners of types 1–4 being called convex and the others concave.

Let q be the concave corner of the picture V. A certain square S_d is called the dissective square (Fig. 3) if (a) $S_d \subset V$, (b) one of the vertices of the square S_d coincides with the corner q, (c) one of the diagonals of the square S_d coincides with the bisectrix of the corner q, and (d) there is no square $S \neq S_d$ for which conditions (a), (b), (c), and $S_d \subset S$ are satisfied simultaneously. Due to the last feature of the square S_d at least one of its two sides not containing q has a nonempty intersection with the contour of the picture V. Let q^* be any point of this intersection. Let us suppose also that points q and q^* belong to the same connected component G_c of the contour of the picture V. In this case points q and q^* split the contour G_c into two parts G_{c1} and G_{c2} and the boundary of the square S_d into two parts G_{s1} and G_{s2}. Splitting the picture V into two pictures V_1 and V_2 (Fig. 3) is called decomposition if (a) $V_1 \cap V_2 = S_d$ and (b) the contour of each picture V_1 or V_2 contains one and only one of the parts G_{c1}, G_{c2} and only one of the parts G_{s1}, G_{s2}. Let us define also a connector as the line segment linking central points of two maximal squares containing the square S_d and belonging to the pictures V_1 and V_2 correspondingly. The following three statements are valid:

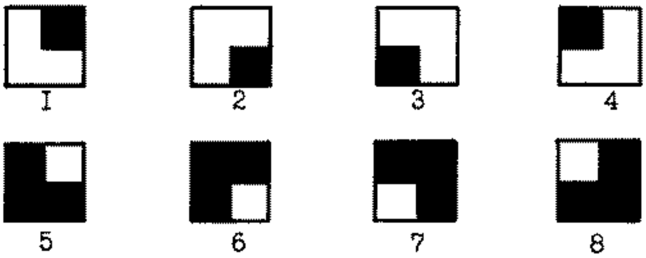


Figure 2. The types of the picture corners.

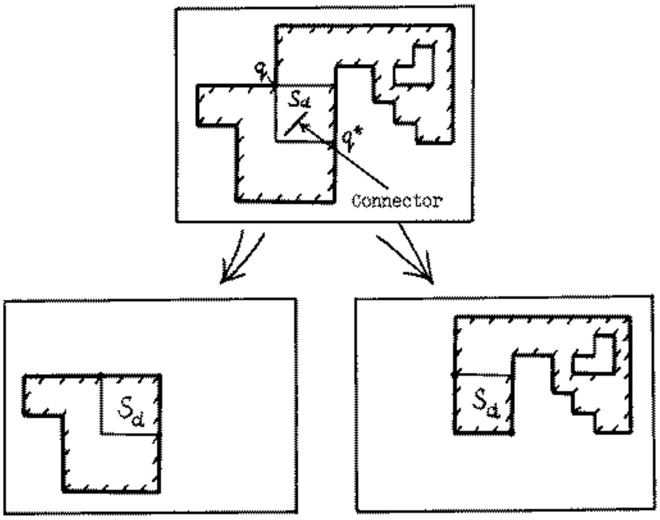


Figure 3. Decomposition of the picture into two subpictures.

Statement 1. The basic skeleton of the picture V is the union of the basic skeletons of the pictures V_1 and V_2.

Statement 2. The auxiliary skeleton of the picture V is the union of the auxiliary skeletons of the pictures V_1, V_2 and a third set containing only the connector.

Statement 3. Perimeter (area) of the picture V is equal to $P_1 + P_2 - P_d$, where P_1, P_2, P_d are perimeters (areas) of the pictures V_1, V_2, and of the dissective square S_d correspondingly.

A certain picture is called a simple picture (Fig. 4) if its contour contains no concave corners (see Fig. 2) of the types 5

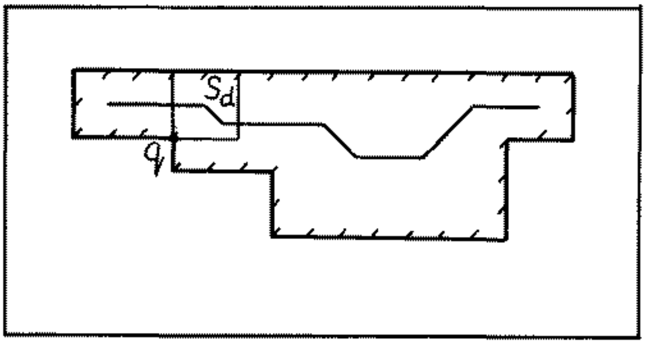


Figure 4. The simple picture V_s and its skeleton.

or 8 and intersects with no vertical sides of dissective squares corresponding to concave corners of the types 6 or 7 in this picture. If the simple picture contains no concave corners, then it is a rectangle and its skeletonization is obvious. Otherwise the skeleton of the simple picture V_s can be detected in the following manner. The concave corner $q \subset V_s$ with the least coordinate x and dissective square corresponding to this corner are determined. The side length of this square equals $(y_n - y)$, where y_n is ordinate of the upper edge of the simple picture and y is ordinate of the corner q. Then the decomposition of the simple picture V_s into two parts is carried out, one of these parts being the rectangle and the other being the simple picture with no concave corner q. These operations are repeated to complete removal of concave corners from this picture.

Let V_{ns} not be a simple picture containing no concave corners of the types 5 or 8 (Fig. 5). The contour of this picture contains one corner of the type 2 and one corner of the type 3. These two corners are connected by a horizontal line segment and a broken line $q_1, q_2, \ldots, q_n$. The algorithm of skeletonization for picture V_{ns} is based also on presentation of the picture as the union of two subpictures. Accordingly, there are two steps of picture processing to this algorithm. The first one consists in successive decomposition of the picture V_{ns} into two subpictures in every corner q_i corresponding to one of two situations presented in first column of Table I. The actions in each of these situations are presented by the pair of pictures in the second and third columns of this table: the fragment illustrating the situation and dissective square corresponding to the corner q_i are shown in the second column, and decomposition of the picture into two subpictures is shown in the third. One of these subpictures is the rectangle with basic skeleton representing vertical line segment and the other is the picture with no concave corner q_i. The set of considered operations is called the horizontal cutoff procedure of the picture. This procedure results in defining vertical line segments of the basic skeleton of the picture V_{ns} as far as in producing of the simple picture $V_s \subset V_{ns}$. The second step of picture V_{ns} processing consists in skeletonization of the simple picture V_s by the above-mentioned algorithm.

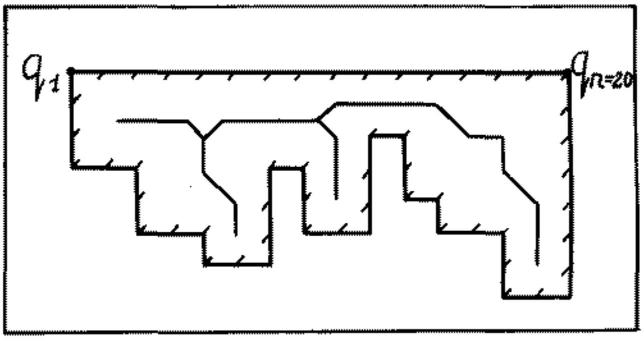
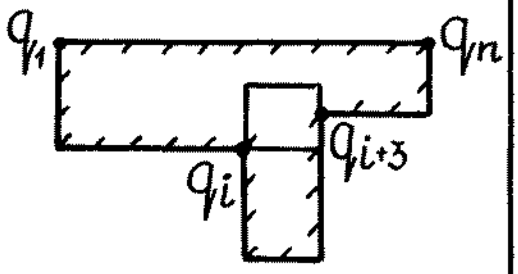
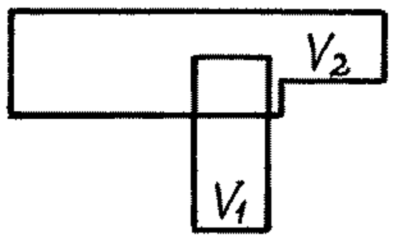
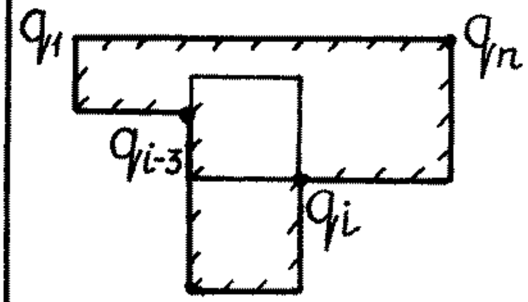
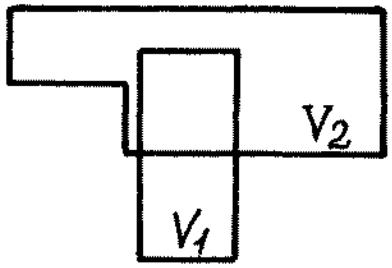


Figure 5. The picture V_{ns} and its skeleton.

The computational complexity of pictures V_s and V_{ns} skeletonization is of order $O(N_{\text{cor}})$, where N_{cor} is the number of corners in these pictures.

Table I. Decomposition operations of the picture V_{ns}.

Situations	Example of the picture V_{ns}	Picture decomposition
$x_{i+3} - x_i \le y_n - y_i$, $y_{i+3} > y_i$		
$x_i - x_{i-3} \le y_n - y_i$, $y_{i-3} > y_i$		

IV. ALGORITHM DESCRIPTION

Let us define a corner line as a horizontal line in the picture V containing two or more of its corners. The algorithm processes the picture using bottom-top scanning, so that the boundary of the scanned part successively occupies positions coinciding with the corner lines in this picture.

Let i be the number of the current corner line and y_i be its ordinate. The processing of the picture $V_b(y_i) \subset V$ below this line results in defining some segments of the skeleton and in forming also the so-called residual picture $V_r(y_i) \subset V$, containing no concave corners of the types 5 and 8 (Fig. 6). The residual picture $V_r(y_i)$ either contains no black points or consists of one or more simple pictures with upper boundary ordinate being equal to y_i. The way to form the residual picture and supplementary properties of this picture will be defined below.

After processing the picture $V_b(y_i)$ the boundary of the scanned part of the picture V will coincide with the next $(i+1)$th corner line. The picture $V_b(y_{i+1})$ below this corner line represents the union of the residual picture $V_r(y_i)$ and the part of picture V between its ith and $(i+1)$th corner lines.

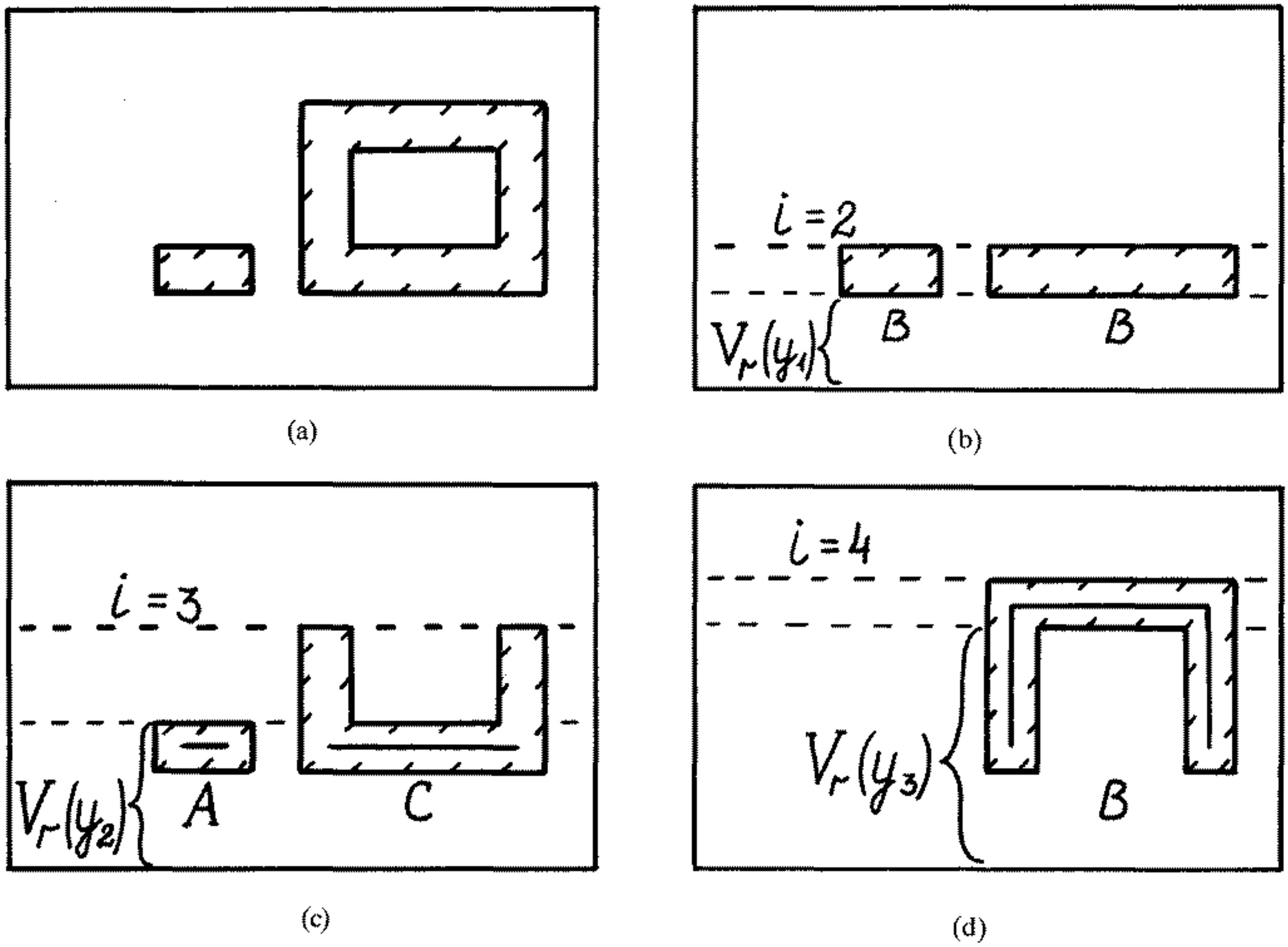
The picture $V_b(y_{i+1})$ consists of one or more simply connected components V_k. Each of these components has one of three following types: (1) type A of the simple pictures with upper boundary ordinate being equal to y_i, (2) type B of the simple pictures with upper boundary ordinate being equal to y_{i+1}, and (3) type C of the pictures with upper boundary ordinate being equal to y_{i+1} and with contour containing one or more corners of the types 5 or 8 with the ordinate y_i (Fig. 6).

The algorithm processes each component V_k as follows. If the picture V_k is of type A then successive decomposition and skeleton detection are carried out in this picture. If the picture V_k is of type B then after horizontal cutoff it will be remembered as the part of the residual picture $V_r(y_{i+1})$. Finally, if the picture V_k is of type C then for the first of its corners q of the types 5 or 8 the dissective square is to be formed. Then the decomposition of the picture V_k into the left V_{lk} and the right V_{rk} parts is carried out.

If the corner q is of type 8 then the picture V_{lk} is simple and its skeleton is defined by the aforesaid algorithm. If the corner q is of type 5 then the picture V_{lk} is of type B. After horizontal cutoff it must be remembered as the part of the residual picture $V_r(y_{i+1})$.

The picture V_{rk} may be of types A, B, or C. In the first two cases actions are the same as considered above. In the third case the same steps previously applied to the initial picture V_k should be applied to the picture V_{rk}, that is, the corner detection of the types 5 or 8 and so on.

The contour of the residual picture contains no concave



(a) (b) (c) (d)

Figure 6. Pictures $V_b(y_i)$ and $V_r(y_{i-1})$ under different positions of the scanned part of the input picture (a). (b) $i=2$, (c) $i=3$, (d) $i=4$.

corners of the types 5 and 8, so the following statement is valid:

Statement 4. The number of the corners in the residual picture does not exceed the value $2 * n$, where n is the corner line length: number of rows in an $n \times n$ picture array. It follows that memory area required for algorithm implementation does not depend on the length of the scanned part of the picture despite the size of defined maximal squares can be however large.

The algorithm has the most effective implementation when it uses an array of the coordinates of the corners $(x_1, y_1, x_2, \ldots, x_m, y_m)$ of the picture as the initial data. These coordinates are arranged in such order $i = 1, 2, \ldots, m$ that $i < j$ if $y_i < y_j$ or $y_i = y_j$, $x_i < x_j$. These data give the compressed coding of the picture, ensuring its error-free restoration and therefore determination of the types of all its corners also. The algorithm processes the picture along its corner lines. Input coordinates of the corners along any current ith corner line actually set the boundary of the scanned part of the picture forward the level of the next $(i+1)$th corner line. When corner coordinates of the last corner line are input, the overall skeleton is completely obtained.

The proposed algorithm possesses the following main features in comparison with the known algorithms: its computational complexity depends not on the thickness of the picture lines and is determined not by the number of the pixels but the essentially smaller number of the picture corners. The additional positive feature is that the algorithm requires comparatively small memory area for the input data depending on no length of the scanned part of the picture. This algorithm is oriented toward a structure-descriptive skeleton representation and its principle is based on the input-time-tracing principle.

V. PERIMETER AND AREA EVALUATION OF BINARY FIGURE

The final result of picture V is a graph $G = (D, N)$ having the same connectivity order of V with arcs $(i, j) \subset D$ being the skeleton line segments and nodes $i \subset N$ being end points of these segments. Each ith node of this graph is labeled by center coordinates (x_i, y_i) and side length t_i of the corresponding maximal square in the picture. It follows from aforesaid Statement 3 that perimeter P of picture patterns may be computed from its skeletal representation by

$$P = \sum_{(i,j) \subset S_b} 2 * (l_{ij} + 2 * t_i) - \sum_{i \in T_o} 4 * t_i \qquad (1)$$

where $S_b = \{((i, j) \subset N \mid i > j,\ x_i = x_j, \text{ or } y_i = y_j\}$ is the set of the basic skeleton line segments; l_{ij} is the length of the basic skeleton line segment connecting ith and jth nodes of the graph G; $T_o \subset N$ is the subset of N such that each ith node belonging to T_o corresponds either to open end of the basic skeleton segment (there is other $i1$th node neighboring to the ith node with $t_{i1} > t_i$) or to intersection of these segments. The area of picture patterns is computed by formula being similar to (1).

In Ref. 13 the algorithm for computing the area and perimeter of any digital figure from its discrete medial transform (MAT) is introduced. This algorithm requires $O(N_p)$ time, where N_p is the number of MAT pixels. Our algorithm requires essentially less $O(N_b)$ time, where N_b is the number of basic skeleton line segments. Its additional property is exact area and perimeter evaluation of binary figures from their skeletons. These distinctive features follow from using in the proposed algorithm more convenient representations of both input picture and resulted skeleton in comparison with Ref. 13.

VI. PERFORMANCE

The algorithm has been implemented in the hardware–software system for automatic input, processing, and semantic interpretations of color graphical pictures [14]. Despite the seeming complexity of the described algorithm the elaborated software implements it using only twofold tracing of the residual picture contour for each corner line in the input picture. Processing speed of this algorithm after some contour noise elimination is about 6×10^4 per second (PDP-11) for coding of sufficiently complex pictures, that is, at least several times faster in comparison with the algorithms in Refs. 4–11. The processing speed is maximal for pictures composed of horizonal and vertical lines and decreases for pictures containing slant lines due to the increased number of corners in these pictures.

REFERENCES

1. H. Blum, "A transformation for extracting new descriptors of shape," in *Models for the Perception of Speech and Visual Form*, edited by W. Whaten-Dunn (MIT Press, Cambridge, 1967), pp. 362–380. (Proc. of Meeting held Nov., 1964.)
2. H. Blum and R. N. Nagel, "Shape description using weighted symmetric axis features," *Pattern Recognition* **10**, 167–180 (1978).
3. P. E. Danielsson, "Euclidean distance mapping," *Comput. Graphics Image Process.* **14**, 227–248 (1980).
4. U. Montanari, "A method for obtaining skeletons using a quasi-Euclidean distance," *J. ACM* **15**, 600–624 (1968).
5. B. Shapiro, J. Piza, and J. Sklansky, "Skeleton generation from *x*, *y* boundary sequences," *Comput. Graphics Image Process.* **15**, 136–154 (1981).
6. T. Pavlidis, "A thinning algorithm for discrete binary images," *Comput. Graphics and Image Process.* **13**, 142–157 (1980).
7. P. S. P. Wang and Y. Y. Zhang, "A fast and flexible thinning algorithm," *IEEE Trans. Comput.* **38**(5), 742–745 (1989).
8. A. Rosenfeld and J.L. Pfaltz, "Sequential operations in digital picture processing," *J. ACM* **13**, 471–494 (1966).
9. C. Arcelli and G. Sanniti di Baja, "A width-independent fast thinning algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-7**, 463–474 (1985).
10. C. Arcelli, "Pattern thinning by contour tracing, "*Comput. Graphics Image Process.* **17**, 130–144 (1981).
11. M.A. Fishler and P. Barrett, "An iconic transform for sketch completion and shape abstraction," *Comput. Graphics Image Process.* **13**, 334–360 (1980).
12. T. Wakayama, "A core-line tracing algorithm based on maximal square moving," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-4**(1), 68–74 (1982).
13. L. P. Cordella and G. Sanniti di Baya, "Geometric properties of the union of maximal neighborhoods," *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-11**(2), 214–217 (1989).
14. V. V. Matsello, Y. L. Provalov, V. M. Sharipanov, and M. I. Shlesinger, "The system for automatic input and processing of the graphical information," *USIM*, No. 6, 56–60 (1986). In Russian.