FAST DILATION AND SKELETONIZATION OF COMPRESSED BINARY PICTURES

V.M. KIIKO and M.I. SCHLESINGER
International Research and Training Centre of Information Technologies and Systems of National Akademy of Sciences of Ukraine
Prospekt Akademika Glushkova, 40,252022,Kiev,Ukraine
E-mail: schles@image.kiev.ua

Abstract. Effective algorithms for dilation (erosion) and skeletonization of binary pictures, represented in compressed form, are described. The proposed line-by-line algorithms are fast and require small memory area, which does not depend on the size of a processed picture. The results of implementation the algorithms for processing of line drawings are presented.

1. Introduction

Let vertices of those polygons, that constitute the contour of a binary picture $v(i,j) \rightarrow \{0,1\}$, be called corners. There are eight types (Fig. 1,a) of the corners and the corners of types 1-4 be called convex and of types 5-8 be called concave ones. The set of coordinates of these corners represents one of the lossless compression forms [8,9] of the binary picture ensuring restoring of the initial pixel representation of this picture.

The corner representation (CR) can be effectively used, for example, for translation, scaling or inversion of the binary picture and for adding or subtraction of two binary pictures. For performing the first two operations only coordinates of the corners in the picture must be changed in some way. For inversion of the picture every of four corners in vertices of the rectangle, bounding the picture, must be added to its corner representation in the absence of this corner and removed otherwise.

Let t be a center of some square consisting of 4 pixels (Fig. 1,b), $\rho_1 - \rho_4$ be the values of these pixels and $d_1 = \rho_2 - \rho_3,\ d_2 = \rho_1 - \rho_4,\ dd = (d_1 - d_2)$ be

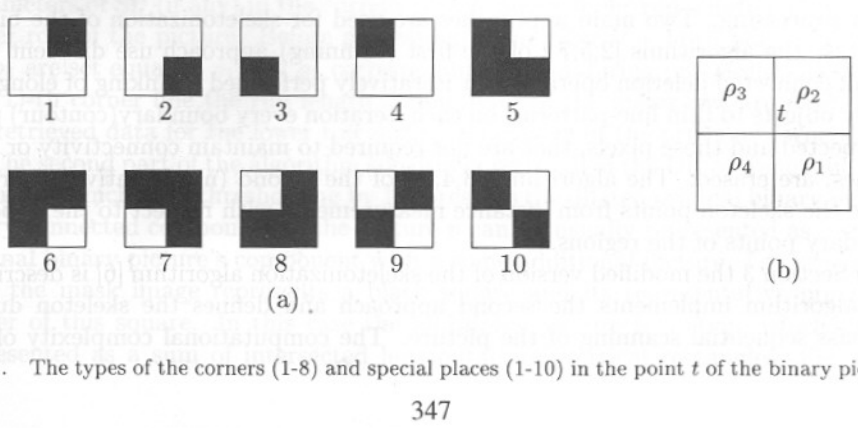

Fig. 1. The types of the corners (1-8) and special places (1-10) in the point t of the binary picture.

correspondingly the first, second and double differences of the pixel's values around this point. Therewith the point t is the corner in the binary picture if $|dd| = 1$ in this point. Some point be called also a special point (SP) in the binary picture, if this point is either the corner or the center of patterns 9-10 in Fig. 1,b and be the SP in the multilevel picture if $|dd| > 0$. A set $\{x, y, dd\}$ of coordinates and double differences of SPs be called a difference representation (DR) of the picture. By sum of two pictures is meant a result of summing of their pixel's values and by sum of two DRs is meant a merging of SPs in these pictures. During this merging instead of some coinciding SPs only one SP with the same coordinates and total double difference must be introduced. It is true also that DR of the sum of two pictures is the sum of DRs of these pictures.

The algorithm for adding of two binary pictures v_1 and v_2 in the corner form consists of the following two parts: 1) summing of DRs of these two pictures and 2) line by line retrieval of the multilevel picture $v = v_1 + v_2$ from its difference representation; binarization of the picture v by the threshold equal to 1 and the corner coding of the result binary picture. In greater detail this algorithm will be considered in Section 2.

An effective algorithm for tracing the contour of the compressed picture and removing the objects with area (number of pixels) less than given threshold is described in [10]. The tracing of the contour is carried out in [10] during one pass scanning of the picture and deletion of the objects is fulfilled by removing their corners from initial data. Considered above examples prove that corner representation can be effectively used in different algorithms but of special interest is the development of the effective algorithms for dilation(erosion) and skeletonization of binary pictures as most time consuming and most often used in the practice.

Dilation of the picture v by mask image x_t with representative point t can be represented as $v_e = \bigcup_{t \subset v_b} x_t$, where v_b is a set of black pixels of the picture v. Usually the number of operations for dilation of the picture depends on the numbers of black pixels in the original and mask images. In Section 2 the effective algorithm for dilation of binary pictures in compressed corner form is described and computational complexity of this algorithm depends on the numbers of the corners in the pictures v and x_t.

Skeletonization [1-7] provides significant geometric features extracted from the binary image data. These features represent the lines that approximate the centers of so called maximal circles or squares, inscribed in the patterns of the picture under processing. Two main approaches are used for skeletonization of the binary pictures: the algorithms [2,5,7] of the first (thinning) approach use different variants of connected deletion operators for iteratively performed shrinking of elongated binary objects to thin line-patterns; on each iteration every boundary(contour) pixel is inspected and those pixels, that are not required to maintain connectivity or ends of lines, are erased. The algorithms [3,4,6] of the second (not iterative) approach define the skeleton points from distance measurements with respect to the opposite boundary points of the regions.

In Section 3 the modified version of the skeletonization algorithm [6] is described. The algorithm implements the second approach and defines the skeleton during one pass sequential scanning of the picture. The computational complexity of the

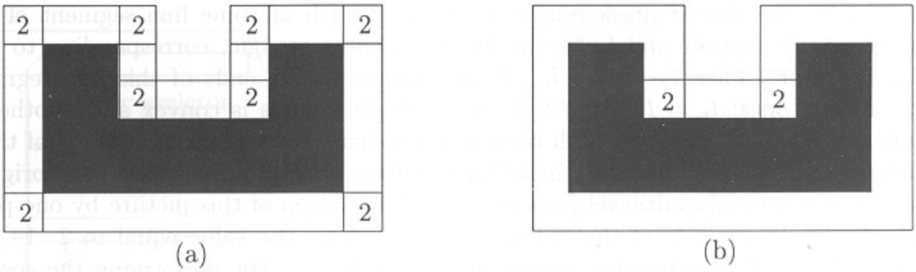

Fig. 2. Two multilevel presentations of the dilated picture.

algorithm depends on the number of the corners in the picture and does not depend, instead of [3], on the size of maximal squares in the picture. Section 4 contains some experimental results and Section 5 some concluding remarks.

2. Description of the dilation algorithm

Let us define corner line as a boundary of two (upper and lower) neighbouring picture's rows that contains two or more corners of this picture. The definition of the line of special places in the picture is similar. The algorithm for dilation of binary pictures consists of three main parts.

The first part of the algorithm consists in transformation of the CR of the picture into its DR, which is carried out during bottom-up and from the left to right scanning of corner lines in the picture. Let us suppose that the most left and bottom point in the picture has coordinates (0,0), and left column and bottom row of the picture are known. Let T_i be a set of points, ordered along some, say, i-th corner line, such that every point is either the corner or the end of a run of pixels (having the same value 0 (white) or 1 (black)) in the lower row of the picture. During processing of this corner line all points $t \subset T_i$ are successively considered and for every of these points the values ρ_1, ρ_2 of to right neighbouring pixels (Fig. 1,b) on the basis of values ρ_3, ρ_4, $|dd|$ and $d_2 = \rho_1 - \rho_4$ are defined: $\rho_1 = \rho_4 + d_2$ and $\rho_2 = 1$, if either a) $|dd| = 1$ and $\rho_1 + \rho_3 + \rho_4$ is the even value or b) $|dd| = 0$ and $\rho_1 + \rho_3 + \rho_4$ is the odd value, and $\rho_2 = 0$ otherwise. The retrieved values $\rho_1 - \rho_4$ are then used for defining parameters of SP (if any) in the current point t and also for run length coding of the upper row of the picture. Before processing of the next $(t+1)$-th point the values ρ_3, ρ_4 are set equal to ρ_1 and ρ_2 correspondingly, and during processing of the next $(i+1)$-th corner line the run length coding data of the upper row are used, being the retrieved data for the lower row during processing of the previous corner line.

The second part of the algorithm consists in retrieval of the DR of some multilevel picture v, which after thresholding by 1 represents the dilated original binary picture. Every connected component of the picture v can be usually represented as a sum of original binary picture's component with a some additional picture. Let us suppose that the mask image represents a black square and its representative point is a center of this square. In this case the additional picture is a strip, which can be represented as a sum of intersected horizontal and vertical rectangles (Fig. 2,a).

Let l_m be the size of mask image, l_c be the length of some line segment of the component's contour and l_r be the length of the rectangle, corresponding to this line segment. Then $l_r = l_c + l_m$, if the corners at the ends of this line segment are convex ones; $l_r = l_c + l_m/2$, if one of these corners is convex and another is concave one, and $l_r = l_c$, if both corners are concave ones.The DR of every of these rectangles consists of four SPs in vertices of this rectangle. In Fig. 2,a the original binary picture and multilevel presentation of expansion of this picture by one pixel are shown. The pixels of the multilevel picture have the value equal to 2=1+1 in intersections of neighbouring rectangles in this figure. For performing the second step of the algorithm it is necessary to define SPs of the additional picture and to merge these SPs with SPs of the initial picture.

Another way for presentation of the additional picture consists in summing pixel's values of every two rectangles, that correspond to the same concave corner, and in joining of all other rectangles. In Fig. 2,b an example of corresponding multilevel representation of dilated picture v is shown. In this figure the pixels in intersected regions of summing rectangles have also the value equal to 2. The DR of the picture v can be received by 1) moving of every SP, that corresponds to convex corner in the original picture, on the distance of mask image's diagonal in certain direction, corresponding to the type of this corner; 2) assigning to every SP: (x, y, dd_0), that corresponds to concave corner of the type t_q, new double difference value $dd_1 = -dd_0$ and introducing of two new SPs: $(x, y+y_1, dd)$ and $(x+x_1, y, dd)$, where a) $dd = -1$, $y_1 = l_m/2$, $x_1 = l_m/2$, if $t_q = 5$; b) $dd = 1$, $y_1 = -l_m/2$, $x_1 = l_m/2$, if $t_q = 6$; c)$dd = -1$, $y_1 = -l_m/2$, $x_1 = -l_m/2$, if $t_q = 7$; and d) $dd = 1$, $y_1 = l_m/2$, $x_1 = -l_m/2$, if $t_q = 8$; 3) ordering of all SPs and summing of the dd values for coinciding SPs. Therewith the SP of the type 9 or 10 (Fig. 1,b) must be considered as a pair of two SPs of the types 1 and 3 or 2 and 4.

Performing of the second part of the dilation algorithm depends in general on the mask image, and for every of these images the algorithm for retrieval of the DR of the multilevel representation of the dilated original image must be defined. For example, if the mask image represents a circle, then concave corners of the original picture will be processed as described above, but processing of every convex corner will result in his replacing on some group of the corners, representing correspondent quadrant of the circle's contour.

The third part of the algorithm is similar to the first one and consists in line by line retrieval of the picture v on the basis of its DR, thresholding of this picture by threshold equal to 1 and corner coding of the result binary picture. This step is carried out during bottom - up and from left to right scanning of the points along the lines of the SPs in the picture, such that every point p is either SP or the end of the run of pixels, having the same value in the lower row of the picture. During processing of every of these points the values ρ_1, ρ_2 of neighbouring pixels (Fig. 2) on the basis of values ρ_3, ρ_4, dd and d_2 are defined: $\rho_1 = \rho_4 + d_2$ and $\rho_2 = \rho_3 + dd + d_2$. The retrieved values $\rho_1 - \rho_4$ are then used for checking if there is the corner in every point p and also for run length coding of the upper row of the picture.

The algorithm of erosion of the binary picture is similar to the considered above algorithm. It can be represented also as dilation of the inversed initial picture, followed by inversion of the dilated picture.

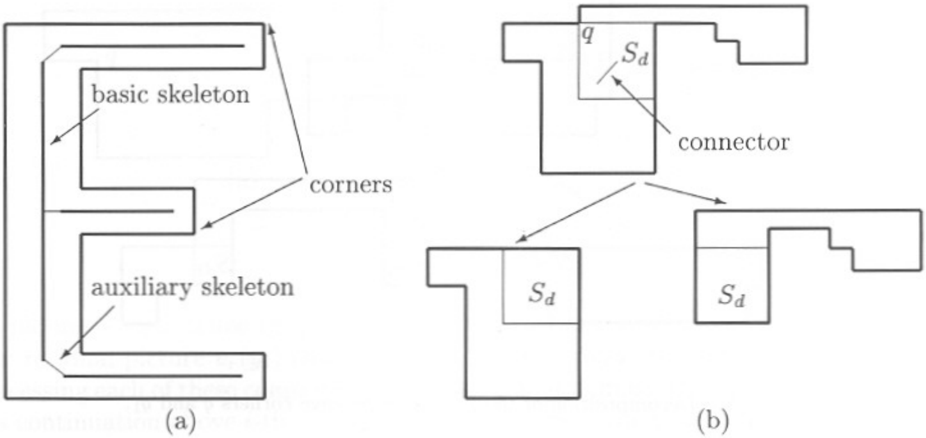

Fig. 3. (a) The basic and auxiliary skeletons of the binary picture. (b) Decomposition of the picture into two subpictures.

3. Description of the skeletonization algorithm

A certain square $S_m \subset R^2$ in the binary picture v is called maximal square [6] if S_m consists of black points and this square can not be increased without violating this feature. A basic skeleton of the picture v is defined as a set of central points of the maximal squares. The basic skeleton represents a set of horizontal and vertical lines, as well as some separate points in the picture and is shown in Fig. 3,a by thick lines. An auxiliary skeleton is a set of additional line segments, ensuring the connectivity of the produced skeleton, and is shown in Fig. 3,a by thin lines. Corner's coordinates represent the compressed form of the picture and also important data for skeleton detection, because to every such corner corresponds (in appropriate diagonal direction) the end of skeleton line segment.

A certain square S_d is called dissective square (Fig. 3,b) in the picture if (a) the vertice of the square S_d coincides with the concave corner q , (b) the diagonal of this square coincides with the bisectrix of the corner q and (c) the square S_d consists of black points and can not be increased without violating this feature.

Let us suppose that intersection $C_v \cap S_d$ belongs to the same connected component of the contour C_v of the picture v. It means, that there is no in the picture v the white component or pattern (the connected set of white pixels), such that its contour contains the concave corner q. In this case splitting the picture v into two connected pictures v_1 and v_2 (Fig. 3,b) is called decomposition if (a) $v_1 \cap v_2 = S_d$, (b) $v_1 \cup v_2 = v$ and (c) $v_1 \neq S_d$, $v_2 \neq S_d$. Connector is the line segment (Fig. 3,b) linking centers of those two maximal squares S_1 and S_2 such that $S_d \subset S_1 \subset v_1$ and $S_d \subset S_2 \subset v_2$. The following two statements are valid: (a) the basic skeleton of the picture v is the union of the basic skeletons of the pictures v_1, v_2; and (b) the auxiliary skeleton of the picture v is the union of auxiliary skeletons of the pictures v_1, v_2 and a third set containing only the connector. That is why skeletonization of the picture v can be reduced to decomposition the picture v into two pictures v_1 and v_2, followed by skeletonization of these two pictures.

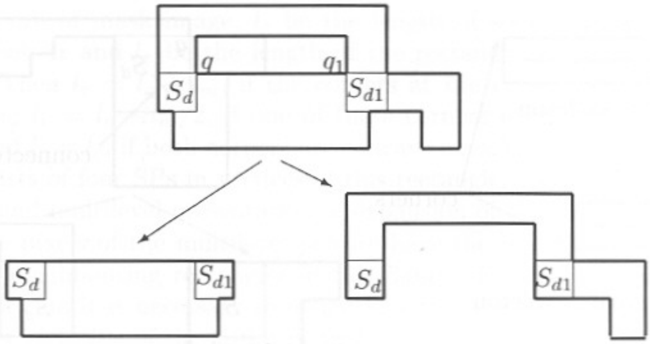

Fig. 4. Decomposition of the picture in concave corners q and q_1.

Otherwise if the picture v contains the white component with the concave corner $q(x, y)$ of the type 5, but below this corner there is no another corner of such type, the picture v has the concave corner $q_1(x_1, y)$ of the type 8, neighbouring to q, and decomposition of the picture v into two pictures can be carried out after defining dissective squares in the corners q and q_1 (Fig. 4).

Let us define also a notion of a residual picture. Let L be a horizontal line with ordinate y dividing the picture v into two parts. The operation of decomposition of lower part $v_b(y)$ of the picture is called permissible, if ordinate of upper side of dissective square, corresponding to this operation, is less than y. After carrying-out all possible decompositions of the picture $v_b(y)$ some set of rectangles below line L and also some set (it may be empty) of the parts of the picture touching to this line will be received. The union of these parts represents the residual picture $v_r(y)$ of the picture $v_b(y)$.

Some picture that does not contain the corners of types 5 and 8 be called a simple picture. At first let us consider the algorithm of skeletonization of the simple picture with the contour which does not intersect vertical sides of dissective squares corresponding to concave corners of types 6-7 in this picture [Fig. 5,a]. If the simple picture does not contain concave corners, then it is rectangle and its skeletonization is obvious. Otherwise the skeleton of the simple picture can be detected in the following manner. The concave corner q with the least coordinate x and the dissective square in this corner are detected. Then decomposition of the picture into two parts is carried out: one of these parts is the rectangle and another is the simple picture, which does not contain the concave corner q. These operations are repeated to complete removal of concave corners in this picture. The skeletonization algorithm of other simple pictures [Fig. 5,b] only somewhat differs from considered above. The computational complexity of skeletonization of any simple picture linearly depends on the number of corners in these pictures.

The algorithm for skeletonization of any (not only simple) picture processes the picture during its bottom-up scanning, so that the boundary of the scanned part successively occupies positions coinciding with corner lines in this picture. In doing so the algorithm processes some, say, i-th corner line with ordinate y_i using not

(a) (b)

Fig. 5. Simple pictures.

all picture $v_b(y_i)$ below of y_i , but only the residual part $v_r(y_i)$ of this picture. The residual picture $v_r(y_i)$ consists of one or some connected components. During processing each of these components $v_{rc}(y_i)$ it can be defined whether this component has continuation above i-th corner line or no. In the second case the picture $v_{rc}(yi)$ is the simple one and its decomposition must be carried out as considered above.

Otherwise the decomposition of the component $v_{rc}(y_i)$, correspondent either to the corner q of the type 5 or 8 with the least absciss and ordinate y_i or to two such neighbouring corners, is carried out. If there is no such corner in the picture $v_{rc}(y_i)$ this picture is joined to the residual picture $v_r(y_{i+1})$. Otherwise, two following pictures as a result of decomposition $v_{rc}(y_i)$ in this corner are produced. One of these pictures does not contain corners of types 5 and 8 and hence it is the simple picture, which must be skeletonized as described above. Another picture does not contain the corner q and it must be processed in the same way as the initial picture $v_{rc}(y_i)$: carrying out the decomposition in the next corner with the least absciss and so on.

4. Performance

The dilation(erosion) and skeletonization algorithms have been implemented in the software system for automatic input, processing and semantic interpretation of line drawings and maps. Fig. 6 shows a 1248x1700 original image on the left, its dilation (isotropic expansion) by two pixels on the middle and its skeletonization on the right. Size of the original picture is 270x170 mm. After scanning this picture with the resolution 200 dpi the volume of noncompressed TIFF file is 570 kbyte and the volume of corner representation of the picture is 97.1 kbyte. Dilation and skeletonization of the picture take 0.6 and 1.5 sec. on IBM PC Pentium 120 The volume of output (postprocessed) skeleton is 31.2 kbyte. The developed software processes the picture part by part and that is why it can be used for processing of large-size line drawings.

5. Concluding remarks

Fast dilation and skeletonization algorithms have been developed and presented in this paper. The proposed dilation algorithm is fast because its computational complexity depends on the number of corners in the picture and does not depend on the number of the pixels in this picture. This algorithm processes the picture line by line and requires small memory area for its performance.

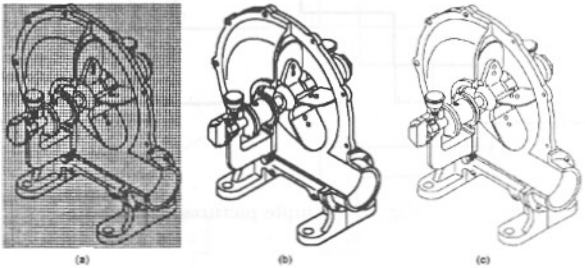

Fig. 6. Results of expansion by two pixels (b) and skeletonization (c) of the original picture (a).

Comparing developed skeletonization algorithm to algorithms [2-5] we find the following.

1. The proposed algorithm is essentially faster, taking in account, that different computers were used, in that it is a) width independent and b) it does not require each pixel of the pattern to be tested except only some points (corners) of its contour.
2. It is structure preserving and is oriented toward a structure-descriptive skeleton representation, because the set of detected maximal squares represents thin lines instead of isolated pixels or points.
3. It requires small memory area, which does not depend on the size of input data.

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-Dumn (MIT Press, Cambridge, 1967), pp. 362-380.
2. P.S.P. Wang and Y.Y. Zhang, "A fast and flexible thinning algorithm," IEEE Trans. Comput. 38(5), 742- 745 (1989).
3. T.Wakayama," A core-line tracing algorithm based on maximal square moving," IEEE Trans. Pattern Anal. Mach. Intell. PAMI-4(1), 68-74 (1982).
4. C. Arcelli and G.S. Di Baja, "A width-independent fast thinning algorithm," IEEE Trans. Pattern Anal. Mach. Intell. PAMI-7(4), 463-474 (1985).
5. S.V. Ablameyko, B.S. Beregov and A.N.Kryuchkov, "Computer-aided cartographical system for map digitizing," Proc. 2nd ICDAR, Tsukuba, Japan, pp. 115-118, 1993.
6. V.M.Kiyko and M.I.Schlesinger, "Width-independent fast skeletonization algorithm for binary pictures," Intern. Journal of Imaging Systems and Technology, vol. 3, 222-226(1991).
7. L.Lam, S-W. Lee, C.Y. Suen, "Thinning methodologies - A comprehensive survey", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 14, No. 9, Sept. 1992, pp. 869-885.
8. M.I.Schlesinger. Matematiceskie sredstva obrabotki izobrazenij, in Russian. (Mathematic tools for image processing). Naukova Dumka, Kiev, Ukraine, 1989.
9. V.Hlavac, J.Fojtik," Adaptive non-linear predictor for lossless image compression. Computer analysis of images and patterns," 7th Int. Conference, CAIP'97, 279-288, 1997.
10. V.V.Matsello, "The algorithm for noise removing in line drawings," in Russian. Avtomatica, No. 2, pp. 86-89, 1986.