Pergamon

Pattern Recognition, Vol. 28, No. 3, pp. 331 341, 1995
Elsevier Science Ltd

0031 3203/95 $9.50 + .00

0031-3203(94)00104-9

A SKELETONIZATION ALGORITHM BY MAXIMA TRACKING ON EUCLIDEAN DISTANCE TRANSFORM†

FRANK Y. SHIH‡ and CHRISTOPHER C. PU§

‡Computer Vision Laboratory, Department of Computer and Information Science, New Jersey Institute of Technology, Newark, NJ 07102, U.S.A.

§Department of Information Management, Ming Hsin Institute of Technology, Hsinchu, Taiwan, R.O.C.

(*Received* 24 *May* 1993; *in revised form* 7 *February* 1994; *received for publication* 22 *August* 1994)

Abstract—A simple and efficient algorithm using the maxima tracking approach on Euclidean distance transform to detect skeleton points is presented. The advantages of the skeleton obtained are: (1) connectivity preservation; (2) single-pixel in width; and (3) its locations as close as to the most symmetrical axes. Besides, the condition of the least slope change of skeleton is used to ensure the fairness of the digital medial axes. With the least effort, the algorithm can be modified to eliminate non-significant short skeletal branches originating from the object contour while the critical shape-informative medial axes are preserved.

Image representation Skeleton Medial axis transformation Distance transform
Pattern recognition Mathematical morphology

1. INTRODUCTION

The studies of distance transformation and skeletonization were motivated from the need of converting a digital image into a linear form in a natural manner. From the skeleton the contour of an object region can be easily regenerated, so that the amount of information involved in image analysis is the same. Besides, the skeleton emphasizes certain properties of the image; for instance, curvatures of the contour correspond to topological properties of the skeleton. The concept of skeleton was first proposed by Blum.[1] Skeleton, medial axis, or symmetrical axis has been extensively used for characterizing objects satisfactorily using the structures which are composed of line or arc patterns. Applications include the representation and recognition of handwritten or printed characters, fingerprint ridge patterns, biological cell structures, circuit diagrams, engineering drawings and the like.

Let us visualize a connected object region as a field of grass and place a fire starting from its contour. Assume that this fire burning spreads uniformly in all directions. The *skeleton* (or *medial axis*, abbreviated MA) is where waves collide with each other in a frontal or circular manner.[1] It receives the name of medial axis because the pixels are located at midpoints or along local symmetrical axes of the region. The points in MA (or skeleton) bear the same distance from at least two contour points which is the smallest among those computed from all background points. The information on the sizes of local object region is retained by associating each skeleton point with a label representing the aforementioned distance value. Intuitively, the original binary image can be reconstructed by a union of the circular neighborhoods centered at each skeleton point with a radius equal to the associated label.

A skeleton will not appear immediately in the wavefront if it is a smooth curve locally. The appearance of a skeleton starts with the minimum radius of curvature in a local contour. The disappearance of a skeleton is encountered when the largest circle can be drawn within a local object region. If the boundary has a regional concave contour, the wave-fronts will not collide with each other. Thus, there exists no skeleton points located surrounding this area. In fact, the skeleton will originate outside the regional concave contour if the skeletonization is considered to deal with the background of the connected object, i.e. the complement of the image.

Many algorithms have been developed to extract the skeleton. The straight-forward procedure to accomplish such a transformation involves an iterative process which shrinks the object region step-by-step until a one-element thick figure is obtained.[2] Naccache and Shinghal[3] proposed the strategy: visiting all the pixels in the bitmap to iteratively delete the edge points which are classified as non-safe points (i.e. the points being deleted without the effectness of connectivity). Zhang and Suen[4] proposed a parallel thinning algorithm which consists of two subiterations: one is to delete the south-east boundary points and the north-west corner points, and the other is to delete the north-west boundary points and the south-east corner points.

†This work was supported by the National Science Foundation under Grant IRI-9109138 and the New Jersey Institute of Technology under Grant 421770.

Several algorithms have aimed to achieve more efficient parallel thinning algorithms for skeletonization.[5-11] Kwok[10] proposed a thinning algorithm using contour generation in terms of chain codes. The advantage of this algorithm is that only the contour tracking is performed in each iteration. Arcelli and Sanniti di Baja[12] used the city-block distance transform to detect the skeleton. Rather than peeling, the skeleton is identified as multiple pixels based on the multiplicity defined, and the recursive procedure calls are applied. An algorithm for generating connected skeletons from approximated Euclidean distances and allowing near reconstruction was developed by Niblack *et al.*[13] Jang and Chin[9] proposed a one-pass parallel thinning algorithm based on a number of criteria including connectivity, unit-width convergence, medial axis approximation, noise immunity and efficiency. Maragos and Schafer[14] used morphological operations to extract the skeleton and to optimize the skeleton for the purpose of image coding. Their morphological skeleton is however not connected and more than 1-pixel wide. Jang and Chin[8] defined digital skeletons based on morphological set transformation and provided the proof of convergence, the condition for 1-pixel thick skeletons and the connectedness of skeletons.

This paper intends to present a simple and efficient algorithm using the maxima tracking approach on Euclidean distance transform to detect skeleton points. The advantages of the skeleton obtained are: (1) connectivity preservation; (2) single-pixel in width; and (3) its locations as close as to the most symmetrical axes. This paper is organized into five parts. In Section 2, the distance transformation using the mathemati- cal morphology approach is briefly introduced. In Section 3, a procedure of thick skeleton generation is described. In Section 4, the basic definitions such as base points, apex points, directional-uphill generation and directional-downhill generation are given. In Section 5, the skeletonization algorithm is presented and its connectivity properties are proved. In Section 6, a slightly modified algorithm is described to eliminate non-significant skeletal branches.

2. DISTANCE TRANSFORMATION

The distance transformation is to convert a binary image, which consists of object (foreground) and non-object (background) pixels, into another image where each object pixel has a value corresponding to the minimum distance from the background. Three types of distance measure for digital images are often used: *city-block*, *chessboard* and *Euclidean*. City-block and chessboard distances are easy to compute by recursively accumulating the distance values in 3×3 neighborhood. A distance transformation algorithm uses an iterative operation by peeling off the border pixels and summing the distances layer-by-layer.[15,16] This is efficient on cellular array computers. Another algorithm only needs two operations in opposite scans of the picture: one scan is in the left-to-right top-to-bottom direction, and the other is in the right-to-left bottom-to-top direction.[16,17] This does not require iterations and is efficient on conventional computers. Other algorithms apply the morphological erosions.[18-20]

We achieve the distance transformation by adopting the mathematical morphology approach which is briefly reviewed next. Mathematical morphology, which is based on set-theoretic concepts, can extract object features by choosing a suitable structuring shape as a probe.[18] The morphological operations deal with two images: one being processed is referred to as the *active image*, and the other being a kernel is referred to as the *structuring element*. With the help of various designed structuring shapes, we can simplify the image data and expose the object shape characteristics. Let F and K be the domains of the gray-scale image f and the gray-scale structuring element k, respectively. The *gray-scale dilation* can be computed by

$$(f \oplus k)(x, y) = \max_{(x-m, y-n) \in F, (m,n) \in K} \{f(x-m, y-n) + k(m, n)\}. \qquad (1)$$

The *gray-scale erosion* can be computed by

$$(f \ominus k)(x, y) = \min_{(x+m, y+n) \in F, (m,n) \in K} \{f(x+m, y+n) - k(m, n)\}. \qquad (2)$$

A binary image f consists of two classes: object pixels (foreground) and non-object pixels (background). The initialized representation of f is: the object pixels are denoted by value '$+\infty$' (or any number sufficiently greater than the object's largest distance value) and the background pixels by value '0'. Let k be the 3×3 structuring element as

$$k_{\text{city-block}} = \begin{matrix} -2 & -1 & -2 \\ -1 & 0 & -1 \\ -2 & -1 & -2 \end{matrix},$$

$$k_{\text{chessboard}} = \begin{matrix} -1 & -1 & -1 \\ -1 & 0 & -1 \\ -1 & -1 & -1 \end{matrix}.$$

The distance transformation algorithm using the morphological erosion is described below. For details refer to reference (18).

2.1. *Distance transformation algorithm*

(1) Perform $d = f \ominus k$.

(2) Repeat (1) until the output remains unchanged or all '$+\infty$' are removed.

The city-block or chessboard distance measures are sensitive to the rotations of an object, but the Euclidean distance measure is rotation invariant. However, its square root operation is costly and the global nature of distance transformation is difficult to decompose into small neighborhood operations because of the nonlinearity of Euclidean distance variations. Hence,

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	**10**	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	**10**	**10**	-
-	**10**	**20**	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	**20**	**10**	-	-
-	10	**20**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	**30**	**30**	**22**	**14**	**10**	-	-
-	10	20	**30**	**40**	**36**	28	22	20	20	22	28	**36**	**40**	**40**	40	40	40	40	40	**40**	**40**	**36**	**28**	20	10	-	-	-
-	10	20	30	**36**	28	22	14	10	10	14	22	28	36	**45**	**50**	50	50	50	**50**	**45**	36	28	**22**	**14**	**10**	-	-	-
-	10	20	**30**	**32**	22	14	10	-	-	10	14	22	32	41	**51**	**60**	**60**	**60**	**51**	41	32	22	14	10	-	-	-	-
-	10	20	**30**	**30**	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	**60**	50	40	30	20	10	-	-	-	-	-
-	10	20	**30**	**30**	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	**61**	51	41	32	22	14	10	10	**10**	**10**	-
-	10	20	**30**	**32**	22	14	10	-	-	10	14	22	32	41	**51**	**60**	**60**	**60**	**54**	45	36	28	**22**	**20**	**20**	**20**	**10**	-
-	10	20	30	**36**	28	22	14	10	10	14	22	28	36	**45**	**50**	50	50	50	**50**	**45**	36	28	**22**	**20**	**20**	**20**	**10**	-
-	10	20	**30**	**40**	**36**	28	22	20	20	22	28	**36**	**40**	**40**	40	40	40	40	40	**40**	**32**	22	14	10	10	**10**	**10**	-
-	10	**20**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	**30**	**20**	10	-	-	-	-	-
-	**10**	**20**	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	**20**	**10**	-	-	-	-	-
-	**10**	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	**10**	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 1. The Euclidean distance function with the ridge points are in bold-face and underlined. For expressional simplicity, we use ten times the Euclidean distance in rounded numbers.

the algorithms concerning the approximation of Euclidean distance transformation have been extensively discussed.[4,15,21,22] Borgefors[15] optimized the local distances obtained within the 3×3, 5×5 and 7×7 neighborhoods by minimizing the maxima of the absolute values of the difference between the proposed technique and the Euclidean distance transformation. A mathematical morphology approach to perform the Euclidean distance transformation by the decomposition of global operation into local operations has been developed by Shih and Mitchell.[20] The decomposition technique employs a set of 3×3 morphological erosions with suitably weighted structuring elements and combines the outputs with a minimum operation. An example of the resulting Euclidean distance transformation is shown in Fig. 1.

3. THICK SKELETON GENERATION

For any pixel P, we associate a set of disks or circles with various radii centered at P. Let C_P be the largest disk which is completely contained in the object region, and r_P be the radius of C_P. There may exist another pixel Q such that C_Q contains C_P. If no such Q exists, C_P is called a *maximal disk*. The set of the centers and radii (or Euclidean distance values) of these maximal inscribed disks is called the skeleton.

3.1. *The skeleton from distance function*

The distance function of pixel P in the object region S is defined as the smallest distance of P from all the background pixels $\bar{S}$. That is

$$D(P) = \min_{Q \in \bar{S}} \{d(P, Q)\}, \tag{3}$$

where $d(P, Q)$ is the Euclidean distance between P and Q. If we visualize the distance values as the altitude on a surface, then the 'ridges' of the surface constitute the skeleton in which a tangent vector cannot be uniquely defined. It can be easily derived that if and only if a point P belongs to the skeleton of an object region S the maximal disk with the radius $D(P)$ hits the contour of $\bar{S}$ at least two places. Let us define the set

$$A(P) = \{Q \mid d(P, Q) = D(P), Q \in \bar{S}\}. \tag{4}$$

If the set $A(P)$ contains more than one element, then P is a skeleton point.

3.2. *Detection of ridge points*

Considering two neighbors of P in anyone of the horizontal, vertical, 45°-diagonal, and 135°-diagonal directions, if the altitude of P is higher than one neighbor and not lower than the other, then P is called the *ridge point*. Figure 1 shows the Euclidean distance transform

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	**10**	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	**10**	**10**	-
-	**10**	**20**	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	**20**	**10**	-	-
-	10	**20**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	**30**	**30**	**22**	**14**	**10**	-	-
-	10	20	**30**	**40**	**36**	28	22	20	20	22	28	**36**	**40**	**40**	40	40	40	40	40	**40**	**40**	**36**	**28**	**20**	10	-	-	-
-	10	20	30	**36**	28	22	14	10	10	14	22	28	36	**45**	**50**	50	50	50	**50**	**45**	**36**	**28**	**22**	**14**	**10**	-	-	-
-	10	20	**30**	**32**	22	14	10	-	-	10	14	22	32	41	**51**	**60**	**60**	**60**	**51**	**41**	**32**	22	14	10	-	-	-	-
-	10	20	**30**	**30**	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	**60**	**50**	**40**	30	20	10	-	-	-	-	-
-	10	20	**30**	**30**	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	**61**	**51**	**41**	**32**	22	14	10	10	**10**	**10**	-
-	10	20	**30**	**32**	22	14	10	-	-	10	14	22	32	41	**51**	**60**	**60**	**60**	**54**	**45**	**36**	**28**	**22**	**20**	**20**	**20**	**10**	-
-	10	20	30	**36**	28	22	14	10	10	14	22	28	36	**45**	**50**	50	50	50	**50**	**45**	**36**	**28**	**22**	**20**	**20**	**20**	**10**	-
-	10	20	**30**	**40**	**36**	28	22	20	20	22	28	**36**	**40**	**40**	40	40	40	40	40	**40**	**32**	**22**	14	10	10	**10**	**10**	-
-	10	**20**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	**30**	**20**	10	-	-	-	-	-
-	**10**	**20**	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	**20**	**10**	-	-	-	-	-
-	**10**	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	**10**	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 2. The ridges points and their trivial uphill generation are in bold-face and underlined. It is indeed connected but too thick.

with ridge points as bold-faced and underlined. Note that the result is not connected.

3.3. *Trivial uphill generation*

The trivial uphill of a point P is the set of all the neighbors with higher altitude. Figure 2 shows the resulting skeleton if we add the uphill of the ridge points, and continuously add the uphill of the new skeleton points until no further uphill can be generated. Finally, a connected skeleton can be obtained but a thick one. In order to make th skeleton to be single-pixel wide, we need get rid of the loose definition of ridge points and take into account the directional-neighborhood.

4. BASIC DEFINITIONS

Prior to the discussion of our algorithm, the basic definitions of base point, apex point, directional-uphill generation and directional-downhill generation are given below.

4.1. *Base point*

The base point is defined as a corner point which has the distance value 1 and is surrounded by a majority of background points. It belongs to one of one of the following three configurations as well as their variations up to eight 45°-rotations, respectively:

$$\begin{matrix}1 & 0 & 0\\ 1 & \underline{1} & 0\\ 0 & 0 & 0\end{matrix}\ ,\quad \begin{matrix}1 & 1 & 0\\ 1 & \underline{1} & 0\\ 0 & 0 & 0\end{matrix}\quad \text{and}\quad \begin{matrix}1 & 1 & 1\\ 1 & \underline{1} & 0\\ 0 & 0 & 0\end{matrix}\ ,$$

where the central pixels underlined represent the 45°, 90° and 135° corner points, respectively. In digital image, only the above three degrees are considered in a local 3×3 window.

It is obvious if all the three configurations are considered as base points, then more non-significant short skeletal branches are produced. In other words, an approach based on the skeletal branches originating from all corner points in a 3×3 window would lead to unmanageable complexity in the skeletal structure. Therefore, an appropriate shape-informative skeleton should reach a compromise among the representativity of the connected object structure, the required reconstructivity, and the cost of deleting the non-significant branches.

In this paper, the sharp convexities are considered as more significant. In detail, if the amplitude of the angle formed by two intersecting wave-fronts of the fireline is viewed as the parameter characterizing the sharpness, the values smaller than or equal to 90° are identified as suitable convexities. More strictly, only 45°-convexities are detected as base points. Using these points as the source to grow up the skeleton, the remaining procedures are acquired to preserve the skeletal connectedness.

4.2. *Apex point*

The apex point is the one being the local maxima in its 3×3 neighborhood. Note that the local-maximum pixels only construct a small portion of a disconnected skeleton. They occur as 45°-corner points or interior elements which have the highest altitude locally. The 45°-corner points also serve as base points. The base points and apex points are considered as sources or elementary cells of the skeleton which grows up emitting from them. Figure 3 shows the Euclidean distance function with the base points labeled '**B**' and the apex points are bold-face and underlined.

4.3. *Directional-uphill generation*

The set of points $\{P_i\}$ is called the *directional-neighborhood* of P, denoted by D_P, if they are in the 8-neighborhood of P and located within $\pm 45°$ slope changes from the current medial axis orientation of P. For example, using the 8-neighbors labeled $P_1, P_2, \ldots,$ and P_8 counterclockwise from the positive x-axis of P, if P_7 and P are the skeleton points, the points $P_2, P_3,$ and P_4 are the directional-neighbors of P, i.e. $D_P = \{P_2, P_3, P_4\}$. Several cases are illustrated below.

$$\begin{matrix}P_4 & P_3 & P_2\\ . & P & .\\ . & P_7 & .\end{matrix}\ ,\quad \begin{matrix}. & . & P_2\\ P_5 & P & P_1\\ . & . & P_8\end{matrix}\ ,\quad \begin{matrix}P_4 & P_3 & .\\ P_5 & P & .\\ . & . & P_8\end{matrix}\ ,\quad \begin{matrix}. & P_3 & P_2\\ . & P & P_1\\ P_6 & . & .\end{matrix}$$

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	B	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	B	-
-	10	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	10	-	-
-	10	20	30	30	30	30	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	30	30	30	30	22	14	10	-	-
-	10	20	30	**40**	36	28	22	20	20	22	28	36	40	40	40	40	40	40	40	40	40	36	28	20	10	-	-	-
-	10	20	30	36	28	22	14	10	10	14	22	28	36	45	50	50	50	50	50	45	36	28	22	14	10	-	-	-
-	10	20	30	32	22	14	10	-	-	10	14	22	32	41	51	60	60	60	51	41	32	22	14	10	-	-	-	-
-	10	20	30	30	20	10	-	-	-	-	10	20	30	40	50	60	**70**	60	50	40	30	20	10	-	-	-	-	-
-	10	20	30	30	20	10	-	-	-	-	10	20	30	40	50	60	**70**	61	51	41	32	22	14	10	10	10	B	-
-	10	20	30	32	22	14	10	-	-	10	14	22	32	41	51	60	60	60	54	45	36	28	22	20	**20**	**20**	10	-
-	10	20	30	36	28	22	14	10	10	14	22	28	36	45	50	50	50	50	50	45	36	28	22	20	**20**	**20**	10	-
-	10	20	30	**40**	36	28	22	20	20	22	28	36	40	40	40	40	40	40	40	40	32	22	14	10	10	10	B	-
-	10	20	30	30	30	30	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	30	30	20	10	-	-	-	-	-
-	10	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	10	-	-	-	-	-
-	B	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	B	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 3. The Euclidean distance function with the base points labeled '**B**' and the apex points are in bold-face.

Note that the set of directional-neighborhood contains always three elements. The directional-uphill generation is to add the point which is the maximum of P and its directional-neighborhood. That is to say

$$P^U_{\text{next}} = \max_{P_i \in D_P \cup \{P\}} \{P_i\}. \tag{5}$$

Figure 4 shows the result of directional-uphill generation of Fig. 3.

4.4. *Directional-downhill generation*

From Fig. 4, we observe that there should exist a vertical path connecting two apex points of value '**40**' in bold-face and underlined on the left side, in which the altitude changes are not always increasing; instead they are a mixture of decreasing and increasing values. The directional-downhill generation which is similar to the directional-uphill generation except the maxima tracking of the set excluding the central point P, is used to produce this type of skeletal branch. That is to say.

$$P^D_{\text{next}} = \max_{P_i \in D_P} \{P_i\}. \tag{6}$$

The directional-downhill generation is initialized from the apex points which cannot be further tracked by the directional-uphill generation. Hence the altitude of the next directional-downhill should be lower in the begining. However, the tracking procedure is continued without taking into account the comparison of neighbors with P; the next directional-downhill altitude could be lower or even higher until a skeleton point appears in the directional neighborhood. Figure 5 shows the result of directional-downhill generation of Fig. 4. The skeleton is now connected and possesses single-pixel in width except for 2 pixels having the same local maximum of Euclidean distance.

5. THE SKELETONIZATION ALGORITHM AND CONNECTIVITY PROPERTIES

The skeletonization algorithm traces the skeleton points by choosing the local maxima on the Euclidean distance transform and takes into consideration of the least slope changes in medial axes. The algorithm is described as follows:

(1) The base points and apex points are detected as the initial skeleton points.

(2) Starting with these skeleton points, the directional-uphill generation in equation (5) is used to add more skeleton points, and continuously add the directional-uphill of the new skeleton points until no further point can be generated. The points which cannot be further tracked are marked.

(3) Starting with the marked points, the directional-downhill generation in equation (6) is used to complete the skeleton tracking.

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	**10**	-
-	10	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	10	-	-
-	10	20	**30**	30	30	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	30	**30**	**22**	14	10	-	-	-
-	10	20	30	**40**	**36**	28	22	20	20	22	28	**36**	**40**	40	40	40	40	40	40	40	**40**	**36**	28	20	10	-	-	-
-	10	20	30	**36**	28	22	14	10	10	14	22	28	36	**45**	50	50	50	50	50	**45**	36	28	22	14	10	-	-	-
-	10	20	30	**32**	22	14	10	-	-	10	14	22	32	41	**51**	60	60	60	**51**	41	32	22	14	10	-	-	-	-
-	10	20	30	**30**	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	**60**	50	40	30	20	10	-	-	-	-	-
-	10	20	30	**30**	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	61	51	41	32	22	14	10	10	10	**10**	-
-	10	20	30	**32**	22	14	10	-	-	10	14	22	32	41	**51**	60	60	**60**	**54**	**45**	**36**	**28**	**22**	**20**	**20**	**20**	10	-
-	10	20	30	**36**	28	22	14	10	10	14	22	28	36	**45**	50	50	50	50	**50**	45	36	28	22	20	**20**	**20**	10	-
-	10	20	30	**40**	**36**	28	22	20	20	22	28	**36**	**40**	40	40	40	40	40	40	**40**	32	22	14	10	10	10	**10**	-
-	10	20	**30**	30	30	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	**30**	20	10	-	-	-	-	-
-	10	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	10	-	-	-	-	-
-	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 4. The results of directional-uphill generation of Fig. 3.

-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	**10**	-
-	10	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	10	-	-
-	10	20	**30**	30	30	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	30	**30**	**22**	14	10	-	-	-
-	10	20	30	**40**	**36**	28	22	20	20	22	28	**36**	**40**	40	40	40	40	40	40	40	**40**	**36**	28	20	10	-	-	-
-	10	20	30	36	28	22	14	10	10	14	22	28	36	**45**	50	50	50	50	50	**45**	36	28	22	14	10	-	-	-
-	10	20	30	32	22	14	10	-	-	10	14	22	32	41	**51**	60	60	60	**51**	41	32	22	14	10	-	-	-	-
-	10	20	30	30	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	**60**	50	40	30	20	10	-	-	-	-	-
-	10	20	30	30	20	10	-	-	-	-	10	20	30	40	50	**60**	**70**	61	51	41	32	22	14	10	10	10	**10**	-
-	10	20	30	32	22	14	10	-	-	10	14	22	32	41	**51**	60	60	**60**	**54**	**45**	**36**	**28**	**22**	**20**	**20**	**20**	10	-
-	10	20	30	36	28	22	14	10	10	14	22	28	36	**45**	50	50	50	50	**50**	45	36	28	22	20	**20**	**20**	10	-
-	10	20	30	**40**	**36**	28	22	20	20	22	28	**36**	**40**	40	40	40	40	40	40	**40**	32	22	14	10	10	10	**10**	-
-	10	20	**30**	30	30	**30**	**30**	**30**	**30**	**30**	**30**	30	30	30	30	30	30	30	30	30	**30**	20	10	-	-	-	-	-
-	10	**20**	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	**20**	10	-	-	-	-	-
-	**10**	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10	**10**	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fig. 5. The results of directional-downhill generation of Fig. 4.

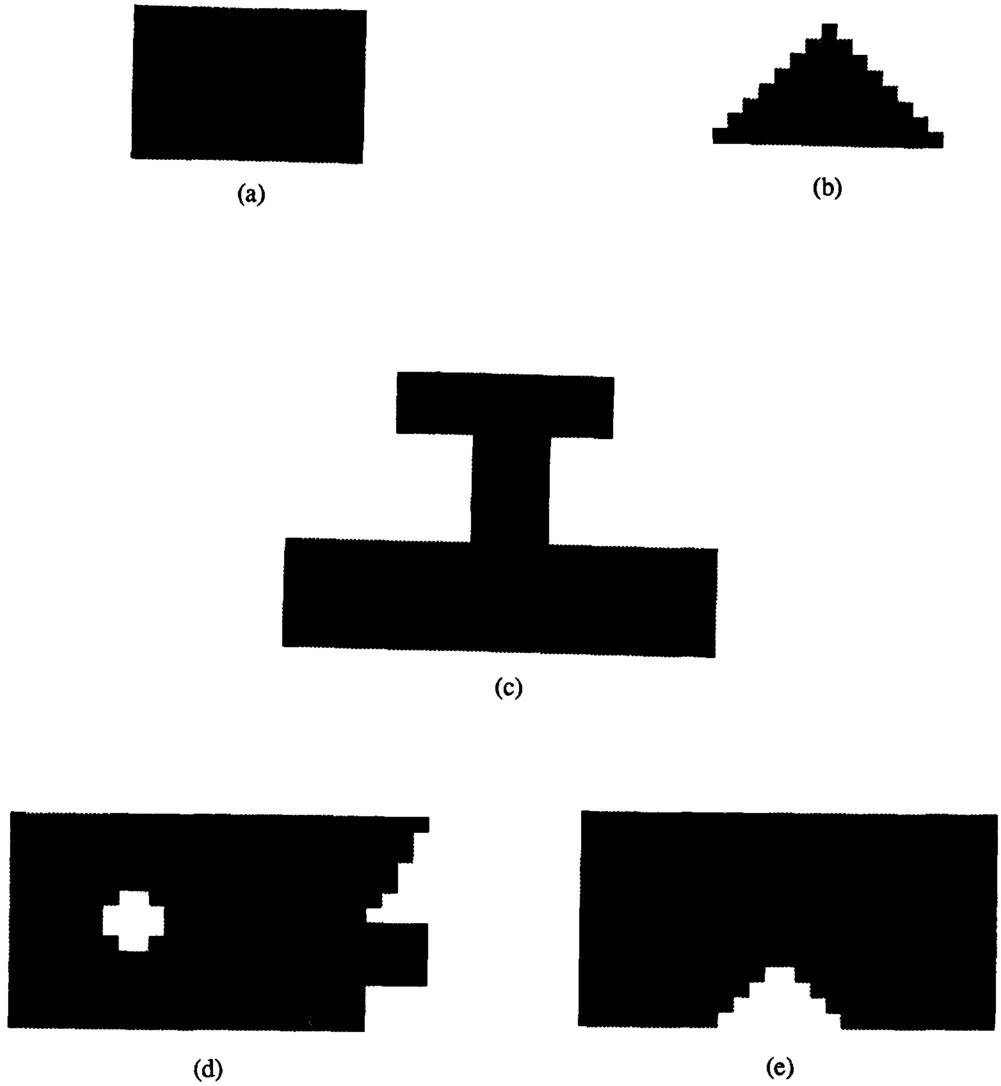

Fig. 6. The resulting skeletons by using the algorithm in Section 5 are illustrated: (a) rectangle, (b) triangle, (c) airplane-like model, (d) and (e) rectangle with holes and notches.

Figure 6 illustrates five skeletons obtained by the skeletonization algorithm. In our experiments, the 8-connectedness is applied for foreground and the 4-connectedness for background. The skeletonization algorithm possesses the following three connectivity properties:

(C1) After skeletonization, an originally connected object is not disconnected into two or more sub-objects.

(C2) After skeletonization, a connected object does not disappear at all.

(C3) After skeletonization, these originally disconnected background components are not 4-connected.

The proof for the above three properties is given as follows:

Proof of C1. *Induction hypothesis*: for any $m (m < n)$ marked apex points of an object, the skeleton obtained will preserve the property C1, where n could be any number.

The base case is $m = 1$. The algorithm starts tracking from each of the base and the apex points, and it will stop when the current point is connected with another skeleton point or touches with the point marked. That means every sub-skeleton branch starting from the base point will connect with some skeleton point or meet with another branch at the marked apex point.

When $m = 2$, it is regarded as two sub-objects, each sub-object contains a marked apex point. As discussed above, each sub-object is 8-connected. The algorithm will trace from each marked-apex and stop when it is connected with some skeleton points contained in the other skeleton subset. The reason that it would not connect with a skeleton point which belongs to the same skeleton subset is the directional-neighbors we use. Using directional-neighbors will enforce the tracking direction to go towards the region which has not been tracked up to now. Finally, it will lead tracking toward another skeleton subset.

As the induction hypothesis claim, when $m = n$, the skeleton obtained is 8-connected. Now consider the case when $m = n + 1$. Those $n + 1$ marked-apex points can be regarded as two sub-objects: one contains n marked-apex points and the other contains only a markedapex [i.e. the $(n + 1)$th point]. As discussed above, the one with n marked-apex points is 8-connected, and the other with a marked-apex is also 8-connected. Tracking from the $(n + 1)$th marked-apex, it will lead the tracking to go towards the skeleton subset with n marked-apex points. That means the whole object is 8-connected.

QED

Proof of C2. Tracking starts from each base and each apex point, the algorithm marks base points as the skeleton points. For any size of an object, there must be at least one base point or at least one apex point, and the skeleton obtained must have at least one point. For the ideal case of a circle, there is no base point but one apex point is present which is the local maximum representing the center of the circle. That is after skeletonization, an 8-connected object will not disappear at all, and the skeleton at least contains a pixel.

QED

Proof of C3. According to the definition of the apex points, there exists at least one apex in the object between any two background components. As the algorithm is performed, it will trace from each apex point which is not 8-connected to any skeleton point. After tracking, there will be one skeleton branch that will make these two background components disconnected. Therefore the skeleton obtained will not allow the originally disconnected background components to be 4-connected.

The algorithm including three procedures and two functions is described in pseudo-codes below.

```
procedure SKEPIK
  /* trace starting from each base and each apex point */
  for i, j in 1···N, 1···N loop
    if BASE_APEX (p(i, j)) then
      Uphill-Generation (p(i, j), p(i, j));
    end if
  end loop
  /* trace from each marked apex point in the procedure UpHill-Generation */
  while (marked pixel)
    DownHill-Generation (marked_apex, marked_apex);
  end while
end SKEPIK
function APEX (p): Boolean
/* apex point is the local maximum point */
  if (p is local maxima) then
    return TRUE
  else
    return FALSE;
  end APEX
function BASE_APEX (p): Boolean
/* base point is the point with distance 1 and has 4 or more zeros in its 8-neighborhood */
  if (distance of p = 1) then
    find 8-neighbors of p;
    if (number of 8-neighbors with distance 0) ≥ 4 then
      return TRUE;
    else
      if APEX (p) then
        return TRUE;
      else
        return FALSE;
      end if
    end if
  end if
end BASE_APEX
procedure Uphill-Generation (current-skeleton-pixel, previous-skeleton-pixel)
  if (number of maximum in 8-neighbors > 1 and
    distance of maximum ≥ distance of current-skeleton-pixel) then
      maximum-pixel = the maximum of the directional-neighbors;
      UpHill-Generation (maximum-pixel, current-skeleton-pixel);
```

```
  else
      if (number of maximum in 8-neighbors = 1 and
         distance of maximum ≥ distance of current-skeleton-pixel) then
         maximum-pixel = the maximum;
         UpHill-Generation (maximum-pixel, current-skeleton-pixel);
      else
         mark-apex; /* mark current-skeleton-pixel for later processing */
      end if
    end if
end UpHill-Generation
procedure DownHill-Generation (current-skeleton-pixel, previous-skeleton-pixel)
  maximum-pixel = the maximum of the directional-neighbors;
  if (maximum-pixel is not a skeleton point) then
    DownHill-Generation (maximum-pixel, current-skeleton-pixel);
  end if
end DownHill-Generation
```

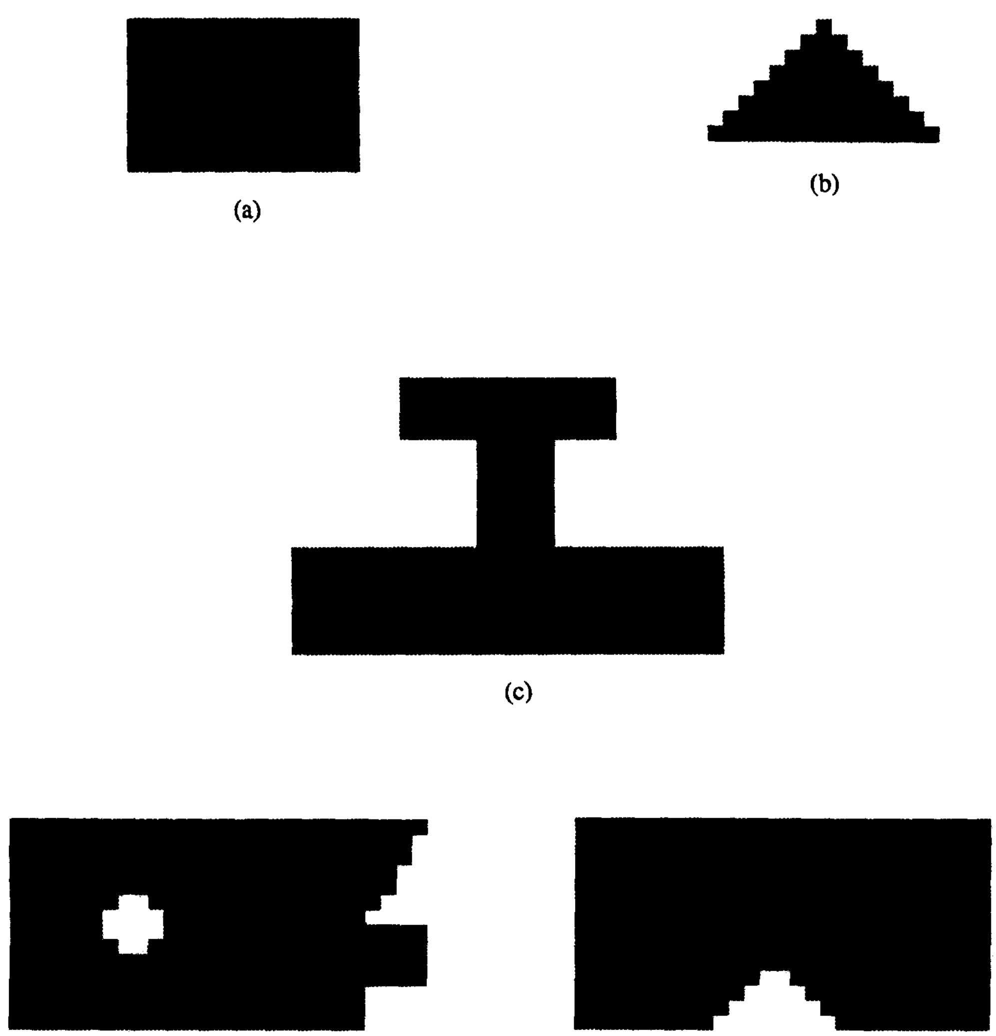

Fig. 7. The resulting skeletons by using the modified algorithm in Section 6 are illustrated: (a) rectangle, (b) triangle, (c) airplane-like model, (d) and (e) rectangle with holes and notches.

6. A MODIFIED ALGORITHM

The modified maxima tracking algorithm will extract the skeleton by eliminating non-significant short skeletal branches which touch the object boundary at corners. It is different from the previous algorithm that detects the base and apex points as initial skeleton points. Instead the maxima tracking starts from the apex points only. The algorithm will recursively repeat the same procedure which selects the maxima in the directional-neighborhood as the next skeleton point until another apex points are reached.

The modified maxima tracking algorithm is given as follows:

(1) The apex points are detected as the initial skeleton points.

(2) Starting with each apex point, we use the directional-uphill generation to generate the skeleton points. Recursively repeat this procedure until an apex point is reached, then the apex point is marked.

(3) Starting with these marked apex points, the directional-downhill generation is used to track the new skeleton points.

Figure 7 illustrates the resulting skeletons by using the modified. As shown in Fig. 7(a), a skeleton of a straight line can be interpreted as a rectangular shape. In Fig. 7(b), a triangle can be represented as a staircase. The distance value of the toppest stair is the radius of the largest enclosed circle in this triangle. Figure 7(c) is a airplane-like shape which is composed of three rectangles. In Fig. 7(d) ther is a closed curve which indicates a hole inside. The skeleton in Fig. 7(e) can be represented by a bent curve which gives us the information that there is a notch.

An example of the skeleton obtained for a wrench is shown in Fig. 8. Figure 9 illustrates that the character 'e' and its rotations by 30° and 45° using the modified algorithm will produce the identical rotated skeleton provided the digitization error is disregarded.

Fig. 8. An example of the modified skeleton for a wrench.

(a)

(b)

(c)

Fig. 9. An example of the modified skeleton for a set of the character 'e' in various rotations. (a) 0°, (b) 30°, and (c) 45°.

7. CONCLUSIONS

A simple and efficient skeletonization algorithm using the maxima tracking technique on the Euclidean distance transform has been presented in this paper. This algorithm can be flexibly modified to delete the non-significant skeletal noises while the primary structure is preserved. The Euclidean distance transform whose attributes to the skeleton has the most equidistance property analogous to the definition for continuous images and possesses the rotation invariance property.

REFERENCES

1. H. Blum, A transformation for extracting new descriptors of shape, in *Models for the Perception of Speech and Visual Forms* (edited by Walthen Dunn), Proc. of Meeting held November 1964. pp. 362–380 MIT Press, Massachusetts (1967).
2. L. O' Groman, K × K thinning, *Computer Vision, Graphics and Image Processing* **51**, 195–215 (1990).
3. N. J. Naccache and R. Shinghal, SPTA: a proposed algorithm for thinning binary patterns, *IEEE Trans. Systems, Man and Cybernetics* **SMC-14**(3), 409–418 (May 1984).
4. T. Y. Zhang and C. Y. Suen, A fast thinning algorithm for thinning digital patterns, *Comm. ACM* **27**(3), 236–239 (March 1984).
5. Z. Guo and R. W. Hall, Parallel thinning with two-subiteration algorithms *Comm. ACM* **32**(3), 359–373 (March 1989).
6. R. W. Hall, Fast parallel thinning algorithm: parallel speed and connectivity preservation, *Comm. ACM* **32**(1), 124–131(January 1989).
7. C. M. Holt, A stewart, M. Clint and R. H. Perrott, An improved parallel thinning algorithm, *Comm. ACM* **30**(2), 156–160 (February 1987).
8. B. K. Jang and R. T. Chin, Analysis of thinning algorithm using mathematical morphology, *IEEE Trans. Pattern Analysis and Machine Intelligence* **12**(6), 541–551 (June 1990).
9. B. K. Jang and R. T. Chin, One-pass parallel thinning: analysis, property, and quantitative evaluation, *IEEE Trans. Pattern Analysis and Machine Intelligence* **14**(11), 1129–1140 (November 1992).
10. P. C. K. Kwok, A thinning algorithm by contour generation, *Comm. ACM* **31**(11), 1314–1324 (November 1988).
11. H. E. Lu and P. S. P. Wang, A comment on: a fast thinning algorithm for thinning digital patterns. *Comm. ACM* **29**(3), 239–242 (March 1986).
12. 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. Pattern Analysis and Machine Intelligence* **11**((4), 411–414 (April 1989).
13. C. W. Niblack, D. W. Capson and P. B. Gibbons, Generating skeletons and centerlines from the medial axis transform, *Proc. 10th Inter. Conf. Pattern Recognition*, Atlantic City, New Jersey, pp. 881–885 (June 1990).
14. P. A. Margos and R. W. Schafer, Morphological skeleton representation and coding of binary images, *IEEE Trans. Acoustics, Speech and Signal Processing* **ASSP-34**(5) 1228–1244 (October 1986).
15. G. Borgefors, Distance transformations in digital images, *Computer Vision, Graphics and Image Processing* **34**, 344–371 (1986).
16. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. Academic Press, New York (1982).
17. A. Rosenfeld and J. L. Pfaltx, Sequential operations in digital picture processing, *J. ACM* **13**, 471–494 (1967).
18. J. Serra, *Image Analysis and Mathematical Morphology*. Academic Press, New York (1982).
19. F. Y. Shih, C. T. King and C. C. Pu, A two-scan algorithm and architecture to a root for morphological filters, *Proc. IEEE Inter. Phoenix Conf. Computers and Communications*, Arizona, pp. 78–84 (March 1990).
20. F. Y. Shih and O. R. Mitchell, A mathematical morphology approach to Euclidean distance transformation, *IEEE Trans. Image Processing* **1**(2), 197–204 (April 1992).
21. P. E. Danielson, Euclidean distance mapping, *Computer Graphics Image Processing* **14**, 227–248 (1980).
22. A. M. Vossepoel, A note on distance transformations in digital images, *Computer Vision, Graphics and Image Processing* **43**, 88–97 (1988).

About the Author—FRANK YEONG-CHYANG SHIH received the B.S. degree from National Cheng-Kung University, Tainan, Taiwan, R.O.C., in 1980, the M.S. degree from the State University of New York, Stony Brook, in 1984, and the Ph.D. degree from Purdue University, West Lafayette, Indiana, in 1987, all in electrical engineering. Then he joined the Faculty of New Jersey Institute of Technology, where he currently serves as Associate Professor of the Computer and Information Science Department, and of the Electrical and Computer Engineering Department. He also serves as the Director of Computer Vision Laboratory.

Dr Shih is on the editorial board of three international journals, *Systems Integration, Information Sciences* and *Pattern Recognition*. He has served as the session chair and program committee for numerous international conferences. He received the Research Initiation Award from the National Science Foundation in 1991 for supporting research in mathematical morphology. He was on the NSF Computer Vision review panel. He has published over 75 technical papers. He is currently active in teaching and research in the areas of image processing, computer vision, computer graphics, expert systems, pattern recognition, computer architecture, fuzzy logic and neural networks.

About the Author—CHRISTOPHER CHAMIM PU received the B.S. degree in 1983 from National Chiao-Tung University, Hsinchu, Taiwan, R.O.C., in computer engineering, the M.S. degree in 1988 from the University of Arizona, Tucson, and the Ph.D. degree in 1993 from New Jersey Institute of Technology, both in computer science.

He joined Ming Hsin Institute of Technology, Hsinchu, Taiwan, R.O.C., in August 1993, where he is now Associate Professor and Chairman of the Department of Information Management. His research interests include image processing and analysis, mathematical morphology, computer vision and computer graphics.