Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution

Xiang Bai, Longin Jan Latecki, *Member*, *IEEE Computer Society*, and Wen-Yu Liu

Abstract—In this paper, we introduce a new skeleton pruning method based on contour partitioning. Any contour partition can be used, but the partitions obtained by Discrete Curve Evolution (DCE) yield excellent results. The theoretical properties and the experiments presented demonstrate that obtained skeletons are in accord with human visual perception and stable, even in the presence of significant noise and shape variations, and have the same topology as the original skeletons. In particular, we have proven that the proposed approach never produces spurious branches, which are common when using the known skeleton pruning methods. Moreover, the proposed pruning method does not displace the skeleton points. Consequently, all skeleton points are centers of maximal disks. Again, many existing methods displace skeleton points in order to produces pruned skeletons.

Index Terms—Skeleton, skeleton pruning, contour partition, discrete curve evolution.

1 Introduction

The skeleton is important for object representation and recognition in different areas, such as image retrieval and computer graphics, character recognition, image processing, and the analysis of biomedical images [1]. Skeleton-based representations are the abstraction of objects, which contain both shape features and topological structures of original objects. Because of the skeleton's importance, many skeletonization algorithms have been developed to represent and measure different shapes. Many researchers have made great efforts to recognize the generic shape by matching skeleton structures represented by graphs or trees [2], [3], [4], [29], [30], [31], [36]. Unfortunately, these approaches have only demonstrated an applicability to objects with simple and distinctive shapes and, therefore, cannot be applied to more complex shapes like shapes in the MPEG-7 data set [37]. The most significant factor constraining the matching of skeletons is the skeleton's sensitivity to an object's boundary deformation: little noise or a variation of the boundary often generates redundant skeleton branches that may seriously disturb the topology of the skeleton's graph. For example, the skeleton in Fig. 1a has many redundant skeleton branches generated by boundary noise.

To overcome a skeleton's instability of boundary deformation, a variety of techniques have been suggested for matching and recognizing shapes. Zhu and Yuille [29] generate more than one possible skeleton graph to overcome unreliability. A similar shape descriptor based on the self-similarity of a smooth outline is presented in [30]. Aslan and Tari [31] posit an unconventional approach to shape recognition using unconnected skeletons in the course level. While their approach leads to stable skeletons in the presence of boundary deformations, only rough shape classification can be performed since the obtained skeletons do not represent any shape details.

The most common approaches to overcome skeleton instability are based on skeleton pruning, (i.e., eliminating redundant skeleton branches). Pruning can either be performed implicitly as a post processing step or implicitly integrated in the skeleton computation. However, none of the existing skeleton pruning methods yields satisfactory results without user interaction. Before describing the existing skeleton pruning approaches, we characterize the desirable properties of skeletons. The skeleton of a single connected shape that is useful for skeleton-based recognition should have the following properties:

1. it should preserve the topological information of the original object,
2. the position of the skeleton should be accurate,
3. it should be stable under small deformations,
4. it should contain the centers of maximal disks, which can be used for reconstruction of original object,
5. it should be invariant under Euclidean transformations such as rotations and translations, and
6. it should represent significant visual parts of objects.

The main goal of this paper is to present a method that extracts the exact skeleton with a new skeleton-pruning method and which will achieve all the above properties. No existing method can provide a skeleton with all these properties. Our proposed method is easy to implement and can be computed efficiently.

The following is a brief overview of skeletonization and skeleton-pruning approaches. The skeletonization algorithms can broadly be classified into four types:

- The first type is thinning algorithms, such as those with shape thinning and the wave front/grassfire transform [8], [9], [10], [34]. These algorithms

- *X. Bai and W.-Y. Liu are with the Department of Electronics and Information, Engineering, Huazhong University of Science and Technology, N1 Hall, D425, HUST, Luoyu Road 1043, Wuhan, Hubei, China, 430074. E-mail: baihouxiang@hotmail.com, liuwy@hust.edu.cn.*
- *L.J. Latecki is with the Department of Computer and Information Sciences, Temple University, 1805 North Broad Street, Philadelphia, PA 19122. E-mail: latecki@temple.edu.*

Manuscript received 25 Oct. 2005; revised 20 Apr. 2006; accepted 5 July 2006; published online 15 Jan. 2007.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0573-1005.

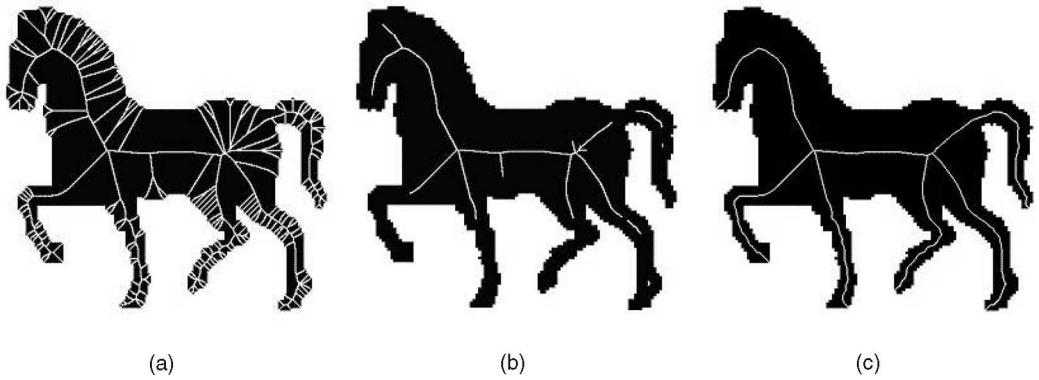

Fig. 1. The skeleton in (a) has many redundant branches. To remove them, usually skeleton pruning is applied. (b) Illustrates the problems of actual pruning approaches (it is generated by a method in [7]). In particular, observe that pruning may change the topology of the original skeleton. (c) Illustrates the pruning result of the proposed method that is guaranteed to preserve topology.

iteratively remove border points, or move to the inner parts of an object in determining an object's skeleton. These methods usually preserve the topology of the original object with many redundant branches, but they are quite sensitive to noise and often fail to localize the accurate skeletal position. In addition, it is important to determine a good stop criterion of this iterative process.

- The second type is the category of discrete domain algorithms based on the Voronoi diagram [5], [12], [27], [28]. These methods search the locus of centers of the maximal disks contained in the polygons with vertices sampled from the boundary. The exact skeleton can be extracted as the sampling rate increases, but the time of computation is usually prohibitive. The obtained skeleton is extremely sensitive to local variance and boundary noise, so that complicated skeleton bunches need to be pruned [5], [28].
- The third type of algorithms is to detect ridges in a distance map of the boundary points [7], [10], [11], [13], [19], [33], [35]. Approaches based on distance maps usually ensure accurate localization but neither guarantees connectivity nor completeness [7], [13]. Under the completeness, the skeleton branches representing all significant visual parts are present (6).
- The fourth type of algorithms is based on mathematical morphology [22], [24], [25], [26]. Usually, these methods can localize the accurate skeleton [24], but may not guarantee the connectivity of the skeleton [22].

All of the obtained skeletons are subjected to the skeleton's sensitivity and many of them also include pruning methods along with the skeletonization. As an essential part of skeletonization algorithms, skeleton pruning algorithms usually appear in a variety of application-dependent formulations [20]. There are two main pruning methods: 1) based on significance measures assigned to skeleton points [5], [6], [7], [20], [28] and 2) based on boundary smoothing before extracting the skeletons [20], [38], [39]. In particular, curvature flow smoothing still has some significant problems that makes the position of skeletons shift and have difficulty in distinguishing noise from low frequency shape information on boundaries [20]. A different kind of smoothing is proposed in [14]. Great progress has been made in the type 1) of pruning approaches that define a significance measure for skeleton points and remove points whose significance is low. Shaked and Bruckstein [20] give a complete analysis and compare such pruning methods. Propagation velocity, maximal thickness, radius function, axis arc length, and the length of the boundary unfolded belong to the common significance measures of skeleton points. Ogniewicz and Kübler [5] present a few significance measures for pruning complex Voronoi skeletons without disconnecting the skeletons. Siddiqi et al. combine a flux measurement with the thinning process to extract a robust and accurate connected skeleton [25].

All presented methods have several drawbacks. First, many of them are not guaranteed to preserve the topology of a complexly connected shape (e.g., a shape with holes). This is illustrated in Fig. 2, where the skeleton in Fig. 2d violates the topology of the input skeleton in Fig. 2c. This skeleton was obtained by the method in [7]. However, many methods described above would lead to topology violation, particularly all methods presented in [20] (including the method of Ogniewicz and Kübler [5]). These methods are guaranteed to preserve topology for simply connected objects (objects with a single contour), but not for objects with more than one contour like the can in Fig. 2. The topology preserving skeleton obtained by the proposed pruning method is illustrated in Fig. 2e. We will prove in the Appendix, which can be found at http://computer.org/tpami/archives.htm, that our method is guaranteed to preserve topology. Even if the input shape is simply connected, some of methods described above are not guaranteed to preserve the original topology (e.g., see in Fig. 1b, generated by the pruning method in [7]).

The second drawback of the methods described above is that main skeleton branches are shortened and short skeleton branches are not removed completely. This may lose important shape information and seriously compromise the structure of the skeletons. These effects are illustrated in Figs. 1b and 3a, e.g., the horse legs in Fig. 1b are shortened too much, although, at the same time, some spurious skeleton branches remained. Thus, shortening of branches may cause branches of significant visual parts to be indistinguishable form branches resulting from noise.

The third drawback is that usually only the local significance of the skeleton points is considered, and the global information of the shape is discarded. However, the

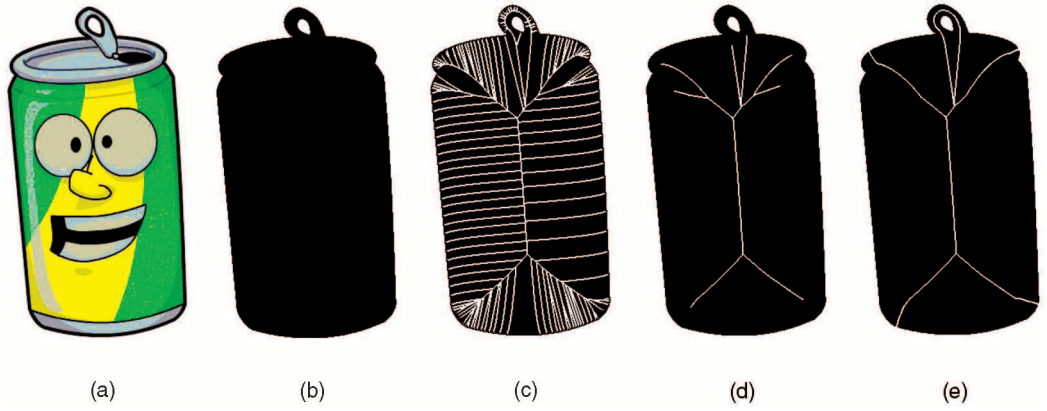

Fig. 2. (a) The input object. (b) Binary object mask. (c) The initial skeleton. (d) A pruned skeleton obtained by the method in [7]. (e) A pruned skeleton obtained by the proposed method. While the skeleton in (d) violates the topology, the proposed method is guaranteed to preserve the topology.

same part may represent an important shape feature for one shape while it may represent noise for a different shape. This is illustrated in Fig. 4. Clearly, the spike in Fig. 4b is less relevant for the overall shape than in Fig. 4a, and consequently, it is more likely to be a result of noise. The proposed pruning method is able to recognize this fact, which leads to the removal of the skeleton branch induced by the spike in Fig. 4d. In contrast, the relevance of skeleton points in the existing pruning methods is computed based only on local contour information, which means that they cannot differentiate the two spike induced skeleton branches in Figs. 4a and 4b. Consequently, their pruning result is very similar to the skeletons shown in Figs. 4a and 4b. The fourth drawback is that pruning results may be different for different scales as pointed out in [40].

An interesting idea, called a fixed topology skeleton, is presented by Golland and Grimson [11]: The process of pruning is skipped and the skeletonization uses a snake-like algorithm for estimating the positions of the skeleton with respect to the fixed skeleton endpoints. Since the fixed endpoints are not changed in the iteration process, a skeleton with the global topology can be extracted. However, the topology of the global shapes must be known before skeletonization for the fixed topology skeleton and the position of the obtained skeleton is not accurate.

To summarize, although the existing skeleton pruning methods have many drawbacks, they are definitely needed to remove inaccurate or redundant skeleton branches. The skeleton generating approaches suffer from the fact that a small protrusion on the boundary may result in a large skeleton branch, which is an intrinsic problem of the definition of the skeleton, since the mapping of boundary points to the skeleton points is not continuous. An obvious solution to this problem is to first remove the protrusions on the boundary and then compute the skeleton. As stated above, various smoothing approaches are either applied to the contour or to the distance map before the skeleton is computed. The problem is that isotropic (e.g., Gaussian) as well as anisotropic smoothing only reduces, but does not remove the protrusions [4]. A common characteristic of the above approaches is that they displace the boundary points and, consequently, displace the location of skeleton points.

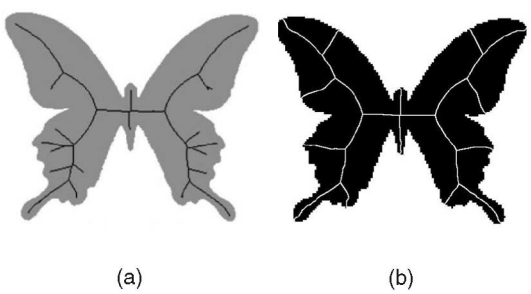

Fig. 3. Comparison on between the result in [7] (a) and our result in (b).

2 Main Ideas of the Proposed Approach

We propose an approach that completely removes protrusions without displacing the boundary points and, therefore, without displacing the remaining skeleton points. Thus, inaccurate or redundant branches are completely removed while the main branches are not shortened. As illustrated above, the proposed method also does not have the other three drawbacks listed above. The main observation of our approach is that it is possible to perform a topology preserving skeleton pruning based on a contour partition into curve segments. Returning to Blum's definition of the skeleton, every skeleton point is linked to boundary points that are tangential to its maximal circle. These are called **generating points**. The main idea is to remove all skeleton points whose generating points all lie on the same contour segment. This works for any contour partition in segments, but some partitions yield better results than other. Fig. 5 illustrated three different pruned skeletons in Figs. 5b, 5c, and 5d) obtained for the same input skeleton in Fig. 5a. The pruned skeletons are based on three different partitions of contour segments whose endpoints are marked with dots. For example, removing all skeleton points all of whose generating points lie on the contour segment CD in Fig. 5c leads to the removal of the entire lower part of the skeleton. Clearly, the contour partition in Fig. 5d leads to a significantly better pruning result than the partitions in Figs. 5b and 5c. Thus, in our framework, the question of skeleton pruning is reduced to finding a good partition of the contour into segments. We obtain such partitions with the process of Discrete Curve Evolution (DCE) [15], [16], [17], which we briefly introduce as follows.

First, observe that every object boundary in a digital image can be represented without the loss of information as a finite

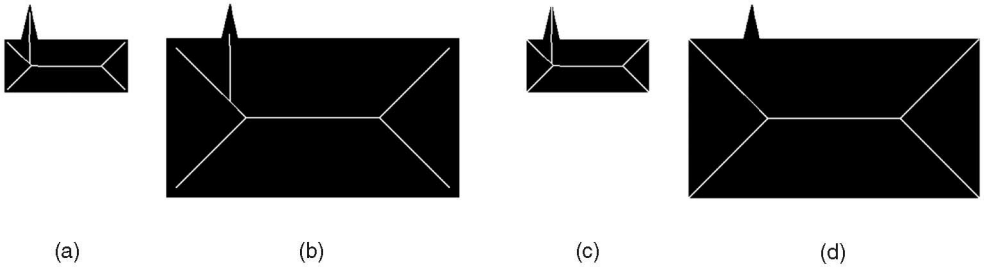

Fig. 4. (a) and (b) show pruned skeletons obtained by the method in [7]. The proposed pruning method can distinguish that shape contribution of the spike in (b) is smaller than in (a) and, therefore, it is possible to prune the branch resulting form the spike in (d).

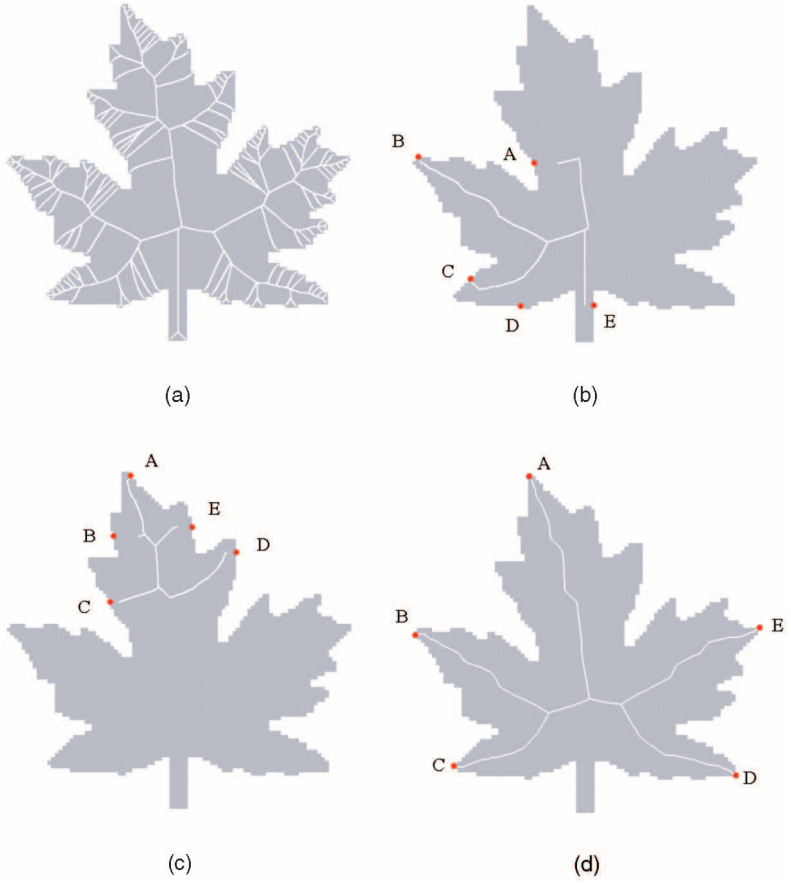

Fig. 5. Pruning the input skeleton (a) with respect to contour partition induced by five random points on the boundary in (b) and (c). The five points in (d) are selected with DCE.

polygon, due to finite image resolution. Let us assume that the vertices of this polygon result from sampling the boundary of the underlying continuous object with some sampling error. There then exists a subset of the sample points that lie on the boundary of the underlying continuous object (modulo some measurement accuracy). The number of such points depends on the standard deviation of the sampling error. The larger the sampling error, the smaller the number of points will lie on the boundary of the continuous object, and subsequently, the less accurately we can recover from the original boundary [15]. The question arises as to how to identify the points that lie on (or very close to) the boundary of the original object or equivalently how to identify the noisy points (that lie far away from the original boundary). The process of DCE is proven experimentally and theoretically to eliminate the noisy points [15], [16], [17]. This process eliminates such points by recursively removing polygon vertices with the smallest shape contribution (which are the most likely to result from noise). As a result of DCE, we obtain a subset of vertices that best represents the shape of a given contour. This subset can also be viewed as a partitioning of the original contour polygon into contour segments defined by consecutive vertices of the simplified polygon. A hierarchical skeleton structure obtained by the proposed approach is illustrated in Fig. 6, where the (red) bounding polygons represent the contours simplified by DCE. Because DCE can reduce the boundary noise without displacing the remaining boundary points, the accuracy of the skeleton position is guaranteed. The continuity, which implies stability in the presence of noise, of the proposed pruning methods follows from the continuity of the DCE. This means that if a given contour and its noisy versions are close (measured by Hausdorff distance), the obtained pruned skeletons will also be close. A formal proof of DCE continuity with respect to the Hausdorff

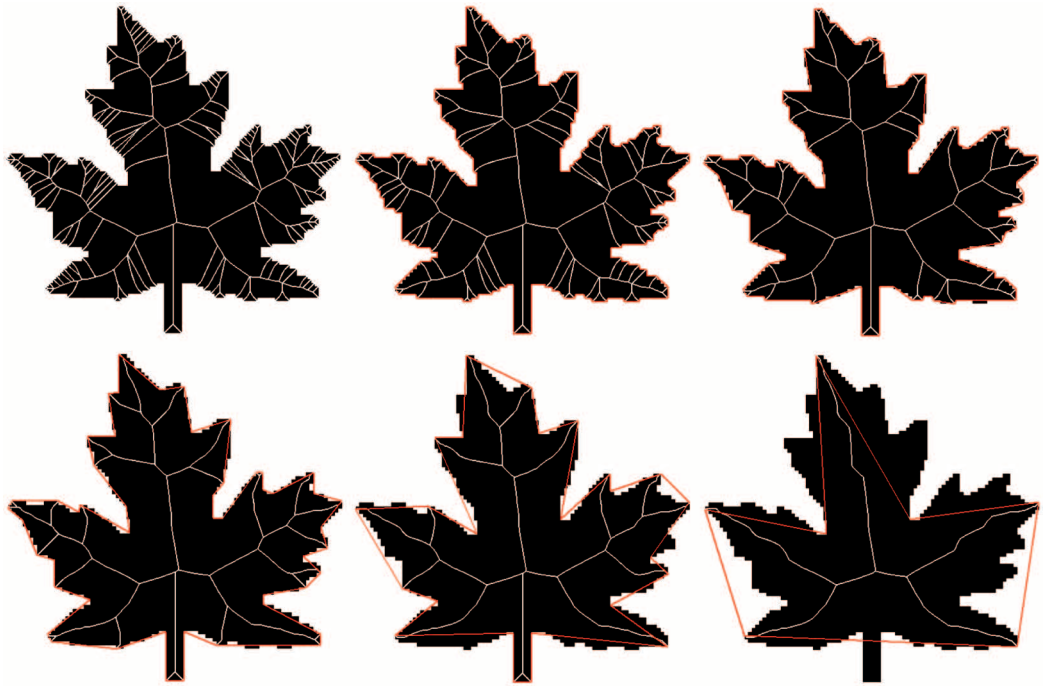

Fig. 6. Hierarchical skeleton of leaf obtained by pruning the input skeleton (top left) with respect to contour segments obtained by the Discrete Curve Evolution (DCE). The outer (red) polylines show the corresponding DCE simplified contours.

distance of polygonal curves is given in [23]. Thus, our approach provides a solution to the instability of the classical skeleton pruning algorithms.

All pruning methods based on a significance measure for skeleton points use local criteria to compute this measure [21], [29], [5], [6], (e.g., the measure in [5] is base on the shortest contour arc between the generating points). Also, all contour smoothing methods are based on local contour information only. In contrast, DCE evaluates global contour information in order to generate the simplified contour. This property is illustrated in Fig. 4. The same spike is on the boundary of Fig. 4a and Fig. 4b, but it has different shape contribution for both objects. While it is more likely to be a shape feature in Fig. 4a, it is more likely to be regarded as noise in the object Fig. 4b. DCE can effectively quantify this difference in shape contribution. Consequently, we obtain the skeletons as shown in Fig. 4c and Fig. 4d.

The proposed pruning method can be applied to any input skeleton. We only require that each skeleton point is the center of a maximal disk and that the boundary points tangent to the disk (generating points) are given. We also present a skeleton growing algorithm that includes an efficient implementation of the proposed pruning method. The main idea is that the pruning is not done in postprocessing (after the skeleton is computed) but is integrated into the skeleton growing process. To implement this idea, we extended the skeleton growing algorithm in [7] based on the Euclidean distance map. First, we selected a skeleton seed point as a global maximum of the Euclidean distance map. Then, the remainder of the skeleton points is decided by a growing scheme. In this scheme, the new skeleton points are added using a simple test that examines their eight connected points. During this process, the redundant skeleton branches are eliminated by the DCE.

3 Background Definitions

Before we define a skeleton, we need to characterize planar sets for which we can determine the skeleton. Following [32], we assume that a planer set D is the closure of a connected bounded open subset of R^2 whose boundary ∂D is composed of a finite number of mutually disjoint simple closed curves. Each simple closed curve in ∂D consists of a finite number of pieces of real analytic curves. We further assume in this paper that each simple closed curve is a polygonal curve, (i.e., the pieces they consist of are line segments). We make this assumption only to simplify some definitions and we stress that all of our results also hold for simple closed curves that consist of a finite number of real analytic curves. This assumption does not introduce any restriction on object contours in digital images since each boundary curve in a digital image can be regarded as polygonal curve with vertices being the boundary pixels.

According to Blum's definition of the medial axis [1], the **skeleton** $\boldsymbol{S}(D)$ of a set D is the locus of the centers of maximal disks. A maximal disk of D is a closed disk contained in D that is interiorly tangent to the boundary ∂D and that is not contained in any other disk in D. Each maximal disc must be tangent to the boundary in at least two different points. We denote as $\boldsymbol{T}an(s)$ the set of the boundary points tangent to the maximal closed disk $\boldsymbol{B}(s)$ centered at $s \in S(D)$. The points in $Tan(s)$ are called **generating points** of the skeleton point s. Due to our assumption that, each boundary curve is a simple closed polygonal curve, $Tan(s)$ is composed of a finite number of isolated boundary points, since $B(s)$ can intersect each boundary line segment in at most one point. (Without this assumption, $Tan(\mathrm{s})$ would be composed of a finite number of isolated contour subarcs.) The **degree** $\deg(s)$ of $s \in S(D)$ is defined as the cardinality of $Tan(s)$, (i.e., as the number of boundary points tangent to the maximal circle centered at s). Let the boundary ∂D of D be composed of

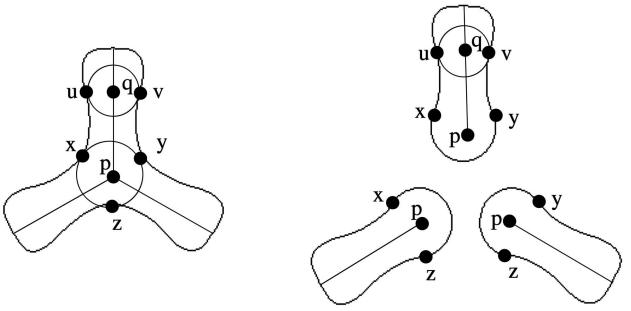

Fig. 7. This figure illustrates Theorem 5.1 in [32], called the Domain Decomposition Lemma.

k simple closed curve polygonal curves $C_1, \ldots, C_k$. Then the degree with respect to C_j $\deg(s, C_j)$ is equal to the cardinality of $Tan(s) \cap Cj$.

For a given boundary point $x \in \partial D$, we define $S(x)$ as the center of the maximal disk that is tangent to ∂D at x. The function $S : \partial D \rightarrow S(D)$ is a strong deformation retraction by Theorem 8.1 in [32]. Moreover, by Theorem 8.2 in [32], the skeleton $S(D)$ is a geometric graph, which means that $S(D)$ can be decomposed into a finite number of connected arcs, called **skeleton branches**, composed of points of degree two, and the branches meet at **skeleton joints** (or **bifurcation points**) that are points of degree three or higher.

We also summarize some of the consequences of Theorem 5.1 in [32], called Domain Decomposition lemma, that will be particularly useful here. For an illustration, see Fig. 7. Given a skeleton point $p \in S(D)$, the maximal closed disk $B(p)$ decomposes $D - B(p)$ into a finite number of connected components $D_1(p), \ldots, D_k(p)$; also $\partial D - B(p)$ is decomposed into a finite number of open contour curves $C_1(p), \ldots, C_k(p)$, and the skeleton $S(D) - \{p\}$ is decomposed into finite number of skeleton curves $S_1(p), \ldots, S_k(p)$, such that $\mathrm{C_i}(p) = D_i(p) \cap \partial D$ and $S_i(p) = D_i(p) \cap S(D)$. A very important consequence of this theorem is that, for two different skeleton points $p, q \in S(D)$, we must have one of the following three cases $C_\mathrm{i}(p) \cap C_\mathrm{j}(q) = \emptyset$ or $C_\mathrm{i}(p) \subset C_\mathrm{j}(q)$ or $C_\mathrm{j}(q) \subset C_\mathrm{i}(p)$. For example, in Fig. 7, $C_1(p) = (x, y)$, which is an open contour segment, $C_1(q) = (u, v)$, and we have $C_1(q) \subset C_1(p)$, while $C_2(p) \cap C1(q) = \emptyset$ and $C_3(p) \cap C_1(q) = \emptyset$, since $C_2(p) = (y, z)$ and $C_3(p) = (z, x)$.

4 Skeleton Pruning with Contour Partition

In this section, we introduce the contour partition into contour segments and skeleton pruning based on it.

Definition 1. *Let the boundary ∂D of a set D be composed of k simple closed curves $C_1, \ldots, C_k$. Let x and y be two contour points lying on the same simple closed curve C_i. With $[x, y]$, we denote the shortest closed contour segment (subarc) of C_i that connects x and y. For simplicity, we assume that x and y are positioned on C_i so that $[x, y]$ is uniquely determined. With (x, y), we denote the segment $[x, y]$ without the endpoints x and y (i.e., the open subarc). (A distinction between open and closed contour segments is unimportant in the digital images, but we need to establish some formal properties on the continuous plane.) A sequence of points $x_0, \ldots, x_{n-1}$ on a simple closed curve C_i forms a partition of C_i if two consecutive segments $[x_i, x_{i+1}]$, $[x_{i+1}, x_{i+2}]$ intersect in $\{x_{i+1}\}$ (the indices are modulo n), nonconsecutive segments have empty intersection, and C_i is the union of these segments. The partition Γ of the boundary ∂D is a sequence of sequences that are partitions of the simple closed curves $C_1, \ldots, C_k$.*

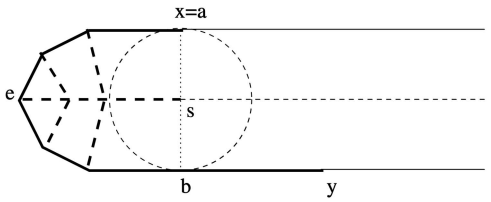

Fig. 8. The initial contour segment [x,y] is marked with a thick continuous line. $\mathrm{CS([x, y])} = \mathrm{[a, b]} = \mathrm{arc(s, [x, y])}$, where $S(a) = S(b) = \mathrm{s}$. The corresponding skeleton part $S([a, b])$ is marked with a thick dashed line.

Definition 2. *Let $[x, y]$ be a contour segment that belongs to some contour partition Γ. In particular, $[x, y]$ is a subsegment of one of the contour curves C of ∂D. For a skeleton point s whose all generating points $Tan(s)$ lie in $[x, y]$, let $\boldsymbol{arc(s, [x, y])}$ be the smallest subarc of $[x, y]$ that contains $Tan(s)$. Observe that $arc(s, [x, y])$ is a contour segment of C (i.e., $arc(s, [x, y]) = [a, b]$ for some $a, b \in C$, since $arc(s, [x, y])$ is an arc connected subset of $[x, y]$.) As a consequence of Theorem 5.1 in [32], we also obtain that $\mathrm{S(a) = S(b) = s}$ (Fig. 8).*

Let $CS([x, y]) = \{z \in [x, y] : S^{-1}(S(z)) \subset [x, y]\}$ be the set of all points z in $[x, y]$ such all generating points of $S(z)$ are contained in $[x, y]$. For example, in Fig. 8, $CS([x, y]) = [a, b]$. Similarly, we can define $CS((x, y))$ for an open segment (x, y).

Definition 3. *Given a partition Γ of the boundary ∂D of a simply connected set D (i.e., ∂D consist of one simple closed curve), the **skeleton pruning** is defined as the removal of all skeleton points $s \in S(D)$ whose generating points lie in the same open segment of the partition. More precisely, the pruned skeleton is composed of all points $s \in S(D)$ such that $Tan(s)$ is not contained in the same open segment of the partition Γ.*

This is a very simple definition of skeleton pruning, and it works with any contour partition. The key issue is to get reasonable partitions. As we will show, DCE provides a very good partition for the pruning. We show in Theorem 1 (in the Appendix which can be found at http://computer.org/tpami/archives.htm) that the topology of a pruned skeleton is preserved for a pruned skeleton generated by any partition of the contour. We illustrate the meaning of Theorem 1 in Fig. 8. The $CS((x, y)) = (a, b)$ is a subsegment of (x, y). Therefore, the thick dashed part of the skeleton $S(CS((x, y)))$ generated by contour segment (x, y) can be removed and the pruned skeleton has the same topology. Observe that the only point in $S(CS([x, y]))$ that connects $S(CS([x, y]))$ to the rest of the skeleton is point s.

The situation is a bit more complicated if D is not simply connected (i.e., ∂D consist of more than one simple closed curve). For example, $CS([x, y]) = [a, c] \cup [d, b]$ shown in Fig. 9 is not a subsegment of $[x, y]$, due to the interior simple closed curve. Therefore, $S(CS([x, y])) = [u, s]$ cannot be removed without violating the topology. Observe that it suffices to additionally check for every partition segment $[x, y]$ whether $CS([x, y])$ is arc connected. When $CS((x, y))$ is arc connected, then we can remove part $S(CS((x, y)))$ of the skeleton without violating the skeleton topology as proven in Theorem 2 (in the Appendix which can be found at http://computer.org/tpami/archives.htm).

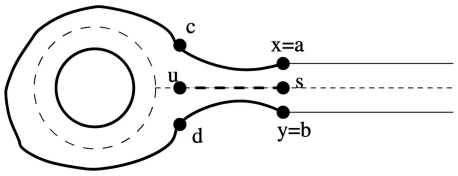

Fig. 9. The initial contour segment $[x, y]$ is marked with a continuous thick line. Observe that $CS([x, y]) = [a, c] \cup [d, b]$ is not a subsegment of $[x, y]$ since it is not arc connected. Therefore, $CS([x, y])$ is not equal to $\text{arc}(s, [x, y])$, where $S(a) = S(b) = s$. Since $CS([x, y])$ is not a subsegment of $[x, y]$, $S(CS([x, y]))$ cannot be removed by Theorem 2. The skeleton part $S(CS([x, y]))$ represented by the segment $[u, s]$ is marked with a thick dashed line. Observe that removing $[u, s]$ disconnects the skeleton.

5 Skeleton Pruning with Discrete Curve Evolution

In this section, we introduce the contour segmentation process based on Discrete Curve Evolution (DCE). The hierarchical decomposition of the boundary of the set D obtained by DCE is the key component in the proposed skeleton pruning method.

5.1 Discrete Curve Evolution

The Discrete Curve Evolution (DCE) method was introduced in [16], [17], [18]. Contours of objects in digital images are distorted by digitization noise and segmentation errors; it is desirable to eliminate the distortions while at the same time preserving the perceptual appearances sufficient for object recognition. DCE accomplishes this goal by simplifying the shape. For example, a few stages of DCE are illustrated in Fig. 6 for the outer (red) polylines. The shape of the leaf becomes more and more simplified by DCE, while preserving the main visual parts.

Since any digital curve can be regarded as a polygon without the loss of information (but, with the possibility of a large number of vertices), it is sufficient to study evolutions of polygonal shapes. The basic idea of the proposed evolution of polygons is simple:

- In every evolutional step, a pair of consecutive line segments s_1, s_2 is replaced by a single line segment joining the endpoints of $s_1 \cup s_2$.

The key property of this evolution is the order of the substitution. The substitution is achieved according to a relevance measure K given by:

$$K(S_1, S_2) = \frac{\beta(S_1, S_2) l(S_1) l(S_2)}{l(S_1) + l(S_2)},$$

where line segments s_1, s_2 are the polygon sides incident to a vertex v, $\beta(s_1, s_2)$ is the turn angle at the common vertex of segments s_1, s_2, l is the length function normalized with respect to the total length of a polygonal curve C. The main property of this relevance measurement is [16], [18]:

- The higher value of $K(s_1, s_2)$, the larger is the contribution of the arc $s_1 \cup s_2$ to the shape.

Given the input boundary polygon P with n vertices, DCE produces a sequence of simpler polygons $P = P^n, P^{n-1}, \ldots, P^3$ such that $P^{n-(k+1)}$ is obtained by removing a single vertex v from P^{n-k} whose shape contribution measured by K is the smallest.

Definition 4. *An important property of DCE is that it introduces a hierarchical partition of the input polygon P. Let $\{v_1, \ldots, v_n\}$ be vertices of P and let $\{u_1, \ldots, u_m\} \subset \{v_1, \ldots, v_n\}$ be the convex vertices of P^{n-k} for $m \leq n - k$. On the level $n - k$ of the partition hierarchy $H_{n-k}(P)$, P is decomposed into m subarcs of $P : H_{n-k}(P) = \{[u_1, u_2], [u_2, u_3], \ldots, [u_m, u_1]\}$. We call these arcs* ***DCE (contour) partition*** *(on DCE level $n - k$). The reason that our partition is based only on convex vertices of P will be explained in the next section, in which skeleton pruning is defined.*

If vertex u_i is deleted in the next evolution step, (i.e., $u_i \in P^{n-k} - P^{n-(k+1)}$), or becomes concave (due to the deletion of one of its neighbors), then the arc $[u_{i-1}, u_{i+1}]$ replaces arcs $[u_{i-1}, u_i]$, $[u_i, u_{i+1}]$ in the partition level $H_{n-(k+1)}(P)$.

Observe that DCE and the hierarchical partition can be also defined for a finite set of polygonal curves. The only difference is that in each DCE step a single vertex is removed from one of the polygons whose actual relevance measure is the smallest. This observation is particularly important for our approach, since the proposed pruning can be applied to a planar set D such that its boundary ∂D is

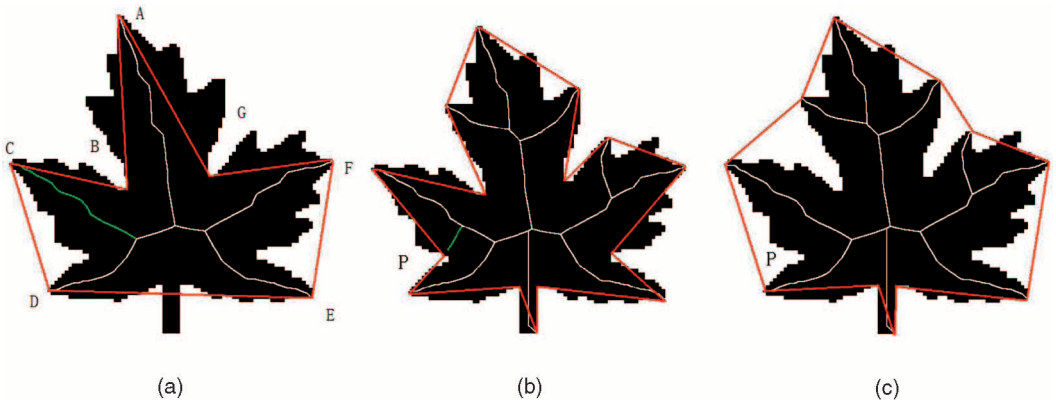

Fig. 10. (a) A simplified polygon with seven vertices (in red) and the skeleton obtained based on this polygon. The green skeleton branch (ending at C) remained, since each of its points has generating points on two different arcs BC and CD of the original contour. A skeleton branch shown in green in (b) does not belong to the skeleton determined by the DCE polygon, since it ends at a concave vertex P. As shown in (c), it would have been removed anyway, but at a later stage of DCE simplification.

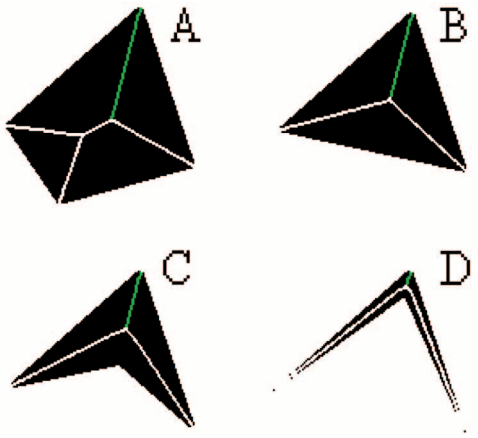

Fig. 11. The same convex vertices may generate different skeleton branches with different importance.

composed of a finite number of simple closed polygons. Thus, the connected set D may have holes. In other words, D does not need to be simply connected.

Though the DCE procedure can effectively remove the noise and visually unimportant portions of the image, a proper stop parameter is still necessary. In other words, we seek such a k so that the simplified polygon P^{n-k} represents the input contours on the adequate level of detail. In order to quantify the level of detail, we define the average distance $D_{av}(P^{n-k})$ between original points of P and their corresponding line segments in P^{n-k}.

Given a threshold T, we can stop DCE if $D_{av}(P^{n-k}) > T$ for some k. Given a sequence of T values, we can obtain a hierarchical sequence of DCE simplified boundary polygons, which leads to a hierarchical sequence of corresponding skeletons. In general, an adequate stop condition depends on the particular application. A stop condition that is adequate for shape similarity is given for DCE in [18]. It is based on the difference of the DCE simplified contour to the original input contour. When the pruned skeletons are input into a shape similarity measure, this stop condition is recommended.

DCE can be viewed as a greedy approach to simplify the contour so that the length difference between the original and the simplified contour is minimal. It is easy to implement a simplification method (using dynamic programming) which is optimal with respect to the length difference. DCE yields very similar results.

5.2 Skeleton Pruning with Discrete Curve Evolution

Given a skeleton $S(D)$ of a planar shape D and given a DCE simplified polygon P^k, we perform **skeleton pruning** by removing all points $s \in S(D)$ such that the generating points $Tan(s)$ of s are contained in the same open DCE segment. Each pruned point s results from a local contour part with respect to the DCE partition and, therefore, s can be considered as an unimportant skeleton point and can be removed. The simplification of the boundary contour with DCE corresponds to pruning complete branches of the skeleton. In particular, a removal of a single convex vertex v from P^{n-k} to obtain $P^{n-(k+1)}$ by DCE implies a complete removal of the skeleton branch that ends at v. We give an example illustrating this fact in Fig. 10a. This figure shows a polygon with seven vertices obtained from a DCE leaf contour and the skeleton is obtained by pruning based on this polygon. There are only five skeleton branches ending in the five convex vertices of the simplified polygon. The pruned skeleton was computed with respect to the DCE segments (A, C), (C, D), (D, E), (E, F), and (F, A). The pruning was applied to the leaf skeleton shown in the first image in Fig. 6. (The skeleton in Fig. 10a is the same as in the last image in Fig. 6.) We can illustrate the main idea of our approach by explaining why the green skeleton branch in Fig. 10a that ends at point C remained. It remained because each of its points has maximal disks tangent to points on two different DCE segments, which are contour arcs (A, C) and (C, D).

We perform contour decomposition into DCE segments based only on convex vertices of the DCE simplification. This means that not only when a given vertex is removed by DCE but also when a convex vertex becomes concave in the process of DCE, the skeleton branch ending in this vertex is removed. This approach allows us to remove minor (small) branches in the earlier stages of the DCE evolution. Fig. 10b illustrates why we only use convex vertices to define DCE segments. The green branch in Fig. 10b that ends at vertex P would be part of

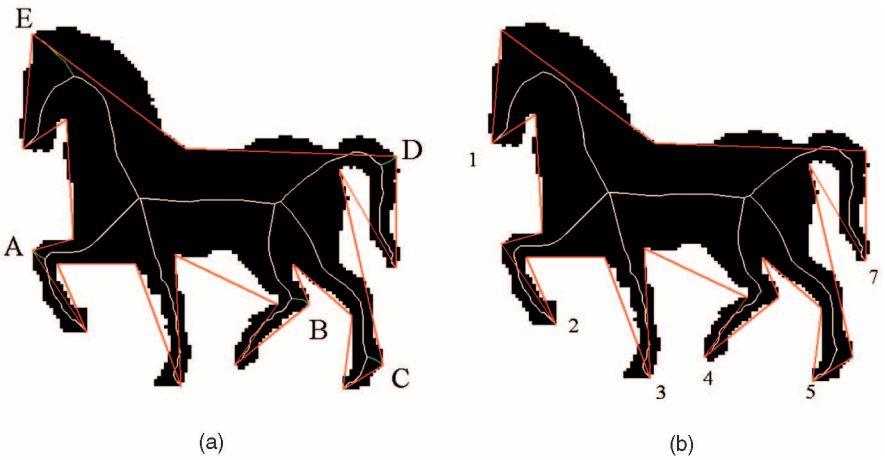

Fig. 12. Removal of unimportant convex vertices for generating an optimal visual skeleton.

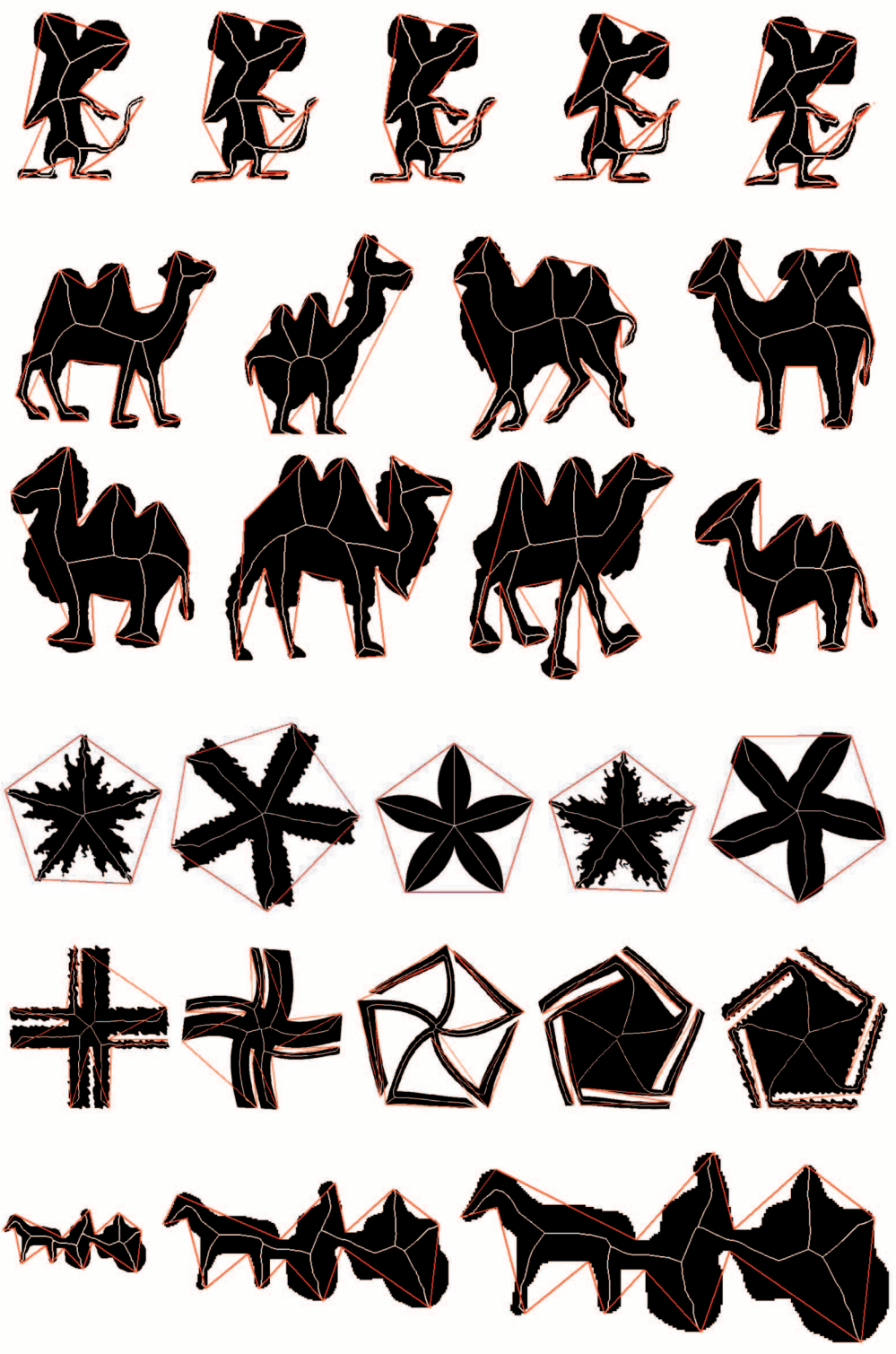

Fig. 13. Our results on Mpeg 7 shape database illustrate the extraordinary stability of pruned skeletons in the presence of significant shape variations and deformations.

the skeleton if we also used concave vertices of the simplified polygon (shown in red) to define DCE segments. This branch would have been removed anyway, since vertex P was removed from the further simplified polygon shown in Fig. 10c. Thus, the fact that DCE segments are defined using only convex vertices of the simplified polygon allows for faster pruning of irrelevant branches.

A very important property of DCE induced contour partition, and every partition that is restricted to vertices of the boundary polygon, is that fact that there is a skeleton branch ending at every partition point. As stated above, if a partition point that is also a polygon vertex u_i is deleted in a DCE evolution step, (i.e., $u_i \in P^{n-k} - P^{n-(k+1)}$), or becomes concave (due to the deletion of one of its neighbors), then

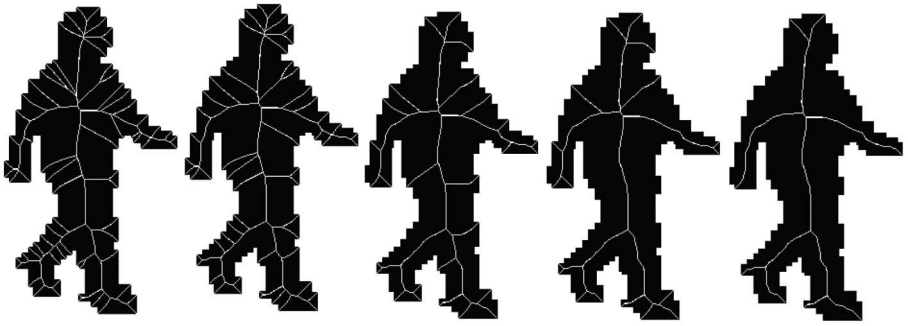

Fig. 14. Hierarchical skeleton of a walking human. The input image is similar to a walking human in [11].

the arc $[u_{i-1}, u_{i+1}]$ replaces arcs $[u_{i-1}, u_i]$, $[u_i, u_{i+1}]$ in the contour partition. Therefore, the whole skeleton branch that ends at vertex u_i is eliminated with skeleton pruning. This fact is proven in Theorem 3 in the Appendix, which can be found at http://computer.org/tpami/archives.htm.

Although convex vertices from DCE can prune skeletons to get clear structures, they may also generate unimportant skeleton branches. We illustrated this problem with Fig. 11. The vertices A, B, C, and D have the same DCE relevance measure K, since K is restricted to directed neighbors of a given vertex. However, the four green skeleton branches ending at them are of differing importance. The branch ending at D has especially and significantly lower importance, and should be removed. Due to the concave vertices inside the shapes with vertices C and D, the importance of the skeleton branches ending at the convex vertices C and D is significantly reduced. Such cases occur in limb shaped parts of visual forms as defined in [41].

To overcome the problem, we introduce an additional relevance measure. For each convex polygon vertex v, we compute the distance $D_l(v)$ between v and the nearest concave vertex u such that the line segment vu is inside the shape if such a vertex u exists. We then remove vertices with low value of the new relevance measure $D_l(v)$.

Fig. 12 illustrates the effect of removing the convex vertices v with low relevance $D_l(v)$. There are five short skeleton branches (in green) that end at A, B, C, D, E in Fig. 12a that have been removed in Fig. 12b. This leads to a contour partition with only seven convex vertices numbered 1-7 in Fig. 12b.

To summarize, the vertices V_f that are used for contour partitioning induced by DCE are computed as: $V_f = V_s - (V_{concave} \cup V_l)$, where V_s denotes all the vertices of the simplified polygon P obtained by DCE, $V_{concave}$ denotes all of the concave vertices of V_s and V_l denotes vertices of V_s with low value of the measure D_l.

5.3 Time Complexity

The contour partition by DCE has a complexity of $O(N \log N)$ [18], where N is the number of the vertices on the original polygon. We can traverse the contour in linear time, $O(N)$, and assign to each contour vertex the label of its partition segment. During skeleton computation, the labels can be passed to each skeleton point as features of generating points. Therefore, the complexity of the proposed pruning is $O(N \log N)$ if DCE is computed, and linear if DCE has been precomputed.

6 Growing a Pruned Skeleton from a Distance Transform

The main goal of this section is to show that it is not necessary to have a separate post-processing step in skeleton pruning, as we can grow a pruned skeleton directly form the distance transform. In this section, we work in the discrete domain of 2D digital images, in which the object contour is still represented with polygons. To achieve our goal, we extend the fast skeleton growing algorithm presented by Choi et al. [7]. We briefly review the skeleton growing algorithm in [7]. First, the Euclidean Distance Transform DT of the binary image of a given shape D is computed. Then the point with the maximal value of $DT(D)$ is selected as a seed skeleton point. Finally, the skeleton is grown recursively by adding points that satisfy a certain criterion, which intuitively means that the added points lie on ridges of the $DT(D)$. The grow process is based on examining every eight-connected point of the current skeleton points. The skeleton continues growing in this way until it reaches an endpoint of a skeleton branch. Next, other skeleton branches starting at other skeleton points are considered.

The proposed extension of the algorithm in [7] is very simple, and it can also be applied to other skeleton growing algorithms. For a point to be added, it must additionally have its generating point on at least two different contour segments of a given contour partition.

7 Experimental Results and Comparison

In this section, we show the performance of the proposed method in three parts: 1) stability in relation to noises and variance, 2) an analysis of our skeletons and comparison to other skeletons, and 3) a discussion of the potential for skeleton matching.

7.1 Stability of Pruning with DCE

Some results on shapes from MPEG-7 Core Experiment CE-Shape-1 database [37] are shown in Fig. 13. For each shape class, we show pruned skeletons for several objects from the same class. Although the objects differ significantly from each other, the obtained pruned skeletons have the same structures. The final DCE simplified polygons are also shown

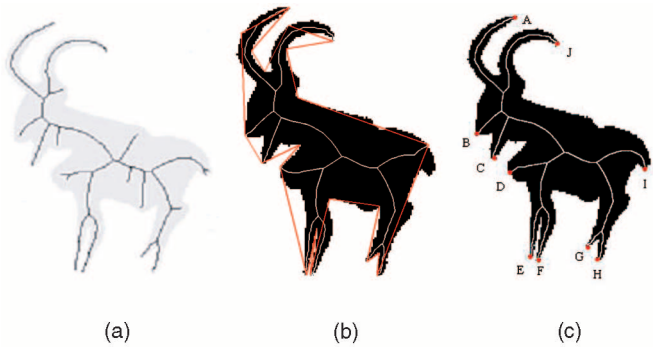

Fig. 15. Comparison between pruning result in [5] in (a) and our results in (b), and (c) is the result of fixed topology skeleton.

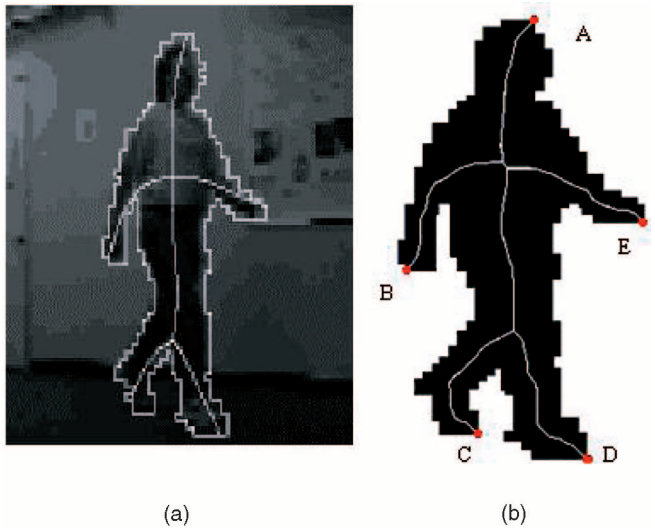

Fig. 16. Comparison between the fixed topology skeleton in [11] in (a) and our skeleton in (b).

overlaid on the shapes with red segments. The skeleton pruning is performed with respect to contour partition induced by the vertices of these polygons. In the first row in Fig. 13, the skeletons of the thin and long tails of rats remained complete. This cannot be achieved by other pruning methods since these may shorten or disconnect the skeleton. Although the camels differ significantly in their shapes, all obtained skeletons have a clear global structure. The extraordinary stability of the skeletons obtained by the proposed pruning method in the presence of significant shape variations and distortions is illustrated for "star" and "plus" shaped objects. These results are possible due to the contour partition stability of DCE. The last row of Fig. 13 shows the DCE's stability to the same shapes in different scales.

7.2 Analysis and Comparison

In this part, we describe our test results with the proposed approach on several binary shape images with the size 500×500. All the images that were tested have significant boundary distortions.

A hierarchy of pruned skeletons is shown for the walking human in Fig. 14. The pruning is preformed with respect to DCE simplified contours with $N = 200$, 100, 50, 30, and 12 vertices. We have also shown a hierarchy of pruned skeletons in Fig. 6. We can see that the results of our algorithm are in accord with human visual perception. Besides

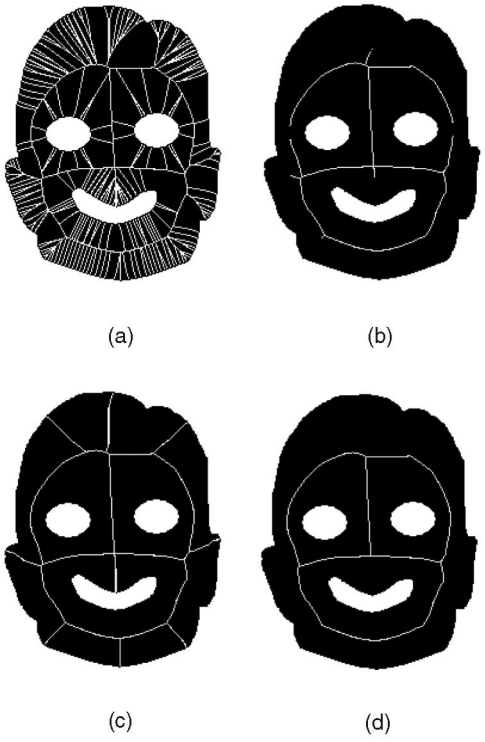

Fig. 17. (a) The input skeleton. (b) A pruned skeleton obtained by the method in [7] violates the topology. (c) and (d) Pruned skeletons obtained by the proposed method, which is guaranteed to preserve the

hierarchical and visual property, our skeleton has a unique property: As proven in Theorem 3 (in the Appendix, which can be found athttp://computer.org/tpami/archives.htm), in the cause of the DCE evolution process, the pruned branches are eliminated completely, (i.e., the obtained skeletons are without the presence of remaining half-shortened small, short branches). For example, in Fig. 14, each skeleton branch is removed, and no remaining fractions are left.

The skeleton in Fig. 15a illustrates a common problem with the existing skeleton pruning approaches [5], which is the problem of inaccurate, half-shortened braches that are not related to any obvious boundary features. It is also shown in Fig. 1b and Fig. 3a. Figs. 15b, 1c, and 3b show that the proposed approach is able to completely eliminate all the unimportant branches and still preserve all main structure. Our method does not suffer from shortening main skeleton branches and it preserves the topology of the skeleton. Moreover, the obtained skeletons seem to be in accord with human perception. Figs. 1 and 3 show a comparison of our method and the method in [7]. The result obtained using the method in [7] also exhibits problems with the skeleton topology in Fig. 1b. Fig. 15 shows a comparison of our method with the method by Ogniewicz and Kübler [5]. It also illustrates that our pruning method can be used in pruning branches of the Voronoi skeleton. As the Voronoi skeleton points are symmetrical to the boundary sample points, the generating boundary points of each skeleton point are known.

Fig. 15c shows an application of our method to generate a fixed topology skeleton introduced in Golland and Grimson [11]. The proposed pruning is not limited to the DCE

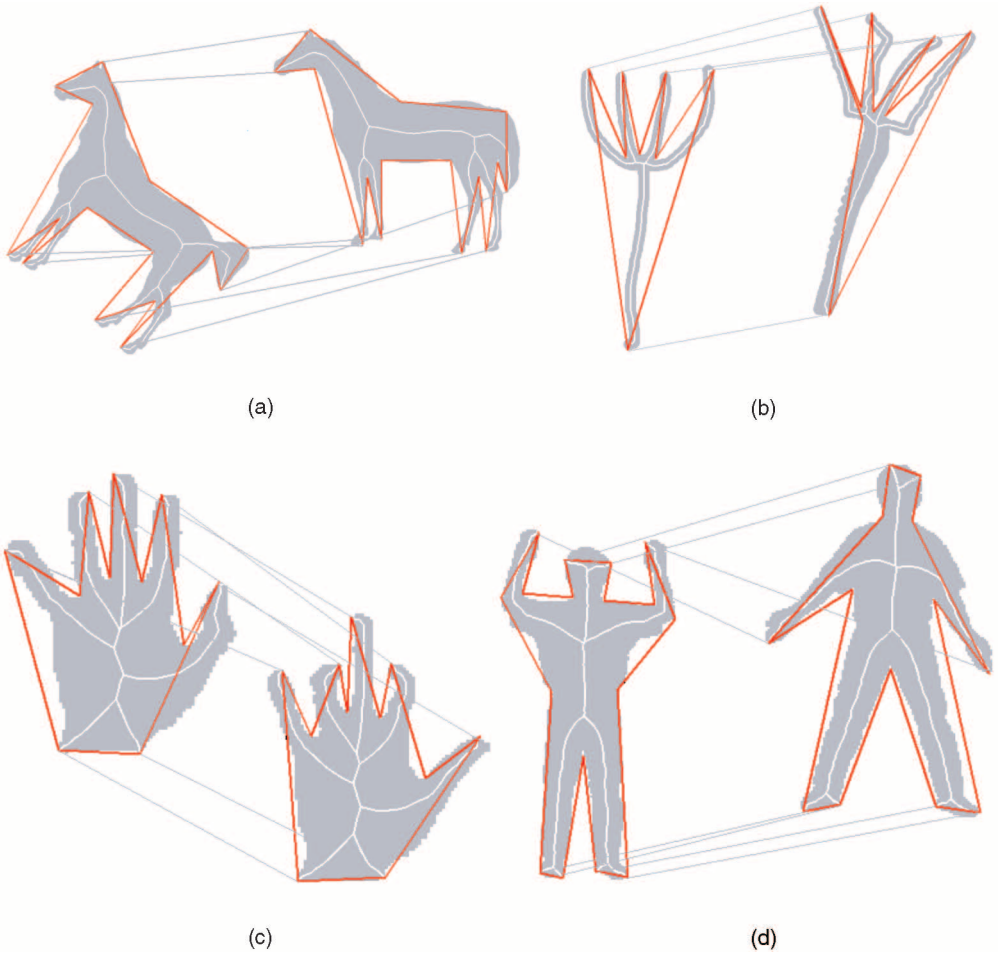

Fig. 18. The high quality of the pruned skeletons obtained by our method makes it possible to match the skeleton structure using existing shape similarity approaches.

induced contour partitioning. Once the positions of the skeleton's endpoints are estimated along the boundary as in the method in [11], the endpoints induce a partition of the boundary curve, and the fixed topology skeleton can be generated by pruning any skeleton with our method with respect to this partition.

The comparison between a result in [11] and our result is shown in Fig. 16. Fig. 16a shows a skeleton obtained by the method in [11], and Fig. 16b shows our result induced by the contour partition (A, B), (B, C), (C, D), (D, E), and (E, F) marked with the red points, which represent the estimated skeleton endpoints. We can see that the position of our skeleton is more accurate than in Fig. 16a since all of our skeleton points are the centers of maximal disks, which are exactly symmetrical to the shape boundary, and which is not the case for the fix topology method in [11]. Moreover, compared with [11], only the endpoints need to be estimated; we do not need to estimate the junction points of the skeleton.

Theorems 1 and 2 (in the Appendix which can be found at http://computer.org/tpami/archives.htm) prove that our method is guaranteed to preserve topology. We illustrate this fact in Fig. 2e above. Fig. 17 shows another example for a shape with three holes that has a total of four contour curves. The result of the method in [7] is shown in Fig. 17b. Fig. 17c shows that the proposed approach can preserves the original topology. In Fig. 17d, the contour partition is only composed of the four boundary curves, (i.e., there are no segments on any of these curves), so that the skeleton points must have their tangent points on the different boundary curves in order to remain.

7.3 The Potential in Shape Similarity

Our skeletons have strong potential for shape similarity, since, in addition to the above stated properties, they have two special properties: 1) Every skeleton branch is generated by contour parts divided by the vertices of the DCE simplified polygon. 2) The convex vertices of the DCE simplified polygon are the endpoints of the skeletons. Therefore, a contour-based shape similarity measure introduced in [17] can be used to match the obtained skeletons. Given a contour partition induced by DCE, the method in [17] establishes the optimal correspondence of the partition segments. Clearly, this also yields a correspondence of skeleton branches. This fact is illustrated in Fig. 18, where the corresponding skeleton branches are linked with lines. The correspondence in Fig. 18d is inspired by an example in Liu at al. example in [30], where complex graph matching algorithms are used to establish correspondences of skeleton braches. The quality of the skeletons obtained by the proposed pruning makes it

possible to apply existing contour similarity measures to problems with the structural similarity of skeletons.

8 Conclusions and Future Work

In this paper, we establish a unique correspondence between skeleton branches and subarcs of object contours. Based on these connections, a skeleton is pruned by removing skeleton branches whose generating points are on the same contour subarc. This has an effect of removing redundant skeleton branches and retaining all the necessary visual branches. We prove that this approach is guaranteed to preserve skeleton topology, does not shift the skeleton, and does not shrink the remaining branches. We use a discrete curve evolution to obtain a hierarchical partitioning of an object contour into subarcs that yields a hierarchical skeleton structure. We provide experimental results that demonstrate the high stability of the obtained skeletons even for objects with extremely complex shapes. The stability of skeletons is the key property required to measure the shape similarity of objects using their skeletons. The proposed definition of the skeleton pruning easily extends to higher dimensions, (e.g., in 3D it only requires a surface partition into patches), but further research on surface partitions is needed.

Acknowledgments

This work received support from the National Natural Science Foundation of China (grant No. 60273099) and was in part supported by the US National Science Foundation under Grant No. IIS-0534929. The authors would like to thank Liu Hairong and Yang Xingwei for their advice and useful discussions.

References

[1] H. Blum, "Biological Shape and Visual Science (Part I)," *J. Theoretical Biology,* vol. 38, pp. 205-287, 1973.
[2] K. Siddiqi, A. Shkoufandeh, S. Dickinson, and S. Zucker, "Shock Graphs and Shape Matching," *Proc. Int'l Conf. Computer Vision,* pp. 222-229, 1998.
[3] C. Di Ruberto, "Recognition of Shapes by Attributed Skeletal Graphs," *Pattern Recognition,* vol. 37, pp. 21-31, 2004.
[4] T.E.R. Hancock, "A Skeletal Measure of 2D Shape Similarity," *Computer Vision and Image Understanding,* vol. 95, pp. 1-29, 2004.
[5] R.L. Ogniewicz and O. Kübler, "Hierarchic Voronoi Skeletons," *Pattern Recognition,* vol. 28, no. 3, pp. 343-359, 1995.
[6] G. Malandain and S. Fernandez-Vidal, "Euclidean Skeletons," *Image and Vision Computing,* vol. 16, pp. 317-327, 1998.
[7] W.-P. Choi, K.-M. Lam, and W.-C. Siu, "Extraction of the Euclidean Skeleton Based on a Connectivity Criterion," *Pattern Recognition,* vol. 36, pp. 721-729, 2003.
[8] C. Pudney, "Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images," *Computer Vision and Image Understanding,* vol. 72, no. 3, pp. 404-413, 1998.
[9] W. Xie, R.P. Thompson, and R. Perucchio, "A Topology-Preserving Parallel 3D Thinning Algorithm for Extracting the Curve Skeleton," *Pattern Recognition,* vol. 36, pp. 1529-1544, 2003.
[10] F. Leymarie and M. Levine, "Simulating the Grassfire Transaction Form Using an Active Contour Model," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 1, pp. 56-75, Jan. 1992.
[11] P. Golland and E. Grimson, "Fixed Topology Skeletons," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 10-17, 2000.
[12] N. Mayya and V.T. Rajan, "Voronoi Diagrams of Polygons: A Framework for Shape Representation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 638-643, 1994.
[13] Y. Ge and J.M. Fitzpatrick, "On the Generation of Skeletons from Discrete Euclidean Distance Maps," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 11, pp. 1055-1066, Nov. 1996.
[14] C.M. Gold, D. Thibault, and Z. Liu, "Map Generalization by Skeleton Retraction," *Proc. ICA Workshop Map Generalization,* Aug. 1999.
[15] L.J. Latecki and R. Lakämper, "Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution," *Computer Vision and Image Understanding,* vol. 73, pp. 441-454, 1999.
[16] L.J. Latecki and R. Lakamper, "Polygon Evolution by Vertex Deletion," *Proc. Int'l Conf. Scale-Space '99,* 1999.
[17] L.J. Latecki and R. Lakamper, "Shape Similarity Measure Based on Correspondence of Visual Parts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 10, pp. 1185-1190, Oct. 2000.
[18] L.J. Latecki and R. Lakamper, "Application of Planar Shape Comparison to Object Retrieval in Image Databases," *Pattern Recognition,* vol. 35, no. 1, pp. 15-29, 2002.
[19] G. Borgefors, "Distance Transformations in Digital Images," *Computer Vision, Graphics, and Image Processing,* vol. 34, no. 3, pp. 344-371, 1986.
[20] D. Shaken and A.M. Bruckstein, "Pruning Medial Axes," *Computer Vision and Image Understanding,* vol. 69, no. 2, pp. 156-169, 1998.
[21] K. Siddiqi, A. Tannenbaum, and S.W. Zucker, "Hyperbolic 'Smoothing' of Shapes," *Proc. Int'l Conf. Computer Vision,* pp. 215-221, 1998.
[22] P. Dimitrov, J.N. Damon, and K. Siddiqi, "Flux Invariants for Shape," *Proc. Int'l Conf. Computer Vision and Pattern Recognition,* 2003.
[23] L.J. Latecki, R.-R. Ghadially, R. Lakämper, and U. Eckhardt, "Continuity of the Discrete Curve Evolution," *J. Electronic Imaging,* vol. 9, no. 3, pp. 317-326, July 2000.
[24] P. Dimitrov, C. Phillips, and K. Siddiqi, "Robust and Efficient Skeletal Graphs," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 1417-1423, 2000.
[25] K. Siddiqi, S. Bouix, A.R. Tannenbaum, and S.W. Zucker, "Hamilton-Jacobi Skeletons," *Int'l J. Computer Vision,* vol. 48, no. 3, pp. 215-231, 2002.
[26] A. Vasilevskiy and K. Siddiqi, "Flux Maximizing Geometric Flows," *IEEE Trans. Pattern Analysis Machine Intelligence,* vol. 24, no. 12, pp. 1565-1578, Dec. 2002.
[27] F.Y.L. Chin, J. Snoeyink, and C. An Wang, "Finding the Medial Axis of a Simple Polygon in Linear Time," *Proc. Sixth Int'l Symp. Algorithms and Computation,* pp. 382-391, 1995.
[28] J.W. Brandt and V.R. Algazi, "Continuous Skeleton Computation by Voronoi Diagram," *Computer Vision, Graphics, and Image Process,* vol. 55, pp. 329-338, 1992.
[29] S.C. Zhu and A. Yuille, "FORMS: A Flexible Object Recognition and Modeling System," *Proc. Int'l Conf. Computer Vision,* 1995.
[30] T. Liu, D. Geiger, and R.V. Kohn, "Representation and Self-Similarity of Shapes," *Proc. Int'l Conf. Computer Vision,* Jan. 1998.
[31] C. Aslan and S. Tari, "An Axis Based Representation for Recognition," *Proc. Int'l Conf. Computer Vision,* 2005.
[32] H.I. Choi, S.W. Choi, and H.P. Moon, "Mathematical Theory of Medial Axis Transform," *Pacific J. Math.,* vol. 181, no. 1, pp. 57-88, 1997.
[33] C. Arcelli and G. Sanniti di Baja, "Euclidean Skeleton via Center of Maximal Disk Extraction," *Image and Vision Computing,* vol. 11, pp. 163-173, 1993.
[34] C. Arcelli and G. Sanniti di Baja, "A Width Independent Fast Thinning Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 7, pp. 463-474, 1985.
[35] R. Kimmel et al. "Skeletonization via Distance Maps and Level Sets," *CVIU: Computer Vision and Image Understanding,* vol. 62, no. 3, pp. 382-391, 1995.
[36] T.B. Sebastian, P.N. Klein, and B.B. Kimia, "Recognition of Shapes by Editing Their Shock Graphs," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 5, pp. 550-571, May 2004.
[37] L.J. Latecki, R. Lakamper, and U. Eckhardt, "Shape Descriptors for Non-Rigid Shapes with a Single Closed Contour," *Proc. Conf. Computer Vision and Pattern Recognition,* 2000.
[38] F. Mokhtarian and A.K. Mackworth, "A Theory of Multiscale, Curvature-Based Shape Representation for Planar Curves," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, pp. 789-805, 1992.

[39] S.M. Pizer, W.R. Oliver, and S.H. Bloomberg, "Hierarchial Shape Description via the Multiresolution Symmetric Axis Transform," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 9, pp. 505-511, 1987.

[40] G. Borgefors, G. Ramella, and G. Sanniti di Baja, "Hierarchical Decomposition of Multiscale Skeletons," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 11, pp. 1296-1312, Nov. 2001.

[41] K. Siddiqi and B.B. Kimia, "Parts of Visual Form: Computational Aspects," *IEEE Trans. on Pattern Analysis and Machine Intelligence,* vol. 17, no. 3, pp. 239-251, Mar. 1995.

Xiang Bai received the BS degree in electronics and information engineering from Huazhong University of Science & Technology (HUST), Wuhan, China, in 2003 and the MS degree in electronics and information engineering from HUST in 2005. He is now an exchange student at Temple University. His research interests include computer graphics, computer vision, and pattern recognition.

Longin Jan Latecki received the master's degree in mathematics from the University of Gdansk, Poland, in 1985, and the PhD degree in computer science from the Hamburg University, Germany, in 1992. He is the winner of the Pattern Recognition Society Award together with Azriel Rosenfeld for "the most original manuscript from all 1998 Pattern Recognition issues." He received the main annual award from the German Society for Pattern Recognition (DAGM), the 2000 Olympus Prize. He cochairs the IS&T/SPIE annual conference series on vision geometry. He has published more than 100 research papers and books. He is an associate professor for computer science at Temple University in Philadelphia. His main research areas are shape representation and similarity, robot mapping, digital geometry and topology, data mining, and video analysis. He is a member of the IEEE Computer Society.

Wen-Yu Liu received the BS degree in computer science from Tsinghua University, Beijing, China, in 1986, and the Diploma and Doctoral degrees, both in electronics and information engineering, from Huazhong University of Science & Technology (HUST), Wuhan, China, in 1991 and 2001, respectively. He is now a professor and associate chairman of the Department of Electronics & Information Engineering, HUST. His current research areas include computer graphics, multimedia information processing, and computer vision.