An Efficient Quick Algorithm for Computing Stable Skeletons

Xiaojun Yang
Graduate School
Academy of Equipment Command and Technology
Beijing, China

Xiang Bai
Dept. of Electronics and Information Engineering
Huazhong University of Science and Technology
Wuhan, China

Xingwei Yang
Dept. of Computer and Information Sciences
Temple University
Philadelphia, USA

Luan Zeng
Key Lab of National Defense
Academy of Equipment Command and Technology
Beijing, China

***Abstract*—A new method to obtain high quality skeletons of binary shapes is proposed in this paper. First, a small set of salient contour points is computed by Discrete Curve Evolution (DCE). These salient points are the stable endpoints of the skeleton. Second, the skeleton is grown between pairs of the endpoints. Examining every eight-connected point of the current skeleton points, Selecte the point, that have equal distance to the contour parts which are partitioned by the two endpoints, as the new skeleton point. The skeleton path continues growing in this way until it reaches the other endpoint or another skeleton branch. The main idea is that the skeleton points are always the center of the maximal disks, and the endpoints of the skeleton are those contour points with high global curvature which is stable to noise and shape variations. The obtained skeletons are in accord with human visual perception and stable, also connected and one pixel thick. They do not require any pruning or any other post-processing. The experimental results clearly demonstrate that the proposed method significantly outperforms other well-known methods for skeleton computation.**

Keywords-skeleton; shape; Discrete Curve Evolution; endpoint; skeleton path; stable; one-pixel wide

I. INTRODUCTION

Skeleton, or Medial Axis, has been widely used for shape analysis and object recognition, such as image retrieval and computer graphics, character recognition, image processing, and analysis of biomedical images. Skeleton-based representations are the abstraction of objects, which contain both shape features and topological structures of original objects. Due to the importance of the skeleton, many skeletonization algorithms have been developed to represent and measure different shapes. However, as the skeleton is sensitive to the noise and deformation of the boundary, which may seriously disturb the topology of the skeleton graph, these methods cannot work on complex shapes or shapes with obvious noise. The most common approaches to overcome skeleton instability are based on skeleton pruning. In Bai's method [1], the algorithm utilized the Discrete Curve Evolution (DCE) to simplify the contour of the image and then pruned skeleton by contour partitioning. The results are very excellent, but the time cost for extraction of skeleton and matching skeleton trees/graphs cannot satisfy the requirement of fast shape retrieval [2].

The skeleton of a single connected shape that is useful for skeleton-based recognition should have the following properties:

(1) it should preserve the topology of the original object,

(2) it should be stable under deformations,

(3) it should be invariant under Euclidean transformations such as rotations and translations,

(4) it should contain the centers of maximal disks, and nothing more than the centers of maximal disks, which can be used for reconstruction of original object,

(5) it should represent significant visual parts of objects. It means that there should be skeleton branches in every significant object part and that there should be no spurious branches that do not correspond to any object parts (which are usually due to noise).

Skeletonization approaches can be broadly classified into four types: thinning algorithms [3,4], discrete domain algorithms based on the Voronoi diagram [5,6], algorithms based on distance transform [7,8], and algorithms based on mathematical morphology [9,10]. 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). The second drawback of the methods described above is that main skeleton branches are shortened and short skeleton branches are not removed completely. 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.

The method introduced by Bai et al. [1] can obtain excellent skeletons which contain most of the properties of ideal skeletons, but it cannot guarantee that the skeleton is one-pixel

978-1-4244-4131-0/09/$25.00 ©2009 IEEE

wide and it need the postprocessing, also, the time cost for skeleton computing and pruning is very large. Recently, Bai et al. present a novel skeleton computing algorithm with particle filters [11]. The algorithm utilized DCE polygon simplification to compute the salient points, then the skeleton is grown between pairs of salient points by examining the posterior probabilities of the skeleton particles. It can obtain excellent skeletons of simple connected shapes, but sometimes, it can encounter difficulties when dealing with complexly shape, especially a shape with holes, also it is time consuming. The main goal of this paper is to present a method that extracts the exact skeleton which will achieve all the above properties. The computation is composed of two major parts. First, a small set of salient contour points is computed. Second, the skeleton is grown between pairs of the endpoints. Our proposed method is easy to implement and can be computed efficiently. The experimental result is shown in Fig. 1. Compared with [11], our method produces one-pixel thick skeletons faster (The method in [11] cannot always guarantee that the skeleton is one-pixel wide, see the horse tail in Fig.1 (a).). Moreover, we can obtain stable skeletons of a shape with holes (c).

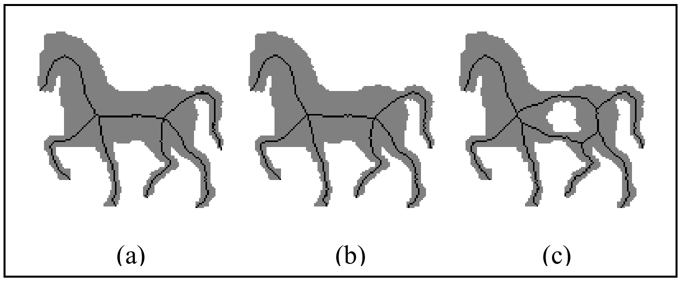

Figure 1. (a) skeleton computed by the method in [11]. (b), (c) by the proposed method.

In the following section, we describe our method in section 2. The experimental results and the comparison are shown in Section 3. Finally, the conclusion is presented in Section 4.

II. Description of our Algorithm

The proposed method first utilizes the Discrete Curve Evolution (DCE) [12] to simplify the contour, and to obtain a small set of salient points as the skeleton endpoints, but other approaches which produce stable salient points could also be used. Then, the skeleton is grown between pairs of the endpoints with greedy algorithm. Our proposed method is easy to implement and can be computed efficiently.

We benefit from a geometric relation between the skeleton and the contour, which is a key observation that motivates our approach: the endpoints of significant skeleton branches coincide with convex salient contour points. We illustrate the main ideas of the proposed method in Fig. 2. Let a and b be two salient contour points. They divide the contour into two parts $C = C_1 \cup C_2$ marked with red and blue colors, respectively. The skeleton path p(a, b) from a to b is composed of centers of maximal disks that are tangent both to C_1 and to C_2. We grow the skeleton path p(a, b) from a to b by examining every eight-connected point of the current skeleton points, and selecting the most conformable point as the new skeleton point. The final skeleton consists of the skeleton paths between all pairs of salient points.

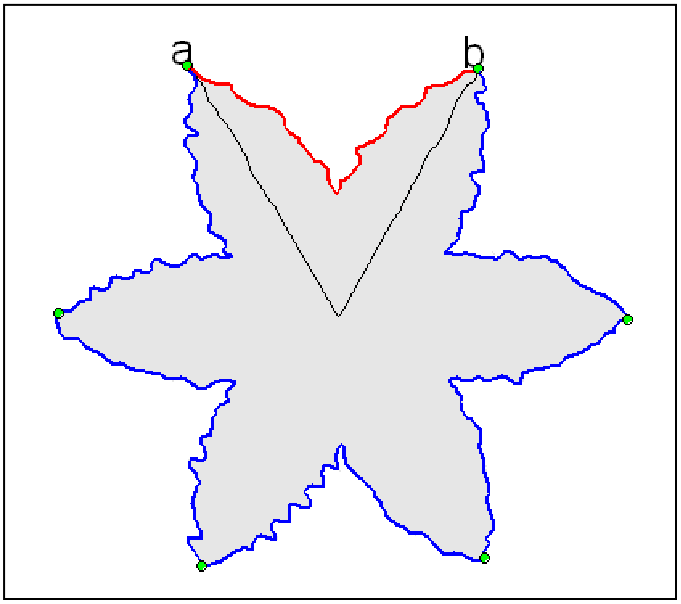

Figure 2. Growing a single skeleton path p(a, b) from a to b computed by our algorithm.

A. *Computing the salient points*

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 DCE [12] is simple: In every evolutional step of DCE, 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 order of the substitution is determined by the relevance measure K given by:

$$K(s_1, s_2) = \frac{\beta(s_1, s_2) l(s_1) l(s_2)}{l(s_1) + l(s_2)}. \quad (1)$$

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 by the total length of a polygonal curve C. 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. During the evolution, we will first remove the arcs with the smallest contribution. Fig. 2. shows salient contour points (the green dot) computed by DCE . In Fig. 3., we show some results to illustrate that each convex vertex of the DCE simplified polygon is guaranteed to be a skeleton endpoint. At different curve evolution stages, the number of skeleton endpoints was reduced. So we can obtain a hierarchical skeleton structure.

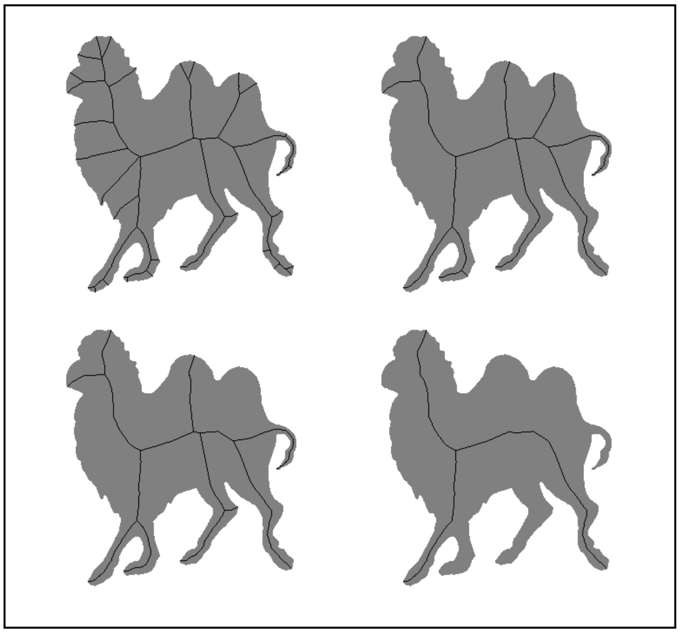

Figure 3. Hierarchical skeleton of camel.

B. *Growing skeleton paths*

Let a and b be two convex, salient contour points. As stated above, we use DCE polygon simplification to compute the salient points, since all convex vertices of the DCE simplified polygon are guaranteed to be skeleton endpoints. Our goal is to obtain a skeleton path from a to b. We use x_i to denote the skeleton points at time step i (initialize $x_1 =$ a). Let $N(xi)$ represent the set of 8-nearest neighbors of x_i. we recall that the contour is divided into two parts C_1 and C_2 by a and b. Let d_1, d_2 represent the minimum distance from the point x_k ($x_k \in N(x_i)$ and x_k not in $x_1 \sim x_i$) to each of the parts, which for a correct skeleton paths both should be equal to the radius of the maximal disk centered at x_k . In particular, we should have $d_1 = d_2$. The growing of the skeleton path is very brief: we select the most conformable point as the new skeleton point x_{i+1}.

$$x_{i+1} = \arg\min_{x_k} \left| d_1 - d_2 \right| . \quad (2)$$

Fig. 2 shows an example of one skeleton path generated by the above algorithm. The skeleton path is in the middle of the two contour parts, which is the main property of an excellent skeleton. Moreover it does not have any redundant branches and is insensitive to boundary noise. The unique selection assures that the skeleton path is one pixel thick.

Let N be the amount of salient contour points, we grow skeleton paths from salient point i+1 to i (i=1,2,…,N-1). The skeleton is the combination of skeleton paths between the salient points. If we have generated one path of the skeleton, the other paths of the skeleton will be generated in the similar way. The only difference is that when the generating skeleton path meets the generated skeleton path, it should stop. This can preserve the property of the one-pixel wide and keep the connectivity of the skeleton. For example, to compute the skeleton of the heart, the skeleton path from 2 to 1 is first generated, in Fig. 4 (a). Then, we grow skeleton path from 3 to 2, but the growing stops when it meets the black generated skeleton path. The white skeleton path in Fig. 4 (b) shows the result. Combine the black skeleton path and the white to form a complete skeleton.

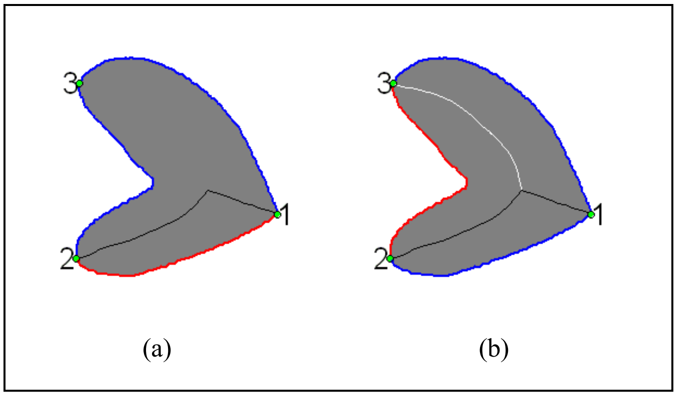

Figure 4. The process of skeleton growing.

C. *Shape with holes*

To generate the skeleton for the shape with holes, all the boundaries should be extracted. We consider shape with one hole. The contour is divided into three parts: $C = C_1 \cup C_2$ divided by a,b (C is the longest outer contour) and the contour of the hole C_3. Let d_1, d_2, d_3 represent the minimum distance from the candidate skeleton point x_k to C_1, C_2, C_3 respectively. To grow the skeleton path from a to b, first we compute the center of C_1, C_2 as stated above. But when $d_3 \le d_1$ or $d_3 \le d_2$, the skeleton path should be separated into two paths. One generate in the center of C_1 and C_3, the other in the center of C_2 and C_3. For example, in Fig.5 (a), the black skeleton path was first computed, then we generated the white path. Fig.5 (b) is the whole skeleton.

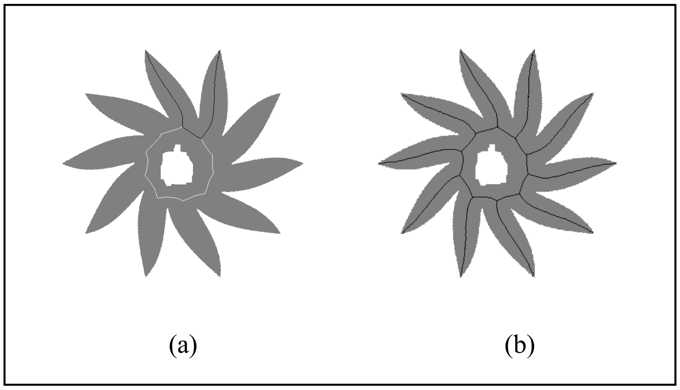

Figure 5. Obtain the skeleton of shape with hole.

III. EXPERIMENTS

In this section, we evaluate the performance of the proposed method in three parts. In Section 3.1, we experiment our method on varied shapes; In Section 3.2, we demonstrate the stability to shape deformations and contour noise; In Section 3.3, we provide a comparison to other methods. From all of the results listed below, we can state that the proposed approach can generate excellent skeletons which satisfy the five properties listed in Introduction. Besides, according to the comparison experiments, the proposed method can obtain much better skeletons than many other approaches.

A. Test on varied shapes

The results in Fig.6 and Fig.7 represent that the proposed method can get the skeleton from different symbols, no matter it is astronaut, fighter plane, satellite, airplane, missile or varied character. The results prove that our algorithm is stability to variance.

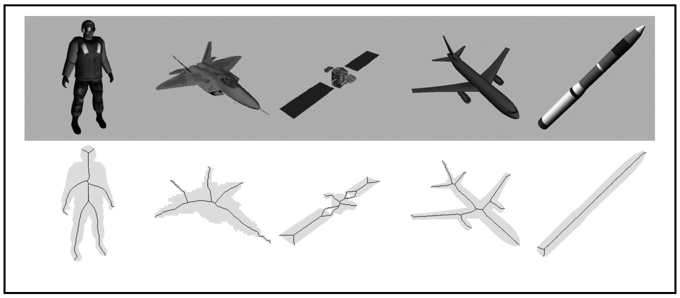

Figure 6. The skeletons of varied shapes.

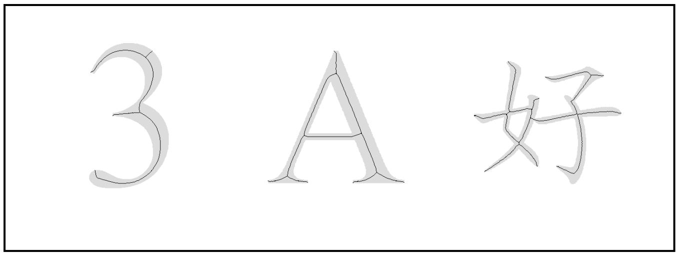

Figure 7. The Skeletons of Arabic numerals, English symbols and Chinese character.

B. Test on noisy images

The results in Fig. 8 show that the proposed method is insensitive to even substantial noise in contours. For each shape, there is one image without noise [13] and one image with substantial noise. The similarity of the obtained skeletons illustrates the stability of the proposed method. In particular, there are no branches generated by the boundary noise, and the skeletons still preserve the topological and geometric structure of the objects.

C. Comparison with other methods

In Fig.9, it's easy to observe that our result is better than other results. We never shorten the skeleton branches, and we can assure one-pixel wide skeleton. Moreover it is faster: Our method only needs 1 second to obtain the whole skeleton, but 6s by Particle Filters. Fig.10 gives the skeletons of a Chinese character. Our grown skeleton in Fig.10 (c) is better than the thinning result (b) obtained with a morphological thinning and the pruned skeleton (a) by DSE [14].

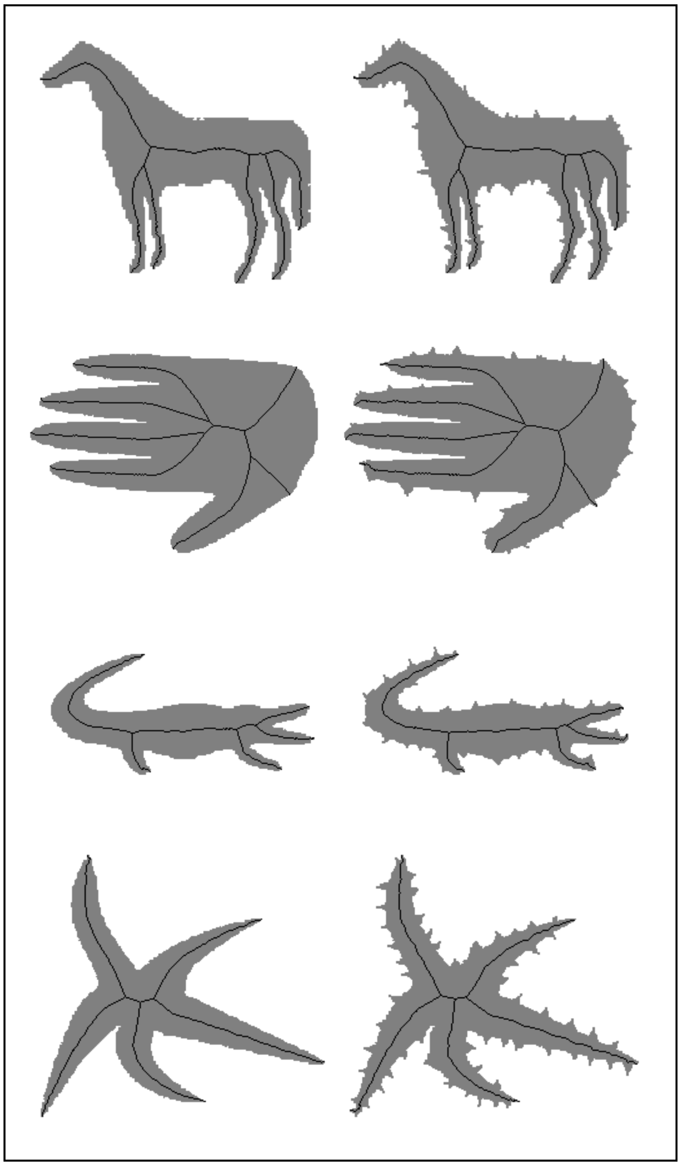

Figure 8. For each shape, there is one image without noise and one image with substantial noise. The obtained skeletons are very similar.

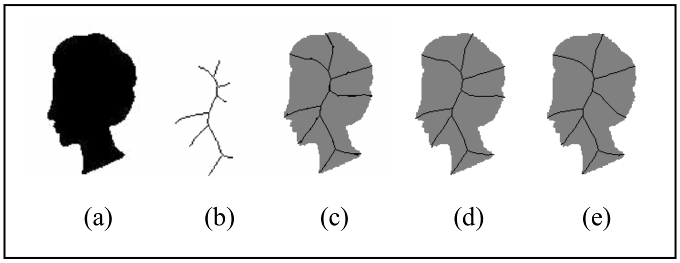

Figure 9. Comparison: (a) is the example original shape, (b) is Tosello's result [10], (c) is computed by DSE [14], (d) is grown by Particle Filters [11], and (e) is our result.

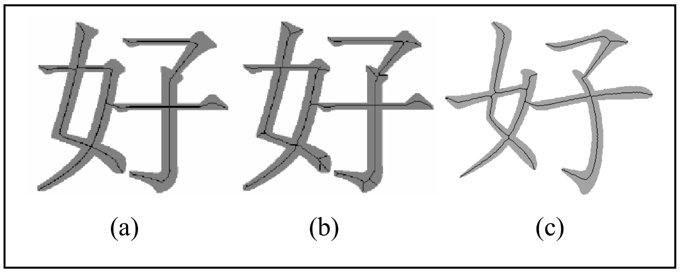

Figure 10. The Skeletons of a Chinese character. (a) is the pruned skeleton by DSE [14], (b) is the result of morphological thinning, and (c) is obtained by our method.

IV. Conclusion

In this paper, we generate skeletons from the salient contour points computed by Discrete Curve Evolution. The obtained skeletons do not have redundant skeleton branches and retain all the necessary visual branches. The experimental results demonstrate high stability of the obtained skeletons even for objects with extremely noisy contours, which is the key property required to measure the shape similarity of objects using their skeletons. Moreover, this method can generate the skeleton for the shape with holes and guarantee the skeleton is one-pixel wide. In future, we will extend the proposed approach to 3D shapes.

References

[1] X. Bai, L.J. Latecki, and W.Y. Liu, "Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution", IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), 29(3), 2007, pp. 449-462.

[2] X. Bai and L.J. Latecki, "Path Similarity Skeleton Graph Matching", IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), 30(7), 2008, pp. 449-462.

[3] X.W. Yang, X. Bai, X.J. Yang, and W.Y. Liu, "An Efficient Quick Thinning Algorithm", 2008 Congress on Image and Signal Processing (CISP 2008), Sanya, China, 2008, pp.475-478.

[4] F. Leymarie and M. Levine, "Simulating the grassfire transaction form using an active Contour model", IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), 14(1), 1992, pp.56-75.

[5] N. Mayya and V.T. Rajan, "Voronoi Diagrams of polygons: A framework for Shape Representation", In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1994, pp. 638-643.

[6] J.W. Brandt and V.R. Algazi, "Continuous skeleton computation by Voronoi diagram", Comput. Vision, Graphics, Image Process 55, 1992, pp.329-338.

[7] G. Malandain, and S. Fernandez-Vidal, "Euclidean skeletons", Image and Vision Computing 16, 1998, pp.317-327.

[8] W.P. Choi, K.M. Lam and W.C. Siu, "Extraction of the Euclidean skeleton based on a connectivity criterion", Pattern Recognition 36, 2003, pp.721-729.

[9] P. Dimitrov, J.N. Damon, and K. Siddiqi, "Flux Invariants for Shape" In: Int. Conf. Computer Vision and Pattern Recognition, 2003.

[10] A. Torsello and E.R. Hancock, "Correcting Curvature-Density Effects in the Hamilton-Jacobi Skeleton", IEEE Transaction on Image Processing 15(4), 2006, pp.877–891.

[11] X. Bai, X.W. Yang, L.J. Latecki, Y.B. Xu, and W.Y. Liu, "Computing Stable Skeletons with Particle Filters", Tenth Pacific Rim International Conference on Artificial Intelligence (PRICAI 2008), Vietnam,Hanoi, 2008, pp.30-41.

[12] L.J. Latecki and R. Lakamper, "Shape similarity measure based on correspondence of visual parts", IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI), 22, 2000, pp.1185-1190.

[13] C. Aslan and S. Tari, "An Axis-Based Representation for Recognition", Tenth IEEE International Conference on Computer Vision (ICCV 2005), pp.1339-1346.

[14] X. Bai and L.J. Latecki, "Discrete Skeleton Evolution", 6th Int. Conf. on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), EZhou, China, August, 2007, pp.362-374.