

Computing Stable Skeletons with Particle Filters

Xiang Bai¹, Xingwei Yang², Longin Jan Latecki², Yanbo Xu¹, and Wenyu Liu¹

¹ Department of Electronics and Information Engineering, Huazhong University of Science and Technology, Wuhan, 430074, China

{xiang.bai,xuyanbohust}@gmail.com, liuwy@hust.edu.cn

² Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19022, USA

{xingwei.yang,latecki}@temple.edu

Abstract. We present a novel method to obtain high quality skeletons of binary shapes. The obtained skeletons are connected and one pixel thick. They do not require any pruning or any other post-processing. The computation is composed of two major parts. First, a small set of salient contour points is computed. We use Discrete Curve Evolution, but any other robust method could be used. Second, particle filters are used to obtain the skeleton. The main idea is that the particles walk along the skeletal paths between pairs of the salient points. We provide experimental results that clearly demonstrate that the proposed method significantly outperforms other well-known methods for skeleton computation.

Keywords: Skeleton, shape, pruning, skeletal paths, particle filters.

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]. The skeleton is an abstraction of objects that at the same time contains both shape features and topological structures of the original object. Therefore, many researchers have worked on matching skeleton structures represented by graphs or trees [2,3,4,5,6]. 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.

We list now properties of the ideal skeleton mentioned in [7].

- (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) the position of the skeleton should be accurate
- (5) it should be composed of 1D arcs (i.e., one-pixel wide in digital images)
- (6) it should represent significant visual parts of objects

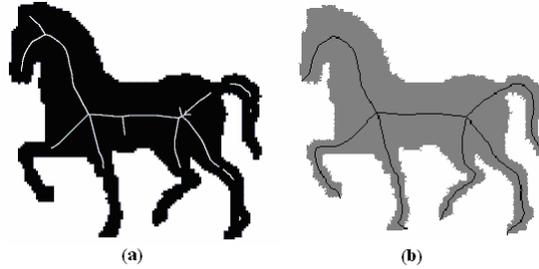


Fig. 1. (a) skeleton computed by the method in [11], (b) by the proposed method

Properties (4) and (5) mean that the skeleton should contain the centers of maximal disks, and nothing more than the centers of maximal disks. Property (6) 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).

Since most of the existing skeleton computation methods are not able to produce skeletons that satisfy property 6, skeleton pruning is applied. Its goal is to remove spurious branches. Clearly, a pruned skeleton should still have properties (1)-(6), which can be will useful for shape matching and recognition [6,8].

Ogniewicz and Kübler [9] presented a few significant measures for pruning complex Voronoi skeletons without disconnecting the skeletons, but it may lead to topology violation. The method in [10] has difficulty in distinguishing noise from low frequency shape information on boundaries. The skeleton generated by [11] cannot guarantee the property of the connectivity, as shown in the experimental results in Fig. 1(a). The skeleton computed by our method is shown in Fig. 1(b).

The method introduced by Bai et al. [7,12] can obtain excellent skeletons which contain most of the properties of ideal skeletons, but it cannot guarantee that the skeleton is one-pixel wide and it need the postprocessing. Compared to it, our method produces one-pixel thick skeletons without skeleton pruning.

Particle filters estimate the posterior probability density over the state space of a dynamic system. The key idea of this technique is to represent probability densities by sets of samples. By sampling in proportion to likelihood, particle filters focus the computational resources on regions with high likelihood, where good approximations are most important. Over the last few years, particle filters have been applied with great success to a variety of state estimation problems including visual tracking, speech recognition, mobile robot localization, robot map building, people tracking, and fault detection. Moreover, Adluru et al have used particle filters in contour grouping [13]. The proposed method is the first one that utilizes particle filters in computing skeletons.

The proposed method first utilizes the Discrete Curve Evolution (DCE) [14] to simplify the contour, and to obtain a small set of salient points as vertices of the simplified polygon, but other approaches which produce stable salient points could also be used. The basic idea of the DCE is simple. In every evolutionary step of DCE, a pair of consecutive line segments s_1, s_2 is replaced by a single

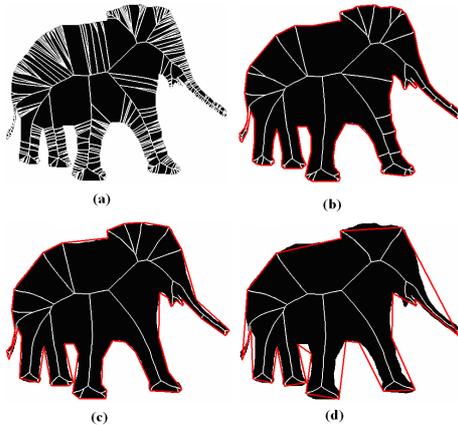


Fig. 2. Hierarchical skeleton of elephant obtained by pruning the input skeleton (left) with respect to contour segments obtained by the Discrete Curve Evolution (DCE). The outer (red) polylines show the corresponding DCE simplified contours.

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. In Fig. 2, we show some results to illustrate that each convex vertex of the DCE simplified polygon is guaranteed to be a skeleton endpoint.

We benefit from a geometric relation between the skeletal path 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. 3. 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 use a particle filter to compute the path $p(a, b)$. The condition that the maximal disks are tangent to two contour parts makes our skeleton insensitive to noise and contour deformations. The computation with particle filters assures that the skeleton paths are connected, vary smoothly, and are one pixel thick.

The final skeleton consists of the skeleton paths between all pairs of salient points. For a given set of salient contour points, we obtain an excellent skeleton without any pruning process. We use DCE to generate salient points, since it is

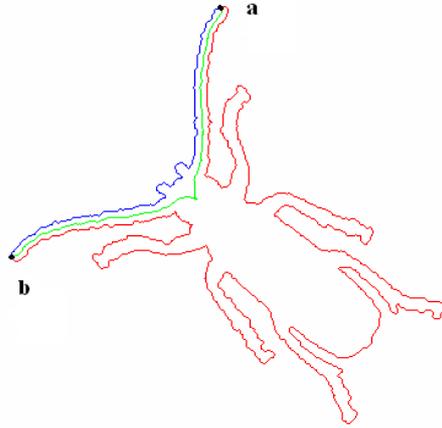


Fig. 3. In green a single skeleton path $p(a, b)$ from a to b computed by our algorithm. The salient points a and b divide the contour into two parts $C = C_1 \cup C_2$ marked with red and blue colors, respectively. $p(a, b)$ is composed of centers of maximal disks that are tangent both to C_1 and to C_2 .

proved in Bai et al. [7], each DCE computed convex salient point is guaranteed to be a skeleton endpoint.

As in our case the target function is nonlinear, the Dynamic Programming (DP), which can only solve the linear function, will carry contour noise to the skeleton. Compared to DP, the particle filter can get rid of the noise and local solutions. It can allow branching and carrying multiple solutions. Therefore, We use a particle filter to find the skeleton path between any pair of the salient points instead of DP. Particle filters are also known as Sequential Monte-Carlo (SMC) methods, which have the ability to carry multiple hypotheses, and are widely used to track multiple targets with cluttered background in image sequences. The first application of particle filters in Computer Vision is in the tracking of object contours (Isard and Blake [15,16]). Tracking of motion boundaries is used for motion estimation in [17]. The first application of particle filters to static images is presented in Pérez et al. [18], where particle filters are applied to perform inference over a spatial chain of edge pixels rather than over a temporal chain. An extension of SMC that performs inferences on arbitrarily structured graphical models has been proposed in [19,20] and applied to an edge linking task in [19].

The rest of the paper is organized as follows: our approach to computing skeleton paths is introduced in Section 2. The construction of the whole skeleton is presented in Section 3. The experimental results are shown in Section 4. Finally, the conclusion is presented in Section 5.

2 Computing Skeleton Paths with a Particle Filter

Let a and b be two convex, salient contour points. As stated in the introduction, 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 $x_{1:t}^j$ to denote a sequence of skeleton points of particle j at time step t , i.e., $x_{1:t}^j = x_1^j, \dots, x_t^j$. Then x_t^j is the current endpoint of the particle j at the step t . Let $N(x_t^j)$ represent the set of 8-nearest neighbors of all of skeleton points of particle j .

We initialize with n particles, each equal to a , and the initial weights of the particles are $1/n$. At each iteration, we consider eight possible continuations of particle $x_{1:t-1}^j$ as the 8-nearest neighbors of x_{t-1}^j . (Here we benefit from the fact that a digital image is a discrete structure.) We obtain an eight extensions of particle $x_{1:t}^k = \{x_{1:t-1}^j, x_t^k\}$ for each of the eight neighbors $x_t^k \in N(x_{t-1}^j)$. The index k of particle $x_{1:t}^k$ may be different from j , since particle j has 8 extensions corresponding to the 8 neighbors $N(x_{t-1}^j)$ of x_{t-1}^j .

Now we derive a particle filter algorithm that is particularly suitable for computation in digital images. Our goal is to estimate the posterior $p(x_{1:t}|z_{1:t})$ over all potential skeleton paths in a given shape. Our observations $z_{1:t} = \{z_1, z_2, \dots, z_t\}$ represent distances to the shape contour (a detailed definition follows below). Each particle represents a particular skeleton path. We will follow the framework of a particle filter algorithm called sampling importance resampling (SIR) filter [21], which can be summarized as follows:

1) *Prediction by Sampling*: The next generation of particles $\{x_{1:t}^k\}_k$ is obtained from the generation $\{x_{1:t-1}^j\}_{j-1}$ by sampling from a proposal distribution π (defined below).

We use prior boosting in prediction by sampling (Gordon et al., [22]). It allows us to capture multi-modal likelihood regions in the posterior. In prior boosting we sample more than one follower for each particle so that different followers can capture different modes of the proposal. As described above, the fact that we work in digital images naturally suggests the eight followers be the eight neighbors of the latest pixel in each particle sequence. Thus, we increase the number of particles from N to $8N$, which is then reduced back to N in the resampling step (3).

2) *Importance Weighting*: An importance weight is assigned to each particle according to the importance sampling principle $w_t^k = \frac{p(x_{1:t}^k|z_{1:t})}{\pi(x_{1:t}^k|z_{1:t})}$. The weights account for the fact that the proposal distribution is usually not equal to the target distribution $p(x_{1:t}|z_{1:t})$.

3) *Resampling*: Particles are drawn with replacement proportional to their importance weights. The weight of each of the eight new particles is defined as:

$$w_t^k = \frac{p(x_{1:t}^k|z_{1:t})}{\pi(x_{1:t}^k|z_{1:t})} = \frac{\eta p(z_t|x_{1:t}^k, z_{1:t-1}) p(x_t^k|x_{t-1}^j) p(x_{1:t-1}^j|z_{1:t-1})}{\pi(x_t|x_{1:t-1}^j, z_{1:t}) \pi(x_{1:t-1}^j|z_{1:t-1})} \quad (2)$$

$$\propto \frac{p(z_t|x_t^k) p(x_t^k|x_{t-1}^j)}{\pi(x_t|x_{1:t-1}^j, z_{1:t})} w_{t-1}^j, \quad (3)$$

where w_{t-1}^j is the weight of particle x_{t-1}^j and $\eta = 1/p(z_t|z_{1:t-1})$ is a normalization factor resulting from Bayes rule that is equal for all particles. Now we make

an important assumption that the proposal distribution $\pi(x_t|x_{1:t-1}^j, z_{1:t})$ is uniform. This is justified in our context by the fact that each point is a pixel that has eight neighbors, and continuation to each of the eight neighbors is equally probable. Therefore, we obtain

$$w_t^k \propto p(z_t|x_t^k)p(x_t^k|x_{t-1}^j)w_{t-1}^j \quad (4)$$

The conditional probabilities in equation (3) are defined below based on digital topology of paths in digital images $p(x_t^k|x_{t-1}^j)$ and on geometric properties of skeletons $p(z_t|x_t^k)$.

The conditional probability of the new particle $x_{1:t}^k$ generated by extending the j th particle is given by:

$$p(x_t^k|x_{1:t-1}^j) = \begin{cases} 1, & \text{if } x_t^k \in N(x_{t-1}^j) - N(x_{1:t-1}^j) \\ 0.01, & \text{else} \end{cases} \quad (5)$$

The main contribution of this probability is to avoid visiting the same pixels again, since we do not want the particle path to go backward, which would create a loop in the skeleton path or perturb it. Hence we assign very low probability to the neighbors of x_{t-1}^j that already belong to the sequence of particle $x_{1:t-1}^j$.

In order to calculate $p(z_t|x_t^k)$, we recall that the contour is divided into two parts C_1 and C_2 . Let d_1, d_2 represent the minimum distance from the point x_t^k 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_t^k . In particular, we should have $d_1 = d_2$. Thus, our observation z_t is composed of two distances d_1, d_2 from the contour parts C_1 and C_2 . Fig. 4 illustrates our computation of $p(z_t|x_t^k)$. Consider two different points P_1 and P_2 as candidates for the skeleton point x_t^k . It is obvious that P_1 is more likely to be the center of a maximal disk with respect to the contour parts C_1 and C_2 than P_2 , since $D' = |d_1' - d_2'|$ is smaller than $D = |d_1 - d_2|$. Therefore, we assume that the observation density is a Gaussian function of the difference $d_1 - d_2$:

$$p(z_t|x_t^k) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d_1-d_2)^2}{2\sigma^2}} \quad (6)$$

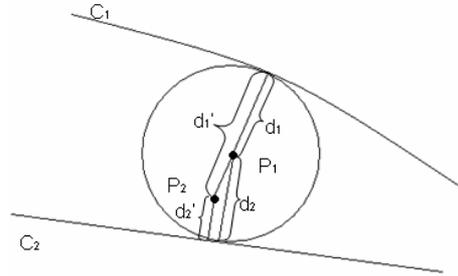


Fig. 4. Point P_1 is more likely to be a skeleton point than point P_2

The outline of the derived particle filter algorithm is as follows:

From the "old" sample set $\{(x_{t-1}^j, w_{t-1}^j) : j = 1, \dots, N\}$ at the time step $t-1$, construct a new sample set $\{(x_t^j, w_t^j) : j = 1, \dots, N\}$ for step t .

For $j = 1$ to N iterate steps (1)-(3):

(1) Prediction by Sampling

For each particle j , we extend it to eight particles by $x_{1:t}^k = \{x_{1:t-1}^j, x_t^k\}$, where $x_t^k \in N(x_{t-1}^j)$.

2) Importance Weighting

Compute the weights $w_t^k = p(z_t | x_t^k) p(x_t^k | x_{t-1}^j) w_{t-1}^j$ and normalize the weights so that $\sum_k w_t^k = 1$.

3) Subsampling

Draw N particles from the current set of $8N$ particles with probabilities proportional to their weights.

Finally, the particle with the highest weights is selected, which represents a skeleton path. There are two important differences in comparison to the standard sampling importance resampling (SIR) filter. First, our prediction by sampling considers all possible extensions to the eight neighbors, this is why our proposal distribution is uniform. Second, since our prediction by sampling increases the number of particles to $8N$, we replaced resampling with subsampling in order to reduce the number of particles to N . We modified the residual resampling to obtain the residual subsampling.

Fig. 3 shows an example of one skeleton path generated by the above algorithm. The blue and red parts represent the two different parts C_1, C_2 of the contour separately, which are divided by the two vertices. The green line is the skeleton path generated by our algorithm. The skeleton path is in the middle of the two contour parts, which is the main property of an excellent skeleton. The skeleton path does not have any redundant branches and it is insensitive to boundary noise. These properties follow from the fact that the observation density $p(z_t | x_t^k)$ is computed with respect to the contour partitions C_1 and C_2 induced by two salient points. The conditional probability $p(x_t^k | x_{t-1}^j)$ is responsible for computing smooth paths that are one pixel thick. The statistical framework of particle filter assures that the local noise on pixel level does not distort the skeleton paths.

3 Combining Skeleton Paths to Form a Complete Skeleton

The skeleton is the combination of skeleton paths between the end nodes. 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, the skeleton of the heart in Fig. 5 (a) is the skeleton of

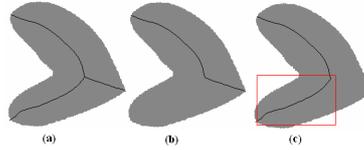


Fig. 5. The skeleton in (a) is constructed by combining the paths in (b), (c) together

the heart. The skeleton path of Fig. 5 (b) is first generated. Then, instead of combining the whole skeleton path in Fig. 5 (c), the propose approach will only take part of the skeleton path of it, which is surrounded by the red rectangle.

4 Experiments

In this section, we evaluate the proposed method in two parts: 1) we show that the skeleton is stable to noise and deformation and 2) we compare it to other methods. From all of the results listed below, we can state that the proposed approach can generate excellent skeletons which satisfy the six properties listed in Introduction. Besides, according to the comparison experiments, the proposed method can obtain much better skeletons than many other approaches.

4.1 Test on Noisy Images

The results in Fig. 6 show that the proposed method is insensitive to even substantial noise in contours. For each shape, there is one image without noise 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. Other methods cannot obtain stable skeletons on noisy images. Most of them will have extra branches or distorted skeletons. The extraordinary stability of our skeletons in the presence of large inner-class shape variations is demonstrated in Fig. 7.

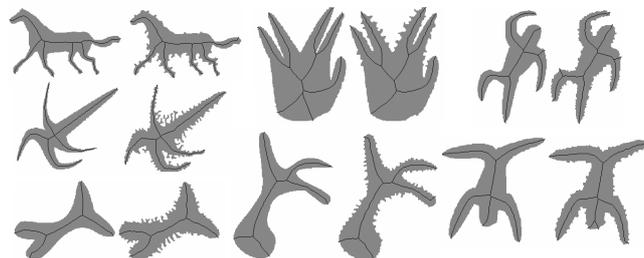


Fig. 6. For each shape, there is one image without noise and one image with substantial noise. The obtained skeletons are very similar.

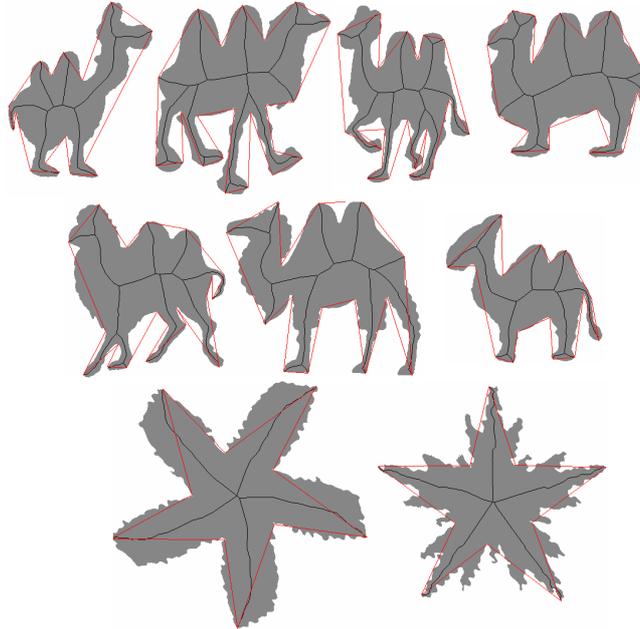


Fig. 7. The results on some shapes from the MPEG-7 database [14] illustrate extraordinary stability of our skeletons in the presence of large shape variances. The red lines illustrate the DCE polygons.

Although the objects differ significantly from each other, the obtained skeletons have the same global structure. Moreover, the thin tails of the camels remained in the skeleton, which cannot be achieved by most of the other pruning methods, since they may shorten or disconnect the skeleton. The final DCE simplified polygons are also shown overlaid on the shapes with red segments.

4.2 Comparing to Other Methods

We compare our method to the fixed topology method in [23], which also starts with a small set of salient points. However, the fixed topology skeleton requires also that the skeleton junction points are estimated. We do not need to estimate the junction points. Two example results of [23] are shown in Fig. 8(a),(c). As can be clearly seen, the obtained skeleton is not positioned accurately in that many skeleton points are not centers of maximal disks. In contrast, as shown in Fig. 8(b),(d) all of our skeleton points are the centers of maximal disks, and therefore they are exactly symmetrical to the shape boundary. In addition, observe the presence of phantom horizontal skeleton branches in Fig. 8(c). They do not reflect any real structural information. Due to the stability of DCE, the proposed method does not introduce any phantom branches.

Fig. 9 shows a comparison of our approach (b) with the method in [9] (a), which has inaccurate, half-shortened branches that are not related to any obvious

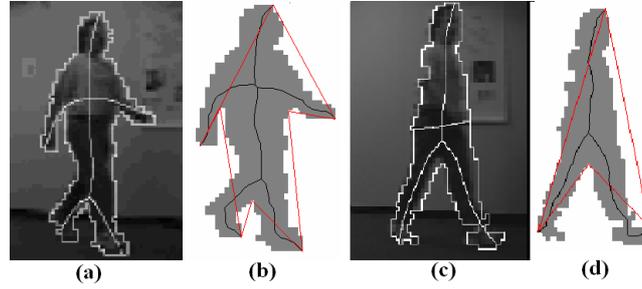


Fig. 8. Comparison between the fixed topology skeleton in [23] in (a), (c) and our skeleton in (b), (d). The red lines illustrate the DCE polygons.

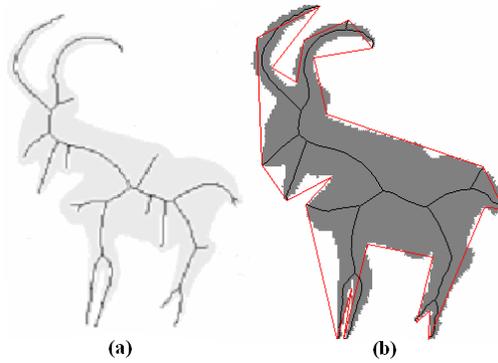


Fig. 9. Comparison between pruning result in [7] in (a) and our results in (b)

boundary features. Other experimental results of the proposed method prove that it is able to completely eliminate all the unimportant branches and still preserve the main structure. Our method does not suffer from the shortening of

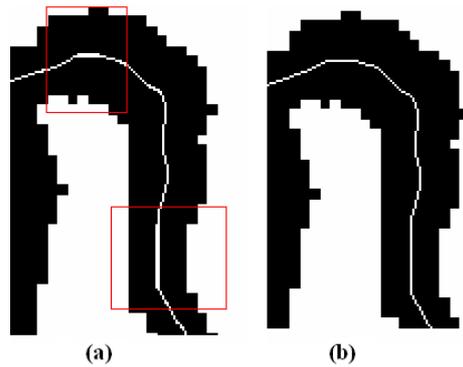


Fig. 10. Comparison between pruning result in [7] in (a) and our results in (b)

main skeleton branches and it preserves the topology of the skeleton. Moreover, the obtained skeletons seem to be in accordance with human perception, as it satisfies the six properties of the skeleton.

The method introduced by Bai et al. [7] can obtain excellent skeletons which contain most of the properties of ideal skeletons, but it cannot guarantee that the skeleton is one-pixel wide, which is illustrated in Fig. 10(a). As shown in Fig. 10(b), our method produces one-pixel wide skeletons.

5 Conclusion

In this paper, we establish a novel framework for skeleton computation that combines the geometric method of Discrete Curve Evolution with the statistical method of particle filters. 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 guarantee the skeleton is one-pixel wide. In future, we will extend the proposed approach to generate the skeleton for the shape with holes and 3D shapes, as the particle filter can deal with the condition that the path between two endpoints are not unique.

Acknowledgements

This work was supported by a grant from the Ph.D. Programs Foundation of Ministry of Education of China (No. 20070487028).

References

1. Blum, H.: Biological shape and visual science (part i). *J. Theoretical Biology* 38, 205–287 (1973)
2. Siddiqi, K., Shkouflandeh, A., Dickinson, S., Zucker, S.: Shock graphs and shape matching. In: *ICCV*, pp. 222–229 (1998)
3. Zhu, S., Yuille, A.: Forms: a flexible object recognition and modeling system. In: *ICCV* (1995)
4. Sebastian, T.B., Klein, P.N., Kimia, B.B.: Recognition of shapes by editing their shock graphs. *IEEE Trans. PAMI* 26, 550–571 (2004)
5. Macrini, D., Siddiqi, K., Dickinson, S.: From skeletons to bone graphs: Medial abstraction for object recognition. In: *CVPR* (2008)
6. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. *IEEE Trans. PAMI* 30, 1282–1292 (2008)
7. Bai, X., Latecki, L.J., Liu, W.Y.: Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Trans. PAMI* 29, 449–462 (2007)
8. Bai, X., Yang, X.W., Yu, D.G., Latecki, L.J.: Skeleton-based shape classification using path similarity. *Int. Journal of Pattern Recog. and Artif. Intell.* 22, 733–746 (2008)

9. Ogniewicz, R.L., Kübler, O.: Hierarchic voronoi skeletons. *Pattern Recognition* 28, 343–359 (1995)
10. Shaked, D., Bruckstein, A.M.: Pruning medial axes. *CVIU* 69, 156–169 (1998)
11. Choi, W.P., Lam, K.M., Siu, W.C.: Extraction of the euclidean skeleton based on a connectivity criterion. *Pattern Recognition* 36, 721–729 (2003)
12. Bai, X., Latecki, L.J.: Discrete skeleton evolution. In: Yuille, A.L., Zhu, S.-C., Cremers, D., Wang, Y. (eds.) *EMMCVPR 2007*. LNCS, vol. 4679, pp. 362–374. Springer, Heidelberg (2007)
13. Adluru, N., Latecki, L.J., Lakämper, R., Young, T., Bai, X., Gross, A.: Contour grouping based on local symmetry. In: *ICCV* (2007)
14. Latecki, L.J., Lakämper, R.: Shape similarity measure based on correspondence of visual parts. *IEEE Trans. PAMI* 22, 1185–1190 (2000)
15. Isard, M., Blake, A.: Contour tracking by stochastic propagation of conditional density. In: *ECCV*, pp. 343–356 (1998)
16. Isard, M., Blake, A.: Condensation – conditional density propagation for visual tracking. *IJCV* 29, 5–28 (1998)
17. Black, M.J., Fleet, D.J.: Probabilistic detection and tracking of motion boundaries. *IJCV* 38, 231–245 (2000)
18. Pérez, P., Blake, A., Gangnet, M.: Jetstream: Probabilistic contour extraction with particles. In: *ICCV* (2001)
19. Isard, M.: Pampas: Real-valued graphical models for computer vision. In: *CVPR*, pp. 613–620 (2003)
20. Sudderth, E.B., Ihler, A.T., Freeman, W.T., Willsky, A.S.: Nonparametric belief propagation. In: *CVPR*, pp. 605–612 (2003)
21. Doucet, A., de Freitas, N., Gordon, N.: *Sequential Monte Carlo Methods in Practice*. Springer, New York (2001)
22. Gordon, N.J., Salmond, D.J., Smith, A.F.M.: Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEE Proceedings-F* 140, 107–113 (1993)
23. Golland, P., Grimson, E.: Fixed topology skeletons. In: *CVPR* (2000)