Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

# A Fast Skeletonization Method

Ju Jia Zou

School of Engineering and Industrial Design
University of Western Sydney
Locked Bag 1797, Penrith South DC, NSW 1797, Australia
j.zou@uws.edu.au

**Abstract.** This paper presents an efficient method for extracting the skeleton of a planar shape. The method is based on computing local symmetries of the shape's linearized contour. The centrelines of the local symmetries form the skeleton of the shape. The method is fast when compared to existing skeletonization methods. The speed-up is usually a factor of more than 10 times for high-resolution images. The method is also robust against noise and geometrical transformations, such as rotation and uniform scaling.

## 1 Introduction

Skeletonization is a global space domain technique for shape representation [1]. It has been studied extensively since skeletons have attractive properties which make them suitable for structural pattern recognition [1][2]. There are two types of skeletonization methods: pixel-based and non-pixel-based. In a pixel-based method, all pixels inside a shape are used in the skeletonization process. Pixel-based methods often use thinning techniques [2]-[4] or distance transforms [4][5]. In a non-pixel-based method, only the contour pixels of a shape are used for skeletonization. The skeleton of the shape is analytically derived from its contour [6][7]. A non-pixel-based method is usually faster than a pixel-based method since less data is used for skeletonization. To extract the skeleton of a shape from its contour, local symmetries of the shape need to be identified accurately [6][7]. Discrete local symmetries (DLSs) can be computed from contour pixels [7]. The technique of identifying DLSs is suitable for low or medium resolution images while it may not be efficient for high resolution images since all contour pixels are used in computation. Skeletons are often sensitive to noise and geometrical transformations, such as rotation and scaling, in particular for high resolution images. This is a common problem for most skeletonization methods [2]. This study is aimed to increase the efficiency and robustness of skeleton computation. Discrete local symmetries are generalized to accomplish the goal.

## 2 Generalized Discrete Local Symmetry

A discrete local symmetry describes a local symmetry between a contour pixel and a contour segment between two adjacent contour pixels [7]. This concept is extended to describe a local symmetry between a contour pixel and a contour segment representing a sequence of contour pixels. A contour is divided into a sequence of segments by a linearization process where two end-points of each contour segment are connected by a straight-line segment. It is assumed that such straight-line segments do not intersect each other at an interior point. Thus, the shape contours of an image can be represented by a planar straight-line graph $G(V, E)$ where the edges $E$ and vertices $V$ are respectively the straight-line segments and their end-points. A vertex $v \in V$ and an edge $e \in E$ form a *generalized Discrete Local Symmetry (gDLS)* if (1) the circumcircle of a triangle $T$ formed by $v$ and the end-points of $e$ does not contain in its interior any other vertex of $G$ which is visible from all vertices of $T$, and (2) $T$ lies inside an object of the original image. Two points are visible from each other if a line segment connecting them does not intersect any edge $e' \in E$. A $g$DLS becomes a DLS if the underlying contour segment contains two and only two contour pixels.

DLSs can be computed using the constrained Delaunay triangulation technique [7]. This technique is also applicable to computing $g$DLSs. To obtain a perceptually more satisfactory result, a triangulation is stabilized by merging appropriate internal triangles. Merging criteria can be found in reference [7]. The skeleton of a shape is composed of the skeletons of the internal polygons of the shape. The skeleton of an internal polygon is a set of straight line segments where each connects the characteristic skeleton point of the polygon with the mid-point of one of the internal edges of the polygon [7]. The merging criteria in reference [7] can be used to determine characteristic skeleton points.

## 3 Implementation Issues and Discussion

Reducing the triangulation time is crucial to increasing the speed of skeletonization. It is obvious that the smaller the number of contour segments, the faster the triangulation process.

Let a contour $\mathcal{C}$ consist of $N$ pixels $\{p_1, p_2, ..., p_N\}$. $\mathcal{C}$ is linearized to generate a new contour $\mathcal{C}'$ which consists of $N'$ vertices $\{v_1, v_2, ..., v_{N'}\}$. Every vertex $v_i$ $(i = 1, 2, ..., N')$ of $\mathcal{C}'$ is a contour pixel of $\mathcal{C}$, i.e.,

$$v_i = p_{(i-1)\lambda+1}, \tag{1}$$

where $\lambda$ is the number of pixels of a contour segment between $v_i$ $(i = 1, 2, ..., N'-1)$ and $v_{i+1}$. The number of vertices in $\mathcal{C}'$ is

$$N' = \left[\frac{N-1}{\lambda} + 1\right], \tag{2}$$

where $[x]$ denotes the integral part of $x$. $\lambda$ is selected to be proportional to the median width $w_m$ of the underlying shape measured by the number of pixels, i.e.,

$$\lambda = [\xi \cdot w_m + 1]. \tag{3}$$

It is found empirically that $\xi \in [0.2, 0.7]$ usually gives a satisfactory result. In this paper, $\xi$ is 0.3.

The number of contour segments of a contour can be rewritten as

$$N' = \left[ \frac{N-1}{\xi \cdot w_m + 1} + 1 \right]. \tag{4}$$

Given an $n \times n$ image, assume that $N$ and $w_m$ are proportional to $n$, i.e., $N = \zeta \cdot n$ and $w_m = \delta \cdot n$. If $n$ is large, then

$$N' \simeq \left[ \frac{\zeta}{\xi \cdot \delta} + 1 \right]. \tag{5}$$

Thus, $N'$ does not depend on $n$. Therefore, for a high resolution image, the triangulation time is not directly related to the absolute size of the image, but depends on the relative size of the shapes in the image. In other words, a similar time would be taken to triangulate different versions of the same shape sampled at different rates. On the other hand, existing skeletonization methods usually take a much longer time to skeletonize a high-resolution image than a low-resolution one. For instance, a thinning method usually takes an $O(n^3)$ time to skeletonize an $n \times n$ image [4].

Two types of noise often exist in a digital image: impulse noise and contour noise. Impulse noise corresponds to small black regions which should be white, or small white regions which should be black. A black region or a white region is ignored in the skeleton computation if its contour contains less than $\tau = \kappa_1 \cdot w_m + \kappa_2$ pixels where $\kappa_1 = 1.5$, $\kappa_2 = 3$ in this paper, and $w_m$ is the median width of the underlying shape. Contour noise is suppressed by contour linearization.

## 4 Experiment Results

The efficiency of the proposed method is tested on a tool pattern sampled at four different rates (72, 150, 300, and 600dpi). The timings are shown in Figure 1.

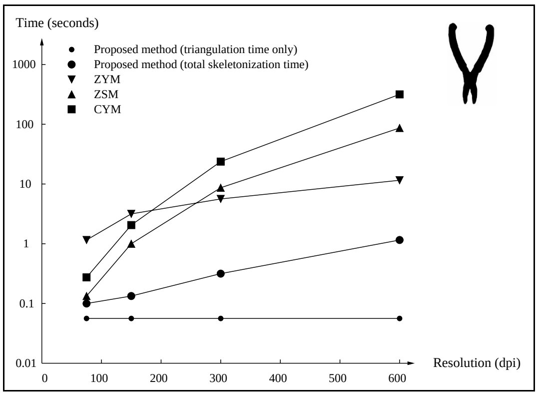


**Fig. 1.** Timings for skeletonizing a tool pattern using different methods.

Results of one non-pixel-based method proposed by Zou and Yan (ZYM) [7], and two pixel-based methods, namely Zhang and Suen's method (ZSM) [3] and Chang and Yan's method (CYM)) [5], are included for comparison. A Pentium II MMX processor with a clock frequency of 233 MHz was used in the experiment. It can be seen that the proposed method is more efficient than ZYM and much faster than ZSM and CYM at high resolutions. At a high resolution (e.g., 300dpi or higher), the proposed method is usually 10 times faster than any of the other three methods. As expected, it takes almost a constant time to triangulate different versions of the tool pattern sampled at different rates.

The proposed method has been tested on various noisy images with shapes on different orientations. Typical results are shown in Figures 2 and 3. Experiment results obtained from ZYM, ZSM, and CYM are included for comparison. It can be seen that the proposed method is robust against geometrical transformations and noise when compared to the other three methods. For instance,

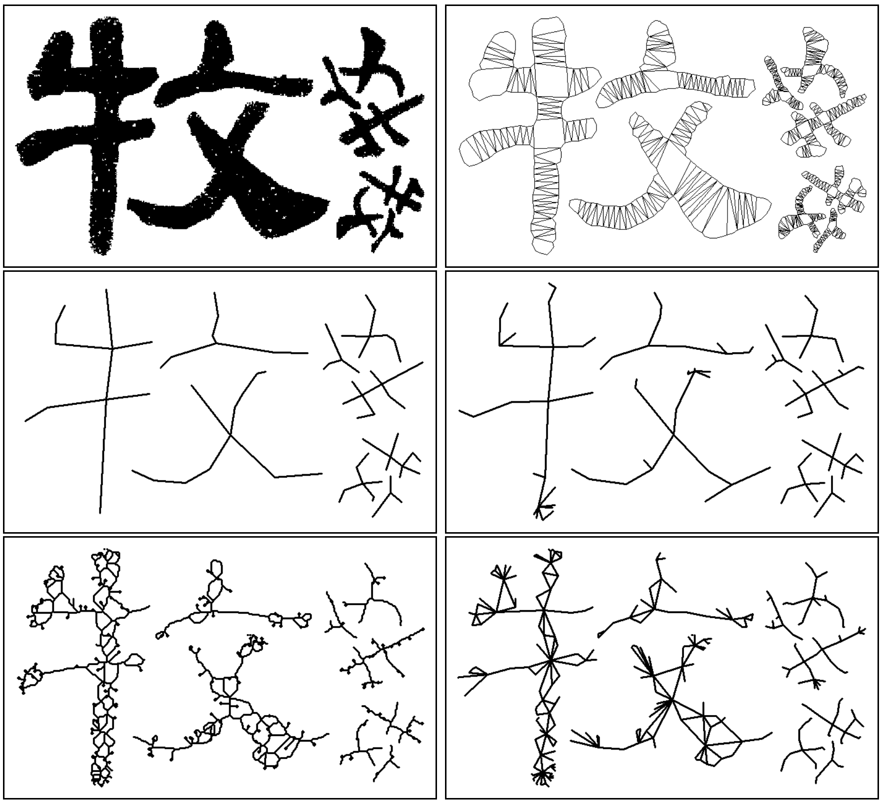

**Fig. 2.** (Upper-left) Three instances of a Chinese character. (Upper-right) Result after triangulation and stabilization. Non-triangles are calculated based on the merging criteria of ZYM. (Middle-left) Skeletons obtained from the proposed method. (Middle-right) ZYM's result. (Lower-left) ZSM's result. (Lower-right) CYM's result.

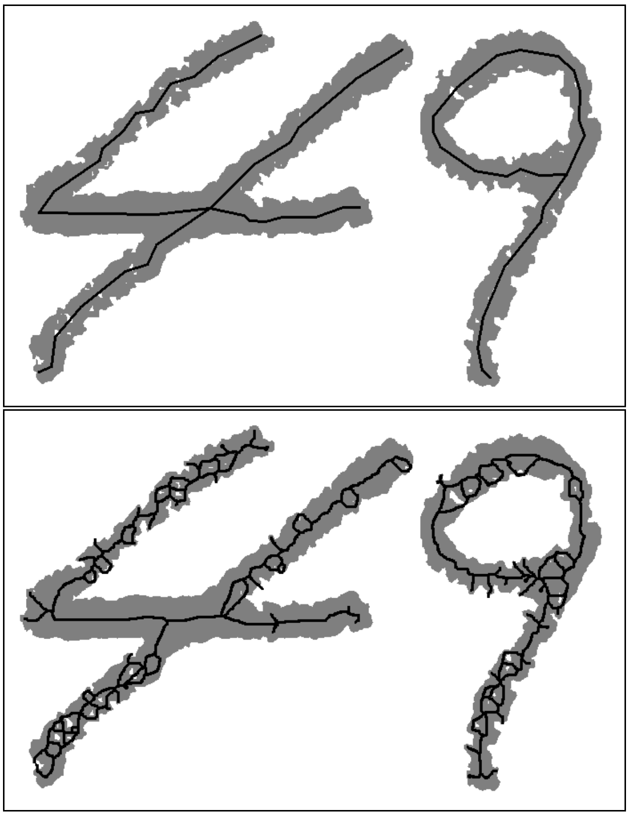

**Fig. 3.** Results on handwritten numbers using (Upper) the proposed method and (Lower) ZSM.

Figure 2 shows the skeletonization results of three instances of a Chinese character with different scales and orientations using four different methods. The proposed method generates structurally identical skeletons on three instances. The essential features of the character, such as the number of branches, the number of intersection points, the number of end points, and the number of branches joining at each intersection are not sensitive to rotation and scaling. They are also not sensitive to noise. On the other hand, the other three methods generate different skeletons on different instances.

## 5 Conclusion

The concept of discrete local symmetry has been generalized to describe local symmetries between a contour point and a contour segment in this paper. The time for skeleton computation can be considerably reduced when generalized discrete local symmetries are used. Experiment results show that a speed-up factor of more than 10 can often be achieved when compared to existing skeletonization methods. The number of vertices for skeleton computation is almost unaffected by the number of pixels in a high resolution image (e.g., 150 dpi or higher). This is attractive for skeletonizing high resolution patterns. Skeletons extracted from generalized discrete local symmetries are robust against geometrical transformations and noise.

## References


1. Loncaric, S.: A Survey of Shape Analysis Techniques. Pattern Recognition **31** (1998) 983–1001.
2. Lam, L., Lee, S.-W., Suen, C.Y.: Thinning Methodologies – A Comprehensive Survey. IEEE Trans. Pattern Anal. Mach. Intell. **14** (1992) 869–885.
3. Zhang, T.Y., Suen, C.Y.: A Fast Parallel Algorithm for Thinning Digital Patterns. Comm. ACM **27** (1984) 236–239.
4. Smith, R.W.: Computer Processing of Line Images: A Survey. Pattern Recognition **20** (1987) 7–15.
5. Chang, H.-S., Yan, H.: Analysis of Stroke Structures of Handwritten Chinese Characters. IEEE Trans. Sys. Man Cybern. (B) **29** (1999) 47–61.
6. Saint-Marc, P., Rom, H., Medioni, G.: B-spline contour representation and symmetry detection. IEEE Trans. Pattern Anal. Mach. Intell. **15** (1993) 1191–1197.
7. Zou, J.J., Yan, H.: Skeletonization of Ribbon-Like Shapes Based on Regularity and Singularity Analyses. IEEE Trans. Sys. Man Cybern. (B) **31** (2001) 391–395.