A pruning algorithm for Voronoi Skeletons

A. Beristain, M. Graña

A robust and efficient binary image shape skeleton computation procedure in 2D, including a pruning procedure, is presented. The procedure is as follows: first the shape contour is subsampled along its arclength, then the Voronoi skeleton is computed from the resulting reduced contour set of points, and finally, a novel two stage pruning procedure is applied to obtain a simplified skeleton. Besides the information reduction, this pruning makes the skeleton more robust to noise. Pruning can be done in real time thanks to some useful properties ofVoronoi skeletons. Here we prove that if the two end points of a Voronoi segment are inside the shape, then the entire segment is contained in the shape. A prototype implementation performs in real-time (60 fps) on an off-the-shelf computer. Testing it on an in-house but publicly available image database shows the stability and robustness of the approach.

Introduction: Skeletons [1,2] summarize all the local and global relevant information of shapes for their representation and recognition, and other purposes. One of the major drawbacks of using skeletons for shape recognition is that they are unstable in the sense that small variations on the shape boundary curves may produce severe changes on the skeleton topology. This sensitivity to shape contour noise has been dealt with by pruning the spurious skeleton branches after its computation. The work in [3] introduces a robust and efficient skeleton pruning procedure, which can be applied to the result of any skeletonization (i.e. skeleton computation) algorithm. The steps in [3] are: (1) The shape contour is simplified using a Discrete Curve Evolution (DCE) [4] procedure, which iteratively removes the shape boundary point with less contribution to the global shape and replaces the two segments connected to it by only one connecting the two remaining points, until a stop criteria is met, (2) skeleton points whose generative points do not belong to different edges of the convex polygon determined by the contour point subset of the DCE are discarded. Disadvantages of the approach in [3] are that not all skeletonization algorithms provide the generative points, or at least efficiently, and that not all skeletonization algorithms produce complete and connected skeletons.

The algorithm presented in this letter improves over the one presented in [3] both in stability and real time performance. It performs a DCE based pruning procedure, but its starting point is a Voronoi skeleton [5] whose Voronoi segments are completely included in the shape. The algorithm is very fast because, (a) the Voronoi skeleton can be computed in $O(n \cdot \log n)$ or even in $O(n)$ [6], (b) only the end points of the Voronoi segments need to be tested in order to determine if the segment belongs to the initial skeleton, (c) generative points are provided without additional computational cost, and (d) generative points are the same for all the points in Voronoi segments.

Skeletonization and pruning algorithm: First we consider the shape contour denoted C, which can be constituted by one or several connected components of one-pixel width curves. It is formally defined as $C = \{p \mid \forall \xi > 0, \exists a \in F, \exists b \notin F \text{ s.t. } \|p-a\| < \xi \wedge \|p-b\| < \xi\}$. (An image is a function $I: D \rightarrow \Re$, a shape is a collection of pixel sites $F \subset D$ identified through some segmentation procedure). The Voronoi Tesselation V is given by a decomposition of the image domain D into convex regions around the Voronoi sites $V_{sites} = \{v_i\}$ called Voronoi polygons $V_{poly}(i) = \{x \mid \|x - v_i\| < \|x - v_j\| \forall v_j \neq v_i\}$. Each Voronoi polygon is bounded by several Voronoi segments and vertices. Each Voronoi segment $s_{ij} = \{x \mid \|x - v_i\| = \|x - v_j\| \wedge \forall k \neq i \neq j, \|x - v_k\| > \|x - v_i\|\}$ is the locus of the intersection of two and only two different Voronoi polygons and is, therefore, determined only by two Voronoi sites. Each Voronoi vertex is the intersection of three or more Voronoi polygons. The set of Voronoi vertices is defined as $T = \{x \mid \exists i,j,k \wedge x \in s_{ij} \cap s_{ik} \cap s_{jk}\}$. According to the original Voronoi skeleton S_V definition [5], the skeleton of a shape F, is formed by the pixel sites of Voronoi segments falling inside the shape F of the Voronoi Tesselation created using the contour points as Voronoi sites ($V_{sites} = C$), that is $S_V = \bigcup_{ij} [s_{ij} \cap F]$.

Our algorithm is as follows:

1. The contour curves are down-sampled uniformly along their arclengths to obtain the set of Voronoi sites $V_{sites} \subset C$.
2. We compute the Voronoi segments $\{s_{ij}; i,j \in V_{sites}\}$ of the Voronoi Tessellation induced

by V_{sites}.

3. We consider as the initial Voronoi Skeleton the set of Voronoi segments fully contained inside the shape F, formally: $S_V = \bigcup_{s_{ij} \subset F} s_{ij}$.
4. As a second pruning step, we apply the DCE pruning procedure [3] based on the original contour C to further reduce the skeleton.

The contour down-sampling reduces both the effect of minor noise on the shape boundary, and the Voronoi Skeleton computational cost. Checking the first pruning condition of a segment is extremely fast, because it involves checking only if two points belong to the shape for each Voronoi segment. We prove in Theorem 1 that if the end points (i.e. its Voronoi vertices) of a Voronoi segment fall inside the shape, the whole segment must be contained inside the shape, when $V_{sites} = C$. This result holds also when the contour is subsampled to obtain the initial Voronoi sites, if the shape is convex or if the width of concave parts of the contour follows some conditions related to the subsampling ratio. These conditions will be studied in detail elsewhere due to present space limitations. We have found that Theorem 1 holds most of the times for practical purposes. The importance of Theorem 1 lies in the great speedup obtained because the rejection of a whole segment can be done testing only its extreme points. We present this theorem in the continuous case for simplicity. To state and prove it for discrete domains requires some further elaboration.

Theorem 1: Let $A_I, A_F \in T \cap s_{ij}$ be the end points of a Voronoi segment s_{ij}. If both A_I, A_F belong to the shape F, then the whole s_{ij} is included in F, when the Voronoi site set is equal to the whole shape contour $(V_{sites} = C)$. Formally: $A_I \in F \wedge A_F \in F \Rightarrow s_{ij} \subset F$.

Proof:

Let us start assuming that there is a point in a Voronoi segment whose endpoints fall inside the shape: $\exists p \in s_{ij} \wedge p \notin F$. The proof consists in showing that s_{ij} cannot be a Voronoi segment. That is, in such a case there must be a contour point lying in the Voronoi segment, formally:

$$A_I, A_F \in F \wedge \exists p \in s_{ij} \wedge p \notin F \Rightarrow \exists q \in s_{ij} \cap C \Rightarrow$$
$$\exists q \in \overline{A_I p} \subset s_{ij} \text{ s.t. } \forall \xi > 0, \exists q^+ \in F, q^- \notin F \left\| q - q^+ \right\| \le \xi \wedge \left\| q - q^- \right\| \le \xi$$

The fact that there are at least one point inside the shape and another outside it in the same segment $\overline{A_I p}$ allows to select q^+, q^- so that the contour condition holds for some q in this segment. Then, as $V_{sites} = C$, it follows that q is also a Voronoi site different from the two that determine the Voronoi segment s_{ij}, that is: $q \equiv v_k \neq v_i \neq v_j$. This is in contradiction with the definition of Voronoi segment, because we have a third site whose distance to some point in the segment is smaller than the distance to the segment generating segments: $\exists q \in V_{sites}$ s.t $\|q - q\| < \|q - v_i\| = \|q - v_j\|$. Thus the theorem holds. ..

An additional speedup is obtained because the DCE pruning criterion needs only to be evaluated once for each Voronoi segment s_{ij} in the initial skeleton, since the generative points are the same for the whole segment, that is the pair of Voronoi sites (v_i, v_j) that determine s_{ij}.

A potential drawback of our algorithm is that skeleton branches do not end in a contour point, because they are shortened by the first pruning phase. But meaningful skeleton branches are not removed and the shortening is limited, as it can be seen in Fig. 1 and Fig.2.

Results: A C++ prototype implementation has been built to prove the real-time performance of this procedure, based on DirectShow and OpenCV libraries.

The prototype is able to compute skeletons at more than 60 frames per second, including all the preprocessing required, on a pc with an Intel Pentium IV (3 GHz.) processor and a 320x240 video resolution. For qualitative comparison we have applied the Matlab implementation of [3] made available at [7]. The skeletons obtained using both approaches are quite similar, as it can be appreciated in Fig. 1 on a well-known benchmark image, but it can be noticed that (1) our skeleton does not touch the shape contour, (2) our implementation produces a simpler skeleton. We have also checked the stability of our approach on an in-house image database of hand gestures for tabletop interaction made publicly available in [8]. The test database contains binary image sequences of hand gestures corresponding to three different dynamic gestures. The computation of the skeletons by both procedures shows that our algorithm is more stable in the sense that small variations of the shape contour introduce less variations of the skeleton than the algorithm proposed in [3]. This can be appreciated in figure 2, where the middle skeletons correspond to [3] and the right ones to our approach. Note the differences in variations in the upper branches of the skeletons.

Conclusion: We propose a robust and efficient skeletonization procedure produced by an innovative pruning of the Voronoi skeleton which can be implemented in real time and shows an improved stability and robustness. Stability is critical for the use of skeletons to obtain features for pattern recognition and shape matching, because it reduces undesired variability in the extracted features, therefore we expect our algorithm to be useful in real time pattern recognition applications.

Acknowledgment: This work have been supported by an Ikertu grant from the Industry Department of the Government of The Basque Country enjoyed at Innovae Vision company (www.innovaevision.com).

References

1 SIDDIQI, K., and PIZER S.: 'Medial Representations. Mathematics, Algorithms and Applications', Springer Verlag, Computational Imaging and Vision series, 2008, Vol. 37
2 HESSELINK, W.H., and ROERDINK, J.B.T.M.: 'Euclidean Skeletons of Digital Image and Volume Data in Linear Time by the Integer Medial Axis Transform', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30, (12), pp. 2204-2217.
3 BAI, X., and LATECKI, L. J., and LIU, W.: 'Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution', IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29, (3), pp. 449-462
4 LATECKI, L.J., and LAKÄMPER, R.: 'Convexity Rule for Shape Decomposition Based on Discrete Contour Evolution', Computer Vision and Image Understanding, 1999, 73, pp. 441-454
5 OGNIEWICZ, R., and ILG, M.: 'Voronoi skeletons: theory and applications', Proceedings of the I992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1992, Champaign,Illinois, June, pp. 63-69
6 ARONOV, B.: 'A Lower Bound on Voronoi Diagram Complexity', Information Processing Letters, 2002, 83, (4), pp. 183-185
7 http://www.cis.temple.edu/~latecki/Programs/skeletonPruning07.htm
8 http://www.ehu.es/ccwintco/index.php/Synthetic_hand_gesture_images_for_tabletop_interaction

Authors' affiliations:

A. Beristain, M. Graña (Grupo de Inteligencia Computacional GIC, Facultad de Informática de San Sebastián. University of The Basque Country, Paseo Manuel de Lardizabal, 1, 20018 San Sebastián. Spain.)

Author’s e-mail address (A. Beristain): beristainandoni@yahoo.es

Figure captions:

Fig. 1 Left: skeleton computed using the Matlab implementation in [7]. Right: Voronoi based approach presented in this letter. The binary image is taken from [3].

Fig. 2 Left column: Hand gesture binary image sequence from [8]. Middle Column: Skeletons obtained using the implementation of algorithm [3] provided in [7]. Right column: Skeletons obtained using the approach in this letter.

Figure 1

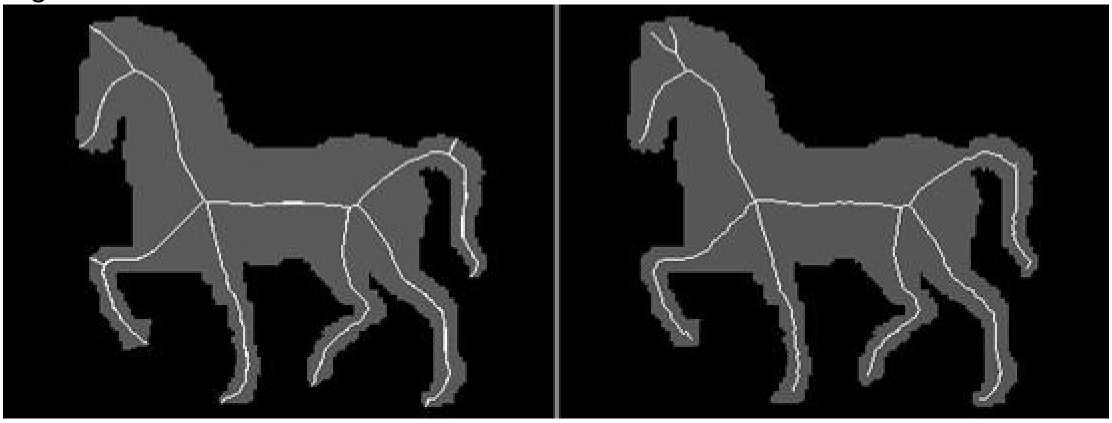

Figure 2