



The Propagated Skeleton: A Robust Detail-Preserving Approach

Bastien Durix¹(✉), Sylvie Chambon¹, Kathryn Leonard², Jean-Luc Mari³,
and Géraldine Morin¹

¹ IRIT-University of Toulouse, CNRS, Toulouse, France
`bastien.durix@enseeiht.fr`

² Occidental College, Los Angeles, CA, USA

³ Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France

Abstract. A skeleton is a centered geometric representation of a shape that describes the shape in a simple and intuitive way, typically reducing the dimension by at least one. Skeletons are useful in shape analysis and recognition since they provide a framework for part decomposition, are stable under topology preserving deformation, and supply information about the topology and connectivity of the shape. The main drawback to skeletonization algorithms is their sensitivity to small boundary perturbations: noise on a shape boundary, such as pixelation, will produce many spurious branches within a skeleton. As a result, skeletonizations often require a second pruning step. In this article, we propose a new 2D skeletonization algorithm that directly produces a clean skeleton for a shape, avoiding the creation of noisy branches. The approach propagates a circle inside the shape, maintaining neighborhood-based contact with the boundary and bypassing boundary oscillations below a chosen threshold. By explicitly modeling the scale of noise via two parameters that are shape-independent, the algorithm is robust to noise while preserving important shape details. Neither preprocessing of the shape boundary nor pruning of the skeleton is required. Our method produces skeletons with fewer spurious branches than other state-of-the-art methods, while outperforming them visually and according to error measures such as Hausdorff distance and symmetric difference, as evaluated on the MPEG-7 database (1033 images).

1 Introduction

A skeleton is an internal structure that describes a shape, defined by the centers of the maximally inscribed balls enclosed within the shape. The set of circle centers and radii gives the medial axis, which completely describes the shape [5]. Because the skeleton can be represented by a graph where each edge represents a feature of the shape, it is useful in applications such as shape analysis and pattern recognition [4, 12, 18, 20, 22]. To be useful in such applications, the skeleton should possess certain properties: centeredness within the shape, thinness, topological equivalence with the shape, and completeness. It should also be robust to noise.

The noise property is one of most difficult to respect, requiring a distinction between noise and details of the shape. We propose a skeletonization method that possesses the above properties, and that is robust to noise on the shape boundary.

Many skeletonization methods have been proposed [16]. Three common categories of computing a skeleton from a discrete structure are:

- Thinning methods that use a morphological operator on a binary shape,
- Distance map methods that consider the distance to the boundary,
- Methods based on the Voronoï diagram of points on the shape boundary.

Thinning methods erode shape pixels in 2D [9] or voxels in 3D [6] according to specific rules and criteria. The resulting skeleton is a 1-pixel-wide structure. We refer to [21] for a survey of these methods. Unfortunately, the resulting skeleton lies on a 1-pixel-wide grid centered only up to a pixel size, thereby inducing a loss of precision. Furthermore, the radius is not directly computed. In contrast, our proposed skeletonization produces a structure not constrained by the grid, and gives the radius for each skeletal point.

Distance maps are defined as the minimal distance between a point and the shape boundary. The singularities of the distance map generate a skeleton. One distance map skeleton uses an active contour evolving inside the shape, following the gradients of the distance transform [13] until the fronts from opposite sides of the shape meet at the skeleton. One drawback to this method is that the structure of the skeleton depends on the local speed of the active contour, which is a function of the local curvature. Another distance map approach computes singularities of the distance transform using image based-methods, then estimates connectivity of the medial axis [15]. Because the distance map is discrete on a discrete image, its singularities can only be computed up to a convolution operator. The choice of this operator is crucial, and can create arbitrarily large branches due to noise. Although our method also uses a distance map, its main principle is to follow the boundary of the shape, which, consequently, makes it more robust to noise.

Voronoï skeletonization is the most popular method for computing a skeleton. Introduced by Ogniewicz [14] *et al.*, the interior edges of the Voronoï diagram of a discrete boundary represent the skeleton. This method is popular because it is efficient and precise. Unfortunately, the resulting skeleton usually includes many uninformative branches and is therefore unnecessarily complex (*cf.* Fig. 1(a)). For more on stability, see [3]. In 3D, the complexity is compounded, which has led to methods such as the power crust [2] to extract a simpler structure. Once a skeleton is computed, most approaches require a second step to prune noise. We next present pruning methods, and then describe approaches to generate a clean skeleton without pruning.

Pruning methods remove noisy portions of a skeleton after computation. Usually, a criterion is evaluated at each skeletal point to determine if that point should be removed. We present three popular pruning approaches. The λ -medial axis [7], with parameter λ , removes a skeletal point if the point's circle intersects the boundary at tangency points that are contained in a circle of radius λ . The

θ -homotopy medial axis [19], with parameter θ , evaluates the angle between the center of the skeletal circle and the tangency points on the boundary. If the angle is less than θ , then the associated skeletal point is removed. Finally, the scale-axis-transform (SAT) [8] grows each skeletal circle by a multiplicative factor s , then computes a skeleton from the new shape. The circles generating the new skeleton are then reduced using the multiplicative factor $1/s$. A variant of SAT that ensures homotopy equivalence with the boundary grows all circles by the factor s and removes grown circles that are contained inside another grown circle. These three methods have the similar drawbacks. First, the homotopy of the pruned skeleton can not be guaranteed, except for SAT variant and the λ -medial axis, for which the connectivity of the skeleton is ensured only if λ is lower than the smallest radius of its skeleton. Second, the choice of the parameter is difficult and often must be tuned to each shape. Finally, no distinction can be made between noise and small shape details, which means that desirable features of the shape may be pruned away (cf. Fig. 1(c) and (d)).

Directly computing a clean skeleton is an appealing alternative, with two main approaches. The first approach includes a pruning criterion within the skeletonization. For example, Digital Euclidean Connected Skeleton (DECS) [11] combines the ridges of the distance map and the centers of the maximal balls to obtain a skeleton on a grid of pixels (cf. Fig. 1(e)). As with thinning methods, the resulting skeleton has the width of one pixel. Furthermore, this method returns a skeleton that tends to be irregular. The second approach modifies the boundary itself, as in the circular boundary representation [1]. The resulting skeleton avoids many noisy branches, but the construction is complicated: the conversion of the boundary into arcs requires that the boundary be represented by polynomial splines which are then converted into circular arcs based on a chosen parameter. In contrast, our method uses the information of the discrete boundary directly without preprocessing.

We propose a new skeletonization method that directly produces a clean skeleton (cf. Fig. 1(f)) from boundaries affected by noise. Two common types of noise are pixelation noise resulting from the rasterization of the shape, and Gaussian noise on the boundary. In 2D, because most shapes are extracted from binary shape images, we focus here on rasterization noise. Our method propagates a circle inside the shape, ensuring continuous contact along the boundary. This “growing approach” has previously been applied only for grid-constrained skeletons [17]. By taking into account the scale of the rasterization noise, we can avoid creating noisy branches. Our method draws from distance map skeletons, searching for singularities of a distance map, and Voronoï skeletons, computing the connectivity of the skeleton directly from the boundary. The method generates a clean skeleton a priori and does not require a pruning step. Contributions of our method are twofold: we explicitly avoid noise on the boundary, and we capture salient geometric information with fewer branches.

Section 2 describes our method, and Sect. 3 compares our method with the most common skeletonization approach, the Voronoï skeletonization [14], pruned by scale-axis-transform [8], λ -medial axis [7] and θ -homotopy medial axis [19]. We also compare with DECS [11].

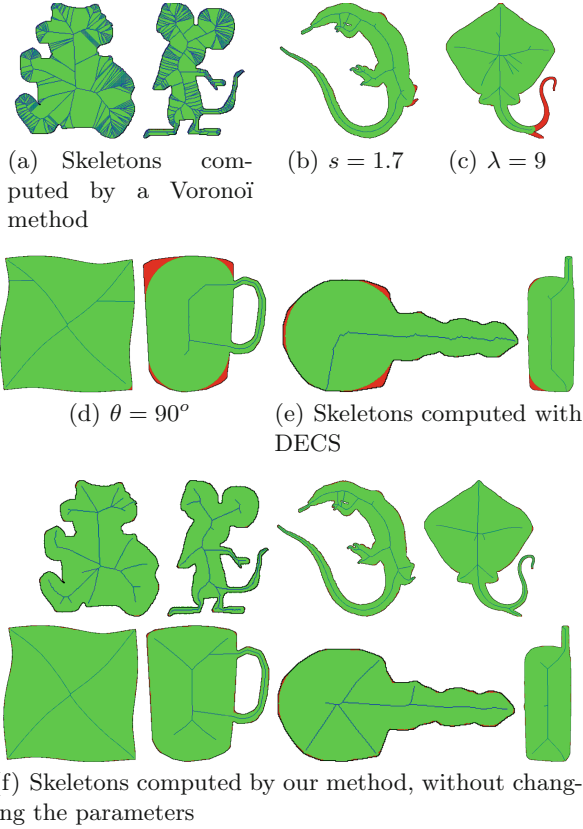


Fig. 1. Illustration of the limits of Voronoi skeletonization (a), the pruning methods (scale-axis-transform (b), λ -medial axis (c), θ -homotopy medial axis (d)), and DECS (e), and the advantages of ours (f). Green parts correspond to the reconstructed shape, and red parts show what has been lost. Voronoi (a) skeletons are noisy, requiring pruning. Some pruning methods struggle to differentiate between noise and shape details, and parts can be lost while noisy branches are preserved (b),(c). Others have shape dependent parameters (d). The DECS method returns a skeleton that is simple, but it tends to miss some parts, and be irregular. Our method (f) distinguishes between noise and details with parameters depending only on the scale of the noise. (Color figure online)

2 Propagating Skeletons

We consider a shape characterized by a matrix of pixels (from a binary image, for example), $\mathbf{S} \in \mathcal{M}_{h,l}(\{0; 1\})$, with h the height of the image, and l its width. We extract a piecewise linear contour \mathcal{B} from the shape pixels. From the boundary contour, we construct a skeleton represented by a graph whose vertices correspond to centers of skeletal balls and whose edges give connectivity information, consistent with boundary connectivity, without branches generated by pixela-

tion noise. Following the philosophy of the Voronoï skeleton, where vertices are centers of maximally inscribed bitangent balls, we propagate circles inside the shape ensuring continuous contact with the boundary on at least two sides.

Before describing the complete algorithm in Sect. 2.4, we present some necessary tools. Section 2.1 defines a smooth distance estimation (SDE), used to compute a smoothed radius of skeletal circles. Section 2.2 explains the notion of a contact set, a key construction for robustness to noise. Finally, Sect. 2.3 describes the circle propagation principle that determines neighbors of a circle given its contact sets and the SDE.

2.1 Smooth Distance Estimation (SDE)

In this section, we consider a rasterized boundary. Typically, radius estimation measures the minimal Euclidean distance to a point on the contour for each point in the interior of the shape. This leads to poor radius estimates for rasterized boundaries, as shown in Fig. 2(a).

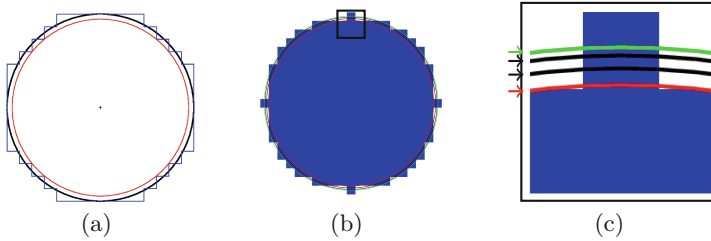


Fig. 2. The limitations of the Euclidean distance function compared to our proposed distance function. (a) The black circle, with center represented by a cross, is rasterized, giving the blue boundary: each pixel is considered inside the binary shape if its center is in the black circle. The red circle has same center as the black circle with radius given by the minimum Euclidean distance to a point of the contour. As the radius of the blue circle is less than that of the black circle, the Euclidean estimation of the radius is incorrect. (b) Estimated circle from the SDE, for several values of σ . In (c), the arrows indicate from top to bottom: ● the original circle ($r = 10$), ● the circle associated with $\sigma = 1$ ($r = 9.89$), ● the circle associated with $\sigma = 0.5$ ($r = 9.55$), ● the circle associated with the classic distance function ($r = 9.51$). When σ approaches 0, the radius of the circle converges to the radius from the Euclidean distance function. (Color figure online)

To improve radius estimation, we consider more than one boundary point to construct a smoothed version of the Euclidean distance:

$$d_{\mathcal{B}}(P) = \min_{B \in \mathcal{B}} \|P - B\|.$$

The proposed distance adds a weight to points on the boundary to capture their relative importance to the particular circle being computed:

$$f_{\mathcal{B},\sigma}(P) = \frac{\sum_{B \in \mathcal{B}} p_{\mathcal{B},\sigma}(P, B) \|P - B\|}{\sum_{B \in \mathcal{B}} p_{\mathcal{B},\sigma}(P, B)},$$

where $p_{\mathcal{B},\sigma}(P, B)$ is a weight assigned to the point B of the boundary. To determine the weight, we consider the distance between B and the circle $\mathcal{C}(P, \rho)$ with $\rho = d_{\mathcal{B}}(P)$. The larger the distance, the less influential the point should be. In other words, the weight function should decrease rapidly as distance to the circle increases. A semi-normal distribution with parameter σ fits these requirements:

$$P_{\sigma}(t_0 \geq 0) = \frac{2}{\sigma\sqrt{2\pi}} \int_{t_0}^{+\infty} e^{-\frac{t^2}{2\sigma^2}} dt.$$

The influence of each boundary point B on P is then given by:

$$p_{\mathcal{B},\sigma}(P, B) = P_{\sigma}(d_{\mathcal{C}}(P, B)), \text{ with } d_{\mathcal{C}}(P, B) = \|P - B\| - d_{\mathcal{B}}(P).$$

The behavior of the function, illustrated in Fig. 2, is related to the values of the parameter σ . For $\sigma = 0$, $f_{\mathcal{B},\sigma}$ is equal to the Euclidean distance. This function is also used to identify an initial point of the skeleton. To estimate a first point P_0 , we randomly choose a point Q_0 inside the shape, then maximize the function $f_{\mathcal{B},\sigma}(P, Q_0)$. Any local maximum is a singular point of the function $f_{\mathcal{B},\sigma}$ and will therefore be a skeletal point. Once P_0 is estimated, we identify its contact set and begin propagation. We define contact set in the next section.

2.2 Contact Between the Circle and the Shape

We now define the contact set between a circle $\mathcal{C}(P, \rho)$, with center P and radius $\rho = f_{\mathcal{B},\sigma}(P)$, and the contour \mathcal{B} . We consider the parameter σ from $f_{\mathcal{B},\sigma}$, and introduce a new parameter, α , which defines the maximal distance from a point to its associated circle. The contact set is the topological closure of boundary points at a distance less than σ to the circle together with points at a distance less than α to the circle, as illustrated in Fig. 3. A contact set corresponds to consecutive boundary points where the first and last points are at most σ , and all points are at most α , from the skeleton point. The points in the contact set are analogous to the bitangent contact points in the continuous model, denoting a set of points “closest” to the skeletal point. The contact sets also determine the directions in which the circle $\mathcal{C}(P, \rho)$ can propagate. The parameter α allows us to avoid noise on the boundary while simultaneously maintaining topological consistency of the skeleton with the boundary (*cf.* Fig. 3).

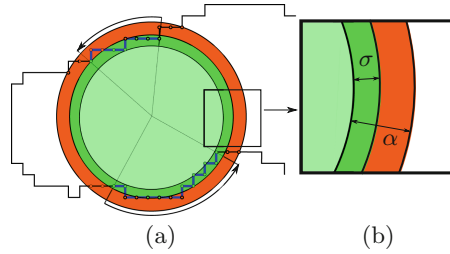


Fig. 3. Contact sets for a skeletal point. In (a), we divide the area around the estimated SDE circle into three parts, see (b) for details: the thick green part $r + \sigma$ (where r is the radius of the original circle), the thick orange part $\alpha - \sigma$, and the outer part. If only the green is taken into account, we find 6 contact sets (in green, 3 at the top, 3 at the bottom). Adding the uncertainty zone using α merges these into 2 contact sets (blue lines), which prevents the creation of four noisy branches. (Color figure online)

The definition of contact sets in our algorithm serves two purposes: reducing noise and maintaining topological equivalence to the shape. First, the contact sets for a point designate the suitable directions for propagation without noise. The value of α , chosen to distinguish between noise on the contour and details of the shape, defines the maximum distance between a point of the contour and the closest skeletal circle, which defines the Hausdorff distance between the true boundary and the approximated boundary produced by the skeleton. Second, as two neighboring circles must have overlapping contact sets, we are guaranteed to consider all points of the boundary, which guarantees the topological equivalence between skeleton and boundary.

2.3 Circle Propagation

After estimating contact sets associated to a point P of the skeleton, we propagate the circle between successive pairs of contact sets. To propagate between two successive contact sets, \mathcal{T}_1 and \mathcal{T}_2 , we search for the farthest circle that maintains contact with points of \mathcal{T}_1 and \mathcal{T}_2 (cf. Fig. 4). This produces a nested loop, with one loop searching for the distance, *i.e.* the circular arc on which there is the next circle, and the other searching for the angle, *i.e.* the position of the center on the given circular arc.

The search region for the next point of the skeleton is the portion of the circle between the last point A_1 of \mathcal{T}_1 and the first point A_2 of \mathcal{T}_2 . For a given distance d , less than the radius $\rho = f_{\mathcal{B},\sigma}(P)$ of the circle, we search for an angle θ_d where $f_{\mathcal{B},\sigma}(Q_d)$ is local maxima (with $Q_d = P + (d \cos(\theta_d), d \sin(\theta_d))^t$). Not every value of d can describe a valid neighbor of P , since contact sets of the neighbors must contain A_1 and A_2 . Using this property, we can validate, for each d , the possibility of Q_d as a neighbor of P . We select the highest value of d so that Q_d is a valid neighbor of P .

To estimate the value of θ_d , we take an initial interval $[\theta_0^1, \theta_0^2]$, where θ_0^i designates the angle of A_i on the circle, and, then, we decrease it with search by

dichotomy. We compute the middle angle θ_i between θ_i^1 and θ_i^2 . If the derivative of $g_d(\theta) = f_{\mathcal{B},\sigma}(P + (d \cos(\theta), d \sin(\theta))^t)$ is positive on θ_i , we update the interval to $[\theta_i, \theta_i^2]$, otherwise we update to $[\theta_i^0, \theta_i]$. The search stops when $(\theta_i^2 - \theta_i^1)$ is less than a given threshold $\epsilon_\theta = \frac{0,01\sigma}{\rho}$. The search for the distance d is done in a similar way, up to a threshold $\epsilon_d = 0,01\sigma$. These thresholds could be chosen as machine precision, but because the scale of noise on the boundary is σ , we can stop the search when the distance given by each interval has less influence than the noise.

Using this algorithm, we find an approximation of a local skeleton around the point P . The number of contact sets gives the degree of connectivity of the skeleton at P . By construction, only the variations of the boundary more than α from the circle are considered as new propagation directions. Finally, forcing contact sets of any two successive circles to intersect guarantees that all the points of the boundary are associated with at least one circle, and we produce a connected skeleton. We summarize the complete algorithm in the next section.

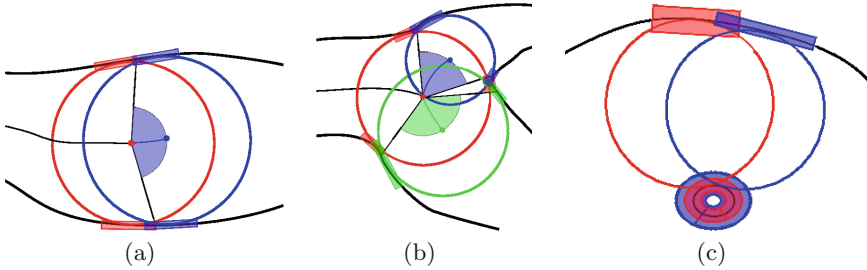


Fig. 4. Propagation of a circle, for 2 contact sets in (a) and 3 contact sets in (b). In (a), as the propagated circle comes from the left, we can propagate the red circle to the right. The blue circle is validated when its contact sets (blue rectangles) intersect the contact sets of the red circle (red rectangles). In the case of 3 contact sets (b), we have two possible directions for the propagation, and both are explored. If there is a small hole in the shape (c) that is completely covered by a contact set, the algorithm still considers that the contact set has a beginning and end and will execute correctly. (Color figure online)

2.4 Main Algorithm

Algorithm 1 summarizes our approach, which uses two parameters, σ and α where σ models the noise on the boundary and α gives a threshold on the Hausdorff distance between the shape modeled by the skeleton and the input shape, in order to discriminate between noise and details on the shape. We have chosen $\sigma = \sqrt{2}/2 \simeq 0.7$ and $\alpha = 3\sqrt{2}/2 \simeq 2.1$ and these choices are explained in Fig. 5.

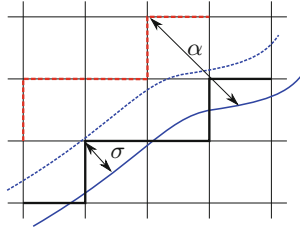


Fig. 5. Justification of the choices of σ and α . σ models the rasterization of the boundary, so it can vary from $-\sqrt{2}/2$ to $+\sqrt{2}/2$. Thus $\sigma = \sqrt{2}/2$. α models random noise: if there is a 1-pixel error on the boundary (like for the red line), the maximal distance between the boundary and the rasterized boundary is $3\sqrt{2}/2$. Thus $\alpha = 3\sqrt{2}/2$. (Color figure online)

Algorithm 1. Propagating skeleton algorithm

Parameters: σ : Noise on the boundary

$\alpha > \sigma$: Maximum distance to the shape generated by the skeleton

Data: \mathbf{S} : Binary grid representing the shape

\mathcal{B} : Boundary of the shape (with 4-adjacency connectivity)

Result: S : Shape skeleton

- 1 Estimation of the point P_0 , local maxima of $f_{\mathcal{B},\sigma}$ (cf. Section 2.1)
 - 2 $l_c \leftarrow \{P_0\}$
 - 3 **while** $l_c \neq \emptyset$ **do**
 - 4 Removing the first element P of l_c
 - 5 Construction of $\mathcal{C}(P, f_{\mathcal{B},\sigma}(P))$, maximal circle with center P and radius $f_{\mathcal{B},\sigma}(P)$
 - 6 Estimation of the tangency points between $\mathcal{C}(P, f_{\mathcal{B},\sigma}(P))$ and \mathcal{B} , with noise parameter σ and maximal distance to the skeleton α (cf. Section 2.2)
 - 7 Estimation by propagation of Q_1, \dots, Q_n the neighbor circles of $\mathcal{C}(P, f_{\mathcal{B},\sigma}(P))$ (cf. Section 2.3)
 - 8 Checking for junctions between Q_1, \dots, Q_n and l_c
 - 9 Adding Q_1, \dots, Q_n (not used in junction) in l_c
 - 10 **end**
-

The algorithm ends when the list l_c is empty. The algorithm can account for loops in the skeleton: For each Q_i that results from propagation of P , we check if there is a neighbor in the current center list l_c . If so, we interrupt the propagation from Q_i and add a loop-closing edge.

The complexity is on the order of $N^2 \log_2(S/\epsilon_d) \log_2(2\pi/\epsilon_\theta)$, where N is the number of boundary points and S their maximum pairwise distance. Note that the number of skeletal points is of the same order as the boundary points, and that the cost of propagation is the cost of the nested loop and the distance function. The complexity of the loop to search for the distance to the next center is $\log_2(d_m/\epsilon_d)$, where d_m is the maximal radius of a skeletal ball which

is bounded above by S . The complexity of the loop searching for the angle is $\log_2(\theta_m/\epsilon_\theta)$, where θ_m is the maximal angle between the extremities of two contact sets, bounded above by 2π . The complexity of the SDE is of order N .

We have described a skeletonization algorithm obtained by propagating centers of skeletal balls under the constraint that each circle maintains contact with the boundary. The resulting skeleton is centered, thin, topologically equivalent to the original shape, and complete. It is also robust to noise. In the next section, we show skeletonization results and quantify the strengths of this method.

3 Results and Evaluation

To evaluate our algorithm, we compare it to DECS [11], a grid-based method, and to pruned versions of the classical Voronoï skeletonization algorithm [14]. For pruning, we choose the three most common methods: scale-axis-transform [8], λ -medial axis [7] and θ -homotopy medial axis [19].

We use three evaluation criteria. The first is the Hausdorff distance between the initial shape and the shape reconstructed by the skeleton. This measures the influence of the parameter α , which sets a maximum distance to the original boundary. The second criterion measures the difference between the initial surface \mathbf{S} and the reconstructed surface \mathbf{S}' :

$$\frac{\mathcal{A}(\mathbf{S}\Delta\mathbf{S}')}{\mathcal{A}(\mathbf{S})},$$

where $\mathcal{A}(\mathbf{S}\Delta\mathbf{S}')$ is the area of the symmetric shape difference, and $\mathcal{A}(\mathbf{S})$ is the area of the original shape. These criteria do not assess noise reduction, so our third criterion counts the number of branches of the skeleton. Taken together, we evaluate the simplicity of the considered skeleton in providing a given level of shape fidelity. We perform tests on 1033 images from the MPEG-7 images [10]. See Table 1 for the averages obtained for each of the criteria and methods.

3.1 Analysis of the Results

Our method produces a simpler skeleton than the Voronoï skeleton while maintaining good shape fidelity. The scale-axis-transform pruning and λ -medial axis, in comparison, require many more branches to maintain a similar or lower fidelity. The θ -homotopy medial axis maintains fidelity with a reasonable number of branches, but our method outperforms it in the Hausdorff distance and produces a similar area difference with comparable number of branches. This shows that our method prioritizes the most informative branches both globally and locally when compared to other Voronoï-based methods. For certain shapes, such as the stingray, this results in a significant difference in shape quality. With similar numbers of branches, DECS gives better precision in terms of area difference, but ours is better for Hausdorff distance.

Table 1. Comparison between propagation skeletonization (with $\sigma = 0.7$ and $\alpha = 2.1$), Voronoi skeletonization with three different pruning methods and DECS [11] on 1033 images from [10]. Evaluation is done with three criteria: the Hausdorff distance in pixels, the area difference in percentages, and the number of branches. Red scores indicate where our algorithm is better, and blue where others are, showing the relation between the number of kept branches and the information lost.

| | Voronoi skeletonization | | | | | | | | | | DECS | Ours |
|-----------------------|-------------------------|-----|-----|-----|-----------|-----|-----|----------|-----|-----|------|------|
| | All | s | | | λ | | | θ | | | | |
| | | 1.1 | 1.2 | 1.5 | 1 | 2 | 4 | 60° | 80° | 90° | | |
| Mean Hausdorff (px) | 0 | 2.2 | 3.3 | 8.3 | 1.0 | 2.1 | 6.8 | 2.5 | 4.9 | 7.5 | 8.1 | 2.0 |
| Median Hausdorff (px) | 0 | 1.9 | 2.7 | 6.3 | 1.0 | 1.5 | 3.1 | 2.1 | 3.9 | 6.0 | 4.6 | 2.1 |
| Mean area (%) | 3.0 | 3.1 | 3.4 | 4.4 | 3.0 | 2.9 | 3.7 | 3.3 | 3.9 | 4.7 | 2.6 | 4.1 |
| Median area (%) | 2.2 | 2.5 | 2.8 | 4.1 | 2.1 | 2.1 | 2.5 | 2.7 | 3.5 | 4.2 | 2.1 | 3.2 |
| Mean branches | 655 | 196 | 96 | 26 | 432 | 192 | 60 | 36 | 16 | 10 | 9 | 16 |
| Median branches | 537 | 115 | 49 | 17 | 355 | 157 | 49 | 19 | 11 | 7 | 7 | 12 |

Computation time of our algorithm averages 982 ms for the MPEG-7 dataset on a i7-4800MQ CPU at 2.70 GHz. By comparison, the mean computation for the full Voronoi is 47 ms and filtering with the scale-axis-transform takes between 451 ms and 1118 ms. The λ -medial axis takes around 86 ms, and θ -medial axis takes between 788 ms and 1171 ms, depending on choices for their respective parameters. The execution time of the DECS is 16 ms.

We note the influence of the parameters σ and α . Lower values for these parameters produce more spurious branches, while higher values produce simpler skeletons that are less accurate. These values can then be tuned appropriately to image resolution or application.

4 Conclusion

We present a new skeletonization method that avoids creating uninformative branches by explicitly modeling boundary noise. The method relies on two parameters that are straightforward to choose and do not require hand-tuning for each shape: σ , the scale of noise on the boundary, and α , distinguishing between noise and detail on the shape. We compare our method with the Voronoi skeletonization pruned with three popular methods, and DECS. We provide quantitative evaluation showing that our method models the shape better in fewer branches in terms of Hausdorff distance.

This work could be extended quite naturally. We focus here on rasterized shapes, but we can generalize our approach to boundary representations with other types of noise. We can also accommodate lack of connectivity on the boundary, such as for point clouds, by redefining contact sets to recognize the directions in which we can propagate. Finally, we can use a similar method to compute 3D skeletons by propagation.

References

1. Aichholzer, O., Aigner, W., Aurenhammer, F., Hackl, T., Jüttler, B., Rabl, M.: Medial axis computation for planar free-form shapes. *Comput. Aided Des.* **41**(5), 339–349 (2009)
2. Amenta, N., Choi, S., Kolluri, R.K.: The power crust, unions of balls, and the medial axis transform. *Comput. Geom.* **19**(2–3), 127–153 (2001)
3. Attali, D., Montanvert, A.: Computing and simplifying 2D and 3D continuous skeletons. *Comput. Vis. Image Underst.* **67**(3), 261–273 (1997)
4. Bai, X., Latecki, L.J.: Path similarity skeleton graph matching. *IEEE Trans. PAMI* **30**(7), 1282–1292 (2008)
5. Blum, H.: A transformation for extracting new descriptors of shape. In: *Symposium on Models for the Perception of Speech and Visual Form* (1967)
6. Borgefors, G., Nyström, I., di Baja, G.S.: Computing skeletons in three dimensions. *Pattern Recognit.* **32**(7), 1225–1236 (1999)
7. Chazal, F., Lieutier, A.: The λ -medial axis. *Graph. Models* **67**(4), 304–331 (2005)
8. Giesen, J., Miklos, B., Pauly, M., Wormser, C.: The scale axis transform. In *Annual Symposium on Computational Geometry* (2009)
9. Lam, L., Lee, S.-W., Suen, C.Y.: Thinning methodologies - a comprehensive survey. *IEEE Trans. PAMI* **14**(9), 869–885 (1992)
10. Latecki, L.J., Lakamper, R., Eckhardt, T.: Shape descriptors for non-rigid shapes with a single closed contour. In: *IEEE Conference on CVPR* (2000)
11. Leborgne, A., Mille, J., Tougne, L.: Noise-resistant digital euclidean connected skeleton for graph-based shape matching. *J. Vis. Commun. Image Represent.* **31**, 165–176 (2015)
12. Leonard, K., Morin, G., Hahmann, S., Carlier, A.: A 2D shape structure for decomposition and part similarity. In: *ICPR* (2016)
13. Leymarie, F., Levine, M.D.: Simulating the grassfire transform using an active contour model. *IEEE Trans. PAMI* **14**(1), 56–75 (1992)
14. Ogniewicz, R., Ilg, M.: Voronoi skeletons: theory and applications. In: *IEEE CVPR* (1992)
15. Rumpf, M., Telea, A.: A continuous skeletonization method based on level sets. In: *Proceedings of the symposium on Data Visualisation* (2002)
16. Saha, P.K., Borgefors, G., di Baja, G.S.: A survey on skeletonization algorithms and their applications. *Pattern Recognit. Lett.* **76**(Suppl. C), 3–12 (2015)
17. Shen, W., Bai, X., Hu, R., Wang, H., Latecki, L.J.: Skeleton growing and pruning with bending potential ratio. *Pattern Recognit.* **44**(2), 196–209 (2011)
18. Siddiqi, K., Shokoufandeh, A., Dickinson, S.J., Zucker, S.W.: Shock graphs and shape matching. *Int. J. Comput. Vis.* **35**(1), 13–32 (1999)
19. Sud, A., Foskey, M., Manocha, D.: Homotopy-preserving medial axis simplification. In: *ACM Symposium on Solid and Physical Modeling* (2005)
20. Sundar, H., Silver, D., Gagvani, N., Dickinson, S.: Skeleton based shape matching and retrieval. In: *Shape Modeling International* (2003)
21. Vincze, M., Kővári, B.A.: Comparative survey of thinning algorithms. In: *International Symposium of Hungarian Researchers on Computational Intelligence and Informatics* (2009)
22. Xu, Y., Wang, B., Liu, W., Bai, X.: Skeleton graph matching based on critical points using path similarity. In: Zha, H., Taniguchi, R., Maybank, S. (eds.) *ACCV 2009*. LNCS, vol. 5996, pp. 456–465. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12297-2_44