Vector-Based Morphological Operations on Polygons Using Straight Skeletons for Digital Pathology

Daniel Felipe González Obando[1,2], Jean-Christophe Olivo-Marin[1], Laurent Wendling[3], and Vannary Meas-Yedid[1(✉)]

[1] BIA, Institut Pasteur, UMR 3691, CNRS, 75015 Paris, France
vmeasyed@pasteur.fr
[2] ED EDITE, Sorbonne Université, Collège doctoral, 75005 Paris, France
[3] Université Paris Descartes, LIPADE, Sorbonne Paris Cité, 75006 Paris, France

Abstract. In this work we present an efficient implementation of vector-based mathematical morphology operators applied to simple polygons by performing wavefront propagation and computing polygon straight skeletons. In Digital Pathology (DP), the slide scanner generates important volume of images from tissues called Whole Slide Image (WSI). The main goal of the DP is to detect the biological stained structures in order to quantify the tissue pathology, such as lesions or cancerous regions. We propose the use of Adapted Straight Skeletons on polygons as an efficient technique in time and memory, to improve image segmentation and image analysis. Thanks to the use of polygons instead of bitmaps to store segmentation results, the performance of straight skeletons depends only on the polygon control points. These straight skeletons can be applied in order to perform fast morphological operations such as dilation, erosion, closing, opening, skeletonizing. When combined, these operations offer different interesting outcomes: (i) multiple disjoint-segmented shapes can be linked together to create a joint skeleton, (ii) the topological structure of segmentation can be extracted as a straight skeleton. Then, it can be used as features for structural and spatial tissue analysis.

Keywords: Polygonal morphological operations · Straight skeletons · Digital Pathology

1 Introduction

The main goal of histopathology is to assess the biological tissue samples by the examination of the tissue in order to diagnose or prognose many diseases, such as cancers, or tissue lesions. Nowadays, slide scanners allow to acquire images with a high speed and high resolution. These images are called virtual slides, but image analysis tools do not exploit the high potential content and the clinical analysis of WSI largely remains the work of human experts. The images resulting from slide scanner devices are of high dimensions (image size greater than 1 GB) containing

M. Couprie et al. (Eds.): DGCI 2019, LNCS 11414, pp. 249–261, 2019.
https://doi.org/10.1007/978-3-030-14085-4_20

up to several thousands of objects and thus, are considered as big data. Therefore these images need to be processed by fast and efficient algorithms, both in time and in memory. In [17], the authors propose a sparse coding and multi-scale dynamic sampling approach. However, this approach is not adapted to object detection, such as capillaries or early tumor areas. In order to cope with this big data issue, object extraction will be coded by polygons which appear to be an efficient structure to represent Regions of Interest (ROIs) within the tissue such as vessels, capillaries, nuclei or pathological regions (fibrosis or cancerous areas). They allow to capture information with a small amount of points in the context of WSI. In this work, we will focus on manipulating and processing area objects extracted from image segmentation of the tissue. In order to tackle this issue, polygons appear to be an efficient structure to manipulate ROIs, in the context of WSI. Indeed, polygons allow to capture region information with a relatively small amount of points.

In this study, we will focus on morphological operations on polygons and we assume that the polygons are the inputs of our algorithms. The notion of skeleton was introduced by Blum [8] as a result of the Medial Axis Transform or Symmetry Axis Transform. The research on mathematical morphology for image processing has been very active, and we can distinguish two kinds of families: raster-image based approach and vector-based approach [28]. In this paper, as the aim is to handle polygons, we will only consider the vector based approach. Among them, we find a popular approach based on the principle of the Voronoï Diagram. The "Voronoï skeleton" is computed on a polygonal representation of an object, which is the sampling of contour points, and is computed through geometry algorithms. This method allows to compute the skeleton based on the edges formed from vertices of the shape, but it does not allow to perform morphological operations such as dilations or erosions. Minkowski sums allow for computing offsets at specific distances using the sum of a circle with the polygon boundaries [29]. Although this method can compute smooth contours, it does not allow to recover the internal structure of the shape and requires recomputing the sum when multiple distances are required, which can be computationally expensive. Another category is to use straight skeletons [2]. The skeleton of a polygon is a thin version of that polygon where every point is equidistant to the polygon boundaries. Unlike skeletons based on Voronoï diagram which can contain parabolic curves [1], straight skeletons contain only straight segments. This property allows for easier and faster operations by avoiding calculus on curves, notably the possibility to compute multiple dilations and erosions at different distances in linear time, with respect to the number of vertices in the polygon. Several works have used this approach to compute straight skeletons (see Table 1). Aichholzer and Aurenhammer [3] set the base grounds by formalizing the concept of straight skeleton as a **planar straight-line graph**, as well as introducing their wavefront propagation method performed with a time complexity of $O(n^2 \log n)$, where n is the amount of vertices in the polygon. Eppstein and Erickson presented a subquadratic algorithm that used efficient closest-pair data structures to optimize processing time up to $O(n^{1+\epsilon} + n^{8/11+\epsilon} r^{9/11+\epsilon}) \subseteq O(n^{17/11+\epsilon})$, where r is the number of reflex vertices in the polygon and $0 < \epsilon \ll 1$ is a fixed value [13].

Table 1. Overview of existing straight skeleton algorithms: n: total number of vertices, r: number of reflex vertices.

Algorithm	Time	Memory space
Aichholzer [2]	$O(nr\log n)$	O(n)
Aichholzer [3]	$O(n^3\log n)$ (pract. $O(n\log n)$)	O(n)
CGAL [9]	$O(n^2\log n)$	$O(n^2)$
Eppstein [13]	$O(n^{1+\epsilon} + n^{8/11+\epsilon}\log n)$	ditto
Cheng	$O(n^{1+\epsilon} + n^{8/11+\epsilon}r^{9/11+\epsilon})$	O(n)
STALGO [18]	$O(n^2\log n)$ (pract. $O(n\log n)$)	O(n)

Cacciola also presented an implementation currently available in CGAL to compute straight skeletons with a complexity of $O(n^2 \log n)$ in time, and $O(n^2)$ in space [9]. More recently, Huber and Held have presented an implementation (called STALGO) by computing the straight skeleton aided by a motorcycle graph in $O(n \log n)$ in time and $O(n)$ in space [18]. We based our solution by following this idea [18], with adaptations to fulfill our needs in Digital Pathology.

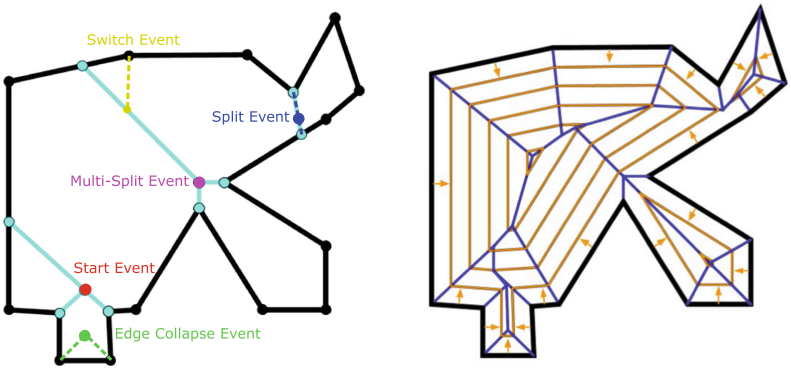
2 Straight Skeletons

The straight skeleton $\mathcal{S}(P)$ of a simple polygon P is defined by a wavefront propagation process. A wavefront $\mathcal{W}_P(t)$ of P is formed by edges that are parallel to those of P moving all at unit speed to the interior of P for erosion computing (resp. to the exterior of P for dilation). $\mathcal{W}_P(t)$ is also defined by the vertices moving on the internal angular bisectors of the vertices of P. These vertices move until either of the following events happen:

- An edge *collapses* when two linked vertices join together at the same point at a given time t. From this event, a new vertex is created following the direction computed from neighboring edges of the collapsed edge.
- An edge is *split* into two new edges when a vertex forming a reflex angle (an angle larger than π on the propagation side) meets a wavefront edge. This results in two new edges with the same speed of the original edge, on each side of the splitting vertex.

This two kinds of events are repeatedly handled, up until all wavefront edges have collapsed and no new events happen. At this point, the skeleton $\mathcal{S}(P)$ of P is defined as the set of loci that are traced out by the wavefront vertices (see the magenta lines on Fig. 2b).

Motorcycle Graph. Computing straight skeletons on convex simple polygons is really straight forward since only edge collapses may happen during the wavefront propagation process. In contrast, non-convex simple polygons require



(a) Motorcycle graph and initial events (b) Wavefront propagation and straight skeleton

Fig. 1. (a) The cyan segments split the initial concave polygon into convex polygons, (b) straight skeletons are represented by blue segments (Color figure online)

detecting potential split events during the propagation. A naive algorithm to detect this kind of events could take $O(n^2 \log n)$ in time by looking all possible splitting events in the planar straight-line graph. However, using a motorcycle graph for this end comes convenient because it allows to detect all the trajectories of splitting vertices that occur during the propagation in $O(n \log n)$ time. Motorcycle graphs divide polygons into convex tessellations that guarantee no split events inside the subdivisions, thus reducing the complexity of the calculation of $\mathcal{S}(P)$ [18].

On a motorcycle graph $\mathcal{M}(P)$ a *"motorcycle"* m_i is launched for each reflex vertex p_i of the P. These motorcycles are allowed to move on the bisectors of the edges leaving a trace behind them until either they crash with another motorcycle or trace, or escape to infinity if no crash happens (see green lines on Fig. 2a). When two or more motorcycles crash at the same time, the pair of successive motorcycles forming a reflex angle is used to compute a new motorcycle. The reader is referred to [18] for more details.

Straight Skeleton Construction. Using all convex and reflex vertices from the input polygon, as well as the motorcycle crash points from $\mathcal{M}(P)$, the wavefront $\mathcal{W}_P(t)$ is initialized. From this wavefront it is possible to compute $\mathcal{S}(P)$ by following the evolution of $\mathcal{W}_P$. Vertices on $\mathcal{W}_P(t)$ move either in the direction of the bisector of the vertices incident edges, or in the inverse direction of the crashed motorcycles from their crash point. The wavefront propagation is then processed in chronological order (see Algorithm 1), handling each crash event to build $\mathcal{S}(P)$ step by step (see event examples in Fig. 1a and the wavefront evolution in Fig. 1b). Once all crash events are processed, $\mathcal{S}(P)$ is fully conformed by the trace of the wavefront vertices. With this result, multiple morphological operations are possible, presented in the following section.

3 Morphological Operations on Polygons

Although straight skeletons are easier to handle and to model, they still have a drawback. As they are built by evolving the wavefront at a constant speed, the speed of reflex vertices tends to be faster than the rest of the wavefront. This causes premature edge splitting and results in a very disturbed skeleton. To address this issue, we propose a refined straight skeleton by adding more than one vertex on reflex angles. This allows to homogenize the speed of vertices on the wavefront avoiding premature split events. As a result, the produced skeleton is smoother (see Figs. 2b and 6b, c).

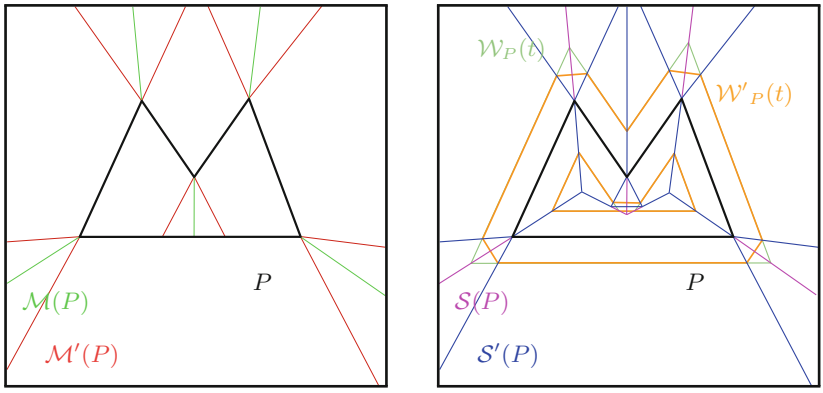
3.1 Refined Straight Skeletons

To provide a more accurate method, we propose to launch two motorcycles instead of just one on reflex angles when creating the motorcycle graph (see red lines against green lines on Fig. 2a). From this, the wavefront is built in the same way as the original straight skeleton. Performing the wavefront evolution with this change allows to prevent excessive evolution on the reflex wavefront vertices (see orange lines against light green lines on Fig. 2b), which can create early polygon divisions when performing dilations and erosions.

Result: $\mathcal{S}(P)$
$\mathcal{W}_P \leftarrow initializeWavefront(P, \mathcal{M}_P)$;
$\mathcal{S}_P \leftarrow initalizeStraightSkeleton(\mathcal{W}_P)$;
Fill event priority queue Q with initial events;
while Q *is not empty* **do**
  $e \leftarrow Q.poll()$;
  if *isEventStillValid(e)* **then**
    $newPossibleEvents \leftarrow processEvent(e)$;
    `// This will update` $\mathcal{W}_P$ `and` $\mathcal{S}(P)$`)`
    $Q.addAll(newPossibleEvents)$;
  else
    continue;
  end
end

Algorithm 1. Event handling on the wavefront propagation when building the straight skeleton $\mathcal{S}(P)$.

Dilating and Eroding Polygons. With this consistent change on the wavefront construction it is now possible to obtain dilations and erosions that are closer to those made by classical bitmap methods [24]. In this case, dilations are created by recreating the wavefront $W_P(t)$ from $\mathcal{S}(P)$. For this, the edges of S are followed from the leafs of the graph S up to the desired time (distance) t. From there the dilation is created by connecting the valid edges at instant t (see Fig. 3).



(a) Motorcycle graph (b) Straight skeleton and offset at time t

Fig. 2. Adaptation of Huber and Held's method to improve offset results. In the motorcycle graph (a), Huber's method creates only one motorcycle for each reflex vertex ($\mathcal{M}(P)$), whereas our method creates two ($\mathcal{M}'(P)$). In the straight skeleton (b), Huber's method creates only one wavefront vertex for each reflex vertex ($\mathcal{S}(P)$), producing sharp edges ($\mathcal{W}_P(t)$). Our method creates two vertices on reflex vertices ($\mathcal{S}'(P)$), accentuating sharp polygon vertices on the wavefront. These changes produce smoother offsets ($\mathcal{W}'_P(t)$). (Color figure online)

3.2 Medial Axis

Since its introduction by Blum [8], the notion of medial axis has proven to be useful for many purposes in morphological analysis. He introduced the medial axis of a shape X as the set of points $x \in X$ that have more than one nearest point on the boundary ∂X of X. To compute this medial axis several solutions have been proposed making use of discrete geometry [14,16,23], digital topology [10,22,26,27], computational geometry [4,5,21], partial differential equations [25], and level-sets [20]. In our case, we focus on the medial axis for skeletons strictly made out of straight line segments, which has not been yet formalized as far as we know.

Here we propose a medial axis based on the straight skeleton $\mathcal{S}(P)$ of a polygon P, with only one parameter for pruning. Let $\mathcal{M}(P)$ be the straight medial axis of the polygon P defined as the edges of $\mathcal{S}(P)$ filtered by the minimum allowed distance from an end-point of any skeleton edge to the shape boundaries. That is, any edge $e \in \mathcal{S}(P)$ with an end-point whose distance to the shape boundary is smaller than a given threshold will be discarded as part of the straight medial axis. This definition can cause topology issues as the connectivity of the medial axis can differ from that of the skeleton (see an example in Fig. 4a). To address this issue we perform the edge filtering using a priority on the minimal distance from the edge to the shape boundaries. This means looking both edge ends and making sure that at least one of them is of degree one. Otherwise the edge is not removed keeping the connectivity of the result (see Fig. 4b).

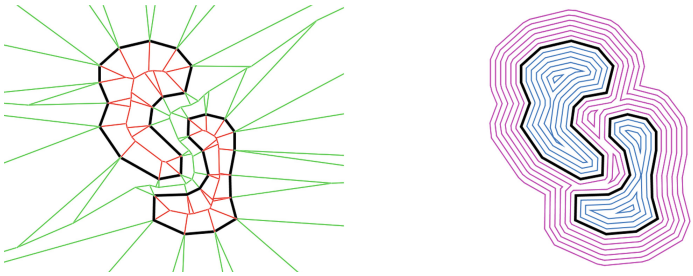
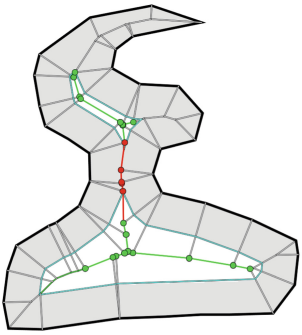


Fig. 3. Refined straight skeletons of two polygons. Inwards skeleton in red and outwards in green. Successive dilations (in magenta) and erosions (in cyan) created from refined straight skeletons. (Color figure online)

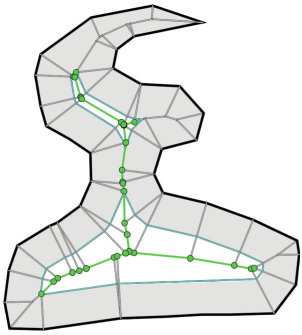
3.3 Weighted Straight Skeletons for Directional Operators

The weighted straight skeleton was first proposed in 1999 by Eppstein and Erickson [13], where the wavefront edges may move with arbitrary but fixed speeds. Since its definition some partial implementations have been proposed to address several of the issues associated to dealing with different speeds on edges. These implementations handle issues such as negative and zero edge weights [19], edge crash event ambiguities [6,7], and algorithm complexity improvements [12]. This kind of skeletons are interesting for oriented morphological operations such as oriented dilations/erosions to restrict the search space of neighbourhood based on the orientation with respect to the shape, etc. We have started the development of the weighted straight skeletons on convex polygon by taking into account two main issues (see Fig. 5):

- The collapse of an edge with two parallel neighboring edges with different weights. In this case a decision must be taken on the speed of the resulting edges joined at this point of the propagation. One option is to take the highest or lowest speed between the involved edges and apply it to them all. Another option (the one chosen for our implementation) is to let the edges following their own propagation speed and adding a zero-speed edge joining them. This way helps to keep speeds without changes and the wavefront continues to be consistent on the propagation.
- On multiple split events, special care must be taken while reconnecting edges in order to keep a consistent topology of the wavefront. If other events happen at the same time, these events must be processed before processing the splitting to avoid confusion when reconnecting edges participating on the event.

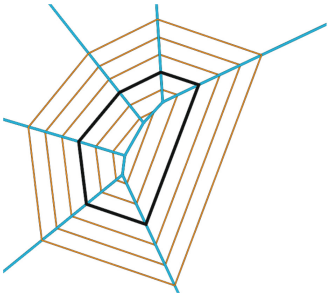


(a) axis without connectivity constraint

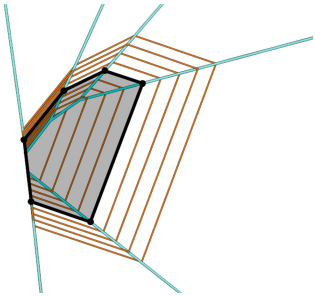


(b) axis with connectivity constraint

Fig. 4. A straight medial axis example. When a connectivity constraint is not applied while pruning the straight skeleton, its topology can differ from that of the original shape. (a) shows in red the edges ignored when no connectivity constraint is applied when building the medial axis (in green) with a threshold criterion expressed by the blue polygons. (b) shows the result of the connectivity-constrained straight medial axis. (Color figure online)



(a) Non weighted straight skeletons.



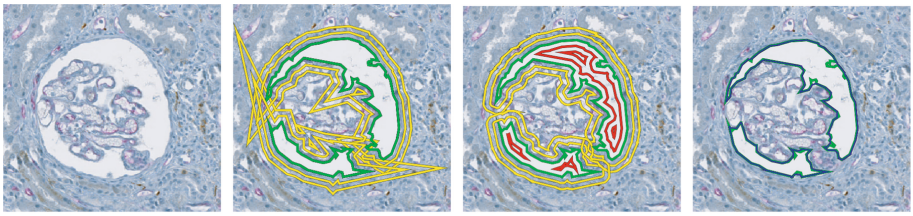
(b) Weighted straight skeletons.

Fig. 5. Comparison between the unweighted and weighted straight skeletons (in cyan) applied to the same polygon, with successive dilations (in orange). (Color figure online)

4 Results

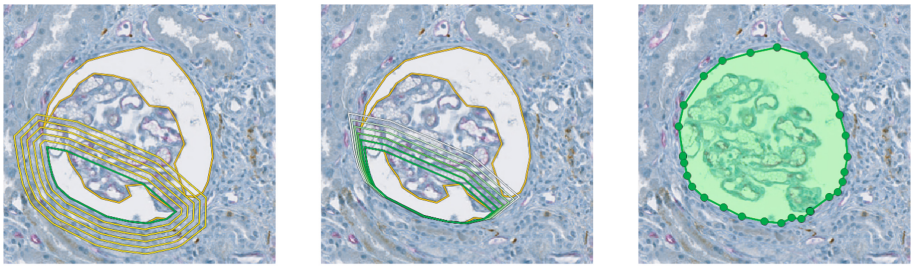
4.1 Applications to Digital Pathology

The structures in the tissue image can be split into several segments due to several factors: the sample slice cut, the staining, the acquisition setup, etc. and can result in over-segmented objects. Figure 7 shows the lumen around glomerulus split into two regions. To help the object detection, we ran the weighted straight skeleton algorithm to perform the directional dilation in order to find the second segmented polygon thanks to this spatial context processing.



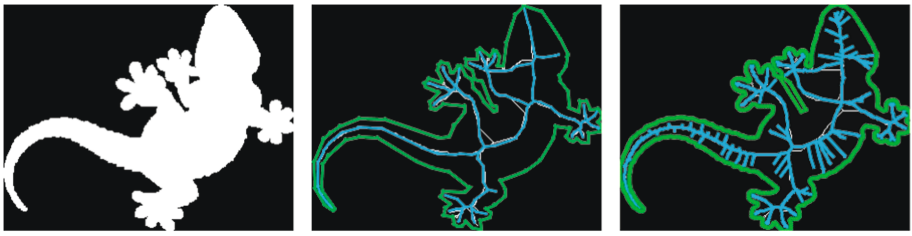
(a) (b) (c) (d)

Fig. 6. Morphological operations applied to a segmentation of a glomerulus from a stained kidney tissue (input polygons in green). (b) Dilation of polygon (yellow) using Huber's method on polygons. (c) Dilations (yellow) and erosions (red) using our method. (d) Closing (*dilation o erosion*) using our method (Blue) (Color figure online)



(a) (b) (c)

Fig. 7. Weighted straight skeletons applied to a glomerulus segmentation (input polygon in orange). (a) Dilations of convex polygon in green using non-weighted straight skeleton algorithm. (b) Directional dilations (green) of weighted straight skeletons. (c) Ellipse detected after fusion of the two lumen segments (Color figure online)



(a) (b) (c)

Fig. 8. The lizzard image used for comparison tests. In green, the polygons used for the straight skeleton algorithm and in cyan, the medial axis with a distance value set to 1. Note that the polygon is not composed by the same amount of points in (b) and (c). (Color figure online)

Table 2. Comparison of skeletonization methods on Fig. 8.

Image size W × H (px)	Points in polygon	[27] Time (ms)	[15] Time (ms)	[30] Time (ms)	Proposed method time (ms)
250 × 194	52	13	137.239	110.254	10.9
300 × 233	57	17	145.683	143.966	11.43
350 × 272	67	33	155.221	145.637	12.3
400 × 311	74	42	172.495	158.056	17.17
450 × 350	83	63	187.624	172.375	15.57
500 × 389	87	89	202.811	182.217	19.45
550 × 428	95	121	224.427	187.787	18.33
600 × 467	98	143	245.708	190.95	21.64
650 × 506	101	183	274.336	199.867	22.63
700 × 545	108	297	299.136	228.684	23.07
750 × 584	118	463	321.907	248.115	24.4
800 × 623	120	646	386.24	289.133	22.05
850 × 662	127	756	423.903	352.17	24.2
900 × 701	169	971	457.716	361.06	38.89
1000 × 779	166	1452	664.056	431.305	38.17
1500 × 1169	169	5394	1625.04	1102.85	34.33
2000 × 1559	172	10407	3528.13	2361.7	39.84
2500 × 1949	174	17504	6714.95	4563.72	39.52
3000 × 2339	181	28505	11456.6	7525.11	44.73

4.2 Evaluation

The complexity of the algorithms in time and memory can be found in Table 1. We have performed a quick performance comparison of skeletonization algorithms. Table 2 presents the comparison between 3 available and considered as fast algorithms of skeletonization: Vincent [27], Guo and Hall [15] and Zhang and Suen [30]. The algorithms have been performed on the lizzard image that shows several levels of resolution (see Fig. 8). The method based on straight skeletons is faster than the other methods especially for big shape sizes. Figure 8 shows the medial axis extracted from straight skeletons. Figure 8b and c highlight the fact that the medial axis could vary according to the number of points describing the polygon: more points will give more branches of skeletons. As we do not have any groundtruth to make a fair quantitative comparison of the skeleton accuracy, we just show some qualitative results. The white points describe the skeleton result from the method [30] in Fig. 8b and the method [15] in Fig. 8c. Figure 9 shows the medial axis of a polygon containing holes (image of letter B). We assume that a good skeleton should lay on the maximum value of the Chamfrein's distance map [8]. Our medial axis lays close to this maximum distance value.

(a) (b) (c)

Fig. 9. Medial axis on a shape containing holes. (a) polygons of B shape and its medial axis derived from the straight skeleton algorithm. (b) medial axis transform (c) overlay of the proposed medial axis on the Chamfrein's distance map.

5 Conclusion and Perspective

Straight skeletons are thus an interesting and fast tool in the process of tissue analysis, but could also be applied as part of a characterization framework in other domains such as Geographical Data or Document Analysis. This approach can be used for map generalization: for example, river or road to line segment simplification. We want to explore other propagation modes for the weighted straight skeleton method. Other polygonal extraction methods to set the initial one are under consideration to show the robustness of the achieved representation, for instance considering blurred segments calculated from different widths of discrete curves [11].

References

1. Aggarwal, A., Guibas, L.J., Saxe, J., Shor, P.W.: A linear-time algorithm for computing the Voronoï diagram of a convex polygon. Discrete Comput. Geom. **4**(6), 591–604 (1989)
2. Aichholzer, O., Alberts, D., Aurenhammer, F., Gartner, B.: Straight skeletons of simple polygons. In: Proceedings of 4th International Symposium of LIESMARS, pp. 114–124 (1995)
3. Aichholzer, O., Aurenhammer, F.: Straight skeleton for general polygonal figures in the plane. In: Samoilenko, A.M. (ed.) Voronoï's Impact on Modern Science, vol. 2, pp. 7–21. Institute of Mathematics of the National Academy of Sciences of Ukraine (1998)
4. Attali, D., Lachaud, J.O.: Delaunay conforming iso-surface, skeleton extraction and noise removal. Comput. Geom. **19**(2), 175–189 (2001)
5. Attali, D., Montanvert, A.: Modeling noise for a better simplification of skeletons. In: Proceedings of 3rd IEEE ICIP, vol. 3, pp. 13–16, September 1996

6. Biedl, T., Held, M., Huber, S., Kaaser, D., Palfrader, P.: A simple algorithm for computing positively weighted straight skeletons of monotone polygons. Inf. Process. Lett. **115**(2), 243–247 (2015)
7. Biedl, T., Held, M., Huber, S., Kaaser, D., Palfrader, P.: Weighted straight skeletons in the plane. Comput. Geom. **48**(2), 120–133 (2015)
8. Blum, H.: A transformation for extracting new descriptors of shape. In: Wathen-Dunn, W. (ed.) Models for the Perception of Speech and Visual Form, pp. 362–380. MIT Press, Cambridge (1967)
9. Cacciola, F.: 2D straight skeleton and polygon offsetting. In: CGAL User and Reference Manual, 4.10.1 edn. CGAL Editorial Board (2017)
10. Davies, E., Plummer, A.: Thinning algorithms: a critique and a new methodology. Pattern Recognit. **14**(1), 53–63 (1981)
11. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. Comput. Graph. **30**(1), 30–36 (2006)
12. Eder, G., Held, M.: Computing positively weighted straight skeletons of simple polygons based on a bisector arrangement. Inf. Process. Lett. **132**, 28–32 (2018)
13. Eppstein, D., Erickson, J.: Raising roofs, crashing cycles, and playing pool: applications of a data structure for finding pairwise interactions. Discrete Comput. Geom. **22**(4), 569–582 (1999)
14. Ge, Y., Fitzpatrick, J.M.: On the generation of skeletons from discrete euclidean distance maps. IEEE Trans. Pattern Anal. Mach. Intell. **18**, 1055–1066 (1996)
15. Guo, Z., Hall, R.W.: Parallel thinning with two-subiteration algorithms. Commun. ACM **32**(3), 359–373 (1989)
16. Hesselink, W.H., Roerdink, J.B.T.M.: Euclidean skeletons of digital image and volume data in linear time by the integer medial axis transform. IEEE Trans. Pattern Anal. Mach. Intell. **30**(12), 2204–2217 (2008)
17. Huang, C.H., Veillard, A., Roux, L., Lomenie, N., Racoceanu, D.: Time efficient sparse analysis of histopathological whole slide images. Comput. Med. Imaging Graph. **35**(7), 579–591 (2011)
18. Huber, S., Held, M.: A fast straight-skeleton algorithm based on generalized motorcycle graphs. Int. J. Comput. Geom. Appl. **22**(5), 471–498 (2012)
19. Kelly, T.: Unwritten procedural modeling with the straight skeleton. Ph.D. thesis, University of Glasgow (2013)
20. Kimmel, R., Shaked, D., Kiryati, N., Bruckstein, A.M.: Skeletonization via distance maps and level sets. Comput. Vis. Image Underst. **62**(3), 382–391 (1995)
21. Ogniewicz, R., Kübler, O.: Hierarchic Voronoï skeletons. Pattern Recognit. **28**(3), 343–359 (1995)
22. Pudney, C.: Distance-ordered homotopic thinning. Comput. Vis. Image Underst. **72**(3), 404–413 (1998)
23. Remy, E., Thiel, E.: Exact medial axis with euclidean distance. Image Vis. Comput. **23**(2), 167–175 (2005)
24. Serra, J.: Image Analysis and Mathematical Morphology. Academic Press Inc., Orlando (1983)
25. Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.W.: Hamilton-jacobi skeletons. Int. J. Comput. Vis. **48**(3), 215–231 (2002)
26. Talbot, H.: Euclidean skeletons and conditional bisectors. In: 1992 Visual Communications and Image Processing, pp. 862–876 (1992)
27. Vincent, L.M.: Efficient computation of various types of skeletons. In: Proceedings of SPIE, vol. 1445, pp. 1445–1445-15 (1991)

28. Vizilter, Y.V., Pyt'ev, Y.P., Chulichkov, A.I., Mestetskiy, L.M.: Morphological image analysis for computer vision applications. In: Favorskaya, M.N., Jain, L.C. (eds.) Computer Vision in Control Systems-1. ISRL, vol. 73, pp. 9–58. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-10653-3_2
29. Wein, R., Baram, A., Flato, E., Fogel, E., Hemmer, M., Morr, S.: 2D minkowski sums. In: CGAL User and Reference Manual, 4.10.1 edn. CGAL Editorial Board (2017)
30. Zhang, T.Y., Suen, C.Y.: A fast parallel algorithm for thinning digital patterns. Commun. ACM **27**(3), 236–239 (1984)