Extracting Noise-Resistant Skeleton on Digital Shapes for Graph Matching

Aurélie Leborgne[1,2], Julien Mille[1,3], and Laure Tougne[1,4]

[1] Université de Lyon, CNRS
[2] INSA-Lyon , LIRIS, UMR5205, F-69621, France
aurelie.leborgne@liris.cnrs.fr
[3] Université Lyon 1, LIRIS, UMR5205, F-69622, France
julien.mille@liris.cnrs.fr
[4] Université Lyon 2, LIRIS, UMR5205, F-69676, France
laure.tougne@liris.cnrs.fr

Abstract. In order to match shapes using their skeletons, these ones should be thin, robust to noise, homotopic to the shape, consequently, connected. However, these properties are difficult to obtain simultaneously when the shape is defined on a discrete grid. In this paper, we propose a new skeletonization algorithm, which has all these properties. Based on the Euclidean distance map, the algorithm extracts the centers of maximal balls included in the shape and uses the ridges of distance map to connect them. A post-processing is then applied to thin and prune the resulting skeleton. The proposed method is compared to three fairly recent methods to highlight the good properties of the obtained skeleton.

1 Introduction

The skeleton is a relevant structure for shape matching. To compare the skeletons of different shapes, the idea is to convert the skeletons into graphs, which will be matched (branches being edges and, junction points and ending points being vertices). However, in order to easily convert the skeleton into a graph, it should have at least the following properties, which are not obviously obtained when the shape is represented by points in $\mathbb{Z}^2$:

- connection: if the skeleton is not connected, the graph is not connected and is not topologically equivalent to the shape;
- thinness (1-pixel width): a thick skeleton generates path extraction problems.

Moreover, in order to obtain effective and pertinent matchings in the context of real objects, it is necessary to construct skeletons robust to noise. Notice that this last property is rarely satisfied by the algorithms of the literature, for which the slightest deformation of the border usually generates a branch.

Let us consider now such algorithms. In $\mathbb{Z}^2$, we can classify them as follows:

- skeletonization methods based on thinning [1,2]: In an intuitive manner, it consists in "peeling" the shape in order to obtain a set of connected points

G. Bebis et al. (Eds.): ISVC 2014, Part I, LNCS 8887, pp. 293–302, 2014.

with a single pixel width, which preserves the topology of the shape. In other words, thinning is an operation that aims to iteratively remove non-terminal simple points. The main problem of these methods is the lack of noise resistance.
– skeletonization methods based on a distance map [3,4,5,6,7]: The objective is to identify the key points on the chosen distance map, where each pixel is labeled with the value of its distance to the nearest background pixel. The problem of these algorithms is to successfully extract enough points to obtain a connected and thin skeleton.

As mentioned previously, object recognition requires a shape representation which is resistant to minor changes, but the main drawback of the skeleton is, generally, its sensitivity to noise on the shape boundary. This is the reason why it is customary to use a regularization procedure, which can be of two types:

– smoothing the boundary of the shape: this is done before the computation of the skeleton points, in order to remove unwanted boundary noise and discretization artefacts [8]. The main drawback is that we do not control the effects of smoothing on the general appearance of the skeleton.
– deleting unwanted branches: this is a post-processing step called pruning [9,10]. It is based on local or global salience measures. The difficulty here is to remove "noisy branches" without removing any meaningful parts of the skeleton.

The proposed algorithm, called DECS, computes the distance map, not only to extract centers of maximal balls, but also to connect them. The obtained skeleton is then connected and thin. Moreover, as the connection method is based on a Laplacian of Gaussian (LoG) filter applied to the distance map, the obtained skeleton is more robust to noise than thinning algorithms.

We describe in details the proposed method in Section 2. Then, three state-of-the-art methods will be compared to ours in Section 3: the K3M method, the extraction of the Euclidean skeleton based on a connectivity criterion, namely Choi*et al*'s method and the Hamilton-Jacobi skeleton method.

2 Extraction of Digital Euclidean Connected Skeleton (DECS)

2.1 Overview

Let us denote $\mathcal{I} \subset \mathbb{Z}^2$ an image of size $M \times N$ and $\mathcal{S} \subset \mathcal{I}$ a shape. More precisely, we consider that $\mathcal{S}$ is 8-connected or 4-connected. We use this hypothesis to ensure the connectivity of the skeleton. In case of non connected shapes, one skeleton can be extracted per connected component. Furthermore, notice that $\mathcal{S}$ can have holes or not. Let $\boldsymbol{p}$ be a pixel of $\mathcal{S}$ and $N_8(\boldsymbol{p})$ the set of 8-connected neighbors of $\boldsymbol{p}$.

Figure 1 summarizes the proposed method.

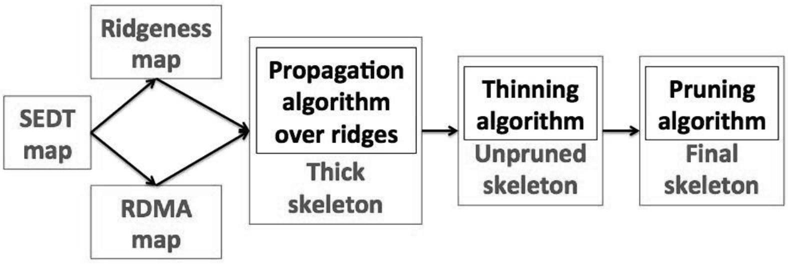


Fig. 1. Flowchart of the DECS algorithm

The proposed method is based on the Squared Euclidean Distance Transformation, namely SEDT, computed using [11] (*cf* Figure 3a). On the one hand, the obtained Reduced Discrete Madial Axis map (RDMA map) contains the set of centers of maximal balls (*cf* Subsection 2.2) [6] that are points belonging to the skeleton, but might not be connected. On the other hand, we use a LoG filter on the Euclidean distance map and compute a Ridgeness map (*cf* Subsection 2.3). Then, the main idea is to combine these two maps to obtain an "over-connected skeleton". In other words, we connect maximal balls using ridge lines of the distance map. Afterwards, we refine this "over-connected skeleton" by a post-processing, which consists in thinning it (*cf* Subsection 2.4) and keeping only the main connected branches (pruning).

2.2 Reduced Discrete Medial Axis (RDMA) [6]

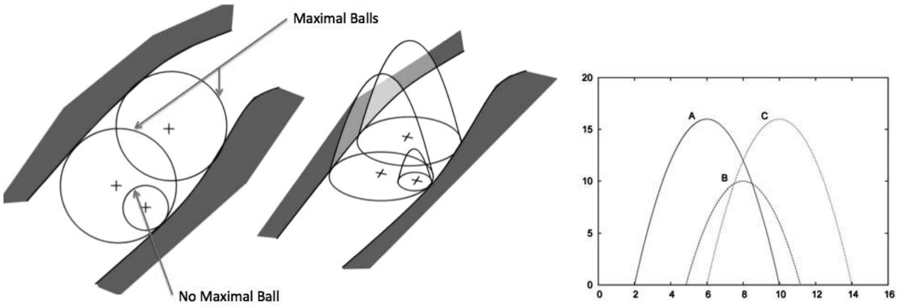
The Squared Euclidean Distance Transformation maps to each point $\boldsymbol{p}$ with coordinates (i, j) in shape $\mathcal{S}$ the square of the radius of the largest ball centered at $\boldsymbol{p}$. The obtained map is called SEDTmap. As the skeleton contains centers of maximal balls, the main idea is to recover such points using SEDTmap. A maximal discrete ball is a discrete ball contained in the shape not entirely covered by another discrete ball contained in the shape.

To obtain maximal discrete balls of a shape, we use Coeurjolly and Montanvert's method [6], which is separable and linearly proportional to n, the number of pixels in $\mathcal{S}$. The general idea is to represent discrete balls by elliptic paraboloids (*cf* Figure 2a) to retain only those belonging to the upper envelope. To illustrate this, if a sheet is placed on the set of paraboloids, hugging perfectly curves, we retain only those in contact with the sheet. Centers of maximal balls are the centers of paraboloids, which have been retained (*cf* Figure 2b).

Figure 3b shows an example of obtained RDMA. We can see that the RDMA is not connected, which is its major drawback. In the following, we extract features on the distance map that will be used to build a connected skeleton.

2.3 Extraction of Ridgeness Map

We can notice that the branches of the skeleton correspond to the ridges of the distance map. The Laplacian operator applied to the Euclidean distance map allows to extract them. However, the Laplacian is very sensitive to noise when



(a) **(b)**

Fig. 2. a) (left): Maximal balls, (right): Representation of balls by elliptic paraboloids, b) $\{A, B, C\}$ is the representation in 2D of three balls that belong to the medial axis. $\{A, C\}$ represent maximal balls, but not $\{B\}$ because $\{B\}$ is covered by the union of $\{A\}$ and $\{C\}$ [6].

computed alone on the distance map. Hence, we rather convolve the Euclidean distance map with the negative Laplacian of a Gaussian (LoG) filter of standard deviation σ.

$$\text{rdg}(x, y) = -(EDT * \text{LoG})(x, y)$$

An example is shown in figure 3c. We can see that only the main branches are highlighted. As the obtained mask is separable, the complexity of the filtering operation is in $O(\sigma n)$.

A simple thresholding of the ridgeness map is not sufficient to extract a connected skeleton because the main branches that should be kept can be disconnected, as the values of the ridgeness map are not constant along branches. The proposed idea is to combine the RDMA and the ridgeness map rdg to determine the situation of the main branches - where there are enough maximal balls or where the ridgeness map has sufficiently high values - and connect them using the ridgeness map that acts as a guide.

2.4 DECS Algorithm

In this subsection we present the heart of our algorithm: this is the propagation of maximal balls over ridges (Algorithm 1). This algorithm generates a labeling $H = \{h(i, j)\}_{i,j}$ indicating whether a point is a center of maximal ball or a ridge. Possible labels are $\{NONE, MAX\_BALL, STRONG\_RIDGE, RIDGE\}$.

$MAX\_BALL - \boldsymbol{p}$ such that $\boldsymbol{p}$ is the center of a maximal ball
$STRONG\_RIDGE - \boldsymbol{p}$ such that $rdg(\boldsymbol{p}) \geq th_{ridge-high}$
$RIDGE - \boldsymbol{p}$ such that $rdg(\boldsymbol{p}) \in [th_{ridge-low}; th_{ridge-high}[$
$NONE - \boldsymbol{p}$ such that $\boldsymbol{p} \notin \{MAX\_BALL, STRONG\_RIDGE, RIDGE\}$

$th_{ridge-low}$ is always equal to 0.05 and $th_{ridge-high}$ varies between $th_{ridge-low}$ and 1.1. These values have been fixed by experimentation. The algorithm uses a propagation technique starting from the center of the largest maximal ball, which is a mandatory part of the skeleton. We consider this is a relevant starting point, as it belongs to the main branch. The propagation algorithm keeps only the points that are connected by a path of ridge points ($STRONG\_RIDGE$ or $RIDGE$) or centers of maximal balls (MAX_BALL). An example of result is shown in figure 3d. At the end of this algorithm, the skeleton is the set of points with a label different from $NONE$. It contains all important branches but has one drawback. Indeed, the skeleton has a thickness that can be greater than one.

To obtain a thin skeleton, we used the MB2 thinning algorithm [12,13] that iteratively suppresses simple points according to their configuration. The result of this algorithm is a skeleton, S_k, such that all the branches are one pixel thick considering 8-connectivity. In other words, we cannot suppress simple points without

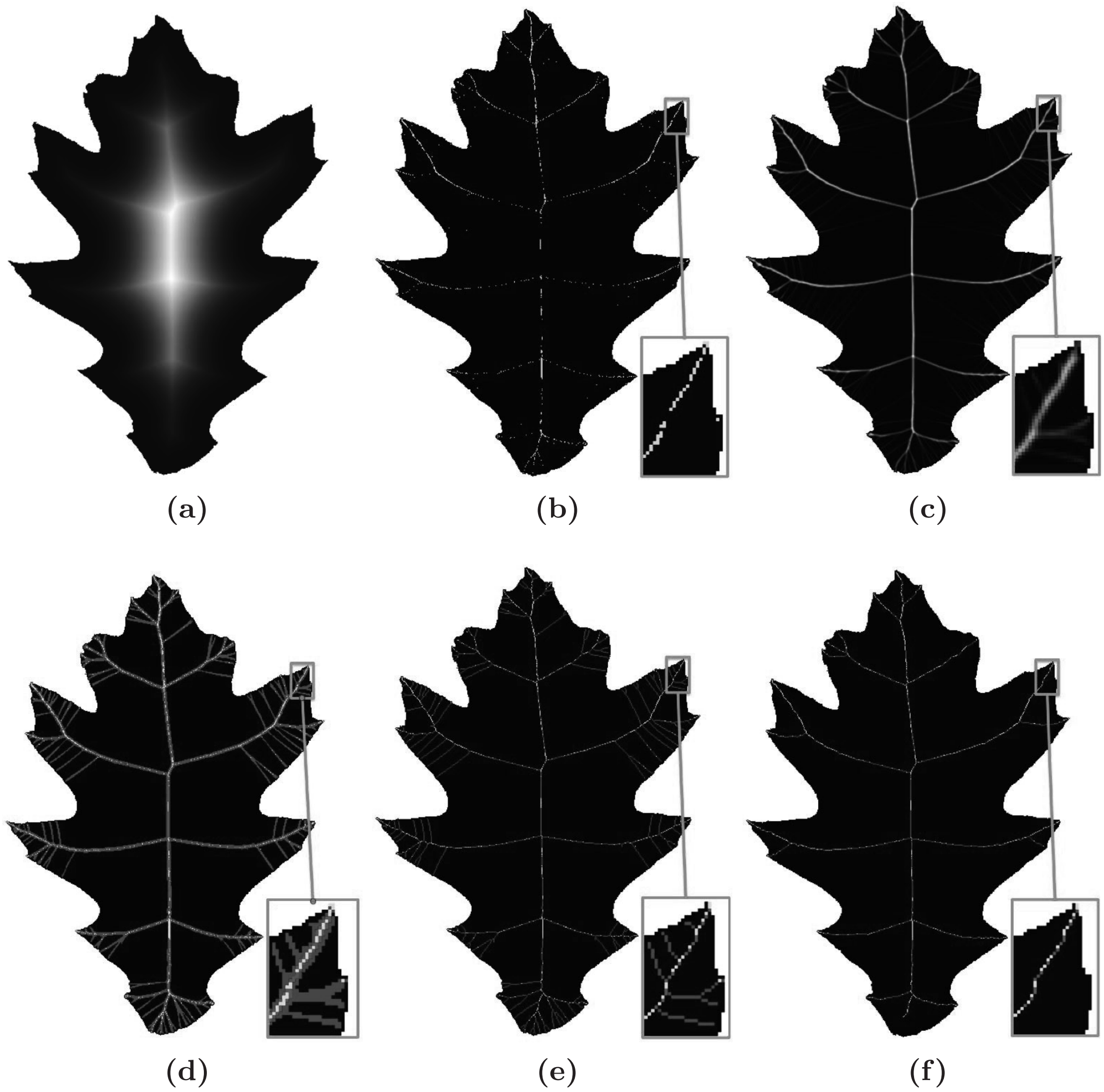


Fig. 3. a) SEDT map b) RDMA, c) Ridgeness map resulting from the convolution of the EDT with a negative LoG filter, d) Result obtained with the propagation over ridges, which generates a thick skeleton, e) Result of the MB2 thinning algorithm and f) Final skeleton of DECS method after pruning. The centers of maximal balls appear in yellow, values of ridgeness map greater than or equal to $th_{ridge-high}$ are visible in red and values of ridgeness map between $th_{ridge-low}$ and $th_{ridge-high}$ in blue.

changing the topology. An example of result is shown in figure 3e. At the end of the current step, the skeleton is thin and connected but may have a high degree of branching. The next step, which is optional, is used to remove spurious branches.

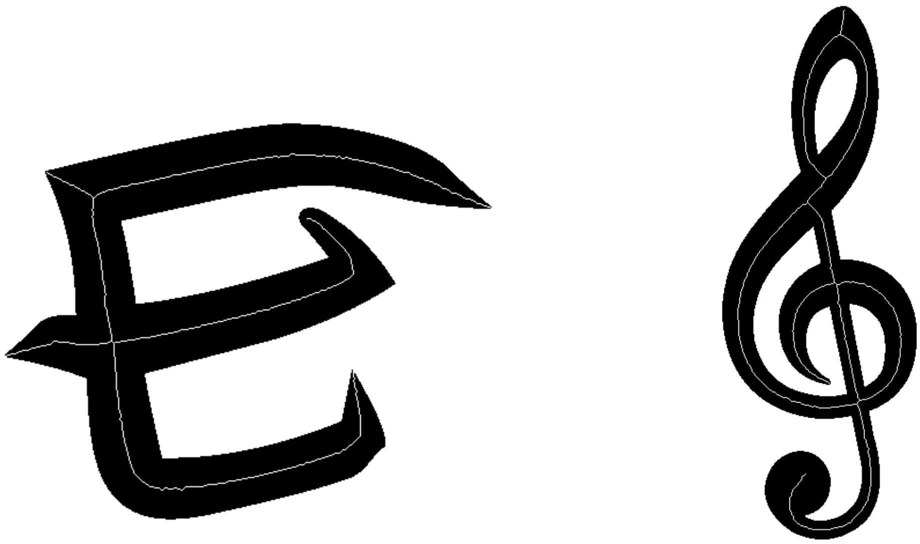
Algorithm 1. Pseudocode of propagation algorithm over ridges

```
Input: RDMA, rdg, th_{ridge-high}, th_{ridge-low}
Output: h : S ⟶ {NONE, MAX_BALL, STRONG_RIDGE, RIDGE}
Variables: p ,q , max_SEDT: points ∈ ℤ²; st: a stack of points;
visited: a set of points

for all p ∈ I do
  h(p):=NONE
end for
visited:=∅
max_SEDT:= argmax_{p∈RDMA} SEDTmap(p)
add(st,max_SEDT)
visited:=visited ∪ {max_SEDT}
while notEmpty(st) do
  p:=popTopElement(st)
  for all q ∈ N_8(p) such that q ∉ visited do
    if q ∈ RDMA then
      h(q):=MAX_BALL; add(st,q)
    else
      if rdg(q)⩾ th_{ridge-high} then
        h(q):=STRONG_RIDGE; add(st,q)
      else
        if th_{ridge-low} ⩽ rdg(q)< th_{ridge-high} then
          h(q):=RIDGE; add(st,q)
        end if
      end if
    end if
    visited:=visited ∪ {q}
  end for
end while
```

The pruning algorithm scans each end branch as long as the skeleton is not stable (*i.e.* until there are no more branches to remove). A branch is removed if all its points have a ridgeness less than $th_{ridge-high}$or if the percentage of maximal balls in this branch is less than $th_{perc-max-ball}$. Thresholds could be learnt on a training dataset in a shape matching process. During this pruning step, each skeleton point is visited once, so, the complexity is linearly proportional to $| S_k |$.

Finally, shape $\mathcal{S}$ has a skeleton made up of all the points of H that are labeled MAX_BALL, $STRONG\_RIDGE$ or $RIDGE$. An example of result is shown in Figure 3f. Throughout this section, we have seen that the total complexity of the proposed method is linearly proportional to n. We can observe, in Figure 4, two examples of skeleton obtained with the DECS method.



(a) $th_{perc-max-ball}$=0,5; $th_{ridge-high}$=1,1 **(b)** $th_{perc-max-ball}$=0,35; $th_{ridge-high}$=1

Fig. 4. Examples of skeletons obtained with DECS method, $th_{ridge-low}$ is always equal to 0.05

3 Results and Comparisons

We chose to compare our method (DECS) against three existing methods, which have equivalent complexities: K3M [2] is a recent thinning method, whereas Choi *et al*'s method [3] and Hamilton-Jacobi Skeleton [5] are two methods based on distance maps, like the proposed method. In this section, we test various properties such as connectivity and noise tolerance. We also make a comparison of their complexity. All tests were made on a database of shapes created by Latecki and Lakamper [14] with 1071 shapes. For tests, the standard deviation σ of LoG filter has been set to one. Because the one value allows to consider sufficient neighboring while maintaining accuracy on the ridges.

3.1 Connectivity

K3M, Hamilton-Jacobi skeleton and DECS methods build a connected skeleton in any situation. However, in Choi *et al*'s method, when the threshold becomes high, connectivity is not guaranteed. Ideal pruning is expected to keep the overall shape while removing unsignificant branches. With Choi *et al*'s method, when pruning is performed, disconnections appear along the skeleton, which corrupts information about the general shape. As regards this criterion, this last method is not interesting because it does not keep the shape topology. Notice that this criterion is dominant for us as our goal is to compute a graph based on the skeleton. This disconnection is highlighted in Figure 5.

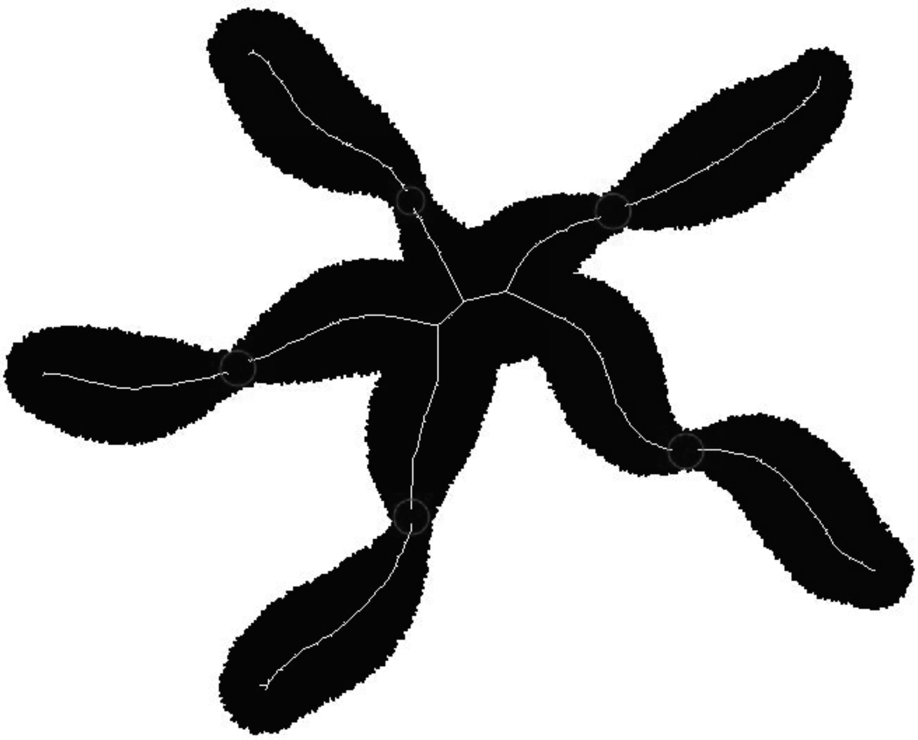
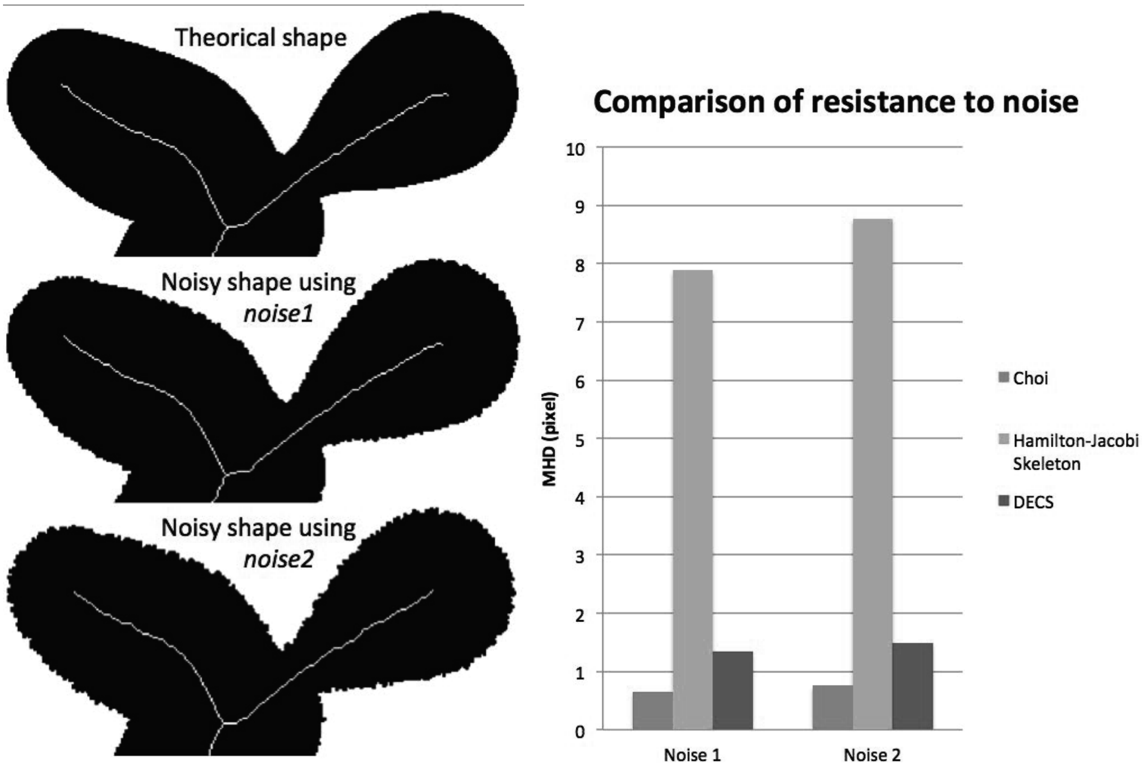


Fig. 5. Highlighting of skeleton disconnection obtained with Choi *et al*'s method ($\rho = 850$)

3.2 Noise Tolerance

The advantage of using a method that needs thresholds is that the tolerance to noise is adjustable. However, the problem is to make distinction, in real situation, between important information related to the actual shape boundary, and the noise. It's for this reason we created theorical situations where we exactly knew where the noise is.

To test the noise tolerance, we created manually 15 theoretical skeletons. Using either linear, sinusoidal or logarithmic functions, we generated the radius of maximal ball along branches. Then, we reconstructed shapes using the value of radius of maximal balls in order to obtain a "theoretical shape". In $\mathbb{Z}^2$, noise is added by randomly moving each contour pixel of k pixels along its normal vector. For our tests, we used $noise_1$ wherein $k \in [-1; 1]$ and $noise_2$ wherein $k \in [-2.5; 2.5]$. For each method, we selected a reference skeleton. For this, we varied thresholds and we retained the skeleton with the smallest Modified Hausdorff Distance (MHD), which allows to compute the Euclidean average error, compared to the theoretical skeleton. Concerning skeletons obtained from noisy shapes by $noise_1$ and $noise_2$ (*cf* Figure 6a), for each figure, for each method, for each noise level, we varied thresholds and we have retained the result giving the smallest MHD associated. The result of this experiment is presented in Figure 6b. This graph allows to bring out that the Hamilton Jacobi Skeleton is not noise resistant compared to other methods. This is its main drawback. Moreover, it allows to note that Choi *et al*'s method and DECS are very close in spite of a slight advantage nearly imperceptible to the naked eye on images for



(a) (b)

Fig. 6. a) Exemple of noisy shapes with skeleton associated, b) Comparison in terms of resistance to noise

Choi *et al*'s method. Nevertheless, the major problem with Choi *et al*'s method is the loss of connectivity of the skeleton when the threshold increases, *i.e.* when unnecessary branches are not taken into account (which is a critical aspect for us). Consequently, Choi *et al*'s method is not usable in our case. K3M was not used in the comparison because this method does not use a threshold, therefore it can not be adapted to be resistant to noise.

4 Conclusion and Future Work

In this paper, we presented a linear algorithm to extract a Discrete Euclidean Connected Skeleton of a shape. To do this, we proposed a propagation algorithm over ridges of the Euclidean distance map and centers of maximal balls, which ensures connectivity. Then we obtained a thin skeleton through the MB2 thinning algorithm, which ensures the thinness of the skeleton. The final step is to reduce the skeleton by pruning branches, based on a criterion using simultaneously ridgeness values and the centers of maximal balls.

By construction, the proposed skeleton always has desirable properties like full-connectivity, thinness and robustness to noise on the shape boundary in order to be used for graph matching. In the literature, no skeleton had all these properties as we mentioned in section 3. Observe that the proposed skeleton is also computed in linear time.

One should notice that the propagation algorithm used to generate a connected skeleton is only usable on connected shapes. For multiple-component shapes, this could be improved by detecting the connected components present in the image and applying DECS on each related component. Our future work is to create a graph from the obtained skeleton. Then, we will use this data structure in order to make shape matching.

References

1. Lam, L., Lee, S.W., Suen, C.Y.: Thinning methodologies - a comprehensive survey. IEEE Trans. Pattern Anal. Mach. Intell. 14, 869–885 (1992)
2. Saeed, K., Tabedzki, M., Rybnik, M., Adamski, M.: K3m: A universal algorithm for image skeletonization and a review of thinning techniques. Applied Mathematics and Computer Science 20, 317–335 (2010)
3. 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)
4. Latecki, L.J., Li, Q., Bai, X., Liu, W.: Skeletonization using ssm of the distance transform. In: International Conference on Image Processing, vol. (5), pp. 349–352 (2007)
5. Siddiqi, K., Bouix, S., Tannenbaum, A., Zucker, S.W.: Hamilton-jacobi skeletons. International Journal of Computer Vision 48, 215–231 (2002)
6. Coeurjolly, D., Montanvert, A.: Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension. IEEE Trans. Pattern Anal. Mach. Intell. 29, 437–448 (2007)
7. Remy, E., Thiel, E.: Exact medial axis with euclidean distance. Image and Vision Computing 23, 167–175 (2005)
8. Dimitrov, P., Phillips, C., Siddiqi, K.: Robust and efficient skeletal graphs. In: Computer Vision and Pattern Recognition, pp. 1417–1423 (2000)
9. Liu, H., Wu, Z., Zhang, X., Hsu, D.F.: A skeleton pruning algorithm based on information fusion. Pattern Recognition Letters 34, 1138–1145 (2013)
10. Bai, X., Latecki, L.J., Liu, W.: Skeleton pruning by contour partitioning with discrete curve evolution. IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007)
11. Meijster, A., Roerdink, J., Hesselink, W.: A general algorithm for computing distance transforms in linear time. Math. Morphology and Its Applications to Image and Signal Processing, 331–340 (2000)
12. Manzanera, A., Bernard, T.M., Prêteux, F., Longuet, B.: Ultra-fast skeleton based on isotropic fully parallel algorithm. In: DGCI (1999)
13. Bernard, T.M.: Improved low complexity fully parallel thinning algorithm. In: International Conference on Image Analysis and Processing, pp. 215–220 (1999)
14. Latecki, L.J., Lakamper, R., Eckhardt, U.: Shape descriptors for non-rigid shapes with a single closed contour. In: CVPR, pp. 1424–1429 (2000)