

Noise-resistant Digital Euclidean Connected Skeleton for graph-based shape matching [☆]



Aurélie Leborgne ^{a,b,*}, Julien Mille ^{a,c}, Laure Tougne ^{a,d}

^a Université de Lyon, CNRS, France

^b INSA-Lyon, LIRIS, UMR5205, F-69621, France

^c Université Lyon 1, LIRIS, UMR5205, F-69622, France

^d Université Lyon 2, LIRIS, UMR5205, F-69676, France

ARTICLE INFO

Article history:

Received 2 July 2014

Accepted 7 June 2015

Available online 24 June 2015

Keywords:

Skeletonization

Squared Euclidean distance map

Maximal ball

Medial axis

Ridge point detection

Digital shape representation

Pruning

Linear complexity

Shape reconstruction

ABSTRACT

The skeleton is an essential shape descriptor providing a compact representation of a shape that can be used in the context of real object recognition. However, due to the discretization, the required properties to use it for graph matching (homotopy to the shape, consequently connectivity, thinness, robustness to noise) may be difficult to obtain simultaneously. In this paper, we propose a new skeletonization algorithm having all these properties, based on the Euclidean distance map. More precisely, the algorithm cleverly combines the centers of maximal balls included in the shape and the ridges of the distance map. Post-processing is then applied to thin and prune the resulting skeleton. We compare the proposed method to three fairly recent methods and demonstrate its good properties.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Let us consider the recognition of 2D shapes, which could result from an image segmentation step. To deal with this recognition problem, one of the methods consists of extracting a set of features, referred to as signature, on the shape to be recognized (to be classified) and on the shapes of the database, or on representative shapes of the database, and comparing such signatures.

The goal here is to represent the shape with as little information as possible while keeping the overall appearance of the shape. In particular, the first properties that we expect for a skeleton is to maintain the topological properties of the initial shape and its geometric properties (ramifications and elongated parts for example).

To compare skeletons extracted from shapes, the idea is to convert skeletons into graphs (branches being edges and, junction

points and ending points being vertices) and then to perform graph matching. In fact, a graph is a representation more compact than the shape itself. Moreover, many effective graph matching methods have been proposed in the literature [1–3].

However, in order to easily convert the skeleton into a graph, it is necessary for this skeleton to have at least the following properties, which are not obviously obtained when the shape is represented by points in \mathbb{Z}^2 :

- it has to be connected: if the skeleton is not connected, the graph obtained from this skeleton will not be connected. Consequently, the graph and the shape will not have the same topology;
- it has to be thin (1-pixel width): a thick skeleton generates path extraction problems.

Moreover, to obtain effective and pertinent matchings in the context of real objects, it is necessary to construct skeletons robust to noise. Note 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 [4].

Let us consider now skeletonization algorithms. We can classify them as follows:

[☆] This paper has been recommended for acceptance by Yehoshua Zeevi.

* Corresponding author at: LIRIS, Université Lumière Lyon 2, 5 Avenue Pierre Mendès France, 69676 Bron Cedex, France.

E-mail addresses: aurelie.leborgne@liris.cnrs.fr (A. Leborgne), julien.mille@liris.cnrs.fr (J. Mille), laure.tougne@liris.cnrs.fr (L. Tougne).

- skeletonization methods based on thinning [5–9]
In an intuitive manner, it consists of “peeling” the shape for the purpose of obtaining a set of connected points with a single pixel width, which preserves the topology of the shape. In other words, thinning is an operation that aims to remove non-terminal simple points in a parallel or sequential manner. The main advantage of these algorithms is the preservation of the shape topology [5,6].
- skeletonization methods based on a distance map [10–15]
The objective is to identify the key points on the distance map, where each pixel is labeled with the value of its distance to the nearest background pixel. Different distance maps approximate or compute exactly the Euclidean distance:
 - Chamfer (approximation of the Euclidean distance by local mask) [16];
 - squared Euclidean distance [17];
 - signed Euclidean distance [18];
 - honeycomb (based on hexagonal grid) [19].

The next step is to search for the medial axis defined as the set of centers of maximal balls contained in the shape. A maximal ball is a ball contained in the shape not entirely covered by another ball contained in the shape.

The extraction of the medial axis is a reversible operation if the information on distance of each point to the nearest background pixel is retained. Hence, the original shape can be obtained with the medial axis [13]. The main advantage is that the skeleton is centered and the reconstruction is possible. However, such algorithms do not guarantee connectivity.

As stated previously, object recognition requires a shape representation which is invariant to minor changes, but the main drawback of the skeleton is 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 skeleton points, for the purpose of removing unwanted boundary noise and discretization artefacts [20,12]. In this case the result is rather biased as the boundary smoothing changes the boundary location. Consequently, the skeleton position will be different from the one computed directly on the shape without smoothing. The difficulty here is differentiating between significant boundary information and noise.
- deleting unwanted branches: this is a post-process called pruning [21–23]. It is based on local or global salience measures. The difficulty here is removing “noisy branches” without removing any meaningful parts of the skeleton.

Even if it could belong to the category of skeletons based on distance map, the proposed algorithm, called Digital Euclidean Connected Skeleton (DECS), computes maximal balls but also exploits the distance map in a novel way to connect centers of maximal balls to each other. The main contribution is the propagation and fusion of centers of maximal balls taking into account the ridges of the distance map, which are obtained by filtering this map. The obtained skeleton is then connected, thin and robust to noise. This is brought out by experiments in Section 4.1.

Before describing the proposed method in detail, in Section 3, we detail in Section 2, three methods from the literature we consider as state of the art for their properties: parallel thinning based on critical kernels, namely Bertrand and Couprie’s method [5], extraction of the Euclidean skeleton based on a connectivity criterion, namely Choi’s method [10] and the Hamilton–Jacobi skeleton method [12]. These methods will be compared to our method in Section 4.

2. Methods used for comparison

We chose to compare our method (DECS) against three existing methods: Bertrand and Couprie’s method is a recent parallel thinning method, Choi et al.’s method and Hamilton–Jacobi Skeleton are two methods based on distance maps, like the proposed method.

2.1. Bertrand and Couprie’s method [5]

This is a recent parallel thinning method based on critical kernels. The main idea is to gradually thin the shape until stability. This algorithm is based on a general framework for the study of parallel thinning in the context of abstract complexes. The principal is to parallelly delete simple points, which are points that may be deleted without changing the topology of the shape. This definition is based on the collapse operation which is a classical tool in algebraic topology and which preserves the topology. It is based on the fact that, if a subset Y of X contains the critical kernel of X , then Y has the same topology as X . We can observe in Fig. 1, an example of skeleton obtained with Bertrand and Couprie’s method.

Note that although Bertrand and Couprie’s method has no parameter to tune, like most thinning algorithms, but experiments, in Section 4, will show its shortcomings in terms of resistance to noise.

2.2. Choi et al.’s method: Euclidean skeleton based on a connectivity criterion [10]

This method generates a connected Euclidean skeleton. This algorithm starts with the computation of the 8-connected Signed Sequential Euclidean Distance map (8SSED) [18].

The next step is the extraction of the skeleton based on a connectivity criterion using a threshold ρ . The complexity of this algorithm is linear with respect to the number of pixels in the image. As illustrated in Fig. 2, the degree of branching of the skeleton decreases as ρ increases. This raises the issue of finding the appropriate threshold value with respect to the desired application. This method is interesting because it has been used for graph matching [3]. Moreover, it is a skeletonization method based on a distance map, like DECS.

2.3. Siddiqi et al.’s method: Hamilton–Jacobi Skeleton [12]

Like the previous method, Siddiqi et al.’s algorithm [12] generates a Euclidean skeleton with a single pixel width. Their method relies on an initial continuous modeling of the Euclidean distance

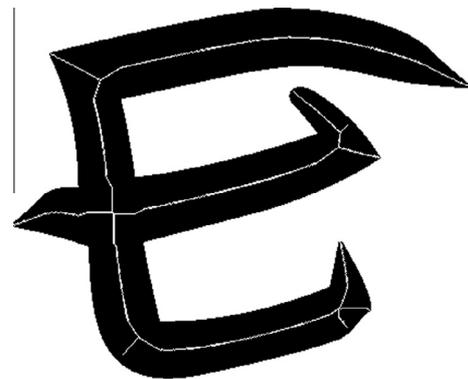
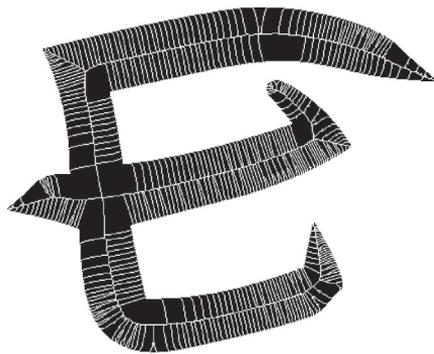
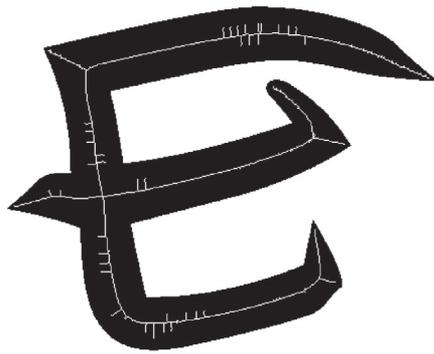


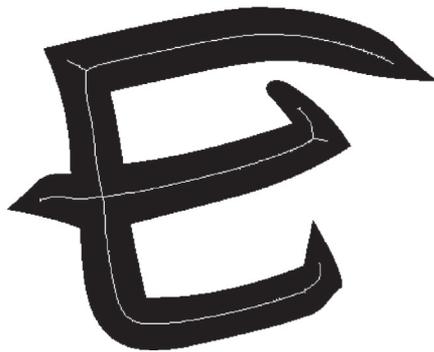
Fig. 1. Skeleton of a letter obtained with Bertrand and Couprie’s method.



(a) $\rho = 4$



(b) $\rho = 50$



(c) $\rho = 800$

Fig. 2. The skeletons obtained by Choi et al.'s method using different thresholds.

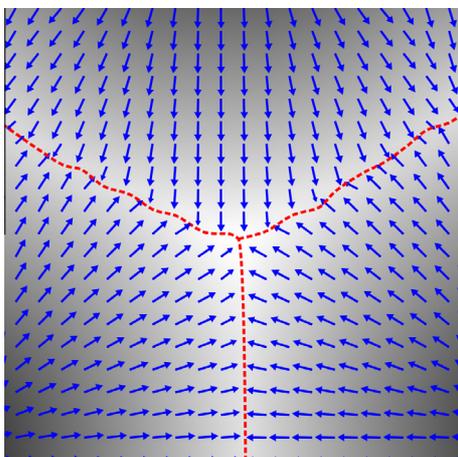
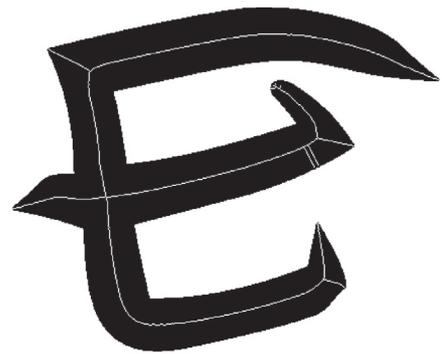


Fig. 3. Vector field generated by computing the gradient of the Euclidean distance transform. Vectors are directed towards the ridges of the distance map. Points of high inward flux are likely to be skeleton points (drawn in dashed red line).



(a) $th_{AOF} = -1$



(b) $th_{AOF} = -3$



(c) $th_{AOF} = -5$

Fig. 4. The skeletons obtained by Hamilton-Jacobi skeleton method using different thresholds.

to the input shape. It is based on the gradient of the Euclidean distance map. The obtained vector field is highly directed towards the ridges of the map, as shown in Fig. 3. The ridgeness of a point is related to the amount of vectors pointing towards this point.

This method shares some properties with the proposed method given that it is based on distance map. It also uses a threshold (th_{AOF}). In Fig. 4, it can be seen that small branches disappear as th_{AOF} decreases.

3. Extraction of Digital Euclidean Connected Skeleton (DECS)

3.1. Overview

Let us now describe our proposed method in detail.

Let us denote $\mathcal{I} \subset \mathbb{Z}^2$ an image of size $M \times N$ and $\mathcal{F} \subset \mathcal{I}$ a shape.

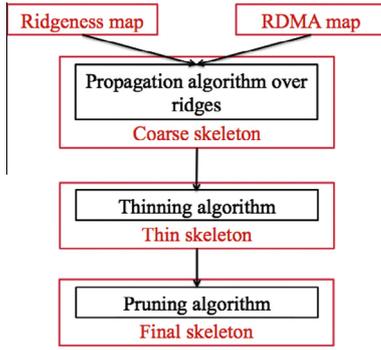


Fig. 5. Flowchart of the DECS algorithm.



Fig. 6. Squared Euclidean Distance map of a holly leaf.

Definition 1. The 8-connected neighborhood ($N_8(\mathbf{p})$) of \mathbf{p} with coordinates (x_p, y_p) is defined by:

$$N_8(\mathbf{p}) = \{\mathbf{q} \mid \max(|x_p - x_q|, |y_p - y_q|) = 1\}$$

Definition 2. A path is a sequence $(\mathbf{p}_0, \dots, \mathbf{p}_{u-1})$ of u pixels of \mathbb{Z}^2 such that for all $i = 0, \dots, (u-2)$, \mathbf{p}_{i+1} belongs to $N_8(\mathbf{p}_i)$.

Definition 3. Let us denote \mathcal{F} a set of pixels of \mathbb{Z}^2 . \mathcal{F} is a shape, if and only if, for all pair of pixels \mathbf{p} and \mathbf{q} belonging to \mathcal{F} , there exists a path linking \mathbf{p} and \mathbf{q} included in \mathcal{F} .

Thus, according to the definitions, our skeleton will be connected. In case there are several shapes in \mathcal{I} , one skeleton can be extracted per shape. Furthermore, note that \mathcal{F} can have holes or not.

As an aid to understanding, Fig. 5 summarizes the proposed method.

The first step consists in computing the squared Euclidean distance map. First, we find the centers of maximal balls and compute the Reduced Discrete Medial Axis (RDMA) map. The proposed method is based on the Squared Euclidean Distance Transformation namely SEDT (cf. Section 3.2) computed using [17]. The obtained map contains the set of centers of maximal balls (cf. Section 3.3) [13] that are points belonging to the skeleton, but that might not be connected. Note that such a set is also called medial axis. Then, we use a Laplacian of Gaussian filter (LoG filter) on the Euclidean distance map and compute a Ridgeness map (cf. Section 3.4). Next, the main idea is to combine these two maps in order 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 of thinning it (cf. Section 3.5) and retaining only the main connected branches (pruning).

3.2. Squared Euclidean Distance Transformation (SEDT)

The Euclidean distance can be estimated thanks to the Chamfer distance [24], or using local update [25]. The major problem with these methods is that they do not produce an exact distance. Take it out to obtain an exact distance, we chose to compute the Squared Euclidean Distance (SED). The technique we used [13,17] is based on a linear-time separable algorithm, which performs calculations on rows then on columns. We specifically chose this method because the SED of an image is calculated in two passes only in linear time. The goal of SEDT is to compute for each point (x_p, y_p) of \mathcal{F} , the following transform:

$$sed(x_p, y_p) = \min\{(x_p - x)^2 + (y_p - y)^2; 0 \leq x < M, 0 \leq y < N \text{ and } (x, y) \in \bar{\mathcal{F}}\}$$

where $\bar{\mathcal{F}}$ is the complement of \mathcal{F} .

An example is shown in Fig. 6. Note that the complexity of the SEDT algorithm is linear in n , the number of image pixels.

3.3. Reduced Discrete Medial Axis (RDMA) [13]

At each point of \mathcal{F} , the SEDT is the square of the radius of the largest ball centered at \mathbf{p} . As the skeleton contains centers of maximal balls, the main idea is to recover such points using the SEDT. For this, it is necessary to have an inclusion test, which decides whether a ball is covered by another one or not.

More formally, let us define the notion of maximal discrete ball.

Definition 4. Let $d^2 : \mathbb{Z}^2 \times \mathbb{Z}^2 \rightarrow \mathbb{N}$ be a discrete squared Euclidean distance.

A discrete ball, with center $\mathbf{c} \in \mathbb{Z}^2$ and a radius $r \in \mathbb{N}$ related to the distance d^2 is defined by:

$$B(\mathbf{c}, r) = \{\mathbf{q} \in \mathbb{Z}^2 \mid d^2(\mathbf{c}, \mathbf{q}) \leq r^2\}$$

A maximal discrete ball is a discrete ball contained in \mathcal{F} not entirely covered by another discrete ball contained in \mathcal{F} .

The set of centers of maximal discrete balls of \mathcal{F} is the Reduced Discrete Medial Axis (RDMA). To obtain the RDMA of \mathcal{F} , we used Coeurjolly and Montanvert's method [13], which is separable and in $O(n)$, where n is the number of pixels in the image. Indeed, RDMA is extracted in only four passes on the image. Moreover, this method uses exact values without approximation.

Fig. 7 is used to highlight the fact 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.

3.4. Laplacian-of-Gaussian filtering

We can note that the branches of the skeleton correspond to the ridges of the distance map. The Laplacian operator, which determines the local variations of second order, allows them to be extracted. When applied to the Euclidean Distance Transform (EDT), it is expected to be highly negative on positive ridges and close to zero on linear slopes. We may thus consider the negative Laplacian of the distance map as a measure of ridgeness. However, the Laplacian is very sensitive to noise when 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 $\sigma = 1$.

$$RDG(x, y) = -(EDT * LoG)(x, y)$$

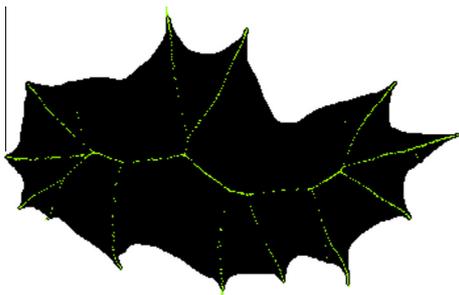


Fig. 7. RDMA of a holly leaf.

with

$$\text{LoG}(x,y) = -\frac{1}{\pi\sigma^4} \left(1 - \frac{x^2 + y^2}{2\sigma^2}\right) \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

An example is shown in Fig. 8. We can see that relevant branches are highlighted. The LoG filter is truncated beyond 3 times the standard deviation. As the obtained mask is separable, the complexity of the filtering operation is linear in σ .

However, a simple thresholding of LoG is insufficient for extracting the skeleton as the main branches that need to be kept can become disconnected, given that the values of the ridgeness map are not constant along branches. The proposed idea is to combine the RDMA and the ridgeness map RDG in order to determine the location of the relevant branches (where there are enough maximal balls and where the ridgeness map has sufficiently high values) and connect them using the ridgeness map that serves as a guide.

3.5. Combination of RDMA map and ridgeness map

In this subsection we present the combination of RDMA map and ridgeness map, which allows the skeleton to be built with good properties for graph matching. The summary of all the steps is shown in Fig. 5.

Let us first give the pseudocode of the propagation algorithm over ridges, which is the core of our proposal (Algorithm 1). This algorithm generates a labeling $h : \mathcal{F} \rightarrow \{\text{NONE}, \text{MAX_BALL}, \text{STRONG_RIDGE}, \text{RIDGE}\}$, where *MAX_BALL* is the label of the centers of maximal balls, *STRONG_RIDGE* is the label of the points having a value in the ridgeness map greater than or equal to $th_{\text{ridge-high}}$, *RIDGE* is the label of the points having a value in ridgeness map between $th_{\text{ridge-low}}$ and $th_{\text{ridge-high}}$, and *NONE* are the points not belonging to the main branches of the future skeleton. Parameters used are explained in Section 4. The algorithm uses a propagation technique starting from the center of the largest maximal ball. We consider this to be a relevant starting point, as it necessarily belongs to the skeleton.

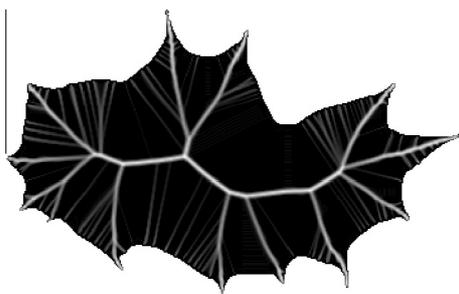


Fig. 8. Ridgeness map resulting from the convolution of the EDT with a negative LoG filter.

Algorithm 1. Pseudocode of propagation algorithm over ridges.

```

1: Input: RDMA, RDG,  $th_{\text{ridge-high}}$ ,  $th_{\text{ridge-low}}$ 
2: Output:
    $h : \mathcal{F} \rightarrow \{\text{NONE}, \text{MAX\_BALL}, \text{STRONG\_RIDGE}, \text{RIDGE}\}$ 
3: Variables:  $\mathbf{p}, \mathbf{q}$ ,  $max\_SED T \in \mathbb{Z}^2$ 
4:    $st$  a stack of points
5:    $visited$  a set of points.
6: for all  $\mathbf{p} \in \mathcal{I}$  do
7:    $h(\mathbf{p}) := \text{NONE}$ 
8: end for
9:  $max\_SED T := \underset{\mathbf{p} \in \text{RDMA}}{\text{argmax}} \text{sed}(\mathbf{p})$ 
10:  $h(max\_SED T) := \text{MAX\_BALL}$ 
11:  $\text{add}(st, max\_SED T)$ 
12:  $visited := \{max\_SED T\}$ 
13: while  $\text{notEmpty}(st)$  do
14:    $\mathbf{p} := \text{popTopElement}(st)$ 
15:   for all  $\mathbf{q} \in N_8(\mathbf{p})$  such that  $\mathbf{q} \notin visited$  do
16:     if  $\mathbf{q} \in \text{RDMA}$  then
17:        $h(\mathbf{q}) := \text{MAX\_BALL}$ 
18:        $\text{add}(st, \mathbf{q})$ ;
19:     else
20:       if  $\text{RDG}(\mathbf{q}) \geq th_{\text{ridge-high}}$  then
21:          $h(\mathbf{q}) := \text{STRONG\_RIDGE}$ 
22:          $\text{add}(st, \mathbf{q})$ ;
23:       else
24:         if  $th_{\text{ridge-low}} \leq \text{RDG}(\mathbf{q}) < th_{\text{ridge-high}}$  then
25:            $h(\mathbf{q}) := \text{RIDGE}$ 
26:            $\text{add}(st, \mathbf{q})$ ;
27:         end if
28:       end if
29:     end if
30:    $visited := visited \cup \{\mathbf{q}\}$ 
31: end for
32: end while

```

The propagation algorithm keeps only the points that are connected by a path of ridge points or centers of maximal balls. An example of result is shown in Fig. 9.

To help in understanding this algorithm, we detail it step by step. Rows 6–8 correspond to the initialization of all pixels with *NONE* label. In rows 9–12, first, we label *MAX_BALL* the center of maximal ball which has the largest radius as this is necessarily a skeleton point. Then, we add it to the stack in order to process its neighbors. Finally, we add it to the set of visited pixels. In rows 13–32, we extract the next point of the stack. We label with

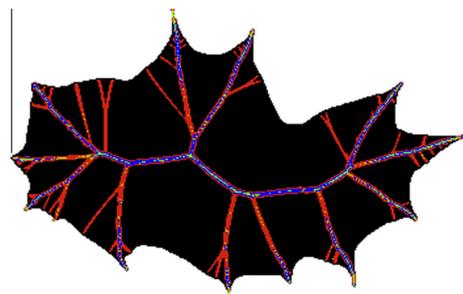


Fig. 9. An example of result obtained with the propagation algorithm. The centers of maximal balls appear in green, values of ridgeness map greater than or equal to $th_{\text{ridge-high}}$ are visible in blue and values of ridgeness map between $th_{\text{ridge-low}}$ and $th_{\text{ridge-high}}$ in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

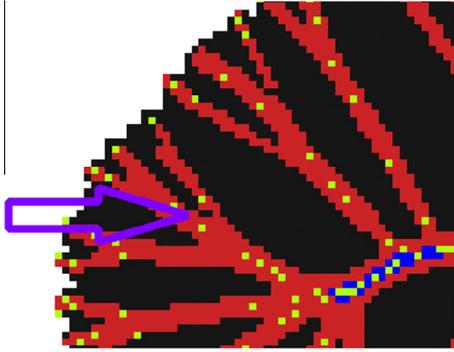


Fig. 10. An example of rare unwanted holes.

MAX_BALL , $STRONG_RIDGE$ or $RIDGE$ the non-visited neighbors of the current pixel, if they correspond to the associated criteria. Now, moreover, they belong to the set of visited pixels and they are added to the stack. These operations are repeated until the stack is empty.

At the end of this algorithm, the obtained temporary skeleton, denoted by S_{coarse} , is the set of points with a label different from $NONE$. It contains all important branches but has two drawbacks. The first one is that rare false holes may appear in the skeleton due to the thresholding and discretization of the ridgeness map. These holes could make the skeleton non-homotopic to the shape. An example is given in Fig. 10.

In order to solve this problem, if p belongs to \mathcal{F} but does not belong to S_{coarse} and which is surrounded by S_{coarse} then p is added to S_{coarse} . In other words, unwanted holes are filled by labeling them $RIDGE$. As a result of this step, shape homotopy is preserved. The second drawback is that the skeleton has a thickness that can be greater than one.

To thin the skeleton, we use the MB2 thinning algorithm [26,27]. This parallel algorithm allows a curve to be obtained with a thickness of one or two pixel(s) (regardless whether the thickness of the corresponding branch on the shape is even or odd). To obtain a curve with one-pixel thickness, post-processing is added. We delete all simple points, which are not endpoints. An endpoint is a point with only one neighbor in the skeleton. We chose this algorithm because its complexity is linear in n . Moreover, it is equivalent to a topology-preserving sequential thinning algorithm with the same deletion rule [6].

The result of this algorithm is a thin skeleton, denoted by S_{thin} , such that all the branches are one-pixel thick considering 8-connectivity. An example of result is shown in Fig. 11. At the end of the current step, the skeleton is thin and connected but it may have a high degree of branching. The next and last step, which is optional, is used to remove spurious branches. In fact, some branches are not necessary for graph matching and, on the contrary, they may disturb it.

We scan the shape and, as soon as one endpoint is detected, its branch is scanned until an intersection is met, while the number of

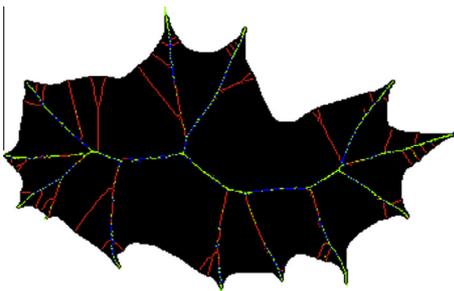


Fig. 11. An example of result of the modified MB2 Thinning algorithm.

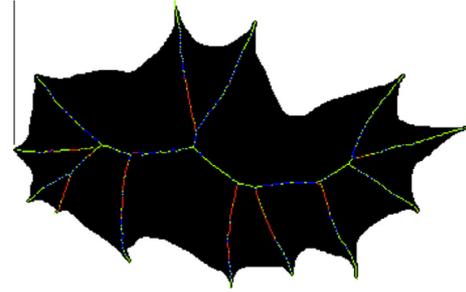


Fig. 12. An example of final skeleton of DECS method after pruning.

points in the branch is stored and the number of maximal balls and the location where the first point labeled $STRONG_RIDGE$ is met. Once at the intersection, the algorithm must take the decision to prune the branch or not. A branch is removed if all its points have a ridgeness less than $th_{ridge-high}$ and if the percentage of maximal balls in this branch is less than a threshold $th_{perc-max-ball}$. In this algorithm, each skeleton point is visited once, which yields a linear complexity.

Finally, the skeleton, denoted by S , is made up of all the points labeled MAX_BALL , $STRONG_RIDGE$ or $RIDGE$. An example of result is shown in Fig. 12. We can observe, in Fig. A.1 in Appendix A, twelve examples of skeletons obtained with the DECS method.

4. Results and comparisons

In this section, we test various properties such as connectivity, reconstruction and noise tolerance. Moreover, we compare our method according to such criteria with the three methods described in Section 2. We also make a comparison of their complexities. All tests are made on a database of shapes created by Latecki and Lakamper [28] with approximately a thousand shapes.

4.1. Noise tolerance

To test the noise tolerance, we created 15 theoretical skeletons: manually, we drew connected skeletons of one-pixel-thickness and using either linear, sinusoidal or logarithmic functions, we assigned a value representing the radius of maximal ball to each pixel (we took care to generate curves where the radius of curvature was high enough everywhere (no high concavities nor high convexities)). Then, we reconstructed shapes using radii of maximal balls to obtain theoretical shapes. Noise was added by randomly moving each pixel inward or outward along its normal vector. We made sure that the thickness of the shape was high enough so as not to break connectivity of the shape.

When all border pixels had been moved, we calculated an 8-connected curve from the list of moved pixels and then we recreated a shape by coloring pixels inside this closed curve. For our tests, we used $noise_1$ with $k = 1$ and $noise_2$ with $k = 3$, where k is the offset. 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) compared to the theoretical skeleton. For each noise, method and image we retained the smallest MHD compared to the reference skeleton. Fig. 13 provides an example of noisy shape skeletonization for each method with adapted thresholds (when these exist) giving the smallest MHD. We can note that Bertrand and Couprie's method is not resistant to noise. This is the reason why we have not quantified the noise tolerance for this method.

The results of this experiment are presented in Fig. 14. This graph shows that the Hamilton–Jacobi Skeleton is not noise

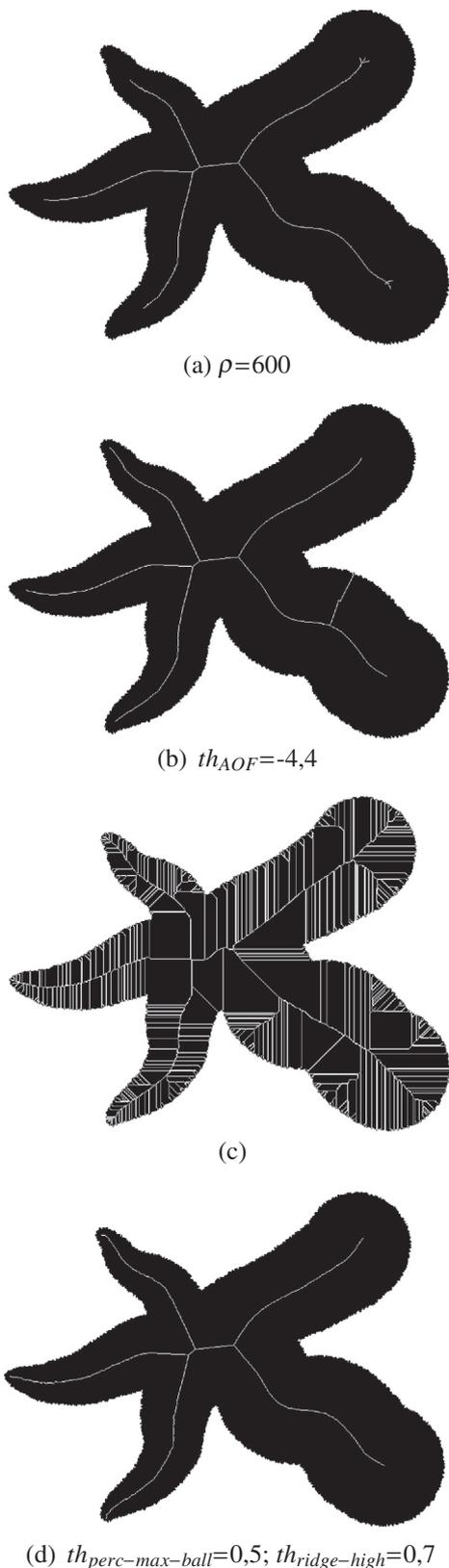


Fig. 13. Result of noisy shape skeletonization for each method with adapted thresholds (when there are) giving the smallest MHD, (a) Choi et al.'s method, (b) Hamilton Jacobi method, (c) Bertrand and Couprie's method and (d) DECS method.

resistant in contrast to the other methods. Moreover, it demonstrates that Choi et al.'s method and DECS are very close in spite of a slight advantage hardly imperceptible to the naked eye on

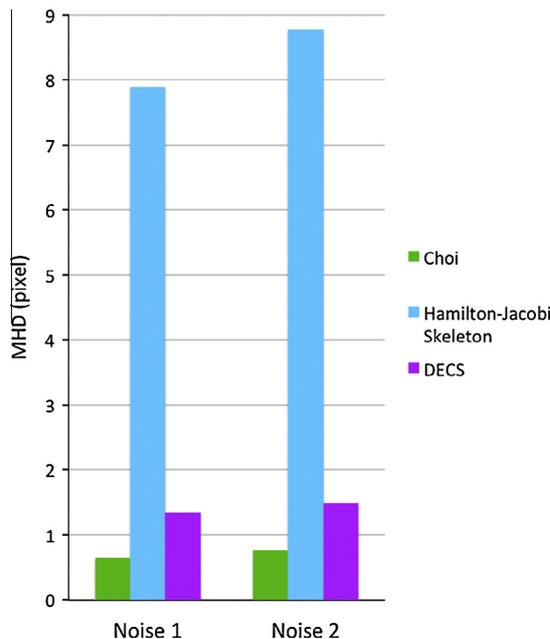


Fig. 14. Comparison of resistance to noise. For $i \in [1 \dots 15]$, for t as $th_{AOF} \in [-6 \dots -0, 1]$ for Hamilton Jacobi Skeleton, for t as $\rho \in [4 \dots 3000]$ for Choi et al.'s method and t is the combination of $th_{perc-max-ball} \in [0 \dots 1]$ and $th_{ridge-high} \in [0, 1 \dots 1, 1]$.

the images for Choi et al.'s method. Nevertheless, the major problem of Choi et al.'s method is the non-connectivity as explained in Section 4.2. Choi et al.'s method is therefore not usable in our case.

4.2. Influence of parameters

In Section 4.1, we present ground truth tests. Each method is tested in its best conditions: for each image, we chose threshold(s) that yielded the best results according to our evaluation criteria (defined in Section 4.1). For Choi et al.'s method, the amount of spurious branches decreases as ρ increases. The threshold which allows the best results to be obtained is $\rho = 803$. For the Hamilton–Jacobi skeleton method, the amount of spurious branches decreases as threshold th_{AOF} decreases. The best results for this method are obtained with average threshold equal to -3.06 . As regards the proposed method, deletion of spurious branches depends on the combination of two thresholds. These two thresholds vary in the same direction, that is to say the higher $th_{perc-max-ball}$ and $th_{ridge-high}$ are, the fewer the spurious branches there are.

$th_{ridge-low}$ is always set to 0.05 and $th_{ridge-high}$ varies between $th_{ridge-low}$ and 1.1. The typical thresholds for this method are $th_{perc-max-ball} = 0.4$ and $th_{ridge-high} = 0.5$. Nevertheless, Fig. 15 shows four DECS skeletons of a same shape with different thresholds in order to give an overview of the influence of the parameters.

4.3. Connectivity

Bertrand and Couprie's, Hamilton–Jacobi skeleton methods and DESC build a connected skeleton in any situation. However, in Choi et al.'s method, when the threshold becomes high, connectivity is not guaranteed. Some disconnections are highlighted in Fig. 16. On the example, the skeleton is disconnected in the middle part even if the threshold is relatively small. Disconnections also appear when the thickness of the shape reduces and the shape forms a “second order shock” [15]. Consequently, the topology of the shape is lost, which is a major drawback for shape matching.

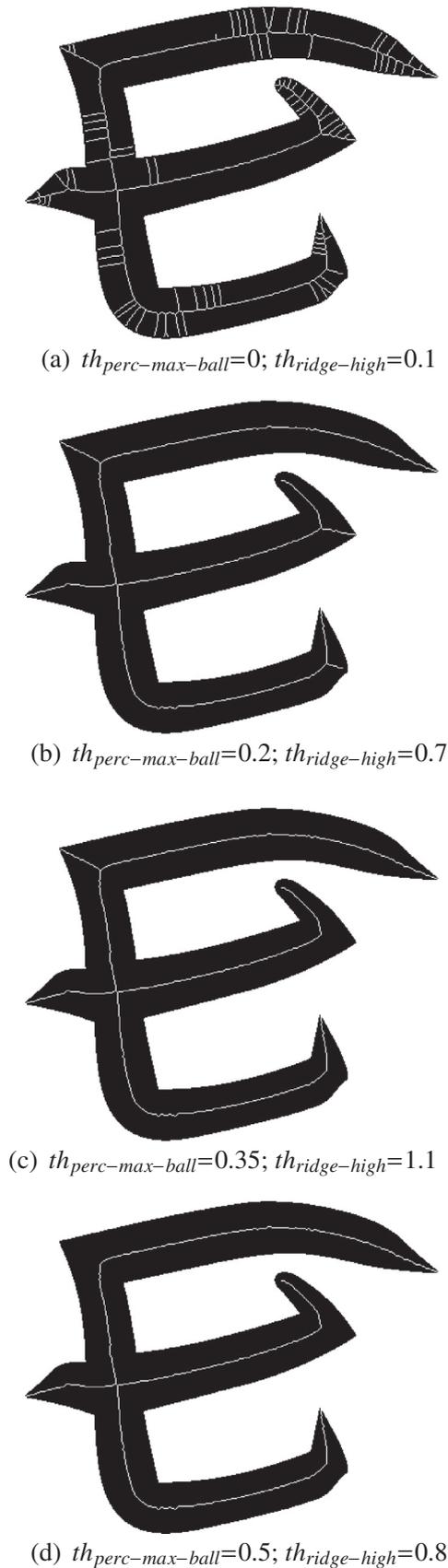


Fig. 15. Examples of skeletons obtained with DECS method using different thresholds. $th_{ridge-low}$ is always equal to 0.05.

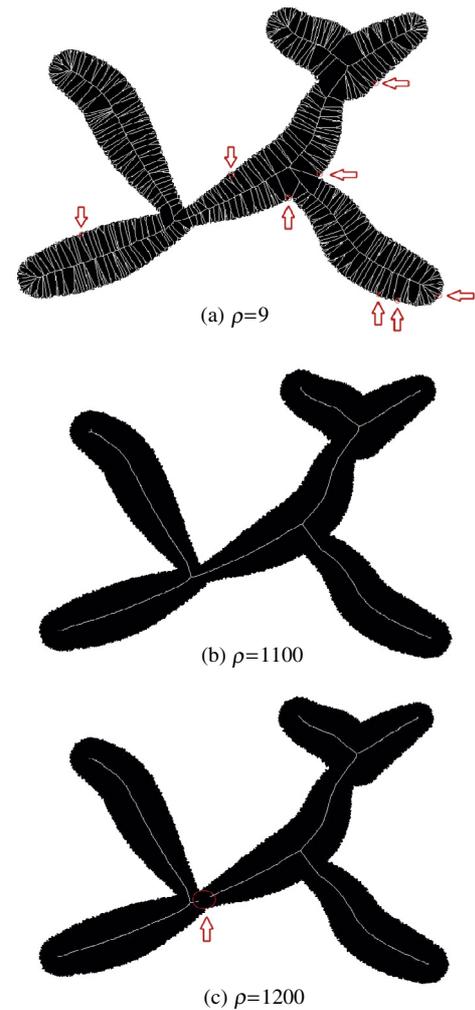


Fig. 16. Evolution of threshold value in order to highlight a disconnection on Choi et al.'s skeletons.

As regards this criterion, this last method is uninteresting because it does not keep the shape topology. Note that this criterion is dominant for us as our goal is to compute a graph based on the skeleton, which should be homotopic to the shape.

4.4. Complexity

In this subsection, let us recall that n is the number of pixels in the image. The proposed method has a linear complexity. An experimental study has been performed on all shapes of the Latecki and Lakamper's database and is shown in Fig. 17(a) to support our demonstration.

To prove this, we detail the complexity of each step. The computation of the *SED*T map is linear [13,17]. The computation of the ridgeness map is also linear as it is a convolution of the Euclidean distance map with *LoG* filter. So the complexity is $O(n\sigma)$. The extraction of centers of maximal balls is also linear as it consists of four linear passes on the image [13]. As regards the propagation algorithm, we select from the maximal ball with the largest radius skeleton points by, at most, scanning all the pixels of the shape. This step also has a linear complexity. The linearity of thinning and pruning is not simple to prove by reasoning, but it is shown experimentally through the graph in Fig. 17(b).

Choi et al.'s method has a linear complexity too. Moreover, we can assimilate the complexity of Bertrand and Couprie's method

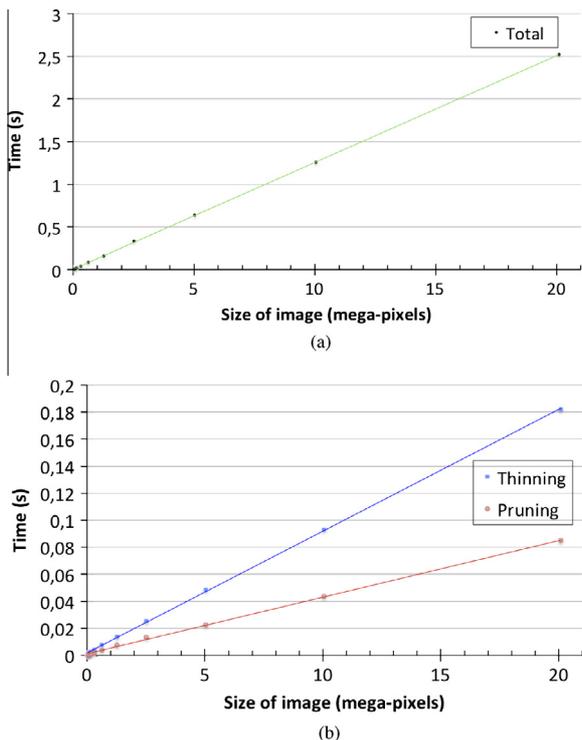


Fig. 17. By varying n , these graphs present the average of computation times of skeletons for each image of the Latecki and Lakamper’s database. To vary n , between two iterations, height and width are multiplied by $\sqrt{2}$. Consequently, n doubles at each iteration. a) Total calculation time, b) Pruning time (in red) and thinning time (in blue). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

to a linear complexity, even if the complexity is not explicitly given in the paper [5], as it is a parallel thinning algorithm. However, the Hamilton–Jacobi skeleton method has a complexity in $O(n \log n)$. Regarding this criterion, this last method is less interesting.

4.5. Reconstruction

One of the important properties of a skeleton is the reconstruction ability. It shows how the geometry of the shape is preserved. In fact, it guarantees that a method preserves the initial information *i.e.* does not suppress nor generate pixels. The need for an accurate reconstruction is questionable given that it depends on the application. In the event of graph matching, we do not need to reconstruct the shape exactly but we need to approximately obtain the initial shape. The advantage of the proposed method is that, depending on the fixed thresholds, we obtain various levels of reconstruction. In fact, a branch of the skeleton is removed if it does not contain any pixel with a value of ridgeness greater or equal to $th_{ridge-high}$ or if the percentage of maximal balls in this branch is less than $th_{perc-max-ball}$. Thus, the level of detail of the skeleton increases as $th_{perc-max-ball}$ and $th_{ridge-high}$ decrease. An example of the skeleton and the generated difference image are shown in Fig. 18.

It should be noted that, in order to perform reconstruction, the value of the *SEDIT* is kept for each skeleton point. Thus, we only have to draw a ball with the corresponding radius, centered at each skeleton point. We now introduce the three measures for assessing reconstruction.

Let P be the set of pixels of the original shape and Q the set of pixels of the reconstructed shape.

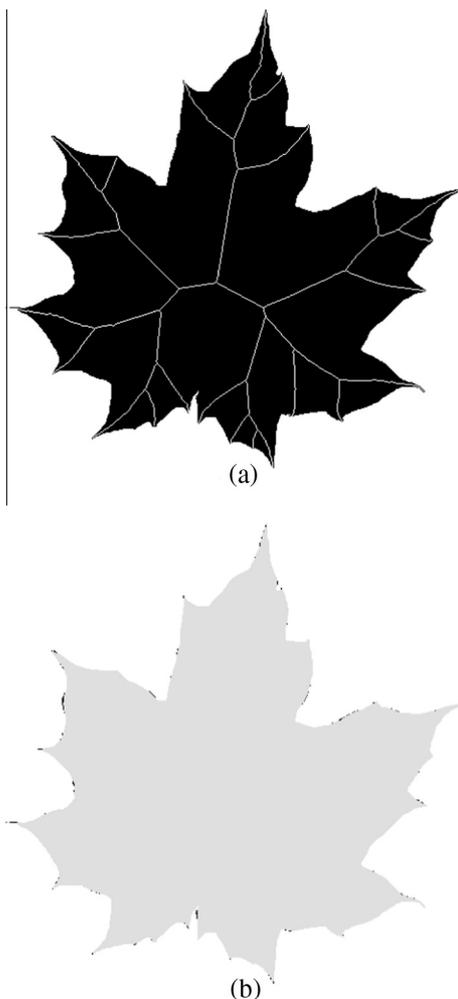


Fig. 18. (a) An example of skeleton extracted from a leaf shape with DECS algorithm, (b) the difference image obtained. In gray, the reconstructed image and in black, the pixels that belong to the original image but not to the reconstructed image.

Definition 5. The **percentage of non-reconstructed shape** is defined as:

$$Perc = \frac{|P \setminus Q|}{|P|} \times 100$$

Definition 6. Let us assume that P and Q are non empty subsets of \mathbb{Z}^2 . We define their **Hausdorff distance** $HD(P, Q)$ by:

$$HD(P, Q) = \max \left\{ \max_{p \in P} \min_{q \in Q} \{d(p, q)\}, \max_{q \in Q} \min_{p \in P} \{d(q, p)\} \right\},$$

where d is the Euclidean distance and, p and q are elements of \mathbb{Z}^2 . In other words, the Hausdorff distance is the largest minimal distance between a point of the shape and the result shape.

Definition 7. Let us assume that P and Q are non empty subsets of \mathbb{Z}^2 . We define their **modified Hausdorff distance** $MHD(P, Q)$ by:

$$MHD(P, Q) = \max \left\{ \frac{1}{|P|} \sum_{p \in P} \min_{q \in Q} \{d(p, q)\}, \frac{1}{|Q|} \sum_{q \in Q} \min_{p \in P} \{d(q, p)\} \right\},$$

Table 1
Quantification of the best reconstructions.

	Perc	HD	MHD
Hamilton–Jacobi skeleton	0.58216	1.20559	1.04608
Choi et al.	0.37558	1.19915	0.95196
Bertrand and Couprie	0.37334	1.17093	1.02878
DECS	0.52732	1.19121	0.99731

where d is the Euclidean distance and, \mathbf{p} and \mathbf{q} are elements of \mathbb{Z}^2 . In other words, the modified Hausdorff distance is the average distance between the original shape and the result shape.

In the remainder of this section, we compared the skeleton generated with DECS algorithm with the ones obtained from the methods of Section 2. In order for the comparisons to be as fair as possible, the skeletons obtained with the three existing methods were post-processed with [27] in order to have a one pixel thickness. The results of this first test can be found in Table 1, where

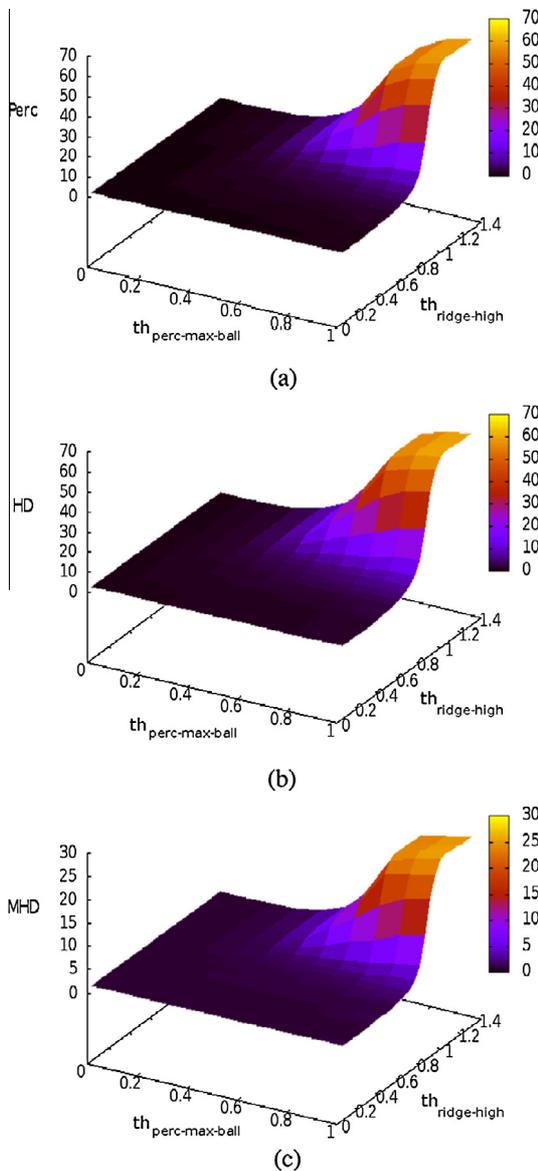


Fig. 19. Variation of Perc (a), HD (b) and MHD (c) measures according to the thresholds $th_{perc-max-ball}$ and $th_{ridge-high}$ of DECS.

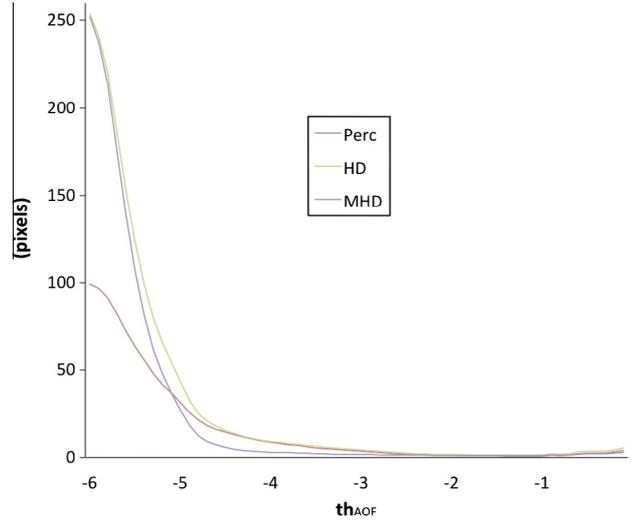


Fig. 20. Variation of Perc, HD and MHD measures according to the threshold th_{AOF} of Hamilton–Jacobi skeleton method.

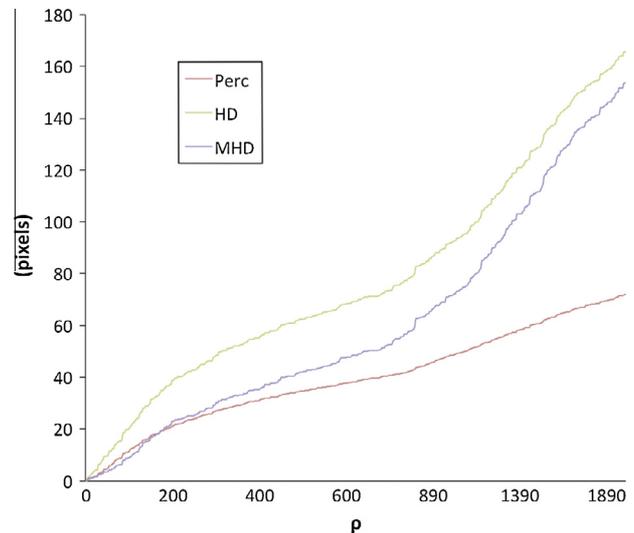


Fig. 21. Variation of Perc, HD and MHD measures according to the threshold ρ of Choi et al.'s method.

the values correspond to the average of measures for the 1071 shapes contained in the database. For each of the four methods, we selected the best reconstruction results by varying the thresholds.

We can already note that perfect reconstruction was impossible given that we had chosen to extract skeletons of one pixel thickness. In fact, when the thickness of the shape is even, in order to be able to reconstruct the original shape exactly, the skeleton should have a thickness of 2 pixels to remain centered. Otherwise, in an 600×400 image, this makes a difference of at most 1 pixel on the reconstructed contour, which is imperceptible during a quick visual inspection. As regards the best reconstruction, we can consider these four methods all comparable.

Figs. 19–21 show that changing thresholds affects the quality of the reconstruction as well as the quantity of detail. These parameters are responsible for the branching density of the skeleton. Decreasing the degree of branching improves the usability of the skeleton (in a matching process, for instance) but it decreases its ability to accurately reconstruct the shape. The objective is to find

a trade-off between the accuracy of reconstruction due to non-pruned skeleton and the usability of the skeleton in a subsequent processing. While a pruned skeleton may not perfectly rebuild the shape, it is often preferable for the purpose of shape matching. The first observation we can make is that the three measures vary in the same way depending on the variation of thresholds. However, the quality of reconstruction of Choi et al.'s skeleton deteriorates very quickly while the parameter of the Hamilton–Jacobi skeleton method can be varied within a large range of values until the results deteriorate.

4.6. Discussion

Let us recall that the skeleton criteria, which are the most critical for shape matching, are:

- Homotopy: the topology of the skeleton should match the topology of the shape. Branches are one-pixel thick. Discontinuities along the branches should be avoided.
- Low complexity: shape matching often involves analyzing many shapes in a minimal time period. A linear complexity is therefore highly desirable.
- Accurate reconstruction, *i.e.* have a skeleton that can rebuild the shape sufficiently close to the original shape.
- Noise tolerance, which eliminates unnecessary branches so as to avoid overloading the skeleton and to save time when matching.

Regarding robustness to noise, we see a clear superiority of distance-based algorithms. This superiority is not confirmed for other criteria since the main drawbacks of the Hamilton–Jacobi skeleton method are its complexity and especially its lack of noise resistance. 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).

5. Conclusion and future work

In this paper, we presented a linear algorithm for extracting a Digital 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. The propagation starts from the center of the maximal ball having the largest radius, thus ensuring connectivity. Then we obtain a thin skeleton through an extension of the MB2 *thinning* algorithm, which thus 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.

The proposed skeleton has desirable properties like connectivity, thinness and robustness to noise on the shape boundary for use in graph matching. As far as we are aware, no noise-robust skeletonization algorithm while preserving connectivity and homotopy to the shape has been previously reported in the literature. Observe that the proposed skeleton is also computed in linear time and it allows the shape to be accurately reconstructed.

One should note 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 will be to create a graph from the obtained skeleton. Then, we will use this data structure to make shape matching.

Appendix A. Examples of skeletons

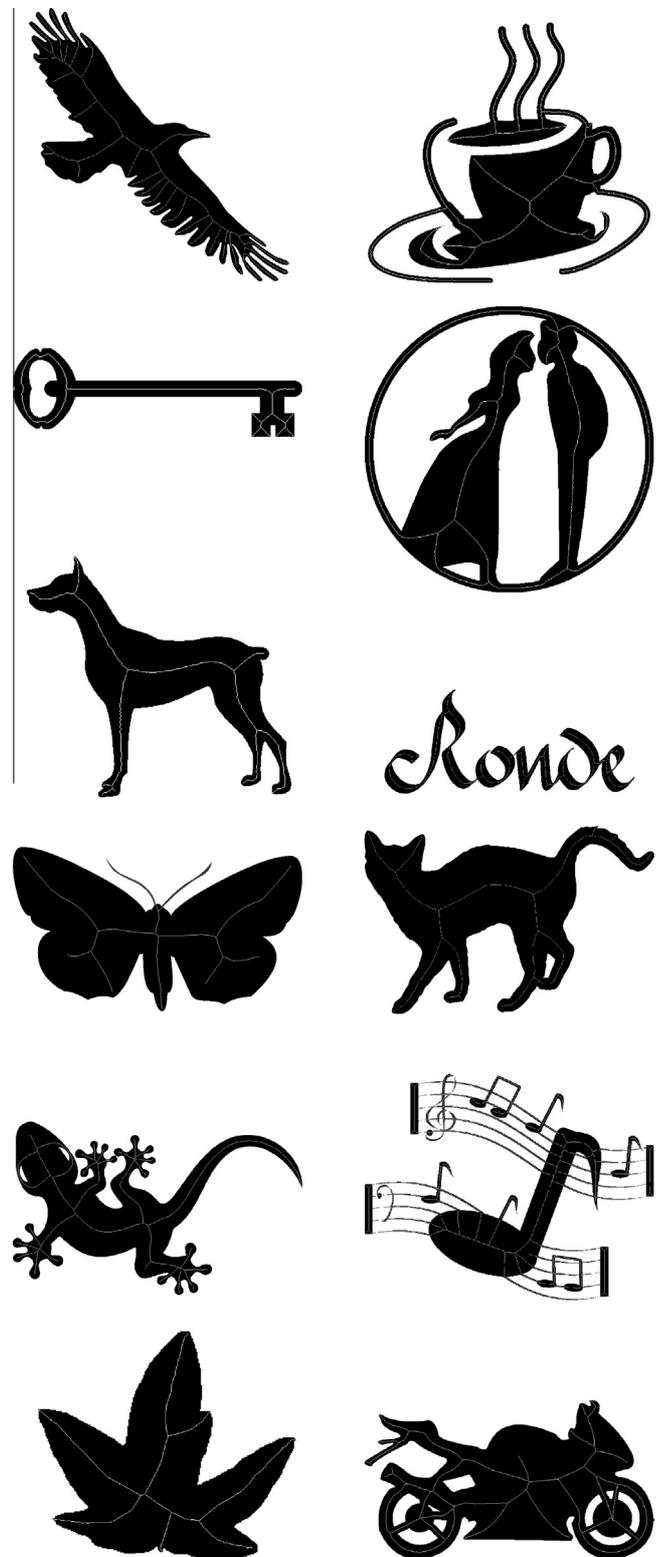


Fig. A.1. Examples of skeletons obtained with DECS method using the same thresholds: $th_{perc-max-ball} = 0,4$; $th_{ridge-high} = 0,5$ and $th_{ridge-low} = 0,05$.

References

- [1] T.B. Sebastian, P.N. Klein, B.B. Kimia, Recognition of shapes by editing their shock graphs, *IEEE Trans. Pattern Anal. Mach. Intell.* 26 (5) (2004) 550–571.
- [2] D. Macrini, S.J. Dickinson, D.J. Fleet, K. Siddiqi, Object categorization using bone graphs, *Comput. Vis. Image Underst.* 115 (8) (2011) 1187–1206.
- [3] W. Shen, Y. Wang, X. Bai, H. Wang, L.J. Latecki, Shape clustering: common structure discovery, *Pattern Recogn.* 46 (2) (2013) 539–550.
- [4] M. Leyton, Symmetry–curvature duality, *Comput. Vis. Graph. Image Process.* 38 (3) (1987) 327–341.
- [5] G. Bertrand, M. Couprie, Powerful parallel and symmetric 3d thinning schemes based on critical kernels, *J. Math. Imag. Vis.* 48 (1) (2014) 134–148.
- [6] K. Palágyi, Equivalent 2d sequential and parallel thinning algorithms, in: *International Workshop on Combinatorial Image Analysis*, Springer, 2014, pp. 91–100.
- [7] L. Lam, S.-W. Lee, C.Y. Suen, Thinning methodologies – a comprehensive survey, *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (9) (1992) 869–885.
- [8] K. Saeed, M. Tabedzki, M. Rybnik, M. Adamski, K3m: a universal algorithm for image skeletonization and a review of thinning techniques, *Appl. Math. Comput. Sci.* 20 (2) (2010) 317–335.
- [9] L. Liu, E.W. Chambers, D. Letscher, T. Ju, A simple and robust thinning algorithm on cell complexes, *Comput. Graph. Forum* 29 (7) (2010) 2253–2260.
- [10] W.-P. Choi, K.-M. Lam, W.-C. Siu, Extraction of the euclidean skeleton based on a connectivity criterion, *Pattern Recogn.* 36 (3) (2003) 721–729.
- [11] L.J. Latecki, Q. Li, X. Bai, W. Liu, Skeletonization using ssm of the distance transform, in: *International Conference on Image Processing (ICIP)*, IEEE, 2007, pp. 349–352.
- [12] K. Siddiqi, S. Bouix, A. Tannenbaum, S.W. Zucker, Hamilton–Jacobi skeletons, *Int. J. Comput. Vis.* 48 (3) (2002) 215–231.
- [13] D. Coeurjolly, A. Montanvert, Optimal separable algorithms to compute the reverse euclidean distance transformation and discrete medial axis in arbitrary dimension, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (3) (2007) 437–448.
- [14] G.S. di Baja, E. Thiel, Skeletonization algorithm running on path-based distance maps, *Image Vis. Comput.* 14 (1) (1996) 47–57.
- [15] K. Siddiqi, A. Shokoufandeh, S.J. Dickinson, S.W. Zucker, Shock graphs and shape matching, *Int. J. Comput. Vis.* 35 (1) (1999) 13–32.
- [16] G. Borgefors, Distance transformations in digital images, *Comput. Vis. Graph. Image Process.* 34 (3) (1986) 344–371.
- [17] A. Meijster, J. Roerdink, W. Hesselink, A general algorithm for computing distance transforms in linear time, *Math. Morphol. Appl. Image Signal Process.* (2000) 331–340.
- [18] Q.-Z. Ye, The signed euclidean distance transform and its applications, in: *Proceedings of the Ninth International Conference on Pattern Recognition*, 1988, pp. 495–499.
- [19] G. Borgefors, S. di baja Gabriella, Skeletonizing the distance transform on the hexagonal grid, in: *Proceedings of the Ninth International Conference on Pattern Recognition*, vol. 1, 1988, pp. 504–507.
- [20] P. Dimitrov, C. Phillips, K. Siddiqi, Robust and efficient skeletal graphs, in: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2000, pp. 1417–1423.
- [21] H. Liu, Z. Wu, X. Zhang, D.F. Hsu, A skeleton pruning algorithm based on information fusion, *Pattern Recogn. Lett.* 34 (10) (2013) 1138–1145.
- [22] X. Bai, L.J. Latecki, W. Liu, Skeleton pruning by contour partitioning with discrete curve evolution, *IEEE Trans. Pattern Anal. Mach. Intell.* 29 (3) (2007) 449–462.
- [23] W. Shen, X. Bai, R. Hu, H. Wang, L.J. Latecki, Skeleton growing and pruning with bending potential ratio, *Pattern Recogn.* 44 (2) (2011) 196–209.
- [24] P.-E. Danielsson, Euclidean distance mapping, *Comput. Vis. Graph. Image Process.* 14 (1980) 227–248.
- [25] I. Ragnemalm, The euclidean distance transform in arbitrary dimensions, *Pattern Recogn. Lett.* 14 (11) (1993) 883–888.
- [26] A. Manzanera, T.M. Bernard, F.J. Prêteux, B. Longuet, Ultra-fast skeleton based on an isotropic fully parallel algorithm, in: *Discrete Geometry for Computer Imagery*, Springer, 1999, pp. 313–324.
- [27] T.M. Bernard, A. Manzanera, Improved low complexity fully parallel thinning algorithm, in: *10th International Conference on Image Analysis and Processing*, IEEE Computer Society, 1999, pp. 215–220.
- [28] L.J. Latecki, R. Lakamper, U. Eckhardt, Shape descriptors for non-rigid shapes with a single closed contour, in: *Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Computer Society, 2000, pp. 1424–1429.