HIERARCHICAL SKELETON FOR SHAPE MATCHING

Aurélie Leborgne[◇ ∮] *Julien Mille*[†] *Laure Tougne*[⋆]

[◇] Univ Lyon, INSA-Lyon , LIRIS, UMR5205, F-69621, LYON, France
[†] INSA-Centre Val de Loire, LI, EA 6300, F-37200, France
[⋆] Univ Lyon, Université Lyon 2, LIRIS, UMR5205, F-69676, LYON, France
[∮] Université d'Auvergne, ISIT, UMR6284, F-63000, France

ABSTRACT

The skeleton is an efficient and complete shape descriptor often used for matching. However, existing skeleton-based shape matching methods are computationally intensive. To reduce the algorithmic complexity, we propose to exploit the natural hierarchy of the skeleton. The aim is to quantify the importance of skeleton branches to guide the shape matching algorithm, in order to match branches having the same order of importance. Our method is based on successive shape smoothing operations and on the deformability of the skeleton to adapt it to each smoothed shape. Moreover, we show that our method is independent from the initial skeleton.

***Index Terms*—** hierarchical skeleton, matching, pruning, pattern recognition

1. INTRODUCTION

Many shape matching methods have been proposed [1]. Matching using skeleton is known to be efficient but has a high computational cost compared to methods based on the boundary curve [2]. In this paper, we use the skeleton to recognize plant species corresponding to binary images obtained from photographs of plant leaves. Consider a typical database where leaves are classified by species. We try to match an unknown leaf with all leaves of the database, in order to determine the species closest to the tested leaf.

A way to reduce the computational time is to use the hierarchy between skeleton branches, to compare branches having the same order of importance. Thus, it is necessary to quantify the importance or significance of each skeleton branch. In the literature, Attali *et al.* measured branches significance by means of their length, elongation or area [3]. More recently, Liu et al. [4, 5] used measures based on the contributions of branches to reconstruction. Shen *et al.* [6] developed a measure denoted bending potential ratio based on the significance of contour segments in terms of contextual information. Other authors used more visual contributions like Bai *et al.* [7], who pre-process the shape boundary using a discrete curve evolution to determine if each branch is significant or not. In the same way, Montero *et al.* [8] proposed a method based on contour approximation and the integer medial axis. One of the most recent methods [9] is based on the distance between an end point and a branch point, and there respective distance to the shape border.

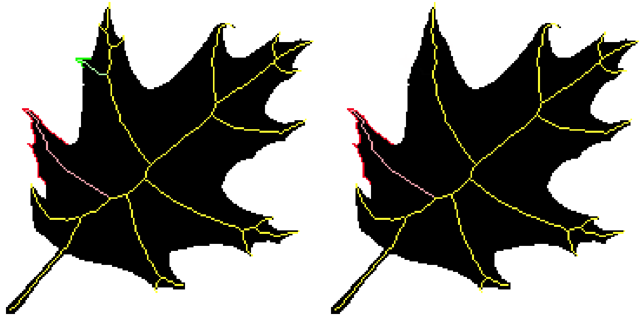
Branch significance can be used to build a hierarchical skeleton, which has exactly the same properties as the initial skeleton. In addition, it is endowed with a weight for each branch, which quantifies the representative power of the branch. In the literature, some authors used hierarchical skeletons in order to describe the shape [10] or, to prune the skeleton [9] in order to remove unnecessary skeleton points or branches caused by noise on the shape contour.

Skeleton pruning methods fall into two categories. The first one consists in smoothing the border of the shape before computing the skeleton [11, 12]. The problem of these methods is that they modify the shape and produce a bias into the skeleton. The second one consists in doing a post-process on the skeleton. To do that, methods can be based on significance values previously assigned to skeleton points, which are removed according to a given threshold [13, 14]. The negative point of these methods is that they are based on the removal of skeleton points. Consequently, in many cases, the skeleton can be disconnected and non-homotopic to the shape. Other methods are based on significance values of each skeleton branch. In this way, the skeleton is pruned branch-by-branch [4, 5, 6, 7, 15]. The proposed method is in this framework. In addition, it is based on successive smoothing operations of the original shape boundary as [11, 12], but keeps track of the initial skeleton in a hierarchical skeleton usable for shape matching.

2. HIERARCHICAL SKELETON

A skeleton is a compact representation of a shape. An ideal skeleton for matching is homotopic to the shape, thin (one-pixel thickness), robust to noise, and should allow to reconstruct the shape. We previously proposed the Digital Euclidean Connected Skeleton (DECS) to fulfill these requirements [15].

THIS WORK IS SUPPORTED BY THE ANR AGENCY (ANR-15-CE38-0004-01)



(a) (b)

Fig. 1. Illustration of skeleton branches, which are more or less important. (a) Skeleton of a leaf, (b) Skeleton of this same leaf with less detail.

When matching two shapes using their skeletons, it seems natural to match branches having the same order of importance. Consider the most important branch of the first shape, *i.e.* the one generated by the general appearance of the shape, an example being the pink branch in Figure 4(a). Obviously, there is no point in matching this branch with a branch of small importance in the second shape, *i.e.* a branch arising from a detail of the shape, an example being a blue branch in Figure 4(b). A detail of the shape is conceptually very different from the general appearance of the shape.

Quantification of the importance of each branch can be applied to skeletons because each branch results from an irregularity more or less important of the border. Consider the example of a tree leaf in Figure 1. The two images are visually similar because three little details have been deleted between Figure 1(a) and Figure 1(b). However, the two obtained skeletons are noticeably different and yet, we would like to match them together. Thus, branches corresponding to these details should have very low importance, to obtain two similar skeletons. More precisely, the green component of the border is responsible for the green skeleton branch. In the same way, the red component of the border is responsible for the pink skeleton branch. The green component is negligible compared to the red one. Consequently, the pink skeleton branch is more important than the green one.

We propose to smooth the shape (Section 2.1) to gradually remove boundary details, and eliminate skeleton branches using the deformable skeleton method (Section 2.2). To quantify the importance of branches, we have chosen the scale at which the skeleton branches disappear when we performed several successive smoothing operations of the border. Doing so, the amount of branches decreases as the boundary is smoothed. Consequently, the longer a skeleton branch persists, the more it is important.

2.1. Shape Smoothing

Let $\mathcal{F}^{(0)} \subset \mathbb{Z}^2$ be the initial shape. Our main application is plant leaves matching, so we deal with connected shapes without hole. We perform successive regularizations of the shape boundary, in order to generate a sequence $(\mathcal{F}^{(1)}, \mathcal{F}^{(2)}, ...\mathcal{F}^{(\lambda)}, ...)$ of increasingly smooth shapes. To extract the contour, we use the works of Roussillon [16], Rosenfeld [17] and Kovalevsky [18]. Then, the shape is smoothed by conserving the area to maintain the shape proportions, which is embodied in the following equation [19]:

$$\frac{\partial C}{\partial t} = \left(\kappa - \frac{2\pi}{\mathcal{L}}\right)\mathcal{N} \tag{1}$$

where $C(s,t) = (x(s,t), y(s,t))$ is the curve varying with respect to time t and spatial parameter s, $\mathcal{L}$ is the Euclidean length of the curve, $\mathcal{N}$ is the inward unit normal of C and κ is the Euclidean curvature defined by:

$$\kappa = \frac{x'y'' - y'x''}{(x'^2 + y'^2)^{3/2}} \tag{2}$$

where $'$ is the partial derivative with respect to s. Under this flow, the curve converges to a circle whose area is equal to the area of the initial shape. The initial curve, which is extracted from the shape, is made up of a sequence of points in $\mathbb{Z}^2$. Eq. (1) is spatially discretized on this sequence of points, which are consequently allowed to take values in $\mathbb{R}^2$. Euler explicit time discretization is used. The curve is resampled after each discrete step of Eq. (1). The shape at smoothing scale λ, denoted by $\mathcal{F}^{(\lambda)}$, is obtained by discretizing the curve back into $\mathbb{Z}^2$ and filling pixels lying inside the discrete closed curve.

2.2. Deformable Skeleton

The general idea is to obtain a skeleton for each smoothed shape and to condense all these skeletons into a hierarchical skeleton. From now on, the number of smoothing operations necessary to come down to a skeleton having only one point is denoted n_{smooth_max}. By abuse of notation, in the remainder of this paper, we can consider that a skeleton S is a set of branches having adjacency relations. A branch is an 8-connected discrete curve. Two branches are adjacent if they have a point in common. Let $\mathcal{F}$ be a shape of $\mathbb{Z}^2$ and let S be its skeleton. Let $N_8(\boldsymbol{p})$ be the 8-connected neighborhood of point $\boldsymbol{p}$.

Definition 1 (Branch) *A branch is a sequence* $(\boldsymbol{p}_0, ..., \boldsymbol{p}_{u-1})$ *of* u *pixels of* S *such that for all* $i = 1, ...(u-2)$, $N_8(\boldsymbol{p}_i) \cap S = \{\boldsymbol{p}_{i-1}, \boldsymbol{p}_{i+1}\}$ *and for* $i = 0$ *and* $i = u-1$, $|N_8(\boldsymbol{p}_i) \cap S| \in \{1, 3, 4\}$. *The extremities of the branch are* $\boldsymbol{p}_0$ *and* $\boldsymbol{p}_{u-1}$.

Definition 2 (End Branch) *A branch* $\flat = (\boldsymbol{p}_0, ..., \boldsymbol{p}_{u-1})$ *is an end branch if* $|N_8(\boldsymbol{p}_0) \cap S| = 1$ *or* $\left|N_8(\boldsymbol{p}_{u-1}) \cap S\right| = 1$, *otherwise* $\flat$ *is an internal branch.*

Definition 3 (Set of Adjacent Branches) *The set of adjacent branches* ($\mathcal{B}_A(\flat_i)$) *of a branch* $\flat_i$ *is defined by:*

$$\mathcal{B}_A(\flat_i = (\boldsymbol{p}_0, ..., \boldsymbol{p}_{u-1})) = \{\flat_j = (\boldsymbol{q}_0, ..., \boldsymbol{q}_{v-1}) \mid \boldsymbol{q}_0 = \boldsymbol{p}_0 \text{ or } \boldsymbol{q}_{v-1} = \boldsymbol{p}_0 \text{ or } \boldsymbol{q}_0 = \boldsymbol{p}_{u-1} \text{ or } \boldsymbol{q}_{v-1} = \boldsymbol{p}_{u-1}\}$$

Definition 4 (Path of Branches) *A path of branches is a sequence* $(\flat_0, ..., \flat_{u'-1})$ *of* u' *branches such that for all* $i = 0, ...(u'-2)$, $\flat_{i+1}$ *belongs to* $\mathcal{B}_A(\flat_i)$.

Definition 5 (Skeleton as set of branches) *Let* S *be a set of branches. If* S *is a skeleton, for every pair of branches* $\flat_i$ *and* $\flat_j$ *belonging to* S*, there exists a path of branches between* $\flat_i$ *and* $\flat_j$.

To build a hierarchical skeleton $S_{hierarchical}$, we implement a method based on a deformation of the initial skeleton in order to adapt it to the successively smoothed shape without recalculating it every time.

Let $\boldsymbol{b}(u,t)$ be the continuous parameterization of a skeleton branch $\flat$ as an open smooth curve and $\boldsymbol{\varrho}_i$ be skeleton points in $\mathbb{R}^2$ initialised from skeleton points $\boldsymbol{p}_i$ in $\mathbb{Z}^2$ to deform them. In this section, superscript (λ) is dropped for convenience. All quantities are considered at current scale λ, unless specified. The deformable skeleton is initialized with the final skeleton of previous scale $\lambda-1$, and each branch evolves according to the following geometric flow:

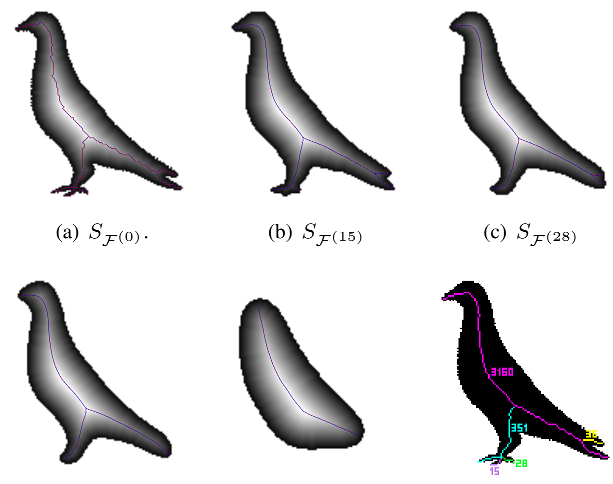
$$\begin{aligned} \frac{\partial \boldsymbol{b}(u,t)}{\partial t} &= (\nabla EDT(\boldsymbol{b}(u,t)) \cdot \mathcal{N}(u,t))\,\mathcal{N}(u,t) \\ &\quad + \alpha \frac{\partial^2 \boldsymbol{b}(u,t)}{\partial u^2} \\ \boldsymbol{b}(\cdot,0) &= \boldsymbol{b}^{(\lambda-1)}(\cdot) \end{aligned} \tag{3}$$

where EDT is the Euclidean Distance Transformation of shape $\mathcal{F}$. Our method being based on active contour, the goal is to minimize total energy. The first term is the external force. The gradient vector field ∇EDT, which points towards ridges of the Euclidean distance map, is projected onto the normal. This makes the branch align itself along the closest ridge. The second term is the internal force, which maintains the smoothness of the curve. Its is weighted by parameter α.

The skeleton is sampled as a set of vertices. The position of the i^{th} vertex is denoted by $\boldsymbol{\varrho}_i \in \mathbb{R}^2$, whereas the set of indices of its adjacent vertices is denoted by η_i. The i^{th} vertex is a branch point if $|\eta_i| = 2$, an endpoint if $|\eta_i| = 1$, and a junction point if $|\eta_i| > 2$. The regularization force is computed as follows:

$$\boldsymbol{r}_i = \begin{cases} \left(\frac{1}{|\eta_i|} \sum_{j \in \eta_i} \boldsymbol{\varrho}_j\right) - \boldsymbol{\varrho}_i & \text{if } |\eta_i| \geq 2 \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{4}$$

The data force attracts vertices towards ridges of the Euclidean distance map $EDT^{(\lambda)}$ along the normal to the branch. The normal vector of branch points is estimated using the two adjacent vertices. For endpoints, the normal considered is the one of the adjacent vertex. For junction points, the gradient of the Euclidean distance map is taken without projection.



(a) $S_{\mathcal{F}^{(0)}}$. (b) $S_{\mathcal{F}^{(15)}}$ (c) $S_{\mathcal{F}^{(28)}}$

(d) $S_{\mathcal{F}^{(36)}}$ (e) $S_{\mathcal{F}^{(351)}}$ (f) $S_{hierarchical}$

Fig. 2. Example of deformable skeleton steps and the associated hierarchical skeleton. Metabranch labels are numbers of smoothing operations necessary to delete metabranches.

Hence, the data force can be written:

$$\boldsymbol{f}_i = \begin{cases} (\mathcal{N}_j \cdot \nabla EDT(\boldsymbol{\varrho}_i))\,\mathcal{N}_j & \text{if } |\eta_i| = 1, \eta_i = \{j\} \\ (\mathcal{N}_i \cdot \nabla EDT(\boldsymbol{\varrho}_i))\,\mathcal{N}_i & \text{if } |\eta_i| = 2 \\ \nabla EDT(\boldsymbol{\varrho}_i) & \text{otherwise} \end{cases} \tag{5}$$

Each vertex evolves according to the following time-discrete equation, with time step Δt

$$\boldsymbol{\varrho}_i^{(t+1)} = \boldsymbol{\varrho}_i^{(t)} + \Delta t \left(\boldsymbol{f}_i^{(t)} + \alpha \boldsymbol{r}_i^{(t)}\right) \tag{6}$$

We took $\Delta t = 0.5$ and $\alpha = 2$. We call $S_{\mathcal{F}^{(\lambda)}}$, the deformable skeleton obtained from smooth shape $\mathcal{F}^{(\lambda)}$.

Definition 6 (Metabranch) *A metabranch* $\flat^m$ *is a sequence of branches* $(\flat_0, ..., \flat_{u'-1})$ *belonging to* $S_{\mathcal{F}^{(0)}}$ *such that for all* $i = 0, ...(u'-2)$, $\flat_i$ *and* $\flat_{i+1}$ *have a same extremity pixel. A metabranch has two extremities*

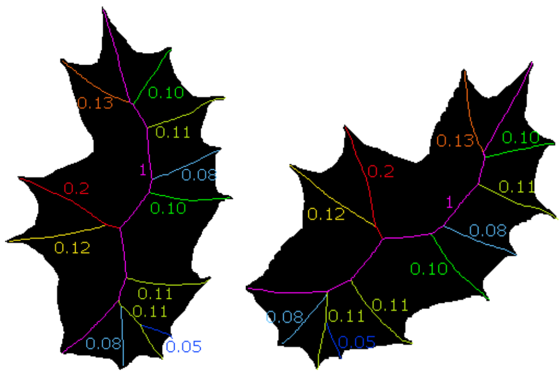
Definition 7 (Hierarchical Skeleton) $S_{hierarchical}$ *is a set of weighted metabranches such that all branches of* $S_{\mathcal{F}^{(0)}}$ *belong to a metabranch. The weight of a metabranch* $\flat^m$ *is denoted by* $w_{\flat^m}$.

Examples of critical steps are presented in Figure 2. Each branch $\flat$ is labelled with a sequence of branches of $S_{\mathcal{F}^{(0)}}$, named a metabranch, corresponding to $\flat$ in $S_{\mathcal{F}^{(0)}}$.

At the end of this step, a sequence of deformable skeletons $h(S)$, including all steps of the deformable skeleton calculated on a shape smoothed between zero and n_{smooth_max} times, is obtained to represent the evolution of the original skeleton. More formally, $h(S) = (S_{\mathcal{F}^{(\lambda)}})_{\lambda \in [0; n_{smooth\_max}]}$. The hierarchical skeleton $S_{hierarchical}$ is obtained from Algorithm 1. An example is presented in Figure 2(f). Then, weights are normalized between 0 and 1.

Algorithm 1 Pseudocode of the creation of $S_{hierarchical}$.

Input: $h(S)$, **Output**: $S_{hierarchical}$
for λ $from\ n_{smooth\_max}\ to\ 0$ **do**
  for all $branch\ \flat\ of\ S_{\mathcal{F}(\lambda)}$ **do**
    if $\flat.metabranch\ \not\subset S_{hierarchical}$ **then**
      $Add\ \flat.metabranch\ to\ S_{hierarchical}$
      $w_{\flat.metabranch} := \lambda + 1$
    end if
  end for
end for
$return\ S_{hierarchical}$



(a) (b)

Fig. 3. (a) Hierarchical skeleton of a tree leaf, (b) Hierarchical skeleton of the same tree leaf as (a), with a 45-degree rotation.

3. RESULTS

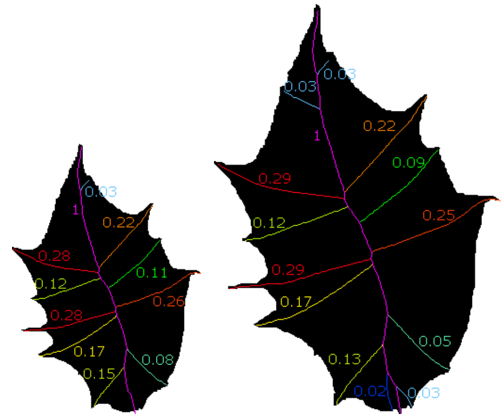
3.1. Stability under Geometrical Transformations

The skeleton obtained with DECS [15] is stable under geometrical transformations. Moreover, the smoothing method being based on continuous space, the shape orientation has no influence on smoothed shapes, the deformable skeleton is thus invariant to rotation. Figure 3 shows two identical shapes, where the second underwent a 45-degree rotation.

We tested our method on different resolutions of the same shape, which is shown in Figure 4. Due to discretization artifacts, the initial skeleton is not necessarily invariant to scaling. When the scale is increased, small details on the boundary may generate additional branches, whereas they did not at the original scale. However, the interest of our method is that these additional branches get a very small weight, nearly zero. This is the case of the dark blue branch in Figure 4(b). An interesting feature of our method is that important branches of the initial shape and the ones of the rescaled shape are the same, and corresponding weights have the same order of magnitude, like fuchsia and red branches in Figure 4(b).

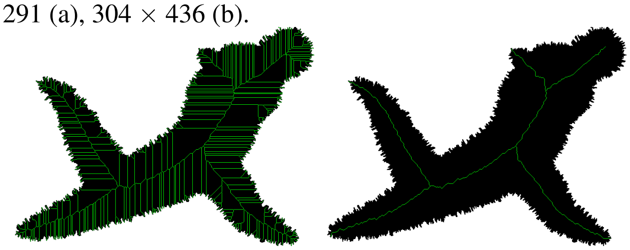
3.2. Skeleton Pruning

As weights quantify the significance of branches, thresholding with respect to weights naturally performs skeleton pruning. By means of a threshold $0 \leqslant \sigma \leqslant 1$, we select important



(a) (b)

Fig. 4. Hierarchical skeleton of a tree leaf at resolution 203 $\times$ 291 (a), 304 $\times$ 436 (b).



(a) $S_{\mathcal{F}(0)}$ (b) $S_{0.3}$

Fig. 5. Example of our method of pruning applied on a Bertrand and Couprie' skeleton [20].

metabranches and obtain a relevant skeleton. The great interest is that this pruning can be performed with a skeleton obtain with another method ([20, 21], for example). Indeed, $S_{\mathcal{F}(0)}$ is not necessary DECS, it can thus be replaced by another skeleton. Figure 5 shows an example of pruning of a skeleton obtained with Bertrand and Couprie's algorithm [20] on a noisy shape (we used $\sigma = 0.3$).

4. CONCLUSION AND FUTURE WORKS

We proposed the construction of a hierarchical skeleton based on successive smoothing operations of the initial shape to reduce details step by step. As smoothing progressively removes shape details, the length of skeleton branches decreases gradually until they disappear. In practing, we do not compute a skeleton at each step but we deform gradually the initial skeleton. We quantified the importance of branches using the normalized scale of disappearance, which is advantageously invariant to image size.

Apart from the fact that our hierarchical skeleton can be used for shape matching, we have also shown that it can be used for skeleton pruning from any kind of skeleton. This work is part of a pattern recognition process and especially tree leaves recognition. We plan to use this hierarchical skeleton for shape matching. Future work also will include an extension to non-simply connected shapes.

5. REFERENCES

[1] R Veltkamp and Michiel Hagedoorn, "4. state of the art in shape matching," *Principles of visual information retrieval*, p. 87, 2001.

[2] Thomas B Sebastian and Benjamin B Kimia, "Curves vs skeletons in object recognition," in *Image Processing, 2001. Proceedings. 2001 International Conference on*. IEEE, 2001, vol. 3, pp. 22–25.

[3] Dominique Attali, Gabriella Sanniti di Baja, and Edouard Thiel, "Pruning discrete and semicontinuous skeletons," in *Image Analysis and Processing*. Springer, 1995, pp. 488–493.

[4] Hongzhi Liu, Zhonghai Wu, D Frank Hsu, Bradley S Peterson, and Dongrong Xu, "On the generation and pruning of skeletons using generalized voronoi diagrams," *Pattern Recognition Letters*, vol. 33, no. 16, pp. 2113–2119, 2012.

[5] HongZhi Liu, Zhong-Hai Wu, Xing Zhang, and D Frank Hsu, "A skeleton pruning algorithm based on information fusion," *Pattern Recognition Letters*, vol. 34, no. 10, pp. 1138–1145, 2013.

[6] Wei Shen, Xiang Bai, Rong Hu, Hongyuan Wang, and Longin Jan Latecki, "Skeleton growing and pruning with bending potential ratio," *Pattern Recognition*, vol. 44, no. 2, pp. 196–209, 2011.

[7] Xiang Bai, Longin Jan Latecki, and Wenyu Liu, "Skeleton pruning by contour partitioning with discrete curve evolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 449–462, 2007.

[8] AndréS SolíS Montero and Jochen Lang, "Skeleton pruning by contour approximation and the integer medial axis transform," *Computers & Graphics*, vol. 36, no. 5, pp. 477–487, 2012.

[9] Luca Serino and Gabriella Sanniti di Baja, "A new strategy for skeleton pruning," *Pattern Recognition Letters*, 2015.

[10] J Park, D Chang, et al., "A hierarchical skeleton-based shape description," *KSPC*, vol. 99, pp. 597–600, 1999.

[11] Pavel Dimitrov, Carlos Phillips, and Kaleem Siddiqi, "Robust and efficient skeletal graphs," in *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*. IEEE, 2000, vol. 1, pp. 417–423.

[12] Doron Shaked and Alfred M Bruckstein, "The curve axis," *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 367–379, 1996.

[13] Robert L Ogniewicz and Olaf Kübler, "Hierarchic voronoi skeletons," *Pattern recognition*, vol. 28, no. 3, pp. 343–359, 1995.

[14] Pavel Dimitrov, James N Damon, and Kaleem Siddiqi, "Flux invariants for shape," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*. IEEE, 2003, vol. 1, pp. I–835.

[15] A Leborgne, J Mille, and L Tougne, "Noise-resistant digital euclidean connected skeleton for graph-based shape matching," *Journal of Visual Communication and Image Representation*, vol. 31, pp. 165–176, 2015.

[16] T. Roussillon, *Algorithmes d'Extraction de Modles Géométriques Discrets pour la Représentation Robuste des Formes*, Thèse de doctorat en informatique, Université Lumière Lyon 2, Nov. 2009.

[17] A. Rosenfeld, "Connectivity in digital pictures," *Journal of the Association for Computing Machinery*, vol. 17(1), pp. 146–160, 1970.

[18] V.A. Kovalevsky, "New definition and fast recognition of digital straight segments and arcs," *Pattern Recognition, Proceedings of the Tenth International Conference on IEEE*, vol. 2, pp. 31–34, 1990.

[19] M. Gage, "On an area-preserving evolution equation for plane curves," *Contemporary Mathematics*, vol. 51, pp. 51–62, 1986.

[20] Gilles Bertrand and Michel Couprie, "Powerful parallel and symmetric 3d thinning schemes based on critical kernels," *Journal of Mathematical Imaging and Vision*, vol. 48, no. 1, pp. 134–148, 2014.

[21] Khalid Saeed, Marek Tabedzki, Mariusz Rybnik, and Marcin Adamski, "K3M: A universal algorithm for image skeletonization and a review of thinning techniques," *Applied Mathematics and Computer Science*, vol. 20, no. 2, pp. 317–335, 2010.