A Continuous Skeletonization Method Based on Level Sets

Rumpf, Martin; Telea, Alexandru

Published in:
EPRINTS-BOOK-TITLE

Joint EUROGRAPHICS - IEEE TCVG Symposium on Visualization (2002)
D. Ebert, P. Brunet, I. Navazo (Editors)

A Continuous Skeletonization Method Based on Level Sets

Martin Rumpf[1], Alexandru Telea[2]

[1] Institut für Mathematik, Duisburg University, Germany
rumpf@math.uni-duisburg.de
[2] Department of Mathematics and Computer Science
Eindhoven University of Technology, The Netherlands
alext@win.tue.nl

Abstract

A robust and efficient method in 2D and 3D for the calculation of skeletons for arbitrary objects is presented. The method is based on the calculation of the distance function with respect to the object boundary. This is combined, in a post processing step, with a new indicator to identify the skeleton, which coincides with the singularity set of the distance map. The indicator is defined as a suitable function of certain local momenta of this distance map and allows a robust and accurate computation of the distance from the skeleton set. This distance is then extended, again via the level set method, onto the whole space. Several applications in 2D and 3D are presented.

1. Introduction

Skeletons and medial axes are of significant interest in many application areas such as object representation, data compression, computer vision, and computer animation. Skeletons provide a simple and compact representation of a 2D or 3D shape that preserves many of the topological and size characteristics of the original. If A is the set to be skeletonized and $d(x,\partial A)$ the distance from A's boundary ∂A at a point x, then the skeleton S is defined via

$$S = \{x \in A \mid \exists y,z \in \partial A,\, y \neq z,\, dist(x,\partial A) = \|x-y\| = \|x-z\|\}$$

Skeletonization methods based on continuous approaches have become increasingly interesting for researchers [3, 9, 10]. These methods detect the skeleton by looking for the singularities (i.e. creases or ridges) of the *distance transform* (DT) of the object's boundary. (see example in Figs. 1, 2,3, and 4). It has been shown [14, 9] that this definition of the skeleton is equivalent with the skeleton definition as the geometric locus of the centers of maximal discs [2]. Moreover, the DT-based skeleton definition is equivalent with the 'prairie fire model' [1], in which the object's boundary evolves in direction of its normal, with constant speed. The skeleton is then defined as the 'shock' points where the moving front collapses onto itself. These points coincide largely with the singularities of the DT, as the evolving front coincides with the DT's isolines, or level sets [9]. Continuous skeletonization methods have several advantages as compared to e.g. thinning methods: the produced skeletons are numerically accurate, distance information to the boundary can be computed to perform object reconstruction, arbitrary topologies are handled by default, and local grid refinement can be easily added to accelerate computations.

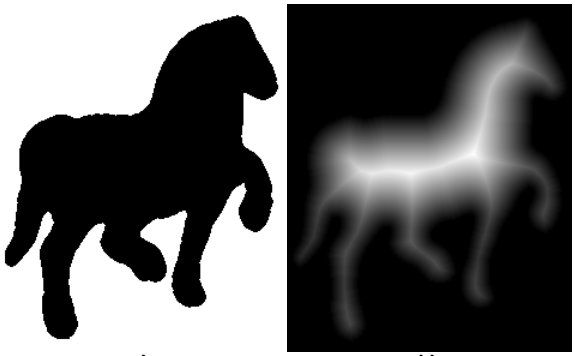
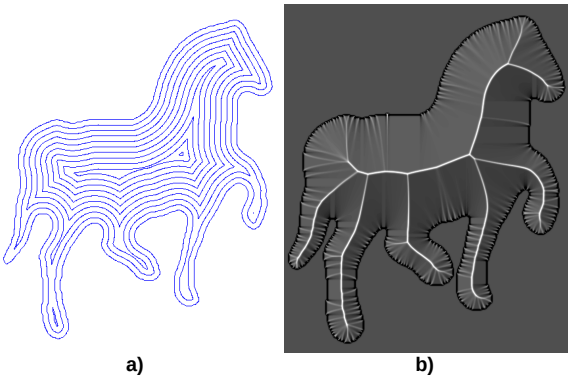


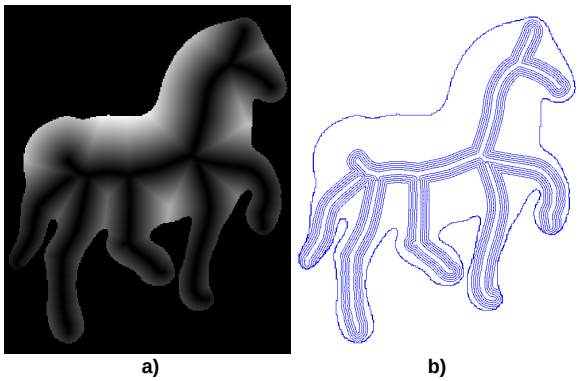
Figure 1: *Object (a) and distance transform (DT) (b)*

The DT, or equivalently the front evolution, can be computed by level set methods, such as the fast marching method (FMM) introduced by Sethian [11]. However, the FMM is not explicitly detecting the DT singularities or shock points of the moving front. Direct computation of the singulari-



a) b)

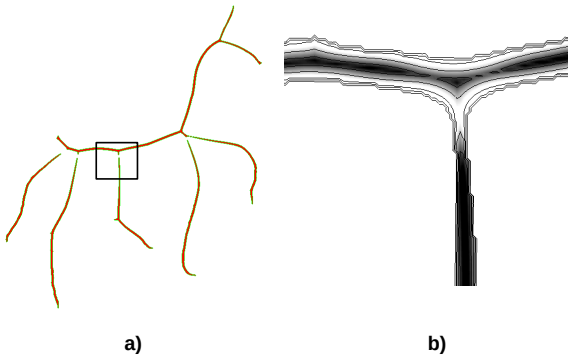
Figure 2: *Evolved boundary under constant speed (a) and skeleton detector (b)*



a) b)

Figure 4: *DT of skeleton (a) and inflated skeleton (b) for object in Fig. 1*

ties is a numerically unstable and delicate process [4, 10, 8, 3] which usually delivers disconnected points [13, 5]. Other continuous skeletonization methods, such as [9], use the observation that skeleton points are generated by compact boundary segments delimited by curvature extrema along the boundary. The skeleton is then built from the separate DTs of the boundary segments separated by the curvature extrema and is restricted to 2D. However, this method relies on accurate detection of curvature extrema along a possibly noisy boundary. Moreover, shapes with holes are treated in a rather complex manner. Another method for front evolution [8] advances the front by tracking marker particles. The skeleton is defined as the shock points where an energy conservation principle is violated. Although numerically more stable than direct singularity detection, this method is hard to implement and computationally expensive: particles must be inserted and removed to preserve a dense and constant particle distribution on the boundary.



a) b)

Figure 3: *Skeleton (a) and detail of distance close to skeleton (b) for object in Fig. 1*

We present a skeletonization technique based on a new method for detecting the singularities of the DT. The method benefits of the full advantages of a continuous formulation. Moreover, we do not make explicit use of derivative, gradient, or divergence operations, so our method is more stable than other continuous methods as cited above. Furthermore, the new approach comes a long with a useful scale space property. In detail, our skeleton indicator relies on a certain filtering of the distance transform, where the filter width acts as a scale parameter. For increasing filter width, more and more skeleton details are ignored and only the dominant parts of the skeleton remain. In Section 2, we introduce the moment concept, with which we build our singularity detector, as shown in Sec. 3. Section 4 presents the complete algorithm. Section 5 shows several 2D and 3D applications.

2. Moment Analysis

In this section, we introduce the *moments* of a d dimensional scalar function. In what follows, we will show how to use differently scaled zero moments to localize the singularities of the boundary's distance transform DT. The first moment encodes directional information of the skeleton. However, in this paper we focus on the zero order moment, the use of the first order moment being subject of a forthcoming paper.

Let us assume that $T : \Omega \to \mathbb{R}$ is a solution of the Eikonal equation [11, 3]:

$$||\nabla T|| = 1 \tag{1}$$

where $\Omega \subset \mathbb{R}^n$ with $n = 2, 3$ has the boundary $\Gamma = \partial\Omega$, and $\Gamma \in C^{0,1}$. As mentioned in Sec. 1, T is the signed distance function to the boundary in the regular case. As a working example, consider the 2D shape in Fig. 1 a. We now regard T as a graph in $\mathbb{R}^{n+1}$ over the whole space $\mathbb{R}^n \supset \Omega$ (see Fig. 1 b). As explained in Sec. 1, the ridges of this graph are the branches of Γ's skeleton. From this graph, we calculate the first moment, which turns out to be a useful indicator for the graph's ridges. The zero-order moment $M_\varepsilon^\circ(x)$ and the

first moment $M^1_\varepsilon(x)$, in the $n+1$ dimensional current point x, are defined as:

$$M^\circ_\varepsilon(x) = \frac{1}{|\mathrm{Graph}B_\varepsilon(x)|}\int_{B_\varepsilon(x)}\begin{pmatrix} y \\ T(y)\end{pmatrix}\sqrt{1+|\nabla T(y)|^2}dy$$

$$M^1_\varepsilon(x) = \frac{1}{|\mathrm{Graph}B_\varepsilon(x)|}\int_{B_\varepsilon(x)}\left[\begin{pmatrix} y \\ T(y)\end{pmatrix} - M^\circ(x)\right] \otimes \left[\begin{pmatrix} y \\ T(y)\end{pmatrix} - M^\circ(x)\right]\sqrt{1+|\nabla T(y)|^2}dy$$

where $B_\varepsilon(x)$ is a small ball of radius ε in the parameter domain $\mathbb{R}^n$, GraphB_ε is the graph under the distance transform, and $|B_\varepsilon(x)|$ and $|\mathrm{Graph}B_\varepsilon(x)|$ are respectively the areas of the ball and the graph. Figure. 5 shows the above for $n=2$, when the graph is a 2-dimensional surface embedded in 3D. Using the Eikonal equation (Eqn. 1), we can simplify the

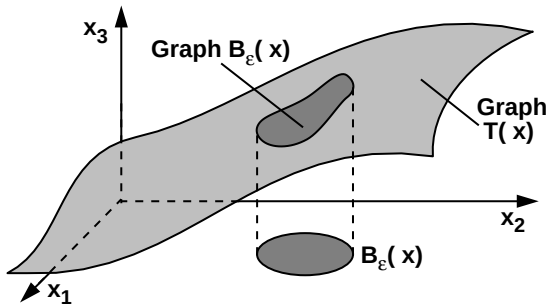


Figure 5: *Definition of zero and first order moments of graph of $T(x)$*

moments' expressions. For the $(n+1)^{\text{th}}$ component of the zero moment – which we denote by $T^\circ_\varepsilon(x)$ and call from now on the zero moment of T – we obtain:

$$T^\circ_\varepsilon(x) := (M^\circ_\varepsilon(x))_{n+1} = |B_\varepsilon(x)|^{-1}\int_{B_\varepsilon(x)} T(y)dy$$

The first moment $M^1_\varepsilon(x)$ gives further information on the normal and tangent space of the skeleton S and allows to identify singularities on the skeleton, such as branching points. In this paper we focus on the zero moment only, which turns out to be a sufficient tool for the calculation of a robust and reliable distance function from the skeleton in areas where the skeleton is a smooth submanifold of $\mathbb{R}^n$.

3. Singularity Detection

Let us now study the relation of the singularities of $T(\cdot)$ and its zero moments evaluated for different scales ε.

3.1. Unidimensional Case

We start with the case $n=1$. Suppose that there is a singularity of the distance map T at a distance d from the current position x, and that $d<\varepsilon$ (Fig. 6 a). We denote by s_+ and s_- the absolute values of the slopes of the graph of T on the left, respectively right side of the singularity. For $n=1$ we deduce from the Eikonal equation (Eqn. 1) that

$$s_+ = s_- = 1.$$

The difference of the actual value of T and its zero moment is:

$$T(x)-T^\circ_\varepsilon(x) = \frac{1}{2\varepsilon}\int_{x-\varepsilon}^{x+\varepsilon} T(x)-T(y)dy = \frac{1}{4\varepsilon}(s_++s_-)(\varepsilon-d)^2$$

where $d = dist(x,S)$ is the distance from the current point x to the skeleton S. Thus we can compute this distance as

$$d = \varepsilon - \sqrt{\frac{4\varepsilon(T(x)-T^\circ_\varepsilon(x))}{s_++s_-}} = \varepsilon\left(1-\sqrt{\frac{2(T(x)-T^\circ_\varepsilon(x))}{\varepsilon}}\right) \tag{2}$$

Outside an ε neighborhood of the skeleton $T(x)-T^\circ_\varepsilon(x)$ vanishes, since T is locally a linear function (for n=1). In conclusion, the quantity $T(x)-T^\circ_\varepsilon(x)$ is a good indicator for singularities in the $n=1$ case: if $T(x)-T^\circ_\varepsilon(x)\neq 0$ then x is closer than ε from the skeleton and the distance can be computed using Eqn 2.

3.2. Higher Dimensional Case

For $n>1$, s_+ and s_- are no longer constant in the considered neighborhood and $T(\cdot)$ is, in general, a nonlinear function. Since ε is small, the graph of T can be approximated by a linear function on both sides of the singularity up to second order in ε in areas where the skeleton S is a smooth submanifold. Consequently, both s_+ and s_- are assumed to be constant quantities up to first order in ε in the considered ε-neighborhood. On the skeleton and close to a point x the sum s_++s_- is computed as:

$$s_++s_- = [\nabla T(x)\cdot N]_S + \mathrm{O}(\varepsilon),$$

where N is the normal to the skeleton and $[\cdot]_S$ the jump operator on S. The error term $0(\varepsilon)$ compensates the variation of the slope at the interface. From now on, we denote by s_+ and s_- respectively the slopes of the distance transform at the point on the skeleton closest to a given point x.

We compute again our singularity indicator $T(x)-T^0_\varepsilon(x)$. Suppose $d=d(x) := \mathrm{dist}(x,S)$ is the distance from the point x to the skeleton S. Thus, consider a ball $B_\varepsilon(x)$ with radius ε, centered in origin, and a spherical cap $C(\varepsilon,d)$ determined by slicing the ball at distance d from the origin. With respect to our smoothness assumption we suppose that inside the ball $B_\varepsilon(x)$ the slice is a approximation of the local skeleton up to first order in position. Denote by h the cap's height (Fig. 6 b,c). Hence we obtain for $T(x)-T^0_\varepsilon(x)$:

$$T(x)-T^0_\varepsilon(x) = |B_\varepsilon(x)|^{-1}\left(\int_{C(\varepsilon,d)}(y_n-d)\ dy\right)(s_++s_-) + \mathrm{O}(\varepsilon^2)$$

The second order approximation in ε is due to the symmetry

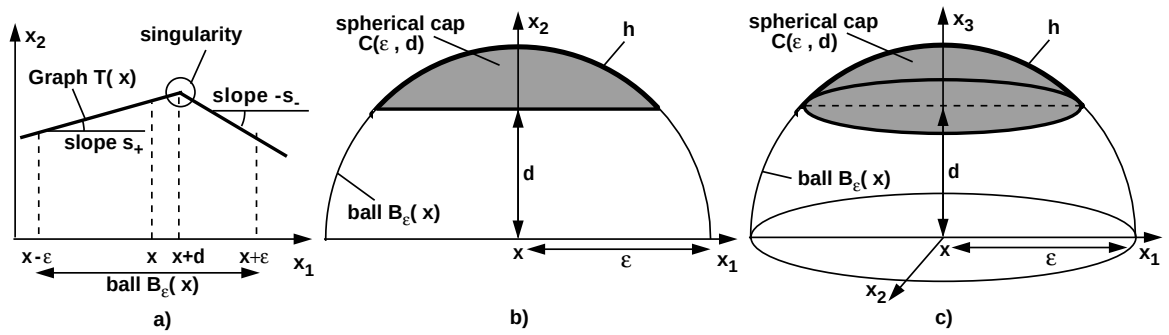


Figure 6: *Singularity detection in 1D (a), singularity detector computation in 2D and 3D (b,c)*

of the geometric configuration. If we introduce $\gamma = \frac{d}{\varepsilon}$, we have

$$T(x) - T_\varepsilon^0(x) = \varepsilon |B_1(x)|^{-1} \left(\int_{C(1,\gamma)} (y_n - \gamma)\ dy \right) (s_+ + s_-) + \mathrm{O}(\varepsilon^2)$$

To simplify the notation in the following discussion, we define

$$g(\gamma) = |B_1(x)|^{-1} \left(\int_{C(1,\gamma)} (y_n - \gamma) dy \right)$$

With this notation, we get

$$T(x) - T_\varepsilon^0(x) = \varepsilon(s_+ + s_-) g(\gamma) + \mathrm{O}(\varepsilon^2) \qquad (3)$$

where $g(\gamma)$ can be explicitly computed based on the integral transform

$$g(\gamma) = \frac{|S_1^{n-2}|}{2|B_1(x)|} \int_0^{\sqrt{1-\gamma^2}} (\sqrt{1-r^2} - \gamma)^2 r^{n-2} dr,$$

where S_1^{n-2} is the $n-2$ dimensional unit sphere and $\|S_1^{n-2}\|$ its $n-2$ dimensional volume (see Appendix for the concrete expressions). In the following section, we shall use $g(\cdot)$ to evaluate the distance to the skeleton in a small skeleton neighborhood.

3.3. Distance Computation

Consider the singularity detector (Eqn. 3) computed using a larger ball $B_{2\varepsilon}(x)$. This gives

$$T(x) - T_{2\varepsilon}^0(x) = 2\varepsilon g\left(\frac{\gamma}{2}\right)(s_+ + s_-) + \mathrm{O}(\varepsilon^2)$$

To solve for γ, we define the ratio f of evaluations of $g(\cdot)$ with respect to dyadic scales. This coincides up to first order in ε with the ratio of the singularity detectors computed with two different balls $B_\varepsilon(x)$ and $B_{2\varepsilon}(x)$, that is:

$$f(\gamma) := \frac{g(\gamma)}{2g\left(\frac{\gamma}{2}\right)} = \frac{T(x) - T_\varepsilon^0(x)}{T(x) - T_{2\varepsilon}^0(x)} + \mathrm{O}(\varepsilon)$$

In the above, f is defined on $[0,1]$,

$$f(0) = \frac{1}{2}, \quad f(1) = 0,$$

and is strictly monotonically decreasing on $[0,1]$ with $|f'| \geq c > 0$. Indeed, $g(\cdot)$ and $g'(\cdot)$ are both monotonically decreasing and $g \geq 0$, thus

$$\begin{aligned} f'(\gamma) &= \frac{1}{2} \frac{g'(\gamma) g\left(\frac{\gamma}{2}\right) - \frac{1}{2} g(\gamma) g'\left(\frac{\gamma}{2}\right)}{g^2\left(\frac{\gamma}{2}\right)} \\ &< \frac{1}{2g^2\left(\frac{\gamma}{2}\right)} \left(g'(\gamma) g(\gamma) - \frac{1}{2} g(\gamma) g'(\gamma) \right) \\ &\leq \frac{g'(\gamma)}{4g(\gamma)} \leq 0 \end{aligned}$$

Hence f is invertible. Since we can numerically compute the ratio $\frac{T(x) - T_\varepsilon^0(x)}{T(x) - T_{2\varepsilon}^0(x)}$, we can compute the distance of x from the skeleton S up to second order in ε

$$\mathrm{dist}(x,S) = \varepsilon\gamma(x) = \varepsilon f^{-1}\left(\frac{T(x) - T_\varepsilon^0(x)}{T(x) - T_{2\varepsilon}^0(x)} \right) + \mathrm{O}(\varepsilon^2). \quad (4)$$

Knowing γ, we can also evaluate the jump of $\nabla T \cdot N$ at the closest point on the skeleton S up to an error $\mathrm{O}(\varepsilon)$:

$$[\nabla T(x) \cdot N]_S = s_+ + s_- = \frac{T(x) - T_\varepsilon^0(x)}{\varepsilon g(\gamma)} + \mathrm{O}(\varepsilon^2)$$

Let us emphasize that the above defined local distance to the skeleton can be considered as a complete solution of the skeletonization problem. First of all, it implicitly defines the skeleton itself as the pre image of the zero distance. Secondly, if we consider interpolation on a discrete grid where the grid points are not *exactly* located on the skeleton, our local distance classifies points close to the skeleton precisely by their distance value. This is obviously the best we can do without generating a new data structure for the skeleton, such as a nonuniform grid refinement, which is not our focus here. Our aim is to stay confined to uniform volumetric respresentations only. Finally, our approach based on moments is robust with respect to discretization, where we replace the

exact distance transform with respect to the boundary ∂A by a numerical solution of the fast marching method (FMM).

4. Skeletonization Algorithm

In this section, we put together the various parts described so far into a complete skeletonization algorithm. The algorithm consists of 5 steps (cf. Fig. 7):

1. We compute the distance T to the boundary of the object to be skeletonized, by using the fast marching method (FMM) to solve Eqn. 1, as in [12, 11]. Concretely, we compute T on a uniform pixel grid in 2D, respectively voxel grid in 3D, inside the boundary (see example in Fig. 1).
2. We evaluate the singularity detector $\frac{T(x)-T_\varepsilon^0(x)}{\varepsilon}$ (Eqn. 3) on T, by numerical integration over all points x (see example in Color Plate i). For accurate results, we use an integration ball size of 5 up to 30 grid points and 1 to 100 subsamples per grid cell.
3. Next, the detector $\frac{1}{\varepsilon}(T - T_\varepsilon^0)$ is thresholded to retain all points where it is larger than a given value. Since $\frac{1}{\varepsilon}(T - T_\varepsilon^0)$ is large close to the skeleton and $O(\varepsilon)$ elsewhere, thresholding outputs an approximate ε-neighborhood of the skeleton S. The threshold must be high enough to limit the extracted neighborhood to a small ε around the skeleton, for the assumptions made in Sec. 3 to hold, but low enough so that all skeleton branches are captured (see example in Fig. 1).
4. We solve for the distance to the skeleton $\mathrm{dist}(x,S)$ for all points x in the extracted neighborhood, as described in Sec. 3.3. Equation 4 is easily solved by e.g. bisection or Newton-Raphson methods, since we know that the function f is invertible.
5. Finally, we propagate $\mathrm{dist}(x,S)$ computed in the previous step to all points of the considered 2D or 3D space. For this, we solve the Eikonal equation starting from an isosurface of $\mathrm{dist}(x,S)$ located within the ε-neighborhood of the skeleton, where $\mathrm{dist}(x,S)$ was computed in step 4. This isosurface is now marched outwards by the FMM algorithm, thus calculating $\mathrm{dist}(x,S)$ until we reach ∂A. Now the distance to the skeleton $\mathrm{dist}(x,S)$ is known everywhere inside the initial set A (see example in Color Plate j).

To check the method's accuracy, we evaluated the difference between the dist field computed by our method for a 2D rectangle discretized on a 512^2 grid, and the exact distance-to-skeleton field. For the rectangle, we can exactly evaluate the latter since we know the exact skeleton. The two distance fields are approximately identical, except close to the skeleton's branching points and tips (Fig. 8), where the error increases to about 2%. This is due to the fact that our local distance transform based on the zero moments is valid only in regions where the skeleton is a smooth submanyfold.

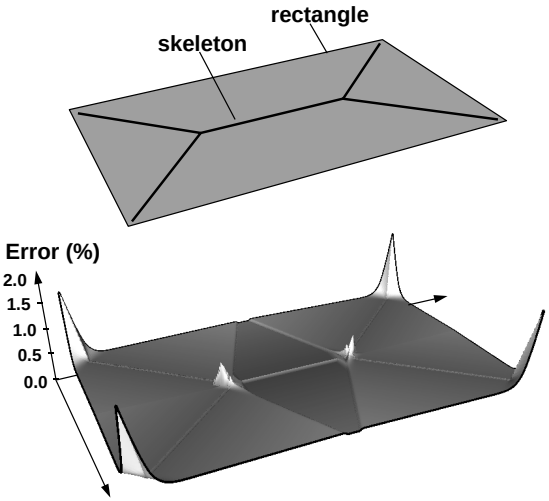


Figure 8: *Difference between exact and computed distance to skeleton fields*

5. Applications

5.1. Smoothing and Reconstruction

We have used the presented skeletonization method for object smoothing and reconstruction (see the 2D examples in Figs. 1, 9, as well as the 3D objects, rendered transparent, together with their skeletons, rendered opaque, in the Color Plate). Objects can be smoothed by computing their skeleton (Fig. 9 b) and then 'inflating' the skeleton (Fig. 9 c), as described in Sec. 4. The three processes involved (DT computation, skeleton computation, and skeleton inflation) deliver a globally smoothed version of the original boundary. Figures 9 d,e shows the difference between the smoothed reconstruction (gray area) and original boundary (thick white line) for the leaf example. The above method can be used to reconstruct simplified objects out of their skeleton, if one avails of a skeleton simplification method. As already sketched in the introduction the parameter ε plays the role of a multiscale parameter: For increasing values of ε, more and more details of the skeleton are disregarded and only the major parts remain. This is a another useful intrinsic feature of our approach requested by many skeletonization applications.

5.2. Morphing

The continuous treatment of the skeletons allows us to easily construct a smooth morphing between an object and its skeleton. For this, consider the blending function B:

$$B = min(\frac{T}{2D}, \frac{1}{2}) + \frac{1}{2} max(1 - \frac{D}{T}, 0)$$

where T is the DT of an object's boundary and D is the DT of the object's skeleton. B achieves a smooth blending between

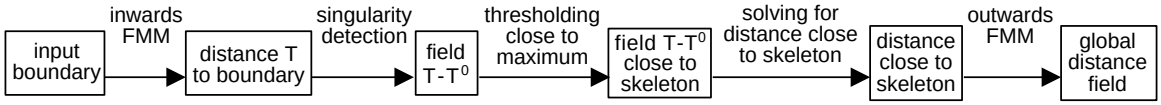


Figure 7: *Skeletonization algorithm pipeline*

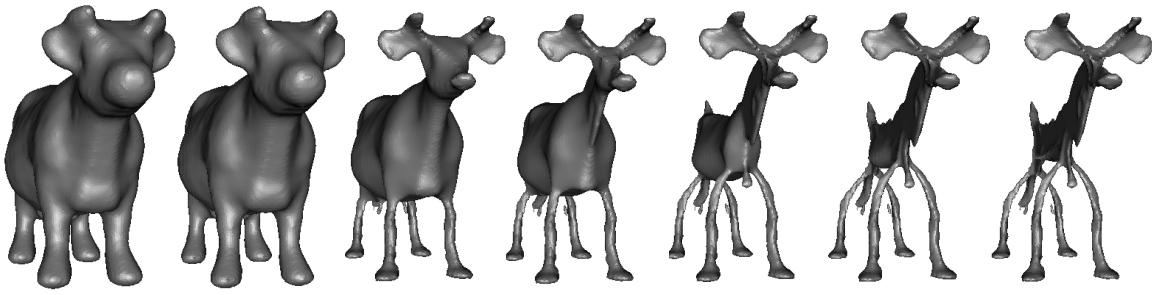


Figure 10: *Morphing between object (leftmost image) and skeleton (rightmost image)*

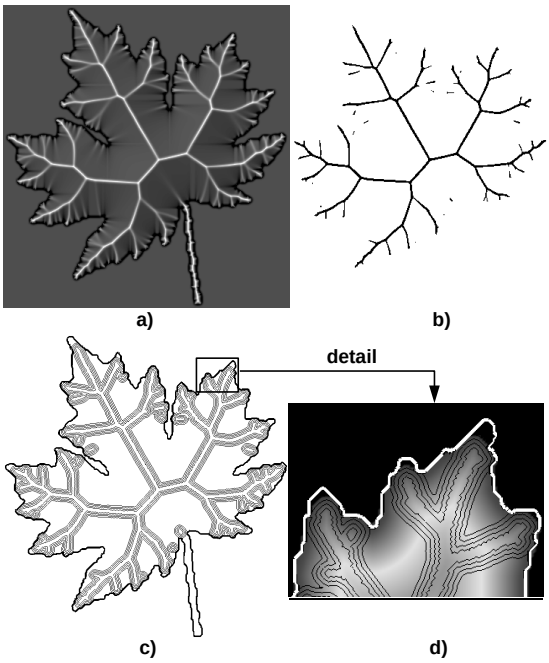


Figure 9: *Singularity detector (a), skeleton neighborhood (b), inflated skeleton and original boundary (c), reconstruction detail (d)*

the boundary and its skeleton. Indeed, for the level set $T = 0$, i.e. the object's boundary, we have $B = 0$. For the level set $D = 0$, i.e. the object's skeleton, we have $B = 1$. For the level set $D = T$, half way between the boundary and the skeleton, we have $B = 0.5$. Extracting consecutive level sets of B between 0 and 1 gives thus a smooth blending of the object to its skeleton, as shown by the 3D example in Fig. 10 (the 3D dataset is taken from the Visualization Toolkit's (VTK) distribution [18]).

6. Discussion

We have presented a robust and simple method to extract an approximate skeleton for a arbitrary 2D or 3D set discretized on a 2D or 3D regular grid. The proposed method is based on a continuous approach and effectively discretized in a second step using approved numerical methodology, such as the fast marching method. No differentiation is required, only integral moments have to be evaluated. This ensures a superior robustness as compared to similar methods. Finally, 2D and 3D skeletons are handled in a single framework.

Concerning complexity, the method requires the application of the fast marching method (FMM) with its cost of $O(N \log N)$. The FMM runs in less than a second for 512^2 2D pixel grids and in a few seconds for 512^3 voxel grids on a Pentium III 800 MHz machine. The moment integration step (Sec. 2 is the slowest part of the process, which is $O(N s k_\varepsilon^3)$ for N grid cells, s subsamples per cell, and an integration ball of radius k_ε grid cells. Using about 20 samples per grid cell, this takes a few seconds for the above mentioned 2D configuration, respectively around two minutes for the 3D configuration. Here is still a huge potential for further improvements of the performance of the algorithm. Finally, solving Eqn 4 numerically takes a few seconds in both the 2D and 3D case.

In the future, we plan to use the presented method in medical imaging applications and for object modeling based on modeling the objects' skeletons. Moreover, we plan to use the higher order moments introduced in Sec. 2 for the topological analysis and simplification of the computed skeletons.

7. Appendix

The following details the computation of g for the 2D and 3D cases by integration (Sec. 3.2).

$$\begin{aligned} g_{3D}(\gamma) &= \frac{2\pi}{2\frac{4}{3}\pi}\int_0^{\sqrt{1-\gamma^2}} (\sqrt{1-r^2}-\gamma)^2 r\,dr \\ &= \frac{3}{4}\int_0^{\sqrt{1-\gamma^2}} r - r^3 + \gamma^2 r - 2\gamma r\sqrt{1-r^2}\,dr \\ &= \frac{3}{4}\left[\frac{r^2+\gamma^2 r^2}{2} - \frac{r^4}{4} + \frac{2\gamma}{3}(1-r^2)^{\frac{3}{2}}\right]_0^{\sqrt{1-\gamma^2}} \\ &= \frac{3}{4}\left[-\frac{1}{12}\gamma^4 + \frac{\gamma^2}{2} - \frac{2}{3}\gamma + \frac{1}{4}\right] \end{aligned}$$

$$\begin{aligned} g_{2D}(\gamma) &= \frac{2}{2\pi}\int_0^{\sqrt{1-\gamma^2}} \left(\sqrt{1-r^2}-\gamma\right)^2 dr \\ &= \frac{1}{\pi}\left[r(1+\gamma^2) - \frac{r^3}{3} - \gamma\left(r\sqrt{1-r^2} + \arcsin r\right)\right]_0^{\sqrt{1-\gamma^2}} \\ &= \frac{1}{\pi}\left[(1+\gamma^2)\sqrt{1-\gamma^2} - \frac{(1-\gamma^2)^{\frac{3}{2}}}{3}\right] - \\ &- \frac{1}{\pi}\gamma\left(\sqrt{1-\gamma^2}\gamma + \gamma\arcsin\sqrt{1-\gamma^2}\right) \\ &= \frac{1}{3\pi}\left[\sqrt{1-\gamma^2}(2+\gamma^2) - 3\gamma\arcsin\sqrt{1-\gamma^2}\right] \end{aligned}$$

References

1. H. BLUM, *A transformation for extracting new descriptors of shape*, In W. Walthen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, MIT Press, 1967.
2. H. BLUM AND R. N. NAGEL, *Shape description using weighted symmetric axis features*, Pattern Recognition, nr. 10, 1978, pp. 167-180.
3. J. GOMES AND O.D. FAUGERAS, *Reconciling Distance Functions and Level Sets*, Journal of Visual Communication and Image Representation, no. 11, 2000, pp. 209–223.
4. C.W. NIBLACK, P.B. GIBBONS, AND D.W. CAPSON, *Generating skeletons and centerlines from the distance transform*, CVGIP: Graphical Models and Image Processing, nr. 54, 1992, pp. 420-437.
5. F. REINDERS, M. E. D. JACOBSON, F. H. POST, *Skeleton Graph Generation for Feature Shape Description*, Proc. IEEE VisSym 2000, Springer, 2000, pp. 73-82.
6. R. L. OGNIEWICZ, *Automatic Medial Axis Pruning by Mapping Characteristics of Boundaries Evolving under the Euclidean Geometric Heat Flow onto Voronoi Skeletons*, Harvard Robotics Laboratory, Technical Report 95-4, 1995.
7. R. L. OGNIEWICZ AND O. KUBLER, *Hierarchic Voronoi Skeletons*, Pattern Recognition, nr. 28, 1995, pp. 343-359.
8. K. SIDDIQI, S. BOUIX, A. TANNENBAUM, S. W. ZUCKER, *The Hamilton-Jacobi Skeleton*, Intl. Conf. on Computer Vision ICCV '99, p. 828-834, 1999.
9. R. KIMMEL, D. SHAKED, N. KIRYATI, A. M. BRUCKSTEIN, *Skeletonization vis Distance Maps and Level Sets*, Computer Vision and Image Understanding, vol. 62, no. 3, pp. 382-391, 1995.
10. S. BOUIX AND K. SIDDIQI, *Devergence-based Medial Surfaces*, Proc. ECCV 2000, pp. 603-618, 2000.
11. J. A. SETHIAN, *A Fast Marching Level Set Method for Monotonically Advancing Fronts*, Proc. Nat. Acad. Sci. vol. 93, nr. 4, pp. 1591-1595, 1996.
12. J. A. SETHIAN, *Level Set Methods and Fast Marching Methods*, Cambridge University Press, 2nd edition, 1999.
13. Y. ZHOU AND A. W. TOGA, *Efficient Skeletonization of Volumetric Objects*, IEEE TVCG, vol. 5, no. 3, 1999, pp. 210-225.
14. U. MONTANARI, *A method for obtaining skeletons using a quasi-Euclidean distance*, Journal of the ACM, 16(4), pp. 534–549, 1969.
15. D. R. MUSSER AND A. SAINI, *STL Tutorial and Reference Guide: C++ Programming with the Standard Template Library*, Addison-Wesley Professional Computing Series, 1996.
16. A. MANZANERA, T. M. BERNARD, F. PRETEUX, B. LONGUET, *MEdial faces from a concise 3D thinning algorithm*, Proc. ICCV'99, pp. 337–343, 1999.
17. T.-C. LEE, R. L. KASHYAP, *Building skeleton models via 3D medial surface/axis thinning algorithms*. CVGIP: Graphical Models and Image Processing, 56(6), pp. 462–478, 1994.
18. *VTK Homepage*, `http://www/kitware.com`.