Skeletal Structures

Silvia Biasotti[1], Dominique Attali[2], Jean-Daniel Boissonnat[3], Herbert Edelsbrunner[4], Gershon Elber[5], Michela Mortara[1], Gabriella Sanniti di Baja[6], Michela Spagnuolo[1], Mirela Tanase[7], and Remco Veltkamp[7]

[1] CNR. - Ist. di Matematica Applicata e Tecnologie Informatiche, Genova, Italy. silvia.biasotti,michela.mortara, michela.spagnuolo@ge.imati.cnr.it
[2] LIS-CNRS, Domaine Universitaire, BP 46, 38402 Saint Martin d'Hères, France. Dominique.Attali@lis.inpg.fr
[3] INRIA, 2004 Route des Lucioles, BP 93, 06904 Sophia-Antipolis, France. Jean-Daniel.Boissonnat@sophia.inria.fr
[4] Department of Computer Science, Duke University, Durham, and Raindrop Geomagic, Research Triangle Park, North Carolina, USA. edels@cs.duke.edu
[5] Technion, Israel Institute of Technology, Israel. gershon@cs.technion.ac.il
[6] CNR - Ist. di Cibernetica "E. Caianello", Pozzuoli, Napoli, Italy. gsdb@imagm.cib.na.cnr.it
[7] Universiteit Utrecht (UU), The Netherlands. remco.veltkamp@cs.uu.nl

Shape Descriptors are compact and expressive representations of objects suitable for solving problems like recognition, classification, or retrieval of shapes, tasks that are computationally expensive if performed on huge data sets. *Skeletal structures* are a particular class of shape descriptors, which attempt to quantify shapes in ways that agree with human intuition. In fact, they represent the essential structure of objects and the way basic components connect to form a whole.

In the large amount of literature devoted to a wide variety of skeletal structures, this Chapter provides a concise and non-exhaustive introduction to the subject: indeed the first structural descriptor, the medial axis, dates back to 1967, which means forty years of literature on the topic.

1 Introduction

The main issue in high-level structuring is to extract an abstract description of the shape that can be more useful for many purposes. For instance, the search in a data base for an object similar to a query shape can be nearly impossible if approached comparing bulks of thousand triangles. Conversely, the process is extremely facilitated when two descriptors of the shapes are compared instead. Of course the performance and the quality of results depends on the conciseness and on the expressiveness of the description. A shape descriptor may be any number, property or function that can be used to discriminate between shapes. For instance, the edge number can be used to classify polygons. Depending on the application tasks and on the shape domain, usually more sophisticated descriptors are needed. In this Chapter we introduce and describe a particular class of shape descriptors, i.e., skeletal structures.

As everybody knows, the word "skeleton" generally indicates the bone structure of vertebrates; in general, skeleton recalls a support structure (e.g., the skeleton of a ship), or the scheme of something (the skeleton of an opera). Translating the concept in the digital context is not straightforward. Intuitively, the skeleton can be defined following two different philosophies: one privileges the aspect of the skeleton of being a *medial structure*, i.e., an entity that always falls inside the shape and is in each point equidistant from the shape boundary. From this point of view, the skeleton of a planar shape is a linear graph, and each point on the skeleton is equidistant from the boundary points of the shape. In the $3D$ space things change: a cylinder with circular base sufficiently far from the bases exhibits a linear skeleton, while the skeleton of an elongated box is conversely a medial surface, i.e., a two dimensional sheet, which extents in the longitudinal direction.

From the other point of view, the skeleton can be regarded as the explicit representation of how the basic components of the shape are glued together to form a whole. A strictly tubular shape has normally one skeletal line, which lays medial to the object and acts as a symmetry axis, usually referred to as a *centreline*. Furthermore, complex objects formed by the arrangement of tubular-like components can be abstracted to a collection of centrelines which split and join, following the object topology.

The definition of skeleton as a medial structure privileges the geometric aspect of the descriptor. Therefore the skeleton retains a strong correspondence with the shape, so that the boundary can be exactly reconstructed, or at least approximated, from the information encoded in the skeletal structure.

Conversely, the second paradigm regards the skeleton as an abstract adjacency graph of salient shape features and relies on shape decomposition in a way that agree with human intuition: recent cognitive research, alongside with new developments in digital imaging and computer vision, has led to a growing consensus that decomposition of shapes into their constituent parts is fundamental to human vision as an early stage of the cognitive process.

Following the previous considerations, a unique formal definition of skeleton in the context of *digital shapes*, i.e., n-dimensional data having a visual representation, can not be given. In this Chapter we will distinguish between *geometric skeletons*, like the medial axis transform, which give a richer encoding of the spatial extent of the shape, and *topological skeletons*, that dismiss some geometric information but make explicit higher level properties of the shape (main features, adjacency relations among parts, number of components, holes, ...).

The choice on which descriptor should be preferred relies on the application context it must cope with. In a variety of applications it is desirable for the skeleton to be linear (e.g., in medical imaging for vascular narrowing detection, in computer-aided screening for early detection of polyps, and so on). Conversely, other applications may require that the skeleton retains a full correspondence with the shape geometry. This is the case of many CAD/CAM applications, where medial surfaces are exploited, for instance, for subdivision of complex solids into simpler pieces for automatic mesh generation and also for the generation of simpler idealised models such as shells and beams for stress analysis.

1.1 Overview

An exhaustive review of the existing literature on skeletal structures would require an effort which is beyond the scope of the Chapter. The goal here is to provide a selection of the methods that are more relevant for subsequent applications in shape modelling. For the classes of methods reviewed we will provide basic definitions and an overview of the structure with respect to different discrete settings. Comparative remarks and examples of their applications will also

be given. The presentation is organized into two main classes: *geometric skeletons* including the medial axis and other medial structures like bisectors, and *skeletons derived from topological structures*, possibly enriched by geometric information to retain a strong correspondence to the shape; the Reeb graph belongs to this category.

Maybe the best known of geometric skeletal descriptors is the *Medial Axis Transform, (MAT)* defined by Blum in the sixties [21]; he first described the medial axis extraction for a 2D shape by analogy with a fire front which starts at the boundary of the shape and propagates isotropically towards the interior. The medial axis is defined by the locations at which the fire fronts collide.

In the planar case the medial axis is a graph, while for shapes in $\mathbb{R}^3$ the MAT is a dimensionally heterogeneous entity composed by curves and surface patches. Small modifications of the input shape can induce large modifications of its medial axis; nonetheless they do not affect the entire medial axis. Typical effects for shapes in $\mathbb{R}^2$ are spurious branches that leave the rest of the medial axis unchanged.

The exact computation of the medial axis is extremely complex in the domain of freeform shapes. Nonetheless, results exist for computing *bisectors* between rational entities exactly. The concept of bisector is strictly related to the medial axis, but while the medial axis can be computed for a given object, the bisector involves more entities, being the locus of points equidistant from two (or more) shapes. We present this approach since bisectors can be effectively used as primitives to construct the MAT and the Voronoi diagram of rational curves (see Section 3.2).

Conversely, many approaches have been adopted to implement Blum's original definition in the discrete case. Basically, we can distinguish them into four categories, depending on the adopted skeletonisation method: *skeleton extraction from Voronoi diagrams; simulation of the grassfire; topological thinning; skeleton extraction from distance maps.* The medial axis of a planar curve can be thought of as the Voronoi diagram generalized to an infinite set of points (the boundary points) [3, 83, 84]. It has been formally shown [28] that the Voronoi diagram becomes an increasingly precise approximation of the continuous medial axis as the density of boundary samples increases. Algorithms which actually try to implement the grassfire process are quite rare; examples are the straight skeleton, first introduced by [1], and the linear axis [97]. Thinning and distance map computation can be directly applied to volumetric discrete representations that are widely used especially in medical applications: most acquisition techniques produce in fact voxel grids, like the Computed Tomography or the Magnetic Resonance Imaging. All these skeletonisation methods are detailed in Section 4.

Concerning topological structures, the *Reeb graph* was defined much before the MAT [89], but its potential in shape description has been understood and formalized later on [94]. Reeb graphs act as a tool for studying shapes through the evolution and the arrangement of the level sets of a real function defined over the shape. This fact relies to Morse theory, [80], that studies the link between the differential properties of a shape and its algebraic topology (in the sense of the number of connected components, number and type of holes, etc.). From this point of view, an object can be partitioned into protrusions, holes and other characteristics and can be efficiently represented as a collection of features with a set of adjacency relations between them. These facts raise the idea that topology-based descriptors, maybe integrated with geometric information, are suitable for dealing with the definition of basic models to represent, generate and manipulate shapes without forgetting the feasibility and the computational complexity of the problem, [15]. In fact, a recent work by Goswami et al. [68] exploits topological structures to locate flat and tubular shaped regions on 3D shapes. Focusing on the level set evolution, we obtain a discrete description which effectively represents the shape

and can be encoded in a topological graph. Some methods follow this paradigm and compute skeletons joining the barycentres of adjacent sections [81, 76, 72].

The remainder of this Chapter is organized as follows: the MAT and geometric skeletons are treated first. In Section 2 the definitions of the main concepts are given of medial axis, Voronoi diagram, shock graphs and bisectors, while the Reeb Graph definition is shifted to the topological skeleton Section (5) for a better reading. Techniques that construct an exact representation of medial structures for particular classes of shapes are detailed in Section 3, while approximated methods are described in Section 4. Skeleton derived from topological structures including the Reeb graph are presented in Section 5. Finally, some concluding remarks and future developments are given in Section 6.

2 Definitions of geometric medial structures

In this Section the concepts of medial axis transform, shock graphs, Voronoi diagrams and bisectors are introduced. All these entities share the property of being *medial* with respect to the shape boundary (medial axis and shock graphs) or to two or more objects (Voronoi diagrams, bisectors); therefore they can be referred to as *medial structures*.

The *medial axis transform (MAT)* has been introduced by Blum [21] as a tool in image analysis. To get an intuitive feeling for this concept, consider starting a grass fire along a curve in the plane. The fire starts at the same time, everywhere along the curve, and it grows at constant speed in every direction. The medial axis is the set of locations where the front of the fire meets itself. Formally, let X be a bounded open subset of the Euclidean k-dimensional space, $\mathbb{R}^k$. The *medial axis*, $\mathcal{M}[X]$, is the set of points that have at least two closest points in the complement of X [78], see Figure 1.

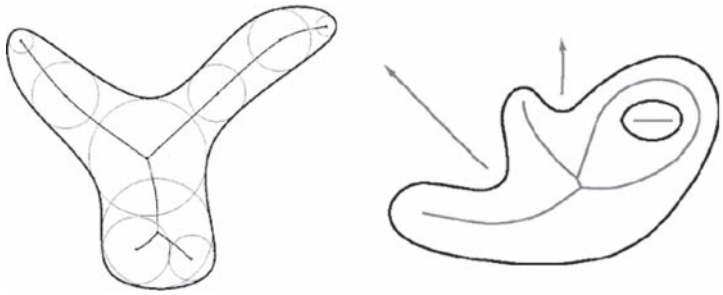


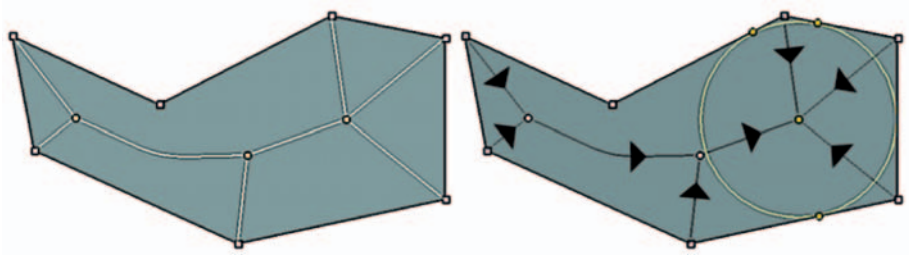
Fig. 1. Medial axis of two planar shapes. In the second example the medial axis is shown also for the external part of the shape.

The medial axis of a shape captures its connectivity, ignoring local dimensionality. More precisely, a shape and its medial axis are homotopy equivalent [78, 91, 101]. In $\mathbb{R}^k$, the medial axis has generically dimension $k-1$, one less than the dimension of the space. In the plane, the medial axis is a (one-dimensional) graph whose branches correspond to regions of the shape it represents. The MAT of planar polygons consists of straight lines and parabolic arcs; each convex vertex of the polygon has an edge of the MAT terminating in it. The MAT structure is very sensitive to noise: the insertion of a new vertex in the boundary of the shape will

cause new edges to appear in the skeleton. In $\mathbb{R}^3$, it is composed of pieces of surfaces, and is sometimes called a *medial surface*. When each point x of the medial axis is weighted with the radius $\rho(x)$ of the maximal ball centered at x, then we have enough information to reconstruct the shape. In other words, the medial axis together with the map ρ provides a reversible coding of shapes. This coding is not necessarily minimal and some shapes, such as finite union of balls, can be reconstructed from proper subsets of their weighted medial axes.

Another medial structure is the *shock graph*, [75], which is obtained by viewing the medial axis as the locus of singularities (shocks) generated during the fire front propagation from the shape boundary. This dynamic view of the medial axis associates a direction and an instantaneous speed of flow to each shock point, [67]. In particular, shock points may be classified according to the number of contact points and to the flow direction, as described in [66]: *source* and *sink* points determine the nodes of the graph while the *links* connect source points to sink ones and define the arcs of the graph. In addition, attributes are associated to the shock graph to store both the intrinsic geometry of the portion of shape corresponding to a link and the radius and the flow direction of each node. Analogously to the MAT, the shock graph structure and the corresponding point classification have been extended to $3D$ shapes [67]. Also, in this case the shock graph structure contains dimensionally heterogeneous components and it is not a planar graph.

The medial axis and the shock graph differ for the interpretation of the structure entities rather than for the geometric abstraction they provide. For example, the shock graph and the MAT of a curve have the same arcs and nodes, but the shock graph associates also to each arc the growing direction of the radius of the bi-tangent spheres, see Figure 2(b). In general, we may consider that the shock graph is a finer partition of the medial axis.



(a) (b)

Fig. 2. The medial axis (a) and the shock graph (b) of two simple curves.

Shock graphs are widely used for image matching, recognition and curve alignment, therefore methods proposed in literature mainly address the problem in the bi-dimensional case and the shape is supposed to be a closed curve.

Strictly related to the medial axis is the *Voronoi diagram* . Given a finite set of points S in $\mathbb{R}^k$, for each point p in $\mathbb{R}^k$ there is at least one point in S closest to p; a point p may be equally close to two or more points in S. For each point in S its Voronoi cell is defined as the subset of $\mathbb{R}^k$ of points closest to it than to any other point in S. The union of Voronoi cells of all points in S is a partition of $\mathbb{R}^k$ called Voronoi Diagram corresponding to the set S.

For instance, in the planar case, given two points a and b, the set of points equidistant from a and b is an infinite line l, the perpendicular bisector of the segment joining a and b. l represents the boundary between the two infinite Voronoi cells of a and b (two half-planes).

The concept of Voronoi diagram is much correlated to the MAT: indeed the MAT of a shape can be approximated by the Voronoi diagram of a finite set of boundary points, as

detailed in Section 4.1; on the other hand, while the MAT is the skeleton of a shape, the Voronoi diagram represents a medial structure between two or more entities (points or objects).

Indeed the Voronoi diagram definition can be easily generalized to set of objects: given m different objects O_1, ... O_m, the Voronoi cell of an object O_i, $(1 \leq i \leq m)$ is defined as the set of points that are closer to the object O_i than to any other object Oj $(1 \leq j \leq m))$. The *bisector* of two objects is the locus of points that are equidistant from the two shapes. The Voronoi cell that contains all points in space that are closer to some object than to any other in space is, therefore, formed out of these bisectors. Similarly, the Voronoi diagram and the medial axis transform are also prescribed by subregions of these bisectors. Therefore, bisectors can be seen as building blocks for the MAT and the Voronoi diagram in such cases where a direct computation of these structures is too complex.

3 Exact representation of medial structures

Indeed, the exact MAT computation was considered for long time affordable only for polygons [77, 65], and more recently for polyhedra [92, 42]. Recently, a few researchers have tackled the problem in the context of freeform (piecewise) rational entities.

Today' accepted approach for computing the planar arrangements of freeform geometry approximates the geometry using piecewise lines and arcs, but this method has noteworthy disadvantages. First, the approach is only an approximation. Second, it is also erroneous. The MAT of a planar shape enclosed by two concentric circles is another mean circle in between them. Yet, by tessellating the two input circles into lines, one introduces numerous C^1 discontinuities along these circles. The resulting MAT will consist of numerous and erroneous edges from the mean circle toward all the C^1 discontinuities in the two boundary circles.

Fortunately, methods exist to compute bisectors of rational entities exactly. For these reason, new approaches aim at computing bisectors between basic freeform shapes as building blocks of every Voronoi Diagram or Medial Axis Transform.

Beyond computing the bisectors between points, lines and arcs in the plane, the current state-of-the-art not only provides complete answers on when an analytic bisector exists between rational manifolds in $\mathbb{R}^n$, but also proposes tractable computational schemes to derive it, as described in Section 3.1.

3.1 Bisectors for freeform shapes

In the following, we will restrict our discussion to rational parametric curves and surfaces, only. Since the rational representation is fully capable of representing all the simple primitives common to the Constructive Solid Geometry (CSG) modelling technique, such as cones, cylinders, spheres, and torii, we will focus on this representation. The fundamental question is whether the bisector between two rational manifolds in $\mathbb{R}^n$ is rational, hence retaining a closure that enables the precise representation of the bisector sheet in the same geometric modelling environment.

The building blocks of every Voronoi diagram or medial axis transform computed in the plane or 3-space must include all cases. These include point-point and point-curve bisectors that are rational in both $\mathbb{R}^2$ and $\mathbb{R}^3$, point-surface and curve-curve bisectors that are rational in $\mathbb{R}^3$, and curve-surface and surface-surface bisectors that are not rational in either space. These non rational curve-curve bisectors in the plane must be differently represented or approximated and such approximations are considered in [54, 62]. [54] maps the problem of computing

the bisector between two planar curves $C(t)$ and $C(r)$ to a zero-set-finding problem in the parameter space of the two curves (t, r). In [62], the planar curve-curve bisector problem is reduced to an envelope of a continuum of point-curve bisectors. The rational surface bisector cases in $\mathbb{R}^3$ are considered in [56].

While the bisectors between points, lines, and arcs have been known for thousands of years, the first real step toward support of freeform geometry was made by Farouki [61]. He showed that the bisector between a point and a rational curve in the plane is indeed rational.

Let $C(t) = (c_x(t), c_y(t))$ be a rational plane curve and $P = (p_x, p_y)$ a point in the plane. The planar bisector sheet, $B(t) = (b_x(t), b_y(t))$, could then be characterized as,

$$\begin{aligned} &< B(t) - P, B(t) - P > = < B(t) - C(t), B(t) - C(t) >, \\ &< B(t) - C(t), C'(t) > = 0. \end{aligned} \tag{1}$$

The first constraint above merely states that the distance between the bisector B and point P should equal the distance between the bisector and curve $C(t)$. The second constraint ensures we measure the distance in an orthogonal direction to the curve, or in the normal space of $C(t)$. It is simple to show that the set of Equations (1) is linear in $B(t)$. Hence one can rewrite Equations (1) as,

$$\begin{bmatrix} c_x(t) - p_x & c_y(t) - p_y \\ c'_x(t) & c'_y(t) \end{bmatrix} \begin{bmatrix} b_x \\ b_y \end{bmatrix} \begin{bmatrix} \langle C(t), C(t) \rangle - \langle P, P \rangle \\ < C(t), C'(t) > \end{bmatrix}. \tag{2}$$

Clearly $B(t)$ in Equation (2) has a rational representation, employing the Cramer rule.

The fact that the number of degrees of freedom equals the number of constraints is a strong hint that the point-rational curve in the plane has a rational representation. Generally speaking, the $(n-1)$-manifold bisector between two input manifolds in $\mathbb{R}^n$ must satisfy three sets of constraints:

1. It must be at an equal distance from the two manifolds.
2. It must be in the normal space of the first manifold.
3. It must be in the normal space of the second manifold.

The distance equality 1 is always there and always imposes one constraint. Constraints 2 and 3 depend on the dimensions of the normal spaces of the two input manifolds. Interestingly enough, Constraints 1-3 are all linear in the bisector function. Hence, the number of constraints for bisectors between zero-, one-, and two-manifolds inputs equal (written as equality constraint + first manifold normal space constraints + second manifold normal space constraints) is listed in Table 1.

	Point	Curve	Surface
Point	1=1+0+0	2=1+0+1	3=1+0+2
Curve	2=1+1+0	3=1+1+1	4=1+1+2
Surface	3=1+2+0	4=1+2+1	5=1+2+2

Table 1. Number of constraints in the bisector computations between points, curves and surfaces. Constraints are listed as distance constraint plus orthogonality constraint(s) to first manifold plus orthogonality constraint(s) to second manifold.

Rational solutions exist whenever the number of constraints, as prescribed in Table 1, is less than or equal to the number of degrees of freedom of the bisector, which is always the

same as the dimension of the space. Every case for which the total number of constraints is less than or equal to two has a rational bisector representation in the plane. The point-point and point-curve bisectors are both rational in the plane. Further, every case for which the number of constraints is less than or equal to three has a rational bisector representation in $\mathbb{R}^3$. Consequently, in $\mathbb{R}^3$, one has a rational representation for point-point, point-curve [61], curve-curve [55] and point-surface [56] bisector cases. Interestingly enough, after inspecting Table 1, we can see that the bisector between two curves is not rational in the plane ($\mathbb{R}^2$), yet is rational in all higher dimensional spaces ($\mathbb{R}^n$, $n > 2$); specifically it is rational in $\mathbb{R}^3$. Figure 3 (see also Figure CP-1 in Appendix E) shows two examples of rational curve-curve and point-surface bisectors in $\mathbb{R}^3$.

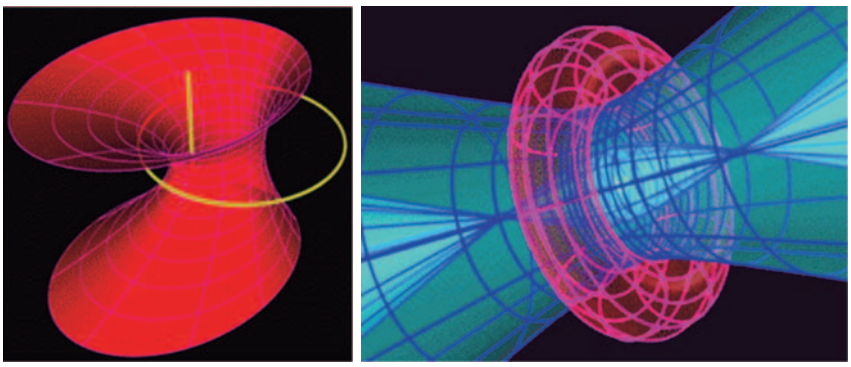


Fig. 3. Curve-curve (left) and point-surface (right) bisector examples in $\mathbb{R}^3$. The curve-curve bisector (in red) on the left is between a horizontal circle and a vertical line (in yellow). The point-surface bisector (in blue) on the right is between a torus (in magenta) and a point at its centre (in yellow). This bisector has two sheets that extend all the way to infinity.

If the number of constraints is less than the number of degrees of freedom, a rational solution still exists. Further readings on these rational cases can be found in [55].

The following set of constraints is defined for the surface-surface bisector, $B = (bx, by, bz)$, in $\mathbb{R}^3$:

$$
\begin{aligned}
0 &= \left\langle \mathcal{B} - S_1(u,v), \frac{\partial S_1(u,v)}{\partial u} \right\rangle, \\
0 &= \left\langle \mathcal{B} - S_1(u,v), \frac{\partial S_1(u,v)}{\partial v} \right\rangle, \\
0 &= \left\langle \mathcal{B} - S_2(s,t), \frac{\partial S_2(s,t)}{\partial s} \right\rangle, \\
0 &= \left\langle \mathcal{B} - S_2(s,t), \frac{\partial S_2(s,t)}{\partial t} \right\rangle, \\
0 &= \langle \mathcal{B} - S_1(u,v), \mathcal{B} - S_1(u,v) \rangle - \langle \mathcal{B} - S_2(s,t), \mathcal{B} - S_2(s,t) \rangle .
\end{aligned}
$$

These five (linear in B) constraints also have seven degrees of freedom: $u, v, s, t, b_x, b_y, b_z$. Hence, having two more degrees of freedom than constraints, the solution space is a two-

manifold, the bisector sheet in $\mathbb{R}^3$ (recall that the bisector sheet in $\mathbb{R}^n$ is an $(n-1)$-manifold). One needs to solve these five equations in seven degrees of freedom – by any means, a non trivial task. In [58], a special non-linear multivariate solver has been employed, presented in [59], which supports cases with non-zero dimensional solution spaces. The solution is given as a dense set of $(u, v, s, t, b_x, b_y, b_z)$ points in $\mathbb{R}^7$. Then, exploiting the given (u, v) parameterisation of S_1, a two-manifold in $\mathbb{R}^3$ is fitted to this data, satisfying the interpolation constraints of $x(u, v) = b_x, y(u, v) = b_y, z(u, v) = b_z$. The non rational bisector between a curve and a surface in $\mathbb{R}^3$ is computed using a similar approach. In Figure 4 (see also Figure CP-2 in Appendix E), the solution point set is shown as yellow points on the fitted bisector sheet in red/magenta.

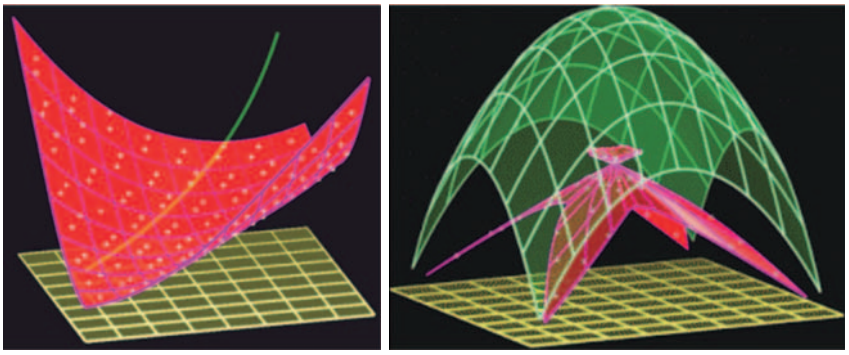
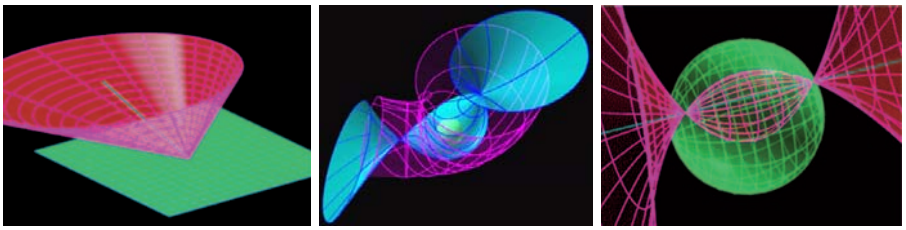


Fig. 4. Curve-surface (left) and surface-surface (right) approximations to the bisector sheets (in red/magenta) in $\mathbb{R}^3$. The dense solution point set is shown as yellow points.

Clearly, being an approximation, the curve-surface and surface-surface bisectors are further more difficult to compute than their analytic counterparts. They become even more difficult when the result is numerically unstable – a not an uncommon case when dealing with bisectors. In many cases, the bisector sheets introduce poles as they vanish at infinity (see the bisector in Figure 3 (right)), and cusps, and hence, self-intersections when the bisector sheet is not regular (see the bisector in Figure 4 (right)). Luckily, many important cases exist where the bisector between a curve and a surface or between two surfaces is indeed rational. One notable simple case is the plane-plane bisector that is another (bisector) plane.

In [57, 86], more special curve-surface and surface-surface rational bisectors in $\mathbb{R}^3$ are identified. The full details of these results are beyond this survey but we will describe a few of the approaches that are presented in [57, 86]. The bisector between a line and a plane in a general position is simply a cone. This is obvious if the line is orthogonal to the plane but also holds for any non-coplanar line (see Figure 5 (a) and Figure CP-3 (a) in Appendix E). An offset is an operation to which the bisector is invariant. The bisector between a sphere and any surface that yields a rational offset could be reduced to a point-surface bisector via the simultaneous offset of the sphere and the other surface by the sphere's radius (see Figure 5 (b) and Figure CP-3 (b) in Appendix E). The bisector computation between a sphere and a canal surface that yields a rational offset is reduced to a bisector computation between a point and rational surface representing the offset of a canal surface. In Figure 5 (c) (see also Figure CP-3 (c) in Appendix E), the line-sphere bisector is similarly reduced to a cylinder-point bisector



(a) (b) (c)

Fig. 5. The bisector sheets (in red/magenta) of a plane and a line (a), a sphere and a canal surface (b), and a line and a sphere (c).

computation, again via an offset operation. Table 2 summarizes the cases known to be rational, as presented in all above references. As can be seen from Table 2, pretty much all CSG primitive shapes yield a rational bisector in $\mathbb{R}^3$ with the exception of the torus, which in most cases has a rational bisector only in special arrangements.

	Point	Line	Plane	Cylinder	Sphere	Cone	Torus
Point	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Line		Yes	Yes	Yes	Yes	Yes	Partial
Plane			Yes	Yes	Yes	Yes	Partial
Cylinder				Yes	Yes	Yes	Partial
Sphere					Yes	Yes	Yes
Cone						Yes	Partial
Torus							Partial

Table 2. The existence of rational bisectors between CSG primitives in $\mathbb{R}^3$.

3.2 Exact computation of the medial axis

As noted above, the construction of the Voronoi diagram and MAT for freeform curves in the plane is more difficult because of the complexity of the bisectors. Ramamurthy and Farouki [63, 64] implemented an incremental algorithm in which the bisectors are inserted one by one and the Voronoi diagram of the curves is updated after each insertion; the MAT is derived from the Voronoi diagram and is represented as a piecewise linear approximation of the actual bisector, computed as the envelope of the point-curve rational bisectors. Ramanathan and Gurumoorthy [87] implemented a different tracing algorithm for the construction of the MAT of a freeform shape. This implementation also approximates the edges of the MAT by computing samples of bisector points on the edges and interpolating these sample points. Piecewise linear curves involve the comparison of expressions with two nested square roots [29]. Efficient and fully robust implementations are few [71]. An exact algorithm for not-necessarily convex polyhedra in $\mathbb{R}^3$ can be found in [41].

A fairly general class of shapes for which it is possible, in principle, to compute the medial axis exactly are the *semi-algebraic sets* . These sets are the solutions of a finite system of

algebraic equations and inequalities. The medial axis of such a set is itself semi-algebraic and can be computed with tools from computer algebra. To describe this, let X be a shape in $\mathbb{R}^3$ whose boundary is a C^1-smooth manifold. We introduce the *symmetry set* of X, consisting of the centers of spheres tangent to the boundary of X at two or more points. It contains all points of the medial axis but also possibly additional points since the spheres are not constrained to bound balls contained in X. Suppose now the boundary of X is defined by the algebraic equation $f(x) = 0$ and 0 is a regular value of f. It follows that the gradient for all points of the boundary is non-zero, $\nabla f(x) \neq 0$. In this case, the symmetry set is the closure of the set of points z for which there exists points x and y that satisfy the following system of algebraic equations:

$$\begin{cases} f(x) &= 0, \\ f(y) &= 0, \\ (x-z) \times \nabla f(x) &= 0, \\ (y-z) \times \nabla f(y) &= 0, \\ \|x-z\|^2 &= \|y-z\|^2, \\ t\|x-y\|^2 &= 1. \end{cases}$$

In the last condition, t is an additional free variable that ensures that x and y are distinct. If 0 is not a regular value of f, we need to add $\nabla f(x) \nabla f(y) s = 1$ as yet another equation, with s as a free variable. Finally, the medial axis is obtained by imposing the additional conditions that $\|u-z\|^2 \geq \|x-z\|^2$, for all points u on the boundary, and z be contained in X. Considering u to be a new free variable, it is possible to remove points from the solution, namely the points z for which $f(z) < 0$ or for which there exists u with $f(u) = 0$ and $\|u-z\|^2 < \|x-z\|^2$. This new set is still semi-algebraic since it is the difference between two semi-algebraic sets.

In [70], the fact that one can express the bisectors of rational curves and the MAT of rational curves as (semi-) algebraic sets is used to derive an algorithm that computes the precise Voronoi cells of rational curves in the plane. Using the precise low degree algebraic formulation offered in [54] to represent the bisector of two planar curves, trimming conditions based on orientation and curvature properties are formulated for these bisectors. The trimmed bisectors are then fed into a lower envelope computation stage in which the Voronoi cells are precisely extracted. The bisector segments are represented as implicit B-spline bivariate forms and hence are algebraic. Further, the locations where adjacent bisectors intersect, and therefore define the corners of the Voronoi cells, are also representable as a set of algebraic constraints. The end result is a precise representation of the Voronoi cells of planar rational curves. Figure 6 shows a few examples of precise Voronoi cells of rational curves.

Dutta and Hoffman [51] proposed a scheme to compute the Voronoi diagram and MAT of CSG primitives. As noted above, their results on bisectors of CSG primitives were partial and, therefore, their work was theoretical and never implemented. A recent result by Ramanathan and Gurumoorthy [88], which is based on their work in [87], constructs the MAT of extruded and revolved shapes. In their work, they exploit the fact that the 3D MAT of an extruded or revolved shape is closely related to the 2D MAT of its creating section curve. This is the only implementation, as far as we can determine, that constructs a MAT in $\mathbb{R}^3$ of surfaces that are not polyhedra.

For the complement of a union of balls in $\mathbb{R}^k$, the medial axis can be derived from the Apollonius diagram of the corresponding spheres or from convex hulls of finitely many points in $\mathbb{R}^{k+2}$ [12, 23]. Perhaps surprisingly, the medial axis of the union of finitely many balls is simpler than that of the complement. As first described in [5], it is piecewise linear and can be constructed from the Voronoi diagram of a finite set of points. As discussed in more detail shortly, the cells of dimension less than k in this diagram may be interpreted as the medial

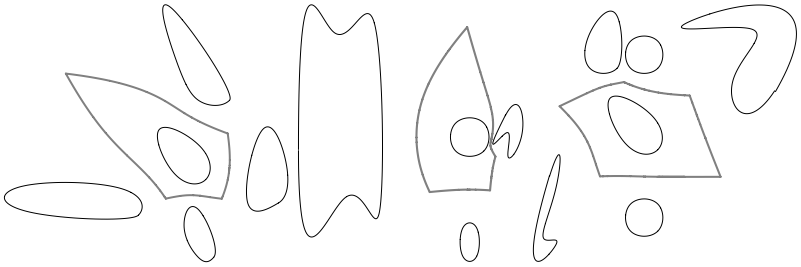


Fig. 6. Three examples of precise Voronoi regions (in gray wide lines) of rational closed parametric curves. The Voronoi region of one curve (the curve inside the Voronoi region) is shown in each example.

axis of a punctured Euclidean space, a case that permits particularly simple exact algorithms. Finally, the MAT of spheres in $\mathbb{R}^3$ was also recently considered in [74].

4 Approximation of the medial axis

Except for the few cases described in the previous Section, when effectively computing a medial representation of a shape, we face the problem of extracting a finite representation of the medial axis. Let $\mathcal{M}[X]$ be the MAT of the shape X. In most cases, we apply an *approximation* of $\mathcal{M}[X]$ that may be either *numerical*, in the sense that our output is always "near" or exactly $\mathcal{M}[X]$, or *geometric*, in the sense that we define new descriptors that are geometrically similar to the skeleton of the shape. The approximation techniques discussed in Section 4.1 refer to numerical approximations of the medial axis. These techniques compute the medial axis as a subset of the Voronoi Diagram of a set of points sampled on the shape boundary. Geometric approximations of the medial axis are shown in Sections 4.2, 4.3 and 4.4. Such approaches are classified on the basis of the skeletonisation method adopted, i.e. implementation of the grassfire propagation, distance map computation and thinning .

Moreover, while techniques based on approximating the Voronoi diagram and on simulating the grassfire represent *continuous* methods that manipulate points with real coordinates (see also Section 4.2), distance maps and thinning constitute *discrete* methods: the object is stored as a collection of pixels/voxels and the resulting skeleton is a connected subset of such pixels/voxels. Working in the discrete space means that we have to face problems specific to this space, which are relevant for medial axis extraction and skeletonisation. It is well known that a different connectivity type has to be used for the shape and for its complement to avoid topological paradoxa. The connectivity type depends on which, among the neighbors of a pixel/voxel, are considered as directly connected to each other. In two dimensions, each pixel p has four neighbors sharing an edge with p, and other four neighbors sharing a vertex with p. The 4-connectivity considers as directly connected to each other pixels sharing an edge, while the 8-connectivity considers both kinds of neighbors. In three dimensions, a voxel v has six neighbors sharing a face with v, twelve neighbors sharing an edge and eight neighbors sharing a vertex. Three connectivity types are hence possible: 26-connectivity, when all three kinds of neighbors are considered, 18-connectivity, when neighbors sharing a face or an edge are considered, and 6-connectivity, when only the neighbors sharing a face are considered. If the

same connectivity type is used for both the object and its complement, a closed curve/surface would not divide its complement into disjoint parts, or an open curve/surface would divide its complement into disjoint parts. For discrete space in two dimensions, the 8-connectivity and the 4-connectivity are generally adopted for the object (and, hence, its skeleton) and for its complement, respectively. In three dimensions, the 26-connectivity and the 6-connectivity are generally used for the object and its complement. Another problem relevant for skeletonisation is strictly related to the nature of the discrete space. In correspondence with regions whose thickness is expressed by an even number of pixels/voxels, the set of centers of maximal balls is 2-pixel/voxel wide. This means that whenever a discrete solution to medial axis extraction or skeletonisation is desired, the resulting set can locally be 2-pixel/voxel wide. Alternatively, which is generally regarded as preferable, the nearly-thin medial axis or skeleton can be reduced to a 1-pixel/voxel thick set by means of final thinning, but the complete reversibility is lost. We remark that the loss in object recovery exclusively regards pixels/voxels on the boundary of the original object. The loss in recovery is generally considered as acceptable, since the actual belonging of pixels/voxels to the boundary of an object obtained after acquisition and digitisation of a continuous object is questionable. We also remark that, in the two-dimensional space, the skeleton is a union of arcs and curves and reversibility is almost completely guaranteed, starting from the 1-pixel wide linear skeleton. In turn, in the three-dimensional space, reversibility is possible only if the so called surface-skeleton, consisting of surfaces and curves, is computed. For solid objects, i.e., objects having no cavities, the surface-skeleton can be furthermore compressed to obtain a linear shape representation (the so called curve-skeleton.) In this case, reversibility is no longer possible. In fact, a large number of centers of maximal balls is unavoidably removed from the surface-skeleton to reduce it to the curve-skeleton. In Sections 4.3 and 4.4 we will mainly focus on linear skeletons.

A more detailed analysis of medial axis extraction and skeleton computation can be found in [31] for objects in the two-dimensional space and in [47] for the three-dimensional case. Other recent contributions on this topic are provided in [40, 43].

4.1 Skeletons from Voronoi Diagrams

We have pointed out that the exact computation of the medial axis runs into obstacles except for certain classes of shapes. Another approach is to approximate the smooth shape with a discrete one, for which the medial axis can be computed exactly.

Despite the intuitive correlation between the Voronoi diagram of a set of points sampling the boundary of a planar shape and its MAT, the formal proof of the Voronoi diagram convergence to the MAT as the number of samples goes to infinite has come rather recently [27]. In this Section we introduce methods that approximate the medial axis of a shape using the Voronoi graph of points sampling its boundary. The role of these methods is twofold: they can either compute the MAT on an approximation of smooth shapes or be applied directly to discrete representations such as triangulations.

In the following we introduce the approximation paradigm; for more details about the stability and computation of medial axes see [6].

Instability and semi-continuity

We think of $\mathcal{M}$ as a transform that maps the shape X to its medial axis, $\mathcal{M}[X]$. As emphasized in [78], geometric shapes are usually not known exactly and represented by approximations of one kind or another. For example, the boundary of a shape may be approximated by a

triangulation obtained by software for surface reconstruction or segmentation. Under these circumstances, it would be important that the transform be continuous. In other words, one should be able to compute an arbitrarily accurate approximation of the output for a sufficiently accurate approximation of the input. Most commonly, one would use the Hausdorff distance to quantify the difference between two inputs and two outputs and this way define what it means for the transform to be continuous. Unfortunately, the medial axis transform is not continuous under this notion of distance: small modifications of the input shape can induce large modifications of its medial axis. This effect is illustrated in Figure 8, where we compare the medial axis of an oval on the left with the medial axis of a set whose Hausdorff distance to the oval is bounded from above by $\epsilon > 0$. The difficulty of approximating the medial axis due to its instability with respect to the Hausdorff distance is a well-known but until recently not well-understood problem.

One can observe experimentally that small modifications of a shape do not affect the entire medial axis. Typical effects for shapes in $\mathbb{R}^2$ are fluctuating branches that leave the rest of the medial axis unchanged. Similarly, for shapes in $\mathbb{R}^3$ we notice fluctuating spikes, added to or removed from the otherwise stable structure. This observation is consistent with the fact that the medial axis is semi continuous with respect to the Hausdorff distance [79, chapter 11]. To explain this concept, we let A and B be subsets of $\mathbb{R}^k$ and write $d_H(A \mid B) \sup_{x \in A} d(x, B)$ for the *one-sided Hausdorff distance* of A from B, where $d(x, B)$ is the infimum of the Euclidean distances between x and points y in B. Observe that $d_H(A \mid B) < \epsilon$ if and only if A is contained in the offset $B^{+\epsilon} = \{x \in \mathbb{R}^k \mid d(x, B) < \epsilon\}$. The *Hausdorff distance* between A and B is $d_H(A, B) \max\{d_H(A \mid B), d_H(B \mid A)\}$. We write A^c and B^c for the complements of A and B and note that the Hausdorff distance between A^c and B^c is generally different from that between A and B. Indeed, $d_H(A^c, B^c)$ is forgiving for small islands of A far away from B, while $d_H(A, B)$ is forgiving for small holes of A far away from B^c. With this notation, we are ready to define the concept of semi continuity. Specifically, a transform $\mathcal{T}$ is *semi continuous* if for every bounded open subset $X \subseteq \mathbb{R}^k$ and for every $\delta > 0$, there exists $\epsilon > 0$ such that for every open subset Y of $\mathbb{R}^k$,

$$d_H(X^c, Y^c) < \epsilon \implies d_H(\mathcal{T}[X] \mid \mathcal{T}[Y]) < \delta. \tag{3}$$

Note that ϵ depends on X. In words, small Hausdorff distance between the complements of X and Y implies that $\mathcal{T}[X]$ is contained in a tight parallel body of $\mathcal{T}[Y]$. As mentioned earlier, this condition is satisfied for $\mathcal{T} = \mathcal{M}$.

Approximation paradigm for the medial axis

The difficulty of computing the medial axis exactly (see Section 3) motivates a serious look at approximation algorithms. A framework that captures a common line of attack to approximating the medial axis is sketched in Figure 7. First, Y that belongs to a class of shapes for which the medial axis can be constructed exactly is found such that it approximates X . Second, the medial axis of Y is constructed. Third, the medial axis of Y is pruned to get a subset $\mathcal{P}[\mathcal{M}[Y]] \subseteq \mathcal{M}[Y]$ that approximates the medial axis of X. The composition of the three steps provides the approximation of the medial axis of X. The most challenging step in this paradigm is the extraction of a subset $\mathcal{P}[\mathcal{M}[Y]$ of $\mathcal{M}[Y]$ that indeed approximates $\mathcal{M}[X]$. Recent mathematical results that rationalize this approach are discussed shortly.

The notion of approximation used in the first step varies between different implementations of the approximation paradigm. It either means that Y is the image of X under a small C^m-perturbation [36], or that the Hausdorff distance between the complements of X and Y is small, as in [35]. Other notions of approximation are conceivable.

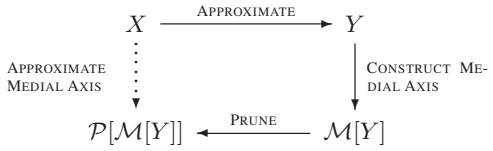


Fig. 7. An approximation $\mathcal{P}[\mathcal{M}[Y]]$ of the medial axis of a shape X can be found as part of the medial axis of a shape Y approximating X.

Punctured Euclidean spaces

We start by identifying a class of shapes for which the medial axis can be constructed exactly and efficiently. We obtain shapes in this class by puncturing the k-dimension real space at a discrete set of locations. Equivalently, we consider the complement of a discrete set of points P in $\mathbb{R}^k$. The medial axis of this space is the *Voronoi graph* of P. Algorithms for constructing the Voronoi graph are well-studied in computational geometry and implementations are available from the geometric software library CGAL [33]. For a set P of n points in $\mathbb{R}^k$, the graph can be constructed in time $O(n^{\lceil k/2 \rceil} + n \log n)$, which is optimal in the worst case because the graph can consist of a constant times $n^{\lceil k/2 \rceil}$ faces. In most practical applications, the number of faces, F, is much less and the output-sensitive algorithm in [34] constructs the graph in $\mathbb{R}^3$ in time $O((n + F) \log^2 F)$. Examples of point sets with provable small Voronoi graphs are so-called κ-light ϵ-samples of compact smooth generic surfaces in $\mathbb{R}^3$, with $F = O(n \log n)$ [9], and κ-light ϵ-samples of polyhedral surfaces in $\mathbb{R}^3$, with $F = O(n)$ [8]. Such samples will be studied in more detail shortly.

Consider a finite point set P whose Hausdorff distance to the boundary of a shape X is less than ϵ and write $\mathrm{Vor}[P]$ for the Voronoi graph of P. Using the semi continuity of the medial axis expressed in (3), the subset of $\mathrm{Vor}[P]$ inside X contains an approximation of the medial axis of X. In the approximation paradigm for medial axes, this subset can be interpreted as part of the medial axis of a shape Y close to X. Following [35], Y is defined to be the parallel body $X^{+\epsilon}$ of X minus the points in P; see Figure 8. Since the Hausdorff distance between P and the boundary of X is less than ϵ, the same is true for the complements of X and the thus constructed space: $d_H(X^c, Y^c) < \epsilon$. In summary, we have $\mathcal{M}[Y] \cap X = \mathrm{Vor}[P] \cap X$.

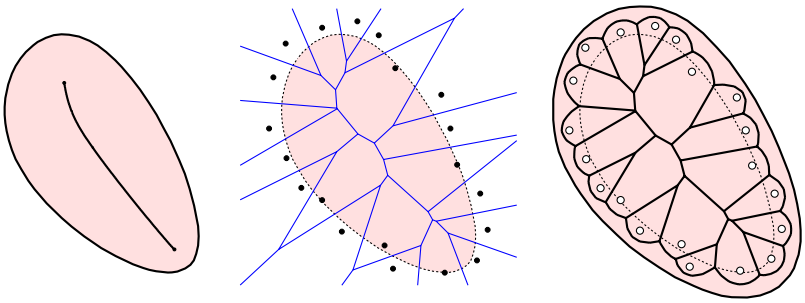


Fig. 8. On the left, a shape X and its medial axis. In the middle, a finite set of points P whose Hausdorff distance to the boundary of X is less than ϵ and its Voronoi graph. On the right, $X^{+\epsilon} - P$ and its medial axis.

Pruning the Voronoi graph

We now consider results that focus on the detailed relationship between the Voronoi graph of a finite point set and the medial axis of the shape whose boundary the points sample. A *sample* of the boundary of a shape X is a finite set of points (exactly and not just approximately) on that boundary. An ϵ-*sample* is a sample whose Hausdorff distance to the boundary of X is less than ϵ. In other words, every point of the boundary is less than distance ϵ away from a point in the ϵ-sample. The ϵ-sample is κ-*light* if the number of sample points within distance ϵ is never more than κ. The ϵ-sample is *noisy* if points are not necessarily on the boundary but at Hausdorff distance less than ϵ to the boundary.

An early result on the connection between the Voronoi graph and the medial axis is due to Brandt [27]. Given a shape in $\mathbb{R}^2$, he takes an ϵ-sample on the boundary curve and considers the Voronoi edges and vertices that are completely contained in the shape; see Figure 9.

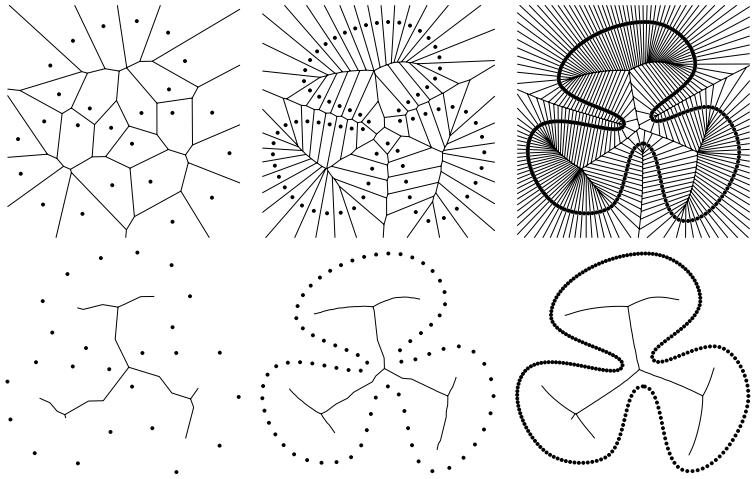


Fig. 9. In $\mathbb{R}^2$, vertices and edges lying inside a shape and extracted from the Voronoi graph of an ϵ-sample of the boundary approximate the medial axis (courtesy of Attali and Montanvert [10]).

Brandt then proves that under some technical conditions on the boundary curve, the portion of the Voronoi graph defined by these edges and vertices approximates the medial axis. Amenta and Bern [2] point out that the direct extension of this result to shapes in $\mathbb{R}^3$ does not hold; see Figure 10. The validity of the extension is spoiled by the existence of slivers in three-dimensional Delaunay triangulations, which occur for ϵ-samples with arbitrarily small $\epsilon > 0$. Roughly, a *sliver* is a tetrahedron whose four vertices are almost co-circular. The location of the Voronoi vertex corresponding to the sliver depends on the four vertices but is generally unrelated to any feature of the surface and does not necessarily lie near the medial axis. As a first step to cope with slivers, Amenta and Bern eliminate all except a few Voronoi vertices they refer to as poles. Every sample point p generates a Voronoi polyhedron and the vertices furthest away from p on the two sides of the surface are the *poles* of p. Clearly, there are at most $2n$ poles for a sample of n points. As proved in [3], for a shape whose boundary is

a smooth C^1-manifold, the poles tend to the medial axis of the shape and its complement as ϵ goes to zero.

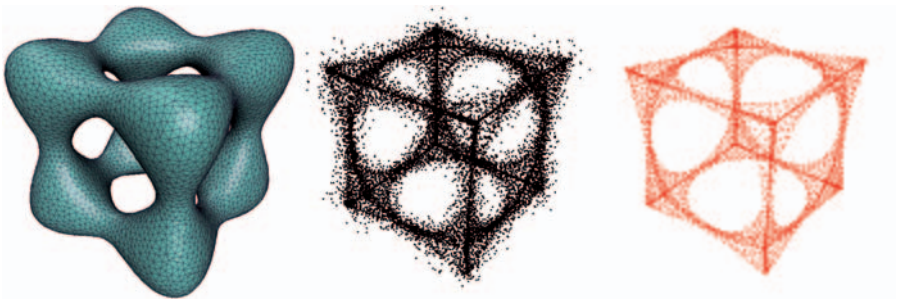


Fig. 10. On the left we see a triangulation of the boundary of a shape in $\mathbb{R}^3$. Its vertices determine a Voronoi diagram whose vertices inside the shape are shown in the middle. The subset of poles inside the shape is shown on the right.

To extend the result of Brandt to $\mathbb{R}^3$, we need more than just points (the poles) near the medial axes, we also need to connect them to form a geometric structure approximating the medial axis. In [3], Amenta, Choi and Kolluri use simplexes of the (weighted) Delaunay triangulation of the poles. To avoid the construction of this weighted Delaunay triangulation and connect the poles directly inside the Voronoi graph, we need to know about its local distance from the medial axis. Bounds on this distance can be found in [7, 22, 38]. Assuming the boundary of the shape is a smooth C^1-manifold and using these bounds, among other things, Dey and Zhao [44] give an algorithm that identifies a sub graph of the Voronoi graph that approximates the medial axis for the Hausdorff distance. We note that the above results are limited to smooth surfaces and to samples of points that lie on that surface. In [35], Chazal and Lieutier obtain a similar result but for more general data: shapes are bounded open subsets and samples are noisy. They introduce a subset of the Voronoi graph, called the λ-Voronoi graph , that approximates the medial axis for a particular sequence of decreasing λ [6]; see Figure 11. Furthermore, for small enough values of λ, this subset is homotopy equivalent to the shape [35].

4.2 Skeleton trough the simulation of the grassfire

Beside methods for the exact computation of a polygon like that proposed in [77], several approximate variations of the medial axis have been proposed in the literature. In particular, in this Section we focus on the straight skeleton and on one of its approximation: the linear axis.

Straight skeleton

Aichholzer and Aurenhammer [1] introduced the *straight skeleton*, a new type of skeleton for polygons. It is closely related to the medial axis, being also based on a wavefront propagation. The wavefront consists of straight line segments and circular arcs (see Figure 12 (a)) and, as it propagates inwardly, the breakpoints between consecutive line segments and circular arcs trace the Voronoi diagram of the polygon. By removing the segments in the diagram incident

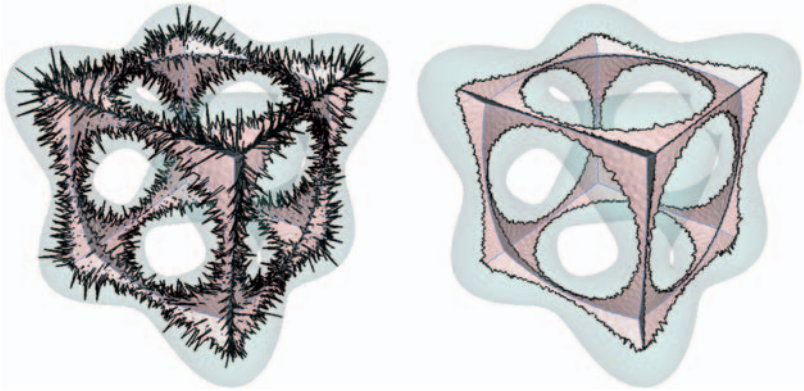
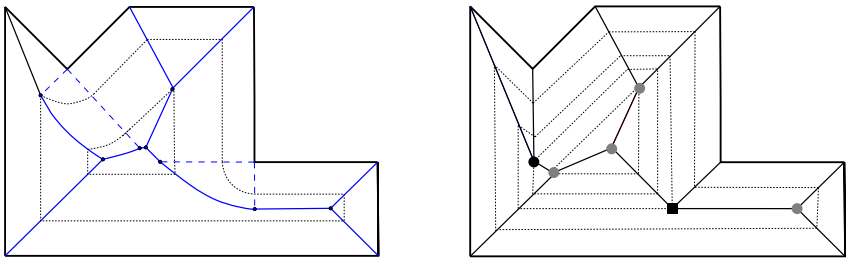


Fig. 11. Two λ-Voronoi graph of the same shape, with λ increasing from left to right, constructed as a subset of the Voronoi graph of a sample of the boundary.



(a) (b)

Fig. 12. Medial Axis (a) vs. Straight Skeleton (b). In (b) the black disk marks a reflex edge annihilation, while gray disks mark convex edge annihilations. An edge-edge collision generates the arc between the black box (vertex-edge collision) and a gray disk (convex edge annihilation)

to the reflex vertices, we obtain the medial axis, which consists of straight line segments and parabolic arcs.

To construct the straight skeleton, we let wavefront edges move parallel to the polygon sides. In contrast to the medial axis, edges incident to a reflex vertex will grow in length. The front remains a polygon, whose vertices during the process trace out the skeleton (see Figure 12(b)). As its name suggests, it consists of straight line segments only. It also has a smaller combinatorial complexity ($n - 2$ internal nodes, with n the number of polygon vertices) than the medial axis ($n + r - 2$ nodes, with r the number of reflex vertices).

A straightforward computation of the straight skeleton consists of simulating the sequence of events occurring in the propagation process described above. Possible edge events are given by intersections of the bisectors of adjacent vertices of the current wavefront. If we maintain a priority queue $\mathbf{E}$ of all these events, indexed by the moment in time when they occur, the next edge event can be detected in constant time. Also after each event occurring in the propagation, only a constant number of updates in $\mathbf{E}$ are necessary. These updates come from changes in the wavefront at the location of the newly occurred event. The priority queue can be created

in $O(n\log(n))$ time, and each update requires $O(\log(n))$ time, where n is the number of vertices in P. Unlike for the edge events, the computation of possible split events can not be done locally. For this purpose we maintain a priority queue $\mathbf{S}$ of all pairs (reflex vertex, wavefront edge), indexed by the moment in time when a split between them would occur. After each event in the propagation a linear number $O(n)$ of updates in $\mathbf{S}$ are necessary. Thus $\mathbf{S}$ can be created in $O(nr\log(n))$ time, and the updates after each event take $O(n\log(n))$ time, where r is the number of reflex vertices of P. The straight skeleton $S(P)$ can thus be computed in $O(nr\log(n))$ time, and the above algorithm requires $O(nr)$ space.

A faster algorithm that uses more complex data structures can be found in [60]. It runs in $O(n^{1+\epsilon} + n^{8/11+\epsilon}r^{9/11+\epsilon})$ time with a similar space complexity, where ϵ is an arbitrarily small positive constant. Eppstein's algorithm simulates the sequence of interactions between edges and vertices in the propagation process. If the polygon P is interpreted as the outline of a building's groundwalls, the straight skeleton is the projection of a roof over P, whose facets are all of equal slope. In simulating the events defining the skeleton, they view time as a third spatial dimension, so that the propagation process becomes an upward sweep of the roof of the polygon with a horizontal plane.

A more recent algorithm by Cheng and Vigneron [37] computes the straight skeleton of a non-degenerate simple polygon in $O(n\log^2 n + r\sqrt{r}\log r)$ expected time. For a degenerate simple polygon, its expected time bound is $O(n\log^2 n + r^{17/11+\epsilon})$.

The Linear Axis

When a simple polygon contains sharp reflex angles with short incident edges, its straight skeleton gives counterintuitive results (see the left column of Figure 14). In [97], Tanase and Veltkamp introduce the *linear axis*. It is based on a linear wavefront propagation like the straight skeleton, but the discrepancy in the speed of the points in the propagating wavefront, though never zero, can decrease as much as wanted.

More formally, let $\{v_1, v_2, \ldots, v_n\}$ denote the vertices of a simple polygon P and let $\kappa = (k_1, k_2, \ldots, k_n)$ be a sequence of natural numbers. If v_i is a convex vertex of P, $k_i = 0$, and if it is a reflex vertex, $k_i \geq 0$. Let $\mathcal{P}^\kappa(0)$ be the polygon obtained from P by replacing each reflex vertex v_i with $k_i + 1$ identical vertices, the end points of k_i zero-length edges, which will be referred to as the *hidden edges* associated with v_i. The directions of the hidden edges are chosen such that the reflex vertex v_i of P is replaced in $\mathcal{P}^\kappa(0)$ by $k_i + 1$ "reflex vertices" of equal internal angle.

Then, the linear axis $\mathbf{L}^\kappa$ of P, corresponding to a sequence κ of hidden edges, is the trace of the convex vertices of the linear wavefront $\mathcal{P}^\kappa$ in the above propagation process. $\mathbf{L}^\kappa$ is a subset of the straight skeleton of $\mathcal{P}^\kappa(0)$; it is sufficient to remove the bisectors traced by the reflex vertices of the wavefront (see Figure 13 (a)). If each reflex vertex v_j of internal angle greater than $3\pi/2$ has at least one associated hidden edge ($k_j \geq 1$), then $\mathbf{L}^\kappa$ is a connected graph. This is because only bisectors incident to reflex vertices of P are removed from the straight skeleton of $\mathcal{P}^\kappa(0)$ in order to obtain $\mathbf{L}^\kappa$.

Obviously, the larger the number of hidden edges, the better the linear axis approximates the medial axis. A thorough analysis of the relation between the number of the inserted hidden edges and the quality of this approximation is given in [97]. They introduce the notion of *ε-equivalence* between two skeletons. Nodes in the two skeletons are clustered based on a proximity criterion, and the ε-equivalence between the two skeletons is defined as an isomorphism between the resulting graphs with clusters as vertices. This allows to compare skeletons based on their main topological structure, ignoring local details. In [97], an algorithm is given

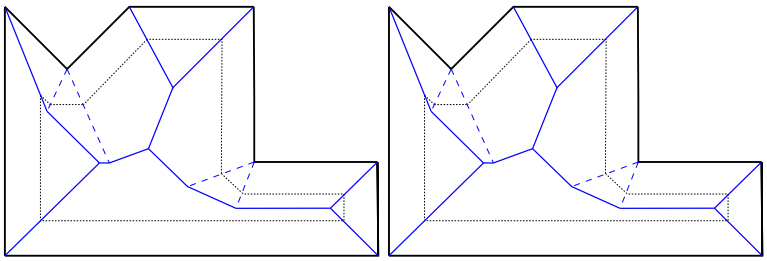


Fig. 13. (a) The linear axis in the case when one hidden edge is inserted at each reflex vertex. A linear wavefront is drawn in dotted line style; the dashed lines are the bisectors that are not part of the linear axis. (b) The linear offset (solid line) of a reflex vertex with 3 associated edges is made of 5 line segments tangent to the uniform offset (dotted line) of this vertex.

for computing the number of hidden edges for each reflex vertex such that the resulting linear axis is ε-equivalent to the medial axis. The whole linear axis computation takes linear time for polygons with a constant number of nodes in any cluster. There is only a limited category of polygons not having this property. Implementation results suggest that in practice only a few hidden edges are necessary to yield a linear axis that is ε-equivalent to the medial axis.

4.3 Skeletons based on topological thinning

Thinning refers to the process of removing pixels or voxels from a discretised object in an attempt to whittle the object down in topological fashion to a more simple representation consisting of connected, unit-wide pathways of pixels or voxels. This process, applied to elongated objects characterized by nearly constant thickness (e.g., printed or hand-written characters, line drawings, blood vessels, or branching patterns of air passageways in the lungs), leads to a set of lines centered within the object and retaining the relevant structural and shape information of the object. For this reason, the main focus of thinning is the preservation of topology, with the primary purpose being to aid in the identification of a basic structure.

Solutions for different grid types such as the rectangular, the triangular and the hexagonal grid have been proposed. Rosenfeld [90] provides a list of over 160 papers on thinning; note, however, that the vast majority of these papers deal with the problem in two dimensions. Ideally, thinning is an isotropic compression process. Since compression takes place from all directions at the same rate, its implementation by means of a parallel algorithm is a natural choice. Actually, both parallel and sequential algorithms have been developed and the literature includes a huge number of papers on this subject (for a survey of two-dimensional thinning algorithms, see e.g. [46]). In parallel algorithms, the processing done at each iteration is a function of the object resulting from the previous iteration only. In sequential algorithms, the elements are processed one after another and are updated in terms both of the object resulting from the previous iteration, and of the modifications produced so far in the current iteration. Thus, the structure of the set resulting from a sequential algorithm depends on the order in which pixels/voxels are processed. Sometimes, spurious branches appear in a particular order of processing, but do not appear in a different order. End-point detection criteria have great importance in this case, to guarantee isotropic object compression and to avoid unwanted shortening of branches in the resulting set.

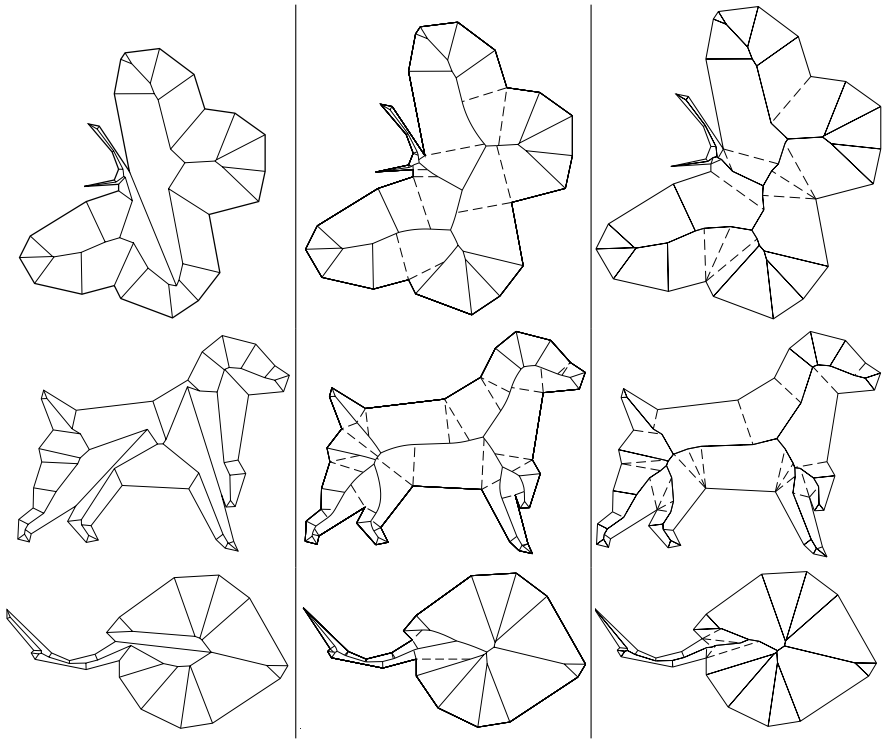


Fig. 14. A comparison of the straight skeleton (left column), the medial axis (middle column), and the linear axis (right column). The skeletons are drawn in solid line style. The dashed lines in the medial axis figures are the Voronoi edges, which are not part of the medial axis. The dashed lines in the linear axis figures represent the bisectors traced by the reflex vertices of the wavefront, which are not part of the linear axis. In these examples, the linear axis is isomorphic with the medial axis ($\varepsilon = 0$).

Topologically oriented thinning consists of repetitive testing and subsequently deletion of pixels or voxels on the boundary of the object, whenever their removal does not alter the topology of the thinned shape. However, as said above, in order the resulting set reflects the geometrical structure of the object, removal operations should be combined with suitable preservation criteria to avoid non isotropic object compression and unwanted shortening of branches in the resulting representation. Practically, in correspondence with every significant protrusion of the object, a branch is expected to be found in the thinned shape. To correctly map protrusions with branches, the tip of each protrusion should be identified and an element in correspondence of each tip (i.e. the end-point of a branch) should be preserved from removal. Unfortunately, most of the existing thinning algorithms do not ensure that the previous correspondence between tips of protrusions and end-points always holds, so their performance is likely to become unacceptable when a wide repertory of objects is to be processed. This behaviour is imputable to the fact that removal occurs during a 'blind' sequential process, that uses the property satisfied by

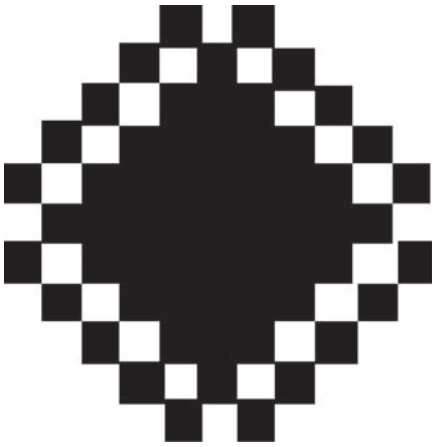


Fig. 15. A lace-shaped 2D object that cannot be reduced to one-pixel wide subset.

the end-points in the resulting set, i.e. the property of having only one neighbor in the skeleton branch, as a criterion to detect the end-points during thinning. This may cause end-points to be originated or not, depending on the order in which the chosen sequence of removal operations is applied to the object's elements. To overcome this problem, the boundary configurations that are assumed to be sufficiently significant to originate end-points, should be identified at the beginning of each iteration of the object compression process, before applying the removal operations. In the opposite case, the sequential way of examining and deleting elements would change the geometry of the neighbourhood the elements are embedded in and may allow the creation of spurious end points, as well as an excessive shortening of significant branches. Effective criteria to correctly identify the tips and mark therein the elements, which will be the end points in the resulting thin set, can be based on the distance of boundary elements from the interior of the object at each iteration of thinning. Boundary subsets including elements whose distance from the interior of the object overcomes a given threshold are preserved from removal, as they correspond to significantly elongated object protrusions [30]. Alternatively, effective criteria can be based on the selection and preservation from removal of all centers of maximal balls in the distance map of the object. In fact, in correspondence with the tip of an object protrusion, a maximal ball of the object exists, whose boundary fits the boundary of the object protrusion for a (wide) connected portion. The center of such a maximal ball can be selected as the endpoint of the branch corresponding to the protrusion(for more details, see next Section).

Topological thinning guarantees connected skeletons; on the other hand, topological thinning does not obligatory produce perfectly thinned output (i.e. one-pixel/voxel-wide paths) since there exist arrangements of pixels/voxels which cannot be further eroded, unless altering object's topology. A 2D example is the lace-shaped object shown in Figure 15, whose border pixels are all non-removable. The alternative approach is based on distance map computation.

4.4 Skeletons from distance maps

Like thinning, skeletonisation based on distance maps is especially suitable for image processing and pattern recognition, and in general for the analysis of discrete objects represented

by grids of pixels or voxels. While thinning is mainly suited to elongated objects, distance map based skeletonisation is also suited to objects that are not elongated as well as to objects that have variable thickness, and provides a representation including also surfaces/branches originating from significant convexities of the boundary of the objects. Distance map based skeletonisation is more directly related to the Blum's notions of a symmetric point and a growth process. In fact, in the distance map the centers of the maximal balls can be easily detected and assigned to the skeleton. The detection of the remaining pixels/voxels necessary to guarantee that the skeleton has the same homotopy type as the object is also an easy task, due to the structure provided by the distance map to the portion of space occupied by the object. Differently from iterative thinning, which requires a number of iterations proportional to the object thickness and, hence, a generally large number of scans of the image when sequential computers are used, distance map based skeletonisation requires a small number of scans, independent of object thickness. Distance map based skeletonisation directly identifies and marks on the distance map of the object the elements that are recognized as belonging to the skeleton, due to the local configuration they are embedded in. The set of elements detected on the distance map includes all the centers of maximal balls (which implies skeleton reversibility), is symmetrically placed within the object and has the same topology as the object. This set is likely to be 2-element wide, in correspondence with object parts characterized by a thickness expressed by an even number of elements. To obtain the unit-wide skeleton, a final thinning, based on topology preserving removal operations, is necessary. We point out that in the three-dimensional case, the skeleton computed by means of the distance map is actually a surface-skeleton. For solid objects, the surface-skeleton can be furthermore compressed to a linear structure, the curve-skeleton, by using an iterative thinning, based on topology preserving removal operations, see e.g. [96]. The so obtained curve-skeleton, though providing a significant representation of the object's shape, does no longer allow object recovery.

In the distance map, each object point is labeled with its distance to the nearest background point. The distance of an element measures the length of a shortest path from that element to the background, where the path consists of elements linked to each other according to the selected connectivity type. Good approximations to the Euclidean distance are obtained by using weighted distances, where suitable integer weights are employed to compute the contribution given to the length of the path by the elements, depending on their relative positions (see, e.g., [24, 25]).

Ridges of the distance map are expected to belong to the skeleton, since they are centrally located within the object. Almost all the elements of a ridge are centers of maximal balls. As such, they can be identified by comparing the distance label of the element z at hand with the distance label of its neighbors, since this is equivalent to comparing the radii of the balls centered on z and on its neighbors. The extrema of a ridge, which are not necessarily centers of maximal balls, can be identified by taking into account that they are placed in saddle configurations. Their detection can be accomplished by counting for each element z, the number of components consisting of neighbors of z with distance labels larger than or equal to the distance label of z, and the number of components of neighbors of z with distance labels smaller than the distance label of z, respectively. Slopes connecting the ridges in the distance map are also expected to belong to the skeleton, to guarantee that the skeleton has the same homotopy type as the object. These linking elements can be found by growing, from the already detected skeletal elements, connecting paths according to the increasing value of the gradient in the distance map. For skeletonisation in the three-dimensional space, besides the linking elements necessary to guarantee skeleton connectedness, also further voxels necessary to prevent the creation of spurious tunnels have to be assigned to the surface skeleton. Roughly speaking, a distance map based skeletonisation algorithm includes three steps:

- distance map computation;
- identification of ridges and slopes;
- reduction of the set of ridges and slopes to unit width.

Obviously different skeletons are obtained depending on the chosen distance function. A number of algorithms can be found in the literature, each of which tailored to a specific distance function (as an example, see [4, 45, 48] for the two-dimensional case, and [49], for the three-dimensional case). Although all distance map based skeletonisation algorithms follow more or less the above scheme, ad hoc rules are often used (for instance to identify the centers of the maximal balls, or to obtain skeleton connectedness through the linking elements), which apply only to the specific distance case.

An important post-processing step is devoted to skeleton simplification [26] and pruning [45, 95]. Simplification is done in the three-dimensional case only, to remove from the surface-skeleton short peripheral curves, whose presence would only make the curve-skeleton structure unnecessarily complex. Pruning is done both in three and in two dimensions and should not be simply regarded as an optional step for a skeletonisation algorithm. In fact, pruning is useful to get rid of superfluous noisy branches and is indispensable to make the linear skeleton stable under object rotation, by eliminating those branches whose presence in the skeleton depends on object orientation. In turn, a post-processing aimed at improving skeleton aesthetics by removing zigzags mostly created by final thinning, can also be performed to favour the use of the skeleton for shape analysis.

In Figure 16, the skeleton of an object in the two-dimensional space is shown, which has been computed according to different distance functions. Namely, the Manhattan distance $d(1,2)$, the chessboard distance $d(1,1)$, the weighted distance $d(3,4)$, which assigns weights 3 and 4 to the steps in the path via the edge-neighbors and the vertex-neighbors respectively, and the weighted distance $d(5,7,11)$, which also consider as possible neighbors along the path pixels that can be reached with the knight move in the game of chess and assigns weights 5, 7 and 11 to the steps via edge-, vertex- and knight-neighbors along the path. In each row, from left to right, the nearly thin skeleton, the unit-wide skeleton and the skeleton resulting after pruning non significant peripheral branches are shown.

In Figure 17, a $3D$ object, its surface skeleton, computed according to D^6, and the curve skeleton obtained by furthermore compressing the surface skeleton by the algorithm [96] are shown. As said before, in the three-dimensional case only the surface skeleton is reversible.

5 Skeletons from topological structures

Methods grouped in this Section have in common the property of coding the evolution and the arrangement of the level set curves of a real, at least continuous, function defined on the shape. The most popular representative of this class of descriptors is the Reeb graph [89].

In principle, topological graphs give an abstract representation of the shape structure, with no information about the geometric embedding. Nevertheless, salient geometric information can be extracted from the shape and attached to the skeleton, thus obtaining a representation that is not only topological but retains also a geometric correspondence with the original shape. In this Section we overview the most popular techniques for constructing skeletons from topological structures related to level sets, distinguishing them in two main classes: those that derive from wave-like expansion techniques and those that more explicitly refer to the Reeb graph definition. Contour trees, a specific kind of Reeb Graphs for scalar fields, are treated in [19].

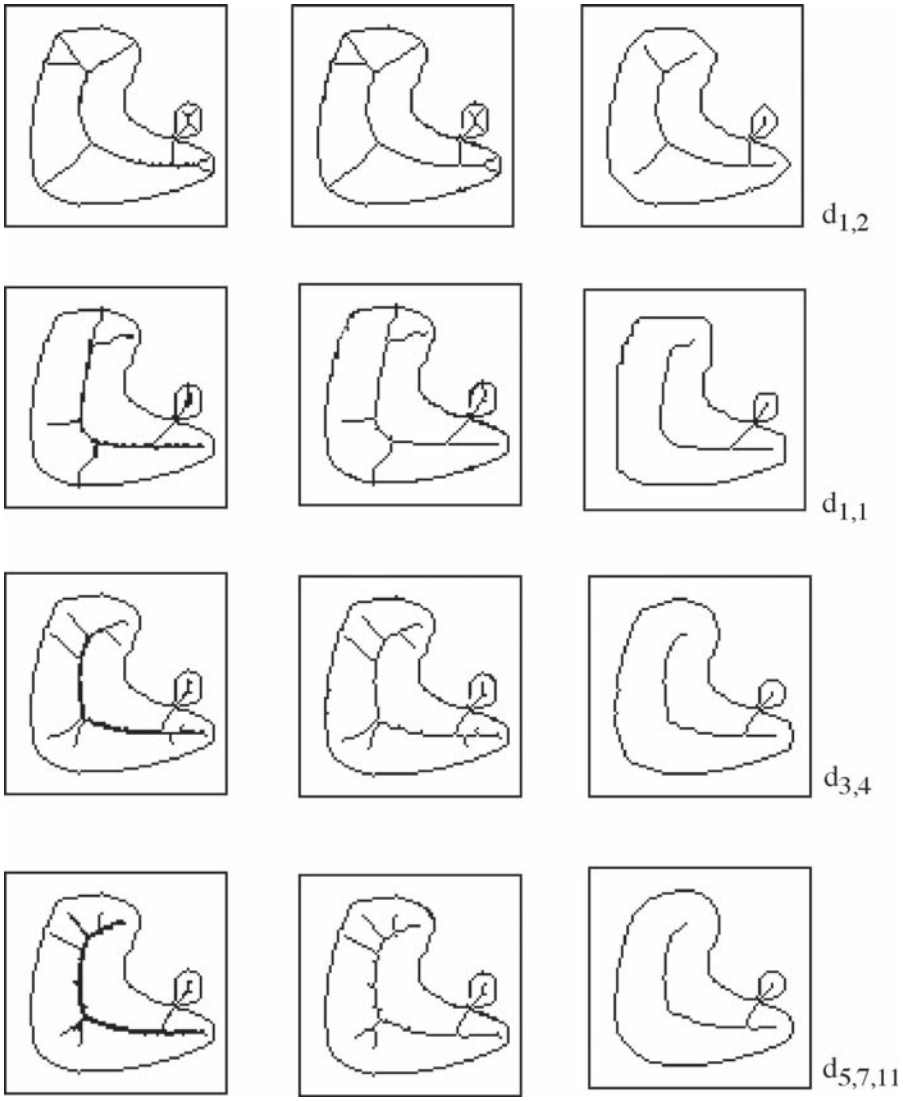
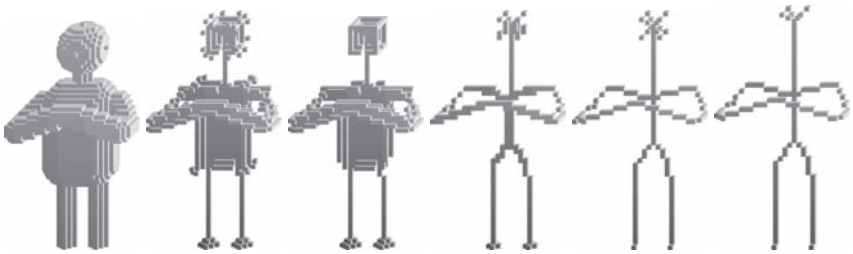


Fig. 16. From top to bottom, skeletons computed by using d(1,2), d(1,1), d(3,4) and d(5,7,11). From left to right, the nearly thin skeleton, the unit-wide skeleton and the skeleton resulting after pruning non significant peripheral branches.

5.1 Methods based on wavefront propagation

Algorithms belonging to this category compute each level set of a continuous function defined over the paradigm of a wave that originates in one point and propagates isotropically with

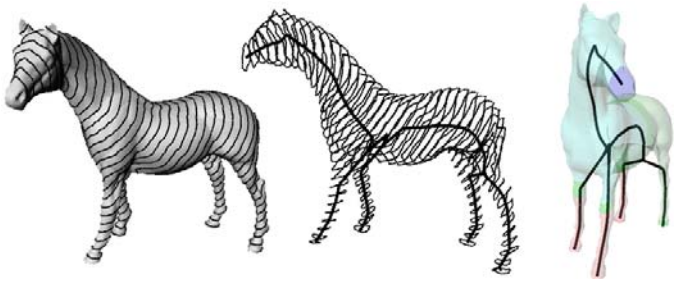


(a) (b) (c) (d) (e) (f)

Fig. 17. A 3D object, a), its surface-skeleton computed according to D6, b), the simplified surface-skeleton, c), the nearly thin curve-skeleton, d), the unit-wide curve-skeleton, e) and the pruned-curve-skeleton, f).

respect to a given function f in each direction on the surface. Points belonging to the same wave-front are characterized by the same function value by construction, and therefore define a level set of f.

The construction of the *Level Set Diagrams* from triangulated polyhedra proposed in [76] uses Euclidean distances for wave propagation [14]. In practice the wave traversal associates to each vertex of a triangle mesh the Euclidean length of the minimal path from that point and a source point. In particular, at the starting point the value of the wave traversal is zero. Each successive wave is a sub-complex and a subset of the *link* of the previous one. The wave propagation process continues until all vertices of the mesh have been selected using the Dijkstra algorithm for finding the paths of minimum length. The wave traversal may be also defined as a distance function. The seed point to start the wave propagation is automatically selected using a heuristic, which works well on elongate tubular shapes. In this case, skeletal lines obtained with different source points are very similar and the resulting skeleton is invariant under rotation, translation and scaling. Anyway, the choice of only one source point determines a privileged "slicing direction", which can lead to the loss of some features if the object is not tubular shaped (like the horse ears in Figure 18(b)).



(a) (b) (c)

Fig. 18. Isolevels (a) and the centreline (b,c) of the horse as computed as described in [76].

An extension of the approaches in [14] and [76] to non-zero genus surfaces has been presented in [72]. In this case, the evaluation of the measuring function, the mesh characterization

(based on local criteria) and the construction of the graph are performed at the same time using the Djikstra's algorithm. The independence on the object position makes this representation suitable to quadrangulate a surface. A similar extension to volume models with through holes has been presented by Wood et al. [103]; there, the graph is implicitly stored for generating high quality semi-regular and multi-resolution meshes from distance volumes. Also in this case, the object topology is achieved by considering a wavefront-like propagation from a seed point, [13] (see figure 19). The calculation of the isosurfaces is obtained by applying the Dijkstra's algorithm; this makes this approach unavailable for non-uniform scaling.

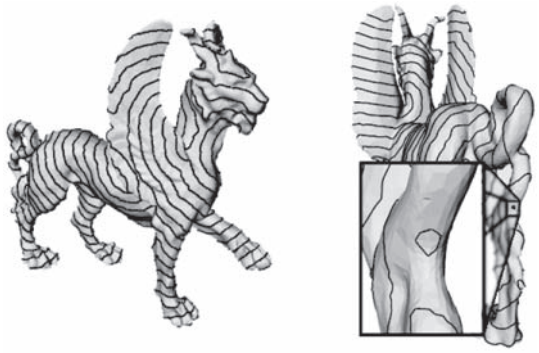
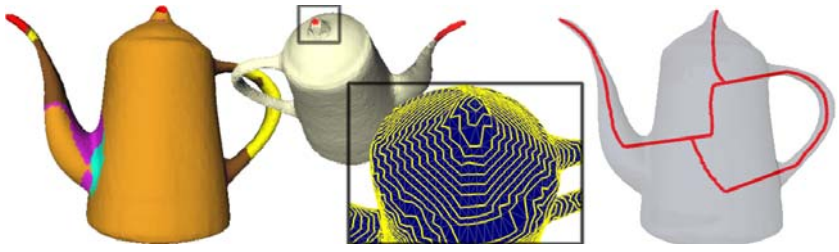


Fig. 19. Simulation of the wave-front propagation in [103].

Finally, a multi-resolution curvature evaluation is introduced in [82] to locate seed points which are sequentially linked by using the natural topological distance on the simplicial complex (see Figure 20(a,b) and also Figure CP-4(a,b) in Appendix E). More precisely, once computed the approximated Gaussian curvature for the mesh vertices, for each high curvature region R_i, $i = 1, \ldots, n$, a *representative vertex* p_i is selected.



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

Fig. 20. (a) Vertex classification based on Gaussian curvature, (b) high curvature regions are highlighted; (c) topological rings expanded from centers of high curvature regions (d) resulting skeleton.

Starting at the same time from all representative vertices, waves made of vertices of increasing neighbourhoods are computed in parallel until the whole surface is covered (see Figure 20(c) and also Figure CP-4(c) in Appendix E), in a way similar to the wave-traversal

technique [13]. Waves growing from different seed points will collide and join where two distinct protrusions depart, thus identifying a branching zone; self-intersecting waves can appear expanding near handles and through holes. A skeleton is drawn according to the wave expansion: *terminal nodes* are identified by the *representative vertices*, while union or split of topological rings give *branching nodes*. Arcs are drawn joining the center of mass of all rings (see Figure 20(d) and also Figure CP-4(d) in Appendix E). This *curvature-based graph* graph is invariant to translation, rotation and scaling. On the other hand, if the curvature evaluation process does not recognize at least one feature region, e.g. surfaces with constant curvature value as spheres, this approach is not meaningful for extracting a description of the shape. Finally, this curve-line representation has at least as many cycles as the number of holes of the surface; however, some unforeseen cycles may appear in correspondence of the wavefront collisions.

5.2 Methods based on the Reeb graph

In the general case, the *Reeb graph* [89] of a n-dimensional manifold M under a mapping function f is defined as a quotient space, which identifies the levels sets of f. More formally: let $f : M \to \mathbb{R}$ be a real valued function on a compact manifold M. Then, the Reeb graph of M with respect to f is the quotient space of $M \times \mathbb{R}$ defined by the equivalence relation "$\sim$", which states that $(P, f(P)) \sim (Q, f(Q))$ iff:

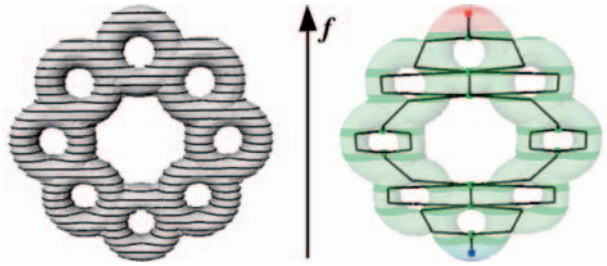
1. $f(P) = f(Q)$;
2. P, Q are in the same connected component of $f^{-1}(f(P))$.

Under the hypotheses that M is smooth and the function f is Morse and simple (i.e., its critical points have different values of f), Reeb demonstrated that the quotient space is a finite and connected simplicial complex of dimension 1, i.e., it is made of a connected collection of vertices and edges. The counter-image of each vertex is a singular connected component of the level sets of f, and the counter-image of an edge is homeomorphic to the topological product of one connected component of the level sets by $\mathbb{R}$ [89]. Under the same hypotheses, the number of cycles of the Reeb graph is an upper bound of the number of loops $\beta_1(M)$ on the manifold [39].

Even if the Reeb graph definition holds in any dimension, in this Chapter we mainly focus our attention to surfaces (bi-dimensional manifolds) embedded in $\mathbb{R}^3$. In the graph representation a node is defined for each creation, merging, split or deletion of a contour, that is, to topological changes affecting the number of connected components in the counter-image of f. Each arc joins two successive critical levels in their own component. If an arc connects two nodes, n_1 and n_2, then the topology of isolevels on M between the levels n_1 and n_2 does not change along the connected component of M joining the corresponding points [69].

From a computational point of view, a centreline representation of the abstract graph may be obtained associating to each contour its centroid; thus providing a geometric embedding of the structure. In this way, the structure roughly sketches the shape, even if some points of the skeletal structure may lye outside the shape. In addition, other geometric entities related to the contour form may be stored in each node so that the original shape may be approximately reconstructed from such a structure.

In Figure 21(a) the points drawn on the manifold represent the equivalence classes of a closed surface with respect to the height function highlighted. In Figure 21(b) the Reeb's quotient space is represented as a *traditional* graph, where the equivalence classes are grouped into arcs.



(a) (b)

Fig. 21. A surface, (a), and its Reeb graph representation with respect to the height function, (b).

From the application point of view, the properties of the Reeb graph strongly depend on those of the function f and the "best" choice for the function f depends on the application context. For a detailed overview on possible choices of the function f and their application in Computer Graphics we refer to [20].

Firstly introduced in Computer Graphics by Shinagawa et al. [94], Reeb graphs have initially been limited to Morse height functions. Methods for extracting Reeb-like graphs have been proposed in [94, 93, 98, 32, 73, 11, 85, 72, 39, 18, 102]. In this Section we focus on methods for constructing the Reeb graph representation of closed surfaces.

A first algorithm, proposed by Shinagawa et al. [93], automatically constructed the graph from surface contours generated by the height function. The extraction algorithm automatically generates the graph arcs relative to a one-to-one correspondence between cross section consisting of only one contour at first. Then the graph is completed using some heuristics based on a weight function and a priori knowledge on the surface genus. Main drawbacks of this algorithm are the need of a priori knowing the genus of the surface and the fact that this procedure is limited to contour levels of the height function [93]. In addition, since information on the shape between two consecutive cross sections is necessarily lost, the frequency of the contours of the surface is critical; therefore, a reasonable computation of the graph requires a high number of surface slices and it is time and space consuming ($O(n^2)$, where n represents the total number of vertices of the scattered contours).

The method proposed by Hilaga et al. in [73] provides a multi-resolution Reeb graph representation of triangle meshes which is independent of the object topology. The construction of the graph begins with the extraction of the graph at the finest resolution desired, then adjacency rules are used to complete the multi-resolution representation in a fine-to-coarse order. First of all, the domain of the mapping function is divided into a set of intervals. Second, triangles whose image under f lies in two intervals are subdivided so that the image of every triangle belongs to only one interval. Third, triangle sets, that is a connected component of triangles whose images belong to the same interval, are calculated. A node of the graph is associated to each triangle set. Then, arcs are detected by checking the region adjacency of triangle sets. The graph extraction is computed in $O(n + m)$ operations, where n and m represent, respectively, the number of triangles of the original mesh and those inserted during the subdivision phase. In Figure 22 an example of the Reeb graph construction method proposed in [73] is shown; in this case the domain of f is subdivided into 4 intervals. The contour insertion in 22(b) determines a set of mesh regions that correspond to the graph nodes 22(c), while their adjacency originates the arcs of the graph 22(d).

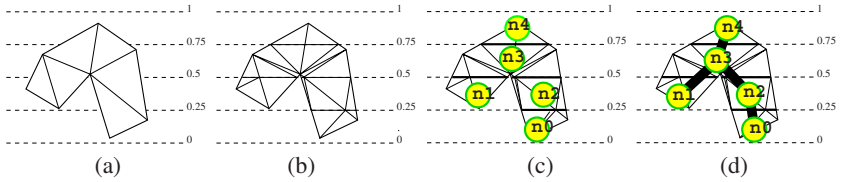
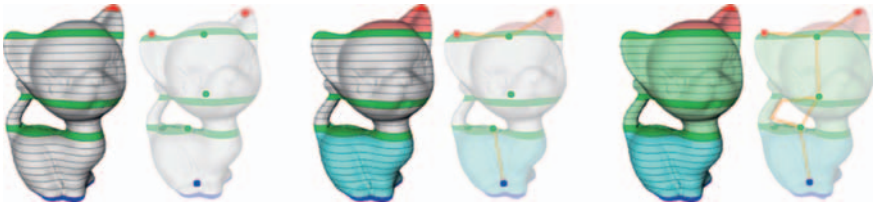


Fig. 22. Pipeline of the Reeb graph extraction in [73]. (b)

In [39] a method that performs also for non-orientable surfaces with or without boundaries, such as the Klein's bottle, has been proposed. Basic assumption of this approach is that the mapping function is Morse, thus critical points have pairwise different function values. Critical points are detected analysing the star of each vertex and non-simple critical points are simplified using the approach proposed in [53]. Once critical points have been identified, all vertices of the model are processed according the increasing value of the function f and the evolution of level sets is tracked. Since operations are done edges the complexity of the algorithm is $O(nlog(n))$, where n is the number of edges of the complex. An extension of this method has been proposed in [52] to analyse the evolution of the Reeb graph when the mapping function varies with time. In this case a point at infinity is added to make the space topologically equivalent to the 3-sphere so that the Reeb graph will be equivalent to a tree. Evolution with time of the graph is coded using a Jacobi curve that collects the birth-death points. Once a Reeb graph is computed, it is updated when an event occurs and stored in a data structure that code the entire evolution. Finally the computational cost of this approach, $O(N + En)$, depends on the number N of simplexes of the triangulation of the space-time data, the upper number n of simplexes at a time t and the amount E of birth-death and interchange events.

The approach proposed in [11, 17, 16] extracts a Reeb graph-like representation, called an *Extended Reeb graph* representation both from a surface with or without boundary through a finite set of contour levels of a given mapping function f. Since the contour levels decompose a surface S into a set of regions, the behavior of their boundaries is used TO detect critical areas and TO classify them as maximum, minimum and saddle areas. The characterization is performed by analysing the number of border components of each region and the values of the function f around them [11]. Critical areas correspond to nodes of the graph. Then arcs between nodes are constructed through an expansion process of the critical areas, in two phases: first arcs from minima/maxima to saddle areas, then the remaining links between saddle areas are inserted. In Figure 23 (see also Figure CP-5 in Appendix E) the main steps of the *ERG* extraction process are depicted; in Figure 23(a) to each critical area is associated a node; Figure 23(b) represents how the maximum (resp. the minimum) is connected to another critical area and the corresponding partial graph representation; finally, Figure 23(c) shows how the expansion process continues until the graph is completed.

On the basis of the ERG representation, a further extension of the domain of the Reeb graph to unorganized point clouds of 3D scan data has been proposed in [100]. The assumption on the point clouds is that they represent a human body. The limitation that the original data are not organized in a polygonal mesh is overcome assuming that the Euclidean distance among a point p and its closest point q, is smaller than a given threshold ϵ, $d(p, q) < \epsilon$. Point sets whose sampling is sufficiently fine are connected in a discrete sense. Therefore, level sets are defined as points that share the same value of the mapping function and are connected in the discrete sense. The resulting graph is called the *Discrete Reeb Graph (DRG)*.



(a) (b) (c)

Fig. 23. The recognition of the critical areas (a), the expansion of maxima and minima (b) and the complete graph. This model comes from the AIM@SHAPE repository: http://shapes.aim-at-shape.net/.

Finally, the method proposed in [102] has been proposed for topologically simplifying and repairing regularly sampled $3D$ grids of scalar values. That is a volumetric model in which each grid cube has 8 neighbor grid points. In this case, the data are swept with a set of parallel planes generating a set of *slices*, which are the sets of grid cubes bounded by two adjacent isosurfaces. Each connected component of a slice is called *ribbon*. The *contours* are given by the intersection of the isosurfaces the slicing plane. In particular, the graph described in this approach is called *augmented* Reeb graph because it codes also geometric information for each contour and each ribbon. Contour nodes in the graph correspond to a distance function traversal of the surface, and cycles, in addition to the geometric information stored in the ribbons, correspond to handles. The traversal is analyzed at discrete z intervals of the volumetric grid along the boundary of a distance function. Therefore, the planar slices are used as an ordered traversal through the slices an may be processed out-of-core of the volume data. Both ribbons and contours correspond to nodes of the Reeb graph while their adjacency is coded in the edges. To avoid that a handle is completely contained within a ribbon, the Euler characteristic of each isosurface component is computed and the sweep is locally refined. In this way the topology of the volume is completely coded and, in each interval, there is the correspondence of the Reeb graph structure with its Euler characteristic.

6 Conclusions and future developments

In this Chapter we have briefly sketched a wide variety of skeletal structures defined in Computer Graphics and Computer Vision. As discussed throughout the Chapter, there is one main structure often referred to as the skeleton, the medial axis transform, and a huge quantity of very similar skeletons that exhibit some (often very small) modifications. Being the MAT unfortunately hard to be computed in the general case and unstable to small perturbations of the shape, a large number of variations of the MAT were introduced: some of them are just approximations of the MAT for facilitating the skeleton computation (e.g. MAT computation through Voronoi diagrams), while others come from different definitions and present different properties. A few descriptors are able to represent the exact medial axis for a small category of input shapes, like the bisectors of parametric curves and surfaces.

In the 3D case the distinction between the MAT and others skeletal structures becomes more evident: in fact, while the MAT in 3D is essentially a medial surface, in many applications a linear skeleton may be preferable. This is the case of path planning for medical applications in which a linear skeleton, as far as possible from the shape boundary, is needed,

maybe to plan the inspection of a human organ [99]. The definitions and properties of these linear 3D skeletons depend mostly on the input data type: for discrete representations like collection of voxels, distance maps and thinning techniques are used; wave-front propagation and level set approaches are preferred in the continuum case.

With reference to the properties that should characterize a descriptor, we highlight that all the skeletal structure described in this Chapter provide a *dimensional reduction* of the original representation. From the storage point of view the MAT (and the shock graph) gives a representation which is also *invertible*, thus paying in terms of spatial and computational costs. Bisectors (that may be seen as an over set of the medial axis) provide the most complete information among the structures considered in this work, but, in practice, their effective computation is limited to a few classes of models. *Linear* structures, like centrelines provided by thinning or distance maps computations, provide a very *compact* and *concise* representation of the shape, even in case of 3D data; unfortunately loosing the *invertibility* property. Nevertheless, as all medial representations, discrete centrelines satisfy the property of being always inside (*centrality*) the shape, which made them popular in applications related to shape animation, deformation and retrieval.

Other structures that are defined on the basis of a real function, like Reeb graphs, capture the topology of the shape. They can always be represented as graphs, eventually enriched with additional geometric attributes, but they are only able to approximate the original dataset. Depending on the application context, the flexibility in the choice of the functions makes these descriptors tunable to different application domains. In particular, there is a growing interest on the definition of functions that do not depend on the (geometric) embedding of the shape, like the so-called Laplacian eigenvalues [50].

In every application context, the choice of the most suitable skeleton must cope with efficiency and expressive power of the representation, and we tried to underline these aspects in the Chapter. Provided that a function, which is independent of object rigid transformations, is suitable for recognition tasks, the Reeb graph it could be anyway preferable to the MAT, which is usually complex (in terms of number of nodes and edges) and is unstable to small perturbations of the shape, thus giving very different skeletons also for similar shapes. Therefore, the relatively simple but topologically effective structure of these descriptions has suggested a large use of them in shape matching and retrieval tasks.

About the *stability* of the representation we observe that bisectors, medial axes and shock graphs intrinsically depend on small shape modifications. Nevertheless, as discussed in Section 4.1, this problem has been partially overcome with pruning strategies that are stable under small shape perturbations. Discrete centrelines derived from thinning or distance maps are usually *robust* to small shape variations.

As far as computational issues are concerned, in table 3 we briefly summarize the complexity of the algorithms described in the Chapter.

In particular, we point out that the complexity of the bisectors may be expressed in terms of degree of the parametric representation. In particular, given two polynomial parametric curves in the plane of degree m, the bisector curve is represented as an implicit function of degree $4m - 2$. For example, for two cubic curves ($m = 3$), the bisector is represented as

[1] For point sets with provable small Voronoi graph, F is reasonably small: $F = O(n \log n)$ [9] or $F = O(n)$ [8].

[2] Once a set of seed points has been recognized, the complexity of the skeleton extraction is linear in the number of mesh vertices but an accurate evaluation of the high curvature points has quadratic cost.

Summary		
Approach	**Description**	**Costs**
[33]	Voronoi graph	$O(n^{\lceil \frac{k}{2} \rceil}) + n \log n$
[34]	Voronoi graph	$O((n+F)\log^2 F)$[1]
[77]	Medial axis of a polygon	$O(n \log n$
	Discrete skeleton 2D images	$O(n^2)$
	Discrete skeleton 3D images	$O(n^3)$
[1]	Straight skeleton	$O(nr \log n)$
[60]	Straight skeleton	$O(n^{1+\epsilon} + n^{8/11+\epsilon} r^{9/11+\epsilon})$
[37]	Straight skeleton	$O(n \log^2 n + r\sqrt{r} \log r)$
[97]	Linear Axis	$O(n)$
[76, 72]	Centerline	$O(n \log(n))$
[103]	Centerline	$O(n \log(n))$
[81][2]	Centerline	$O(n)$
[93, 94]	Reeb graph	$O(n^2)$
[73]	Reeb graph	$O((n+m))$
[11, 16]	Reeb graph	$O(\max(m+n, n \log(n)))$
[39]	Reeb graph	$O(n \log(n))$
[102]	Reeb graph	$O(n \log(n))$

Table 3. Classification of the methods for skeleton extraction. Symbols: n represent the number of vertices or points or pixels (voxels); m the number of vertices inserted in the mesh during an eventual contouring phase; e the number of edges in the neighborhood tree, r is the number of reflex vertices.

an implicit of degree 10. Hence, the Voronoi cells and diagrams of cubic curves could be represented as subregions of degree 10 implicits.

To conclude, we would like to emphasize that skeletal structures will play a fundamental role in the development of specific tools for the (future) semantic annotation of shapes, or shape parts, according to the concepts formalised by a domain ontology. In fact, the computation of a skeleton and the extraction of features automatically provide a way for decomposing a shape into significant parts, which may be further analysed and annotated.

References

1. O. Aichholzer, D. Alberts, F. Aurenhammer, and B. Gartner. A novel type of skeleton for polygons. *Journal of Universal Computer Science*, 1:752–761, 1995.
2. N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *Discrete and Computational Geometry*, 22:481–504, 1999.
3. N. Amenta, S. Choi, and R.K. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19:127–153, 2001.
4. C. Arcelli and G. Sanniti di Baja. Euclidean skeleton via center-of-maximal-disc extraction. *Image and Vision Computing*, 11:163–173, 1993.
5. D. Attali. *Squelettes et graphes de Voronoi 2-D et 3-D*. PhD thesis, University Joseph Fourier, 1995.

6. D. Attali, J.-D. Boissonnat, and H. Edelsbrunner. Stability and computation of medial axes — A state-of-the-art report. In T. Möller, B. Hamann, and B. Russell, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*. Springer-Verlag, 2005. To appear.
7. D. Attali and J.D. Boissonnat. Complexity of the Delaunay triangulation of points on polyhedral surfaces. *Discrete and Computational Geometry*, 30(3):437–452, 2003.
8. D. Attali and J.D. Boissonnat. A linear bound on the complexity of the Delaunay triangulation of points on polyhedral surfaces. *Discrete and Computational Geometry*, 31:369–384, 2004.
9. D. Attali, J.D. Boissonnat, and A. Lieutier. Complexity of the Delaunay triangulation of points on surfaces: the smooth case. In *SCG '03: Proc. of the 19th Annual Symposium on Computational Geometry 2003*, pages 201–210. ACM Press, 2003.
10. D. Attali and A. Montanvert. Modeling noise for a better simplification of skeletons. In *ICIP '96: Proc. of the International Conference on Image Processing*, volume 3, pages 13–16, 1996.
11. M. Attene, S. Biasotti, and M. Spagnuolo. Shape understanding by contour-driven retiling. *The Visual Computer*, 19(2-3):127–138, 2003.
12. F. Aurenhammer and H. Imai. Geometric relations among Voronoi diagrams. *Geometria Dedicata*, 27:65–75, 1988.
13. U. Axen and H. Edelsbrunner. Auditory Morse analysis of triangulated manifolds. In *Mathematical Visualization*, pages 223–236. Springer-Verlag, 1998.
14. Ulrike Axen. Computing Morse functions on triangulated manifolds. In *SODA '99: Proc. of the 10th ACM-SIAM Symposium on Discrete Algoritms 1999*, pages 850–851. ACM Press, 1999.
15. M. Bern, D. Eppstein, P.K. Agarwal, N. Amenta, P. Chew, T. Dey, D.P. Dobkin, H. Edelsbrunner, C. Grimm, L.J. Guibas, J. Harer, J. Hass, A. Hicks, C.K. Johnson, G. Lerman, D. Letscher, P. Plassmann, E. Sedgwick, J. Snoeyink, J. Weeks, C. Yap, and D. Zorin. Emerging challenges in computational topology. In Report from the NSF-funded Workshop on Computational Topology, 1999.
16. S. Biasotti. *Computational Topology Methods for Shape Modelling Applications*. PhD thesis, Università degli Studi di Genova, May 2004.
17. S. Biasotti. Reeb graph representation of surfaces with boundary. In *SMI '04: Proc. of Shape Modeling Applications 2004*, pages 371–374, Los Alamitos, Jun 2004. IEEE Computer Society.
18. S. Biasotti, B. Falcidieno, and M. Spagnuolo. Surface shape understanding based on extended Reeb graphs. In *Topological Data Structures for Surfaces: An Introduction for Geographical Information Science*, pages 87–103. John Wiley and Sons, 2004.
19. S. Biasotti, L. De Floriani, B. Falcidieno, and L. Papaleo. Morphological representations of scalar fields. In *Shape Analysis and Structuring*. Springer, 2007.
20. S. Biasotti, S. Marini, M. Mortara, and G. Patane. An overview on properties and efficacy of topological skeletons in shape modelling. In M.S. Kim, editor, *SMI '03: Proc. of Shape Modeling International 2003*, pages 245–254, Los Alamitos, May 2003. IEEE Computer Society.
21. H. Blum. A transformation for extracting new descriptors of shape. In Weiant Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form. Proc. of a Symposium*, pages 362–380, Cambridge MA, Nov 1967. MIT Press.
22. J.D. Boissonnat and F. Cazals. Natural neighbor coordinates of points on a surface. *Computational Geometry-Theory and Applications*, 19(2-3):155–173, 2001.

23. J.D. Boissonnat and M. Karavelas. On the combinatorial complexity of Euclidean Voronoi cells and convex hulls of d-dimensional spheres. In *SODA '03: Proc. of the 14th ACM-SIAM Symposium on Discrete Algoritms*, pages 305–312. ACM Press, 2003.
24. G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.
25. G. Borgefors. On digital distance transform in three dimensions. *Computer Vision and Image Understanding*, 64(3):368–376, 1996.
26. G. Borgefors, I. Nyström, G. Sanniti di Baja, and S. Svensson. Simplification of 3D skeletons using distance information. In L. J. Latecki, R. A. Melter, D. M. Mount, and A.Y. Wu, editors, *Proc. of SPIE - Vision Geometry IX*, volume 4117, pages 300–309, San Diego - USA, 2000.
27. J.W. Brandt. Convergence and continuity criteria for discrete approximations of the continuous planar skeletons. *CVGIP: Image Understanding*, 59:116–124, 1994.
28. J.W. Brandt and V.R. Algazi. Continuous skeleton computation by Voronoi diagram. *CVGIP: Image Understanding*, 55:329–337, 1992.
29. C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections*. Ph.D thesis, Universität des Saarlandes, March 1996.
30. C.Arcelli and G. Sanniti di Baja. A thinning algorithm based on prominence detection. *Pattern Recognition*, 13(3):225–235, 1981.
31. C.Arcelli and G. Sanniti di Baja. Skeletons of planar patterns. In T.Y. Kong and A. Rosenfeld, editors, *Topological Algorithms for Digital Image Processing*, pages 99–143. North-Holland, 1996.
32. H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. In *SODA '00: Proc. of the 11th ACM-SIAM Symposium on Discrete Algoritms 2000*, pages 918–926. ACM Press, 2000.
33. *The CGAL 3.1 User Manual*.
34. T.M. Chan, J. Snoeyink, and C.K. Yap. Primal dividing and dual pruning: Output-sensitive construction of 4-D polytopes and 3-D Voronoi diagrams. *Discrete and Computational Geometry*, 18:433–454, 1997.
35. F. Chazal and A. Lieutier. Stability and homotopy of a subset of the medial axis. In *SMA '04: Proc. of the 9th ACM Symposium on Solid Modeling and Applications 2004*, pages 243–248. ACM Press, 2004.
36. F. Chazal and R. Soufflet. Stability and finiteness properties of medial axis and skeleton. *Journal on Control Dynamics and Systems*, 10:149–170, 2004.
37. S.W. Cheng and A. Vigneron. Motorcycle graphs and straight skeletons. In *SODA '02: Proc. of the 13th ACM-SIAM Symposium on Discrete Algoritms 2002*, pages 156–165. ACM Press, 2002.
38. S.W. Choi and H.P. Seidel. Linear one-sided stability of MAT for weakly injective 3D domain. In *SMA '02: Proc. of the 7th ACM Symposium on Solid Modeling and Applications 2002*, pages 344–355. ACM Press, 2002.
39. K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in Reeb graphs of 2-manifolds. In *SCG '03: Proc. of the 19th Annual Symposium on Computational Geometry 2003*, pages 344–350. ACM Press, 2003.
40. N. D. Cornea, D. Silver, and P. Min. Curve-skeleton applications. In *Proceedings IEEE Visualization*, pages 95–102, 2005.
41. T. Culver. *Computing the medial axis of a polyhedron reliably and efficiently*. PhD thesis, University North Carolina, Chapel Hill, North Carolina, 2000.
42. T. Culver, J. Keyser, and D. Manocha. Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design*, 21(1):65–98, 2004.

43. T. Dey and j. Sun. Defining and computing curve-skeletons with medial geodesic function. In *Proceedings of the Symposium on Geometry Processing*, pages 143–152, 2006.
44. T.K. Dey and W. Zhao. Approximating the medial axis from the Voronoi diagram with a convergence guarantee. *Algorithmica*, 38, 2004.
45. G. Sanniti di Baja. Well-shaped, stable and reversible skeletons from the (3,4)-distance transform. *Visual Communication and Image Representation*, 5:107–115, 1994.
46. G. Sanniti di Baja. Representing shape by line patterns. In P. Wang and A. Rosenfeld, editors, *Advances in Structural and Syntactical Pattern Recognition*, volume 1121 of *Lecture Notes in Computer Science*, pages 230–239. Springer-Verlag, 1996.
47. G. Sanniti di Baja and I. Nyström. Skeletonization in 3D discrete binary images. In C.H. Chen and P.S.P. Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, Chapter 2.2, pages 137–156. World Scientific, Singapore, 3rd edition, January 2005.
48. G. Sanniti di Baja and E. Thiel. Skeletonization algorithm running on path-based distance maps. *Image and Vision Computing*, 14:47–57, 1997.
49. G.Sanniti di Baja and S. Svensson. Surface skeletons detected on the D6 distance transform. In F.J. Ferri et al., editor, *Proc. of SSSPR'2000 - Advances in Pattern Recognition*, volume 1121, pages 387–396, Alicante, 2000. LNCS, Springer-Verlag.
50. S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. Hart. Spectral surface quadrangulation. *ACM Transactions on Graphics*, 25(3):1057–1066, August 2006.
51. D. Dutta and C. Hoffmann. On the skeleton of simple CSG objects. *ASME J. of Mechanical Design*, 115:87–94, 1993.
52. H. Edelsbrunner, J. Harer, A. Mascarenhas, and V. Pascucci. Time-varying Reeb graphs for continuous space-time data. In *Proceeding of the 20-th ACM Symposium on Computational Geometry*, pages 366–372, 2004.
53. H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30:87–107, 2003.
54. G. Elber and M. S. Kim. Bisector curves of planar rational curves. *Computer Aided Design*, 30(14):1089–1096, December 1998.
55. G. Elber and M. S. Kim. The bisector surface of freeform rational space curves. *ACM Trans. on Graphics*, 17(1):32–50, January 1998.
56. G. Elber and M. S. Kim. Computing rational bisectors. *Computer Graphics and Applications*, 19(6):76–81, November-December 1999.
57. G. Elber and M. S. Kim. Rational bisectors of CSG primitives. In *The Fifth ACM/IEEE Symposium on Solid Modeling and Applications, Ann Arbor, Michigan*, pages 159–166, June 1999.
58. G. Elber and M. S. Kim. A computational model for nonrational bisector surfaces: Curve-surface and surface-surface bisectors. In *Geometric Modeling and Processing 2000, Hong Kong*, pages 364–372, April 2000.
59. G. Elber and M. S. Kim. Geometric constraint solver using multivariate rational spline functions. In *The Sixth ACM/IEEE Symposium on Solid Modeling and Applications, Ann Arbor, Michigan*, pages 1–10, June 2001.
60. D. Eppstein and J. Erickson. Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. *Discrete and Computational Geometry*, 22:569–592, 1999.
61. R. Farouki and J. Johnstone. The bisector of a point and a plane parametric curve. *Computer Aided Geometric Design*, 11(2):117–151, April 1994.
62. R. Farouki and R. Ramamurthy. Specified-precision computation of curve/curve bisectors. *Int. J. of Computational Geometry & Applications*, 8(5-6):599–617, October-December 1998.

63. R. Farouki and R. Ramamurthy. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries I. theoretical foundations. *J. of Computational and Applied Mathematics*, 102:119–141, 1999.
64. R. Farouki and R. Ramamurthy. Voronoi diagram and medial axis algorithm for planar domains with curved boundaries II. detailed algorithm description. *J. of Computational and Applied Mathematics*, 102:119–141, 1999.
65. S. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.
66. P. Giblin and B.B. Kimia. A formal classification of 3D medial axis points and their local geometry. In *CVPR 2000: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2000*, volume 1, pages 566–573, Los Alamitos, 2000. IEEE Computer Society.
67. P.J. Giblin and B.B. Kimia. On the local form and transitions of symmetry sets, medial axes, and shocks. *International Journal of Computer Vision*, 54:143–157, 2003.
68. S. Goswami, T. K. Dey, and C. L. Bajaj. Identifying flat and tubular regions of a shape by unstable manifolds. In *SPM '06: Proceedings of the 2006 ACM symposium on Solid and physical modeling*, pages 27–37, New York, NY, USA, 2006. ACM Press.
69. A. Gramain. *Topologie des surfaces*. Presses Universitaires de France, 1971.
70. I. Hanniel, R. Muthuganapathy, G. Elber, and M. S. Kim. Precise Voronoi cell extraction of free-form rational planar closed curves. In *ACM Symposium on Solid and Physical Modeling*, pages 51–59, June 2005.
71. M. Held. Vroni: An engineering approach to the reliable and efficient computation of Voronoi diagrams of points and line segments. *Computational Geometry: Theory and Applications*, 18:95–123, 2001.
72. F. Hetroy and D. Attali. Topological quadrangulations of closed triangulated surfaces using the Reeb graph. *Graphical Models*, 65(1-3):131–148, 2003.
73. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes Los Angeles, CA. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 203–212, Aug 2001.
74. D.S. Kim, Y. Cho, and D. Kim. Euclidean Voronoi diagram of 3D balls and its computation via tracing edges. In *Computer Aided Design*, pages 1412–1424, November 2005.
75. B. Kimia, A. Tannenbaum, and S. Zucker. Shapes, shocks, and deformations, I: The components of shape and the reaction-diffusion space. *International Journal of Computer Vision*, 15:189–224, 1995.
76. F. Lazarus and A. Verroust. Level set diagrams of polyhedral objects. In W.F. Bronsvoort and D.C. Anderson, editors, *SMA '99: Proc. of the 5th ACM Symposium on Solid Modeling and Applications 1999*, pages 130–140. ACM Press, 1999.
77. D. T. Lee. Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(4):363–369, 1982.
78. A. Lieutier. Any open bounded subset of $\mathbb{R}^n$ has the same homotopy type as its medial axis. In *Proc. 8th ACM Sympos. Solid Modeling Appl.*, pages 65–75. ACM Press, 2003.
79. G. Matheron. Examples of topological properties of skeletons. In *Image Analysis and Mathematical Morphology, Volume 2: Theoretical Advances*, pages 217–238. Academic Press, 1988.
80. J. Milnor. *Morse Theory*. Princeton University Press, New Jersey, 1963.
81. M. Mortara and G. Patané. Shape-covering for skeleton extraction. *International Journal of Shape Modelling*, 8:245–252, 2002.

82. M. Mortara, G. Patane, M. Spagnuolo, B. Falcidieno, and J. Rossignac. Blowing bubbles for multi-scale analysis and decomposition of triangle meshes. *Algorithmica*, 38(1):227–248, 2004.
83. R.L. Ogniewicz. Skeleton-space: A multi-scale shape description combining region and boundary information. In *CVPR '94: Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1994*, pages 746–751, Los Alamitos, 1994. IEEE Computer Society.
84. R.L. Ogniewicz and O. Kubler. Hierarchic Voronoi skeletons. *Pattern Recognition*, 28:343–359, 1995.
85. V. Pascucci and K. Cole-McLaughlin. Parallel computation of the topology of level sets. *Algorithmica*, 38:249–268, 2003.
86. M. Peternell. Geometric properties of bisector surfaces. In *Graphical Models and Image Processing*, 2000.
87. M. Ramanathan and B. Gurumoorthy. Constructing medial axis transform of planar domains with curved boundaries. *Computer Aided Design*, 35:619–632, 2002.
88. M. Ramanathan and B. Gurumoorthy. Constructing medial axis transform of extruded and revolved 3D objects with free-form boundaries. *Computer-Aided Design*, 37(13):1370–1387, 2005.
89. G. Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences*, 222:847–849, 1946.
90. A. Rosenfeld. Digital geometry: Introduction and bibliography. In *Advances in Digital and Computational Geometry*, 1998.
91. E. Sherbrooke, N. M. Patrikalakis, and F.-E. Wolter. Differential and topological properties of medial axis transforms. *Graphical Models and Image Processing*, 58:574–592, 1996.
92. E.C. Sherbrooke, N.M. Patrikalakis, and E. Brisson. An algorithm for the medial axis transform of 3D polyhedral solids. *IEEE Trans. on Visualization and Computer Graphics*, 22:44–61, 1996.
93. Y. Shinagawa and T.L. Kunii. Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics and Applications*, 11:44–51, 1991.
94. Y. Shinagawa, T.L. Kunii, and Y.L. Kergosien. Surface coding based on Morse theory. *IEEE Computer Graphics and Applications*, 11:66–78, 1991.
95. S. Svensson and G. Sanniti di Baja. Simplifying curve skeletons in volume images. *Computer Vision and Image Understanding*, 90:242–257, 2003.
96. S. Svensson, I. Nystr om, and G. Sanniti di Baja. Curve skeletonization of surface-like objects in 3D images guided by voxel classification. *Pattern Recognition Letters*, 23(12):1419–1426, 2002.
97. M. Tanase and R. C. Veltkamp. A straight skeleton approximating the medial axis. *Lecture Notes in Computer Science*, 3221:809–821, Sep 2004.
98. M. van Kreveld, R. Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface transversal. In *SCG '97: Proc. of the 13th Annual Symposium on Computational Geometry 1997*, pages 212–220. ACM Press, Jun 1997.
99. M. Wan, Z. Liang, Q. Ke, L. Hong, I. Bitter, and A. Kaufman. Automatic centerline extraction for virtual colonoscopy. *IEEE Trans. on Medical Imaging*, 21(12):1450–1460, December 2002.
100. N. Werghi, Y. Xiao, and J. P. Siebert. A functional-based segmentation of human body scans in arbitrary postures. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 36(1):153–165, 2006.

101. F.E. Wolter. Cut locus & medial axis in global shape interrogation & representation. Technical Report Design Laboratory Memorandum 92-2, MIT, 1992.
102. Z. Wood, H. Hoppe, M. Desbrun, and P. Schroeder. Removing excess topology from isosurfaces. *ACM Trans. on Graphics*, 23:190–208, 2004.
103. Z.J. Wood, M. Desbrun, P. Schroder, and D. Breen. Semi-regular mesh extraction from volumes. In *VIS 2000: Proc. of IEEE Conference on Visualization 2000*, pages 275–282, Los Alamitos, 2000. IEEE Computer Society.