

# Curve-Skeleton Applications

Nicu D. Cornea<sup>1</sup>, Deborah Silver<sup>1</sup>  
Rutgers University, New Jersey, USA

Patrick Min<sup>2</sup>  
John Cabot University, Rome, Italy

## ABSTRACT

Curve-skeletons are a 1D subset of the medial surface of a 3D object and are useful for many visualization tasks including virtual navigation, reduced-model formulation, visualization improvement, mesh repair, animation, etc. There are many algorithms in the literature describing extraction methodologies for different applications; however, it is unclear how general and robust they are. In this paper, we provide an overview of many curve-skeleton applications and compile a set of desired properties of such representations. We also give a taxonomy of methods and analyze the advantages and drawbacks of each class of algorithms.

CR Categories and Subject Descriptors: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling -- Curve, surface, solid, and object representations; I.4.10 [Image Processing and Computer Vision]: Image Representation

Additional Keywords: skeleton, curve-skeleton.

## 1 INTRODUCTION

3D models are common in many disciplines ranging from computer aided design, medical imaging, computer graphics, scientific visualization, computational fluid dynamics, and remote sensing. While the 3D representation is invaluable, many applications require alternate “compact” representations of these models. One such representation is a line-like or stick-like 1D representation which is sometimes referred to as a “skeletal representation” or “curve-skeleton” [84]. This is different from the skeletal-surface representation (medial surface). This type of representation captures the essential topology of the underlying object in an easy to understand and very compact form. Examples of applications which use a curve-skeleton include virtual navigation, registration, animation, morphing, scientific analysis, recognition, and retrieval.

One of the difficulties is that a “curve-skeleton” is an ill-defined object. This has led to a large number of algorithms and heuristics in the literature and many more constantly being proposed. On most of the visualization discussion boards, there are periodic requests for such techniques. Many of the algorithms in the literature use different definitions, parameters and thresholds and test the algorithm on a limited number of diverse 3D objects. Additionally, some are fine-tuned for a specific application. Understandably, many of these algorithms can not be replicated and most major visualization packages do not use them. It is hard to decide which algorithm to implement since there are no criteria for evaluation, thereby causing a further proliferation of new algorithms. What is needed is an analysis of the desired properties of the curve-skeleton as required by the various applications and an overview of the existing methods.

In this paper, we compiled a list of properties for curve-skeletons based upon numerous applications. We also categorized many of the existing algorithms into classes based upon implementation, and we discuss how these classes achieve the various properties. In addition, one algorithm from each class has been implemented and tested on the same set of 6 3D shapes. The results are shown in Section 6. Our goal in this paper is to provide an overview of curve-skeletonization applications and implementations to help guide visualization users and developers.

## 2 DEFINITIONS

In 2D, the medial axis of a shape is defined as the locus of the centers of maximal inscribed discs [14], in other words, in the continuous case it is the locus of points which are equidistant from at least two points on the boundary of the object. The medial axis of a 2D shape consists of a set of curves. In 3D, however, the problem becomes more complicated. Here we talk about a medial surface, defined in terms of the centers of maximal inscribed balls, which in addition to a set of curves can also contain surface patches. Figure 1 shows the medial axis for a 2D shape (a rectangle) and the medial surface of a 3D shape (a box). Note that in Figure 1(b) only the horizontal patch of the medial surface is shaded, but in fact the medial surface consists of nine different patches determined by the lines drawn in red and the edges of the box. The medial axis (2D) and the medial surface (3D) are also known as the *skeleton*.

A definition of the medial surface can be formulated as follows [45]: let  $X \subset R^3$  be a 3D object. A ball of radius  $r$  centered at  $x \in X$  is defined as  $S_r(x) = \{y \in R^3, d(x, y) \leq r\}$ , where  $d(x, y)$  is the distance between two points  $x$  and  $y$  in  $R^3$ . A ball  $S_r(x) \subset X$  is maximal if it is not completely included in any other ball included in  $X$ . The medial surface is then the set of centers of all maximal balls included in  $X$ . A more illustrative definition of the skeleton is given by the grass-fire analogy [14], where the boundary of an object made entirely of dry grass is set on fire and the skeleton consists of the loci where the fire fronts meet and quench each other. The process of obtaining a skeleton is called *skeletonization*.

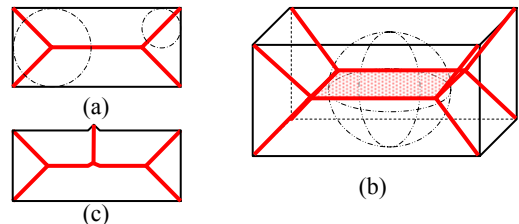


Figure 1. The medial axis in 2D (a and c) and the medial surface in 3D (b) and a few examples of inscribed discs (2D) and ball (3D).

If the medial surface is augmented at each point with the radius of the maximal ball centered at that point, a complete and accurate reconstruction of the original object is possible from the medial surface alone, by growing balls [14][32]. This property has an immediate application in shape compression and volume animation [32].

<sup>1</sup> cornea, silver @ece.rutgers.edu

<sup>2</sup> min@johncabot.it

A major disadvantage of the medial surface (axis) is its intrinsic sensitivity to small changes in the object’s surface due to the way it is defined [22]. An illustrative example in 2D is shown in Figure 1(c) where it can be observed how a small change in the object’s shape can generate a large change in the medial axis.

In many applications however, a concise representation of 3D objects with curve arcs or straight lines is desirable because of its simplicity. For example animation traditionally uses an IK (inverse-kinematics) skeleton consisting of a small number of connected line segments representing the torso, arms and legs, for example. Other applications, such as virtual navigation, also require a set of curve paths for traversal. This line-like representation of a 3D object is also known as the *centerline* or the *curve-skeleton* [84] and is a simplified 1D representation of its medial surface, consisting only of curves. Figure 2 shows curve-skeletons of several 3D objects.

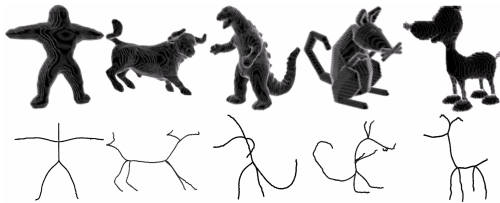


Figure 2. Examples of curve-skeletons of different 3D objects

## 2.1 The Discrete Case

The above definitions were formulated in continuous space. However, many of the applications which need skeletonization have discrete 3D datasets, such as those acquired using medical scanners. In discrete space, the definitions are analogous to the continuous case. However, problems may occur because of discretization. For example a maximal ball may touch the discrete boundary of an object in a single point. As a result, in order to include all centers of maximal balls, the discrete skeleton may be more than one image element (pixel or voxel) thick. Furthermore, resolution can cause a loss of detail for certain objects.

Some skeletonization algorithms work on geometric data (continuous), others deal with discrete objects only (distance field, thinning). In this paper, we will consider mainly the discrete case, but for the sake of completeness we will also include references to geometric methods. Geometric data can be voxelized [80], and similarly, voxelized data can be polygonized [55].

## 3 CURVE-SKELETON PROPERTIES

Below we describe the desirable properties of the curve-skeleton as required by the various applications (see Section 4): homotopic, invariant under isometric transformations, reconstruction, thin, centered, reliable, component-wise differentiation, robust, efficient to compute, and hierarchic.

For the following discussion, we will consider the discrete 3D case unless otherwise specified. We will use  $Sk(O)$  to denote the curve-skeleton of a 3D object  $O$ .

**Homotopic** (topology preserving): There are various definitions of topology preserving (see [45][52][72]). Two objects have the same topology if they have the same number of connected components, tunnels and cavities [45]. A cavity is a background (white) connected component surrounded by an object (black) component (an empty space inside the object). As pointed out in [45], the above formulation applied to two objects  $O_1$  and  $O_2$ : “object  $O_2$  preserves the topology of object  $O_1$ ” is meaningful only if an additional constraint is added: object  $O_2$  is obtained from  $O_1$  by only removing object voxels (no adding). Otherwise, object  $O_2$  could end-up having a completely new configuration,

but still have the same topology (for example  $O_2$  may grow limbs where  $O_1$  did not have them). With this observation, the above definition of topology preserving is meaningful in the context of skeletonization, where the skeleton  $S$  is a subset of the original object  $O$ . Algorithms to compute the number of connected components, tunnels and cavities in an object are given in [72] and [85].

Of course, we cannot have cavities in a 1D curve, so in a strict sense, a curve-skeleton cannot preserve the topology of an object with cavities. To accommodate objects with cavities, we propose a relaxed definition of topology preserving: the curve-skeleton should have at least one loop around each cavity of the original object. Think of a hollow sphere: the curve-skeleton can be just a circle – a single loop – or many circles in different orientations but all surrounding the same cavity. The latter version may better convey the true shape of the original object. However, tunnels in the original object also create loops in the curve-skeleton. Thus, we will reformulate the relaxed definition as follows: the curve-skeleton  $S$  preserves the topology of the original object  $O$  if it has the same number of connected components and at least as many loops as tunnels and cavities in the original object. Of course if the object does not have any tunnels or cavities, the skeleton should have no loops at all. The number of loops in a curve-skeleton can be determined by performing a depth-first search on the skeleton.

An important issue in topology preservation is the definition of connectivity. Two adjacent voxels  $x$  and  $y$  are *6-connected* if they share a face, *18-connected* if they share a face or an edge and *26-connected* if they share a face, an edge or a vertex. An *n-path* is a sequence of voxels  $x_1 \dots x_k$  with  $x_i$  n-adjacent to  $x_{i+1}$ , where  $n$  is 6, 18 or 26. A *n-connected component* is then a set of voxels such that any two such voxels are connected by an n-path included in that component. In order to avoid topological paradoxes such as objects being both connected and disconnected [45], different connectivities must be chosen for the object and for the background. Common choices are (26,6) and (18, 6), where the first number in a pair represents the object connectivity and the second represents the background connectivity.

**Invariant under isometric transformations:** Given an isometric transformation  $T$ , the skeleton of the transformed object  $T(O)$ , denoted by  $Sk(T(O))$ , should be the same as the transformed skeleton of the original object. Formally, the invariance criterion is given by:  $T(Sk(O)) = Sk(T(O))$ .

**Reconstruction** [31][62] refers to the ability to recover the original object from the curve-skeleton. The reconstruction operation involves growing balls centered at each skeleton point (or discrete medial axis point). The radius of each ball is given by the “distance transform value” which specifies the distance to the closest point on the boundary of the object. If we denote the reconstruction operation by  $Rec(skeleton)$ , then accurate reconstruction means that  $Rec(Sk(O)) = O$ .

In general, accurate reconstruction is not possible from the curve-skeleton alone since it is only a subset of the medial surface. To test the degree of reconstruction possible from a given curve-skeleton, every point must be equipped with the distance transform value determined in the original object. Then, the difference volume  $O - Rec(Sk(O))$  will provide a quantifiable measure of the ability to reconstruct the object.

**Thin:** Curve-skeletons should be one-dimensional, that is at most one voxel thick in all directions, except at joints where the skeleton might become thicker to ensure connectivity between the different branches.

We can distinguish three types of curve-skeleton points: regular points on a 1D curve-arc that have exactly two neighbors, end-points of a curve that have exactly one neighbor, and junction points (where curves meet) which can have three or more

neighbors. The thinness property can be easily checked if the junction points are known in advance. Some curve-skeletonization methods directly identify junction points [23][50]. If junction points are not known in advance, they have to be identified with another method.

Thinness and reconstruction are two conflicting properties. Even for objects whose medial surface actually contains only curves (like tubular objects), a one-voxel thick curve-skeleton may not contain all the necessary maximal balls to accurately reconstruct the object (remember that a discrete skeleton is usually more than one voxel thick owing to the discrete nature of the object).

**Centered:** One important characteristic of a skeleton is its centeredness within the object. This is especially true for scientific applications where the “core” of an object is desired. Clearly the curve-skeleton should lie on the medial surface of the object, but this criterion alone does not guarantee centeredness. One possible way to quantify the centeredness of a curve-skeleton is to seed a number of uniformly distributed radial rays at each skeleton point and measure the distance to the boundary along each of these rays.

**Reliable:** This property is useful for virtual navigation. *Reliability* [38] refers to the property of the curve-skeleton that every boundary point is visible from at least one curve-skeleton location. In other words, for any boundary point, there exists a straight line connecting it to a curve-skeleton point that does not cross any boundary. The term reliable is used in relation to virtual endoscopy where it ensures that the interior organ surface is fully (reliably) examined by the physician performing the virtual procedure. A straight-forward algorithm to test the reliability of the curve-skeleton checks the visibility of each boundary point with a straight line to every skeleton point. Boundary points which cannot be connected without intersecting the surface are not visible. Efficient visibility computation can be done following the solutions from [38].

**Junction Detection and Component-wise Differentiation:** The curve-skeleton should be able to distinguish the different components of the original object, reflecting its part/component structure. This says that the *logical components* of the object should have a one-to-one correspondence with the logical components of the curve-skeleton (which are curve arcs). There is no rigorous definition of logical components of a 3D shape, although several attempts have been made. For example in [86], meaningful components are defined as components that can be perceptually distinguished from the remaining object. In [44], the component structure of a 2D shape is defined using a combination of substance and connection measures computed around junction points of the medial axis using “visual conductance”. As long as the curve-skeleton has identifiable joints or junction points, a partitioning of the original object can be performed to produce a one-to-one correspondence between the different components in the object and the skeleton (for use in animation or mesh decomposition).

Component-wise differentiation is different from homotopy in that the first one deals with logical perceptual components of a single connected object while the latter is concerned with geometrical connected components forming different objects.

**Connected:** This is a consequence of homotopy. If the skeleton corresponds to a single connected object, then by maintaining the topology of that original object the skeleton would have to consist of a single connected component itself.

**Robust:** As shown in Figure 1(c), the medial axis is very sensitive to noise. A desirable property of the curve-skeleton is to exhibit weak sensitivity to noise on the boundary of the object, that is, the curve-skeletons of a noise-free object and the same object with noise should be similar.

There are another three criteria that relate to the algorithm used to compute the skeleton. Clearly, the algorithm should be **efficient**, i.e. many applications need real-time computations. Because the curve-skeleton is an approximation of the complex components of an object, the skeletonization process should reflect the natural **hierarchy** of these complexities [25][43]. A hierarchical approach is useful because it can generate a set of skeletons of different complexities that could be used in many different applications. Furthermore, some algorithms can handle **point sets** (i.e., where the connectivity is not specified and there is no inside/outside information), not just a voxelized representation.

Not all properties described above are essential to all types of applications. Furthermore, some of the properties may be conflicting, such as discussed for thinness and reconstruction. As a result, various algorithms that extract curve-skeletons usually enforce only a subset of these properties, depending on the application.

#### 4 USES OF CURVE-SKELETONS IN VISUALIZATION

Since they were first introduced, curve-skeletons have found uses in many areas of image processing and visualization. Below we present some of these applications. Many others exist.

One of the first uses of the curve-skeleton was in **virtual navigation** [66][92], exploiting its centeredness property to generate collision-free paths through a scene or through an object. Given a scene composed of 3D objects, the curve-skeleton of the background gives a collision-free path through the scene. In **virtual endoscopy**, curve-skeletons are used to specify collision free paths for navigation through human organs. Traditional endoscopic methods are invasive and often uncomfortable to patients. A virtual endoscopy system can produce images similar to those obtained using the traditional technique but in a non-invasive way. After imaging, the organ is “skeletonized” and a virtual camera is translated along this skeleton path allowing the inspection of the respective organ. Clinical applications include colonoscopy [40] [now available from GE, Viatronix, Philips, Siemens], bronchoscopy [66], angiography [8] and others. A *reliable* navigation path ensures the interior organ surface can be fully examined by the physician performing the virtual procedure [38][92].

In **traditional computer graphics**, skeletons are used extensively to specify animation [1][12][60]. These skeletons are sometimes referred to as IK-skeletons and they control the polygonal representation of the character being animated. Surface polygons are attached to and manipulated through this simple stick-like figure. While most of the IK-skeletons are specified by an animator, recently, there have been methods to compute the skeleton and the “skinning” (polygon correspondence) automatically [13][53][86][91]. A simplification of the curve-skeleton can be successfully used as an IK skeleton, by replacing curve arcs with straight lines. Volumetric objects can also be animated and manipulated using the same type of paradigm [32].

Surgical planning and radiation treatment require accurate extraction (**segmentation**) and **quantification** of specific anatomical structures from CT (computed tomography), MRI (magnetic resonance imaging), MRA (magnetic resonance angiogram) or ultrasound data. This is especially true for blood vessels and nerve structures. Since these structures have a characteristic tubular shape, methods aimed specifically at extracting the centerline of such tubular objects from medical images have been developed [5][6][28][29] using field-specific knowledge (intensity variation of the blood vessels, connectivity). The centerline can also be used to aid in other image processing operations such as edge detection and segmentation [67][68]. Other uses include curved planar reformation (flattening) [41][42], detection of stenosis [63][79], aneurisms or vessel wall

calcifications [81], deforming volumes: unwinding convoluted objects to allow a more efficient inspection of the overall structure or to remove occlusion (e.g., colon straightening [78]).

A common operation in medical imaging is the **registration** of two images from the same patient taken with different modalities (MRI, CT, MRA). Registration is performed by aligning some structures that are visible in both images. One approach is to reduce the dimensionality of the problem by extracting the skeleton of the structure from both images and then aligning the skeletons [6][30][67].

Another application is **matching** of 3D objects: given a query object, the task is to find similar or identical objects in a database by using the curve-skeleton [19][24][39][82]. If the curve-skeleton can differentiate the part structure of the original object, part matching is also possible, where only parts of the objects are matched against the query. In addition to matching, it directly provides registration of the part in the whole object [24][82].

Shape metamorphosis (**morphing**) is the process of generating smooth transitions between two shapes, creating the impression that one object is being smoothly transformed into another. One of the most difficult tasks in generating a successful morph is determining the correspondences between the two shapes used to drive the interpolation process. Various trade-offs are made between allowing the user full control over the process (and turning it into a mostly manual process) and completely automating the correspondence finding algorithm. The curve-skeleton can be used in this context for its simplicity, allowing the user to quickly specify correspondences on the skeletons or enabling matching algorithms to find correspondences more efficiently. Additionally, the interpolation process can be performed directly on the skeleton [11][47][95].

**Decomposing a polygonal mesh into components** is desirable for applications which treat objects as a sum of components. Such a decomposition can be assisted by using the curve-skeleton if it has the ability to distinguish the components of the original object [20][51]. In [43] an inverse approach is taken, where a 1D skeleton is extracted using the mesh decomposition results. Related geometric uses of skeletons include **surface reconstruction** [3][89] and **mesh repair** [50].

In [83], the curve-skeleton is used to define a “skeletal dimensional reduction” for the CAD field. It is also shown how such a representation can be used to reduce boundary value problems over complex solids to lower-dimensional problems over the skeleton. Skeletons have also been used to improve the efficiency of collision detection of volumetric objects [33] or in surgical simulations [93] and as a general data structure for graphical objects [69]. One of the biggest uses of skeletons is in **analysis of scientific data** where complex topologies can be easily explained using line-like drawings. Furthermore, skeletons can be used for reduced modeling and to explain simple physical phenomena. Examples include **plume visualization** [74], **vortex core extraction** [7], **feature tracking** [90] and many others.

The previous discussion is by no means exhaustive, but gives a sample of popular uses of skeletons in visualization. There are many other examples as well. Some applications have extra data available to help in the curve-skeletonization process such as velocity fields in the case of vortex core extraction [7] or blood flow data in the case of vessel tracking (e.g., [5][6][28]), while others use only the 3D object. In this paper, we concentrate on the more general problem where extra information is not available.

## 5 ALGORITHM CLASSES

There are many different skeletonization algorithms for both 2D and 3D. Although some of the 2D algorithms reportedly scale to 3D, we restrict our discussion to algorithms explicitly designed for 3D. The discussion below reviews general 3D curve-

skeletonization algorithms, i.e. the generation of a 1D curve-like representation from a 3D object. However, for completeness we do include some medial surface algorithms since these medial surfaces could be further reduced to a curve-skeleton [84]. Unless otherwise stated, we consider the 3D objects to be represented by voxels on a regular grid.

A commonly used classification scheme present in the literature divides the skeletonization algorithms into the following classes [57][87]: topological thinning (grassfire propagation), distance transform based (ridge detection) and Voronoi diagram based. However, many of the surveyed methods that produce curve-skeletons use pieces from several classes listed above to obtain a curve-skeleton. For example, there are thinning algorithms which use the distance field information to determine the thinning order, or some distance field methods which use thinning to prune the skeleton. Instead, we categorize the algorithms based on the underlying implementation into the following classes: (1) pure thinning and boundary propagation (2) distance field based (3) geometric and (4) general-field functions.

### 5.1 Thinning and Boundary Propagation

Thinning methods attempt to produce a curve-skeleton by iteratively removing simple points from the boundary of the object. A simple point [9][45][46] is an object point which can be removed without changing the topology of the object (see [45] for a complete review of digital topology). The process starts from the object’s boundary and continues inward until no more simple points can be removed. At every iteration, each boundary voxel is tested against a set of topology preserving conditions and possibly removed. The conditions are usually implemented as *templates* (or *masks*), of size 3x3x3 or larger. Additional conditions are used to prevent removal of *surface* or *curve endpoints* in order to maintain the geometrical properties of the object. Directional thinning methods remove voxels only from one particular direction in each pass (for example, North, South, Up, Down) using different numbers of directions and conditions to identify endpoints [9][20][36][48][54][64][65][71][88]. Because of this, these methods are sensitive to the order in which the different directions are processed and the resulting skeletons may not be centered within the object. Fully parallel [27][56][59] and non-directional thinning methods [16][83] do not have this disadvantage. Some thinning methods produce a surface-skeleton in the first stage and continue to thin until a one voxel wide skeleton is obtained [16]; others directly produce a curve-skeleton. Most thinning algorithms are designed and proven correct for a specific connectivity.

### 5.2 Using a Distance Field

The distance transform or distance field is defined for each interior point  $P$  of a 3D object  $O$  as the smallest distance from that point to the boundary  $B(O)$  of the object:

$$D(P) = \min_{P \in O} (d(P, Q)), \text{ where } d \text{ is some distance metric.}$$

Various distance functions can be used such as the Euclidean distance or an approximation such as the <3,4,5> chamfer metric [15]. A distance field can also be approximated using fast marching methods [76][87].

Ridges in this distance field correspond to voxels that are locally centered within the object. Most of the methods in this class attempt to find these voxels. These act as potential candidates (from the larger pool of object voxels) for curve-skeleton points. The candidates must then be somehow “pruned” or thinned to produce a 1D skeleton. The resulting values are then connected using a path connection or minimum spanning tree approach [82][92][96]. Therefore, most of the algorithms have

three steps: 1. find ridge points (local maxima, saddles), 2. prune and 3. connect. Other methods explicitly maintain the connectivity by combining steps 2 and 3. Methods used to find candidate voxels include: distance ordered thinning [26][31][32] [70], gradient searching [10], divergence computation [17], geodesic front propagation [66], thresholding the bisector angle [58] or shrinking the surface along the gradient of the distance field [75]. In voxel coding approaches, the distance field is combined with a distance-from-a-source field to generate a skeleton [97].

After obtaining candidate values from the original volume, these points are clustered and connected. For connectivity, most use minimum spanning trees, shortest paths [38][91] or other graph algorithms. In [96] an “LMpath” defines the connectivity of local maxima clusters. An alternative is provided by the fixed topology skeleton which is a set of a fixed number of connected active contours driven by the underlying distance field [35].

The distance field method can accurately extract the medial surface; however they cannot extract a curve-skeleton from arbitrary objects without employing additional techniques to prune the medial surface. For example, for the box in Figure 1(b), the voxels along the center plane (shaded) all have the same distance field value. Therefore, some sort of pruning must be used to simplify it into a line.

The main advantage of these methods is that computation of the distance field is very fast and it is usually needed by the application. Furthermore, for tubular objects, the distance field approach works very well.

### 5.3 Geometric methods

Geometric methods usually apply to objects represented by polygonal meshes or scattered point sets in continuous space.

A popular approach is to use the Voronoi diagram [18] generated by the vertices of the 3D polygonal representation or directly by a set of unorganized points [3][4][61]. The Voronoi diagram represents a subdivision of the space into regions that are closer to a generator element (a mesh vertex in the case of a 3D model) than to any other such element. The internal edges and faces of the Voronoi diagram can be used to extract the skeleton of the shape.

Cores and M-reps [21][69] are also medial-axis/surface approaches. A core is a locus in a space whose coordinates are position, radius, and associated orientations. The location of the core represents the middle of the figure and the spread of the core represents the width of the figure. M-reps are a generalization of the Core concept. The M-rep models the medial surface using a “web” of connected atoms. Each atom describes the position, width, local figural frame which implies the figural directions, and an object angle between opposing, corresponding positions on the implied boundary.

A similar structure is the *shock scaffold*, which relies on the concept of contact spheres [49][50] and represents the medial axis/surface by a set of shock curves, defined as the intersection of medial surface sheets (not the curve-skeleton).

The methods described above can be labeled as medial-axis/surface based. The main disadvantage of medial-axis based geometric methods is their sensitivity to noise. For example, Amenta’s *power shape* [3] contains a large number of unwanted branches that need to be pruned to extract a simple skeleton [4]. Additionally, these methods are more computationally intensive than the thinning/distance field based methods.

There are other geometric methods that avoid the medial axis altogether. One approach is based on computing level sets of a geodesic graph and then extracting a skeleton by connecting the centers of adjacent levels [89]. The resulting skeleton is not unique for a given object, being dependent on the location of the

source point used to construct the geodesic graph. Li *et al.* [51] construct a line segment skeleton by collapsing edges in length order (shortest first). This method is sensitive to the mesh tessellation. Katz and Tal [43] first decompose a mesh surface into segments using clustering, and then use this segmentation to construct a skeleton (each clustering can be represented by a centered vertex). But if the input mesh is not 2-manifold, the results may be unpredictable.

### 5.4 General-Field Functions

Various types of fields generated by functions other than the distance transform can also be used to extract curve-skeletons. Included in this class are potential field function [2][23][25] where the potential at a point interior to the object is determined as a sum of potentials generated by point charges on the boundary of the object; electrostatic field function [37]; visible repulsive force function [94]; radial basis function [57]. The skeleton points are found by determining the “sinks” of the field and connecting them using a force following algorithm [25] or minimizing the energy of an active contour [57].

The main advantage of these functions over the distance field is that they can produce nice curves on medial sheets where the distance field is constant. This is because they take into account larger boundary areas, not just the distance to the closest point on the boundary. This also creates an averaging effect that makes these algorithms less sensitive to boundary noise. However, they are much more expensive to compute.

Resolution of the voxel grid also affects the field functions which tend to be more sensitive to noise in thin regions of the object because significant contributions toward the final field value at an interior point come from fewer boundary voxels.

## 6 DISCUSSION

In this section, we discuss the described curve-skeletonization methods in terms of the properties presented in Section 3.

**Homotopy** is explicitly checked only by the thinning methods while removing points. The geometric methods based on the Voronoi diagram also preserve the topology. None of the other methods provide any guarantees regarding homotopy.

**Invariant under isometric transformations:** Directional thinning methods are sensitive to the object orientation and thus are not invariant under transformation. The other methods are.

**Reconstruction:** It should be obvious that regardless of the method used to compute it, a complete accurate reconstruction of the original object is not possible from the information retained in a curve-skeleton alone, except for certain objects (cylindrical objects for example). But clearly, a denser curve-skeleton will generate a better reconstruction [31].

**Thinness (1D)** is an implicit property of the general-field methods which use force-following or active contours to generate 1D skeleton branches. Thinning algorithms can also directly produce a curve-skeleton or further thin the surface-skeleton to a 1D representation. However, both algorithms are resolution dependent. Distance field methods and geometric methods do not produce a 1D representation directly. Therefore, both require significant post-processing.

**Centeredness** is achieved in the first steps by the methods using a distance field, however, once clustering and spanning trees are used, centeredness may be lost (for example [91]). The geometric methods can better achieve centeredness since contact points are directly computed [50] and can be incorporated more easily into the pruning steps. However, Voronoi-based methods are dependent on the sampling rate of the object’s surface: a dense sampling produces a more centered skeleton [3]. Thinning and general field methods do not guarantee centeredness (for example,

for directional thinning this would depend on the order of the directions).

**Reliability** must be checked for all of the classes. It is unclear which of the methods produces the most reliable skeleton (i.e., without adding more points to the initial skeleton).

**Junction Detection and Component-wise Differentiation:** Potential field methods and some thinning algorithms that specifically identify joints have the ability to distinguish the different components of the skeleton and determine a corresponding part structure of the original object. The other methods must test for joints after significant pruning and clustering [91]. The joints placement in these cases is very sensitive to slight changes in the object.

**Connectivity** is usually checked by all the algorithms to ensure at least a minimal degree of homotopy with the original object. Some algorithms (e.g., thinning) explicitly maintain connectivity during computation, while other methods check and enforce connectivity in a post-processing step.

**Robust:** Thinning, distance field and geometric methods are sensitive to noise, generating unnecessary branches in the skeleton as a result and methods have been proposed to filter the resulting skeletons [4][26]. The general field approaches are less susceptible to noise because of the large amount of averaging included in the underlying computation. Field based methods are more sensitive to resolution because thin regions in the objects can cause numerical instabilities in the computations.

Many of the algorithms described in the literature are usually illustrated with only a few examples and are not tested on a large database of general 3D objects, like [77]. Thus it is unclear how robust and general these algorithms are with respect to the choice of their parameters.

**Efficiency:** The Euclidean distance field of a 3D object can be computed in linear time using the algorithm of Saito and Toriwaki [73]. The subsequent steps of filtering and reconnecting the skeleton may, however, have a higher complexity but they usually operate on a greatly reduced set of voxels. Thinning is also a linear process in the number of object voxels. Computation of the Voronoi diagram of a set of  $n$  points in 3D is  $O(n^2)$  in the worst case [3]. The complexity of potential field computation is  $O(n^2)$ [25], where  $n$  is the number of object voxels.

## 6.1 Implementation

In section 5, four classes for the curve-skeletonization algorithms are given. The classes are divided into a “core” part and then a “post-core” step which is necessary to prune, cluster, connect or smooth the skeleton. In this section, we describe the results of comparing one algorithm from each class on 6 objects including one “real” object (a colon dataset) and one object with noise (the chess piece). Because many of the algorithms described in the literature are difficult to implement (typically not all of the details are given such as specific thresholds, epsilon values and cluster parameters), we have only implemented the “core” part of the algorithms.

From the methods using a distance field, we implemented the parameter controlled filtering of the distance function described by Gagvani and Silver [31]. From the thinning class, we implemented the 12-subiteration curve thinning algorithm described by Palágyi and Kuba in [65]. To represent the geometric methods, we used Amenta’s implementation of the *power shape* (a Voronoi diagram based medial axis) [3]. Finally, we used our implementation of the potential field method described in [25].

The purpose of this comparison is to get a sense of what each method-type can offer in terms of extracting a curve skeleton. Clearly this is not a fair comparison of the various algorithms in the different classes, since typically additional pre- and/or post-processing steps are performed that improve the results

significantly. It does, however, convey a sense of how much additional processing may be required to extract a curve-skeleton.

The results on a small set of test objects are shown in Figure 3. Note that for the *power shape* algorithm, only the surface voxels were given as input to the program and the results shown in the figure are the inside poles determined by the algorithm [3]. In the case of the potential field method, we show only the core skeleton generated by connecting the critical points of the vector field [25] (i.e., the “extra branches” have to be added).































	Test Objects	Distance Field	Thinning	Geometric	Potential Field
Thin Block (34x9x54)					
Monster (54x87x75)					
Mushroom (80x87x59)					
Colon (204x132x260)					
Chess Piece (40x39x87)					
Chess Piece with 10% noise on the surface (39x38x86)					

Figure 3. Skeletons of various objects. Only the “core” of each class of methods was implemented.

From these results, it is clear that if one is interested in a thin skeleton, the potential field method yields the thinnest and cleanest skeleton at the initial stage. The distance field and the geometric methods do not generate a curve-skeleton directly and the resulting skeleton points need to be pruned and then connected to obtain a curve-skeleton. Thinning produces a connected curve but the curve needs to be smoothed and, as can be seen, is sensitive to noise. In terms of running time, the potential field method is the slowest. In Figure 4 we show a comparison of the running times (in milliseconds, on a logarithmic scale) recorded for each method as a function of the total number of object voxels. (note that no optimizations were done for the implementations.)

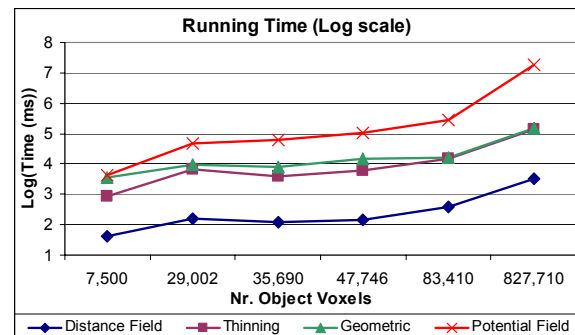


Figure 4. Running time (log scale) vs. number of object voxels.

## 7 SUMMARY

There have been many different curve-skeletonization algorithms given in the literature. In this paper, we have summarized the

visualization applications that use curve-skeletons and have distilled the list of skeleton properties necessary for these applications. We have then classified the algorithms for computing the curve-skeleton based upon their implementations and have discussed how each methodology achieves the different skeleton properties, using an implementation of the “core” of each class.

## 8 ACKNOWLEDGEMENTS

This work is supported in part by NSF 0118760 and NSF EIA-0205178. We would also like to thank Dr. Raman Balasubramanian and Xiaosong Yuan for their help.

## REFERENCES

- [1] 3D Studio Max, Discreet, (<http://www4.discreet.com/3dsmax/>).
- [2] N. Ahuja, J. Chuang. Shape Representation Using a Generalized Potential Field Model, IEEE PAMI, 19(2): 169-176, 1997.
- [3] N. Amenta, S. Choi, R. Kolluri. The Power Crust, Proceedings of 6th ACM Symposium on Solid Modeling, 249-260, 2001.
- [4] D. Attali J.-O. Lachaud. Delaunay Conforming Iso-surface, Skeleton Extraction and Noise Removal, Comp. Geometry, 19(2-3):175-189, 2001.
- [5] S.R. Aylward, E. Bullitt. Initialization, Noise, Singularities, and Scale in Height Ridge Traversal for Tubular Object Centerline Extraction, IEEE Trans. on Medical Imaging 21(2), 2002.
- [6] S.R. Aylward, J. Jomier, S. Weeks and E. Bullitt, Registration and Analysis of Vascular Images, IJCV, 55(2-3), 2003.
- [7] D.C. Banks, B.A. Singer. Vortex tubes in turbulent flows: identification, representation, reconstruction, IEEE Vis., 1994.
- [8] D. Bartz, W. Straßer, M. Skalej, D. Welte. Interactive Exploration of Extra- and Intracranial Blood Vessels, IEEE Visualization, 1999.
- [9] G. Bertrand and Z. Aktouf. A three-dimensional thinning algorithm using subfields, Vision Geometry III, 2356:113-124. SPIE, 1994.
- [10] I. Bitter, A.E. Kaufman, M. Sato. Penalized-Distance Volumetric Skeleton Algorithm, IEEE TVCG, 7(3), 2001.
- [11] R.L. Blanding, G.M. Turkiyyah, D.W. Storti and M.A. Ganter. Skeleton-based Three-Dimensional Geometric Morphing, Computational Geometry, 15:129-148, 2000.
- [12] J. Bloomenthal. Medial Based Vertex Deformation, SIGGRAPH/Eurographics Symp. On Computer Animation, 147-151, 2002.
- [13] J. Bloomenthal, C. Lim. Skeletal Methods of Shape Manipulation, Shape Modeling Int'l, 1999.
- [14] H. Blum. A Transformation for Extraction New Descriptors of Shape, Models for the Perception of Speech and Visual Form, MIT Press, 1967.
- [15] G. Borgefors. On Digital Distance Transforms in Three Dimensions, Computer Vision and Image Understanding 64(3):368-376, 1996.
- [16] G. Borgefors, I. Nyström, G.S. Di Baja. Computing skeletons in three dimensions, Pattern Recognition, 32(7), 1999.
- [17] S. Bouix, K. Siddiqi. Divergence-Based Medial Surfaces, ECCV 1842:603-618, Springer-Verlag, 2000.
- [18] J.W. Brandt, V.R. Alazi. Continuous Skeleton Computation by Voronoi Diagram, CVGIP: Image Understanding, 55:329-338, 1992.
- [19] A. Brennecke, T. Isenberg. 3D Shape Matching Using Skeleton Graphs. Simulation and Visualization 2004, 299-310, 2004.
- [20] D. Brunner, G. Brunnett. Mesh Segmentation Using the Object Skeleton Graph, Proc. IASTED International Conf. on Computer Graphics and Imaging, 48-55, ACTA Press 2004.
- [21] C.A. Burbeck, S.M. Pizer. Object representation by cores: Identifying and representing primitive spatial regions. Vision Research, 35(13):1917-1930, 1995.
- [22] S.W. Choi, H.P. Seidel. Linear Onesided Stability of MAT for Weakly Injective 3D Domain, Proc. ACM SMA, 2002.
- [23] J. Chuang, C. Tsai, Min-Chi Ko. Skeletonization of Three-Dimensional Object Using Generalized Potential Field, IEEE PAMI, 22(11):1241-1251, 2000.
- [24] N.D. Cornea, M.F. Demirci, D. Silver, A. Shokoufandeh, S.J. Dickinson, P.B. Kantor. 3D Object Retrieval using Many-to-many Matching of Curve-skeletons, To appear in SMI, 2005.
- [25] N. Cornea, D. Silver, X. Yuan, R. Balasubramanian. Computing Hierarchical Curve-Skeletons of 3D Objects, Submitted for publication, The Visual Computer, 2005.
- [26] M. Couprie and R. Zrouf. Discrete Bisector Function and Euclidean Skeleton, Lecture Notes in Computer Science, vol. 3429, Springer-Verlag, 2005.
- [27] U. Eckhardt, G. Maderlechner. Invariant Thinning, Pattern Recognition and Artificial Intelligence (7):1115-1144, 1993.
- [28] A.F. Frangi, W.J. Niessen, R.M. Hoogeveen, T. van Walsum, M.A. Viergever. Model-based quantitation of 3-D magnetic resonance angiographic images, IEEE Trans. on Medical Imaging, 18(10):946-956, 1999.
- [29] Y. Fridman, S.M. Pizer, S. Aylward, and E. Bullitt. Extracting Branching Tubular Object Geometry via Cores, Medical Image Analysis, 8(3):169-176, Elsevier, 2004.
- [30] D.S. Fritsch, S.M. Pizer, B.S. Morse, D.H. Eberly, A. Liu. The Multiscale medial axis and its applications in image registration, Pattern Recognition Letters, 15:445-452, 1994.
- [31] N. Gagvani and D. Silver. Parameter Controlled Volume Thinning, Graphical Models and Image Processing, 61(3):149-164, 1999.
- [32] N. Gagvani and D. Silver. Animating volumetric models, Academic Press Professional 63(6):443-458, 2001.
- [33] N. Gagvani, D. Silver. Shape-based volumetric collision detection, Proc. IEEE Symp. On Volume Visualization, 2000.
- [34] P. Giblin, B.B. Kimia. A formal classification of 3D medial axis points and their local geometry, Proc. IEEE CVPR, 566-573, 2000.
- [35] P. Golland, W.E.L. Grimson. Fixed Topology Skeletons, IEEE CVPR, 2000.
- [36] W. Gong and G. Bertrand. A simple parallel 3d thinning algorithm. Proc. IEEE Pattern Recognition, 188-190, 1990.
- [37] T. Grigorisshin, Y.H. Yang. Skeletonization: An Electrostatic Field-Based Approach, Pattern Analysis and App., 1:163-177, 1998.
- [38] T. He, L. Hong, D. Chen, Z. Liang. Reliable Path for Virtual Endoscopy: Ensuring Complete Examination of Human Organs, IEEE Trans. Visualization and Comp. Graphics, 7(4):333-342, 2001.
- [39] M. Hilaga, Y. Shinagawa, T. Kohmura, T. Kunii. Topology Matching for Fully Automatic Similarity Estimation of 3D Shapes, Proc. ACM SIGGRAPH, Aug. 2001.
- [40] L. Hong, S. Muraki, A. Kaufman, D. Bartz, T. He. Virtual Voyage: Interactive Navigation in the Human Colon, SIGGRAPH, 1997.
- [41] A. Kanitsar, D. Fleischmann, R. Wegenkittl, P. Felkel, M. E. Gröller – CPR: Curved Planar Reformation, IEEE Visualization, 2002.
- [42] A. Kanitsar, R. Wegenkittl, D. Fleischmann and M. E. Gröller - Advanced Curved Planar Reformation: Flattening of Vascular Structures, IEEE Visualization, 2003.
- [43] S. Katz, A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts, Proc. ACM SIGGRAPH 2003.
- [44] R.A. Katz, S.M. Pizer. Untangling the Blum Medial Axis Transform, IJCV, 55(2-3):139-153, 2003.
- [45] T.Y. Kong, A. Rosenfeld. Digital topology: Introduction and survey. Comp. Vision, Graphics, and Image Proc., 48(3):357-393, 1989.
- [46] T.Y. Kong, A.W. Roscoe, A. Rosenfeld. Concepts of digital topology, Topology and its App., 46(3):219-262, Elsevier, 1992.
- [47] F. Lazarus, A. Verroust. Metamorphosis of Cylinder-like Objects, Journal of Visualization and Comp. Animation. 8(3):131-146, 1998.
- [48] T. Lee and R.L. Kashyap. Building skeleton models via 3-d medial surface/axis thinning algorithms. CVGIP: Graphical Models and Image Processing, 56(6):462-478, November 1994.

- [49] F.F. Leymarie, B.B. Kimia. Computation of the Shock Scaffold for Unorganized Point Clouds in 3D, Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2003.
- [50] F.F. Leymarie. 3D Shape Representation via Shock Flows, Ph.D. thesis, Brown University, May 2003.
- [51] X. Li, T.W. Woon, Z. Huang. Decomposing polygon meshes for interactive applications, Proc. ACM Symp. On Interactive 3D Graphics, 35-42, 2001.
- [52] A. Lieutier. Any open bounded subset of  $R^n$  has the same homotopy type than its medial axis, Proc. ACM SMA, 2003.
- [53] P. Liu, F. Wu, W. Ma, R. Liang, M. Ouhyoung. Automatic Animation Skeleton Construction Using Repulsive Force Field, 11th Pacific Conference on Computer Graphics and Applications, 2003.
- [54] C. Lohou and G. Bertrand, A 3D 12-subiteration thinning algorithm based on P-simple points, Discrete Applied Mathematics 139:171-195, Elsevier, 2004
- [55] W.E. Lorensen, H.E. Cline. Marching cubes: A high resolution 3D surface construction algorithm, ACM SIGGRAPH Computer Graphics, 21(4):163-169, 1987.
- [56] C.M. Ma and M. Sonka. A fully parallel 3d thinning algorithm and its applications. Computer Vision and Image Understanding, 64(3):420-433, 1996.
- [57] W. Ma, F. Wu, M. Ouhyoung. Skeleton Extraction of 3D Objects with Radial Basis Functions, IEEE SMA, 2003.
- [58] G. Malandain, S. Fernandez-Vidal. Euclidean Skeletons, Image and Vision Computing, vol. 16:317-327, 1998.
- [59] A. Manzanera, T. Bernard, F. Preteux, B. Longuet. A unified mathematical framework for a compact and fully parallel n-D skeletonization procedure, Vision Geometry VIII, Vol. 3811: 57-68, SPIE, 1999.
- [60] Maya, Alias, (<http://www.alias.com>).
- [61] M. Näf, O. Kubler, R. Kikinis, M.E. Shenton, G. Szekely. Characterization and Recognition of 3D Organ Shape in Medical Image Analysis Using Skeletonization, IEEE Workshop on Math. Methods in Biomedical Image Analysis, 139-150, 1996.
- [62] C.W. Niblack, P.B. Gibbons, D.W. Capson. Generating skeletons and centerlines from the distance transform, Graphical Models and Image Processing, 54(5):420-437, 1992.
- [63] I. Nyström, Ö. Smedby. Skeletonization of Volumetric Vascular Images – Distance Information Utilized for Visualization, Journal of Combinatorial Optimization, vol. 5:27-41, 2001.
- [64] K. Palagyi and A. Kuba. Directional 3d thinning using 8 subiterations. Proc. DGCI, vol. 1568 of Lecture Notes on Computer Science, 325-336. Springer-Verlag, March 1999.
- [65] K. Palagyi and A. Kuba. A parallel 3d 12-subiteration thinning algorithm. Graphical Models and Image Proc., 61(4):199-221, 1999.
- [66] D. Perchet, C. I. Fetita, F. Preteux. Advanced navigation tools for virtual bronchoscopy, Proc. SPIE Conf. on Image Processing: Algorithms and Systems III, vol. 5298, 2004.
- [67] S.M. Pizer, G. Gerig, S. Joshi, S. Aylward. Multiscale Medial Shape-Based Analysis of Image Objects, Proc. IEEE, 91(10):1670-1679, 2003.
- [68] S.M. Pizer, D. Fritsch, P. Yushkevich, V. Johnson, E. Chaney. Segmentation, Registration, and Measurement of Shape Variation via Image Object Shape, IEEE Trans. Medical Imaging, 18:851-865, 1999.
- [69] S.M. Pizer, A.L. Thall, D.T. Chen. M-reps: A new object representation for graphics, Tech. Rep. TR99-030, University of North Carolina, Chapel Hill, NC, 17, 1999.
- [70] C. Pudney. Distance-Ordered Homotopic Thinning: A Skeletonization Algorithm for 3D Digital Images, Computer Vision and Image Understanding, 72(3):404-413, 1998.
- [71] P.K. Saha, B.B. Chaudhuri, D. Dutta Majumder. A new shape preserving parallel thinning algorithm for 3d digital images. Pattern Recognition, 30(12):1939-1955, 1997.
- [72] P.K. Saha, B.B. Chaudhuri. 3D Digital Topology under Binary Transformation with Applications, Computer Vision and Image Understanding, 63(3):418-429, 1996.
- [73] T. Saito, J. Toriwaki. New algorithms for Euclidean Distance Transformation of an n-Dimensional Digitized Picture with Applications. Pattern Recognition, 27:1551-1565, 1994.
- [74] K. Santilli, K. Bemis, D. Silver, J. Dastur, P. Rona. Generating realistic images from hydrothermal plume data, IEEE Vis., 2004.
- [75] H. Schirmacher, M. Zöckler, D. Stalling, H. Hege. Boundary Surface Shrinking - a Continuous Approach to 3D Center Line Extraction, Proc. of IMDSP, 25-28, 1998.
- [76] J.A. Sethian. Fast Marching Methods, SIAM Review, 41(2):199-235, 1999.
- [77] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser. The Princeton Shape Benchmark, Shape Modeling International, 2004.
- [78] D. Silver and N. Gagvani. Unwinding the Colon, Medicine Meets Virtual Reality (MMVR) 2002.
- [79] E. Sorantin, C. Halmaj, B. Erdohelyi, K. Palagyi, L.G. Nyul, K. Olle, et.al.. Spiral-CT-based assessment of tracheal stenoses using 3-D-skeletonization, IEEE Trans. Medical Imaging, 21(3):263, 2002.
- [80] M. Sramek, A.E. Kaufman. Alias-Free Voxelization of Geometric Objects, IEEE Trans. Vis. and Comp. Graph., 5(3): 251 – 267, 1999.
- [81] M. Straka, M. Cervenansky, A. La Cruz, A. Kochl, M. Sramek, E. Groller, D. Fleischmann. The VesselGlyph: focus & context visualization in CT-angiography, IEEE Visualization, 2004.
- [82] H. Sundar, D. Silver, N. Gagvani, S. Dickinson. Skeleton Based Shape Matching and Retrieval, Proc. Shape Modeling Int'l, 2003.
- [83] K. Suresh. Automating the CAD/CAE Dimensional Reduction Process, ACM Symp. On Solid Modeling and Applications, 2003.
- [84] S. Svensson, I.Nystrom and G.S. Di Baja. Curve-skeletonization of Surface-like Objects in 3D Images Guided by Voxel Classification, Pattern Recognition Letters, 1419-1426, 2002.
- [85] S. Svensson, C. Arcelli, G.S. Di Baja. Finding cavities and tunnels in 3D complex objects, Int'l Conf. on Image Analysis and Processing, 2003. pp. 342 – 347.
- [86] A. Tal and S. Katz. Hierarchical Mesh Decomposition using Fuzzy Clustering and Cuts, ACM Trans. on Graphics, 22(3):954-961, 2003.
- [87] A. Telea, A. Vilanova. A robust level-set algorithm for centerline extraction, Eurographics/IEEE Symp. On Data Visualization, 2003.
- [88] Y.F. Tsao and K.S. Fu. A parallel thinning algorithm for 3d pictures. Computer Vision, Graphics and Image Proc., 17:315-331, 1981.
- [89] A. Verroust, F. Lazarus. Extracting skeletal curves from 3D scattered data, The Visual Computer, 16:15-25, 2000.
- [90] B. Vrolijk, F. Reinders, F.H. Post. Feature tracking with skeleton graphs, in Data Visualization: The State of the Art, pp. 37--52, Kluwer Academic Publishers, 2003.
- [91] L. Wade, R.E. Parent. Automated generation of control skeletons for use in animation, The Visual Computer 18(2):97-110, 2002.
- [92] M. Wan, F. Dachille, A. Kaufman. Distance-Field Based Skeletons for Virtual Navigation, IEEE Visualization 2001.
- [93] R. Webster, M. Harris, R. Shenk, J. Blumenstock, J. Gerber, C. Billman, A. Benson, R. Haluck. Using an Approximation to the Euclidean Skeleton for Efficient Collision Detection and Tissue Deformations in Surgical Simulators, Medicine Meets Virtual Reality, vol. 13, IOS Press, 2005.
- [94] F. Wu, W. Ma, P. Liou, R. Liang, M. Ouhyoung. Skeleton Extraction of 3D Objects with Visible Repulsive Force, Eurographics Symp. On Geometry Processing, 2003.
- [95] Y. Zhao, H. Ong, T. Tan, Y. Xiao. Intuitive interfaces for animation: Interactive control of component-based morphing, SIGGRAPH, 2003.
- [96] Y. Zhou, A. Kaufman, A. W. Toga. Three-dimensional Skeleton and Centerline Generation Based on an Approximate Minimum Distance Field, The Visual Computer, 14, pp. 303-314, 1998.
- [97] Y. Zhou, A.W. Toga. Efficient skeletonization of volumetric objects, IEEE Trans. Visualization and Comp. Graphics, 5(3):196-209, 1999.