3D Skeletonization as an Optimization Problem

Denis Khromov[*] Leonid Mestetskiy[†]

Abstract

We present a novel approach to 1D curve-skeletonization of 3D objects. The skeletonization process is reduced to a numerical optimization problem. The method provides a strict way to evaluate and compare various 1D skeletons of the same 3D object. We describe a particular implementation of our approach and discuss experiment results.

1 Introduction

A skeleton of a shape is a graph that captures major topological and metrical properties of the shape. It may be considered as a 1D thinning of the original shape. Skeletons are useful in computer vision since it is much easier to extract a shape's features from a graph rather than from its boundary description. The skeleton of a two-dimensional shape is usually defined as a shape's medial axis. The medial axis is the set of all points having more than one closest point on the shape's boundary. The medial axis of a 2D shape is always a 1D set. Such a set can be computed efficiently. However, a 3D medial axis contains 2D sheets and therefore is not a graph.

A curve-skeleton is a 1D skeleton of a 3D shape. It is known to be an ill-defined object [3]. There is no common strict definition recognized by a significant number of papers on curve-skeletons. A curve-skeleton is usually defined as an object produced by some particular algorithm. Different algorithms produce different skeleton graphs; those graphs have different properties. Examples of such algorithms can be found in [3, 11]. Therefore it is almost impossible to compare different algorithms with each other. It is only possible to evaluate the computational performance and the visual quality of different skeletons.

On the other hand, there is an intuitive idea of what a correct curve-skeleton should be. It is clear that different curve-skeletons of the same 3D object are visually similar, even if they are defined differently. Our goal is a mathematical formalization of this intuitive idea.

In this article we present a method to evaluate the quality of a curve-skeleton. We define a curve-skeleton as a continuous thinning of the original object. We also describe the procedure of reconstruction which allows to produce a 3D shape from a 1D skeleton. The shape produced from a curve-skeleton is called a silhouette. The measure of similarity between the original shape and the skeleton's silhouette is an evaluation of the skeleton's quality. Thus the problem of skeletonization can be formulated as a numerical optimization problem. To construct a skeleton of the object means to find the best approximation of the object by the skeleton's silhouette. We describe our implementation of this idea. The implementation includes the measure of similarity, the first approximation algorithm, and numerical methods for the minimization process.

The key feature of our method is the numerical evaluation of the curve-skeleton's quality. However, our work shares some common ideas with the recent papers on the subject. We use Reeb graphs to compute the first approximation of our skeletons. Reeb graphs are widely used in topological shape analysis [1, 4, 5, 10, 14]. The primitives we use to approximate the original object are very similar to the sphere swept volumes (SSV). There are papers presenting the usage of SSV for 3D shape approximation [2, 7]. In [9] a deformable model is used to describe the object's shape; the curve-skeleton itself is obtained from this deformable model. There are very few methods involving the optimization of a skeleton's quality. One example is the paper [12], where an iterative least squares optimization is performed to shrink the model and obtain its accurate thinned representation.

2 Definitions

As mentioned above, there are many definitions of a 3D curve-skeleton. The common idea of these definitions is that the curve-skeleton is a thinned 1D representation of the 3D object [3]. It can be formalized in terms of homotopy.

Let Ω be a connected open set embedded in $\mathbb{R}^3$ with boundary $\partial\Omega$. Let $\overline{\Omega}$ be its closure:

$$\overline{\Omega} = \Omega \cup \partial\Omega. \tag{1}$$

Let $\Gamma \subset \overline{\Omega}$ be a 3D representation of some graph G such that every edge of G is mapped onto a smooth curve $\gamma \in \mathbb{R}^3$. We denote the Euclidean distance between points $x, y \in \mathbb{R}^3$ by $\rho(x, y)$.

Definition 1 Γ *is a curve-skeleton of* Ω *if there is a*

[*] Moscow State University, denis.v.khromov@gmail.com

[†] Moscow State University, l.mest@ru.net

continuous function

$$H : [0;1] \times \partial\Omega \to \overline{\Omega} \qquad (2)$$

such that

$$H(0,x) = x, H(1,\partial\Omega) = \Gamma. \qquad (3)$$

The function

$$\sigma(x) \equiv H(1,x) \qquad (4)$$

is called a skeleton mapping.

A skeleton mapping describes the correspondence between the surface of the object and its skeleton.

One of the advantages of a 2D skeleton is the ability to recover the original shape from a skeleton. That is possible due to a distance transform function (DTF). A DTF defines ”width” of the shape for every skeleton point. Then the shape is a union of discs: the centers of these discs are situated on the skeleton branches, and the radii are defined by the DTF. We need some analogue of a 2D distance transform in order to preserve the reconstruction possibility for 3D curve-skeletons.

Definition 2 *A radial function r is a non-negative real-valued function defined on a curve-skeleton:*

$$r : \Gamma \to \mathbb{R}, r(x) \geq 0 \ \forall x \in \Gamma. \qquad (5)$$

Definition 3 *A silhouette of a curve-skeleton Γ with a radial function r is a set*

$$S(\Gamma, r) = \bigcup_{x\in\Gamma} B_{r(x)}(x), \qquad (6)$$

where $B_{r(x)}(x)$ is a ball with the center in x and the radius $r(x)$:

$$B_{r(x)}(x) = \{y \in \mathbb{R}^3 : \rho(x,y) \leq r(x)\}. \qquad (7)$$

A single curve with its silhouette is called a fat curve [8]. Fat curves can be used for an approximation of tubular objects [6]. An example of a silhouette of a skeleton is shown in Figure 1.

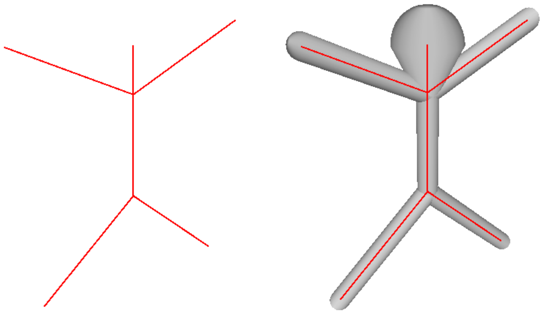


Figure 1: A 3D graph (left) and its silhouette produced by some radial function (right).

A silhouette can be considered as a reconstruction of the original 3D object. Unlike 2D skeletons, a 3D silhouette is merely an approximation of Ω. So we need a numerical measure of similarity between the original shape Ω and the silhouette $S(\Gamma, r)$. It is possible to define it as a distance between two sets in $\mathbb{R}^3$ (for example, the Hausdorff distance). But this approach may produce skeletons that do not correspond to the intuitive idea of what a correct curve-skeleton is. An example is shown in Figure 2. The graph Γ consists of one parabola, and the radial function r is such that the silhouette $S(\Gamma, r)$ looks like a straight tubular object because of its self-intersection. It can be considered as an ordinary subset of $\mathbb{R}^3$. In that case its curve-skeleton is expected to look like a single straight line segment, not a parabola. But the silhouette of the parabola is a perfect approximation for this object if we use something like the Hausdorff distance.

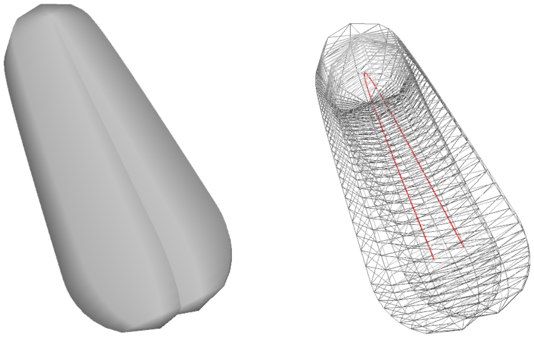


Figure 2: A silhouette of the parabola (red): solid (left) and wireframe (right).

In this paper we propose the following function.

Definition 4 *Let Γ be a curve-skeleton of Ω with a radial function r. An approximation error of (Γ, r) is a value*

$$\mathcal{E}(\Omega, \Gamma, r) = \int_{x\in S=\partial\Omega} \Big(\rho^2\big(x, \sigma(x)\big) - r^2\big(\sigma(x)\big)\Big)^2 dS. \qquad (8)$$

This function takes into account the correspondence σ between $\partial\Omega$ and Γ. It takes small values when the distance between $x \in \partial\Omega$ and $\sigma(x) \in \Gamma$ is close to $r\big(\sigma(x)\big)$. It is satisfied when the skeleton's curve gives the right representation of the approximated shape. Thus Equation 8 gives a mathematical meaning to the intuitive concept of a 3D curve-skeleton.

3 First Approximation

The main idea of our method is a numerical optimization of a curve-skeleton in order to minimize the approximation error given in (8). This approach suggests that there is a first approximation of a curve-skeleton. It can be constructed by almost any existing skeletonization algorithm.

We construct the initial curve-skeleton using Reeb graphs.

Definition 5 *Let f be a continuous function defined on a compact manifold M. A Reeb graph is a quotient space*

$$R_f = M/\sim_f, \tag{9}$$

where $\sim_f$ is an equivalence relation such that

$$x \sim_f y \Leftrightarrow \exists l \subset \partial\Omega : x, y \in l, f(z) \equiv \text{Const}\ \forall z \in l, \tag{10}$$

and l is a connected curve in $\partial\Omega$.

Algorithms based on Reeb graphs are rather effective. They always produce topologically correct skeletons. This is important since it is difficult to change the topology of a skeleton during the numerical optimization process. Another important advantage of Reeb graphs is that the skeleton mapping σ is generated explicitly.

The first approximation is computed in a few steps. A continuous scalar function

$$f : \partial\Omega \to \mathbb{R} \tag{11}$$

is called a mapping function. The corresponding Reeb graph R_f is a graph-like space. It describes the topology of Ω. We need to embed it into $\mathbb{R}^3$ in order to generate a 3D representation of the graph.

The embedding is defined as follows:

$$\Gamma = \bigcup_{l \in L} \text{MEB}(l), \tag{12}$$

$$\sigma(x) = \text{MEB}(l \in L : x \in l). \tag{13}$$

MEB(l) is the center of a minimum enclosing ball covering the set l. Here minimum enclosing balls play the same role as maximum inscribed balls do in the definition of a medial axis.

The appearance of a skeleton is determined by the mapping function f. In our implementation it is defined as follows. Let $p \in \partial\Omega$ be a fixed point called a pole. The mapping function $f(x)$ equals the geodesic distance over $\partial\Omega$ between x and the pole p. This function does not depend on the object's orientation in space. This is a serious advantage over functions like a height map. Another benefit of our mapping function is that it has a low computational complexity.

4 Implementation

In this section we discuss the implementation of our method for 3D objects represented by polygonal meshes. Let M be a triangulated closed surface with a set of vertices V and a set of edges E. We assume that V is uniform and detailed enough.

To avoid confusion with the mesh and its vertices and edges, in this section we will use the term "joint" for a curve-skeleton vertex and the term "bone" for an edge. By J denote the set of joints and by B denote the set of bones. In our implementation, all bones are straight line segments and all radial functions are linear. Each joint

$$j = (x_j, r_j),\ x_j \in \mathbb{R}^3, r_j \geq 0 \tag{14}$$

is defined by 4 real values: the position x_j and the value of the radial function (or radius) r_j. Each bone

$$b = (j_1, j_2),\ j_1, j_2 \in J \tag{15}$$

connecting joints j_1 and j_2 describes a curve and a radial function given by equations

$$\gamma_b(t) = tx_{j_1} + (1-t)x_{j_2}, \tag{16}$$

$$r_b(t) = tr_{j_1} + (1-t)r_{j_2}, \tag{17}$$

where $t \in [0; 1]$.

The skeleton mapping value for a vertex $v \in V$ is defined by the bone $b^v = (j_1^v, j_2^v)$ and the parameter t_v:

$$\big(j_1^v, j_2^v, t_v\big), (j_1^v, j_2^v) \in B, t_v \in [0; 1], \tag{18}$$

$$\sigma(v) = \gamma_{(j_1^v, j_2^v)}(t_v) = t_v x_{j_1^v} + (1-t_v)x_{j_2^v} \tag{19}$$

The computation of this mapping will be explained below.

The integral in (8) is approximated with a sum $\tilde{\mathcal{E}}$ over all vertices:

$$\begin{aligned}\tilde{\mathcal{E}} = \sum_{v \in V}\Big(\rho^2\big(v, \sigma(v)\big) - r^2\big(\sigma(v)\big)\Big)^2 = \\ = \sum_{v \in V}\Big(\rho^2\big(v, t_v x_{j_1^v} + (1-t_v)x_{j_2^v}\big) - \\ -\big(t_v r_{j_1^v} + (1-t_v)r_{j_2^v}\big)^2\Big)^2.\end{aligned} \tag{20}$$

4.1 First Approximation

In order to find the first approximation we need to

1. select the pole vertex;
2. compute the mapping function f and its contour lines;
3. construct the skeleton itself.

The geodesic distance between two vertices $v_1, v_2 \in V$ is approximated by the length of the shortest chain of edges from E connecting v_1 and v_2. The advantage of this method is that it is rather fast. In some cases it may produce a rough approximation. However, that is not an issue since we don't need the exact values of geodesic distance. The accuracy of this method is enough to produce contour lines which define an acceptable Reeb graph.

The pole vertex is computed in three steps:

1. let v_1 be a random vertex from V;
2. let v_2 be the vertex farthest from v_1;
3. the pole vertex p is the vertex that is the farthest from v_2.

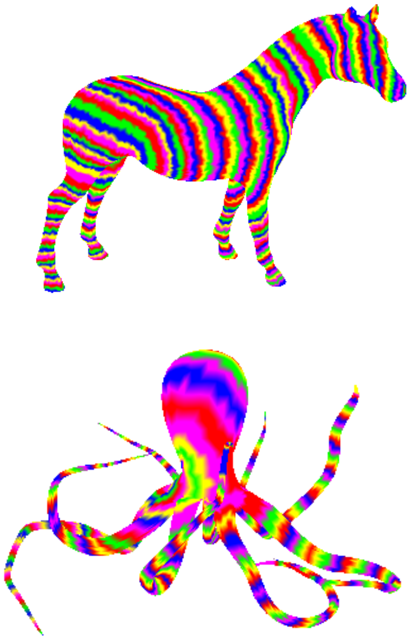


Figure 3: Contour lines of the mapping function.

The set V is divided into contour lines of f. It is impossible to obtain exact contour line of f because we work with the discrete set of points V. So the following algorithm is performed. Let $l_k \subset V$ be some contour line. The contour line l_{k+1} produced on the next step of the algorithm is a maximum connected subset of V such that

$$\exists u \in l_{k+1} : \tag{21}$$

$$\exists v_1 \in l_k \quad (v_1, u) \in E, \tag{22}$$

$$\forall v_2 \in l_{k+1} \quad f(v_2) \leq f(u). \tag{23}$$

More then one connected subset satisfying these conditions means that we have reached a joint where a group of skeleton branches converge: each subset corresponds to a new branch. Figure 3 demonstrates an example: contour lines are painted in different colors.

Each contour line corresponds to a skeleton joint $j \in J$. The position of this joint x_j is computed as a center of the minimum enclosing ball. We use Welzl's algorithm to find the MEB [13]. The radius of this ball is taken for the first approximation of the radial function value r_j.

4.2 Numerical Optimization

This step of the algorithm involves the minimization of the function (20). This function is a polynomial of degree 4 with $4|J|$ variables (since we have $|J|$ joints and each is defined by 4 scalar variables). It can be represented in the form of the sum

$$\begin{aligned}\mathcal{E}(z) = \sum_{v \in V} \left((a_v z + b_v)^2 - c_v \right)^2 = \\ = z^4 \left(\sum_{v \in V} a_v^4 \right) + \\ + z^3 \left(\sum_{v \in V} 4 a_v^3 \right) + \\ + z^2 \left(\sum_{v \in V} (6 a_v^2 b_v^2 - 2 b_v^2 c_v) \right) + \\ + z \left(\sum_{v \in V} (4 a_v b_v^3 - 4 a_v b_v c_v) \right) + \\ + \left(\sum_{v \in V} (b_v^4 - 2 b_v^2 c_v + c_v^2) \right).\end{aligned} \tag{24}$$

with respect to each single variable z. For example, for $z = r_j$

$$a_v = t_v, b_v = (1 - t_v) r_{j_2^v}, c_v = \rho^2 \left(v, \sigma(v) \right) \tag{25}$$

if $j_1^v = j$,

$$a_v = (1 - t_v), b_v = t_v r_{j_1^v}, c_v = \rho^2 \left(v, \sigma(v) \right) \tag{26}$$

if $j_2^v = j$ and

$$a_v = b_v = c_v = 0 \tag{27}$$

otherwise.

We use a gradient descent to minimize $\tilde{\mathcal{E}}$. Let J_k be a $4|J|$-dimensional vector describing the set of joints computed in step k. Then J_{k+1} is given by

$$J_{k+1} = J_k - \lambda \nabla \tilde{\mathcal{E}}(J_k), \tag{28}$$

where λ is a solution of the minimization problem

$$\arg \min_{\lambda \in \mathbb{R}} \tilde{\mathcal{E}}(J_k - \lambda \nabla \tilde{\mathcal{E}}(J_k)). \tag{29}$$

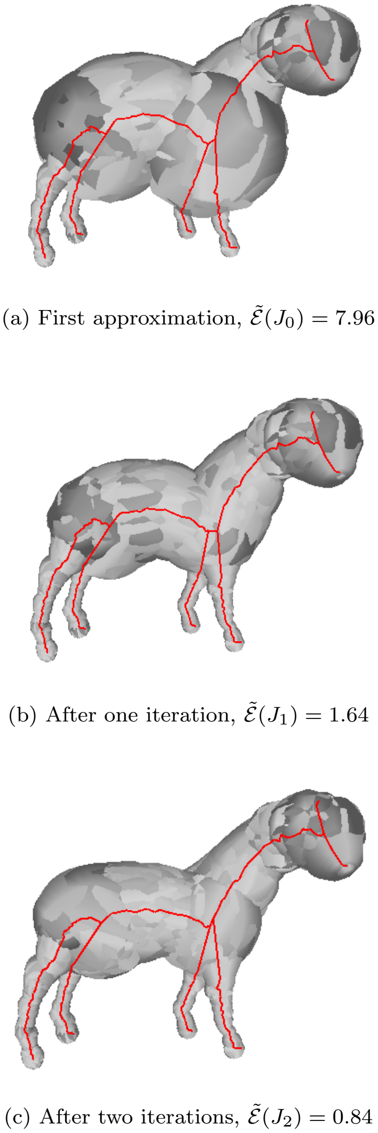
It is a polynomial of degree 4 with respect to only one variable. Therefore it is very easy to find its minimum analytically. $\tilde{\mathcal{E}}$ and $\nabla \tilde{\mathcal{E}}$ are computed using (24). The procedure (28) is performed until the condition

$$|\tilde{\mathcal{E}}(J_k) - \tilde{\mathcal{E}}(J_{k+1})| < \varepsilon \tag{30}$$

is satisfied, where $\varepsilon > 0$ is a fixed parameter of the algorithm. In our experiments

$$\varepsilon = 10^{-3}\tilde{\mathcal{E}}(J_0), \qquad (31)$$

where J_0 is the first approximation. The optimization process is demonstrated in Figure 4. It shows the first approximation and two serial iterations of the gradient descent for the horse model from Figure 3.



(a) First approximation, $\tilde{\mathcal{E}}(J_0) = 7.96$

(b) After one iteration, $\tilde{\mathcal{E}}(J_1) = 1.64$

(c) After two iterations, $\tilde{\mathcal{E}}(J_2) = 0.84$

Figure 4: The curve-skeleton of the horse and its silhouette.

Note that this process may lead to negative radial functions. Negative values in radii have no physical meaning. We change the sign of each negative r_j since they are included in $\tilde{\mathcal{E}}$ with the second degree.

5 Experiments and Discussion

Some examples of our algorithm's work are shown in Figure 5. The resulting curve-skeletons are demonstrated with their corresponding silhouettes. These 3D models are often used in articles on curve-skeletons, so they are suitable for visual comparison of our skeletonization method with other ones.

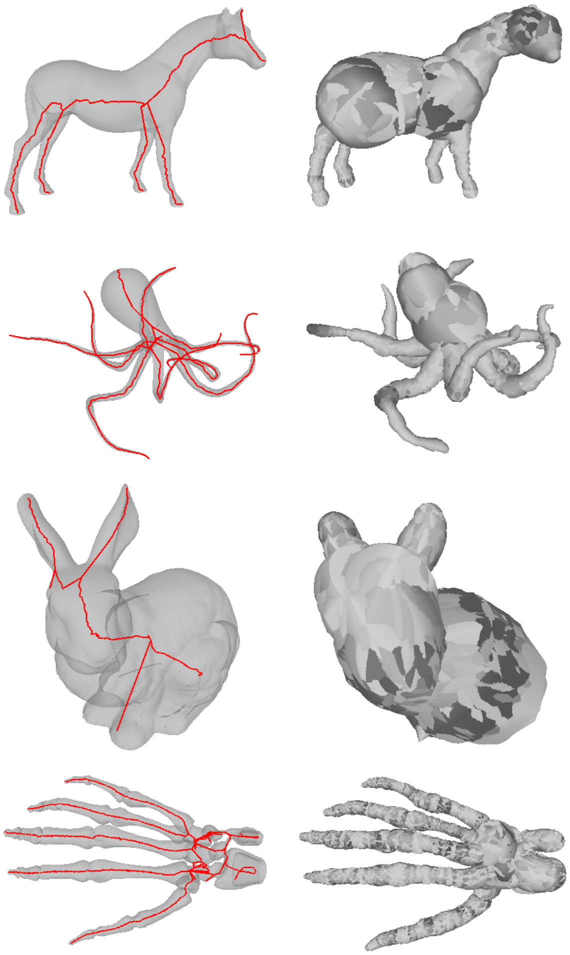


Figure 5: The results produced by our algorithm: the skeletons (left) and their silhouettes (right) for various 3D objects.

An object with a full-dimensional medial axis is shown in Figure 6. It does not have any significant tubular fragments. Curve-skeletons are not very useful for objects like this. The resulting silhouette is not a valuable approximation.

Some desirable properties of a curve-skeleton are listed in [3]. Our approach guarantees some of them:

- the topological properties (such as homotopic invariance) are provided by the usage of Reeb graphs;
- the invariance under isometric transformations is not always guaranteed by methods based on Reeb

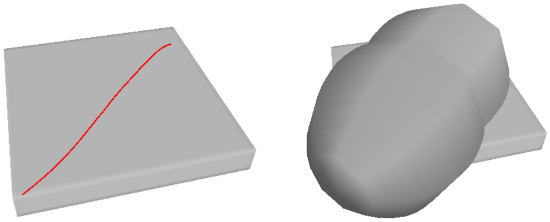


Figure 6: An object with a full-dimensional medial axis.

graphs (for example, the mapping function can depend significantly on one of the coordinates), but in our case this property is satisfied;

- the possibility of reconstruction is a key feature of our method;
- the skeletons are thin by definition;
- the centeredness is the value that is strictly measured and optimized with our algorithm.

Some properties are not satisfied:

- robustness is not guaranteed since some images may produce skeleton branches for visually insignificant protrusions of the shape. These branches should be considered as noise which can't be compensated by the minimization of (20);
- smoothness is another property that is provided neither by Reeb graphs nor by the optimization process. However, the usage of curves of higher degrees instead of piecewise linear curves could fix that problem.

6 Conclusions and Future Work

We have demonstrated an approach to 3D curve-skeletonization which allows to evaluate skeletons and to choose the best one among others. This method is implemented. The implementation and experiments prove the practical utility of the method. Below we list several open issues for further research.

1. Tuning the skeleton mapping σ during the minimization process. Currently it is fixed, which imposes strict requirements on the quality of the first approximation and its skeleton mapping.

2. Taking mesh edges and faces into account. Current algorithm uses vertices only. This means that skeletons of the detailed meshes are more accurate than for the low polygon meshes.

References

[1] M. Attene, S. Biasotti, M. Spagnuolo. Shape understanding by contour driven retiling. *The Visual Computer*, 19:2–3, 2003.

[2] M. Bae, J. Kim, Y.J. Kim. User-guided volumetric approximation using swept sphere volumes for physically based animation. *Computer Animation and Virtual Worlds*, 2012.

[3] N.D. Cornea, D. Silver. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13:530–548, 2007.

[4] M. Hilaga, Y. Shinagawa, T. Kohmura, T.L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 203–212, 2001.

[5] R.E. Khoury, J. Vandeborre, M. Daoudi. 3D mesh Reeb graph computation using commute-time and diffusion distances. *Proc. SPIE* 8290, 82900H, 2012.

[6] D. Khromov. Curve-Skeletons based on the fat graph approximation. *Lecture Notes in Computer Science*, Volume 6915/2011, 239–248, 2011.

[7] E. Larsen , S. Gottschalk , M. C. Lin , D. Manocha. Fast proximity queries with swept sphere volumes. *IEEE International Conference on Robotics and Automation*, 3719-3726, 2000.

[8] L.M. Mestetskii. Fatcurves and representation of planar figures. *Computers & Graphics*, Volume 24, Issue 1, February 2000.

[9] A. Sharf, T. Lewiner, A. Shamir, L. Kobbelt. On-the-fly curve-skeleton computation for 3D shapes. *In Proceedings of Comput. Graph. Forum*,323-328, 2007.

[10] Y. Shi, R. Lai, S. Krishna, N. Sicotte, I. Dinov, A.W. Toga. Anisotropic Laplace-Beltrami eigenmaps: bridging Reeb graphs and skeletons. *Computer Vision and Pattern Recognition Workshop*, pp. 1-7, 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, 2008.

[11] K. Siddiqi, S. M. Pizer . Medial Representations: mathematics, algorithms and applications. *Springer*, 2008.

[12] Y. Wang, T. Lee. Curve-Skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics*, 14:926–936, 2008.

[13] E. Welzl. Smallest enclosing disks (balls and ellipsoids). *Results and New Trends in Computer Science*, 1991.

[14] Y. Xiao, J.P. Siebert, N. Werghi. A discrete Reeb graph approach for the segmentation of human body scans. *Fourth International Conference on 3-D Digital Imaging and Modeling*, 2003.