Skeleton Extraction of 3D Objects with Radial Basis Function

Fu-Che Wu, Wan-Chun Ma, Ming Ouhyoung
Communication and Multimedia Lab
Dept. of Computer Science and Information Engineering
National Taiwan University
{joyce, firebird}@cmlab.csie.ntu.edu.tw, ming@csie.ntu.edu.tw

ABSTRACT

Medial axis transform (MAT) is the locus of centers of all spheres that have maximal radius in the interior of an object. A lower dimension shape in medial axis representation is a skeleton of an object. The skeleton itself in general can be used in 3D object retrieval. The requirements of the skeleton differ with applications. For pattern recognition, it requires primitive features for similarity comparison. For surface reconstruction, it requires detailed information to reduce the approximation error. Whereas many previous works have been done, most of these methods are time consuming and sensitive to noise, or restricted to specific 3D models.

A practical approach for extracting primitive skeleton from general 3D polygonal models using radial basis function (RBF) is proposed. This approach can generate skeletons suitable for information retrieval. Given a 3D polygonal model, we use the model vertices as centers to construct a RBF level set. Next, we shrink the surface vertices to local maximums in the RBF following the gradient vector field. With the inherited connectivity from the original model, we connect these local maximums with lines that deform by active contour model. Skeleton construction is done when the potential energy of these lines is minimized.

Categories and Subject Descriptors

I.5.2 [**Design Methodology**]: Pattern analysis

Keywords

Skeleton, radial basis function, active contour model, variational implicit surface, implicit level set

1. INTRODUCTION

Since the computation and graphics power is greatly improved, three dimensional (3D) related applications become more and more popular in computer generated virtual world. To use a large 3D object database efficiently, MPEG-7 concerns the concept of object description and object retrieval. The related techniques are surface reconstruction and information retrieval of objects. Intuitively, the skeleton of an object is a good feature for these requirements. How to extract the skeleton from 3D objects becomes one of the most important subjects in recent years. In traditional approaches, medial axial transform (MAT) is a way to extract a medial axis from a closed shape.

Different applications require different skeletonization methods. If the application is for object shape retrieval, we will focus on the skeletons that represent the objects both topologically and geometrically. Exact skeleton representation such as MAT is not necessary for this kind of application. On the other hand, for surface reconstruction, we require a skeleton from which object can be reconstructed that has minimal error compared to original one. In this point of view, MAT seems to be the best choice.

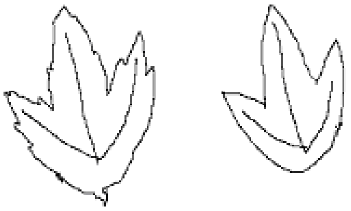


Figure 1: two arbitrary shapes

For example, given two arbitrary shapes in Figure 1, they are similar if we use the two skeletons to compare with each other. However, for the surface reconstruction, the error may not be acceptable if we use the left skeleton to reconstruct the left object. In this paper, we will focus on extracting a primitive skeleton that follows the topology and geometry of an original 3D object. A 3D object can be represented by an implicit function. We construct this function as the following form:

$$f(x) = h$$

where x is a point in $\Re^3$ and h is a level value of some surface. The function itself is regarded as a height level set. Given an implicit function level set of a 3D object, we can extract the skeleton by using surface shrinking and active contour model techniques.

Submit to ACM Multimedia 2002 in the Juan les Pins, France. December 1-6, 2002.

1.1 Previous Works

The notion skeleton was first introduced by Blum [4]. The skeleton, as the union of the centers of maximum spheres inside a 2D object, is a kind of reduced representation and provides region-based shape features. A good skeleton needs to fulfill the following two requirements:

1. Topology: The skeleton must retain the topology of the original object.
2. Geometry: Force the skeleton being in the middle of an object and being invariant under the affine transformations including translation, rotation, and scaling.

Many previous works focused on 2D shape skeletons. Traditionally, there are two classes of approaches to extract the skeleton of a shape. One approach is to compute the skeleton directly from the object surface points, such as using the Voronoi diagram. The other is to extract skeleton by constructing a distance field of an object.

Direct computation method: Many previous works construct the skeleton of 2D shapes by using the Voronoi diagram [18, 17]. The idea is straightforward to be extended into 3D space. The skeleton in 3D is the union of the center of maximum spheres inside a 3D object where the surface is just ”touched” by these balls. With this concept, Edelsbrunner and Mücke [11] introduce an α-shape to represent an object. The concept of alpha shapes is an approach to formalize the intuitive notion of shape for sample points. The α-shape is a generalization of the convex hull and a subgraph of the Delaunay triangulation. The real number α which is represented as the radius of a ball with $0 < \alpha < \infty$, controls the desired level of detail of shapes.

Following the idea of α-shape, Amenta et al. [2] proposed the ”power crust” algorithm for 3D surface reconstruction and MAT approximation. The algorithm computes the Voronoi diagram of the surface sample points first and retrieves the set of polar balls by selecting from the Voronoi balls that have maximal distance to the sampled surface. By labelling polar balls with power diagram, the object contains the surface sample points now can be transformed into the polar ball representation. Connecting the centers of the polar balls forms a good discrete approximation to the MAT.

Tracing approach is another direct computation method. Sherbrooke et al. [19] and Culver et al. [7] propose similar approaches. They first find a seam as start point. From the arithmetic computation, they search for the exact medial axis. Another tracing approach, Chung et al. [6] proposed a force field model. They assume each polygonal face of an object produces a force field. They push each node following the force field to a force balanced position as the skeleton candidate. Finally, they connect all of these nodes to get the result. However, practical use of above approaches show that the results are not reliable among general 3D objects.

Distance field method: Constructing a Voronoi diagram needs $O(n^3)$ in complexity where n is the input point number. Due to this reason, direct computing the medial axis from geometry is not suitable for the conditions of large number of input points or arbitrary shapes. Another approach is to use distance field. It is common to use discrete approach to construct the distance field and extract the skeleton. The result is similar to the medial axis. The basic methodology is as following:

1. Construct the distance field of an arbitrary object.
2. Find the local maximums of the distance field.
3. Connect these local maximums to generate the skeleton.

Several distance field methods are invoked to extract skeleton. Leymarie and Levine [15] implemented a 2D MAT (they called it grassfire transformation) by minimizing the distance field energy of an active contour model with initial position at the boundary of a given shape. Zhou and Toga [24] proposed a voxel-coding method which is a recursive voxel by voxel propagation. The algorithm starts from a set of seed voxels. Then, they design a coding schema to detect object connectivity and geometric features, such as centerlines and skeleton. Bitter et al. [3] proposed a penalized-distance algorithm to extract skeleton from volumetric data. Based on pre-calculation of distance field, the algorithm finds the farthest voxels as the centerline, which is an initial skeleton. Then, they discard voxels near the skeleton recursively to refine the skeleton.

2. PROBLEM ANALYSIS

Let S be a surface. A skeleton of surface S is a set of points that is denoted as $M(S)$. Each point q on surface S we can find a point p which belong to skeleton $M(S)$. We denote it as $M(q) = p$. For each p on skeleton, we can find the inverse skeleton that are set of points denoted as $M^{-1}(p)$. Then, if $M(S)$ is a well define of skeleton of surface S, it satisfies some of the following properties:

1. Neighborhood property: if p and q are neighbors in $M(S)$. $M^{-1}(p)$ and $M^{-1}(q)$ are neighbors in S.
2. Uniform property: Let $q \in S$, the distance between q and $M(q)$ is the radius of q, denoted as $\rho(q)$. Let $p \in M(S)$, for $q_i \in M^{-1}(q)$ the radius $\rho(q_i)$ is uniform. Therefore, the goal is to minimize the variant of $\rho(q)$.
$$\min \left(\int\limits_{\forall q \in M^{-1}(p)} \|\rho(q) - Average(\rho(q))\| dA \right)$$
3. Compactness property: The size of $M(S)$ is minimal.

The uniform property will prefer to produce symmetrical skeleton. Leyton [16] has inferred the Symmetry-Curvature Duality, The theorem indicates that each curvature extremum can be assigned a unique symmetry axis leading to the extremum. If the shape has more curvature variation, it will generate more skeletons. This explains that properties 2 and 3 are actually in conflict. In traditional definition, the MAT prefers the uniform property which is suitable for the surface reconstruction. In this paper, we want to extract a skeleton that can be applied to object retrieval.

Let Ω_0 be a connected bounded domain in $\Re^3$. If S_0 is an initial surface, $S_0 = \Omega_0$. A point p inside of S_0 can be denoted as $p \to S_0$. If S_n belongs to S_0, for all p, $p \to S_n \Rightarrow p \to S_0$. If there exists a distance function $D_s(p)$ that calculates the distance between point p and surface S, we can define a shrinking function $\nabla D_s(S_n)$ that can be applied to surface S_n. The shrinking function is a gradient vector operator in the distance field. Then, $S_n = S_{n-1} + \nabla D_s(S_{n-1})$. If S_n is a skeleton of a object then $D_s(S_{n-1}) = 0$.

3. IMPLICIT SURFACE WITH RADIAL BASIS FUNCTION

Many previous studies have been done for the implicit surface and radial basis functions [21, 22, 23, 10]. Given a set of constraint points that define the shape of the surface, an implicit surface can be created by using 3D scatter data interpolation. Turk and O'Brien construct implicit surfaces with radial basis function by solving a linear system [23]. This method is widely adopted for height interpolation or deformable models. In this paper, we invoke radial basis functions as height (distance) interpolation for building a distance field.

3.1 Implicit Surface Construction with Radial Basis Function

An implicit surface can be defined by a set of radial basis functions. The implicit surface function has the form:

$$f(x) = \sum_{j=1}^{n} d_j w(x - c_j) + p(x)$$

where c_j are the surface constraints, d_j are the weights, $w(x)$ is the radial basis function and $p(x)$ is a degree one polynomial for the linear and constant terms of $f(x)$. To solve the weights and polynomial parameters we need to assign interpolation constraints as following:

$$h_i = f(c_i)$$

then

$$h_i = \sum_{j=1}^{n} d_j w(c_i - c_j) + p(c_i)$$

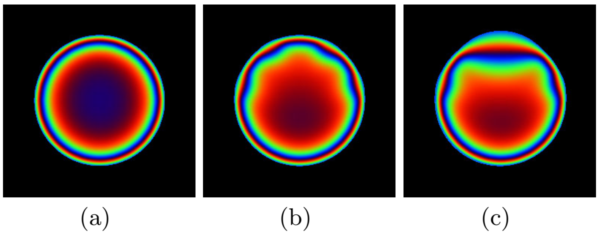


Figure 2: Cross section pictures of RBF level set for different models. (a) Level set from a 10x10 sampled sphere. (b) Reduce the vertex density of a sphere by 50%. (c) Reduce the vertex density of a sphere by 75%.

Assume that there are n interpolation constraints, the equation above can be formulated as a linear system:

$$\begin{bmatrix} W & C \\ C^T & 0 \end{bmatrix} \begin{bmatrix} D \\ P \end{bmatrix} = \begin{bmatrix} H \\ 0 \end{bmatrix}$$

where

$$\begin{array}{ll} C_i = (1, c_i^x, c_i^y, c_i^z) & i = 1, ..., n \\ W_{ij} = w(c_i - c_j) & i, j = 1, ..., n \\ D = (d_1, ..., d_n)^T & \\ P = (p_0, p_1, p_2, p_3)^T & \\ H = (h_1, ..., h_n)^T & \end{array}$$

The system itself is symmetric and positive semi-definite, so there is always a unique solution for D and P. We may solve this linear system to get D and P using LU factorization [8] or singular value decomposition (SVD) [9].

As for the interpolation constraints, we apply the normal constrains method described in [23]. There are two kinds of constraints assigned to the linear system in this method:

1. Surface constraints: We use the vertex set V of a arbitrary 3D model as surface constraints. For each vertex $v_i \in V$, we set the value $h_i = 0.0$.
2. Normal constraints: For each vertex v_i, we create the paired positive-valued normal constraint $r_i = v_i + k \times (-\overrightarrow{n})$, where $\overrightarrow{n}$ is the normal at vertex v_i. For each r_i, we set the value h_i a positive value, usually 1.0. The value of k depends on the size and resolution of an object. In our case, we set $k = 0.01$ for the 3D objects which are scaled within a $2 \times 2 \times 2$ cube.

We can choose many basis functions for the interpolation. Usually the appropriate basis function to use is $w(r) = |r|^3$. Dinh et al. [10] proposed a multiple order radial basis function that can control the smoothness of the implicit surface by changing δ and τ value. The basis function we invoke here is $w(r) = (|r|^2 + c^2)^{-0.5}$, where c is the control radius. This function has a simple form and similar behavior to the multiple order one. Notice the basis function must be continuous and partially differentiable.

3.2 Property of Constraints and Radius in Radial Basis Function

In this section, we will discuss about how the RBF parameters, such as interpolation constraints, radius and regularization parameters influence the RBF level set.

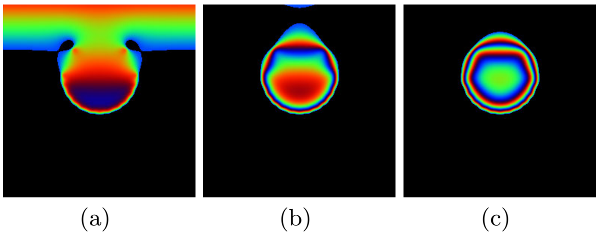


Figure 3: Cross section pictures of RBF level set with different RBF radius. (a) R = 0.1 (b) R = 0.2 (c) R = 0.3. The model is the same with Figure 2 (c). The radius of the sphere is 0.5.

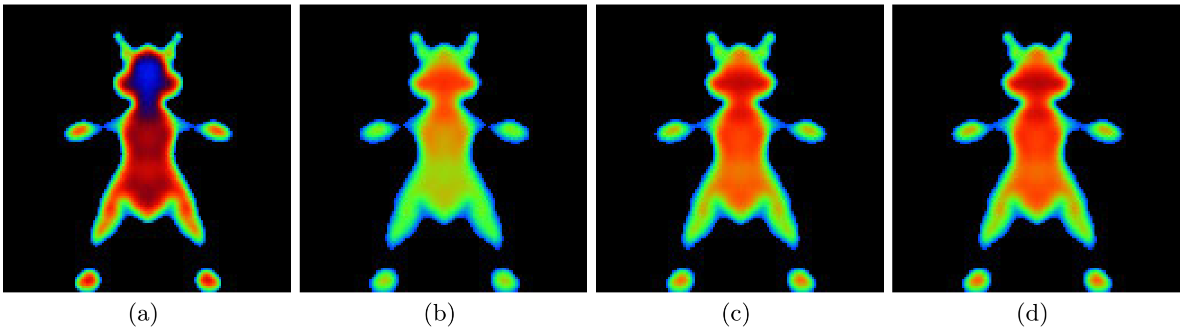


Figure 4: Cross section pictures from a mouse model coded in color, where green color means smaller while red means large values. (a) Exterior constraint λ (λ_{ex}) = 0.0, surface constraint λ (λ_{surf}) = 0.0, (b) λ_{ex} = 1.0, λ_{surf} = 1.0, (c) λ_{ex} = 1.0, λ_{surf} = 0.1 and (d) λ_{ex} = 1.0, λ_{surf} = 0.01. The darker the color, the higher the RBF value. The RBF level set seems to have better distribution when proper λ values are applied. The radius of all the above RBF level set is 0.05.

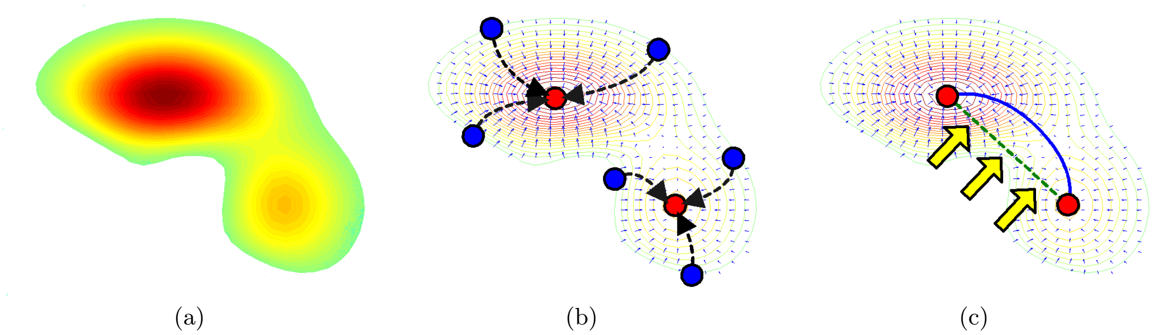


Figure 5: Illustrations of the skeletonization algorithm, where color coding is the same as in Figure 4. (a) The object and its RBF level set function in color visualization. (b) A gradient vector field (blue small arrows) is constructed with partial differential RBF. Surface points (blue points) shrink to the local maximums (red points) enforced by the gradient vector field. (c) After the local maximums being gathered, we link two local maximums with connectivity relationship together with a Snake (active contour model, the green dash line). Yellow arrows indicate the external force that pushes the Snake toward the direction that decreases the potential energy. The Snake itself deforms continuously until the potential energy is minimized (blue solid line), and a skeleton is found.

3.2.1 Interpolation Constraints

The RBF level set behavior is concerned with constraints distribution. The RBF of a sphere with equally constraint distribution is prefect. All the level set surfaces are homocentric spheres, which are shown in Figure 2 (a). As the reduced sphere models in Figure 2 (b) and (c), distortion happens in the RBF level set although the reconstructed surface boundary still seems the same. The RBF level set relaxes at the locations without the constraints, shown in the top region of the sphere in Figure 2 (c).

3.2.2 RBF Radius

Figure 3 shows the effect of the RBF control radius when applied to a reduced sphere model. As shown in Figure 3 (a), RBF surface with small radius sometimes may become out of control. If we increase the control radius, the result will be better and smoother (Figure 3 (b) and (c)). Using large radius to construct a RBF surface, however, may lose some geometry details due to the global smoothing effect.

3.2.3 Regularization Parameter

We apply the method proposed by Dinh et al [10] for the RBF surface smoothing. Several λ values can be assigned to the diagonal elements in the RBF system matrix. The smoothness of the surface is proportional to the λ values. As the λ values become larger, the RBF surface is smoother. From the viewpoint of the RBF level set behavior, the level set seems to distribute better if we apply this technique. The level set distribution behavior is important because it affects the RBF gradient vector field, which directs the convergence of the surface points.

4. SKELETONIZATION ALGORITHM

Our skeletonization algorithm states as following:

1. Compute the RBF level set from a given 3D model.
2. For each surface shrinking points, move it along the gradient vector direction until a local maximum position is reached.
3. Cluster the local maximums in a given range.

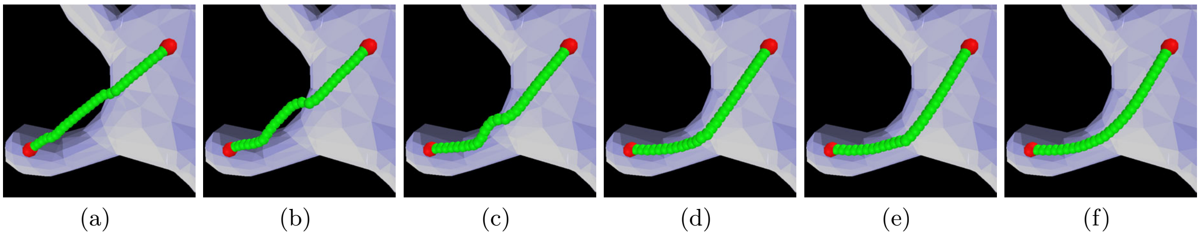


Figure 6: The skeleton deforming process, where it deforms successively until an energy-minimized position is found (From left to right).

Model	Constraint number	RBF radius	RBF construction (sec.)	Shrinking (sec.)	Skeleton construction (sec.)
Mouse	1,912	0.1	51.05	44.80	182.45
Bull	1,982	0.2	55.93	79.96	210.33
Octopus	2,004	0.1	57.44	41.25	344.62
Cobra	1,752	0.2	39.17	62.13	110.04
Frog	1,890	0.06	48.14	21.39	312.27
Pig	2,170	0.1	74.83	74.28	294.30

Table 1: Execution time statistics on a desktop computer with Intel Pentium-4 processor running at 1.5 GHz and with 256 Mbytes RAMBUS memory. We apply MathWork MATLAB C++ Library for the RBF construction matrix computation.

4. Link any two maximums which have connectivity relationship together with the Snake (active contour model) method [14] and minimize the Snake energy.

5. Output the final positions of the Snakes as skeleton.

4.1 Surface Shrinking

As previously discussed, RBF $f(x) = c$ is a continuous, partial differentiable level set where c describes a height value for any given x. An interior point has higher RBF value than exterior one. With this idea, we can regard a cross section plane from RBF as a height contour map shown in Figure 5 (a). The most interior points are the highest peaks. Each point from the surface moves inside an object along the gradient direction until a corresponding peak (local maximum) is found.

A gradient vector field can be easily built by only substituting partial differentiable basis function for the original one without any changes to the basis weight and the linear term parameters. We choose the vertices of the model as the surface shrinking points without losing generality. Sometimes the vertices of an arbitrary model may not be well distributed. Methods such as dense sampling or random points relaxation [20] can be applied to make the shrinking points somewhat equally spaced.

For a shrinking point v_i in the i_{th} iteration of the shrinking process, we calculate the new position of v_{i+1} by:

$$v_{i+1} = v_i + gradient(v_i) \times step$$

where *step* is a small discrete factor. The iteration stops when the RBF value of v_{i+1} is smaller than that of v_i. The final position of the v is recorded as a local maximum. The whole process stops when all of the shrinking points stayed in their corresponding local maximums. Figure 5 (b) illustrates the shrinking process.

4.2 Clustering

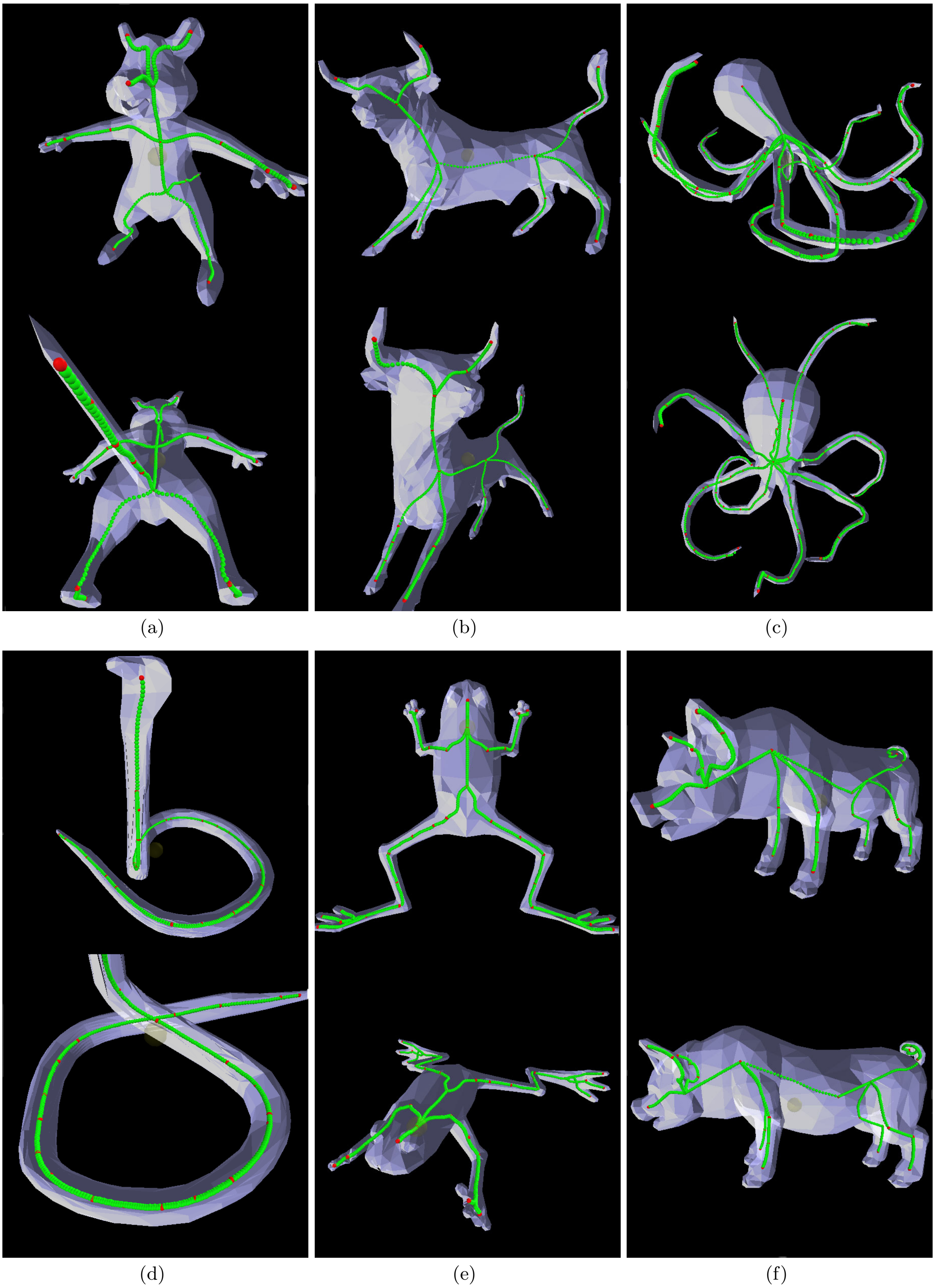
In practice, each shrinking point converges to the local maximum points within a small area. The range between the point and the local maximum depends on the precision of shrinking step. Due to this reason, we merge the final positions of the shrinking points in a given range to become a single local maximum point.

4.3 Connecting

As we use the vertices of the 3D model to be the shrinking points, it is intuitive that the edges of a model become the connectivity relationship. For two shrinking points v_i and v_j and their corresponding local maximums position m_i and m_j, the connectivity relationship is defined as that if v_i and v_j share an edge in the original 3D model, then m_i and m_j have connectivity relationship. We invoke a Snake (active contour model) method to link two local maximums which have connectivity relationship. An active contour model is generated by adjusting a contour to minimize an energy function:

$$E_{Snake} = E_{Internal} + E_{External}$$

where $E_{Internal}$ and $E_{External}$ means the internal external energy respectively. The internal energy is the part that depends on intrinsic properties of the snake, such as length or curvature. $E_{External}$ means the external energy. The external energy depends on many factors such as distance field. Initially, we connect two local maximums as a straight line. We sample some points on the straight line as candidates. The position of each point is adjusted on the constrained plane which is perpendicular to the original straight line described in Appendix A. When the energy of the active


(a) (b) (c)

(d) (e) (f)

Figure 7: Skeletons extracted from 3D models. (a) ”Mouse”, (b) ”Bull”, (c) ”Octopus”, (d) ”Cobra”, (e) ”Frog”, (f) ”Pig”. Models are rendered in semi-transparent style. The red and green spheres represent local maximums and skeleton points respectively.

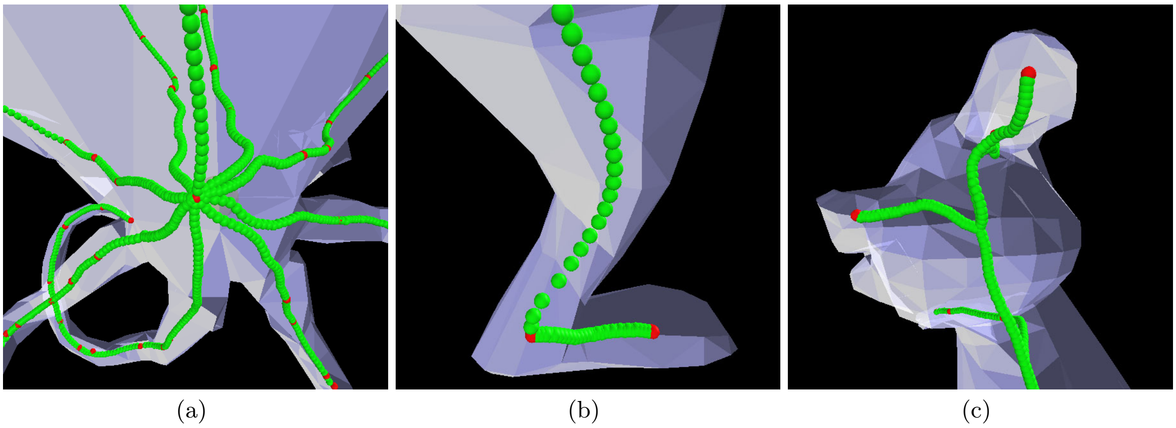


Figure 8: Feature details of the selected skeleton. (a) The connection joint of the head and the eight tentacles of an octopus model. (b) The leg of a mouse model. (c) The head of a mouse model.

contour model is minimized, we regard the Snake itself as a part of the skeleton (Figure 5 (c)). Figure 6 shows the deforming process of a Snake.

5. RESULT

The results are shown in Figure 7. Detailed views of some models are shown in Figure 8. The execution time statistics for each model in Figure is shown in Table 1. The error evaluation algorithm is described in Appendix B, and the skeleton we generated as compared to the ”ideal” medial axis has a deviation from 2.9% to 8.5%.

Some detailed features, such as the hand part of the ”mouse” and the ”frog” are not shown because of we use larger RBF radius to make the level set smooth and well distributed. The globally smoothing effect has been applied to the fine features of the model. However, significant shape features are preserved well in the skeleton representation, and these are important to be used in object retrieval algorithms.

6. CONCLUSIONS AND FUTURE WORK

Creating implicit surface by radial basis function provides us a simple and robust method to generate arbitrary 3D object distance field, which is needed for building gradient vector field and then, surface points convergence.

Our skeletonization method extracts both topological and geometrical information. Significant geometry detials, such as arm and leg in the model, can be well preserved.

There are many works need to be done in the future. From the viewpoint of research, we will keep working on the complicated relationship between the implicit surface, constraint distribution, RBF radius and regularization parameters. As for the basis functions, currently we only use the basis function $w(r) = (|r|^2 + c^2)^{-0.5}$ for all the RBF level set generation. In the future, we may continue to explore the RBF level set property with different basis functions. Computation efficiency is another area to be optimized.

Besides, from the viewpoint of applications, there exist many projects related to this method. For example, a 3D object retrieval system is going to take the skeletons as main comparison keys. We still need to evaluate the results compared with the method proposed by Hilaga et al. in [13], which is further implemented by Chen [5] in our lab.

We also want to apply the skeleton to the original model for animation use. We may select several points on the skeleton as joints. When some parts of the skeleton move, the surface vertices also move following the relationship between vertices and the skeleton. With this idea, we want to develop a system for artists to simply edit the skeleton movements and accurately animate the models with skin mesh method. This may become a useful tool for key frame animation.

7. ACKNOWLEDGEMENT

We would like to thank the 3DCAFE website [1] for providing all the models used in this paper.

8. REFERENCES

[1] http://www.3dcafe.com.

[2] N. Amenta, S. Choi, and R. Kolluri. The power crust. *Proceedings of the sixth ACM Symposium on Solid Modeling and Applications*, pages 249–260, 2001.

[3] I. Bitter, A. E. Kaufman, and M. Sato. Penalized-distance volumetric skeleton algorithm. *IEEE Transaction on Visualization and Computer Graphics*, 7(3):195–206, 2001.

[4] H. Blum. *A Transformation for Extracting New Descriptors of Shape*, pages 362–380. MIT Press, 1967.

[5] D. Y. Chen. http://3dsite.dhs.org/∼dynamic/.

[6] J. H. Chung, C. H. Tsai, and M. C. Ko. Skeletonization of three-dimensional object using generalized potential field. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 22(11):1241–1251, 2000.

[7] T. Culver, J. Keyser, and D. Manocha. Accurate computation of the medial axis of a polyhedron. *Proceedings of the fifth ACM symposium on Solid Modeling and Applications*, pages 179–190, 1999.

[8] B. N. Datta. *Numerical Linear Algebra and Applications*, pages 111–132. Brooks/Cole, 1995.

[9] B. N. Datta. *Numerical Linear Algebra and Applications*, pages 551–599. Brooks/Cole, 1995.

[10] H. Q. Dinh and G. Turk. Reconstructing surfaces by volumetric regularization. *GVU-00-26, College of Computing, Georgia Tech*, December 2000.

[11] H. Edelsbrunner and E. Mucke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13(1):43–72, 1994.

[12] H. Goldstein. *Classical Mechanics*, pages 164–166. Addison-Wesley, 1980.

[13] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. *SIGGRAPH 2001 Conference Proceedings*, pages 203–212.

[14] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1987.

[15] F. Leymarie and M. Levine. Simulating the grassfire transform using an active contour model. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 14(1):56–75, 1992.

[16] M. Leyton. Symmetry-curvature duality. *Computer Vision,Graphics, and Image Processing*, 38:327–341, 1987.

[17] N. Mayya and V. T. Rajan. Voronoi diagrams of polygons: A framework for shape representation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 638–643, 1994.

[18] R. Ogniewicz and M. Ilg. Voronoi skeletons: theory and applications. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 63–69, 1992.

[19] E. C. Sherbrooke, N. M. Patrikalakis, and E. Brisson. Computation of the medial axis transform of 3d polyhedra. *Proceedings of the third ACM symposium on Solid Modeling and Applications*, pages 187–199, 1995.

[20] G. Turk. Generating textures on arbitrary surfaces using reactiondiffusion. *SIGGRAPH 1991 Conference Proceedings*, pages 289–298.

[21] G. Turk, H. Q. Dinh, J. F. O'Brien, and G. Yngve. Implicit surfaces that interpolate. *Proceedings of the seventh International Conference on Shape Modeling and Applications*, pages 62–71, 2001.

[22] G. Turk and J. F. O'Brien. Shape transformation using variational implicit surface. *SIGGRAPH 1999 Conference Proceedings*, pages 335–342.

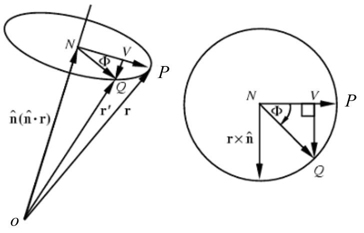
[23] G. Turk and J. F. O'Brien. Variational implicit surfaces. *GVU-99-15, College of Computing, Georgia Tech*, May 2000.

[24] Y. Zhou and A. Toga. Efficient skeletonization of volumetric objects. *IEEE Transaction on Visualization and Computer Graphics*, 5(3):195–206, 1999.

APPENDIX

A. COORDINATE TRANSFORMATION

To adjust a point which moves in a plane, we can define a point as $\mathbf{r} = (p_x, p_y, 0)_p$ in plane coordinate. The normal of the plane is as $\mathbf{n} = (n_x, n_y, n_z)$ that is also the straight line between two local maximums. A rotation axis is $\hat{\mathbf{n}} = \mathbf{z} \times \mathbf{n} = (-n_y, n_x, 0)$. A rotation angle is $\Phi = \cos^{-1}(\mathbf{z} \cdot \mathbf{n})$. As Figure 9 show, we can find the relation between before and after rotation[12].



N
V
Φ
P
Q
$\hat{\mathbf{n}}(\hat{\mathbf{n}}\cdot\mathbf{r})$
$\mathbf{r}'$
$\mathbf{r}$
o
N
V
P
Φ
$\mathbf{r}\times\hat{\mathbf{n}}$
Q

Figure 9: Coordinate rotation

$$\mathbf{r}' = \overrightarrow{ON} + \overrightarrow{NV} + \overrightarrow{VQ} = \mathbf{r}\cos\Phi + \hat{\mathbf{n}}(\hat{\mathbf{n}}\cdot\mathbf{r})(\mathbf{1} - \cos\boldsymbol{\Phi}) - (\mathbf{r}\times\hat{\mathbf{n}})\sin\boldsymbol{\Phi}$$

let $\hat{\mathbf{n}} = (x, y, z)$, then

$$\mathbf{r}' = \mathbf{M} \cdot \mathbf{r}$$

where $\mathbf{M}$ equals to

$$\begin{bmatrix} x^2 + \cos\Phi(1 - x^2) & xy(1 - \cos\Phi) - z\sin\Phi & zx(1 - \cos\Phi) + y\sin\Phi \\ xy(1 - \cos\Phi) + z\sin\Phi & y^2 + \cos\Phi(1 - y^2) & yz(1 - \cos\Phi) - x\sin\Phi \\ xz(1 - \cos\Phi) - y\sin\Phi & zy(1 - \cos\Phi) + x\sin\Phi & z^2 + \cos\Phi(1 - z^2) \end{bmatrix}$$

$$= \begin{bmatrix} n_y^2 + \cos\Phi(1 - n_y^2) & -n_x n_y(1 - \cos\Phi) & n_x\sin\Phi \\ -n_x n_y(1 - \cos\Phi) & n_x^2 + \cos\Phi(1 - n_x^2) & n_y\sin\Phi \\ -n_x\sin\Phi & -n_y\sin\Phi & \cos\Phi \end{bmatrix}$$

Based on this relation, we can transform a point in plane coordinate into world coordinate $\mathbf{p_w}$ as following:

$$\mathbf{p_w} = \mathbf{M} \cdot \mathbf{r} + \mathbf{T}$$

where $\mathbf{T}$ is the position of a shrinking candidate.

B. ERROR EVALUATION

Here we describe the error evaluation method used in this paper:

1. Surface of the input model are densely sampled to n points (In our evaluation, the models are sampled more than 20,000 points).
2. For each surface point v, find the nearest point k on the skeleton then define $v \in M^{-1}(k)$ and the distance between v and k is $\rho(v)$.
3. For each k, sort each $v \in M^{-1}(k)$ and find the smallest $\rho(v)$ and define it as $r(k)$, which is the zero error position. For each surface point v we calculate the error $e(v) = \frac{\rho(v)}{r(k)}$.
4. The total MSE equals to $\sqrt{\frac{\sum e^2(v)}{n}}$