Extraction of Skeletal Meshes from Volumetric Data by Sparse Polynomial Approximation

Yukie Nagai* Yutaka Ohtake*,** Kiwamu Kase** Hiromasa Suzuki*

*The University of Tokyo
{ nagai, yu-ohtake, suzuki }@den.rcast.u-tokyo.ac.jp
**VCAD Modeling Team, RIKEN
kiwamu@riken.jp

Abstract

The skeletal structures of solid objects play an important role in medical and industrial applications. Given a volumetrically sampled solid object, our method extracts a well-connected and not-fragmented skeletal structure represented as a polygon mesh. The purpose is to achieve a noise-robust extraction of the skeletal mesh from a real-world object obtained using a scanning technology such as the CT scan method. We first approximate the input image intensity through a set of spherically supported polynomials that provide an adaptively smoothed intensity field, and then perform a polygonization process to find the extremal sheet of the field, which is regarded as a skeletal sheet in this research. In our polygonization, a subset of the weighted Delaunay tetrahedrization defined by a set of spherical supports is used as an adaptively sampled grid. The derivatives for detecting extremality are analytically evaluated at the tetrahedron vertices. We also demonstrate the effectiveness of our method by extracting skeletal meshes from noisy CT images.

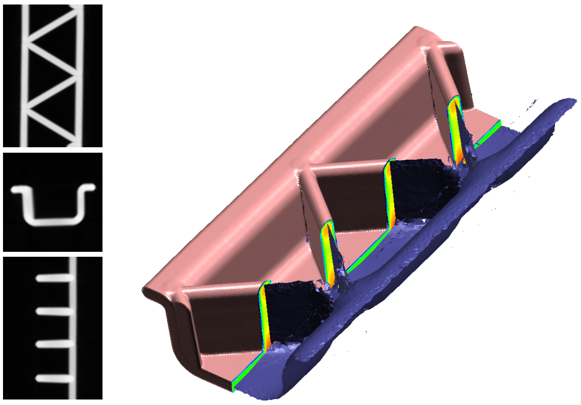

Figure 1. Left: three cross sections of a CT image. Right: an isosurface (light pink) and a skeletal mesh extracted using our method (dark blue). The coloring on the half cross section shows the input CT values (red for high and blue for low).

1. Introduction

The skeletal structures of solid objects have various applications in shape modeling and geometry processing. A shape simplified from a 3D solid to 2D skeletal sheets or 1D skeletal curves gives us more intuitive shape information for use in deformation, recognition, matching etc.

Skeletal-sheet extraction is important in industrial applications as well as in the field of computer graphics. Recent advances in X-ray-based CT scanning technology have enabled accurate measurement of mechanical objects consisting of thin parts, as shown in the images on the left of Fig. 1. We can easily extract the boundary of a scanned object using isosurface extraction techniques such as [16, 13]. However, as covered in [9, 10], it is often necessary to extract the scanned object as single sheets (rather than as a two-sided thin object) for further digital manipulation such as reverse engineering, quality evaluation or physical simulation. Skeletal-sheet extraction is one possible solution to this problem. The goal of this research is to extract a polygon mesh approximating the skeletal sheets of a CT-scanned thin object as demonstrated in Fig. 1.

Previous work. As reviewed in [5], many good methods for extracting skeletal structures have been developed so far. These are briefly reviewed below.

Given a voxelized solid, one method performs a thinning algorithm to obtain voxelized 1D/2D skeletons [12]. Another applies a distance transformation from the boundary voxels, and the voxels taking the local maxima of the distance field are then detected [22]. These techniques extract skeletal structures as voxels, thus requiring the application of an additional polygon mesh generation if the output model is desired as a piecewise linear approximation [10].

A polygonal approximation of the medial axis (i.e. skeletal sheets) can be directly obtained using a Voronoi diagram of sampling points on a solid boundary [1, 6]. Although this technique does not require grid sampling inside the solid, the topology of the resulting mesh is sensitive to small perturbations of the solid boundary. The method is therefore not suited to the processing of scanned objects, which typically have noisy boundaries.

Using a potential field [17, 4] is another possible approach. The skeletal points are extracted as the stationary points of a gradient descent flow defined by the sum of radial functions centered at points on a given solid boundary. This approach is mainly used for extracting 1D skeletal curves.

Our approach. Let a 3D scalar field $f(\boldsymbol{x})$ be a C^2- continuous function defined in a domain including a given solid object. f can be constructed from CT images of the object. Our skeletal mesh extraction is simply to find the locus of points satisfying the following conditions:

$$\langle \boldsymbol{e}, \nabla f \rangle = 0, \ \ \lambda < 0$$

where $\boldsymbol{e}$ is the eigenvector of the Hessian associated with the minimum eigenvalue λ. This condition means that f takes the local maximum along direction $\boldsymbol{e}$, which is often referred to as a height ridge in image processing [7, 14]. In geometric modeling, surfaces defined using such extremal conditions are generally called extremal surfaces [2].

A typical choice for f is the distance field from a solid boundary, but the skeletal sheet of a distance field is topologically too complicated (in terms of the number of branches) if the boundary surface is not smooth enough. Our simple choice for f is therefore an approximation of the intensity values of CT images, which are higher inside scanned objects. We also assume these images are almost noise free though our method can tolerate a certain amout of noise in real CT images as shown in the examples in Sec. 3.

For a smooth approximation of the input values assigned to 3D points, we use a partition of unity technique [23] consisting of a set of spherically supported quadratic polynomials. This approach is similar to that of sparse low-degree polynomial implicit surfaces [18], but local approximations are generated inside the solid as well as on the boundary surface since we want to find the object's internal characteristics. Our approximation with adaptively supported polynomials has the following benefits:

- Adaptive smoothing : Our error-driven support-size decision has a smoothing effect on scanning noise while preserving important geometric features. Smoothing with a fixed support size (such as Gaussian convolution of the image [15]) often leads to over/undersmoothing.

- Adaptive grid generation : Regular polygonization of skeletal sheets is performed on a uniformly sampled cubical grid [11]. In contrast, the version used in our polygonization is an adaptively sized tetrahedral grid obtained from the spherical supports of local approximations. Polygonization using the adaptively sampled grid gives us adaptive sampled skeletal meshes.

2. Algorithm

2.1. Algorithm overview

We take a set of spatial points equipped with normalized values $\mathscr{P} = \{\boldsymbol{p}_i = (\boldsymbol{x}_i, v_i) | \boldsymbol{x}_i \in \mathbb{R}^3, v_i \in [0, 1]\}$ as input, such as CT scan data and grayscale images. Fig. 2 (a) shows a 2D analogous example. A set consists of two kinds of point, i.e. those belonging to the object and those that make up the background. We assume that points with a value greater than the user's specification T represent the object. Taking T as input is reasonable because each material has a unique CT value. These points are referred to as *object points*, and are described as $\mathscr{P}_{\text{obj}} = \{\boldsymbol{p}_i | v_i > T\}$.

First, our algorithm constructs a set of locally supported quadratic functions that approximate the values of $\boldsymbol{p} \in \mathscr{P}$. A support is a sphere centered at a randomly selected point among $\boldsymbol{p}_i \in \mathscr{P}_{\text{obj}}$, and its radius is adapted to the distribution of values in the vicinity of the center (Fig. 2 (b)). The number of generated functions is much lower than the number of input points. We explain details in Sec. 2.2.

Next, a tetrahedral mesh representing the object is generated by connecting the centers of the function supports, as a 2D version (triangular mesh) is shown in Fig. 2 (c). The connectivity of this mesh depends on the adjacency of the supports. More details are described in Sec. 2.3.

Finally, the algorithm extracts the skeletal mesh using the tetrahedral mesh as a grid. The results for 2D data are shown in Fig. 2 (d). Points on the skeletal sheet are detected on mesh edges. In this step, we use the gradient and Hessian of the functions, which are estimated well with the help of the function set. If a skeletal point is detected on an edge, a small patch with vertices inside the tetrahedra incident to the edge is generated around the edge. The details are given in Sec. 2.4.

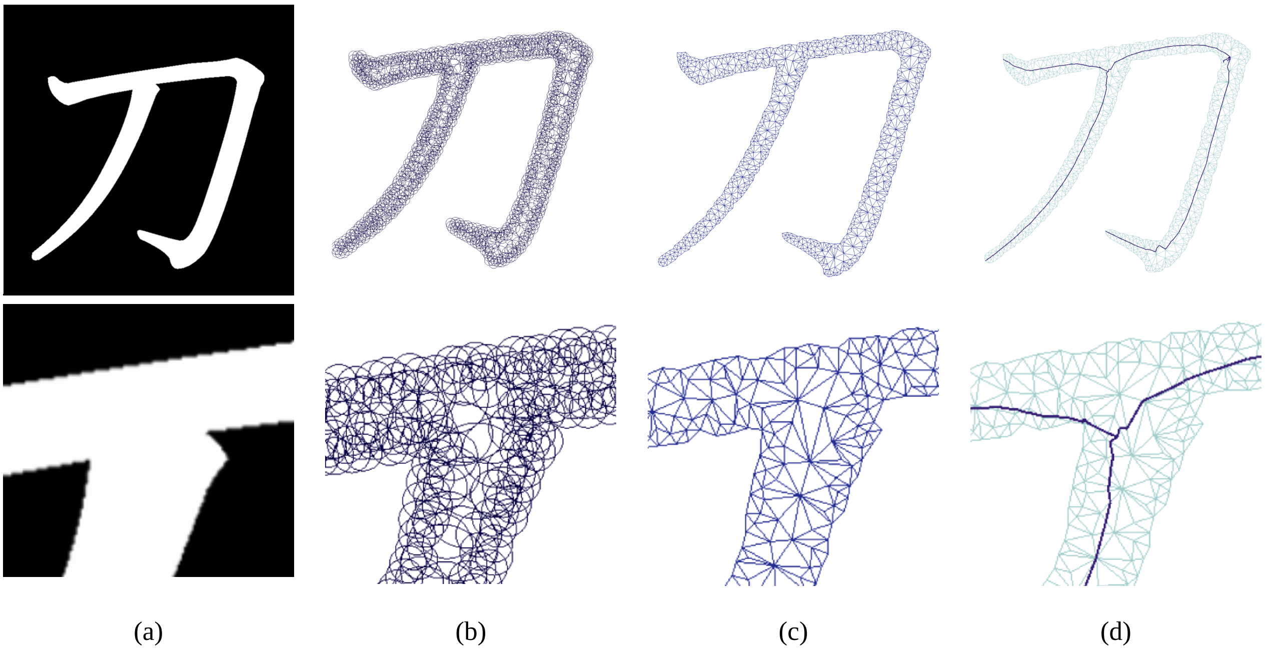

Figure 2. (a) An input, a 650×650 binalized image of a Chinese character. The number of $\mathscr{P}_{\mathrm{obj}}$ is $61,300$. (b) The support circles of approximation functions. The total number of functions is 919. (c) The triangular mesh of (b). (d) The skeletal structure. Nonmanifold parts are well detected.

2.2. Approximation

A set of spherically and adaptively supported functions is generated to approximate the values of $\mathscr{P}$. The approximation function is represented as a partition of unity based on polynomial approximations[23], i.e.

$$f(\boldsymbol{x}) = \sum_i \phi_i(\boldsymbol{x}) g_i(\boldsymbol{x}),$$

where $\boldsymbol{x} = (x, y, z) \in \mathbb{R}^3$. Expressing a radius and a center of a support in r_i and $\boldsymbol{c}_i$, weight function $\phi_i(\boldsymbol{x})$ is described as

$$\phi_i(\boldsymbol{x}) = \frac{w_{r_i}(\|\boldsymbol{x} - \boldsymbol{c}_i\|)}{\sum_j w_{r_j}(\|\boldsymbol{x} - \boldsymbol{c}_j\|)},$$

$$w_{r_i}(d) = \begin{cases} \frac{1}{(2\pi)^{\frac{2}{3}} r_i^2} e^{-\frac{d^2}{2r_i^2}} & (d \le r_i) \\ 0 & (otherwise) \end{cases}$$

and $\sum_i \phi_i(\boldsymbol{x}) = 1$. This truncation of w_{r_i} causes a discontinuity of $f(\boldsymbol{x})$. In order to make $f(\boldsymbol{x})$ continuous, cubic B-Spline may be a good candidate $w_{r_i}(d)$. We experimented a B-Spline version but could not find discernible differences of the quality of results.

We use a second-order equation as a polynomial g_i,

$$\begin{aligned} g_i(\boldsymbol{x}) =& c_{xx}x^2 + c_{yy}y^2 + c_{zz}z^2 + c_{xy}xy + c_{yz}yz \\ &+ c_{zx}zx + c_x x + c_y y + c_z z + c_0 \end{aligned}$$

where each coefficient is in $\mathbb{R}$.

The generation of a set of local approximations in f is summarized as below.

1. Set $\mathscr{P}_{\mathrm{obj}}$ as a center candidate list denoted by $\mathscr{C}$. Mark all points in $\mathscr{C}$ as *uncovered*.

2. Select a point $\boldsymbol{p}_i$ randomly from $\mathscr{C}$ and decide the radius and function (more details are given below).

3. Remove some points from $\mathscr{C}$ (details below). If $\mathscr{C} = \emptyset$ then the algorithm terminates, otherwise go back to step 2.

Decision of radius and function. We generate supports following the error-driven approach proposed in [20]. Original CT data include noises therefore using approximated data is better. To determine the support radius with noise tolerance ε we solve Eq. (1) with ε of 5% of the range of the obtained CT scanned value.

$$E_i(r_i) = \varepsilon \tag{1}$$

where

$$E_i(r)^2 \equiv \sum_j \phi_j(\boldsymbol{x})(g_i(\boldsymbol{x}_j) - v_j)^2 \Big|_{g_i(\boldsymbol{x}) = \underset{g_i}{\operatorname{argmin}} E_i^2(r)}.$$

We sum over the values of points whose distances to the center are less than or equal to r. $g_i(\boldsymbol{x})$ is a minimizer of $E_i(r)$.

Under the assumption that the approximation error monotonically increases, Eq. (1) is solved by a bisection method [21]. This equation is a means to maximize the radius of the support without exceeding the error tolerance.

We describe the minimum and maximum radius as $r_{\min}$ and $r_{\max}$ respectively. Setting a large value for r_{max} can decrease the number of supports. We set $r_{\max}$ to the maximum thickness of the object which is typically obtained by the scanned object itself. If thickness cannot be known, a user can estimate it by a distance transformation of the inside of the object. $r_{\min}$ should be small enough with keeping for the set of supports to be a covering of the object.

The intensity changes rapidly on boundaries, so circles whose centers are near a boundary tend to have small radii. In contrast, for centers that are placed far enough from the boundary, the corresponding radii become large because the intensity changes slightly or may not change in its support.

In the Fig. 3, a part of a binalized object is shown toward the lower left. The radii of the small green circles are the minimum radius $r_{\min}$, while the large circles drawn in blue dotted lines have the maximum radius $r_{\max}$. The top graph shows the change of intensity in the horizontal cross section, while the one on the right is for the vertical. The cross sections are drawn with red lines in the object figure. The original intensities (the gray lines in the two graphs) are step functions for this binalized object. The approximated intensities shown with black lines have Gaussian-like shapes.

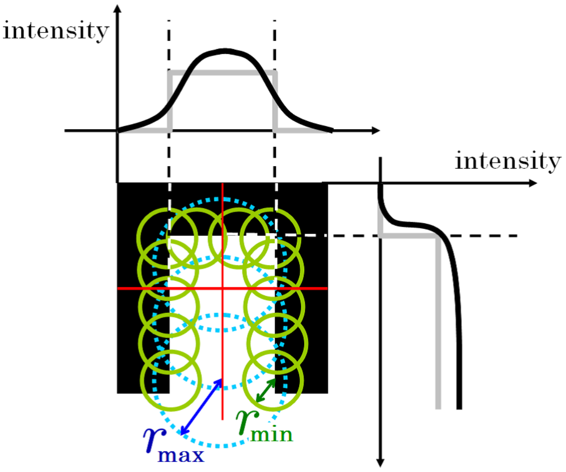

Figure 3. Lower left: a part of an object and supports. Top and right: original (gray) and approximated (black) intensities.

Covering. To generate a covering of $\mathscr{P}_{\mathrm{obj}}$, we re-mark some points inside the support sphere as *covered*. In 2D, a similar method is proposed in [24], and we extend it to 3D. The points to be re-marked as *covered* are those inside the convex hull defined by the object points in this support. In the image on the left of Fig. 4, we show a state after a new support (red) is generated. Supports already existing before this adding are drawn in black. Gray points are already covered by existing supports because these are inside the convex hull (blue) defined by the set of points included in each support. The red points are included inside the orange convex hull, and are newly marked as *covered*. Points not yet covered are shown in black. In the next step, the center of the new support will be selected from among the black points.

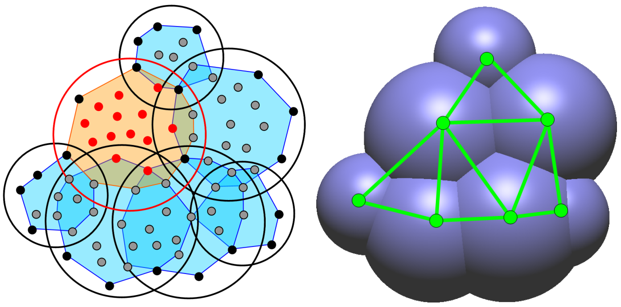

Figure 4. Left: changes in the status of points after a new 2D support (red) is added. Right: the one dimension raised supports (spheres) and the triangulation corresponding to the support circles in the image on the left.

2.3. Tetrahedrization

After generating the supports, we obtain a tetrahedrization by connecting their centers. An example for a 2D case is shown in the image on the right of Fig. 4 and a 3D case is in Fig. 5.

The tetrahedral meshes used in our method are a subset of weighted Delaunay tetrahedrization by the definition of [8]. The weighted Delaunay triangulation of $\mathscr{P}$ is defined by the dual of the weighted Voronoi diagram which uses a weighted distance

$$d_s(\boldsymbol{p}) = d(\boldsymbol{p}, \boldsymbol{s})^2 - w_s^2$$

for the distance between a point $\boldsymbol{p}$ and a site $\boldsymbol{s}$ whose weight is w_s instead of the Euclidean distance. In our method, w_s is the radius of the support whose center is $\boldsymbol{s}$. Since we need only a mesh representing the object, it is sufficient to mesh only the domain covered by supports. More details are described in the Appendix.

The image on the left of Fig. 5 is a part of generated supports. The number of all supports is $44,095$. The corresponding part of tetrahedral mesh is shown in the right side. The number of vertices and tetrahedra are $44,084$ and $226,474$ respectively.

The image on the left of Fig. 6 shows a triangulation generated using the above strategy. In this triangulation, some skinny elements (with a low aspect ratio) appear around the boundaries of the object. This is because some small spheres are absorbed by larger spheres, meaning that their centers cannot appear as vertices.

The mesh quality can be improved by smoothing the size changes of elements. Multiplying all radii of supports by $\alpha < 1$ and using these new radius values facilitates the avoidance of small sphere intersections being included in other spheres. The result is that more vertices appear in the generated mesh. Good results are obtained using $\alpha = 0.75$ in all examples in this paper. Such scaling enables a better quality of tetrahedrization (see the right-hand part of Fig. 6).

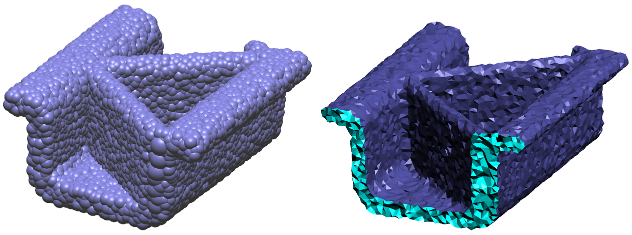

Figure 5. Left: the supports for the same input as Fig. 1. Right: the tetrahedral mesh generated from the supports on the left.

2.4. Skeletal mesh extraction

To obtain a skeletal mesh, we first check each edge of the tetrahedrization to see whether a crossing point with a skeletal sheet exists. If such a point is detected, a small patch is created around the edge. The set of patches created by this process becomes the skeletal mesh.

Crossing points. Let p_1, p_2 be the end points of an edge and $\boldsymbol{x}_1, \boldsymbol{x}_2$ the coordinates. See the image on the left of Fig. 7. Using polynomial approximation functions is a powerful method of estimating quality gradients and curvatures. To extract a smooth skeletal mesh, we use the first- and second-order derivatives:

$$\boldsymbol{g} = \nabla f, \;\; H = \nabla \boldsymbol{g}.$$

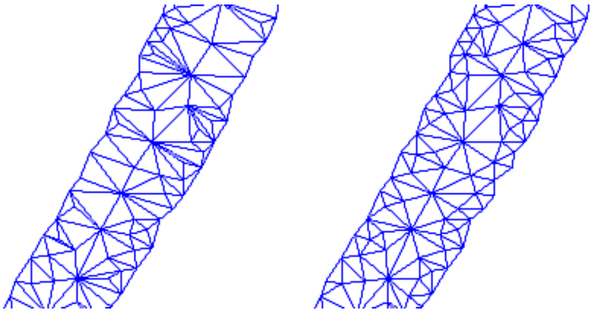

Figure 6. Left: many skinny triangles appear along the boundary. Right: by shrinking the support sizes, some points appear as mesh vertices and the size of the triangles changes slightly.

Let the eigenvectors corresponding to the minimum eigenvalues of Hessian H at the vertices p_1, p_2 be $\boldsymbol{e}_1, \boldsymbol{e}_2$. We assume the inner product $\langle \boldsymbol{e}_1, \boldsymbol{e}_2 \rangle > 0$. If this is not true, flip $\boldsymbol{e}_2$ to make $-\boldsymbol{e}_2$ and satisfy this condition. The existence of zero-crossing of $\langle \boldsymbol{e}, \boldsymbol{g} \rangle$ is tested by the condition

$$\langle \boldsymbol{e}_1, \boldsymbol{g}_1 \rangle \langle \boldsymbol{e}_2, \boldsymbol{g}_2 \rangle < 0.$$

Next, for edges that satisfy the above condition, we test whether the extremum is the maximum or minimum. As skeletal sheets correspond to maximum extrema, we need only the crossing points where the skeletal sheet corresponds to the maxima. We use a condition proposed in [19] for this test: if the condition

$$\langle \boldsymbol{e}_i, \boldsymbol{g}_i \rangle \langle (\boldsymbol{x}_j - \boldsymbol{x}_i), \boldsymbol{e}_i \rangle > 0$$

holds for $(i, j) = (1, 2)$ or $(2, 1)$, the edge has a crossing point to be obtained. If a maxima exists between these endpoints the values of f increase toward it along e, so above condition holds. This condition cannot decide the maximality if an edge is perpendicular to $\boldsymbol{e}_i$, so we test this condition on the both endpoints of the edge. By assuming that $\langle \boldsymbol{e}, \boldsymbol{g} \rangle$ changes linearly along the edge, the coordinates of the crossing point are calculated by

$$\frac{|\langle \boldsymbol{e}_2, \boldsymbol{g}_2 \rangle|}{|\langle \boldsymbol{e}_1, \boldsymbol{g}_1 \rangle| + |\langle \boldsymbol{e}_2, \boldsymbol{g}_2 \rangle|} \boldsymbol{x}_1 + \frac{|\langle \boldsymbol{e}_1, \boldsymbol{g}_1 \rangle|}{|\langle \boldsymbol{e}_1, \boldsymbol{g}_1 \rangle| + |\langle \boldsymbol{e}_2, \boldsymbol{g}_2 \rangle|} \boldsymbol{x}_2.$$

Skeletal patches. Next, a skeletal patch is made around crossing point c as shown in the image on the right of Fig. 7. The vertices of the patch are points inside the tetrahedra incident to the edge where c is found. A vertex is the centroid of all crossing points on the edges belonging to each

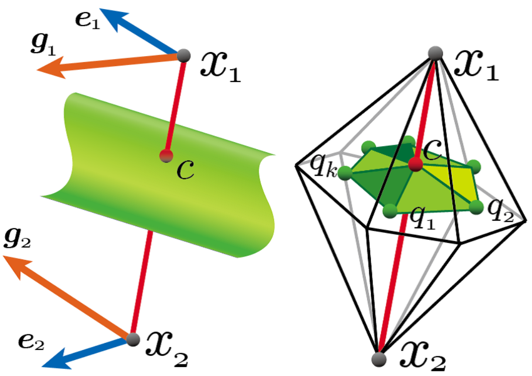

Figure 7. Left: an edge intersecting with a skeletal sheet (green). Right: a skeletal patch around the red edge.

tetrahedron, and a tetrahedron incident to an edge that has a crossing point also has at least one crossing point. After all vertices have been obtained, a patch is made by traversing all the incident tetrahedra. For a sequence of k vertices $q_1, q_2, \ldots, q_k$, we split this patch into k triangles sharing the crossing point c, $\{q_1, q_2, c\}, \{q_2, q_3, c\}, \ldots, \{q_k, q_1, c\}$. This procedure is a tetrahedron-based variation of the dual contouring proposed in [13].

Smoothing. To obtain well-connected and not-fragmented skeletal patches, smoothing the fields of derivatives is an effective choice. We can easily obtained such a smoothing effect by enlarging the support sizes as $\{\sigma\, r_i\}$ $(\sigma > 1)$ before evaluating $\boldsymbol{g}$ and H, which means the local approximations are spread out more large regions. The all results demonstrated in this paper are obtained by $\sigma = 4$.

3. Results and Discussion

Experimental results. Some results for CT scan data are shown in Figs. 8, 9 and 10. Each of extracted skeletal meshes reaches to the ends of a thin object without branching. Fig. 8 gives the results for objects with uniform and nonuniform thickness. Our algorithm works well for both examples. The object in Fig. 10 has a noisy boundary. Our algorithm extracts a smooth skeletal mesh from such data.

Comparison with image processing approaches. Since input data is a CT image, it is also possible to extract skeletal voxels using a conventional image processing approach. Given a certain value and direction at each voxels, non-maximum suppression [3] is one of the most popular method for finding the voxels taking local maximum value along the directions. For extracting skeletal voxels, we use the standard Gaussian convolution of CT values to obtain the values f and direction $\boldsymbol{e}$ at each voxel [15].

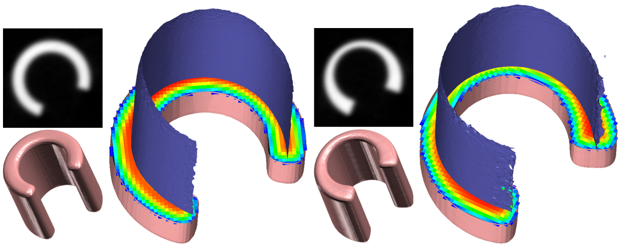

Figure 8. Left: a CT scanned object with a uniform thickness. Right: a CT scanned object with a non-uniform thickness. The coloring on the half cross section shows the input CT values (red for high and blue for low).

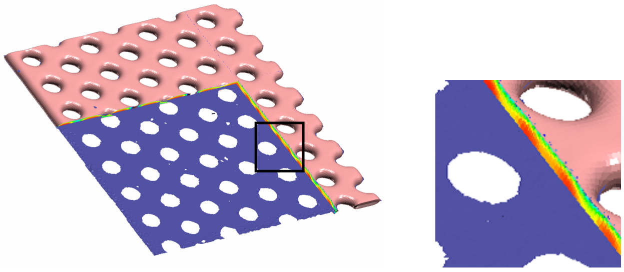

Figure 9. Skeletal mesh for a perforated metal plate.

As shown in the top images of Fig.11, the image processing approach works well for the model with a uniform thickness if we choose a proper kernel size of Gaussian convolution. However, it is hard to obtain a good result for the model with a non-uniform thickness because the fixed kernel size causes under-smoothing (fail to eliminate unwanted branches) or over-smoothing (skeletal voxels go out of the object). See Fig.12 for an example of such a problem. In contrast, our method automatically adjusts the support sizes of local approximations based on the error analysis Eq. 1; thus we can obtain a proper smoothing effect adaptively.

To obtain a skeletal mesh from extracted voxels, we have to apply a surface reconstruction method to connect the centers of the voxels with polygonal patches. The bottom images of Fig.11 show triangular meshes obtained from the skeletal voxels. Since the mesh vertices are located on the

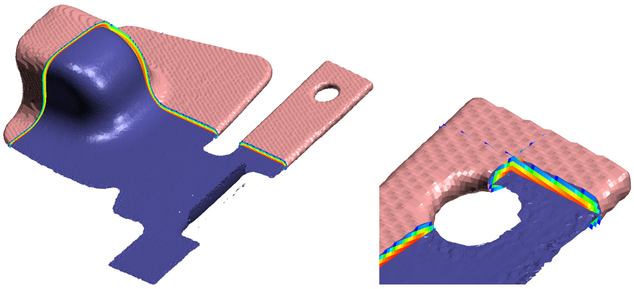

Figure 10. Skeletal mesh for a thin plate with a hole. The bumps on the surface of the original object do not affect the result extrema.

grid points, the meshes are not smooth enough unlike the meshes obtained by our method. Further, some of surface reconstruction methods are fail to reconstruct meshes because the grid artifact behaves as a large noise.

Compared with the Gaussian convolution of a 3D image, generating our polynomial approximation is an expensive task. For example, it takes about ten minutes on a standard PC for computing the result shown in Fig. 10 while about three minutes for Fig.11. This is the only drawback of our method against the non-maximum suppression with the Gaussian convolution. However, our method does not require a grid sampling structure to the input volume, thus it is possible to apply our method to scattered volume samples (irregularly sampled points with values) which can not be handled by image processing approaches.

4. Conclusion and Future Work

In this paper, we have proposed an algorithm to extract the skeletal meshes of CT-scanned thin objects using a set of approximation functions and an adaptive grid. This algorithm can generate a smooth surface mesh of the skeletal sheet directly from raw data. Using our data approximation strategy, the technique enables detection of smooth skeletal meshes that extend almost to the end of thin objects.

The improvement of crossing-point detection is another issue. Theoretically speaking, our algorithm can handle even T-junctions as long as edges that should have a crossing point are detected. But actually we observed that the derivatives around such parts are very unstable, so detection for non-manifold parts is still a difficult issue.

Acknowledgement

We thank Takashi Michikawa and Hiroshi Kawaharada for fruitful discussions.

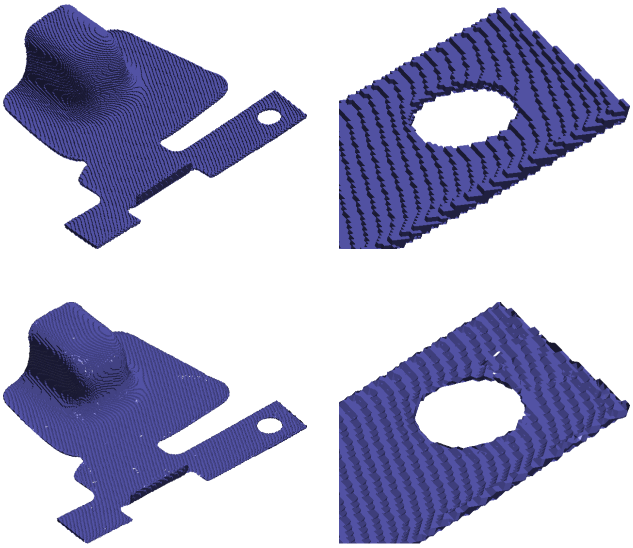

Figure 11. Top: skeletal voxels extracted by non-maximum suppression (the input CT image is the same as Fig. 10). Bottom: a triangular mesh reconstructed by connecting the skeletal voxels.

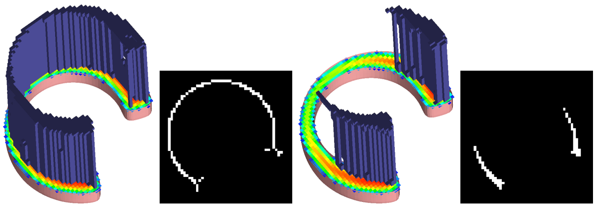

Figure 12. Skeletal voxels extracted by non-maximum suppression for a model with a non-uniform thickness (same as the right model in Fig.8).

References

[1] N. Amenta, S. Choi, and R. Kolluri. The power crust. In *Proceedings of 6th ACM Symposium on Solid Modeling*, pages 249–260, 2001.

[2] N. Amenta and Y. Kil. Defining point-set surfaces. *ACM Transactions on Graphics*, 23(3):264–270, 2004. Proceedings of ACM SIGGRAPH 2004.

[3] J. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[4] N. Cornea, D. Silver, X. Yuan, and R. Balasubramanian. Computing hierarchical curve-skeletons of 3d objects. *The*

Visual Computer, 21(11):945–955, Octber 2005.

[5] N. D. Cornea, D. Silver, and P. Min. Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization And Computer Graphics*, 13(3):530–548, May/June 2007.

[6] T. K. Dey and W. Zhao. Approximating the medial axis from the voronoi diagram with a convergence guarantee. *Algorithmica*, 38:179–100, 2003.

[7] D. H. Eberly. *Ridges in Image and Data Analysis*. Kluwer Academic, Dordrecht, Nederlands, 1996.

[8] H. Edelsbrunner. The union of balls and its dual shape. In *Proc. 9th ACM Sympos. Comput. Geom.*, pages 218–231, San Diego, California, United States, 1993.

[9] T. Fujimori, Y. Kobayashi, and H. Suzuki. Separated medial surface extraction from ct data of machine parts. *Lecture Notes in Computer Science*, 4077:313–324, 2006. Geometric Modeling and Processing - GMP 2006.

[10] T. Fujimori, H. Suzuki, Y. Kobayashi, and K. Kase. Contouring medial surface of thin plate structure using local marching cubes. *ASME Journal of Computing and Information Science in Engineering*, 5(2):111–115, 2005.

[11] J. D. Furst and S. M. Pozer. Marching ridges. In *Proceedings of the IASTED International Conference on Signal and Image Processing*, pages 22–26, 2001.

[12] T. Ju, M. Baker, and W. Chiu. Computing a family of skeletons of volumetric models for shape description. *Computer-Aided Design (accepted)*, 2007.

[13] T. Ju, F. Losasso, S. Schaefer, and J. Warren. Dual contouring of hermite data. *ACM Transactions on Graphics*, 21(3):339–346, July 2002. Proceedings of ACM SIGGRAPH 2002.

[14] T. Lindeberg. Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156, 1998.

[15] G. Lohmann. *Volumetric Image Analysis*. Wiley, 1998.

[16] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 163–169, 1987.

[17] W.-C. Ma, F.-C. Wu, and M. Ouhyoung. Skeleton extraction of 3d objects with radial basis functions. In *Shape Modeling International 2003*, pages 207–215, May 2003.

[18] Y. Ohtake, A. Belyaev, and M. Alexa. Sparse low-degree implicit surfaces with applications to high quality rendering, feature extraction, and smoothing. In *Proc. 3rd Eurographics / ACM SIGGRAPH Symposium on Geometry Processing*, pages 149–158. Eurographics Association, Aire-la-Ville, Switzerland, 2005.

[19] Y. Ohtake, A. Belyaev, and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics*, 23(3):609–612, 2004. Proceedings of ACM SIGGRAPH 2004.

[20] Y. Ohtake, A. G. Belyaev, and H.-P. Seidel. Multi-scale and adaptive CS-RBFs for shape reconstruction from cloud of points. In *MINGLE Workshop on Multiresolution in Geometric Modelling*, pages 337–348, Cambridge, UK, September 2003.

[21] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1993.

[22] S. Prohaska and H.-C. Hege. Fast visualization of plane-like structures in voxel data. In *Proceedings of the conference on Visualization*, pages 29–36, 2002.

[23] R. J. Renka. Multivariate interpolation of large sets of scattered data. *ACM Transactions on Mathematical Software*, 14(2):139–148, June 1988.

[24] J. Wu and L. P. Kobbelt. Optimized sub-sampling of point sets for surface splatting. *Computer Graphics Forum*, 23(3):643–652, 2004. Proceedings of EUROGRAPHICS 2004.

A Tetrahedral mesh generation

The adjacency of supports is used to decide the connectivity of mesh. The supports are spheres as seen in the previous section, but we raise their dimension by a factor of one and consider the adjacency of these 4D spheres that have the same radii and centers as the original 3D spheres. For example, an original sphere centered at $(c_x, c_y, c_z) \in \mathbb{R}^3$ is described as $(x-c_x)^2+(y-c_y)^2+(z-c_z)^2 = r^2$. This sphere will become $(x-c_x)^2+(y-c_y)^2+(z-c_z)^2+(u-0)^2 = r^2$. The centers are the same points as the original ones on the x, y, z space, and the connectivity is based on the adjacency of these 4D spheres. From here on in this section, the term sphere refers to an 4D sphere unless otherwise specified.

First, we select 4 spheres with intersections that are at most two points in 4D space. If an intersection point is not included in other spheres, we connect the centers of these 4 spheres to create an tetrahedron. As the centers have a value of zero in the fourth coordinate, the tetrahedron is in a 3D space. The triangulations gained by this one-dimensional increase method are subsets of weighted Delaunay triangulations.

As the vertices came from uniformly sampled grid points $\mathscr{P}$, in most cases the triangulations have a number of degenerate areas. To avoid this degeneracy, in intersection testing only we use coordinates perturbed $0.01 \times r_i$ in random directions from the original location rather than the original coordinates. Good results are obtained using this method.