Adaptive Meshing Based on the Multi-level Partition of Unity and Dynamic Particle Systems for Medical Image Datasets

Zhong Chen*, Zhiwei Hou, Quanquan Yang, Xiaobing Chen

Jiangsu Key Laboratory of Advanced Manufacturing Technology
Huaiyin Institute of Technology
Huaian, China
E-mails: chenzhong2006@126.com, zw_hou@163.com, qqyang@hyit.edu.cn, chenxiaobing@hyit.edu.cn

Corresponding author

Received: December 17, 2017 **Accepted: August 15, 2018**

Published: September 30, 2018

Abstract: *Surface meshes extracted from sparse medical images contain surface artifacts, there will produce serious distortion and generate numerous narrow triangle meshes. In order to eliminate the impact of the above factors, this paper presents a novel method for generating smooth and adaptive meshes from medical image datasets. Firstly, extracting the stack of contours by means of image segmentation and translating the contours into point clouds. The improved Multi-level Partition of Unity (MPU) implicit functions are used to fit the point clouds for creating the implicit surface. Then, sampling implicit surface through dynamic particle systems based on Gaussian curvature, dense particles sampling in the high curvature region, sparse particles sampling in the low curvature region. Finally, generating triangle meshes based on particle distribution by using the Delaunay triangulation algorithm. Experimental results show that the proposed method can generate high-quality triangle meshes with distributed adaptively and have a nice gradation of triangle mesh density on the surface curvature.*

Keywords: *Medical computed tomography, Point clouds, Multi-level partition of unity, Dynamic particle systems, Gaussian curvature.*

Introduction

Medical computed tomography (CT) scanner produced 2D CT sequential images, which containing a large number of spatial relation information of the tissues and organs of the human body. From the current situation of medical finite element (FE) simulation and 3D printing technology, the original 2D image can not be effectively used for surgical analysis and planning. It must be preprocessed to generate the model in higher dimensional space, which can be used for high-fidelity simulation and 3D printing. Therefore, we need for analysis and processing these 2D CT images, and reconstruct the 3D model of the organ and tissue, providing an intuitive means of technology for high level simulation and surgical planning, improving the effectiveness and accuracy of the diagnosis and treatment for doctors. However, medical imaging, it is possible to produce anisotropic data where the slice resolution in z-direction is commonly lower than the imaging resolution in the *x*-*y* plane. In other words, in-plane resolution is only accurate to the pixel, whereas slices can be more pixels apart from each other. Therefore, CT scanning will produce sparse sampling data because of missing data between CT slices [7]. Then surface meshes extracted from sparse CT images contain surface artifacts, which will produce a serious distortion with regard to 3D medical surface reconstruction.

At present, there are two main categories for 3D reconstruction of medical tissue and organs: Volume rendering and Surface mesh rendering. There are mainly three kinds of Volume rendering algorithm: ray-casting [17], cell projection [13] and the fast volume rendering algorithm [9]. The volume rendering method is shading processes to all of the volume data directly, maintains the details of the medical images, and achieves better render results. However, the virtual model formed by volume rendering can only be used for the naked eye observation, because of the meshes can not be generated, the method cannot be used for finite element analysis and 3D printing. Surface meshes rendering is further classified into three sub-categories:

(1) extracting iso-surface algorithm for regular volume data, e.g., marching cubes (MC) algorithm [12], marching tetrahedra (MT) algorithm [18]. The main drawback of the MC algorithm (or MT algorithm) is that it generates tremendously dense meshes and numerous narrow triangles.
(2) Contour-based surface mesh reconstruction. e.g., contour stitching algorithm [14]. The surface meshes are formed by stitching the vertices of adjacent contours. There is branching problem that needs to be resolved.
(3) Extracting explicit or implicit surface mesh algorithm from the point clouds, explicit method: e.g., Delaunay triangulation [4], advancing front triangulation [5], ball-pivoting algorithm [1], and so on. Implicit method: moving least squares (MLS) algorithm [10], level-set methods [20], radial basis function (RBF) algorithm [15], multi-level partition of unity (MPU) algorithm [16].

The comparison of the above methods, MPU algorithm is very suitable for large datasets in medical imaging, because it uses an adaptive octree-based subdivision of the space to conform to local detail features adaptively. Furthermore, the reconstruction approximation errors can be controlled through the preference settings. Most important, the MPU implicit algorithm can reconstruct closed smooth implicit surface for non-uniform density of point clouds (or incomplete sampled data). So that MPU algorithm can fit a smooth surface over missing data between CT slices. However, Ohtake et al. [16] proposed MPU algorithm, which uses quadratic surface function to compute the local surface, less robustness, and prone to produce spurious sheets. Moreover, Ohtake et al. [16] adopted Bloomenthal's method [2] for triangulation of implicit surfaces, but it generated numerous narrow triangle meshes.

In this paper, a novel strategy that efficiently reconstructs smooth, adaptive meshes from large medical image datasets is presented. Our method is based on improved MPU implicit functions and improved dynamic particle systems. An overview of our approach to adaptive triangle mesh reconstruction is illustrated in Fig. 1. As seen from Fig. 1, we begin with a stack of closed contours by image segmentation. Point clouds are extracted from the set of contours. Then, implicit surfaces are reconstructed from point clouds. The implicit surface is sampled by using dynamic particle system. Finally, triangle meshes are generated from the particles through the Delaunay triangulation algorithm.

Point clouds preprocessing

Contours extraction of CT images

In our implementation, medical images were segmented by using active contours without edges algorithm [3]. The algorithm contains the following advantages: (1) robustness to noise; (2) we can get a closed contour curve by image segmentation under any conditions. The key to the algorithm is to evolve an interface, which divides the image into two homogeneous regions.

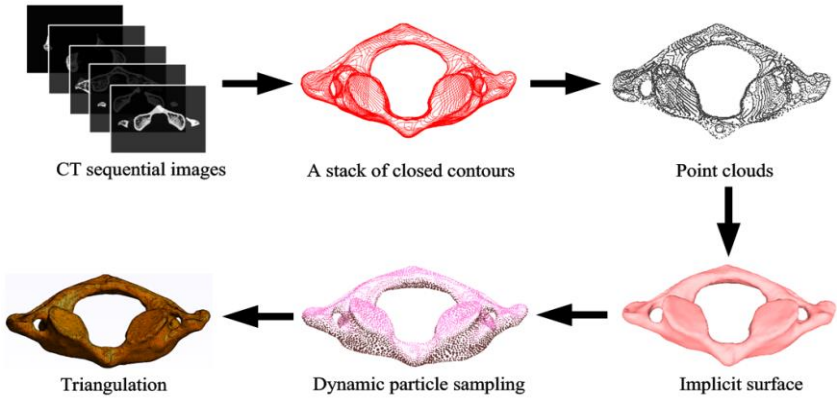


Fig. 1 Overview of the proposed method

This is the solution to minimize the energy functional $F(c_1, c_2, C)$, defined by:

$$F(c_1, c_2, C) = \mu Length(C) + \nu Area(insideC) + \lambda_1 \int_{inside(c)} |u(x,y) - c_1|^2 dxdy + \lambda_2 \int_{outside(c)} |u(x,y) - c_2|^2 dxdy \quad , \tag{1}$$

where C is equivalent to the closed curve, $u(x, y)$ is equivalent to the image, c_1 and c_2 are the average intensity of $u(x, y)$ inside and outside of C, respectively. Furthermore, μ, $\nu \geq 0$, λ_1, $\lambda_2 > 0$ are fixed parameters.

In this work, we set the parameters as: $\lambda_1 = 1$, $\lambda_2 = 7$ (bone) and $\mu = 0.1 \times 255^2$, $\nu \geq 0$, $\triangle t = 0.01$ ($\triangle t$ is the time step), for 100 iterations.

Point clouds extraction and normal estimation

To reconstruct medical implicit surface by implicit functions. There are required to obtain point clouds from CT images firstly [11]. We extract contours from CT sequential images, a single pixel of the contour as x, y coordinates in a 2D plane, calculating the z coordinate according to the slice number. Finally, we acquire 3D point coordinates and the contour points translate into 3D point clouds $P = \{p_i \in R^3\}_{i=1}^n$. The normal n_i of each point p_i can be estimated through Principal Component Analysis (PCA) [6].

Medical implicit surface reconstruction

MPU implicit functions

Given medical point clouds $P = \{p_i \in R^3\}_{i=1}^n$ equipped with the normal n_i of each point p_i, let adopting the octree subdivision process to adapt to local detail features adaptively. The point clouds P are divided into m overlapping subsets. Each subset is included within a sphere with a support radius of R_i. In each sphere, the corresponding local approximation functions $Q_i(\mathbf{x})$ are fitted by the least square method. In fact, the MPU method defines a global implicit function $f(\mathbf{x})$, which is consist of overlapping local approximation functions $Q_i(\mathbf{x})$ that are blended together by a partition of unity functions $\varphi_i(\mathbf{x})$ that sum up to 1 on a finite Euclidean domain Ω, as shown in Fig. 2.

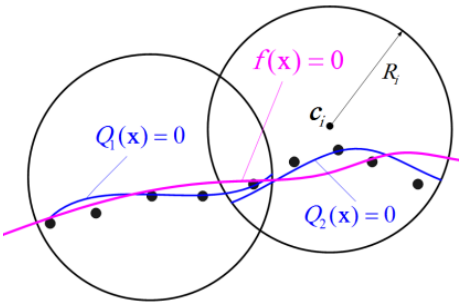


Fig. 2 A global implicit function $f(\mathbf{x})$ is consisting of overlapping local approximation functions $Q_i(\mathbf{x})$ that are blended together

A global implicit function is defined by:

$$f(\mathbf{x}) = \sum_{i=1}^{m} \varphi_i(\mathbf{x}) Q_i(\mathbf{x}) \,,\ \mathbf{x} \in \Omega \ , \tag{2}$$

where a partition of unity function $\varphi_i(\mathbf{x})$ is computed as:

$$\varphi_i(\mathbf{x}) = \frac{w_i(\mathbf{x})}{\sum_{j=1}^{m} w_j(\mathbf{x})} \,,\ \sum_{i=1}^{m} \varphi_i(\mathbf{x}) \equiv 1 \,. \tag{3}$$

In order to approximation, a quadratic B-spline $b(t)$ is used to denote each weight function $w_i(\mathbf{x})$, which centered at c_i in the subdivision, and have a spherical support of radius R_i :

$$w_i(\mathbf{x}) = b(t_i(\mathbf{x})) \ \text{, where} \ \ t_i(\mathbf{x}) = \frac{3|\mathbf{x} - \boldsymbol{c}_i|}{2R_i} \,. \tag{4}$$

According to different needs, the local implicit functions $Q_i(\mathbf{x})$ have three forms [16]: (a) a general 3D quadric for fitting large-area surface or unbounded surface, as shown in Eq. (5); (b) a bivariate quadric polynomial in local coordinates for fitting a local smooth region, as shown in Eq. (6); (c) a piecewise quadric surface that fits an edge or a corner for the reconstruction of sharp features. The function $Q_i(\mathbf{x})$ is chosen based on the number of points and normals within the sphere.

The local approximation functions $Q_i(\mathbf{x})$, defined by:

$$Q_i(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x} + \mathbf{b}^{\mathrm{T}} \mathbf{x} + c \,, \tag{5}$$

where $\mathbf{A}$ is a symmetric 3×3 matrix, $\mathbf{b}$ is a space vector, and c is a coefficient.

$$Q_i(\mathbf{x}) = Q_i(u,\, v,\, w) = w - (a_1 u^2 + 2a_2 uv + a_3 v^2 + a_4 u + a_5 v + a_6) \,, \tag{6}$$

where (u, v, w) are local coordinates, $a_1, \ldots, a_6$ are unknown coefficients.

Improved MPU implicit functions

In the natural form of internal bones or organs do not have sharp features, so we need not consider the sharp features reconstruction, only the general 3D quadric (the first case) and the bivariate quadric polynomials (the second case) are utilized in our work. The original MPU algorithm can reconstruct 3D medical model, but the reconstructed model surface is not smooth and prone to produce spurious sheets. To improve the surface smoothness, we need to increase the order of the local fitting function. For the regular polynomial, the regular equation is often ill conditioned equation when in higher order condition. It cannot reconstruct the desired surface.

To solve the above problems, in our work, the orthogonal polynomials as the local fitting function instead of regular polynomial, and the polynomial order can adjust adaptively based on the actual situation in the local fitting process, and does not require the solution of equations, this strategy can avoid the ill conditioned equations.

Defining local coordinates (u,v,w) for each sub region, the center of the coordinates is the support center, (u,v) for the 2D local coordinates, w is the 2D orthogonal direction. Constructing orthogonal polynomial basis, let

$$\beta_1 = 1,\ \beta_j = x_j\beta_{j-1} - \sum_{n-1}^{j-1} b_{j,n}\beta_n\ ,\ j = 1,\ 2,\ldots, \tag{7}$$

where $b_{j,n} = \dfrac{\langle x_j\beta_{j-1}, \beta_n\rangle}{\langle \beta_n, \beta_n\rangle}$, $n = 1,\ 2,\ldots,\ j-1$.

Calculating the orthogonal polynomials in the local coordinate system by using the orthogonal polynomial basis.

Due to

$$h_i(u,\ v) = \sum_{\beta_j \in \beta} \beta_j(u,\ v)\frac{\sum_{i=1}^{m} w_j\beta_j(u_i,\ v_i)}{\sum_{i=1}^{m} \beta_j(u_i,\ v_i)\beta_j(u_i,\ v_i)}, \tag{8}$$

we have

$$Q_i(\mathbf{x}) = Q_i(u,v,w) = w - h_i(u,v)\,, \tag{9}$$

where (u,v,w) are the coordinates of **x** in the new coordinates.

We use the Eq. (9) instead of the original Eq. (6) in our implementation.

Adaptive meshing of implicit surfaces

Dynamic particle sampling based on Gaussian curvature

The dynamic particle system is one of the most used methods for implicit surface rendering, the implicit surface is sampled by using particle systems, according to the sample points to construct meshes to approximate the implicit surface, sampling quality directly affect the final quality of surface. Witkin et al. [19] proposed a method based on dynamic particle systems for

sampling implicit surface, which constrain a set of particles on the surface, the repulsion force between the particles so that the particles are uniformly distributed on the surface, so as to achieve a uniform surface sampling. This uniform sampling can be a good triangulation result for simple surfaces, but it is not so practical for the complex medical surface model.

In order to get more reasonable sampling particles for complex surface, making in the region with dense particles sampling in the high curvature region, sparse particles sampling in the low curvature region, we introduce Gaussian curvature as the sampling constraints. Arbitrary points of the principal curvatures k_1 and k_2 can be calculated using Kindlmann's method [8]. We use Gaussian curvature $K = k_1 k_2$ as a standard to measure the changes of surface. The definition of repulsion energy is only related to the distance between particles. The Gaussian curvature *K* is introduced to particle sampling. It is desirable that smaller energy of particles in the high curvature region, so that the distance between the particles is smaller. Similarly, the larger repulsion energy of particles in the low curvature region, the distance between the particles is larger.

Therefore, we define a weight coefficient to determine the size of repulsion energy function for each particle. The size of weight coefficient is inversely proportional to the repulsion energy between the particles.

The weight coefficients λ_i defined are as follows:

$$\lambda_i = \exp\left(-\left|K_i\right|\right). \tag{10}$$

The new repulsion energy of the particle *i* after introducing a weight coefficient is defined as:

$$E^i = \lambda_i \cdot \sum_j \left(E^{ij} + E^{ji}\right) = \lambda_i \cdot \left(\mu \exp\left(-\frac{\left|r^{ij}\right|^2}{2(\sigma^i)^2}\right) + \mu \exp\left(-\frac{\left|r^{ij}\right|^2}{2(\sigma^j)^2}\right)\right), \tag{11}$$

where E^{ij} is the repulsion energy between particle *i* and particle *j* and $E^{ij} \neq E^{ji}$ because of $\sigma^i \neq \sigma^j$, μ is the repulsion parameter, $\left|r^{ij}\right|$ is the distance between the particles, σ^i is repulsion radius with the change according to the local energy of the particle.

The velocity of particle movement v^i by using the derivative of repulsion energy is:

$$v^i = -\lambda_i \cdot \left(\sigma^i\right)^2 \sum_{j=1}^{n} \left(\frac{r^{ij}}{(\sigma^i)^2} E^{ij} - \frac{r^{ij}}{(\sigma^j)^2} E^{ji}\right). \tag{12}$$

The only value for repulsion radius of each particle is $\sigma_{Witkin} = 0.3\sqrt{S/N}$, where *S* is the surface area, *N* is the number of particles in Witkin's method. In this paper, we set a threshold value σ_i for each particle's repulsion radius, so that the particles can be fissioning or killing particles dynamically depending on the size of the curvature $\sigma_i = \sigma_{Witkin} / \lambda_i$. The obvious change of surface curvature in the region of the particles, σ_i will become larger. At this time, the particle will continue to fission, produce smaller and more intensive distributed particles; otherwise it will stop fissioning.

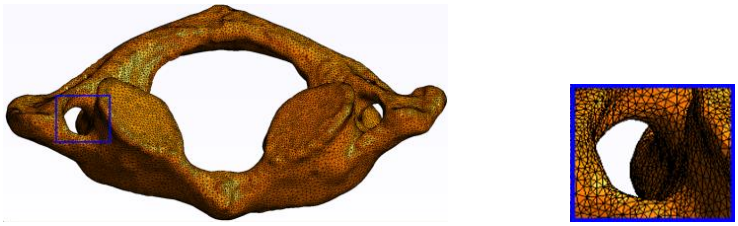
Triangulation

In order to triangulate a distribution of particles, we need geometry information about particles. The position of each particle is denoted as p^i, and the normal vector $N_i(p^i)$ of the sample particle is calculated by the gradient $\nabla f\ (p^i)$. Finally, a Voronoi diagram based on the triangle mesh generation algorithm [4] is utilized to generate triangle meshes.

Experimental results

In this section, we show the results of our experiments with several methods. The proposed method has been implemented in VC ++ 6.0. Experiments were performed on a PC with a 2.53 GHz Intel Core (TM) i3 CPU and 2.0 GB of RAM. We evaluated our approach based on several medical CT datasets that include human first cervical vertebrae (C1 for short) and scapula (the large medical image datasets used in this experiment are from the Laboratory of Human Anatomy and Embryology, University of Brussels (ULB), Belgium). We used medical CT image slices of C1, as shown in Fig. 1 (Top left), with an anisotropic voxel size of 0.352×0.352×0.5 mm. In our implementation, we set diameter factor $\alpha = 0.75$, $N_{\min} = 30$ (the minimum number of points needed to be extracted in the subdivision) and radius factor $\lambda = 0.1$ in the process of implicit surface reconstruction. At dynamic particle sampling stage, we set time step $\Delta t = 0.3$, repulsion parameter $\mu = 6$, equilibrium speed $\gamma = 4$. The experimental results showed in Fig. 3 and Fig. 4.

(a) Dynamic particle sampling based on Gaussian curvature



(b) Meshing distribution of particles by Delaunay triangulation

Fig. 3 Dynamic particles on the C1 model and the resulting triangulation

In Fig. 3a, a C1 implicit surface model has been sampled by means of dynamic particle sampling based on Gaussian curvature. The C1 model is sampling with 10103 particles. As shown zoomed region of C1 model, there are dense particles sampling in the high curvature region, sparse particles sampling in the low curvature region. Finally, generating high-quality triangle meshes from the particles through Delaunay triangulation algorithm [4], as shown in Fig. 3b. Similarly, the scapula model is sampling with 25419 particles. We compare our results with MPU method, and the marching cubes algorithm (see Fig. 4). The MPU method can be generated numerous narrow triangle meshes and prone to generate spurious sheets. The triangle mesh density has achieved the voxel level by means of MC algorithm, the number of meshes is considerable. In addition, these meshes contain numerous narrow triangles. Our method can make the best of the characteristics of particle distribution in the process of triangulation. The high-quality triangle meshes can be distributed adaptively and have a nice gradation of

triangle mesh density on the surface curvature.

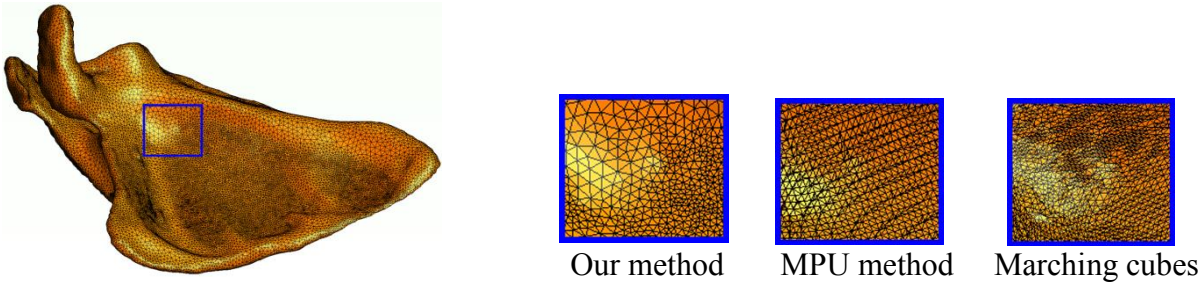


Fig. 4 The scapula mesh model is generated by our method.
Three zoomed regions of the scapula mesh models are generated using our method, the MPU method, and the Marching Cubes algorithm separately.

Conclusions

A novel method for generating smooth and adaptive meshes from large medical image datasets is proposed. Our method is a multi-stage process. Firstly, we extract the stack of contours by means of image segmentation. The contours translate into point clouds. The normals of each point can be estimated through PCA. Then, an improved MPU implicit function is applied to reconstruct the implicit surface. The implicit surface is sampled using dynamic particle sampling based on Gaussian curvature. Finally, the Delaunay triangulation algorithm is implemented to generate triangle meshes. Our method has been used for CT image datasets of C1 and scapula. Experimental results show that the proposed method can make the best of the characteristics of particle distribution in the process of triangulation and generate high-quality triangle meshes with distributed adaptively. In a further work, we will develop a mesh optimization algorithm for our method.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No. 51705184), and the Natural Science Foundation of Jiangsu Higher Education Institution (No. 15KJA460003).

References

1. Bernardini F., J. Mittleman, H. Rushmeier, C. Silva, G. Taubin (1999). The Ball-pivoting Algorithm for Surface Reconstruction, IEEE Transactions on Visualization and Computer Graphics, 5(4), 349-359.
2. Bloomenthal J. (1994). An Implicit Surface Polygonizer, Graphics Gems IV, 324-349.
3. Chan T. F., L. Vese (2001). Active Contours without Edges, IEEE Transactions on Image Processing, 10(2), 266-277.
4. Dey T. K., S. Goswami (2003). Tight Cocone: A Water-tight Surface Reconstructor, SM'03 Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, Seattle, USA, 127-134.
5. Foucault G., J. C. Cuillière, V. François, J. C. Lèon, R. Maranzana (2013). Generalizing the Advancing Front Method to Composite Surfaces in the Context of Meshing Constraints Topology, Computer-aided Design, 45(11), 1408-1425.
6. Hoppe H., T. DeRose, T. Duchamp, J. McDonald, W. Stuetzle (1992). Surface Reconstruction from Unorganized Points, SIGGRAPH'92 Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques, 71-78.
7. Hossain A. B. M., M. H. Cho, S. Y. Lee (2015). Study on Compression Induced Contrast in

X-ray Mammograms Using Breast Mimicking Phantoms, International Journal Bioautomation, 19(3), 385-396.
8. Kindlmann G., R. Whitaker, T. Tasdizen, T. Moller (2003). Curvature-based Transfer Functions for Direct Volume Rendering: Methods and Applications, Proceedings of IEEE Visualization, Seattle, USA, 513-520.
9. Lacroute P., M. Levoy (1994). Fast Volume Rendering Using a Shear-warp Factorization of the Viewing Transformation, SIGGRAPH'94 Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, Orlando, USA, 451-458.
10. Levin D. (1998). The Approximation Power of Moving Least-squares, Mathematics of Computation of the American Mathematical Society, 67(224), 1517-1531.
11. Li X., K. Wu, Y. Ma (2016). Quantitative Analysis of Geometric Structures and Experimental Evaluation of Rooster Beak, International Journal Bioautomation, 20(2), 205-214.
12. Lorensen W. E., H. E. Cline (1987). Marching Cubes: A High Resolution 3D Surface Construction Algorithm, SIGGRAPH'87 Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, USA, 163-169.
13. Ma K. L., T. W. Crockett (1997). A Scalable Parallel Cell-projection Volume Rendering Algorithm for Three-dimensional Unstructured Data, IEEE Symposium on Parallel Rendering, Phoenix, USA, 95-104.
14. Meyers D., S. Skinner, K. Sloan (1992). Surfaces from Contours, ACM Transactions on Graphics, 11(3), 228-258.
15. Ohtake Y., A. Belyaev, H. P. Seidel (2003). A Multi-scale Approach to 3D Scattered Data Interpolation with Compactly Supported Basis Functions, Shape Modeling International, 153-161.
16. Ohtake Y., A. Belyaev, M. Alexa, G. Turk, H. P. Seidel (2010). Multi-level Partition of Unity Implicits, ACM Transactions on Graphics, 22(3), 463-470.
17. Ray H., H. Pfister, D. Silver, T. A. Cook (1999). Ray Casting Architectures for Volume Visualization, IEEE Transactions on Visualization and Computer Graphics, 5(3), 210-223.
18. Treece G. M., R. W. Prager, A. H. Gee (1999). Regularised Marching Tetrahedra: Improved Iso-surface Extraction, Computers & Graphics, 23(4), 583-598.
19. Witkin A. P., P. S. Heckbert (1994). Using Particles to Sample and Control Implicit Surfaces, SIGGRAPH'94 Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques, Orlando, USA, 269-277.
20. Zhao H. K., S. Osher, R. Fedkiw (2001). Fast Surface Reconstruction Using the Level Set Method, IEEE Workshop on Variational and Level Set Methods in Computer Vision, Vancouver, BC, Canada, 194-201.

Zhong Chen, Ph.D.
E-mail: chenzhong2006@126.com

Zhong Chen received his Ph.D. degree from Chongqing University in 2013. He is currently working at Huaiyin Institute of Technology, China. His research interests include reverse engineering and digital geometry processing.

Prof. Zhiwei Hou, Ph.D.
E-mail: zw_hou@163.com

Zhiwei Hou received his Ph.D. degree from Nanjing University of Aeronautics and Astronautics in 2014. Currently, he is a Professor in Huaiyin Institute of Technology, China. His research interests include reverse engineering and mechanical control.

Quanquan Yang, Ph.D.
E-mail: qqyang@hyit.edu.cn

Quanquan Yang received his Ph.D. degree from Nanjing University of Aeronautics and Astronautics in 2013. He is currently working at Huaiyin Institute of Technology. His current research interests include mechanics of functional materials and additive manufacturing technology.

Prof. Xiaobing Chen, Ph.D.
E-mail: chenxiaobing@hyit.edu.cn

Xiaobing Chen received his Ph.D. degree from Nanjing University of Aeronautics and Astronautics in 2012. Currently, he is a Professor in Huaiyin Institute of Technology, China. His research interests include reverse engineering and digital geometry processing.