Computer-Aided Design and Applications

http://www.cadanda.com

Heterogeneous Object Slicing with Geometric Contour Constraint

Xiao J. Wu[1], Michael Y. Wang[2,1] and Wei J. Liu[3]

[1]Harbin Institute of Technology Shenzhen Graduate School, wuxj@hitsz.edu.cn
[2]The Chinese University of Hong Kong, yuwang@mae.cuhk.edu.hk
[3]Shenyang Institute of Automation, CAS, wjliu@sia.cn

ABSTRACT

Slicing contour generation for geometric model is a vital process in rapid prototyping manufacturing. When slicing a heterogeneous object (HO), we take into account not only the geometric slicing but also the gradient material information. In this paper, we develop a new method to slice a heterogeneous object where the geometric contours are taken as constraints to resample the heterogeneous information through pixel overlap interpolation and tri-linear interpolation, which can maintain the original heterogeneous material information as much as possible and the slicing boundary as accurate as possible. The novelty of our method is that the gradient material resampling with geometry constraint can guarantee the crisp boundary of the HO slices when the gradient information is converted for geometric space to image space. Experimental results validate that our method can create heterogeneous object slices with more accurate boundary than simple interpolation scheme.

Keywords: heterogeneous object modeling, rapid prototyping, geometric slicing, heterogeneous object slicing.
DOI: 10.3722/cadaps.2009.137-145

1. INTRODUCTION

A heterogeneous object is referred to a solid component consisting of two or more material primitives distributed continuously or discontinuously within an object. Modeling and manufacturing of heterogeneous object (HO) have gained much attention recently as the advent of rapid prototyping manufacturing technologies. Rapid prototyping can offer possibility to fabricate the component with material variations because of the characteristic of layer by layer manufacturing. Techniques such as shape deposition manufacturing (SDM) and laser engineered net shaping (LENS) have been used to produce FGM parts with variety of metallic powders.

Approaches to modeling and representing a heterogeneous object have been extensively studied in the computer and manufacturing community. Kumar and Dutta proposed an approach to model multi-material objects based on R-m sets and R-m classes primarily for application in layered manufacturing. Boolean operators to facilitate the modeling procedure were defined [1, 2]. Jackson et al. proposed a local composition control (LCC) approach to represent heterogeneous object in which a mesh model was divided into tetrahedrons and different material compositions were evaluated on the nodes of the tetrahedrons by using Bernstein polynomials [3,4]. Chiu [5] developed material tree structure to store different compositions of an object. The material tree was then added to a data file to construct a modified format suitable for RP manufacturing. Marsan and Dutta presented a method to model material properties in the form of tensor product surfaces within the framework of heterogeneous solid modeling [6]. Siu and Tan developed a scheme named 'source-based' method to distribute material primitives, in which any material within an object could have varying properties [7]. The feature-based modeling scheme was extended to heterogeneous object representation. By controlling boundary conditions of a virtual diffusion problem in the solid, designers could control its material distribution [8, 9]. Kou and Tan suggested a hierarchical representation for heterogeneous object modeling by using B-rep to represent geometry and a heterogeneous feature tree to express the material distributions [10]. Samanta suggested a scheme to represent and design heterogeneous object by using a free-from functions to describe complex shapes of geometry and material features [11]. Zhang et al. address the problem of heterogeneous material object modeling and analysis through a constructive approach [12]. Wu et al developed a heterogeneous object modeling system based on volumetric date set [13]. Various methods for designing and optimizing objects composed of multiple regions with continuously varying material properties have been developed. Wang and Wang proposed a level-set based variational scheme [14]. Biswas et al. presented a mesh-free approach based on the generalized Taylor series expansion of a distance field to model a heterogeneous object to satisfy the prescribed material conditions on a finite collection of material features and global constraints [15].

To build an object by a particular rapid prototyping technology, certain paths must be generated to guide the executor to add the material layer by layer, called path planning [16]. It is necessary to slice the model into contours through a serial of parallel plane to intersect with the object. Conventionally, we do not consider material information when doing slicing operation. But the heterogeneous object can not be dealt with the same strategy. Siu et al. proposed a contour sub-division algorithm on each layer from slicing a heterogeneous object where the material gradient is decomposed into sub-contours according to the different grading variation [17][26]. However, the “grading step-width” based method separates the continuous material domain into discrete strips on each slice. In this paper, we develop a new method to slice a heterogeneous object where the geometric contours are taken as constraints to resample the heterogeneous information through pixel overlap interpolation and tri-linear interpolation. The novelty of our method is that it can maintain the original heterogeneous material information as much as possible and improve the slicing boundary as accurate as possible. This method is called material resample with geometric constraint (MRGC). The output of our scheme is bitmap liked raster image. Then lossless data compression can be utilized to store the heterogeneous object slices.

The subsequent sections of this paper are structured as follows: A geometric slicing algorithm is introduced in section 2. Section 3 describes in detail the heterogeneous object slicing with geometric constraint. Experiment and analysis are given in section 4. Section 5 concludes our paper.

2. GEOMETRY MODEL SLICING

2.1 Setup of the Weighted Directed Graph

One of the prerequisites of rapid prototyping manufacturing is slicing the geometry model into contours such that the path planning can be generated [18,19,20]. The geometric slicing algorithm was extensively studied in RP community in the past decades. It can be classified into two categories: slicing polygonal meshes with same thickness or adaptive layers; and direct slicing of CAD models. Jamieson developed the first direct slicing method on the basis of solid modeling core of UG [21]. Zhao proposed an adaptive direct slicing scheme for CAD models by using ARX [22]. In this paper, we use the Weighted Directed Graph (WDG) to recoding the STL model such that the face table, edge table and vertices table can be well established. WDG is a directed graph that has a weight, or numeric value, associated with each edge in order to eliminate the redundancy and facilitate the traversal of STL models. The benefits of WDG are as follows.

- Only one intersection point is needed for a triangle when calculating the intersection points as we can inherit another intersection point from the facet connectivity in WDG.
- All the intersection points are connected as a closed loop sequentially and the reordering is not needed.
- Once the WDG is constructed, it can be reused when the thickness of the slices does not meet the requirement, which is highly efficient.

Taking a tetrahedron as example, shown in Fig. 1(a), we set the triangle facet as the connection node of the graph, and then give numbers the corresponding edges, seeing Fig. 1(b), and set the weights for the common edges of the neighboring faces such that the WDG can be constructed according to this connectivity attributes, show in Fig. 1(c). The tetrahedron is unfolded as Fig. 1(b), then we denote the four triangles △BCD, △ABC, △ACD, and △ABD as *Tri*0, *Tri*1, *Tri*2 and *Tri*3, and the vertices of each triangles as v_0, v_1 and v_2. In Fig. 1(c), the nodes of the WDG are the four triangles denoted as N_0, N_1, N_2 and N_3, and the arrows point to the neighboring triangle facet. The number on each edge is the weight which is the summation of the subscripts. The weights must comply with the following regulations according to the subscript of the vertex notation.

$$1+0=1 \text{ for } v_0v_1 \text{ or } v_1v_0$$

$$2+0=2 \text{ for } v_0v_2 \text{ or } v_2v_0$$

$$1+2=3 \text{ for } v_1v_2 \text{ or } v_2v_1$$

For efficiently traversing the WDG, we construct an adjacency list to store the nodes, shown in Fig.2, and each node is a singly linked list as shown in Fig. 3. In this case, the redundancy of the STL can be get rid of and mitigate the slicing computation.

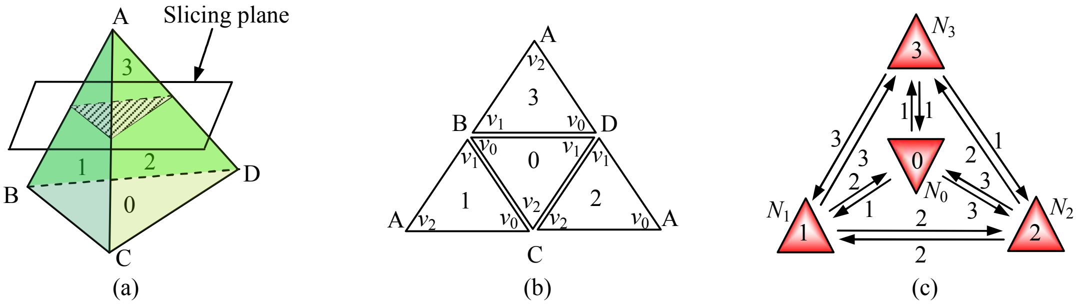


Fig. 1: A solid model: tetrahedron, (b) connectivity of triangles, (c) weighted direct graph of the triangles.

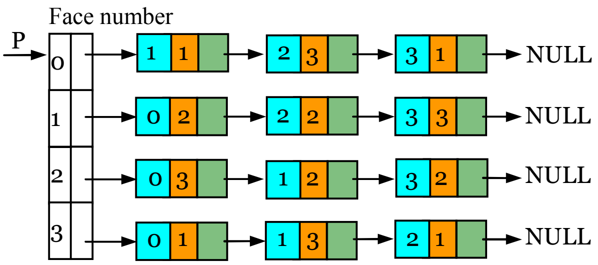


Fig. 2: Adjacency list for triangles.

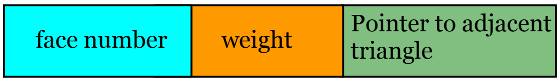


Fig. 3: Node of the adjacency list.

2.2 Recursively Search the WDG

After WDG is set up, we can traverse it to calculate the intersection point with depth first search (DFS). The traversal starts from node N_0, then to the next unvisited node N_i neighboring N_0. It is serial to process the next node of the graph. A flag is set to 1 for the traversed triangle as the slicing algorithm need not visit all of the triangles, process the triangles intersected with the slicing plane instead. As illustrated in Fig. 4, when the intersection point, A, locates on the common vertices of face f_1 and f_4, the flag of face f_2 has been set to 1 if we assume f_1 is the current triangle such that the face f_4 can not be processed. The next face should be f_3. The intersection points can be calculated according to the weights and stored in a doubly linked list. Therefore, it is important to test the intersection between triangle and slicing plane and set a flag to each triangle for efficiently slicing the model. If *flag*=0 or miss the intersection of the slicing plane with a triangle, the pointer of adjacency list moves to the next triangle to continue the search. When the traversal is accomplished, the slicing contour can be generated. Fig. 5 gives an example to validate the geometric slicing algorithm.

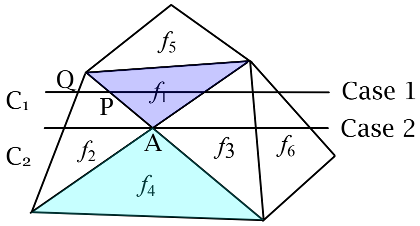


Fig. 4: Triangles search schemes.

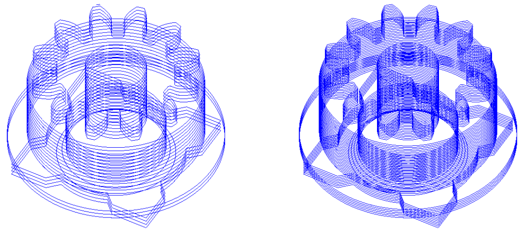


Fig.5: Geometric slices with different thickness.

3. HMP SLICING WITH GEOMETRIC CONSTRAINT

3.1 Material Slicing of a Heterogeneous Object

In our HO modeling system, the material information is represented in voxel based model. Because the voxel grid is axis aligned, it is very easy to obtain the material information along x, y, and z axis as illustrated in Fig. (b), (c), and (d) which is obtained by extracting a voxel layer every some continuous layers. Figure (a) is a gear mesh model, and (e) is one of the slices along z axis. On the one hand, this volumetric heterogeneous model can not be manufactured directly on account of the discrete representation and the roughness on the boundary. On the other hand, the volumetric model is very memory consuming, and unfeasible to compression, store, and transmission. Therefore, we must transform the volumetric object into portable format to meet the requirement of manufacturing and exchanging. In digital image processing, there are lots of lossless compression techniques to facilitate the data store and transmission.

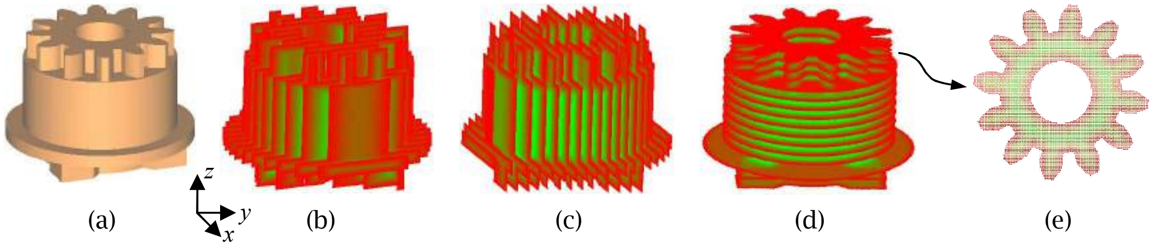
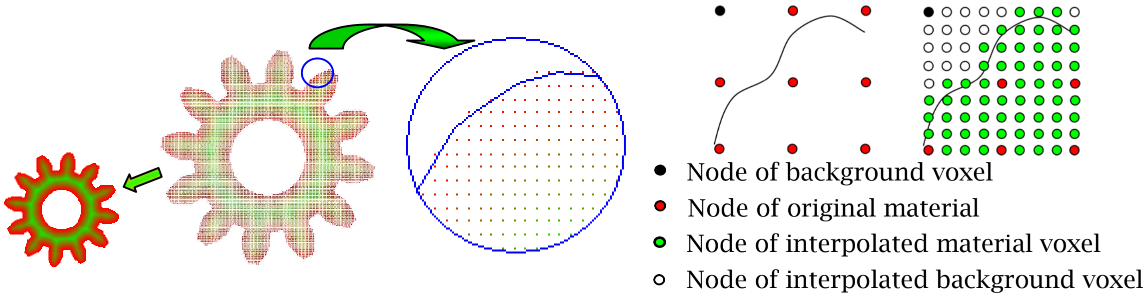


Fig. 6: Material slicing of heterogeneous object, (a) STL mesh, (b) heterogeneous object layers along *x* axis, (c) heterogeneous object layers along *y* axis, (d) heterogeneous object layers along *z* axis.

3.2 Material Layer Interpolation

As aforementioned, the discrete heterogeneous model can not offer adequate accuracy for fabrication because of the low resolution of voxel layers. For example, Fig. 7(a) is a bitmap liked image formed directly from a voxel layer (b) which is in object space. But the resolution is relatively low. Figure (c) is portion of zoomed voxel layer, from which we can find that the limited resolution can not completely represent the contour boundary. The blue curve is the geometry slice where some of the voxels locate outside the curve boundary. Fig. 7 (d) is representation of the geometric curve and pixels whose color of the dot represents the different attributes demonstrated as the notations in Fig. 7. Figure (e) is an enlarged image of the figure (d) from which we can conclude that the boundary of the solid model can be accurately represented with more interpolations. However, with the increase of the resolution, more and more object pixels become background pixels, the white dots in Fig. 7(d). We must offer a scheme to determine whether a pixel belongs to object or background. In this case, the geometry slicing contours can be used as constraints to restrict the interpolation. It is implemented by image interpolation method to resample the material slicing, which can be carried out in 2D and 3D. In 2D, it is regarded as interpolation inside an image, and in 3D one or more new images can be constructed with interpolation between neighboring two images.



(a) (b) (c) (d) (e)

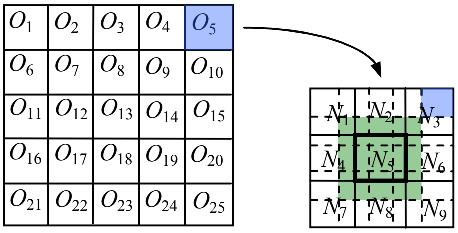
Fig. 7: Original voxel layer image and the interpolation.

Traditionally, by using nearest, bilinear, bicubic interpolation methods can improve the image resolution [23,24]. However, the pixels in new interpolated image can not maintain the information of the original image. Yokoya proposed an image interpolation method based on fractal geometry [25], which used the statistical self-similarity between gray levels of neighboring pixels to interpolate. Whereas it is difficult to accurately compute the self-similarity transformation using the traditional fractal scheme. Furthermore, the information can not be well maintained between the interpolated image and the original voxel layer after several times resampling. In this paper, we employ a method called pixel overlap interpolation (POI) to maintain the material information to the greatest extend, which can keep all of the original pixel values when up-sampling.

The pixels are regarded as rectangular instead of pure points in POI, as shown in Fig. 8(a). When executing the interpolation, the original image is extended like a plastic membrane to the same size with the interpolated image. Thus, the original image can cover all the area of the new image, shown in Fig. 8(b)., and Fig. 9(b). The new interpolated image can be obtained by calculating the information percentage of the overlapped rectangles. For example Fig. 8(a)., if we contract the image into 3×3, more pixels cover the overlapped rectangles; the number is 9 pixels at most, see Fig. 8(b). Original image includes 25 pixels, from O_1 to O_{25}. The pixel, N_1, in new image consists $36\%O_1$, $24\%O_2$ and O_6, and $16\%O_7$ respectively. That is

$$N_1 = 0.36O_1 + 0.24(O_2 + O_6) + 0.16O_7 .$$

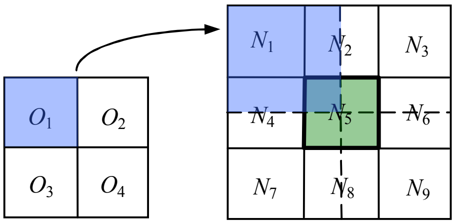
$$N_2 = 0.12(O_2 + O_4) + 0.36O_3 + 0.08(O_7 + O_9) + 0.24O_8$$



(a) 5×5

(b) 3×3 interpolated image, dot line indicates the original pixels.

Fig. 8: 5×5 image contracts into 3×3, the bold box is covered by nine pixels.



(a) 2×2

(b) 3×3 interpolated image, dot line indicates the original pixels.

Fig. 9: 2×2 image is scale into 3×3 , the bold rectangle is occupied by four pixels.

If the new image is enlarged, the maximum pixels in the new image are not more than four overlapped pixels, see pixel N_5 in Fig. 9(b). In this case, the new image can be constructed $N_1 = O_1$, $N_3 = O_2$, $N_7 = O_3$, $N_9 = O_4$, $N_2 = 0.5(O_1 + O_2)$ (so do N_4, N_6, N_8) and $N_5 = 0.25(O_1 + O_2 + O_3 + O_4)$. The advantage of this interpolation method is maintaining the information to the greatest extend which is very suitable to store the heterogeneous material compositions.

As our heterogeneous object is represented as volumetric data set, the material information in the inter-layer can be computed by trilinear interpolation which is a method of multivariate interpolation on a 3D regular gird. It approximates the value of an intermediate point within the local axial rectangular prism linearly. A sample point B is on the geometry slice plane Π and B is within a voxel, indicated in Fig. 10. The material information on B can be calculated using Eqn. (1).

$$B = P_0(1-u)(1-n)(1-w) + P_1u(1-v)(1-w) + P_2v(1-u)(1-w) + P_3w(1-u)(1-v) + P_4uv(1-w) + P_5uw(1-v) + P_6vw(1-u) + P_7uvw, \quad (1)$$

where u, v and w are the local coordinates on a voxel grid. They are computed using

$$u = x_b - x_{p0},\ v = y_b - y_{p0},\ w = z_b - z_{p0}, \quad (2)$$

where (x_b, y_b, z_b) and (x_{p0}, y_{p0}, z_{p0}) are the coordinates of sample point B and voxel vertex P_0.

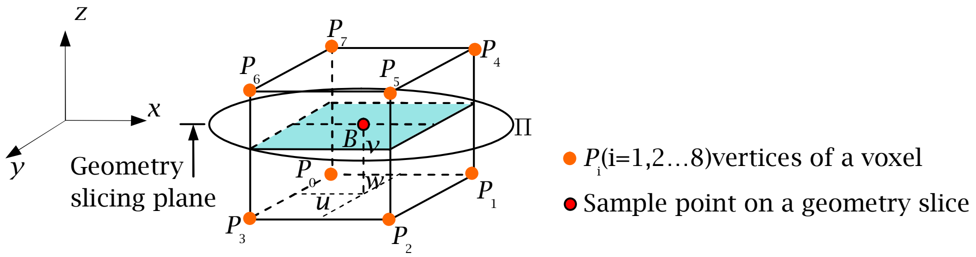


Fig. 10: Sample point and trilinear interpolation.

3.3 HO Slicing with Constraint of Geometry Slices

In the above section, we describe the interpolation methods in 2D inner-layer and 3D inter-layer. Now, we can start to construct HO slicing with any desired precision. Fig. 11(a)., is a STL model, (b) voxel layers with heterogeneous material and geometric slicing contours, (c) a new view of (b) along x axis, and (d) a zoomed view of a portion of (c), and (e) one of layers of (d) viewed along z axis. From figure (c)(d) and (e), we can conclude that our method implies two stages. Firstly, when the voxel layer resolution along the direction of geometry slicing is less than that of geometric slices, trilinear interpolation method is utilized to get the heterogeneous material information on the geometric planes such that the thickness of the voxel layer is same as the geometric slices. Secondly, when the resolution and shape accuracy of the voxel layer can not satisfy the requirement of manufacturing, the voxel layer should be resampled by using POI method on the geometry slice plane.

The most complicated part of HO slicing is the resampling operation on the voxel layers as we have to consider the boundary to grantee the accuracy and smoothness. The initial heterogeneous material layer is denoted as I_0, I_1 is the first sampled image, then next I_2,…,I_k. If the resolution can satisfy the requirement of manufacturing, the sampling procedure can be stopped, illustrated in Fig. 12. Figure (d) is the third resampling from which we can find that the pixels outside the geometric boundary have not any contribution to the shape representation. The eight pixels under the shadow is an example in Figure (d). These pixels can be changed into background pixels. In practice, a threshold *d* is set to filter the non-contributive pixels. If the distance from the pixel to the geometric contour is greater than d, this pixel is set to background pixel; otherwise, this pixel is maintained as object pixel. In this scheme, the accuracy of the final HO slice is totally decided by the number of resampling, demonstrated in Fig. 12(e).

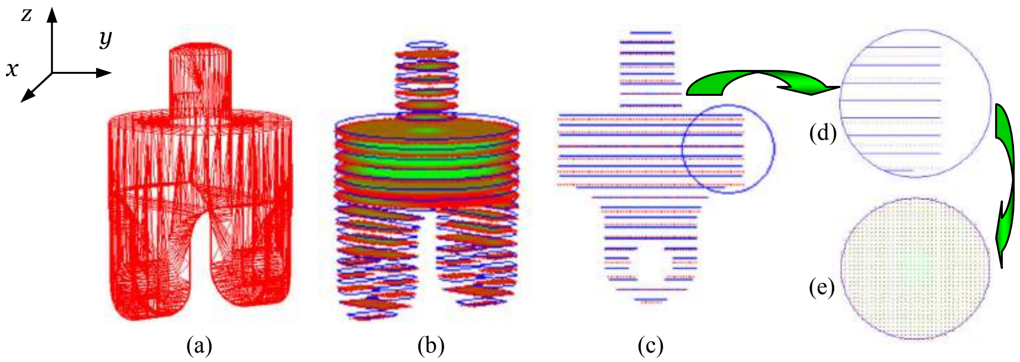


Fig. 11: HMP slicing and geometric slicing, (a) geometric model, (b) geometric slices and voxel layers, (c)(d)(e) different view of the geometric slices and voxel layers.

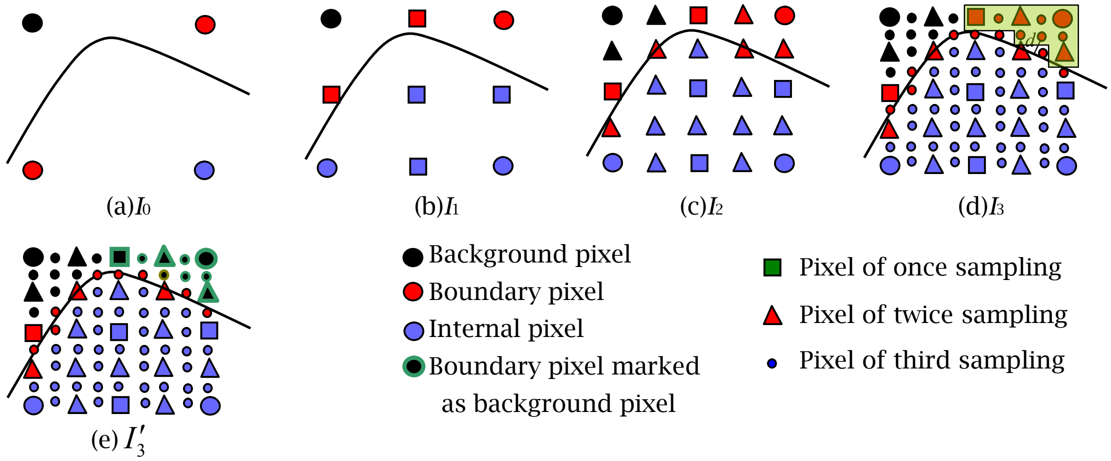


Fig. 12: Iteratively sampling in 2D, (a) is the original image, (b)(c)(d) are three times resampling (d) is the final high resolution image after eliminating the non-contributive pixels.

4. EXPERIMENTS AND ANALYSIS

The examples of using the proposed scheme are implemented with VC++6.0 and OpenGL. Fig.13 is an example of heterogeneous object slicing and resampling with geometric contour constraint, where (a) is STL mesh model, (b) the corresponding voxel model, (c) heterogeneous object and geometric slices, thickness of slice is 0.15mm and the number of layers is 117. The amount of voxel layer along z axis is 73. (d) is a hatched view of the heterogeneous object. (e) is a layer of geometric slice contour and a voxel layer with same height. Figure (f) is an image constructed from the voxel laryer directly but with low resolution, only 80×124. Then the image is resampled four times using aforementioned technique, the resolution can achieve 320×496. After the invalid pixels are abandoned, we obtain an image with high resolution and clear boundary exactly with the corresponding geometric contour, for clearness only three portions of the image are displayed in figures (g), (h) and (m). Figure (n) is an enlarged part of image (f) using simple interpolation scheme without geometry constraint, from which we can find that the edge of the image is very blurry. We can also use the geometric contours produced by direct slicing or adaptive slicing algorithm as constraints to reconstruct the HO slices. In this case, the accuracy of HO slices is determined completely by the resample resolution. It is clear that we can theoretically construct accurate slices with heterogeneous information exactly as long as the resample resolution is high enough. However, it will increase the computational and storage cost. It is unnecessary to resample the material voxel layer to extremely high resolution. As long as the accuracy of the layers satisfies the manufacturing requirement, it should be stopped. The slices can be employed to produce the path planning using halftone or other methods.

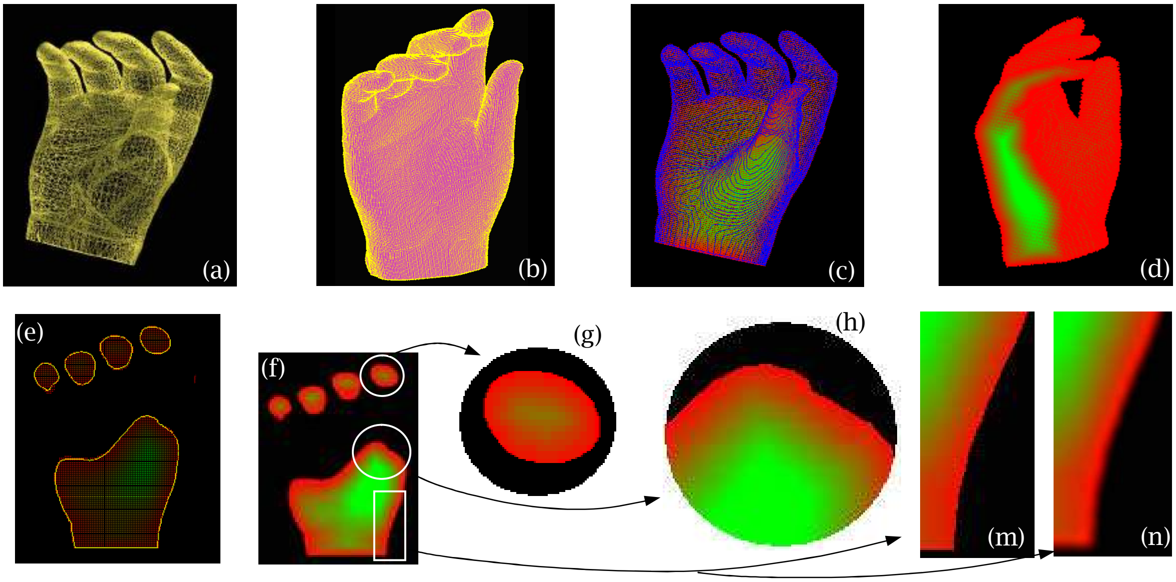


Fig.13: An example of MRGC, (e) a layer of volume dataset and a geometrical contour, (f) a HO slice constructed from (e) directly, (g)-(m) are three parts of the enlarged image with clear boundaries, (n) is an enlarged part of image (f) using simple interpolation scheme without geometry constraint.

5. CONCLUSIONS

We have presented a framework to generate slices for heterogeneous object. This framework combines the geometric slice contours and gradient material information for heterogeneous object slicing. In our new method, the geometric contours are taken as constraints to resample the heterogeneous information through pixel overlap interpolation and trilinear interpolation to avoid blur on the boundary owing to simple interpolation scheme. The novelty of our method is that it can maintain the original heterogeneous material information as much as possible and the slicing boundary as accurate as possible. The output of our method is images with lossless compression property, which is most suitable for 3D printing technology to manufacture the heterogeneous object through digital half-toning method.

6. ACKNOWLEGEMENT

This project is partially supported by Natural Science Foundation of China (NSFC No.50805031)

7. REFERENCES

[1] Kumar, V.; Dutta, D.: An approach to modeling multi-material objects, Proceedings of the fourthACM symposium on Solid modeling and applications, 1997, 336-345.
[2] Kumar, V.; Dutta, D: An approach to modeling and representation of heterogeneous objects, Journal of Mechanical Design, 120, 1998, 659-667.
[3] Jackson, T. R.: Analysis of functionally graded material object representation methods, PhD thesis, Massachusetts Institute of Technology (MIT), Cambridge, MA, January 2000.
[4] Liu, H.: Algorithms for design and interrogation of functionally graded material solids, Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, January 2000.
[5] Chiu, W. K.; Tan S. T.: Multiple material objects: from CAD representation to data format for rapid prototyping, Computer-Aided Design, 32, 2000, 707-717.
[6] Marsan, A.; Dutta, D.: On the application of tensor product solids in heterogeneous solid modeling. Proceedings of 1998 ASME Design Engineering Conferences, Atlanta, Georgia, Sep. 1998, 1-9.

[7] Siu, Y. K.; Tan, S. T.: 'Source-based' heterogeneous solid modeling, Computer-Aided Design, 34(1), 2002, 41-55
[8] Qian, X.; Dutta, D.: Physics-Based Modeling for Heterogeneous Object, Journal of Mechanical Design, 125(3), 2003, 416-427.
[9] Qian, X.; Dutta, D.: Feature-based design for heterogeneous objects, Computer-Aided Design, 36(12), 2004, 1263-1278.
[10] Kou, X.Y.; Tan, S. T.: A hierarchical representation for heterogeneous object modeling, Computer-Aided Design, 37(3), 2005, 307-319.
[11] Samanta, K.; Koc, B.: Heterogeneous object design with material feature blending, Computer-Aided Design and Applications, 1(1-4), 2004, 429-437.
[12] Zhang, X. F.; Subbarayan, G.: A Constructive Approach for Heterogeneous Material Modeling and Analysis, Computer-Aided Design and Applications, 1(1-4), 2004, 171 -178.
[13] Wu, X.,J.; Liu, W. J.; Wang, M. Y.: Modeling Heterogeneous Objects, CAD Computer-Aided Design and Applications, 4(6), 2007, 731-740.
[14] Wang, M. Y.; Wang, X. M.: A level-set based variational method for design and optimization of heterogeneous objects, Computer-Aided Design, 37(3), 2004, 321-337.
[15] Biswas, A.; Shapiro, V.; Tsukanov, I: Heterogeneous material modeling with distance fields, Computer Aided Geometric Design, 21(3), 2004, 215-242.
[16] Marsan, A. L.; Allen, S.; Kulkarni, P.; Dutta, D.: An integrated software system for process planning for layered manufacturing, Proceedings of the Solid Freeform Fabrication Symposium, 1997, 661-668.
[17] Siu, Y. K.; Tan, S. T.: Modeling the material grading and structures of heterogeneous objects for layered manufacturing, Computer-Aided Design, 34(10), 2002, 705-716.
[18] Adams, S.-H.; Yang, M.-Y.: A study on a generalized parametric interpolator with real-time jerk-limited acceleration, Computer-Aided Design, 36(1), 2004, 27-36.
[19] Mani, K.; Kulkarni, P.; Dutta, D.: Region-based adaptive slicing, Computer Aided Design, 31(5), 1999, 317-333.
[20] Tata, K.; Fadel, G.; Bagchi, A.; Aziz, N.: Efficient slicing for layered manufacturing, Rapid Prototyping Journal, 4(4), 1998, 151-167.
[21] Jamieson, R.; Hacker, H.: Direct slicing of CAD models for rapid prototyping, Rapid Prototyping Journal, 3(1), 1995, 12-19.
[22] Zhao, Z.; Laperriere, L.: Adaptive direct slicing of the solid model for rapid prototyping. International Journal of Production Research, 38(3), 2000, 89-98.
[23] Gmoldwasser, S. M.; Reymolds, R. A.; Talton, D. A.; et al.: Techniques for the rapid display and manipulation of 3-D biomedical data, Comut. Med. Imag. Graph., 12(1), 1988, 1-24
[24] Maeland, E.: On the comparison of interpolation methods, IEEE Transactions on Medical Imaging, 7(3), 1988, 213-217.
[25] Yokoya, N.; Yamamoto, K: Fractal-based analysis and interpolation of 3D natural surface shapes and their application to terrain modeling, Computer Vision, Graphics and Image Processing, 46(3), 1989, 284-302.
[26] Siu, Y. K.; Tan, S. T.: Slicing and contours generation for fabricating heterogeneous objects, Proceedings of the Geometric Modeling and Processing—Theory and Applications (GMP'02), 2002, 219-225.