

Comparison of Surface Reconstruction Method for CT based on VTK

Tingting Xu, Ning Wei*, Tingyao Jiang

College of Computer and Information Technology, University of Three Gorges, YiChang 443002, China
ctgu_xtt@163.com, weininglz@163.com

Abstract - The technology of 3D reconstruction of medical images is an important application in computer graphics and image processing, especially in biomedical engineering. In this paper, two types of surface construction algorithms for medical sequences images are reviewed in this paper, including Marching Cubes (MC) and Contour reconstruction. We describe the algorithms implementation framework by using Visualization Toolkit (VTK). After testing these algorithms on actual CT (Computed Tomography) data, we analyze their reconstructed results and make a comparison on each of their advantage and disadvantage for scope of application.

Index Terms - 3D reconstruction, Surface reconstruction, VTK

1. Introduction

With the emergence and development of medical imaging technologies, such as computer tomography, nuclear magnetic resonance and ultrasound, people can get the human body and its internal organs of the 2D digital CT (Computed Tomography) image sequence. The clinical applications of these technologies made great development in the medical diagnosis and treatment. Although the 2D CT images can express the anatomical information of a section, it is difficult to establish a 3D structure. Therefore, it can't provide relative position of 3D space of the organs and makes it hard for doctors to locate and diagnose diseases by these images. In order to provide true and reliable diagnosis and make rational treatment planning, the medical image 3D reconstruction technology arises, and become the hot point of research.

The 3D reconstruction process refers to the generation of a 3D image from a digitized set of 2D images. The most widely used method for rendering such 3D geometrical model is iso-surface rendering which uses a surface model (always planar patch especially triangular patch) to approximate the surface. At present, the commonly used iso-surface construction algorithms are Marching Cubes [5], Contour based reconstruction [6-7], Marching Tetrahedron, Dividing Cubes and Delaunay Triangulation. Due to the fast speed in reconstruction, iso-surface rendering has been widely applied to the fields such as surgical navigation, virtual endoscopy and radiotherapy plan, etc [1-4]. In this paper we describe the detail of the Contour reconstruction and Marching Cubes algorithm.

To apply these algorithms in research, the paper uses VTK (Visualization Toolkit) to achieve the visualization of 3D complicate medical rendering system. The paper summarized 3D reconstruction based on VTK and its framework.

At last, we compared the rendering results of using the MC and CR, and analysis their individual merit and weaknesses. These conclusions will helpful to our further research in the field of medical image segmentation and registration.

The paper is structured as follows, in section II we describe the algorithms of Contour based reconstruction and Marching cubes reconstruction; In III we introduce the visualization toolkit VTK and its 3D reconstruction framework; We show the results of the two methods in the section IV; At last, we draw conclusion at the V.

2. Reconstruction Algorithm based on Surface Rendering

A. Contour reconstruction

Contour reconstruction algorithm [8] was put forward earlier than other reconstruction algorithms. The main concept of Contour reconstruction algorithm is to extract contours from 2D slices, and then to connect the adjacent sampling vertices on contours to form triangles. The connecting rule is based on the geometrical corresponding relationship between contours. After that, we can get the triangle mesh surface of an object.

The first step of the algorithm is to read data slices into memory. In order to avoid pixilated, it is need to interpolate the pixels to achieve structured slices. The next step is to extract contours of each slice and then build the classification tree for topological reconstruction. The aim of topological reconstruction is to determine the geometrical corresponding relationships between contours. At last, we can construct triangular patches of the surface by geometrical reconstruction based on the tree and geometrical corresponding.

The Flowchart of contour reconstruction algorithm is showed in Fig. 1.

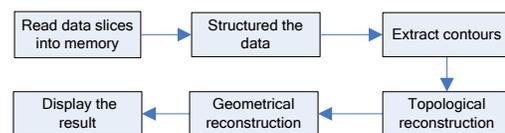


Fig. 1 Flowchart of Contour reconstruction algorithm

1) Topological reconstruction

The purpose of topology reconstruction is to determinate the geometrical corresponding relationships between the contours. We first build the classification tree. After that we

* The corresponding author is Ning Wei.

determine the corresponding relation of contours based on the connecting type of the nodes in the tree.

(a) Build the classification tree

Classification tree is used to determine the topological relations between contours. The number of tree layers equals the number of slices. Every node on each layer corresponds to a contour in each slice. The edge in the tree indicates that there exists geometrical relation between the two contours.

Fig. 2 shows an example to construct a classification tree from slices. The number of slice layers in Fig. 2 (a) is 4. The contour of the first layer is expressed with a symbol a , the second layer with b, c and d , the third are e, f and g , the fourth is h . So the layer number of corresponding classification tree is 4. To determinate the edge between nodes, we project the adjacent slices into a plane. If there is an intersection between adjacent contours, that means they have corresponding relationship and there exists an edge between the nodes in the tree, as shown in figure (b). This process does not consider nested contours. For example, the contour c of the second slice layer (with color red) nested in b , but we do not consider the corresponding relation between a and c . The contour a and b , as well as a and d have intersections, so there is an edge between a and b , also a and d . The other edges in Fig. 2 (c) can be got analogously.

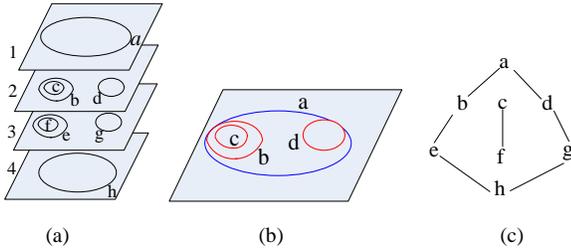


Fig. 2 Constructed diagram of the classification tree. (a) is the original slices, (b) is the projected contours and (c) is the classification tree

(b) Determinate the geometric corresponding relations

After get the classification tree, the next step is to determinate the specific relationship represented by the edges. Contour algorithm only considers four basic correspondences which are pole, cylinder, independent branch and connected branch respectively, as shown in Fig. 3.

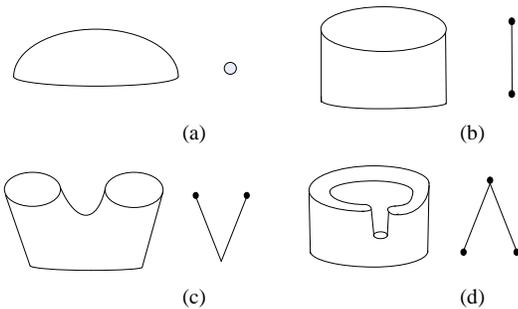


Fig. 3 Four kinds of basic geometric correspondence. (a) pole; (b) cylinder; (c) independent branch; (d) connected branch

For the classification tree in Fig. 2 (c), there is an edge between the node a and b , as well as a and d . So there are corresponding relationships of independent branch correspondence shown in Fig. 3 (c). It means that the sample point on contour corresponds to two nearest points at contours b and d each, as shown in Fig. 4. We assume that there are two contour collections in two adjacent layers, and denoted them as $S = \{S_0, S_1, \dots, S_m\}$ and $P = \{P_0, P_1, \dots, P_n\}$, as shown in the Fig. 4. The bigger contour coloured with blue and smaller two circles are coloured with red respectively. For S_0 in point of S , its two nearest points P_0 and P_1 are selected from P by using the rule of geometric corresponding and classification tree. Then $S_0P_0P_1$ forms a triangle of the surface. Take the same rule, the triangle can be formed for each point in the set S .

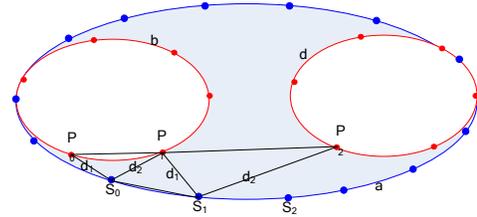


Fig. 4 Corresponding relationship of contour point

2) Geometrical reconstruction

The task of geometry reconstruction is to build the triangular geometric surface on the basis of topology reconstruction. That means connect the contour point and its corresponding points to build triangles mesh surface. The connected result of adjacent layers is showed in Fig. 5.

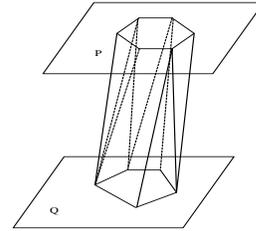


Fig. 5 Triangular geometric surface

B. Marching cubes reconstruction

MC (Marching cube) uses the spatial decomposition method to cut the spatial using voxel elements (cubes), and then approximate the object using the voxel elements and cutted voxel elements. MC is widely used because of its simple implementation.

The main concept of MC [5] is to regard the 2D medical image sequences as independent layers. The vertices of the cube are regularly distributing on the layers. We select a cube using the eight vertices on two adjacent layers, the cube is defined as voxel elements. Then the slices are divided into many small voxels. The next step is to compare the pixel value of the eight vertices with a given threshold value, and then label its states with 0 if the pixel value less than threshold or 1

conversely. For an edge if the states of two vertices are labeled with different states, it indicates the edge has an intersection with the iso-surface, as shown in Fig. 6. The position of the intersections can be calculated by interpolating using the two vertices. Finally we can get the surface by connecting all the intersections into triangles. In a 3D space, there are 256 different situation are enumerated for MC representation. All these situations can be generalized in 15 families by rotations and symmetries [11].

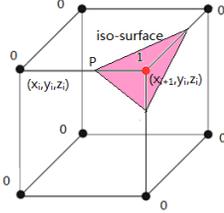


Fig. 6 Intersections of cube and iso-surface

The implementation process of the Marching cubes algorithm is showed in Fig. 7.

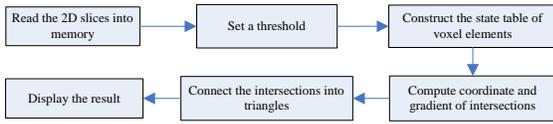


Fig. 7 Flowchart of Marching cubes algorithm

1) Calculate the coordinates of intersections

The vertex where the edge intersects with the iso-surface can be calculated using the interpolation method as follow.

Assuming that the threshold value of iso-surface is C and the coordinates of the two vertices at one edge of voxel represent as p_1 and p_2 , and the pixel value are denoted by v_1 and v_2 . Using linear interpolation, the coordinate of intersection vertex p_0 can be computed as (1):

$$p_0 = p_1 + \frac{(C - v_1)(p_2 - p_1)}{v_2 - v_1} \quad (1)$$

3. VTK Visualization Mechanism

A. VTK visualization framework

VTK (Visualization Toolkit) is a graphic library designed for 2D or 3D graphical visualization. It has been widely used in 3D computer graphics, image processing and visualization.

VTK encapsulates some surface rendering algorithms, such as the Marching cubes and Contour reconstruction, are commonly used in medical data visualization. Moreover, the toolkit provides reliable interface of input and output for the secondary development.

The 3D visualization framework [8-9] of VTK mainly includes two parts: the visual model and the graphic model. As shown in Fig. 8.

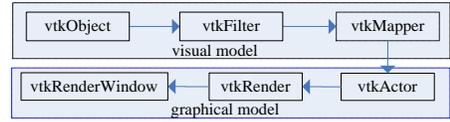


Fig. 8 Visualization framework of VTK

Visual model mainly processes data and generates geometries used for rendering. But the graphical model is used to draw the geometry. The mainly objects contained in the VTK's visualization framework include: source object, filter object, mapping object, render object, render window object and so on. Among these objects, mapping object is the interface of visual model and graphical model.

In Fig. 8, the `vtkObject` is the base class of VTK, It provides basic method for the visualization process; `vtkSource` is the derived class of `vtkObject` and the base class of `vtkFilter`, and it defines the specific behaviors and interfaces to the whole visual process definition. Besides this, `vtkSource` is mainly used to read the input data and generate object. `vtkFilter` is also an important class in VTK. After dealing with `vtkFilter`, the source data will be converted to a form which can be processed by VTK directly. `vtkMapper` is also a derived class of `vtkObject`. It maps the data into geometric data through `vtkFilter` and defines the interface of source data and graphic data. `vtkActor` is the 'actor' in VTK. It used to draw graphics and set the geometric data attributes by using the method of `SetMapper`. The demonstrator in VTK is `vtkRender`. It uses `AddActor` to add the actor to the stage and show it through `vtkRenderWindow` class. In addition, another important object `vtkRenderWindowInteractor` is used for interacting the target graphics, and the `SetRenderWindow` method can set the window to be interactive.

B. 3D reconstruction based on VTK

On the basis of data pipeline in VTK, 3D reconstruction algorithms can be used to process and model the slice images after reading the sequence images. Then combined with the rendering pipeline, the processed data can be mapped into geometrical data with topological structure. And after computing in the renderer, the rendering result will be displayed in the render window. The process of reconstruction in VTK is shown in Fig. 9.

The 3D reconstruction algorithm in Fig. 9 is implemented by using the built-in surface rendering algorithms in VTK. The MC is implemented by the class `vtkMarchingCubes` and the contour algorithm is by `vtkContourFilter`. The two algorithms are the subclass of `vtkAlgorithm` which is the derived class of `vtkObject`. According to the visualization framework, as shown in Fig. 8, we can learn that the original slice data processed by reconstruction algorithms can enter to the `vtkFilter` class to be processed.

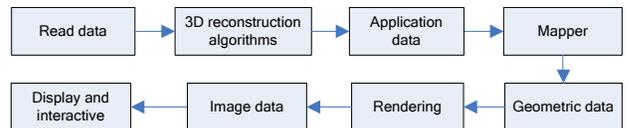


Fig. 9 Flowchart of Marching cubes algorithm

4. Experiment Results

In this paper, the data used in tests come from the database of National Institutes of Health [10]. The detailed information of the testing data is showed in Table I.

TABLE I Testing data

Models	Slice Resolution	Pixel spacing (mm)	Number of images	Data size (MB)
ankle	512×512	1.0	150	75.1
head	512×512	1.0	234	117
knee	512×512	1.0	350	175

The head, knee and ankle models are rendered respectively with the above two kinds of surface rendering reconstruction algorithms. The results are shown in Fig. 10.

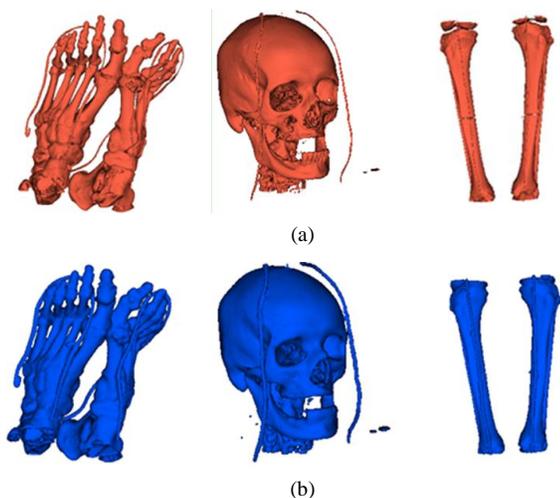


Fig. 10 Iso-surface rendering results. (a) MC algorithm and (b) Contour algorithm. The models from left to right are ankle, head and knee.

The rendering time is showed in Table II.

TABLE II Rendering time

Models \ Time	MC reconstruction (s)	Contour reconstruction (s)
ankle	0.468	12.70
head	0.546	15.245
knee	0.577	18.53

By comparing the rendering results and rendering time of the two algorithms, we concluded that the reconstruction efficiency of MC is relatively higher than the Contour reconstruction. Since the rendering time of MC is shorter, so it is suitable for the reconstruction of larger data. But MC algorithm has its own disadvantages, such as how to set the threshold during the construction. Always, a reasonable threshold is selected empirically. In addition, the

reconstruction effect of it is poorer than Contour reconstruction algorithm. As in Fig. 10, some parts appeared cracks, such as the ear and eye in the head model, as well as the middle part of the knee model. While the Contour reconstruction algorithm extracts the contour information of slices to get external characteristics of an object firstly. The reconstruction result of Contour algorithm has no cracks and it is more accurate than MC. So it can be proved that the Contour reconstruction algorithm is suitable for applications where the high accuracy is needed or the perfect object segmentation slice results are provided.

5. Conclusions

This paper mainly studies the marching cubes and Contour algorithm among the 3D reconstruction surface rendering algorithms. Especially describes the principle and process of the two algorithms in detail. After that, we simply introduce the software package VTK and summarized the procedure of 3D reconstruction using it. With the built-in marching cubes and contour algorithm, we implement them. At last, we compare and analysis the result of the two algorithms and get the advantages and disadvantages of each other. All of these provide the basis for medical image 3D reconstruction.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (Grant No. 6120214 and Grant NO. 41172298).

References

- [1] Huang, Yuxuan, Zhongpan Qiu, and Zhijun Song, "3D reconstruction and visualization from 2D CT images," IT in Medicine and Education (ITME), 2011 International Symposium on, vol. 2, 2011.
- [2] Hafizah, Mahani, Tan Kok, and Eko Supriyanto, "Development of 3D image reconstruction based on untracked 2D fetal phantom ultrasound images using VTK," WSEAS transactions on signal processing, vol. 6, no. 4, 2010, pp. 145-154.
- [3] Preim, Bernhard, and Dirk Bartz, "Visualization in medicine," Theory, Algorithms, and Applications. Amserdam: Morgan Kaufmann, 2007.
- [4] Wee, Lai Khin, Hum Yan Chai, and Eko Supriyanto, "Surface rendering of three dimensional ultrasound images using VTK," Journal of Scientific & Industrial Research, vol. 70, no. 6, 2011, pp. 421-426.
- [5] Lorensen, William E., and Harvey E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," ACM Siggraph Computer Graphics, vol. 21, no. 4, 1987.
- [6] Keppel, Eric, "Approximating complex surfaces by triangulation of contour lines," IBM Journal of Research and Development, vol. 19, no. 1, 1975, pp. 2-11.
- [7] Fuchs, Henry, Zvi M. Kedem, and Samuel P. Usetlon, "Optimal surface reconstruction from planar contours," Communications of the ACM, vol. 20, no. 10, 1977, pp. 693-702.
- [8] William J. Schroeder, VTK user's guide, Kitware inc, 1998.
- [9] Hansen, Charles D., and Christopher R. Johnson, eds, The visualization handbook. Access Online via Elsevier, 2005.
- [10] <http://www.nlm.nih.gov/research/visible/>.
- [11] Hafizah, Mahani, Tan Kok, and Eko Supriyanto, "3D ultrasound image reconstruction based on VTK," Proceedings of the 9th WSEAS International Conference on SIGNAL PROCESSING, 2010.