

Visualizing Volume in CT Images Using Vascular Modelling Toolkit

Yusman Azimi Yusoff^{a,b}, Bong Tze Wee^a, Mohd Khalid Mokhtar^{a,b}, Farhan Mohamed^{a,b,*}

^a Faculty of Computing, Universiti Teknologi Malaysia, Skudai 81310, Johor, Malaysia

^b Media and Game Innovation Centre of Excellence, Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, Malaysia *

Corresponding author: farhan@utm.my

ABSTRACT

Current medical images processing software moving toward with a complete package from medical images data segmentation to patient specific blood flow and analysis. Each software package has its own user interface design while containing almost same medical imaging processing pipeline. Volume visualization is one of the processes in the pipeline that requires computationally intensive method such as isosurfaces and marching cube. We suggested using Vascular Modelling Toolkit application to perform 3D volume rendering technique for Computed Tomography images in order to extract vascular structure in 3D. Seamless and user-friendly of the tool allow any beginner with basic knowledge able to use the toolkit and understand the process behind of visualizing 3D vascular structure.

INTRODUCTION

Computed Tomography (CT) is a non-invasive medical screening method. The captured cross-sectional images are generated using the customs x-ray hardware. CT image can be classified into several types such as coronary CT, calcium-score screening heart scan, single photon emission CT, x-ray CT and total body heart scan. X-ray CT used to capture the image of the human heart has increased exponentially over the last decade (Nicol et al. 2011).

In order to view the images, the viewer needs to navigate slice by slice. The problem becomes worse because the user needs to do mental construction to imaging the structure of the human organ which quite troublesome for the newbie user. One of the solutions is to extract the important features and show it in a 3D environment. Surface visualization able to solve this problem by processing the image and come with a 3D visual representation of the human organ extracted from the CT images. This process can assist the user to accomplished tedious task in short period of time.

This chapter will cover knowledge related to surface visualization technique and medical visualization application. A review of the Vascular Modelling Toolkit (VMTK) and its implementation also will be covered in this chapter. Last but not least, a discussion is prepared to provide additional information about the reviewed toolkit and its features as well as the available method and application that can achieve a similar result.

VOLUME VISUALIZATION TECHNIQUE

Volume visualization falls under scientific visualization area of knowledge. It is the process of converting scientific data set into 2D or 3D informative visualization (Kolivand and Sunar, 2014). Some of the areas that apply scientific visualization knowledge are meteorology, oceanography, aeronautic, automotive and much more. Volume visualization usually implemented on to the data sets that have scalar value. Examples of data set that contain scalar values from medical imaging hardware are CT, computed confocal microscopy over ultrasound and magnetic resonance imaging (MRI) scans. The data of material testing or control field comes from industry-CT.

Other example data set that contain scalar value are generated from a simulation using finite element method and computational fluid dynamics knowledge.

Tomographic volume data is used in volume visualization such as the data found in MRI or CT is an important benefit for research, education, treatment planning, diagnosis and surgery rehearsal (Pommert and Höhne, 2002). The fundamental idea of volume visualization is to make the boundaries of three-dimensional object become transparent and so the user can gain insight into three-dimensional data. The purpose of volume visualization in medicine is to create detailed and realistic views of objects from medical volume data (Stergiopoulos 2009).

Isosurface

Isosurface is technique transform volumetric data to the surface model. The vital step for indirect volume rendering algorithm is the extraction of isosurfaces from volume datasets (Bischoff & Kobbelt 2002). An isosurface displays the three-dimensional volume bounded by a particular isovalue. Isosurface computes and draws a surface within the volumetric data on three-dimensional surface which corresponding to points with a scalar value. In short, isosurface is a closed surface which separates the outside and inside. The isosurface has two levels which the volume inside holds a value greater than isovalue and volume outside holds value less than the isovalue. The surface defined by the isovalue is called isosurface. Figure 1 shows the demonstration of iso-boundary on a cross-section of a smooth sphere.

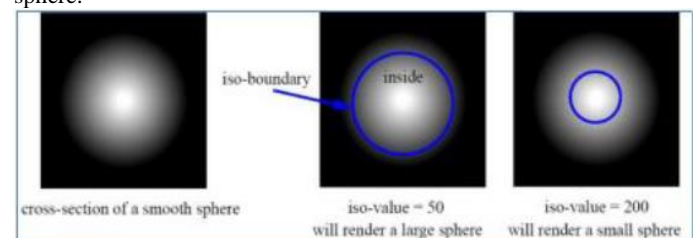


Fig. 1 Demonstration of iso-boundary on a cross-section of a smooth sphere.

The surface represents the constant value of a point such as velocity, pressure, density, and temperature within the volume.

Isosurface usually displayed by computer graphics and is used to visualize computational fluid dynamics. This technique allows people to study the fluid flow features such as liquid and gas around the objects.

Isosurface is among the most prominent techniques for volume visualization and is widely used in medicine, engineering, computational physics and others (Klein et al. 2004). Isosurface can use to represent areas of specific density in three-dimensional CT scan in medical imaging. It can visualize the organs, bones or other structures. But, the limitation of isosurface is only geometry features of the isosurface are visualized. The big limitations of implements isosurface in applications are this technique unable to illustrate the material property and internal structures behind an isosurface. It may be suitable for particular datasets such as CT scan datasets but not always appropriate.

Marching cubes algorithm is one of the general methods to construct the isosurface from volumetric datasets. Besides that, marching tetrahedrons and asymptotic decider are methods of construct isosurface.

Marching Cube

Marching cube algorithm is developed by Lorensen and Cline in 1987 (Raman & Wenger 2008). Marching cubes algorithm is an indirect rendering technique which use the triangular faces as basic elements to construct the isosurface from nonlinear and irregular data, particularly suitable for medical imaging data sets (Gong & Zhao 2010). Marching cubes are used to extracting the polygonal mesh of the isosurface from three-dimensional scalar volume data. In other words, marching cubes algorithm estimate the isosurface with triangle mesh which generated by comparing the isovalue and voxel densities. The algorithm exam each cell in the volume one by one and create triangles inside the cell by configuring the voxel values. The applications with this algorithm mainly focus on medical image visualization such as CT scan and MRI scan images.

The idea of marching cubes is to create the triangular mesh that will approximate the isosurface. Besides that, it can estimate the normal to the surface at each triangle. First, the algorithm assumes all voxels above the isovalue are one and others are zero. The voxel that inside the surface and the appropriate bit is set to one if the scalar value is higher than the isovalue while the voxel located outside the surface is set to zero if the scalar value is lower than the isovalue. Then, the next step is to find the triangular surface includes all 1s and excludes all 0s. Since each cube has 8 vertexes and the vertex can be either 0 or 1, so totally have 28 cases which are 256 cases. By exploited symmetry and rotation case, the 256 cases reduced to 15 cases. Figure 2 shows the 15 cube combinations. Linear interpolation is the approach to find the surface intersection.

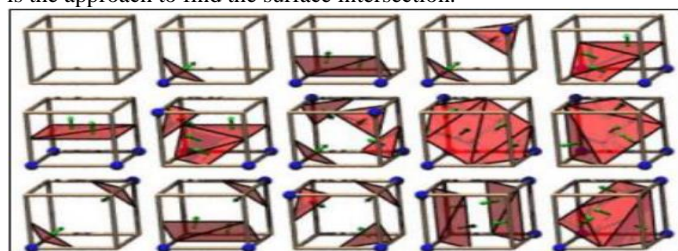


Fig. 2 The 15 cube combinations (Lorensen & Cline 1987).

Marching cubes algorithm described as a test to define whether the arbitrary point is within the object. The space within the bounds divides into cubes. Then, test the vertex of every cube whether it is inside of the object. Vertexes of every cube are either inside or outside of the object, the surface of the object must pass through the cube and intersecting the edges of the cube. After that, a surface draws between each cube and connect the intersections. Researchers can understand well about marching cubes algorithm works by looking at the two-dimensional case first which is called marching squares algorithm.

The two-dimensional image is shown in Figure 3(a). First of all, the image divides into squares (Figure 3(b)). The vertexes which are

located inside the image labeled as red dot while the vertexes which are located outside the and computational labeled as a blue dot (Figure 3(c)). But, the edges between the inside and outside vertexes have intersected with the image. The midpoints between the inside and outside vertexes have been labeled as a purple dot. The approximation of the original image can obtain through the connection of the purple dots (Figure 3(d)).

Marching cubes work much the same as marching square. The input volume datasets consist of samples organized into a regular three-dimensional Cartesian grid (Patera and Skala, 2004). Suitable size must choose for cubes which align with the slices in order to test the vertices of each cube. The surface intersects the edges in between the vertexes of the cubes opposing classification. The cube looks like Figure 4.

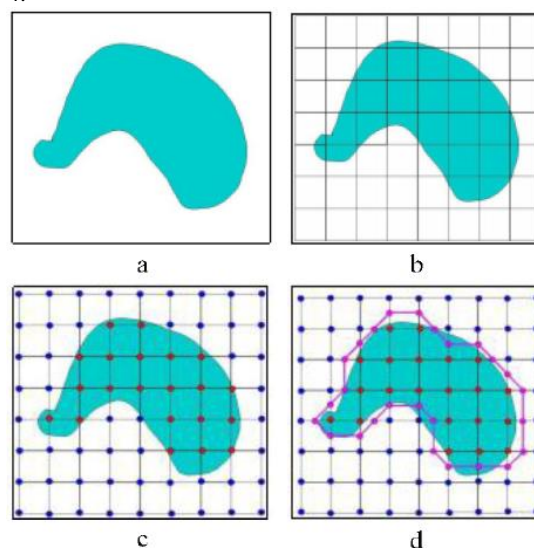


Fig. 3 a) The two-dimensional image. (b) Divide the image into squares. (c) Differentiate the vertices either inside or outside the image. (d) Approximation of the original image.

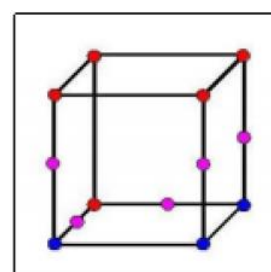


Fig. 4 The cube labeled the intersection points (purple dots) of object with edges between the vertices

In marching square technique, the lines draw between each purple dot to approximate the surface. In marching cube, the lines drawing like Figure 5(a). Triangulate the cube where the filled triangles are used to represent the surface passing through the cube as shown in Figure 5(b).

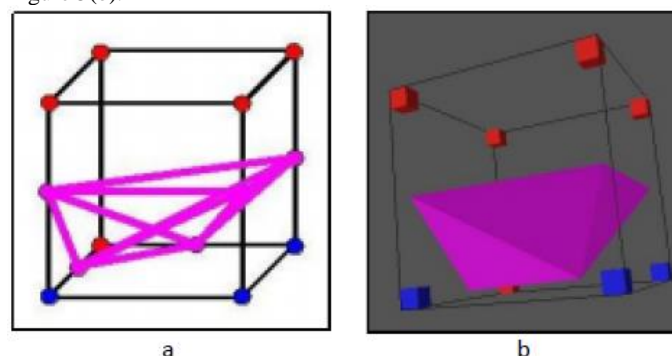


Fig. 5 a) The lines connect the purple dots in marching cube. b) Triangulate cube

But, it is actually challenging to draw the line in connect all the purple dots in the cube. It is difficult to tell which three dots to connect in which triangle. If randomly connect the dots in the triangle, incorrect shape of the object will obtain as shown in Figure 6.

The problem can be solved by a lookup table. As I mentioned above, there are 256 cases. By eliminate the symmetry and rotation cases, 15 unique triangulations can be obtained as shown in Figure 2. Based on the 15 triangulation cubes combination as shown in Figure 3.2, a linear interpolation method is used to calculate the surface intersection point with the edge of the cube (Faming and Xinglong, 2010).

Figure 7(a) shows a blocky rendering of a foot. Better rendering results can be achieved by using a smaller cube which increases the result accuracy. The rendered image by smaller cubes is shown in Figure 7(b). Besides that, there is another way to improve the rendered image which moves the intersection points towards the actual intersection points of the surface with the edges instead of placing the intersection points in the midpoint of inside and outside vertices.

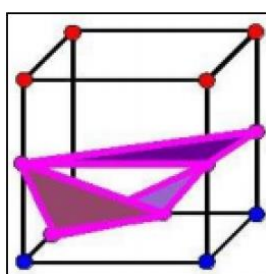


Fig. 6 Bad Triangulation object

Figure 7(c) display the very tiny and pretty rendered foot image. According to Congote et al. (2010), by using the equivalent grid size, it can produce a preferable resolution which can increase the number of cells that containing the surface by putting the vertices of the cube grids towards the surface. But, it is not perfect yet. In other words, the problem is that cannot estimate the exact intersection points in between the slice. The only choice is to guess the location and place the points in the midway between the data slices. The only information between the layers is every point on the top of the cube is inside and on the bottom is face out.

Fortunately, Gouraud shading can use to render the object, which computes the average surface normal for each vertex at that point by average the normal of all triangles sharing that vertex. Besides that, the vector normal is used to compute the lighting at that point. The effect is to blur the appearance of steps without changing the actual geometry of the object.

As a conclusion, the realism is achieved by calculated the normalized gradient and number of triangle can be reduced by surface cutting and connectivity. The advantages of marching cubes are that it is a simple rendering and manipulation and it produces high solution output. However, the disadvantage of this technique is the complexity of the model and required high memory usage to store the resulting surface.

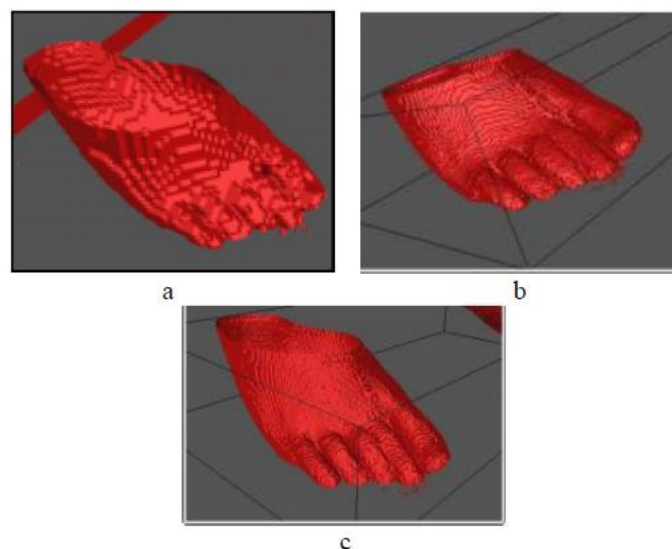


Fig. 7 Different rendering method for foot data. a) Blocky rendered foot. b) Improve rendered image by using smaller cube. c) Rendered image with improvement by move the midpoint of inside and outside vertexes towards the actual intersection points.

VASCULAR MODELLING TOOLKIT (VMTK) – TOOL FOR VOLUME VISUALIZATION

VMTK contained the libraries and tools for geometric analysis, three-dimensional reconstruction, surface data analysis and mesh generation for image-based modeling of blood vessels. The toolkit was composed of VTK and Insight Segmentation and Registration Toolkit (ITK) algorithm in C++ classes. VMTK was used to do the vascular segmentation structures from medical images.

There are others medical visualization toolkits such as MedVis and Simvascular. These toolkits also can be used to extract surface of vascular structures but VMTK comes with cleaner and simpler user interface, allowing beginners to use the software easily and produce the same result with fewer steps.

This section focused on segmenting the aorta from the medical image and extract the surface for future use. The tool used to assist this task is the Vascular Modelling Toolkit (VMTK). VMTK is a script based toolkit, but there is GUI-enabled interface for certain operation that requires specific user input. This data is collected from a healthy patient. Figure 7 is the screenshot of the image data on three planes. The aorta structure does not have any dilation or ballooning on the wall.



Fig. 7 Aorta data from healthy patient

The data are loaded into the toolkit using VMTK terminal. Module `vmtklevelsetsegmentation` is chosen for this process. A window is open after the data is successfully loaded. The image plane is adjusted until the aorta is visible. Window level also adjusted to

enhance the image. Initialization method used to start the segmentation process is the colliding fronts. Two points are placed on the image before starting the initialization process. Figure 4.17 shows the position of the points in red color. The point will be used to test the segmentation process of lower terminal of the aorta.

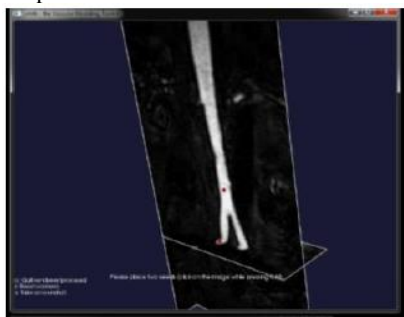


Fig. 8 Colliding fronts point placement

The result of the initialization process is mapped on the image so that the volume can be observed and analyzed. The process is then followed by the evolution process which the volume will be extended toward the edge of the aorta wall. Figure 9 shows the result of the initialization and the evolution process. The surface after evolution process has improved a lot compared to the result of the initialization process.

Because of the colliding fronts initialization technique limitation, the process needs to be repeated in different areas of the aorta. The initialization point can be placed inside the result of the past evolution process, making merge process easier. The result after merging the partial evolution is shown in Figure 10. The renal arteries also included in the initialization process since the size is large enough for the initialization process.

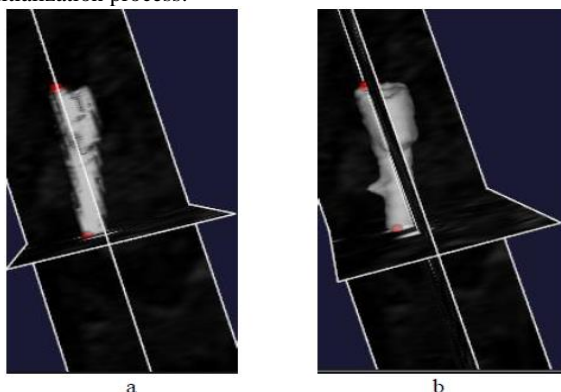


Fig. 9 Result comparison between before evolution (a) and after evolution (b)

ADVANCES AND ISSUES

Volume visualization technique has evolved throughout the years. Researchers are focusing toward automated volume visualization technique which allows human organ or structure to so user. The challenge in this area is to reduce the processing time as the medical image resolution increases. On the other hand, having high-resolution medical image allow researchers to produce high-quality volume visualization because there is less noise in the images.

The user also can use the available tool instead of creating a medical visualization application from scratch. There are many available functions in the application that allow volume visualization of soft tissues up to the bone structure. This application also comes with an additional function to enhance the visualization result. Users are suggested to survey the application based on their needs and knowledge. Some of the application require certain technical knowledge to use the application. So, having basic knowledge related to data visualization may help the user to navigate around the visualization application. For user point view, friendliness interfaces make one of the characteristics should be determined.

CONCLUSION

This chapter cover basic knowledge about the volume visualization techniques suggested visualization application and a specialized tool used to extract vascular structure from the medical images. Section 2 explained briefly 3D volume rendering technique which is isosurface and marching cube technique used to generate volume visualization. Lastly, section 3 describes a specialized tool which is used to extract vascular structure and export it into other type file format. Volume rendering visualization has been part medical imaging pipeline. The technique behind this process area that unchanged and by describing this knowledge provide a start point for the clinical researcher.

ACKNOWLEDGEMENT

This work was financially supported by the Universiti Teknologi Malaysia under the Research University Grant and Ministry of Higher Education Malaysia and eScienceFund from Ministry of Science, Technology and Innovation Malaysia with reference number R.J130000.7928.4S135.

REFERENCES

- Bischoff, S. & Kobbelt, L.P., 2002. "Isosurface reconstruction with topology control". In *10th Pacific Conference on Computer Graphics and Applications*, 2002. *Proceedings*. IEEE Comput. Soc. 246–255.
- Gong, F. & Zhao, X., 2010. "Three-Dimensional Reconstruction of Medical Image Based on Improved Marching Cubes Algorithm". In *2010 International Conference on Machine Vision and Human-machine Interface*. IEEE. 608–611.
- Klein, T., Stegmaier, S. & Ertl, T., 2004. "Hardware-accelerated reconstruction of polygonal isosurface representations on unstructured grids". In *12th Pacific Conference on Computer Graphics and Applications*, 2004. *PG* 2004. *Proceedings*. IEEE. 186–195.
- Lorensen, W.E. & Cline, H.E., 1987. "Marching cubes: A high resolution 3D surface construction algorithm". *ACM SIGGRAPH Computer Graphics*, 21(4).163–169.
- Nicol, E. et al. eds., 2011. *Cardiovascular Computed Tomography*, Oxford University Press.
- Raman, S. & Wenger, R., 2008. "Quality Isosurface Mesh Generation Using an Extended Marching Cubes Lookup Table". *Computer Graphics Forum*, 27(3). 791–798.
- Stergiopoulos, S., 2009. "Signal Processing Concept Similarities among Sonar, Radar, and Noninvasive Medical Diagnostic Systems". 1–22.
- Kolivand H. and Sunar M.S. 2014. "Realistic Real-Time Outdoor Rendering in Augmented Reality." *PLoS ONE*, Vol. 9, Issue 9.