RECONSTRUCTION OF THE BOUNDARY REPRESENTATION OF THE GREAT VESSEL FROM CARDIAC CT IMAGES

Yu-Tai Ching Yu-Sun Wong Min-Ta Yu
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan R. O. C.

Abstract

We present a prototype system for reconstruction of the boundary representation for a great vessel from a set of CT images. The system includes segmentation of great vessel, finding the contours for the boundary of the great vessel, and triangulation a pair of corresponding contours. In this system, we tried to reduce the user interface. The subtasks that still need many user interfaces are the segmentation step and determining the corresponding contours that need to be connected. Many Computational Geometry techniques are employed so that we can generate a good quality triangulation.
Key Words Medical Imaging,Surface Reconstruction, Solid Modeling, Computational Geometry.

1 Introduction

CT images contains the information of the anatomic structure of human body. This property increases the applications of CT image in clinic especially with the help of a computer. Marching cube method for volume rendering is one of such applications. Marching cube method finds the isosurface for given thresholds and present the isosurface by using polygonal patches. We then are able to use the computer graphics technique to render the polygonal patches so that we can see the area of interest non-invasively.

Recently, people look for more sophisticate applications such as planning and simulation of a surgery[3]. Suppose we can have a solid representation of the area of interest, then we could simulate a surgery process or we can do the planning of the surgery and predict the result of the surgery. This paper describe a prototype system toward this goal.

The subject we studied in this paper is the great vessel of a heart in the CT images. We plan to reconstruct the boundary representation of the great vessel from a set of CT images. Our prototype system includes the following subtasks.

1. Segmentation: We have to identify the great vessel from CT images and find the boundary of the great vessel.

2. The boundaries of the great vessel are actually boundary points. We need to calculate the contours (polygons) from the boundary points.

3. It is hard to have an image processing technique which is able to identify a precise great vessel. We design an user interface method to remove those not on the great vessel.

4. To reconstruct the boundary representation for the great vessel. Since CT images are parallel to each other, it is a natural approach to connect two contours in two

consecutive slices by using triangle patch. We use [5] approach which converted this problem to a problem of finding the shortest path in a grid graph. We also use 3D Delaunay triangles to improve the quality of the solid been reconstructed.

In the next section, we shall describe these subtasks in details. In the last section, we present some results.

2 The Prototype System

In this section, we present each subtask in details.

2.1 Segmentation

Before we present the segmentation subsystem, we briefly describe the CT images we studies. The areas of interest in the images are the great vessel and the heart. In radiology, the great vessel and the heart are intensified by putting contrast media into the circulation system. The contrast media makes the area of interest generally having intensity greater than the soft tissue but less than bone. However, the intensity for the area of interest is not a constant for all the cases. The intensity depend on the dose of the contrast media, the weight of the patient, and the time between the contrast media was putting into the patient and the images were acquiring. Each CT scan has dimension 512 by 512 and there are 4k grey scales in a set of volume data.

Since we are dealing with CT images, we take advantage of the fact that different material has different CT value. The segmentation process is based on the intensity. We also know that the area of interest has intensity less than bone but more than soft tissue. We segment the images points whose intensity is in between a given interval. Not all the image points having intensity in the interval is an image point of area of interest. This problem is solved by finding the connected components of the thresholded image points. Generally speaking, the largest connected component is the set of image points of heart and great vessel. Since the heart and the great vessel are connected in natural, we cannot identify those image point of great vessel only. User interface is the only way to separate the heart and the great vessel. The boundary points of the connected component is a set of points of the endocardial surfaces.

2.2 Closed Contour for Boundary Points

We found a set of boundary points of the area of interest. However, what we need is a close contour which represents the boundary of the area of interest. To convert the set of boundary points, we design and implement a system which systematically calculate the closed contour no mater what the shape is. The proposed method is based on the α-shape and need a *doubly connected edge list* (DCEL) data structure for implementation [1]. The proposed method finds two closed contours that the set of boundary points are between these two contours.

α-shape mathematically define the shape of a set of points in space. The α-shape of a set of points can be obtained from the Delaunay triangulation of the set of points. Our approach takes advantage of the concept of the α-shape but it does not follow the definition of the α-shape. Recall that we are looking for the pair of contours which sandwiches the set of boundary points. We first calculate the Delaunay triangulation of the set of boundary points and the four corners of the image. For a given α, we consider the following cases of a Delaunay triangle t.

1. All sides of t are greater than α. In this case, the 3 points of the triangle are not connected. We discard these 3 edges.

2. All sides of t are less than α. In this case, the 3 points of the triangle are connected and these 3 sides should be considered in the interior of a component. We also discard these three edges.

3. One of two edges are greater than α and the others are less than α. In this case, the sides which are less than α are boundary of the shape. These boundary edges should be connected to form the contours.

Note that, the above approach does not always find two closed contours. There are cases that an edge with only one end point anchored at a contour or two contours connected by an edge. In order to deal with all these cases, we need a formal data structure to represent the planar graph produced by the above method. The data structure is the doubly connected edge list. This data structure support the operation to find the polygonal path that enclose a face. By using DCEL, we are able to find all the polygonal paths that enclose all the surfaces. In most of the cases, the data structure reports two contours, i.e., one is enclosed by the other. In this case, we take the path that has more vertices on it. If there are more than one path reported, we take the outermost path.

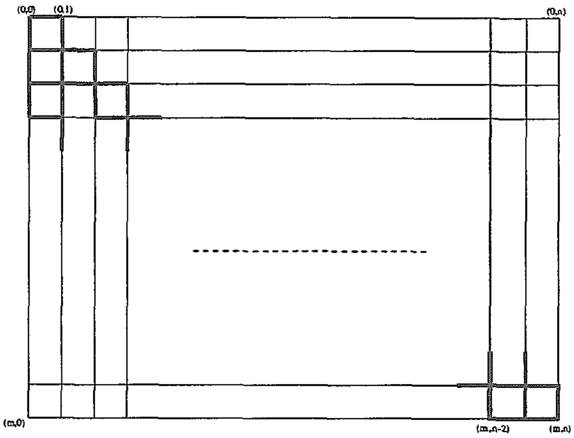

Figure 1: A toroidal graph corresponding to two contours c_1 and c_2. Assume that the line segment connecting vertex 0 in c_1 and vertex 0 c_2 must be in the triangulation, we find the shortest path from $(0,0)$ to (m,n). The bold edges corresponding to the triangles in the Delaunay Triangulation.

2.3 User Interfaces

The user interface subsystems are required in the following cases.

1. To specify the pair of contours in consecutive slices that should be connected by triangles.
2. A contour encloses more than one material, for example heart and great vessel. This case occurs when the image processing technique couldn't separate two different material in an image.
3. When there are branches of the area of interest between a pair of consecutive slices.

Very often, there are more than one contour that should be connected together. It is very hard to give a rule (a piece of computer code) to determine the connectivity of contours especially when we are processing patients with congenital hear problem. It is required that a physician to point out the connectivity between contours. Our user interface system displays two consecutive slices and let a user to select a contour from each slice for triangulation.

Case 2 and case 3 are handled identically that we split part of the contour away from the original contour. We always split a contour to two contours by adding a line segment which has both end points on the contour. The splitted contours are part of the contour concatenate with the line segment. The user interface allow us to select the location of the points by using keyboard input. In case 2, we simply split the contour in to two. In case 3, we split a contour followed by a case 1 selection. Although the length of the newly added line segment is generally relative longer than the line segment originally on the contour, the surface reconstructed from the contours by using this method does not have unpleasant artifact.

2.4 Triangulation

2.4.1 Shortest Path approach

Reconstruction of the boundary of a solid from two contours in the parallel slices is a problem received intensively studied. In this work, we use a combinatorial optimization approach. We transform the triangulation problem to a problem of searching shortest in a *toroidal graph*.

As shown in Figure 1, a toroidal graph consists of squared meshes. An m rows by n columns toroidal graph corresponds to a pair of closed contours c_1 and c_2 of length m by n which should be connected by using triangle patches. A horizontal edge $(i, i+1)$ on row j corresponds to a triangle whose 3 vertices are respectively the i'th and the $(i+1)$'th vertices on c_2 and the j'th vertex on c_1. Similarly vertical edge $(j, j+1)$ on column i corresponds to a triangle whose 3 vertices are the j'th and $(j+1)$'th vertices on c_1 and the i'th vertex on c_2. A sequence of triangles representing the boundary of solid between these pair of slices corresponds to a path from $(0,0)$ to (m,n) in the Toroidal graph. The toroidal graph is a weighted graph that each edge has a weight corresponding to the cost of a triangle. The possible costs for a triangle can be:

1. the total length of the sides connecting the two contours, or
2. the area of the triangle connecting the two contours.

In our experiment, there was no significant difference between the results obtained from these two cost functions.

Suppose that we look for the shortest path in the graph from $(0,0)$ to (m,n), we assume that the path must contain a line segment connecting vertex 0 in c_1 and vertex 0 in c_2. The optimal triangulation may not contain those edge containing the two vertex. In order to find the optimal triangulation, we find all the shortest paths containing $(0,i), i = 1 \ldots n$. In our experiment, the triangulation obtained from the short path approach always looks good in it shape and the variation of surface normal (looks smooth). The difference between the optimal triangulation and any other "good" triangulation may nor be significant. But its performance is very stable and definitely better than an "heuristic" triangulation most of the time.

2.4.2 Delaunay Triangulation

Very often, we wish that the triangles in the triangulation do not contain a very "sharp" triangle. The shortest path approach stated above does not prevent a triangulation containing such triangles. We propose to employ the Delaunay Triangulation. In k-D space, the Delaunay Triangulation of a set of points has a property that a "sphere" in k-D space passing through $k+1$ points of a Delaunay Triangle does not contain any other points. This nice property suggest that a triangle in the Delaunay Triangulation has little chance that it is very sharp (otherwise the sphere passing through the triangle could be large enough to enclose other points). Based on the observation, we prefer to include as much as the Delaunay Triangles in the triangulation.

For a given pair of contour c_1 and c_2, we first calculate the Delaunay Triangulation in 3-D space[13]. Note that, since there are only 2-D triangles in a triangulation, for each Delaunay Triangle, there are four 2-D triangles. Let the set of 2-D triangles be denoted $T = \{t_i | i = 1, \ldots, M\}$. A triangle t_i can be included in a triangulation if t_i contains an edge in c_x and a vertex in c_y, $x, y = 1, 2$ and $x \neq y$. Each such triangle has a cost so that we can assign the cost to the weight of an edge in the toroidal graph. Since there are cases that vertex $(0,0)$ and (m,n) in the toroidal graph may not be connected through those edges of Delaunay triangles, a triangulation must contain some other edges not in the Delaunay Triangulation. However we prefer to include an edge in the Delaunay Triangle first. Our algorithm to achieve this requirement is to give those edges not in the Delaunay Triangulation having a cost 10 times larger than the cost if it is a Delaunay Triangle.

3 Conclusion and Result

We develop a prototype system that construct the boundary representation of the great vessel from CT images. The system includes segmentation of the great vessel, find the boundary point of the segmented great vessel, convert the boundary points to closed contours, and find a triangulation to connect a pair of contours. For the subtasks converting boundary points to contours and finding the triangulation, the proposed approached do not need any user interface. There are still processes that need user interfaces especially when the great vessel did not correctly segmented and when we need to decide which two contour should be connected. A "smart" system to automatically do these tasks are proposed as future work.

The reconstructed great vessel are shown in the Figure 2. The number of triangles is 10000. The time to reconstruct a case is about an hour. However, the time required can be reduced by improvement of the software. The bottle neck will be the process of determine the corresponding contours.

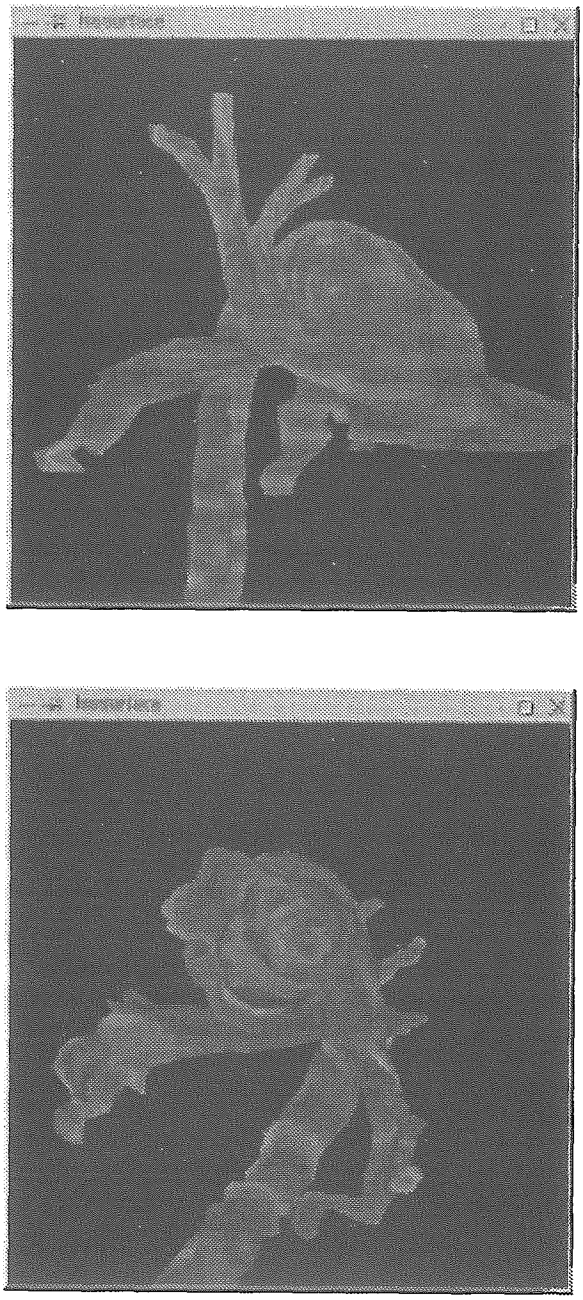

Figure 2: Two rendered result of a reconstructed great vessel.

References

[1] F. P. Preparata and M. I. Shamos, *Computational Geometry,* Springer-Verlag, New York, 1985.

[2] D. Knight and J. Kim, "Surface reconstruction of concealed objects," Twenty-fifth Southeastern Symposium on System Theory, 1993. Proceedings on SSST '93, pp. 529-532, Mar. 1993.

[3] P. Block and J. K. Udupa, "Application of computerized tomography to radiation therapy and surgical planning," Proc. IEEE, vol. 71, pp. 351-372, Mar. 1983.

[4] E. Keppel, "Approximating complex surfaces by triangulation of contour lines," IBM Journal of Research and Development, vol. 19, pp. 2-11, Jan. 1975.

[5] H. Fuchs, Z. M. Kedem, and S. P. Uselton, "Optimal surface reconstruction from planar contours," Commun. Ass. Comput. Mach., vol. 20, pp. 693-702, Oct. 1977.

[6] H. N. Christiansen and T. W. Sederberg, "Conversion of complex contour line definitions into polygonal element mosaics," Comput. Graphics, vol. 13, no. 2, Aug. 1978.

[7] L. T. Cook, S. J. Dwyer III, S. Batnitzky, and K. Rak Lee, "A three-dimensional display system for diagnostic imaging applications," IEEE Comput. Graphics Applications, vol. 3, no. 5, pp. 13-19, Aug. 1983.

[8] B. Wyvill, C. McPheeters, and G. Wyvill, "Data structure for soft object," The Visual Computer, vol. 2, no. 2, pp. 235-242, 1986.

[9] W. E. Lorensen and H. E. Cline, "Marching cubes: a high resolution 3-D surface construction algorithm," Computer Graphics, vol. 21, no. 4, pp. 163-169, 1987.

[10] J. D. Boissonnat, "Shape reconstruction from planar cross sections," Computer Vison, Graphics, and Image Processing, vol. 44, pp. 1-29, 1988.

[11] W. C. Lin, C. C. Ling, and C. T. Chen, "Dynamic elastic interpolation for 3-D medical image reconstruction from serial cross sections," IEEE Trans. On Medical Imaging, vol. 7, no. 3, Sep. 1988.

[12] D. T. Lee and B. J. Schacter, "Two algorithms for constructing a Delaunay triangulation," Intl. J. Comput. Inform. Sci., vol. 9, no. 3, pp. 219-242, 1980.

[13] K. L.Clarkson, K.Mehlhorn, and R.Seidel, "Four results on randomized incremental constructions," Comp. Geom.: Theory and Applications, pp. 185-121, 1993.

[14] K. L.Clarkson, "Safe and effective determinant evaluation," Proc. 31st IEEE Symposium on Foundations of Computer Science, pp. 387-395, Oct. 1992.

[15] Y. K. Choi and K. H. Park, "A heuristic triangulation algorithm for muliple planar contours using an extended double branching procedure," The Visual computer, vol. 10, pp. 372-387, 1994.