

Tessellated Surface Reconstruction from 2D Contours

Chee Fatt Chan¹, Chee Keong Kwoh², Ming Yeong Teo¹ and Wan Sing Ng¹

¹School of Mechanical and Production Engineering
p7105753d@ntu.edu.sg
<http://mrcas.mpe.ntu.edu.sg/>

²School of Applied Science
Computer Integrated Medical Intervention Laboratory
Nanyang Technological University, Nanyang Avenue, Singapore 639798

Abstract. This paper presents a new triangulation method to define surfaces of 3D object from parallel 2D contours. These contours represent the boundary of a human organ segmented from 2D images acquired from radiological volumetric data using ultrasound (US), computer topography (CT) and magnetic resonance (MR) imaging. Many papers have identified and looked into the problems of generating surfaces from 2D contours. They are correspondence, tiling, branching and surface-fitting problems. Our new algorithm dealt with these problems in three steps. First, several adjacent contour mapping positions are used to reveal the correspondence of contours. Second, a tessellation algorithm approximates contours into line segments. Finally, surface meshes, in the form of strips and fans, use the gradient of line segments to optimise surface quality without affecting the rendering time.

1 Introduction

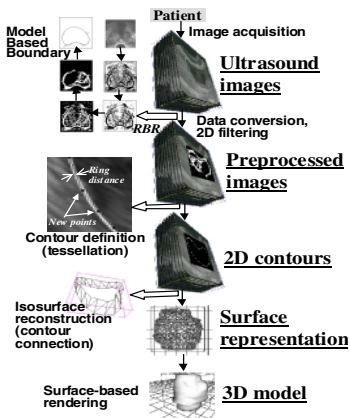


Fig. 1. URO-ART 3D-imaging pipeline.

3D models can provide more detailed and precise anatomical information than 2D image as they are important in determining the type of surgical indications, e.g. surgery planning, radiation therapy planning and volumetric measurements. Figure 1 shows a 3D-imaging pipeline approach in which a 3D prostate model is reconstructed from its contours from 2D-ultrasound images. Currently, this pipeline is implemented on a system called Augmented Reality for Therapy in Urology (URO-ART) where 3D prostate model is used for prostate cancer biopsy [1]. After the acquisition of a series of topographic images from a patient, the data usually undergo some pre-processing that

This work is supported by National Science and Technology Board (NSTB) in Singapore and the Ministry of Education (MOE) of Singapore for a joint funding under the strategic research grant JT ARC17/97.

include data conversion and image filtering. Then a set of object contours is defined on every topographic image. In our case, outlining the object boundary (e.g. prostate) of ultrasound images is done by employing a set of segmentation processes consisting of image pre-processing, edge enhancing using Radial Bas-Relief [2] followed by a model-based boundary extraction [3]. To form a 3D structure, the contours that use polygonal approximation [4] from adjacent cross sections are connected. One of the disadvantages of this approach is that ambiguities arise when attempts are made to connect equivalent points on contours. For medical visualisation, shapes are often complex and vary greatly from one cross section to the next. Furthermore, there is no guarantee that the edge segmentation will detect only one contour in each slice.

2 Reviewing 3D Surface Reconstruction Techniques

It is common to extract from multiple parallel 2D slices of image into digitised 2D contours representing the object boundary. Ways to reduce the amount of points as to represent the 2D digitised contours are using polygonal approximation [4], contouring [5] and 2D marching squares [6]. 2D Marching squares algorithm (has been extended to 3D voxel known as 3D marching cubes) uses a divide-and-conquer technique that treat cells independently. After defining the 2D contours, the next step is to do edge linking. Running along the contours, surface triangles are constructed between adjacent planes. One of the different linking techniques available is proposed by Giraudon [7] based on an efficient data structure. Various conditions may be imposed: maximise the volume, minimise the surface, and minimise the edge length or angles, as proposed by Keppel et al. and Fuchs et al. [8,9]. However, their solutions are limited to one single contour on each cross section.

One of the difficulties in current techniques in surface modelling is that they are not efficient enough to handle complex topology of natural objects. For example, the surface reconstruction step throws away most of the valuable information on cross sectional images. Once the surface representation is created, there is no way of getting back to the original intensity values. Even simple volume cuts are meaningless because there is no information about the interior of an object. Furthermore, every change of surface definition criterion requires a recalculation of the whole data structure. However, as compared with volume reconstruction, surface reconstruction is fast in the creation of images for a wide variety of data and objects. A clear advantage of surface based methods is that a very high data reduction from volume to surface representations may be achieved. This affects both memory requirements and computing times. Computing times can be further reduced if the surface representations are based on common data structures such as triangle meshes, which are supported by computer graphics workstations. For a successful surface reconstruction, the following problems have to be resolved:

Tiling problem consists of finding the best set of triangulated facets to define the surface between a pair of connected contours in adjacent sections. Keppel et al. [8] proposes a graph search optimisation algorithm and Christiansen et al. [10] designs a heuristic approach to deal with this problem. The optimisation methods have computational overhead in comparison with the heuristic approach. On the contrary, the heuristic algorithms require pairs of contours that are to be aligned and similar in shape to work successfully. Base on Keppel's optimising algorithms to solve the

problem of tiling two contours, Fuchs et al. [9] expresses the graph search using Euler tour of a toroidal graph. Ekoule et al. [11] develops an approach to tile two dissimilar contours on the assumption that their convex hulls are similar. Bajaj et al. [12] defines a set of criteria to constrain the undesired tiling and to allow detection of branching regions and dissimilar portions of a contour.

Correspondence problem involves finding the correct connections between the contours of adjacent slices. Meyers et al. [13] and Soroka [14] approximate the contours by fitting them with ellipses and then assembling them into cylinders to determine their correspondence. Bajaj et al. [12] addresses this problem assuming that reconstructed surface between adjacent slices can have at most one intersection with any line perpendicular to the slices. This limits some of the actual contours to correspond correctly in a topological manner.

Branching problem occurs when a contour in one slice can correspond to more than one contour in an adjacent slice. Boissonnat [15] uses Delaunay triangulation to compute vertices of contours in two adjacent slices. Lin et al. [16] combines surface interpolation to transform vertices of contours to create smooth surfaces that can handle the branching problem. This requires large number of triangles. Meyers et al. [13] and Ekoule et al. [11] use composite contours to connect the adjacent contours at the closed points. Their branch handling do not correspond well to the actual physical object. Bajaj et al. [12] dealt with branching problem by using a tiling algorithm that makes several passes. His initial few passes are able to tile as much of the surface as possible before tiling the “difficult” regions such as branching regions, holes and dissimilar areas of contours.

Surface fitting problem involves finding a smooth surface that either interpolates or approximates the vertices of the mesh and maintain the same topology. Miller et al. [17] approximates an isosurface of volume data by fitting a mesh homomorphic to a sphere that can be extended to finding of isosurfaces of topological type. Meyers et al. [13] uses a surface fitter on final constructed surface to regenerate surface. Bajaj et al. [12] uses 2D marching squares algorithm [6] to generate contour segments and links them to form simple polygons. The generated contour segments have coarsely spaced vertices for an accurate surface fitting of different types of curvature.

3 New Reconstruction Method

In this paper, we propose a new surface reconstruction algorithm based on the following general requirements:

1. Shorter computational time,
2. Maximise the use of triangular mesh and triangle strips,
3. Minimise the use of large sampling for a given set of points,
4. Surface quality can be maintain even data resolution is change or become spatial,
5. Restrain complexity of algorithm for easy implementation in an NT system.

To obtain a topologically accurate surface model of an object using its stacks of parallel 2D contours, the model must approximate the physical boundary as well as possible. Ideally, this can happen if the distributing distance between surface vertices is small and the vertices are set on the 2D contour line. Therefore, for our implementation, we assume that our 3D-reconstruction method is built from accurate and complete (no opening and intersection of contours) 2D digitised contours that

represented the boundary of the physical object. The digitised contours are extracted using segmentation methods such as the Model based RBR method [3] for ultrasound images, and global intensity threshold method (found in Materialise Mimics software¹) for CT and MRI images. To ensure our tiling method is correct, our algorithm follows the guidelines stated in Bajaj et al. [12]:

1. For two dissimilar contours, tiling problem can occur when connecting every vertex of one contour to another contour,
2. Re-sampling of the reconstructed surface should yield the original contours.
3. Composite contours should not be formed; they do not correspond well to the actual physical object.

3.1 Contour Tessellation

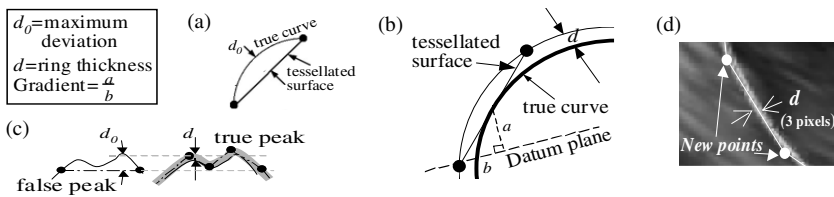


Fig. 2. (a) Principal of curve tessellation. (b) New contour tessellation. (c) Key advantage of new tessellation, showing the peak point, over the method of using only the deviated height. (d) New points created within 3 pixels error of ring thickness.

Figure 2(a) shows the basic principal of contour tessellation. Tessellated surface results because of the deviation (d_0) between the true curve and the tessellated straight line. Figure 2(b) shows the new approach of contour tessellation, this method can represent surfaces that deviation (d) from the straight line rather than representing the maximum deviation. The key step is by enlarging the true curve with a bounding ring thickness d . This new curve is set as a new boundary that can force the tessellation process to distribute a new set of point according to the curvature of the contour. One of the most important aspects of this new approach is that the topological consistency is checked by using gradient value ($\nabla = \frac{a}{b}$) of the true curve whenever a new point is set. Each vertex point is defined with reference to its neighbour linkage. Figure 2(c) shows the difference between curve tessellation and new approach of contour tessellation. The objective of this process is to use optimal number of line segment to represent a raster object data (e.g. point cloud data) without losing the shape and nature of the object. Advantages of this process to construct a surface are as follows:

1. It optimises the connecting of new vertices over a surface,
2. It uses an estimate of surface curvature to distribute more new vertices at regions of higher curvature.

Let the true digitised curve to be approximated be denoted as $P, [p_1, p_2, \dots, p_n]$. The tessellated curve is $P', [p'_1, p'_2, \dots, p'_m]$ such that m is significantly smaller than n , and the chain of line segments is defined as $[p'_1, p'_2], [p'_2, p'_3], \dots, [p'_{m-1}, p'_m]$. The vertices of P' are ordered subsets of the vertices of P . The proposed tessellation algorithm starts at $[p_1, p'_1]$, which can be selected arbitrarily, until $[p_n, p'_m]$, or a new

¹ Mimics 6.0 is a trademark of Materialise Software

initial vertices causing the number of polygon sides to increase (This is explained in Section 3.3). In order to demonstrate the effectiveness of this new algorithm, the choice of method used to measure the error of approximation of curve P and P' are subject to the pixel density region error criterion. This criterion defines the goodness of fit in terms of pixel density deviations of the output curve P' from the input curve P . The error measure used is the pixel density (e = number of deviated pixels per line segment $[p'_{m-p}, p'_{m}]$) between a vertex of p and the line collinear with the corresponding approximating line segment. It is difficult to use mathematical functions to represent the digitised curve; the implementation is depended on the pixel resolution of the input data. Therefore, the idea of contour tessellation algorithm (illustrated in Figures 2(c) and (d)) can be divided into two-stages:

1. Sampling the true curve with a dynamic searching pixel mask ($d \times$ two pixels) within the bounding ring of thickness d .
2. Setting the new vertex point and joining line segments $[p'_{i-p}, p'_i]$ by sampling the optimal pixel gradient that satisfied any of the following situation:
 - Convex: p'_{i-1} on outer, $p'_{i/2}$ on inner and p'_i on outer ring positions.
 - Concave: p'_{i-1} on inner, $p'_{i/2}$ on outer and p'_i on inner ring positions.
 - Convex to concave: p'_{i-1} on outer, $p'_{i/2}$ on outer and p'_i on inner ring positions.
 - Concave to convex: p'_{i-1} on inner, $p'_{i/2}$ on inner and p'_i on outer ring positions.

3.2 Tiling Using Triangle Strips and Fans

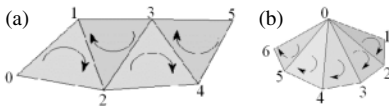


Fig. 3. (a) Triangle strip and (b) fan.

The speed of constructing triangulated surfaces for displaying is crucial to interactive visualisation; this is bounded by the rate at which triangulated data can be sent to the graphics subsystem for rendering. Each triangle can be specified using three vertices, but to maximise the use of the available data bandwidth, it is desirable to order the triangles so that the next triangle shares an edge. Using such an ordering, only the incremental change of one vertex per triangle need to be specified, potentially reducing the rendering time by factor of three by avoiding redundant clipping and transformation computations. Besides, such an approach also has benefits in compression for storing and transmitting models. Consider the triangulation in Figure 3(a). Without using triangle strips, this triangulation will be represented by specifying four triangles with three vertices each. By using triangle strips, as supported by the OpenGL² graphics library [18], the triangulation can use the strip (0,1,2,3,4,5). Such a sequential strip can reduce the cost to transmit n triangles from $3n$ to $n+2$ vertices. The problem of constructing quality triangle strips has received attention from both the graphics and the computational geometry communities. For example, Akeley et al. [19] has written a SGI² program that converts triangle meshes to triangle strips. According to Evans et al. [20], it is a “greedy” algorithm, which always chooses the triangle that is adjacent to the least number of neighbours (e.g. minimises the number of adjacencies) as the next triangle in a strip. Evans also describes other triangle strip generation algorithms as well as presenting some of the most effective techniques to date. He

² OpenGL 1.1 and SGI program are trademark of Silicon Graphics, Inc.

proposes an algorithm for constructing triangle strips from fully triangulated models. This has proved to be quite useful for generating efficient triangle strips for the OpenGL cost model. In the case of generating triangulated mesh from contours, the algorithm to determine sequential of strip can be simpler than Evans’s algorithm. Figure 4 shows an example of linking a contour with concave arc and a straighten contour. This requires extension of tessellation gradient checking process that will close the connectivity surfaces between the low curve and the high curve. The connectivity surface can then be created through using one sequential triangle strip and one sequential triangle fan. Getting good sequences of triangle strips when multiple dissimilar contours are involved is discussed in the next Section.

3.3 Topological Consistency Checks

Fig. 4. Triangulating top contour to bottom contour by using one sequential triangle strip and one sequential triangle fan.

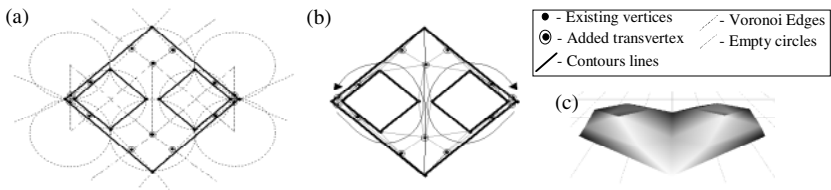
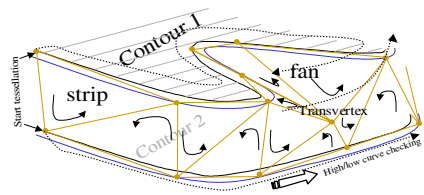


Fig. 5. Topological consistency checking. (a) Voronoi Diagram edges. (b) Triangulating top (2 diamond) to bottom contour (1 diamond) using two sequential strips. (c) Rendered model.

Gradient checking is used when tessellation shows that adjacent contour curvature has changed drastically. This change can be determined by the following conditions:

1. There is poor triangulation of any existing neighbour vertices that strayed away. This is checked using suspect line segments matching with Voronoi diagram [15] (Delaunay triangulation is not used) as a 2D template from the top, as shown in Figure 5(a). The Intersection of suspect line and Voronoi edge is used.
2. There is discontinuation of triangle strips along one contour.
3. When the triangulation meets the edge of a gap that is between two contours on the same slice or they are contours that overlapping one contour on an adjacent slice.

If the above conditions are met during gradient checking, new vertices are needed to maintain the surface topological consistency. The additional vertex is named Transvertex. In practice, Transvertex can be effective to handle branching and tiling problems as shown in Figure 4 and Figure 5. The automatic contour tessellation method, mentioned in Section 3.1, is able to create vertices for surface according to the curvature of the contour. For instance, one contour may have a large number of vertices from an original description of an object while adjacent contour may have lesser vertices. By enforcing the use of triangle strips, one might face a problem of tempting to add new vertices on the adjacent tessellated contour line segment. This

problem is also stated in Bajaj et al. [12] first guideline. Furthermore, if more vertices are added to an existing tessellated line segment, this will propagate more new vertices to other consecutive contours even though their existence do not represent the curvature of the surfaces. Figure 6 shows examples of connection paths with and without triangle strips (normal triangulation). Transvertices are used to fulfil the first guideline. Smooth shaded models demonstrate that using tessellated triangle strip has better topological surface appearance. For instance, Figure 6(d) shows concave cone stop at the middle without relying on the adjacent contour and 6(e) shows that the top straight line can be easily identified while the other seems bent in the shadow.

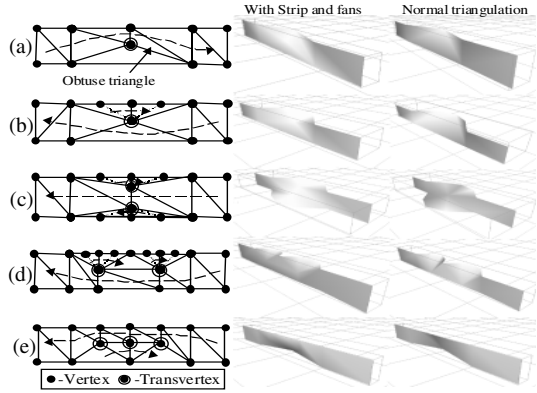


Fig. 6. (a-e) Examples of connection path, with Transvertices, that facilitate the sequential triangle strips within two dissimilar contours. On the right hand side shows the smoothed shaded results when using triangle strips and without using triangle strips.

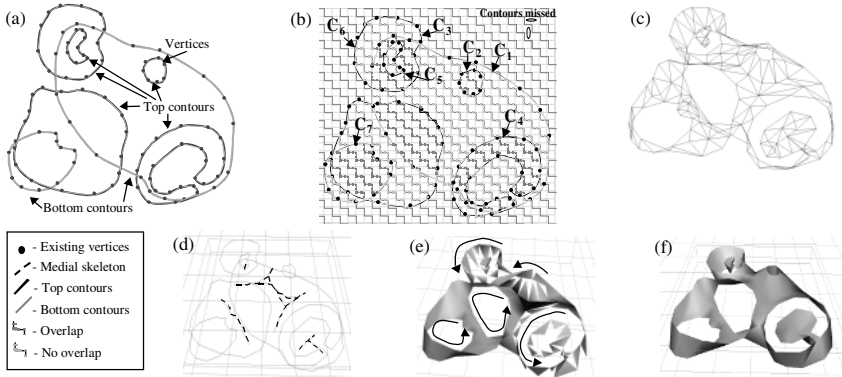


Fig. 7. (a) Two slices, top and bottom, with complex contours (from [12]’s Figure 13(a)). (b) Top slice overlaps bottom slice into one slice for zigzag scan for contours intersection relationship. (c) Wireframe model. (d) Medial axis, the skeleton lines, are created for the canyon and concave terrain island. (e) 6 triangle strips (along C_2 , C_3 , C_4 , C_5 , C_7 and $C_1 \cup C_6$, represented by interval chain of blank triangles). (f) Smooth rendered results using algorithm.

The Voronoi diagram has the property that every vertex in the region around a site is closer to that site than to any of the other sites. This can also be used for approximation of the medial axis called Edge Voronoi Diagram (e.g. shown in Figure 7(d)). Medial axis of a polygon is the locus of points with equal distance to at least two contour points [15]. Calculating the real medial axis can assist the tiling of canyons, islands and any complex branching surfaces. It represents the metric and topological properties of shapes well. There is no composite contour form; therefore it fulfils Bajaj et al. [12] third guideline. Accurate medial axis is numerically difficult to achieve when existing contour vertices are far apart. Bajaj et al. [12] solves this problem using a rough medial axis. We use Bajaj’s calculation of rough medial axis, but in our case, Transvertices need not be located on the medial axis, depending on where the running vertices of sequential triangle strips are. Figure 7 shows the result of our algorithm using two slices of contours of the shape used by Bajaj et al. [12]. Figure 7(e) shows an example of 3 strips meeting in-between canyons and additional transvertices are added by matching Voronoi diagram as mentioned earlier.

3.4 Correspondence of Contours Using Contours Cross-Shadow Mapping

Section 2 shows that many solutions for the correspondence problem use a global view of the data set and incorporate some assumptions about the objects to be reconstructed. This allows a reasonable solution to be generated where no uniquely correct solution is possible. These methods may become more difficult to implement if the distance between slices becomes large. Thus, automatic solution of the

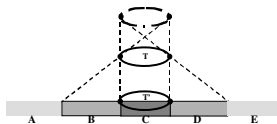


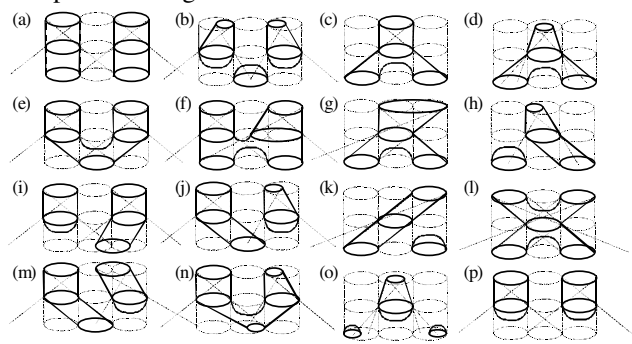
Fig. 8. Regions of cross-mapping method.

correspondence problem in its general form is difficult. Zigzag grid scan (Figure 7(b)) is used to define the location and the overlapping relationship of adjacent contours. Cross mapping (Figure 8) is used to solve the correspondence problem by satisfying the condition that: T is not connected to T' only if T' falls within the constraint,

$$S = AB + \overline{BCD} + DE \tag{1}$$

For Example, (in Figure 9(h)), T is connected to right T' as $S = CDE$, but T is not connected to left T' as $S = AB$. Advantage of this method is that it takes the properties of similar triangles and the slope surface gradient from the last two adjacent contours are included to map the new correspondence region.

Fig. 9. The correspondence problem is solved by determining which of the contours from each section should be connected together. (a-p) Examples of different outcomes possible in using cross-shadow mapping method.



4 Preliminary Results of Implementation

The algorithm has been implemented in Microsoft Visual C++ 5.0³ using WindowNT 4.0³ operating system. The graphics hardware of the system, Dual Intel Pentium II 333MHz TDZ2000 NT graphic workstation⁴, is exploited by using the OpenGL library [18]. The new tessellation-meshing algorithm presented in this paper has been implemented to reconstruct a prostate gland (Figure 10). The model is generated from 19 slices of parallel-digitised 2D ultrasound images, each consisting of 256x256 elements of 8-bit depth, evenly spaced at 3 mm apart. 2D contours are segmented from these images without discontinuity using our new RBR model based method [3]. In order to illustrate and test the full practical detail features of the new algorithm, a human CT scanned pelvic bone is used. The pelvic images consisting of 83 slices (approximately 3mm for slice-to-slice thickness) from 512x512x512 with pixel size of 0.46mm, data are from Materialise's Mimics software (NV, Kapeldreef 60 B3001L). Only 38 slices with slice-to-slice thickness of 6mm are used. We used Mimic's intensity thresholding (range 1250 to 4095) to segment the pelvic bone. 2D Contours polylines (Figure 11(a)) and 3D models (Figure 11(b)) using 3D marching cube [6] are exported using Mimic's CTM to our software. The new tessellation algorithm is then tested to generate the smooth shaded pelvic model shown in 11(c).

Fig. 10. A smooth rendered 3D prostate model using the new tessellation algorithm with 3 pixel error built from 19 slices of ultrasound images with a slice-to-slice thickness of about 3mm. 1118 facets are generated, with 78% of them in 20 triangle strips. The model takes 1-3 seconds to generate from the 2D contours. Arrows indicate the undesired artifacts. (Note: This model has only single branch for connection of contours.)

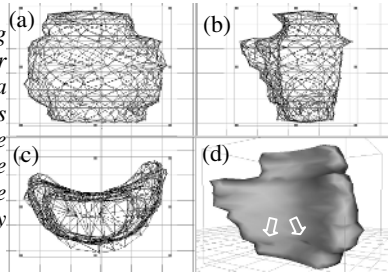
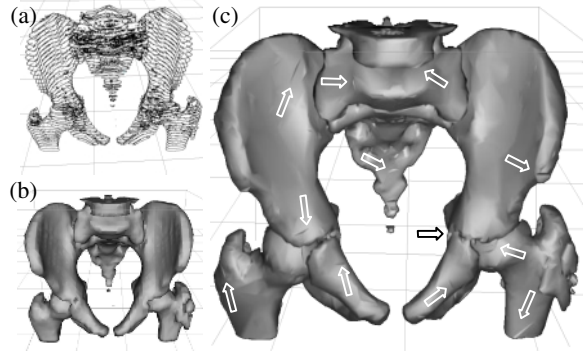


Fig. 11. (a) 38 cross-sections of a pelvis with slice-to-slice thickness 6mm. (b) Model reconstructed using marching cube with 25572 facets. Approximately 16 seconds to generate in Mimics. (c) The model is generated using the new tessellation algorithm with 3-pixels error. 11329 facets with 67% of them in 194 triangle strips. Approximately 9 seconds to generate model from contours. Arrows indicate undesired artefacts.



³ Microsoft Visual C++ 5.0 and WindowNT 4.0 are trademark of Microsoft Corporation.

⁴ TDZ2000 is a trademark of Intergraph Corporation.

5 Limitations and Contributions

In the prostate model (Figure 10(d)) and pelvic model (Figure 11(d)), there are several deep “cut” artefacts that are denoted by a narrow and long shadow groove (indicated by block arrows). These are resulted by the triangulation strips used. We identified that the triangulation strips path used here are similar to the triangulation path as shown in the Figure 6(a). The middle Transvertex have located towards the left-hand side causing obtuse triangle to form. The “cuts” become longer if the upper contour line segment also has obtuse triangles. Obtuse triangles are caused during 2D contour tessellation. Different contours have different starting points that are dependent on the sequence of triangle strips. In some cases, this caused the last point to be located near the starting point.

The new tessellation algorithm generated the pelvic models with 28% less facets than marching cube model. The total running time for generating the pelvic model, using the new method without including segmentation time, has shown a significant improvement of only 9 seconds are needed to complete the whole task, while the marching cube model in Mimics uses 16 seconds. However, the running time may not indicate the performance of algorithm. E.g. the Mimic software may not be using OpenGL (graphical hardware supported). In terms of quality, our model shows much more topological surface details (visually examine) than the marching cube model. In terms of rendering speed, it is much faster than the marching cube model (all models are rendered on our software). This is because the graphics hardware handles 194 triangle strips instead of every facet. This results in the improvements in generating and rendering speed, and reduction in data handling.

6 Conclusion and Future Directions

The new tessellation algorithm automatically determines the topological type of the surface, and the presence and location of the sharp features. In addition, the tessellated line segment defined between adjacent contour determines the choice of triangulation that can significantly impact the cost of the resulting strips. The gradient check also plays an important part when exercising a complete “walk” of triangle strips around a chosen contour. The algorithm is tested using ultrasound images of a prostate and CT images of a pelvic bone. The constructed pelvic model surface quality and data handling performance is compared with Lorensen’s marching cube method [6]. It can be seen that the new algorithm is efficient for creating strips for polygon models that are supported by computer graphics workstations. This can be attributed largely to the first step of implementation of contour tessellation and the optimal use of triangle strips that can significantly reduce rendering time [20]. There are a number of areas for future research that include:

1. Eliminate obtuse triangulation; possible use of 3D delaunay triangulation or reversing of contour tessellation for the last few points.
2. Experiment with different sparse data. E.g. branching nerve or blood vessel.
3. Generate a graph consisting of running time (including rendering time) versus number of triangles used and the number of triangle strips used.
4. Determine the important of model’s pixel error criterion with clinical application.

References

- [1] C. F. Chan, C. G. Zhu, C. K. Kwok, M. Y. Teo, W. S. Ng, Y. C. Wong, C. Cheng and K. T. Foo. Prostate Biopsy using Augmented Reality 3D-Visualisation. *18th Southern Biomedical Engineering Conference and 2nd International Conference on Ethical Issues in Biomedical Engineering, Clemson, U.S.A., May 21-23, 1999*. To appear.
- [2] Y. J. Liu, W. S. Ng, M. Y. Teo and H. C. Lim. Computerised prostate boundary estimation of ultrasound images using radial bas relief method. *Medical Biology Engineering and Computing*, 35:445-454, 1997.
- [3] R. Y. Wu, K. V. Ling, and W. S. Ng. Prostate Edge Strength Enhancement In Ultrasound Images Using New Radial Bas-Relief Method. *Technical report. Nanyang Technology University*, 1999.
- [4] Y. Zhu and L. D. Seneviratne. Optimal polygonal approximation of digitised curves. *IEE Proceedings of Vision Image Signal Process*, 144(1):8-14, 1997.
- [5] D. F. Watson. Contouring: A Guide to the Analysis and Display of Spatial Data. *Pergamon Press*, 1992.
- [6] W. Lorensen and H. Cline. Marching cubes: a high-resolution 3D surface construction algorithm. *Computer Graphics*, 21:163-169, 1987.
- [7] G. Giraudon. An efficient edge chaining algorithm. *Scandinavian Conference on Image Analysis, Stockholm*, 5(2):163-169, 1987.
- [8] E. Keppel. Approximating complex surfaces by triangulation of contour lines. *IBM Journal Resolution Development*. 19(1):2-11, 1975.
- [9] H. Fuchs, Z. M. Kedem and S. P. Uselton. Optimal surface reconstruction from planar contours. *Association for Computing Machinery*, 20(10):693-702, 1977.
- [10] H. Christiansen and T. Sederberg. Conversion of complex contour line definitions into polygonal element mosaics. *Computer Graphics*, 12:187-192, 1978.
- [11] A. B. Ekoule, F. C. Peyrin and C. L. Odet. A triangulation algorithm from arbitrary shaped multiple planar contours. *ACM Transition Graphics*, 10(2):182-199, 1991.
- [12] C. L. Bajaj, E. J. Coyle and K. N. Lin. Arbitrary Topology Shape Reconstruction from Planar Cross Sections. *Graphical Models and Image Processing*, 58:310-319, 1996.
- [13] D. Meyers, S. Skinner and K. Sloan. Surfaces from contours. *Association for Computing Machinery Transition Graphics*, 11(3):228-258, 1992.
- [14] B. I. Soroka. Generalised cones from serial sections. *Computer Vision Graphics and image Processing*, 15:154-166, 1981.
- [15] J. D. Boissonnat. Shape reconstruction from planar cross sections. *Computing Vision Graphics Image Proceeding*. 44(1):1-29, 1988.
- [16] W. C. Lin and S. Y. Chen. A new surface interpolation technique for reconstructing 3D objects from serial cross-sections. *Computer Vision Graphics and image Processing*, 48:124-143, 1989.
- [17] J. V. Miller, D. E. Breen, W. E. Lorensen, R. M. O'Bara and M. J. Wozn. Geometrically deformed models: A method for extracting closed geometric models from volume data. *Computer Graphics, SIGGRAPH '91*, 24(4):217-226, 1991.
- [18] OpenGL Architecture Review Board, OpenGL Reference Manual. *Addison-Wesley, Reading, MA*, 1993.
- [19] K. Akeley, P. Haeblerli and D. Burns. *C program on SGI developer's toolbox CD*, 1990.
- [20] F. Evans, S. Skiena and A. Varshney. Optimizing Triangle Strips for Fast Rendering. *In Proceedings of Visualization '96*, 319-326, 1996.