Radek Sviták[1], **Václav Skala**
University of West Bohemia; Department of Computer Science and Engineering;
Faculty of Applied Sciences; Univerzitní 8, 306 14 Plzeň
{rsvitak|skala}@kiv.zcu.cz

SURFACE RECONSTRUCTION FROM ORTHOGONAL SLICES

Abstract.
This paper is concerned with the problem of reconstructing surfaces of 3D objects, given sets of planar parallel slices representing cross sections through the objects. We present a new approach, which is based on considering more than one mutually non-parallel sets of slices. This paper is an introduction to this problem and includes our proposal of solution for orthogonal sets of slices. The properties and sample results are discussed.

keywords : *surface reconstruction, orthogonal slices, polygonal contours, visualization.*

1 INTRODUCTION

The problem of reconstructing a three dimensional surface from a set of planar contours is an important problem in many fields. For example in clinical medicine, the data generated by various imaging techniques such as computed axial tomography (CAT), ultrasound, and nuclear magnetic resonance (NMR) provide a series of slices through the object of study. Biologists try to understand the shape of microscopic objects from serial sections through the object. In Computed Aided Design (CAD), lofting techniques specify the geometry of an object by means of series of contours.

One set of planar slices, where each slice contains several contours, is usually used as an input. It is possible to say that there are three basic approaches to the problem: volume based, surface based and implicit solid modeling. Volume based approaches [7], [9], [11], [18] require data available as a 3-D grid. Surface based approaches [1] - [6], [8], [14] - [16] assume the data define the intersection of a surface and a plane of sectioning. Quite a new approach is reconstruction using implicit modeling [10], [12], [19].

Which approach is the most applicable depends on the nature of the data. When the available data are a dense 3-D lattice of values, a volume based approach such as the marching cubes algorithm of Lorensen and Cline [13] or the geometrically deformed models of Miller et al. [17] may be the best. If the available data are a set of closed contours

[1] This work was supported by the Ministry of Education of the Czech Republic – project MSM 235200005

denoting the surfaces of the objects to be reconstructed then a surface-based or implicit modeling approach may be preferred. If the spacing between slices is comparable to the resolution within the plane of the contours, a volume-based approach can be used.

A large set separation between the planes of adjacent sections causes problems for volume-based approaches, since their solution to the problem of the determining the adjacency topology of contours in the data set (i.e., which contours should be connected by a surface - the correspondence problem) rely on overlap of projected contours. If widely spaced sections slice obliquely through an object then the reconstruction is quite difficult and results of these methods have shortcomings, such as incorrect contour correspondence. If it is possible to use more than one set of planar slices (i.e., if the original object was scanned in more than one plane) then the reconstruction is easier and the quality of its results increases.

We found the existing methods have problems solving contour correspondence. The aim of our work is to prove that the considering more information given as mutually non-parallel sets of slices can overcome these shortcomings. Our effort is to propose a new method handling such kinds of input. We have restricted to orthogonal sets of parallel slices (i.e. 3 sets of slices parallel to *xy*, *yz*, *zx* in 3D) at this moment.

2 ORTHOGONAL SETS OF SLICES

If we slice an object by more then one set of parallel slices and especially when these sets are mutually orthogonal, we get orthogonal sets of parallel planar slices. See fig. 1 for illustration of the problem or [20] for more detailed definition. We suppose polygonal contours in each slice.

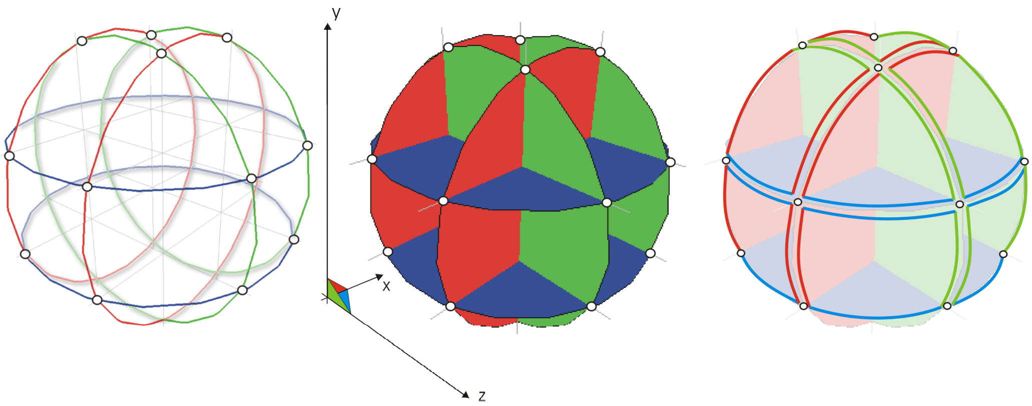


Fig. 1: Orthogonal sets of slices. In the middle contours of one set are filled by a unique color. On the right there is an example of obtained spatial polygons to be patched.

2.1 Principle of the algorithm for the surface reconstruction from orthogonal slices

The motivation for our algorithm was the study of the situations similar to the one on fig. 1. The planes of slices divide space into a set of spatial cells. We distinguish two kinds of cells, the active and the empty cells. On some sides of an active cell are located parts of contours, which means that the resultant surface intersects the cell, see fig. 2.

The intersection of two orthogonal slices is a set of points. Such points we call node points, see fig. 2. Now we focus on an active cell. The important discovery is that parts of input contours and the node points form spatial polygons placed on the surface, see fig. 2.

We take a graph view on the contours. Each contour is a subgraph (or component) of graph G. Graph G is directed, its set of vertices V_G is formed by contour vertices, the set of edges E_G is formed by contour edges. Note that as long as we have one set of parallel slices, each contour represents one component of the graph and for each vertex v_i is $d^-(v_i) = d^+(v_i) = 1$, where d^- is the input degree and d^+ is the output degree of a vertex.

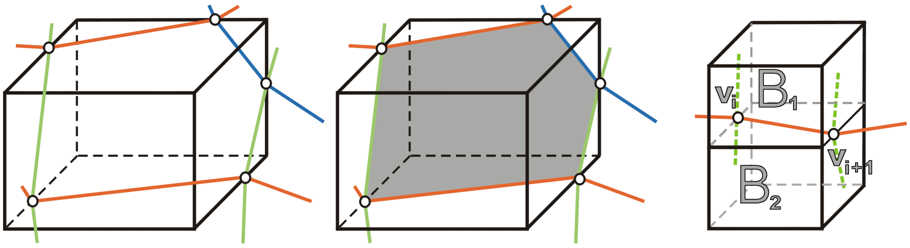


Figure 2: Active cells. Parts of contours on sides of the cell together with node points form spatial polygons to be patched. Node points are denoted as white circles.

If we add a node point (node vertex) to graph G then the number of components decreases. Node vertex represents a connection of two components. Our aim is to show that if graph G is connected then we are able to get surface with the same structure of the sampled part of the input object.

Our algorithm consists of 4 steps. In the first step we compute node vertices of each contour, in the second step we find all pairs of corresponding node vertices (each of the two points is a part of a different contour, which are orthogonal). These pairs are transformed to valid node vertices of graph G. In the third step we find (using a set of criteria) those parts of graph G (circles) that form spatial polygons lying on the object surface. In the last step we patch the polygons.

2.2 Node points computation

In this step node vertices of each contour are computed and added among current contour vertices to the right position. Our algorithm works on the same principle as the Cohen-Sutherland's line clipping algorithm.

An intersection of a slice plane and all other orthogonal slice planes form a lattice with cells, see fig. 3. Each node point arises as the intersection of a contour and a side of a cell. Since the contour is supposed to be polygonal, a node point is computed as an intersection of two lines. Singular cases when a contour crosses a cell at its corner are handled separately.

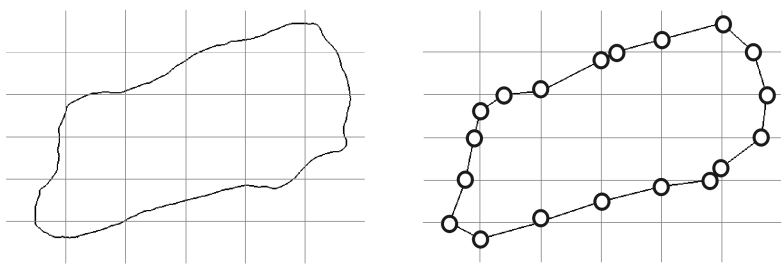


Figure 3. On the left there is an input contour, the lattice represents positions of orthogonal slices planes. On the right there is the contour formed by its node points (white circles).

The algorithm starts at an arbitrary vertex v_k of a contour and cell B_{ij} containing this vertex is determined. Then it marches along the contour until it gets over the border of cell B_{ij}. At this moment a new node point is computed and then added to the contour as a new vertex. The algorithm continues in this manner until it reaches the last vertex of the contour. The time complexity is linear $O(n+m)$, where n is the number of vertices of the contour and m is the number of intersections of the contour and the lattice, i.e. number of node vertices.

2.3 Searching for node vertices correspondence

As we said before a node vertex is an intersection of two mutually orthogonal contours. However, there is a problem that we have computed all node vertices, but we do not know the correspondence between them, since each vertex of a pair of corresponding vertices belongs to a different contour. Each pair of corresponding node vertices should be replaced with a single vertex. As soon as we connect both crossed contours through this vertex, its degree is $d^- = d^+ = 2$.

A simple algorithm searching for the pairs of corresponding node vertices works with quadratic complexity. For each node vertex w_i is the corresponding slice s determined. Then all node vertices w_j of s are searched until the pair (w_i,w_j) is found. We can use the space subdivision defined by the spatial cells to replace searching over all node vertices of s with searching among a set of vertices adjacent with one edge of a cell. It decreases the time complexity to linear.

Now it is necessary to describe an important singular case. As we said above, it is possible that more than two contours cross at one point. This case can appear only when there are 3 mutually orthogonal sets of slices. Each of such contours belongs to a different set. It is obvious that they cross at a node point of the spatial lattice formed by the slice planes.

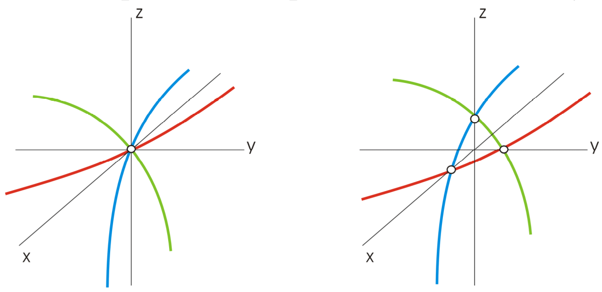


Figure 4. Replacing one intersection of three contours with three intersections of two contours.

An ideal solution would be to accept these cases and to deal with the two kinds of node vertices. Since the occurrence of such singular cases is not frequent, we handle them by moving the critical parts of the contours slightly. Thus we replace one intersection of three orthogonal contours by three intersections of two orthogonal contours, see fig. 4.

2.4 Searching and patching spatial polygons

In this step we assume graph G with correctly connected node vertices. From now on we define that one edge of graph G represents a part of a contour between two node vertices, see fig 3. The geometrical shape of the edges still corresponds to the real part of contours. The task is to find those spatial polygons that lie on the surface. From the graph view, we can regard these polygons as circles of the graph.

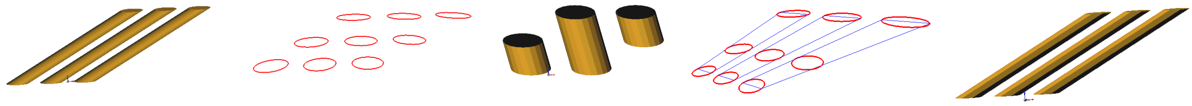
We suppose each edge e is adjacent with cells B_1 and B_2, see fig. 2, on the right. Each cell from $\{B_1, B_2\}$ includes one circle c, which is adjacent with e. The circle c represents the spatial polygon being searched.

In the first step the algorithm finds adjacent cells B_1 and B_2 for each e. The following steps are then applied to each cell b from $\{B_1, B_2\}$. If e has not been used in b yet, we can start to build polygon *p*. All the edges that can be appended to *p* must be adjacent with *b* and must not have been used yet.

Since we have obtained all polygons, we can start to patch them. We can use any arbitrary patching technique. Our algorithm linearly approximates each edge of *p*.

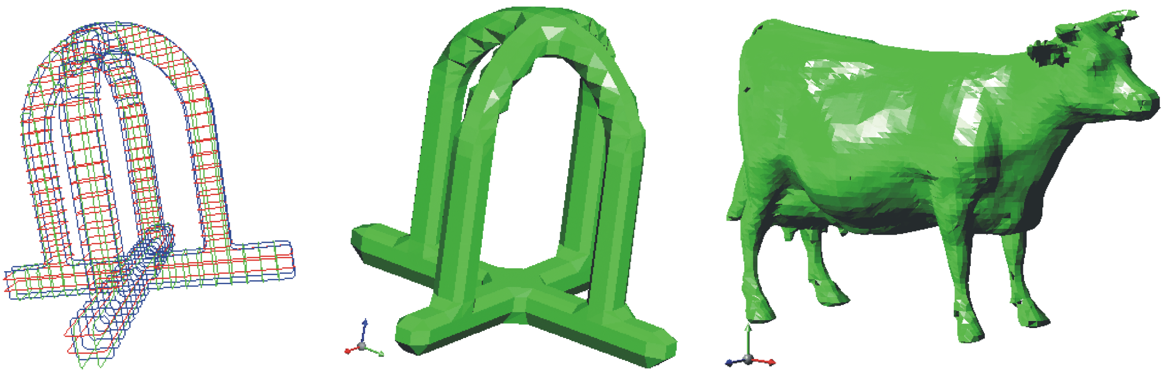
3 RESULTS

First we have created slices of a scene containing oblique pipes (their axes are significantly leaned from *z*-axis); see fig. 6a. Then we have used a usual algorithm reconstructing the surface from parallel slices. The part of the algorithm responsible for determining the correspondence of contours is based on contour overlapping. The correspondence has been determined incorrectly and so the resulting surface does not resemble the original; see fig 6c. Our algorithm processing orthogonal sets of slices (fig. 6d) has produced surface with the expected structure, fig. 6e.



a) b) c) d) e)

Figure 6a-e. a) the initial scene, b) set of parallel slices, c) surface reconstruction from parallel set of slices (contour overlapping correspondence), d) orthogonal slices, e) surface reconstruction from orthogonal sets of slices



a) b) c)

Figure 7a-c. Orthogonal slices (a) as a source of reconstruction (b), c) another experimental result

In order to prove the capabilities of proposed algorithm we have verified it for more complicated data sets as well; see fig. 7a-c. The artifacts on the surface such as holes or overlaying triangles are caused by incorrect graph *G*, which, in fact, is the most problematic part of the algorithm. Linear approximation of the graph edges causes loss of details of the original surface.

4 CONCLUSION

We have presented a new approach for surface reconstruction using orthogonal sets of slices. The experiments have proven that the presented algorithm reconstructs typical and also more complex data sets while more applicable results than the other methods. In further research we would like to remove incorrectness in the graph construction and to propose a robust volume-based algorithm.

REFERENCES

[1] Bajaj, C. L., Coyle, E., Lin, K. N. *Arbitrary topology shape reconstruction from planar cross sections*. Graphical Models and Image Processing 58 (1996) 524-543.

[2] Bajaj, C. L., Coyle, E., Lin, K. N. *Surface and 3D Triangular Meshes from Planar Cross Sections*, 5th International Meshing Roundtable, Sandia National Laboratories, pp.169-178, October 1996.

[3] Bajaj C., Coyle E., Lin K. N. *Tetrahedral Meshes from Planar Cross Sections*, Computer Methods in Applied Mechanics and Engineering, Vol. 179 (1999) 31-52.

[4] Barequet, G., Sharir, M. *Piecewise-linear interpolation between polygonal slices*, Proceedings of the 10th Annual Symposium on Computational Geometry, pp. 93-102, 1994.

[5] Barequet, G., Shapiro, D., Tal. A. *History consideration in reconstructing polyhedral surfaces from parallel slices*. In Proceedings of Visualization, pages 149--156, October 1996.

[6] Cheng, S. W., Dey, T. K. *Improved constructions of Delaunay based contour surfaces*. Manuscript 1999.

[7] Cong, G., Parvin, B. *Robust and efficient surface reconstruction from contours*, The Visual Computer (2001) 17:199-208, Springer-Verlag 2001.

[8] Ekoule, A. B., Peyrin, F. C., Odet, C. L. *A Triangulation Algorithm from Arbitrary Shaped Multiple Planar Contours*, ACM Trans. on Graph., Vol.10, No.2, pp.182-199, 1991.

[9] Floater, M. S., Westgaard, G. *Smooth surface reconstruction from cross sections using implicit methods*. SINTEF report, Oslo, 1995.

[10] Hirsch, B.E., Thoben, K. D., Sheng, X., Joppe, M. *Reconstruction of volume boundary from layers by triangulation*, In Journal of Manufacturing - Special Issue on Layered Manufacturing, Volume 16, Number 4, pp. 249-259, Society of Manufacturing Engineers (SME), Dearborn, Michigan, August 1997.

[11] Klein, R., Schilling, A. G. *Fast Distance Field Interpolation for Reconstruction of Surfaces from Contours*, in: Eurographics '99, Short Papers & Demos proceedings, 1999.

[12] Lim, C.T., Ensz, M.T., Ganter, M.A., Storti, D.W. *Object reconstruction from layered data using implicit solid modeling*. Accepted by Journal of Manufacturing Systems, Special Issue on Layered Manufacturing, 1997.

[13] Lorensen, W. E., Cline, H. E. Marching Cubes, *A high resolution 3D surface reconstruction algorithm*. Computer Graphics 21, 4 (July 1987), 163-169.

[14] Meyers, D. *Multiresolution Tiling*. Computer Graphics Forum 13, 5 (December 1994), 325-340.

[15] Meyers, D. *Reconstruction of Surfaces From Planar Contours*, disertation, University of Washington, 1994.

[16] Meyers, D., Skinner, S., Sloan, K. *Surfaces from Contours*. ACM Trans. on Graphics, Vol. 11 (3), 1992, pp.228-258.

[17] Miller, J. V., Breen, D. E., Lorensen, W. E., O'Bara, R. M., Wozny, M. J. *Geometrically deformed models: A method for extracting closed geometric models from volume data*. Computer Graphics 25, 4 (1991), 217-226.

[18] Schilling, A., Klein, R. *Fast Generation of Multiresolution Surfaces from Contours*, proceedings of The 9th Eurographics Workshop on Visualization in Scientific Computing, April 1998, pages 88-99.

[19] Savchenko, V., Pasko, A. *Reconstruction from contour data and sculpting 3D objects*, Journal of Computer Aided Surgery, Vol. 1, Suppl., Proceedings of Second International Symposium on Computer Aided Surgery ISCAS'95, 1995, pp.56-57.

[20] Sviták, R., (supervisr Skala, V). *Surface reconstruction from slices*. In Czech. MSc. Thesis. Department of Computer Science and Engineering, West Bohemia University, Plzeň, 2001.