A New Approach for Surface Reconstruction Using Slices

Shamima Yasmin and Abdullah Zawawi Talib

School of Computer Sciences, Universiti Sains Malaysia,
11800 USM Penang, Malaysia
{shamima, azht}@cs.usm.my

Abstract. This paper describes a novel algorithm for surface reconstruction from slices. A number of slices are extracted from a given data oriented along any of the principal axes. Slices are projected onto the XZ plane and equal number of traversals takes place for each slice by a cut plane oriented along the X axis. As the cut plane traverses along each slice, cut points are extracted. To establish correspondence between two consecutive slices, firstly domain mapping takes place. Then a heuristic approach is taken which is based on the comparison of the number of occurrences of particular cut points between slices. Optimization is performed on the basis of minimal differences of the number of occurrences of particular cut points between consecutive slices. Although heuristic approach is not flawless, this algorithm is able to construct surface of fairly complex objects. The algorithm is dynamic enough as the number of slices and the number of traversals can be adjusted depending on the complexity of the object.

Keywords: Contours, Surface Reconstruction, Boundary.

1 Introduction

Contours are usually a sparse representation of an object. Surface reconstruction from slices is performed by taking two consecutive slices into consideration. Surface reconstruction from slices has its application in medical science for surface recovery, for the reconstruction of joint in human body, for volume calculation of human organs etc. In geological field, surface reconstruction is useful to develop terrain model of a particular area from a number of contours. Surface reconstruction has its application in industry for developing synthetic models. Many surface reconstruction algorithms have been developed which construct surface from slices [1-7]. Our proposed algorithm was originally developed to construct surface of morphed object [8]. Later it was tested separately for surface reconstruction of a number of objects and found quite sound for constructing surface of fairly complex objects.

2 Related Work

Three dimensional surface reconstruction methods can be classified into the following two categories: (a) Volume-based Surface Reconstruction and (b) Surface Reconstruction

from Contours. Volume-based surface reconstruction methods can again be subdivided into three categories: (i) Image Processing Technique, (ii) Distance Field Interpolation (DFI) Method and (iii) Marching Cube Method.

In image processing technique, the whole volume is decomposed into a number of cross-section along a given direction. Slice interpolation or slice replication is performed and these interpolated/ replicated slices are inserted in between the extracted slices [9]. Distance Field Interpolation (DFI) method computes the distance of each voxel within the volume to the surface of the object and interpolates the distance field of the consecutive contours [10]. To take into account the warping of the object, warp guided distance field interpolation is also used [11]. In marching cube method [12], voxel structures within a volume helps to establish the connectivity between the adjacent voxels. Adjacent voxels are connected by a number of triangles which form the constructed surface.

Surface-based reconstruction from contours consists of mapping between adjacent contours. This method consists of the following major pre-processing steps: (i) Discretization of the vertices along the contours, (ii) Finding matched portion of the adjacent contours and (iii) Processing separately of the unmatched portions. Once pre-processing is done, polygonal triangulation is performed. Barequet et al. [5] use some heuristics such as calculation of minimum area, minimum area², (minimum area² + Length), maximal minimum angle to find the optimum output between the adjacent contour vertices. Sometimes two adjacent contours are converted into a directed toroidal graph whose minimum cost cycle determines the optimal area [2]. Bajaj et al. [3] impose a set of constraints to construct the optimal tiling vertex table between two adjacent contours. To ensure smooth surface, partial differential equation is applied across the contours [6]. Keppel et al. [1] derive optimal triangulated surface for two convex contours from the maximum volume of the polyhedron between them and vice versa for concave contours. Turk et al. [4] construct surface by first converting adjacent slices to two separate implicit functions. Then surface reconstruction takes place between these two functions. Wang et al. [7] at first generate unconstrained delaunay triangulation of all vertices followed by constrained recovery of the boundary edges. Constrained delaunay triangulation is performed for mapping along the surface.

3 Algorithm Overview

Our algorithm is a surface-based reconstruction method. Each contour is first discretized into a number of contour points. Firstly domain mapping between two adjacent contours takes place on the basis of contour overlay. Then on the basis of the number of occurrences of particular cut points, point clouds in each contour are demarcated. Two adjacent contours may differ in the number of cut points. Mapping takes place on the basis of the optimum differences between the number of occurrences of particular cut points in corresponding regions of adjacent slices. After the necessary mapping, rectangular cells are constructed from two pair of points with each pair from each of the adjacent contours.

The algorithm consists of the following two main steps: (a) Contour Pre-processing and (b) Surface Reconstruction. Contour pre-processing step is further subdivided into

three steps: (i) Contour Projection, (ii) Contour Traversal and Extraction of Contour Points and (iii) Orientation and Translation of Contours. Surface reconstruction is also subdivided into three steps: (i) Separation of Disconnected Regions, (ii) Establishing Correspondence between Adjacent Contour Points and Cell Construction and (iii) Consideration of Critical Points.

4 Contour Pre-processing

The detail of this step is as follows:

(i) Contour Projection. All contours are projected onto the XZ plane and centered at the origin.

(ii) Contour Traversal and Extraction of Contour Points. Each contour is traversed along the direction of their minimum X ($X_{\min}$) to maximum X ($X_{\max}$) with a traversal plane defined as (1, 0, 0). Equal number of traversals is performed for each contour. Traversal spacing for each contour is determined as follows:

$$\text{Spacing} = (X_{\max} - X_{\min}) / \text{Number of Traversals};$$

Boundary points are extracted from the traversals. If the number of extracted points from any cut plane happens to be odd, it is made even. As the cut plane traverses, regions are demarcated. Region is an area where the number of points extracted by the cut plane is the same while traversing along the X axis. In Figure 1(a), the contour consists of 3 regions i.e. two 2-point region and one 4-point region.

(iii) Orientation and Translation of Contours. Each contour defined by a point cloud already projected onto the XZ plane is now oriented along the normal of each contour and translated back to the center of that contour (Figure 1 (b)).

Fig. 1.(a) Region Separation and (b) Orientation and Translation of All Contour Point Clouds

5 Surface Reconstruction

From the stack of oriented and translated point cloud, surface is constructed. Surface reconstruction is performed by only considering each of the two consecutive point clouds. This simplifies the overall surface reconstruction process as where data is highly irregular, necessary modification among cell coordinates is limited to only two consecutive boundaries.

5.1 Separation of Disconnected Regions

Each consecutive boundary may have regions which are disconnected from one another (Figure 2(a)). Nearest neighbor searching is carried out to find out this kind of regions. The disconnected regions are laterally mapped (for better effect) and the other regions are to be mapped across the boundary (vertical mapping), (Figure 2(b)). The detail of vertical mapping is discussed next.

Fig 2. Mapping Separated Regions : (a) Point Clouds with One Region (Top) and Two Regions (Bottom), (b) Vertical Mapping (Left) and Lateral Mapping (Right)

5.2 Establishing Correspondence between Adjacent Contour Points and Cell Construction

After region separation, two consecutive point/ cell arrays (representing two consecutive slices) are obtained. Empty region is obtained when the element of the number array (representing number of cut points) is '0'. If the region is empty in one of the arrays while the corresponding element of the other array is not empty (not '0'), then the value '-1' is inserted in the corresponding portion of the other array. By doing so, we obtain equal number of '0' and '-1' regions for both arrays and we extract equal number of sub arrays from these two consecutive number arrays. Now the following operation is carried out on each corresponding extracted sub arrays.

The two arrays which contain the number of interpolated points at each index need to be compressed so that the process of mapping can be carried out in an easier and straightforward manner. Figure 3(a) shows the process of compressing two consecutive 'Region Number' arrays (where region numbers are stored) and 'Number of Occurrences' arrays (where the number of occurrences of each region number are stored). Firstly, the size of both arrays should be made equal. This is discussed in detail next.

Each index value of the number of occurrences of the first sub array is compared against the corresponding number of occurrences in the second sub array. To meet the optimum situation, one index value in one of the number of occurrences array may correspond to more than one index values of the other number of occurrences array. Both region number sub arrays and number of occurrences sub arrays are again divided into a number of smaller sub arrays on the basis of corresponding optimum index matching in the number of occurrences arrays. As shown in Figure 3(a), index '0' in the first region number array and the first number of occurrences array correspond to index '0' to index '2' in the second region number array and the second number of occurrences array respectively.

index values to the corresponding region number values and the values of the region number of the second portion are the remaining region numbers resulting from the split. In the example (Figure 3(d)), the first discrepancy occurs at index '1' and the nearest matched values are at index '0' and index '2' with a value of '8' and '8'. The current value i.e. 8 in the first array needs to be split into two portions. The first portion is made equal to '6' and the second portion is assigned the remaining value ($8-6=2$). At the end of the entire processing, two sets of region number arrays are obtained. The top set (Figure 3(d)) now consists of equal region number and can therefore be vertically mapped whereas the bottom set (Figure 3(e)) is to be laterally mapped separately.

5.3 Consideration of Critical Points

5.3.1 Empty Space Consideration

When there is an empty space in both corresponding cells, spaces are separated by direct vertical mapping in between them (Figure 4(a)). On the other hand, when solid cells in one boundary meet empty space in another boundary during vertical mapping, corresponding imaginary empty space is calculated for the solid portion. Now empty spaces are separated by direct vertical mapping and lateral cell construction takes place for the imaginary empty portion (Figure 4(b)).

Fig. 4. Empty Space Consideration during Surface Reconstruction: (a) Reconstructed Surface from Slice No. 15 and Slice No. 16 and (b) Reconstructed Surface from Slice No. 13 and Slice No. 14

5.3.2 Transition between Different-Numbered Regions

Discontinuity occurs at the transition point between two different-numbered regions. To ensure smooth transition between the regions, the last point array of the first region and the first point array of the second region have to be scanned so that the portions which are continuous and the portions which are not continuous are identified (Figure 5).

5.3.3 Distinguishing Connected/ Disconnected Regions across the Direction of Traversal of Slices

When region number values in both corresponding region number arrays are equal and region number value is greater than two, and there exists an empty region next to

Fig. 5. Surface Reconstruction while Transiting between Different-numbered Regions

it, there should also exist empty space across the section. However there may be ambiguity on the number of empty spaces across the section. Figure 6 shows two different cases of two adjacent point clouds where the region number in the point clouds is '4'. In Figure 6(a), there are two empty spaces across the section whereas in Figure 6(b), there is one empty space across the section.

Fig. 6. Surface Reconstruction Considering the Number of Empty Spaces across the Slice: (a) Two Empty Spaces and (b) One Empty Space

In addition to these cases, it may happen that only one of the two consecutive slices has empty spaces. In this case the same method as mentioned above is applied and in addition, lateral mapping takes place for the non-empty portion.

6 Implementation and Results

The algorithm is implemented using C++ with Visualization Tool Kit (VTK) as graphics platform. WindowsXP is used as the operating system with the following hardware configurations: Pentium 4-M CPU 2.40GHZ, 1 GB of RAM. In Figure 7(a) and Figure 7(b), surfaces are reconstructed from 129 slices with the number of traversals 201 with a runtime of 94 seconds and 161 seconds respectively. In Figure 7(c) and Figure 7(d) two parametric surfaces i.e. “crosscap” and “boy” are constructed from 129 slices with the number of traversals of 201 and with the runtime of 84 seconds and 88 seconds respectively. In Figure 8(a) and Figure 8(b) two surfaces i.e. “Heart” and “Human head” are reconstructed from 129 slices with the number of traversals 201 and 401 respectively and a runtime of 97 seconds and 200 seconds respectively.

Fig. 7. Reconstruction of Surface from Slices Arranged along the Y-axis

Fig. 8. Reconstruction of Two Surfaces from Slices Arranged along (a) the Z-axis and (b) the Y-axis respectively

Figure 9 shows how the output as well as runtime vary as the number of slices and the number of traversals vary. Figure 9(a)(i) shows the output for a parametric surface “Figure-8 Klein” with a runtime of 10 seconds with the number of slices 33 and the number of traversals 101. In Figure 9(a)(ii), the same parametric surface is reconstructed from 65 slices with the number of traversals 201 and a runtime of 39 seconds whereas in Figure 9(a)(iii) runtime is 166 seconds when the number of slices and the

number of traversals are 129 and 401 respectively. Runtime increases as the number of traversals as well as the number of slices increase as shown in the Figure 9(b).

As it is discussed in Section 5.3.3, slices may also need to be traversed along Z-axis in order to find out the number of empty spaces along that direction when there is ambiguity about the number of empty spaces along that direction. In Figure 10, a surface called “schwarz” is reconstructed from 129 slices with the number of traversals 201

Fig. 9. Variation in (a) the Output and (b) Runtime with the Variation in the Number of Traversals and the Number of Slices

Fig. 10. Reconstruction of a Surface from Slices Arranged along the Z-axis

Table 1. Variation in the Number of Cells and Runtime with Variation in the Number of Slices and the Number of Traversals

Name of Object	Number of Slices	Number of Traversal along Each Slice	Direction of Arrangement of Slices	Number of Cells	Runtime (in seconds)
(a) Human Head	129	401	Y-axis	124739	200
(b) Heart	129	201	Z-axis	71805	97
(c) Cow	129	201	Y-axis	42928	161
(d) Teapot	129	201	Y-axis	49044	94
(e) Parametric Surface (crosscap)	129	201	Y-axis	52000	84
(f) Parametric Surface (boy)	129	201	Y-axis	62170	88
(g) Schwarz	129	201	Z-axis	19859	340
(h) Parametric Surface (Figure-8 Klein)	33	101	Y-axis	7108	10
	65	201	Y-axis	28238	39
	129	401	Y-axis	112547	166

and a runtime of 340 seconds. There is an increase in the runtime because of this excess of traversing along the Z-axis. Table 1 summarizes the variation in the runtime and the number of cells with the variation in the number of traversals and the number of slices for the different objects mentioned above.

Let us analyze the algorithm in terms of efficiency. Let us suppose that 'n' is the number of traversals along the X-axis, 'm' is the number of traversals along the Z-axis, 'p' is the tuple number of 'Region Number Array' in two consecutive slices. During contour point extraction, each slice is traversed along the X-axis. During tuple equalization, comparison of each index value of 'Region Number Array' takes place. During vertical mapping, rightward and leftward traversals take place for each slice which in worst case can be equal to the tuple number in the 'Region Number Array'. Again traversal along the Z-axis can take place for each slice during vertical mapping. Considering the number of constructed cells for each of the two consecutive slices is equal to the number of traversal along the X-axis i.e. 'n', the complexity per slice is $\Theta((n + p + pm) + n)$. Here 'p' signifies the tuple number of the 'Region Number Array' and its value is usually much smaller than 'n' and 'm'. Hence the complexity is reduced to $\approx \Theta(n + pm)$.

7 Conclusion and Future Work

The proposed algorithm shows a simple method of constructing surface from slices. Some compromises have been made in developing this algorithm as it is already mentioned that this algorithm was originally developed to reconstruct surface for morphed objects where accuracy is not so stringent. Still this method works well as a stand-alone algorithm for reconstruction of surfaces ranging from simple to fairly complex objects. No triangulation is performed in this method. Depending on the complexity of the object, the number of slices and the number of traversals can be adjusted and the runtime can also be varied. As surface reconstruction is limited between two adjacent slices, discontinuities may be noticeable near the edges. This method also works well when slices are not parallel. Future works involve strengthening the algorithm so that it works well as a full fledged standalone algorithm for surface reconstruction.

References

1. Keppel, E.: Approximating Complex Surface by Triangulation of Contour Lines. IBM Journal of Research and Development 19, 2–11 (1975)
2. Fuchs, H., Kedem, Z., Uselton, S.: Optimal Surface Reconstruction from Planar Contours. Communications of the ACM 20(10), 693–702 (1977)
3. Bajaj, C., Coyle, E., Lin, K.: Arbitrary Topology Shape Reconstruction from Planar Cross Sections. Graphical Models and Image Processing 58, 524–543 (1996)
4. Turk, G., O'Brien, J.: Shape Transformation Using Variational Implicit Functions. In: Proc. SIGGRAPH, pp. 335–342 (1999)
5. Barequet, G., Shapiro, D., Tal, A.: Multi-level Sensitive Reconstruction of Polyhedral Surfaces from Parallel Slices. The Visual Computer 16(2), 116–133 (2000)
6. Hormann, K., Spinello, S., Schröder, P.: C-1-continuous Terrain Reconstruction from Sparse Contours. In: Proc. Vision, Modeling and Visualization, pp. 289–297 (2003)

7. Wang, D., Hassan, O., Morgan, K., Nigel, W.: Efficient Surface Reconstruction from Contours Based on Two-Dimensional Delaunay Triangulation. *International Journal for Numerical Methods in Engineering* 65, 734–751 (2006)
8. Yasmin, S., Talib, A.Z.: A Method for 3D Morphing Using Slices. In: *Proceedings of International Conference on Computer Graphics Theory and Applications (GRAPP)*, pp. 292–301 (2009)
9. Levoy, M.: Display of Surface from Volume Data. *IEEE Computer Graphics and Applications* 8, 29–37 (1988)
10. Payne, B., Toga, A.: Distance Field Manipulation of Surface Models. *IEEE Computer Graphics and Applications* 12(1), 65–71 (1992)
11. Cohen-Or, D., Levin, D.: Guided Multi-Dimensional Reconstruction from Cross-sections. In: Fontanella, F., Jetter, K., Laurent, P.J. (eds.) *Advanced Topics in Multivariate Approximation*, pp. 1–9. World Scientific Publishing Co., Singapore (1996)
12. Lorensen, W., Cline, H.: Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics* 21, 163–169 (1987)