OPTIMUM SLICE REDUCTION ALGORITHM FOR FAST SURFACE RECONSTRUCTION FROM CONTOUR SLICES

by

TANG SZE LING

Thesis submitted in fulfillment of the requirements for the degree of Master of Science

MAY 2007

ACKNOWLEDGEMENTS

I would like to express my gratefulness to the following imitation I have gained during the period of study at Universiti Sains Malaysia (USM). Special mention of acknowledgements to my beloved parents, Mr Tang Siam Boon and Mdm Lim Wan Hwa, for their never-ending support throughout this study.

I am very grateful to my supervisor, Dr. Bahari Belaton at the School of Computer Sciences, USM. I appreciated his wisdoms on how to develop my study. His guidance, support and inspiration were the crucial to this study. Dr. Bahari was kind enough to read and comment upon early versions until the latest version of this thesis. The next person who comes to my mind is Assoc. Prof. Dr. Abd. Rahni Mt. Piah at the School of Mathematics who recommended me to this research field and also introduced me to Dr. Bahari. I would like to thank Prof. Rahni and Dr. Zainul at the School of Dental Sciences, USM, Kubang Kerian for supplying us with dataset used in experiments.

I must express my great appreciation to School of Computer Sciences and the Institute of Postgraduate Studies (IPS) for releasing the Graduate Assistant Scheme in my graduate study.

In addition, I would like to express my gratitude to all the supporting staff of School of Computer Sciences and IPS. Last but not least, I must thank all my friends who showed concern and gave encouragements during the period of my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ALGORITMA PENGURANGAN HIRISAN YANG OPTIMUM BAGI PEMBINAAN SEMULA PERMUKAAN YANG PANTAS DARIPADA HIRISAN - HIRISAN KONTUR

ABSTRAK

Tesis ini memfokus kepada pembinaan semula permukaan daripada siri hirisan kontur, dengan tujuan mempercepatkan proses pembinaan semula di samping mengekalkan kualiti output pada tahap yang boleh diterima. Teknik yang dicadangkan dalam tesis ini memproses hirisan-hirisan kontur sebelum pembinaan semula permukaan. Fokus kepada input ini boleh dianggap berpatutan berasaskan kepada beberapa perkara. Pertama, modaliti pengimejan moden telah dibina untuk merekod data yang berketepatan tinggi, sesetengah daripada data tersebut berulangan atau tidak menambah nilai secara visual. Kedua, saiz bagi set data sering mempengaruhi masa pemprosesan pembinaan semula, oleh itu jika kita mulakan dengan bilangan hirisan yang kurang tetapi signifikan, ini merupakan satu peluang untuk meningkatkan kelajuan proses pembinaan semula. Ketiga, kebanyakan penyelidikan yang sedia ada hanya tertumpu sama ada pada algoritma pembinaan semula atau pada pemprosesan selepas menghasilkan jaringan segitiga, kaedah yang dicadangkan dalam tesis ini memberi pilihan yang lain untuk mengimbangkan antara kelajuan pemprosesan komputer dan kejituan bagi output.

Kaedah yang dicadangkan ini dibahagikan kepada dua fasa: Pengurangan Hirisan Yang Optimum dan Pembinaan Jaringan Segitiga Secara Blok. Dalam fasa pertama, hirisan-hirisan yang berlebihan akan dikurangkan berdasarkan kepada perbezaan bilangan titik kontur daripada sepasang hirisan yang bersebelahan. Bagi sesuatu set data yang diberikan, satu nilai tahap yang optimum akan dipilih, ini akan digunakan untuk menentukan hirisan-hirisan sama ada untuk dibuang (atau disimpan). Apabila proses ini diselesaikan, outputnya merupakan satu set hirisan-hirisan yang berturutan dengan jarak antara hirisan yang tidak tetap. Ini memerlukan algoritma

pembinaan semula yang khas untuk pembinaan semula permukaan daripada hirisan-hirisan kontur yang tidak tetap, dan masalah ini diselesaikan dalam fasa kedua, iaitu satu pengubahan pembinaan semula permukaan untuk hirisan-hirisan kontur yang tidak tetap. Pengubahan algoritma pembinaan jaringan segitiga ini berdasarkan modul dalam VTK yang dikenali sebagai *vtkDelaunay3D*. Perkara yang penting dalam pengubahan tersebut ialah penghasilan nilai alfa secara automatik untuk "blok-blok" yang berlainan dalam hirisan-hirisan kontur yang tidak tetap. Untuk blok masing-masing, algoritma ini menghasilkan satu nilai alfa yang sesuai, seterusnya melaksanakan pembinaan semula permukaan dengan *vtkDelaunay3D*.

Tesis ini menilaikan kecekapan kaedah tersebut melalui sepuluh (10) set data yang mempunyai pelbagai jenis disiplin, kerumitan and saiz yang berlainan. Bagi setiap set data, tesis ini telah mengumpulkan pengukuran kuantitatif seperti masa pemprosesan komputer untuk pembinaan semula permukaan dan pengukuran kualitatif melalui pembandingan visual antara imej asal dan imej yang dihasilkan daripada hirisan yang dikurangkan. Bagi semua set data, algoritma ini mampu mengurangkan hirisan-hirisan dalam lingkungan antara 40 – 60 % and keputusan ini mempercepatkan masa pemprosesan pembinaan semula sebanyak 60 – 90%. Melalui perbandingan visual, kami menunjukkan secara visual kawasan-kawasan dalam output yang bertindihan atau terpisah (antara yang asal dan yang dikurangkan). Hasil ini adalah setara dengan prinsip-prinsip bagi kaedah yang dicadangkan.

Kami membuat kesimpulan bahawa kaedah yang dicadangkan telah meningkatkan kelajuan pemprosesan komputer bagi proses pembinaan semula dengan kualiti output pada tahap yang boleh diterima, peningkatan utama dibuktikan dalam kes yang set datanya besar dan rumit.

OPTIMUM SLICE REDUCTION ALGORITM FOR FAST SURFACE RECONSTRUCTION FROM CONTOUR SLICES

ABSTRACT

This thesis is concerned with the reconstruction of surface from a series of contour slices, with the aim to speed up the reconstruction process while preserving the output quality at an acceptable level. The proposed technique in this thesis, pre-processes the slices of contour prior to surface reconstruction. This shift of focus to the input seems reasonable for several reasons. First, modern imaging modalities are built to capture high precision data, some of these are redundant or visually insignificant. Second, the dataset's size often determines the computational time of the reconstruction process hence if we provide with lesser but significant number of slices, there is a chance that we can improve the speed of the reconstruction process. Third, majority of the existing works focus either on the reconstruction algorithm itself or on the post-processing of the resulting meshes, the method proposed here provides an alternative way to address the issue between speed of computation and accuracy of output.

Our proposed method is divided into two phases: Optimum Slice Reduction and Block Triangulation. In the first phase, redundant slices are reduced based on the number of contour point's differences for a pair of adjacent slices. For any given dataset, an optimum threshold value is extracted, this value is used to determine the slices to be removed (or retained). Once this process is completed, a set of non-consecutive slices is produced. This requires special reconstruction method to rebuild surface(s) from sparse contour slices, and this problem is addressed in the second phase of our algorithm. Our modified meshing algorithm is based on the VTK's module called vtkDelaunay3D. The crucial part in the modification is the automatic generations of alpha value for different "blocks" of the sparse contour slices. For each block, we

derived a suitable alpha value, which we then applied the vtkDelaunay3D to reconstruct the surface from the block of slices.

We assess the effectiveness of the proposed method through ten (10) datasets with various types, complexities and sizes. For each dataset, we collect quantitative measure like the reconstruction time and the qualitative measures via visual comparison. Through the experiments, we achieved slice reduction in the range between 40 – 60 % and this result has improved the reconstruction time between 60 - 90 %. Through visual comparison, we showed the regions in visual output that are overlapping or detaching (between the original vs. reduced). These are consistent with the principles set in our proposed method.

We conclude that the proposed method improves the computational speed of the reconstruction process with an acceptable level of output quality; major improvement is evident in the case of large and complex dataset.

CHAPTER 1

INTRODUCTION

Traditional clinical method of medical image viewing used two-dimensional (2D) images to analyze patients' illness. These images are captured in a stack of two-dimensional (2D) data sets by the medical imaging modalities such as Computer Tomography (CT), Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET) and others. Then, the radiologist applies some post-processing with the help of computer and print out the resulting 2D pictures. These pictures are then analyzed by experts in 2D view using a light box for instance. While this process is simple (and perhaps fast) it requires the expert to form the 3D "image" mentally, and such visual thinking skills depends on experts proficiency and experience in the area.

The developments of three-dimensional (3D) surface reconstruction techniques solve this 3D mental image formation problem – the low-level preconscious visual processing is now replaced by constructing 3D structures from 2D data sets using computer. This provides accessibility to non-experts, free the users to focus on high-level conscious tasks of image interpretations and analysis in a consistent manner. There exist many types of surface reconstruction techniques, some work directly with the 2D slices (treated as a volumetric data set) to extract and display the 3D structures – methods in this category includes isosurface e.g. Marching Cube (Lorensen and Cline, 1987) or Volume Rendering e.g. ray-tracing. While on the other end, 3D structures can also be constructed on a slice-by-slice basis, relying on some pre-processing steps to extract feature(s) of interest from the slices e.g. a segmentation operation performed to extract bony structure from CT human head dataset for instance. Usually the former category of methods is used to reconstruct 3D structures of complex features – a tumour cell for instance is difficult to separate from a healthy cell, for this we require a good pre-segmentation process to extract the tumour “structure” from the slices.

A brief introduction of surface reconstruction from 2D slices is described in the next section. Figure 1.1 summarizes surface reconstruction process into a diagram.

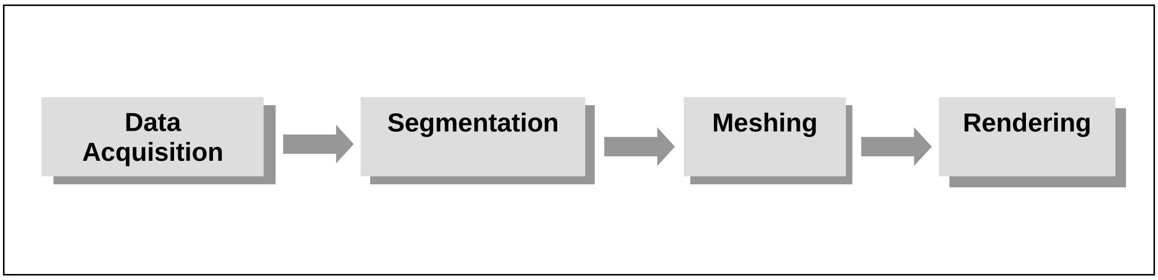

Figure 1.1 Surface reconstruction from 2D slices pipeline.

1.1 Brief Introduction of Surface Reconstruction

Surface reconstruction rebuilds the model stage by stage from Data Acquisition, Segmentation, Meshing and Rendering. More detailed introductions are presented as followed:

Data Acquisition

In our context, data acquisition is to capture sampled data from real world to be manipulated by computer. The acquired data is stored (various formats) in computer for display or further analysis.

The most common standard format is Digital Imaging and Communications in Medicine (DICOM) which is widely used in many hospitals. The DICOM files contain both header (which store patient’s information) and image data (which can contain information in three dimensions). In addition, many scientific packages used ANALYZE format which is a single multivolume image with a header file. For reconstruction purposes, the only required data are the images data, identified as slices or cross sections. Thus, the first step of models reconstruction is to sample a stack of 2D images dataset from DICOM or ANALYZE files.

Most of the imaging modalities (especially CT) capture the images in *axial* orientation or projected on Z-plane rather than *coronal* (Y-plane) orientation and *sagittal* (X-plane) orientation. Nowadays, these modalities are capable of producing the 2D images with high resolution in order to preserve the accuracy. For instance, a craniofacial CT dataset consists of approximately 124 slices with 512 X 512 pixel and distance measuring in 0.35 X 0.35 X 1.25 mm.

Segmentation

Segmentation is known as the process of isolating multiple regions from a digital image. The aim of segmentation is to extract the features of interest from an image for further analysis. The typical output of image segmentation is a set of contours extracted from the image where each of the pixels from the region is similar with respect to some characteristic such as colour, intensity or texture. Image segmentation is important and used in various practical application, e.g. treatment planning, study of anatomical structure, tumours identification and computed-aided surgery.

Many image segmentation techniques have been developed for specified purpose such as *thresholding, region growing, edge-based, clustering* and *histogram-based* approaches. Thresholding utilized the similar pixel's intensity values to detach the same region from an image. Region growing used intensity information and/or edges in the image to extract the connected region. Edge-based approaches use image gradient to extract the boundary of the object. Clustering is used to partition an image into K clusters using an iterative technique. In histogram-based approaches, the peaks and valleys of the histogram are used to establish the clusters in the image.

Meshing

A surface is built by stacking segmented closed contours data (refer Figure 1.2). The output is a mesh triangle which is appropriate to form the 3D object approximating the contour's structure.

There are two classes of methods used to generate surfaces from closed contours data. The first group is *Slice-Based* approaches. In Slice-Based approaches, surface is directly connected by contour points of adjacent slices. Contour points are joined with only straight lines where each triangle consists of two points from a slice and one point from the consecutive slice. In other words, Slice-Based approaches produced surface which is only c^0 continuity (surface with patches just touching). The smoothness or the accuracy is influenced by the number or density of contour points in the slices.

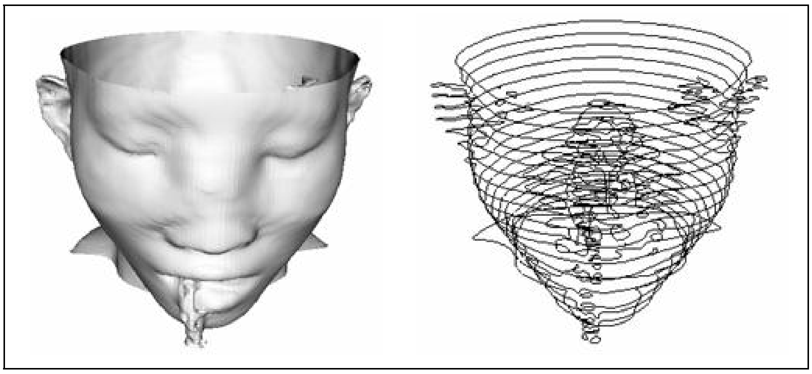

Figure 1.2 Closed contours data of human's head skin dataset.

The second group is *Volume-Based* approaches which estimate a 3D function from the contour points to generate triangular mesh. The 3D function represents some distances measurement from any point to surfaces. Once the function has been created, the isosurface in zero set can be triangulated to expose the mesh surface. It involves more computational power especially with dense dataset. The generated mesh surfaces are smoother compared with Slice-Based approaches. Volume-Based approaches regularly required distance field and/or medial axis to establish the correspondence region of adjacent slices and employed specified mathematical formulations such as "Eikonal" equation in the process. In addition, algorithms such as Marching Cubes (Lorensen and Cline, 1987) extract the surfaces directly from dataset.

Rendering

Rendering referred as the final step of surface reconstruction to display 3D geometry into pixel that can be viewed on screen. Various visual cues are calculated in this stage such as shading, lighting, transparency, colouring and texturing.

1.2 The Main Concerns of 3D Reconstruction

In this thesis, we considered two main concerns of surface reconstruction: the *Speed of Computation* and the *Accuracy of Output*.

1.2.1 Speed of Computation

Computational speed refers to the processing time to reconstruct 3D surface(s). It is influenced by the several factors such as the *data size* and the *complexity of the algorithm employed* on which this process is deployed.

- *Data Size:* In slice-based surface reconstruction, the slices are stacked in volume and surface reconstructed from the slices. The number of slices determines the size of dataset. The larger the size, more information can be used to produce accurate output. However, it takes longer to produce complete 3D reconstruction from the larger data size. In addition, many surface reconstruction algorithms exhaustively process every single slice even in the case where these slices are considered insignificant contributions to the overall output accuracy.

- *Complexity of the Algorithm Employed*: Some of the existing surface reconstruction algorithms employ complex equations to assist 3D reconstruction. The computational time for these equations affected the speed of computation. The computation time of complex equation is longer than simple equation. Some algorithms are quite complex in term of finding the point's correspondence for mesh generation e.g. distance transformation, medial axis or normal vector calculation.

1.2.2 Accuracy of Output

Output accuracy can be identified as minimal errors of the approximated surface compared to the original surface. Good output accuracy is where the reconstructed meshes resemble closely to the original surface. Output accuracy is influenced by several factors including *density of dataset*, the *shape of contour structure* and the *size of mesh*.

- *Density of Dataset*: In surface reconstruction, contour data are used to construct a triangular mesh representing surface. Surface more approximate to data points produced by denser dataset.

- *Shape of Contour Structure*: Extracted contour points provide the base dataset where surface is to be reconstructed. Hence, indirectly the accuracy of surface(s) reconstruction from contour slices is influence by the segmented structure.

- *Size of Mesh*: A mesh consists of large number of triangles primitives. Merging these triangles into larger polygon often increased the rendering speed yet decreased the accuracy of approximated surface.

1.3 Research Objective

In this research, we aim to first speed up the computation time of 3D surface reconstruction from 2D slices, and second we want to preserve accuracy of the output while satisfying the first objective.

We address the first objective by proposing a slice filtering algorithm to reduce the number of slices from a stack of 2D images with certain criteria, called Optimum Slice Reduction algorithm. The word Optimum refers to select the slices optimally in order to balance between the speed of computation and accuracy of output. The second objective is tackled by proposing a modified meshing algorithm (Block Triangulation algorithm) to triangulate our reduced slices. Details of these algorithms will be discussed in chapter 3.

1.4 Thesis Organization

This thesis consists of six chapters. The following is an overview of each chapter:

Chapter 1 – Introduction

A brief introduction of 3D reconstruction is discussed in the first section of this chapter to provide a refreshment and notation of the process for a good understanding of following chapters. In the second section, two main concerns of reconstruction are established with details explanation. Lastly, the goal of this research is presented to guide a proper inspiration of this research.

Chapter 2 – A Survey of Surface Reconstruction

In this chapter, we specify the discussion in surface reconstruction pipeline. We focus the discussion in related procedure in surface reconstruction pipeline which is pre-meshing, surface reconstruction itself and post-meshing. Besides, we highlight several existing methods which the surface reconstruction from sparse contours to accomplish the survey related to this thesis.

Chapter 3 – An Optimum Slice Reduction Algorithm

The proposed Optimum Slice Reduction algorithm is introduced here with fundamental discussion and description in detail. With occurrence of this algorithm, a modified triangulation algorithm is proposed, called Block Triangulation to reconstruct mesh(s) from sparse and inconsistent data. Demonstrations are provided to advance the schemes of the methodology.

Chapter 4 –Implementation

A brief introduction of related tools is presented in the beginning of this chapter followed by design and implementation of proposed methodology. Essentially, the implementation evaluates the proposed method practically, included Data Preparation, Optimum Slice Reduction, Surface Construction and Output Visualization. Flowcharts are adapted in each module to improve reader's brainwave for the architecture.

Chapter 5 – Experiments, Results and Performance Analysis

This chapter starts with explanation of experiment setup including the execution equipments, sampled datasets for testing and evaluation metrics collected to analyze the performance. The results are collected according to two case studies to evaluate the performance of proposed method in two portions with short discussion. Coloured images are included for visual comparison. This chapter ends up with a summary of the overall performance.

Chapter 6 – Conclusion and Future Work

A summary of this research work with the overall work flow of research, the exact contributions and weaknesses of methodology are discussed. Beside, the outline of future works to improve the performance is listed too.

CHAPTER 2

A SURVEY OF SURFACE RECONSTRUCTION

In this chapter we survey techniques for reconstructing surface(s) from contour slices. In addition, we also discuss procedures related to surface reconstruction, specifically on techniques that perform *pre-meshing* and *post-meshing* operations. Since our interest in this thesis is to trade-off between visual fidelity with speed of computation, it is crucial that we highlight the entire pipeline of surface reconstruction process. Doing this should assist us to categorize the main processes in a typical surface reconstruction, more importantly however, we can identify stages where an improvement can be made, hence we can position our contributions in the context of surface reconstruction pipeline.

This chapter is divided into several sections, guided by the general pipeline of surface reconstruction. We start with our view of a typical pipeline for surface reconstruction; this is then followed by sections that discuss the various stages in surface reconstruction pipeline – detailed reviews of existing techniques in each stage are further elaborated in the corresponding subsections. Finally, we conclude this chapter with several points that leads to the idea proposed in this thesis.

2.1 Surface Reconstruction Pipeline

The issue of accuracy versus speed has long been discussed in the literature. In the context of surface reconstruction from contour slices, the solutions to this trade-off can be divided into three categories: pre-meshing, meshing itself (surface reconstruction), and the post-meshing stage. Figure 2.1 illustrates these stages in a typical surface reconstruction pipeline. The figure also shows (in dark colour) two optional stages – pre-meshing and post-meshing.

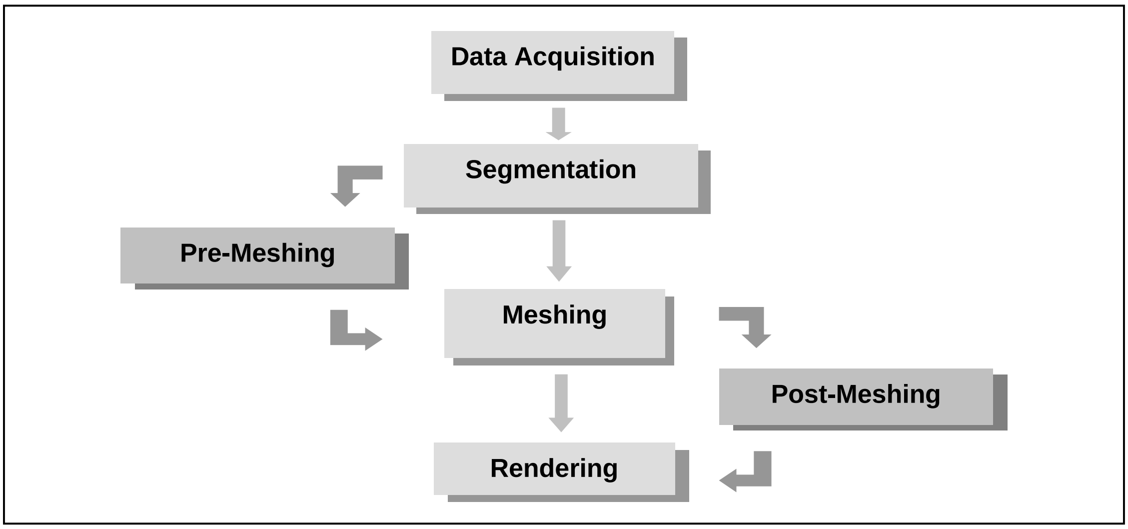

Figure 2.1: Pre-meshing, meshing and post-meshing in 3D reconstruction pipeline. Darker colour indicated optional process while lighter colour indicated essential process in the pipeline.

While it is known that the stages of *Data Acquisition* and *Segmentation* could have some influence on the performance of the subsequent processes, in this thesis we will assume their influence in term of visual accuracy and computational speed is relatively small compared to the influence imposed by the other stages in the pipeline. Instead we will focus our discussion on the pre-meshing, meshing and post-meshing processes - the stages where we anticipate will dominate the computational time of surface reconstruction pipeline. These are also the stages where we can control visual fidelity and traded it off with computational requirements.

The following subsections survey related works in each of these stages.

2.1.1 Pre-Meshing

We broadly classify techniques or methods that belong to pre-meshing as those that do some processing to the contours (extracted from the previous segmentation stage) prior to the meshing or surface reconstruction process. In the context of surface reconstruction pipeline (refer to Figure 2.1), techniques belong to this category are optional. The sole purpose of pre-meshing is mainly to prepare the data (contour slices) for subsequent processing. Typical processing includes removing noise/outliers from segmented data, fixing missing data via interpolation or approximation, converting scattered data into structured format, re-samplings dataset into coarser or finer resolutions, and many others.

In general a pre-meshing solution modifies original dataset in some ways, and the derived data will then form the input to the surface reconstruction algorithms. There are two ways to address pre-meshing: one is performing pre-meshing on the slice/plane, also known as intra-plane pre-meshing and the other one is inter-plane pre-meshing where this is analog to reduce data point along the third dimension, the number of slices.

Intra-plane Reduction

Eck et al. (1995) and Meyers (1994) performed multi-resolution analysis to simplify the contour curves on each slice, this ultimately help to accelerate the tiling process in the surface reconstruction stage. The technique starts with a very simple polygonal approximation to the original curves, known as base model. This model is derived by sub-sampling the original curves using multi-resolution analysis. The algorithm proceeds on subdividing the base model until the resulting surface lies within user specified threshold from the original surface. The drawback of these algorithms is that the recursive subdivision of the base model may not be able to capture the exact geometry of the original model. Figure 2.2 shows the multi-resolution tiling of Meyers.

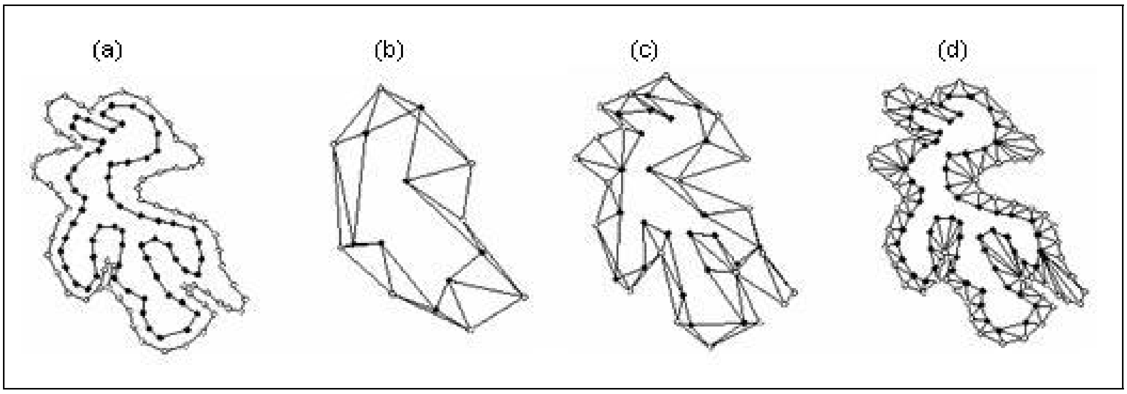

Figure 2.2 Multi-resolution tiling of Meyers. (a) Original contours. (b) Lowest resolution for tiling. (c) Intermediate level. (d) Final tiling.

It is probably too early in the surface reconstruction pipeline to do geometry simplifications, and worse the simplification is based on the geometric properties of the curves – the relation to the original data features is probably very weak at this stage.

Inter-plane Reduction

A common processing at inter-plane is to resample data point into either coarser or finer resolution in the third dimension. Sub-sampling to coarser grid is one of the examples of inter-plane dataset reduction (implicitly this implies coarser grid resolution per slice). In VTK (Schroeder and Lorenson, 1996), a filter object known as **vtkExtractVOI** for instance, allows sub-sampling at a coarser grids along the z-direction. While this solution in general reduces the number of slices to process, and potentially improves the overall computational cost of reconstructing surface over the original dataset, it has certain drawbacks. The sub-sampling is exhaustively done over the entire dataset, local variations of the feature(s) between adjacent slices often linearly interpolated to get derived slice(s) which now form the coarser grids along the z-direction. The result is often structured grid with half or less the size of the original dataset. Depending on the type of interpolation method employed, this approach can incur substantial cost to the surface reconstruction computation time. A better approach is perhaps to adaptively sample the dataset to preserve local variations while keeping the cost of computation relatively low for the entire surface reconstruction pipeline.

One major disadvantage of this approach is once the original data is re-sampled, it is considered lost – unrecoverable or lossy reduction – making it impossible for the subsequent processes to recover the original features in the dataset.

2.1.2 Surface Reconstruction

Attempt to trade-off visual fidelity with speed has been done at the surface reconstruction algorithms themselves. The three universal problems addressed in surface reconstruction are correspondence, tiling and branching problems (Bajaj et al., 1996 a). These problems are graphically illustrated in Figure 2.3.

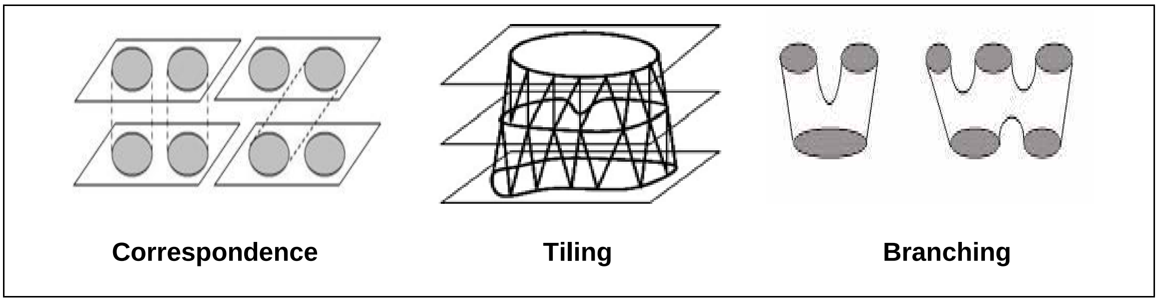

Figure 2.3: The three universal problems in surface reconstruction.

- *Correspondence* problem referred to the correct connection of contours between the adjacent slices

- *Tiling* problem indicates vertices linking and edges joining between the adjacent slices
- *Branching* problem occurs when different number of contours appeared in two adjacent slices

In general, matching or connecting correspondence contour points in adjacent slices is a difficult and time consuming to resolve. This is further amplified by the nature of the data (constant contour features vs. irregular contour features between adjacent slices) and indirectly influence by the size of the input data. Thus, whenever possible, we try to prepare the input data i.e. the contours, either by simplifying the contours or by reducing the input size prior to reconstructing surfaces.

The rest of this section surveys techniques for reconstructing surfaces from contour slices. The techniques to be discussed are further divided into two categories: slice-based and volume-based techniques. In a slice-based category, surface is reconstructed by direct triangulation of points on the cross-sections or contours. On the other hand, volume-based category approaches surface reconstruction based on implicit function or 'level set' where distance function is calculated, and zero isosurface is triangulated to reveal the surface.

Slice-based

Since 1970's, the problems of reconstructing models from contour data have been investigated by Keppel (1975) with limit to a triangulation from each pair of consecutive slices. Other methods were proposed later and can be divided in two groups: optimal and heuristic. Optimal methods aim to provide the best triangulation with given criterion. These methods are based on description of all possible arrangement from a graph and an optimal solution with respect to specified criterion. For instance, Keppel declared an expression to demonstrate the number of possible triangle permutation and permit an exhaustive search for the optimal triangulation. He models the pair wise contour tilling method using directed graph and optimized the triangulation by searching maximal weighted path. Two years later, Fuchs et al. (1977) presented an optimal solution by finding certain minimum cost of cycles in directed toroidal graph. His method requires a large number of steps. These optimal methods essentially provide fairly good results, but are very time consuming.

Heuristic methods can be described in the same way as optimal methods in tiling process but not require searching for maximal or minimal weighted path. However the weighted path must be computationally inexpensive and based on local constraints. Ganapathy and Dennehy (1982) introduced a heuristic algorithm which uses parameters to guide selection of contour points on either upper or lower slices. Most of these methods are adequate when the contours have similar shape and orientation and are mutually aligned. Incorrect results occurred for contours with significant changes in shape, orientation and position. Furthermore, this method only solved a single contour reconstruction whereas the branching problem is not resolved. Ekoule et al. (1991) proposed a new heuristic general algorithm which provides solution to handle single branching of dissimilar shapes and orientation contours. He assumed that branching from a section of model has practically the same position in their consecutive slices and created an intermediate level to interpolate mesh surface. His method suffers from variety of problems and works only when with two slices closely resemble each other. Barequet (1994) used a partial curve matching technique to match contour portions and remain with the unmatched portions. Then he used a minimum spanning tree heuristic to interpolate between non-connected regions. His method successfully handled multiple contours reconstruction, yet it is time consuming to process.

In 1980's, a new paper by Boissonnat (1988) presented the first contour-based surface reconstruction based on Delaunay triangulation for each pair of slices. Figure 2.4 shows an example of Delaunay triangulation with Voronoi diagram. The Delaunay triangulation of a set of points is a triangulation of a set of points with the property that no point is interior of the circum sphere of any triangle in the triangulation. The union of tetrahedral formed the desired surface. The complexity of his algorithm depends on the number of tetrahedral in the Delaunay triangulation between two adjacent slices.

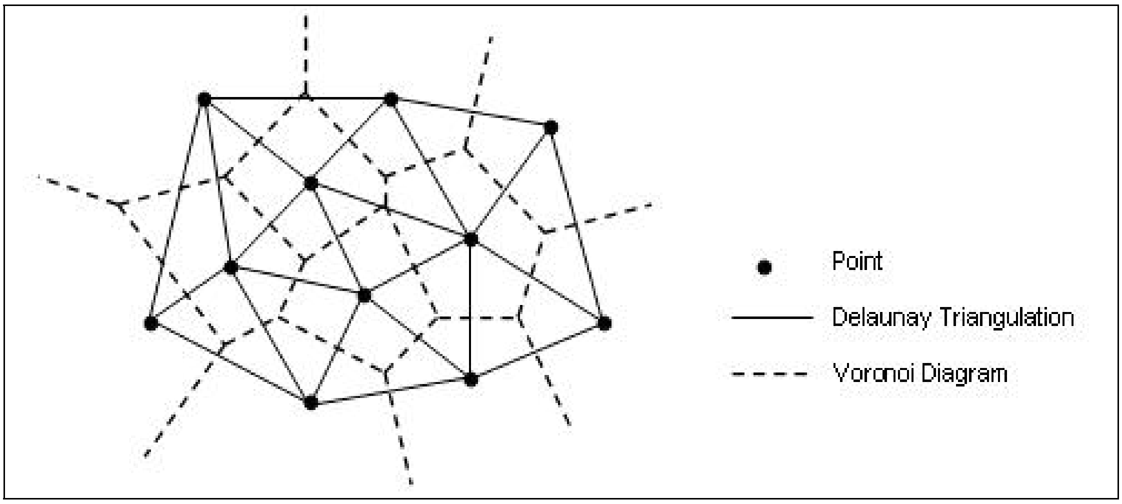

Figure 2.4: Voronoi diagram with Delaunay Triangulation.

Geiger (1993) modified Boissonnat's method to design a reconstruction algorithm. He showed its underlying theory and its applications to tomography data. He revealed that the complexity of his implementation has O(n) per triangle where n referred to number of points. Bajaj et al. (1996 a; 1996 b) tiled the untiled region using Voronoi diagram. In his paper, he implements two algorithms, one is to construct surface element and the other is to construct tetrahedral finite elements from contour data. The time complexity of this approach is O(n Log n) in average case and increased in worst case. Dance and Prager (1997) presented the Delaunay reconstruction from arbitrary planar contour data with complex holes and branching. The technique has been illustrated with parallel cross sections CT pelvis data. A polyhedral surface between arbitrary pair of contours with *n* vertices in total can be obtained in O(n) time. Thus, the number of contour points affects the triangulation time.

Volume-based

Raya and Udupa (1990) reported a new shape-based interpolation for multidimensional images and compared with existing gray-level interpolation. Their proposed method is able to preserve the shape of model even in the temporal domain. Yet, an additional step reason by computationally more expensive, the calculation of distances. Chatzis and Pitas (1999) proposed a shape-based interpolation which uses mathematical morphological operators to interpolate the object's skeleton. The skeletonization consists of a small percentage of the total model's points. Therefore, various additional computations are requested to accomplish the skeletonization and reconstruction process.

More recently, Treece et al. (2000 a; 2000 b) improved shape-based interpolation with a new approach referred to as Maximal Disc Guided Interpolation. Each contour is represented by a number of maximal discs to involve in correspondence calculation assisted by distance transformation. This method predicts the shape accurately and solved the multiple branching problems. Unsurprisingly, this method is time consuming to accomplish the calculation for each slice; long thin objects take more time to process than more spherical objects.

Lee and Wang (2000) proposed a surface interpolation technique based on mathematical morphology operators to solve branching problems. They first employed dilation to calculate the morphology difference area then applied erosion-based interpolation to this region. This proposed method is simpler in computational complexity and created smooth surface but in some situation, the method cannot handle well, for example an image with narrow concavity. Figure 2.5 is an example that cannot be handled well by this method. The morphology operator cannot provide an appropriate interpolation for this case.

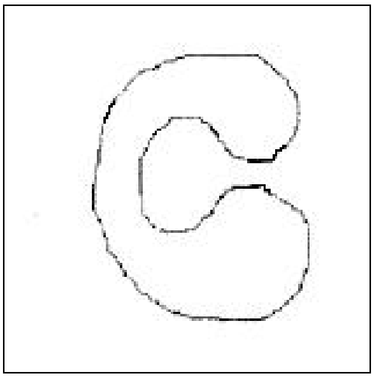

Figure 2.5: An example of narrow concavity which cannot be handled well by Lee and Wang's method.

Another work reported by Albu et al. (2003) utilized morphological morphing to generate intermediate slices between two adjacent input slices and linked the adjacent slices into a triangular surface mesh via iterative triangle generation. Figure 2.6 illustrates a sequential of dilation based morphing. But again, some cases are excluded and computationally expensive to process.

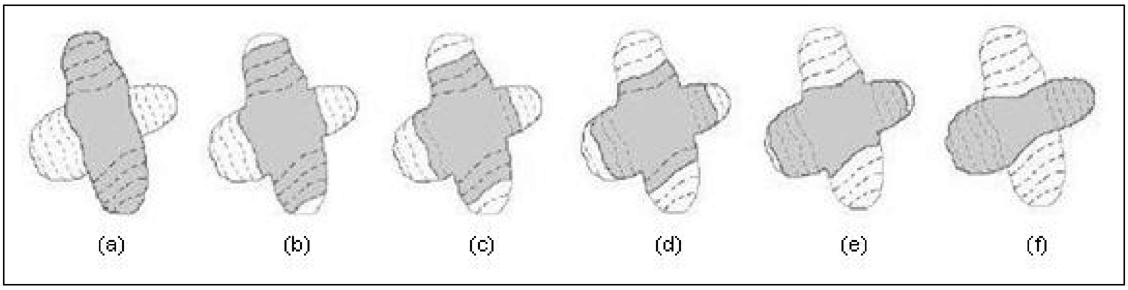

Figure 2.6: Morphing between two objects by creating intermediate slices presented by Albu et al.. (a) Two objects in initial stage; (b), (c), (d) and (e) Intermediate slices creation; (f) Final object.

An extension of morphology based surface reconstruction, a new numerical technique, Level Set method is designed. Sethian (1996) presented a fast marching level set method which leads an extremely fast scheme for solving the numerical equation in level set. Zhao et al. (2001) developed a fast algorithm based on variation and partial different equation (PDE) to reconstruct implicit surface. Unlike typical techniques, his variation level set formulation is executed with optimal efficiency. No doubt, this method is computationally expensive.

Surface Reconstruction from Sparse Contours

Most of the surface reconstruction techniques presented above work well for a regular and relatively close contour slices (especially for the slice-based category). This type of data is quite typical for medical data capture through scanning devices such as CT, MRI, fMRI, and others. There have been several recent research interests in reconstructing surface from non-regular sparse contour slices. Here we will provide a brief look at recent research in this area since these works are relevant to idea presented in this thesis, particularly in the context of reconstructing surface after the slice optimization process is done.

We can adapt the technique proposed by Hoppe et al. (1992; 1994) to reconstruct surface(s) from sparse contour, if we assume these sparse contour points as clouds of unorganized points. Indeed, the difficulty of employing this approach will depend on setting of the various parameter values in the algorithm. Breen et al. (2001) employed 3D metamorphosis (morphing) technique based on level set methods to reconstruct several types of geometric models such as polygonal meshes, CSG model and MRI scans from sparse contour slices. The method first converts the data into distance volumes and then utilizes level set morphing techniques to represent volumetric mesh. While this method has been shown to produce a smooth and close surface, its main drawback is the excessive computational requirements.

The latest effort to move away from 'level-set' in Breen et al. (2001) work is presented by Nilsson et al. (2005). They avoid the excessive computational requirement of level set to morph sparse contours by employing MPU (Ohtake et al., 2003). A local function is fit into a set of points to represent part of contour's curve, and then by selectively choosing the appropriate weight, they blend these local functions to approximate the shape of contour per slice. The morphing between contours is achieved through high degree interpolations, making sure the transition between base contours is smooth, and creating a well defined close surface geometry. They claim the technique can work well even for small number of contour slices, as low as four base contours.

2.1.3 Post-Meshing

Post-meshing methods are concerned with reducing or simplifying the output (polygonal data type) produced by surface reconstruction algorithms. The aim is mainly to improve speed of rendering. The meshing techniques discussed in previous subsection 2.1.2 often produce geometry of low quality (badly proportioned triangles) and in addition surfaces of medical data are often composed of a large number of triangles, thus proportionally taking longer time to render. Methods in this category may improve the computational cost of the rendering part whereas the performance of the surface reconstruction (meshing) part is unaffected. Unlike the case for pre-meshing as discussed in see subsection 2.1.1, research that deal with post-meshing received more attentions where numerous solutions and implementation has been proposed. In this subsection, we will only touch the major works in the area. For more detail treatment of the subject refers to (Heckbert and Garland, 1997). We categorize the related techniques to geometry removal and sampling.

Geometry Removal

The most popular geometry removal simplification method is decimation which starts with a triangulated mesh and consecutively simplifies it until the desired level of approximation is achieved. Decimation methods are classified into vertex decimation, edge decimation, triangle decimation and patch decimation (Heckbert and Garland, 1997).

- Vertex decimation deletes a vertex and re-triangulates the deleted vertex's neighbourhood.
- Edge decimation merges two edges after deleting one edge and two triangles.
- Triangle decimation deletes one triangle and three edges, then the three vertices are merged and the neighbourhood is re-triangulated.
- Patch decimation deletes more than one adjacent triangle. The boundary is then re-triangulated.

Schroeder et al. (1992) explored vertex decimation which iteratively removes vertices from mesh and re-triangulation the resulting holes. Essentially, their decimation method is designed to Marching Cubes (Lorensen and Cline, 1987) algorithm‘s output. Multiple vertices are deleted per pass. This method is fast and less expensive but less accurate error measure probably has lower quality.

Hoppe et al. (1993) proposed a mesh optimization method based on minimizing global energy measure. The method proceeds in three nested loops and optimizes the topology by using random selection and heuristics to choose an edge and either collapses it, splits it or swaps it. The geometric optimization finds the vertex positions that minimize the global error for a given topology. In other words, this method repeated twice changes to the mesh in order to allow better fit and simpler mesh. This method is considered to be slow but accomplishes very good simplification.

Progressive meshes presented by Hoppe (1996) are a simplified version of the previous method (Hoppe et al., 1993). Rather than preserving the geometry of the original mesh, it preserves the overall appearance such as colour, texture and normal discontinuities over the surface of the mesh. He proposed that the resulting meshes are just as good as or even better than the meshes from mesh optimization algorithm (Hoppe et al., 1993). Yet the preservation of extract attributes rather than geometry is not in our concentration.

Hamann (1994) presented an iterative triangle removal principle to reduce the number of triangles from the triangulation. In this algorithm, triangles in nearly planar surface regions are priority to reduce. The degree of reduction can be specified either by percentage or by an error tolerance, in the case of bivariate functions. The algorithm is rather complicated and has not been generalized to higher dimensional triangulation.

With multi-resolution hierarchical computing, Varshney (1994) created an efficient algorithm to generate level-of-details (LOD) approximate to a provided polygonal mesh. The validity of all triangles defined by three points are checked. Once valid, the qualified triangle is inserted and the old triangle is deleted and small triangles are used to fill the crack between new and old triangles. This algorithm generates good approximations yet it is very slow with time cost ranging from $O(n^2)$ to $O(n^6)$.

Simplification envelopes proposed by Cohen (1996) which is initially envisioned by Varshney consist of two envelopes construction, an outer offset surface and inner off surface. The constructions of each envelope are not more than a user defined distant from the original surface and not self-intersection to avoid computational expense and correction of the self-intersection. The envelopes are simplified by triangles decimation principle. The resulting simplifications tend to have very good fidelity yet it is computationally expensive especially during the envelope construction phase.

Many simplification algorithms fall in this category (decimation) but vary from their selection mechanism and measuring the approximation error where these processes may be time consuming. For instance, Garland and Heckbert (1997) proposed a surface simplification using quadric error metrics. They developed an iterative vertex pair's contraction to simplify the models as well as join unconnected regions and track surface error approximations using quadric metrics. In this paper, they declared that this algorithm strategy as middle ground between vertex clustering which