Founded in 1905

A HYBRID DIGITIZATION MEHTOD FOR REVERSE ENGINEERING

BY

WANG YUE

(B.Eng., M.Eng.)

DEPARTMENT OF MECHANICAL ENGINEERING

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

NATIONAL UNIVERSITY OF SINGPAORE

2006

ACKNOWLEDGEMENT

I would like to express my sincere respect and gratitude to my research supervisors, A/Prof. Zhang Yunfeng and A/Prof. Loh Han Tong for their invaluable guidance and advice in the entire duration of the project. A/Prof. Zhang Yunfeng has shared his knowledge and provided me with strong feedbacks and invaluable advice on my research. A/Prof. Loh Han Tong has provided me with strong encouragement and inspiration on my study.

I would like to give my thanks to Ms M. Shi who gave me much help on the programming of triangulation part. My senior Ms L. L. Li and Mr. Y. F. Wu would gain my lots of thanks for their selfishless and grateful help on my study and research. My thanks will also be given to Dr. Z. G. Wang, Mr. T. Li, my junior Miss H.Y. Li and all the lab-mates for their cheerful accompany and help. In addition, I would like to thank Mr. Wong and all the other staffs in Advanced Manufacturing Lab for their technical help in my research.

I would like to thank my family for their love. They always support and encourage me to step forward in my study and in my life.

Finally, I would like to express my acknowledgement to the National University of Singapore for the research scholarship.

TABLE OF CONTENTS

SUMMARY

Methods for acquiring shape data generally can be classified into two categories: contact and non-contact. Laser scanners are popular non-contact devices due to its fast acquisition rate. However, there is no guarantee that the important feature information (e.g., boundaries and holes) is captured because of the reflection and the topology of the part. Furthermore, merging of data points from multiple views also introduces errors and leads to redundant data points. On the other hand, CMMs are more accurate devices but with low acquisition rate. Therefore, it is preferable to combine the use of scanner and CMM for digitisation. The scanner can be used to quickly capture a set of rough shape data, which is then used as a reference model for planning the probe-path of the CMM to capture the feature information. A more complete and accurate set of shape data can be obtained by combining both data sets.

In this thesis, a hybrid digitization method is developed. Firstly, a filtering algorithm is developed that is able to thin the merged data points by eliminating noise and spurious data points with a user controlled tolerance bound. Secondly, we applied a region growing based triangulation algorithm to this set of thinned cloud data to form a triangular meshed surface model, with explicit feature information (boundaries & holes). Finally, we developed an algorithm to generate the probe paths for a CMM to recapture the key features of the object based on the information obtained from the triangulation process.

Algorithms for the developed methods have been implemented using C/C++ on the OpenGL platform. Simulation results and actual case studies demonstrate the efficacy of the algorithms.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Reverse Engineering and its Applications

Reverse engineering (RE) is the process that creates a digital model from an existing physical object, which can be used in engineering analysis, manufacturing and rapid prototyping. RE has played an important role in modern industry and biomedical field in recent years. The typical process of reverse engineering begins with collecting point data from the surfaces of a physical object. Either contact or non-contact method is used to obtain the object's surface data. Contact type devices are generally more accurate but slow in data acquisition especially for free form surfaces. After data acquisition, usually a pre-procession process, such as noise filtering, smoothing, merging and data thinning is applied to the obtained cloud data. The resulting cloud data is then put into a modelling package, which creates a geometric model suitable for rapid prototyping and/or CNC machining.

Today, RE technology has become a very useful tool and it can be used in many situations. Some typical examples are listed as follows:

(1) Replicate the part: when the product exists in a designer's medium, such as clay or wood, it must have its surfaces digitized so that it can be converted to a computer-based representation for manufacturing or when part drawings are not available in computerized form, such as for some proven old designs or antiques.

(2) Modification of the existing part: when a product design changes greatly during the process of development and modifying the original design needs too much work or it is hard to get the final computerized model by revising the initial one.

(3) Products designed are according to the customer' request (such as shoe and hearing-aided instrument). RE technology is a very timesaving and economical method to realize it.

(4) RE also can be widely used in medical field such as producing artificial bones.

1.2 Data Acquisition Approaches in RE

There are mainly two steps in RE: (1) part shape data capturing and (2) shape data modelling. Data capturing is a crucial step in RE, and there are mainly two different approaches for acquiring shape data: non-contact methods (e.g., optical, acoustic and magnetic methods) and contact approaches, (e.g., CMM). Light, sound or magnetic fields are used in the former method, and in the later methods the surface is touched by mechanical probes at the end of an arm, which is a time consuming process for full-size models that may require thousands of points to adequately define the surface.

Measurement with a CMM is fast and highly repeatable when the measured geometric elements are line, plane, cylinder, sphere and cone etc., because the process requires only a limited number of probing points and the probe radius compensation is quite straightforward. The machine can be programmed to follow paths along a surface, which can usually produce accurate results up to 10μm or better and collect very accurate, nearly noise-free data. However, in the case of complex surfaces requiring numerous measurement points, this kind of measurement encounters difficulties (Yan and Gu 1996).

With comparison to contact measurement approach, the non-contact method has a more rapid measurement speed, e.g., the VIVID 900 laser scanner can capture over 300,000 points in 2.5 seconds with a fast mode scan. However, the corresponding measurement accuracy is usually in the range from several hundredths to several tenths of a mm. Moreover, there are also some problems for the non-contact measurement methods (Varady et al., 1997, Lee and Woo 2000). This kind of method tends to pick up redundant points and the distributed density of points measured in the digitization steps often does not meet the surface geometric trend. The bright and shiny regions are difficult to be measured by optical methods. A typical digitisation process requires the range sensor to be re-positioned six-to-eight times, at various positions and viewing angles, to completely sample the surface patches. The resulted cloud data file is a mosaic of range data patches. The obtained wrap-around model consisting of data points from registered range map has shortcomings as follows:

(1) The obtained cloud data model contains regions of overlapping data.

(2) The mosaic of range data points that forms the global model does not have the ordered row and column structure of a single range map.

(3) The model consists of unconnected data points – the connectivity of points in adjacent views is not readily established.

In summary, laser scanner and CMM are two kinds of method widely employed to obtain the cloud data of the object and each has its pros and drawbacks.

1.3 Data Compression Approaches

When the laser scanner is used to collect the data points, it often results in a large mount of redundant data points. To achieve the trade-off between the efficiency and accuracy in the following process of the data points, such as triangulation, control and manufacturing, the copious data set is required to be thinned from the range [300,000-30,000] to the range [30,000-3,000] points.

There are many approaches that can achieve the goal of compressing cloud data. Some approaches are based on mesh simplification. In those methods, vertices are deleted/inserted and then meshes are merged/split. And any level of reduction can be achieved by estimating and control errors locally or globally. The input of these approaches is usual structured/unstructured triangular/quadratic meshes and the output is simplified meshes within a given tolerance. Hinker and Hansen (1993) merge several polygons into one polygon by implementing co-planarity criteria. Soucy and Laurendeau (1996) remove vertices that meet minimal distance or curvature criteria by analyzing geometry and topology. Hoppe (1996) and Eck et al. (1995) developed a PM algorithm to reduce the mesh size by subdividing connectivity for structured meshes.

Multi-level representation, which has two common multilevel structures (multi-resolution wavelet model for surface representation and level of detail model for meshes representation), is proposed to cope with the development of remote rapid prototyping. In these methods, a multi-level model is firstly rebuilt based on a surface or meshed surface representation as input and then any level of detail can be achieved to satisfy the requirement of data compression, modelling or prototyping. The wavelet method (Lucier 1992, Stollnize et al. 1995) is one kind of multi-resolution approach that does local measurement in wavelet spaces to build triangulated spaces. As for the LOD (level of detail) method, data structure such as Octree (Williams 1983, Tnamminen et al. 1984) and representation of structured meshes such as Quadtree (Samet 1990, Shephard and Georges 1991) are applied to store data. Local or global error control can be applied by implementing error estimator at each node, so that geometric details can be preserved even at any level of resolution. For the case of unstructured meshes, it will be pre-processed by mapping directly to a hierarchical structured mesh so that each node in the Quadtree includes at most one sample point.

The voxel binning method proposed by Weir et al. (1996) directly reduces a set of unstructured cloud data to a uniform data points set. In this method, a maximum bounding box is created around the whole data in 3D space. Then this volume of data is subdivided into uniform cubes, called *voxel bins*. Furthermore, the data points are allocated into each bin and the point closest to the bin centre is retained. Finally a close to uniform data point set is obtained. In this method, selection of the bin size is crucial to the accuracy of the reconstructed surface. Adjusting the bin to different size can lead to different numbers of points left for triangulation and the more points remained for triangulation the more accuracy the reconstructed model can achieve. The voxel binning method is very efficient in subtracting the redundant cloud data and can dramatically reduce the data set. But because it simply thins the data set by arbitrarily selecting a bin size, it cannot guarantee the accuracy between the subsequent triangulation and original data.

The voxel binning method was further enhanced by Sun et al. (2001) who established the relationship between the bin size and the triangulation error, which ensures the accuracy of the triangulation by a user input tolerance, therefore gaining both the efficiency and the accuracy of the cloud data thinning.

1.4 Objectives of Our Research

Date capturing is the crucial part of reverse engineering process. There are many different methods for acquiring shape data. Generally, it can be classified into two broad categories: contact and non-contact. The no-contact optical method of shape capture is most popular used because of its fast acquisition rate. But there are several problems associated with quality of the acquired data. First, it has a lot of noise because of the reflection and the merge of multiple views. For example, the Minolta VIVID-900 scanner has a tolerance of 0.2mm on the data collected. Second, the data acquired is unorganized and has a lot of redundant data due to the scanner itself and multiple views. Third, the collected data has inherently incompleteness, because the important feature information (e.g., holes) may not be captured. On the other hand, CMMs can be programmed to follow paths along a surface and collect very accurate, nearly noise-free data, but they are also the slowest method for data acquisition. Currently, people reconstruct the computerized model of physical object using the point data acquired by only one kind of measuring method.

Therefore, a nature way to overcome this problem is to combine the use of scanner and CMM touch-probe. The scanner can be used to quickly capture a set of rough shape data. A thinning algorithm is used to filter this set of multiple views point data to eliminate redundant data points. Then a triangulation method is implemented to set up the computerized model for the thinned data set. The feature information (boundaries and holes) can be identified from the triangular model. This triangulated shape data is then be used as a reference for planning the probe-path of the CMM to re-capture the feature information (e.g., boundaries and holes). Finally, a more complete and accurate set of shape data can then be obtained by combining both sets of data.

1.5 Organization of the Thesis

The thesis contains six chapters as follows:

Chapter 1 introduces the major processes and applications, data capturing methods, and data compression approaches in reverse engineering. The research objectives are then outlined.

Chapter 2 presents the literature view of three major surface reconstruction methods in RE: segmentation, surface fitting and triangulation. An outline of our hybrid digitization method is presented in this chapter.

Chapter 3 presents the data thinning method, i.e., a shape error-controlled filtering algorithm. This algorithm is based on a most curved region identification obtained by Least Square Surface Fitting of a series of neighbor points for a given point.

Chapter 4 describes the triangulated surface construction method, i.e., a region growing based triangulated model constructed from filtered cloud data. The algorithm can identify the patch boundaries and holes during triangulation process.

Chapter 5 illustrates the probe path planning algorithms which is used to generate reasonable probe path for the CMM to re-capture the feature information (edges & holes) by using the triangulated model's information as a reference.

In Chapter 6, the conclusions and directions for future work are presented.

CHAPTER 2

LITERATURE REVIEW

Shortening the cycling time between design and manufacturing of a new product is one of the challenging topics in today's manufacturing and prototyping industry. RE is one of the technologies that can meet the demands of reducing development time.

The typical process of RE begin with data capturing which aims at acquiring sample point coordinates from the surface of a physical object. Then usually a data segmentation process is applied which divides the acquired cloud data into several smooth regions for surface fitting purpose according to the similarity of the property of the cloud data point. Finally surface modelling methods are used to generate the geometric models such as CAD model for the cloud data.

2.1 Modeling of the Surface

After capturing the data points of the physical object's surface, it is necessary to reconstruct a computerized model that can be used for the subsequent manufacturing or rapid production, which is the work of surface modelling.

According to the property of the obtained data points, the modelling approaches used in surface reconstruction can be divided into two classes: interpolation (in which method, cloud data lie on the corner of triangles or rectangles and give a continuous global surface) and approximation (in which method, surface traverses between the cloud data points, while maximal error bound is given by user). The interpolation method provides high performance for noiseless data. However, if the noise is significant, approximating the surface is preferable to interpolating it through the data points. And during surface approximating, if the object is composed of several free-form surface patches, it is difficult to obtain the equation that represents the whole data set, so data segmentation is usually used to group the data points into several regions in the first place. Then surface fitting approaches can be implemented to obtain the equation of each region and combine them together.

In the following sections, we will introduce the data segmentation, surface fitting and triangulation (which achieves the reconstructive surface by either interpolation or approximation) approaches individually.

2.1.1 Data segmentation

To divide the measurement data points into regions according to shape-change detection is the main purpose of data segmentation process, which generally has three categories of methods: edge-based, face-based and feature based data segmentation approaches (Milroy et al., 1997, Yang and Lee 1999, Jun et al., 2001).

The edge-based method is usually a two-stage approach that includes edge detection and linking. In the edge detection process, curvature is selected as the mathematical basis for edge detection and local surface curvature properties are used to identify boundaries present in the measured data. One popular method developed by Milroy et al. (1997) tries to find boundaries in the point data representing edges between surfaces. Hamann (1993) presented a method for curvature estimation from 3D meshes, and Kobbelt (2000) extracted curvatures from a locally fitted quadratic polynomial approximant. Yang and Lee (1999) extended the edge-based method by using parametric quadric surface approximation to identify the edge points. Woo and Kang (2002) used the Octree-based 3D-grid method to extract edge-neighbourhood points by subdivision of cells using the normal values of points. After edges are being sought, an edge-linking process usually follows to connect the disjoint edge points to continuous edges.

In the face-based segmentation approaches (Chen and Schmitt 1994, Peng and Loftus 1998), a group of points are firstly classified into a distinctive region according to similar geometrical properties, such as normal vectors or curvatures, and each region is then transformed into appropriate surfaces using a region-growing algorithm. Finally the exact boundary edges can be derived by intersection or other computations from surfaces. In these methods, triangulated meshes are firstly generated from input scanned cloud points. Furthermore the cost value of each polygonal mesh is computed by the area and the normal of each polygonal and the cost value for each edge from these triangles is taken as reference value. Then all the meshes whose cost value is higher than a certain level are selected to be boundary meshes. At last, a region-growing process is imposed to aggregate a polygonal mesh into sub-region until the area of the sub-region reaches the user-defined area criterion.

In the feature-based segmentation method (Jun et al., 2001), intelligent algorithms, such as artificial neural network or genetic algorithm, are used to extract or reconstruct geometric features directly from scanned point set.

2.1.2 Surface fitting

Surface fitting techniques are used to obtain surface equations from point data that belong to each particular region obtained from the segmentation process. Then all the equations are combined to a presentation for the whole cloud data set.

A surface fitting problem can be defined as follows: let D be a domain in the x-y plane, and suppose F is a real valued function defined on D. Suppose we know the values F(x_i, y_i) (i = 1, 2,…, N) located in D. Find a function f defined on D which reasonably approximates F. Polyhedral and curved surface are two commonly used geometric models in surface fitting.

Curved surface fitting methods may be classified into three types: algebraic, parametric, and dual. In algebraic surface fitting, the surfaces are approximated using polynomial equations, and there are two approaches for algebraic surface fitting (Menq and Chen 1996). In parametric surface fitting (Chivate and Jablokow 1993), parametric functions are applied to fit appropriate surface to the patches of data. In general the algebraic surfaces have infinite domain while parametric surfaces are bounded. In dual surface fitting, the representation of surface combines both algebraic and parametric surfaces. However, most methods assume the surface has simple topological structure or require user intervention to build the patch work. Many surface fitting algorithms, such as quadratic surface fitting, B-spline surface fitting, rotational surface fitting and lofted surface fitting etc., have been discussed (Weir et al 1996, Piegl and Richard 1995, Ueng et al., 1998).

2.1.3 Triangulation

Triangulation involves the reconstruction of the target surface by creating the triangular meshes that approximate or interpolate the point cloud. There are generally three triangulation approaches: sculpting-based, region-growing, and contour-tracing.

In sculpting-based approach, Delaunay triangulation or Voronoi diagram is first constructed from a set of cloud points, and then it extracts a collection of triangles or triangular patches representing the target object surface. Boissonat (1984) applies Delaunay triangulation to the points' convex hull. And then, redundant tetrahedron is removed and at the same time the manifoldness of the exterior surface is preserved. Fang and Piegl (1995) proposed a 3D Delaunay triangulation with implementation steps. Cignoni et al. (1998) described a Delaunay triangulation based on a divide and conquer paradigm. Some geometrical measures are predefined to make clear the order of the tetrahedral to be eliminated. Amenta and Bern (1998) removed triangles from the Delaunay triangulation using Voronoi filtering. Later, an approach based on medial axes transform was proposed by Amenta et al. (2001). Dey et al. (2001) proposed another Delaunay-based approach to reconstruct surface from large-scale data.

When dealing with a set of unorganized points with the absence of geometric information, the sculpting-based method is more systematic and robust because of the structural characteristics of the Delaunay triangulation and the Voronoi diagram. However, the computation is quite time-consuming, because (1) the Delaunay-based method needs to construct not only the Delaunay/ Voronoi diagram, it also needs multiple Delaunay computation and (2) The extraction process based on Delaunay/Voronoi structure is complex.

In region-growing approach, it begins with a seed triangle patch, and then new triangles are progressively added to the patch boundary according a series of rules. Lawson (1977) used geometric reasoning to construct a triangular facet mesh. Choi et al. (1988) developed a signed distance function by estimating the local tangential plane and using a marching cube method to extract a triangular polyhedral mesh. Sun et al. (2001) generated triangulated mesh from the data sorted by Euclidean distance.

If we directly use the region-growing method, it is extremely faster compared with the sculpting-based method. But when the cloud data set is obtained by merging multiple range images digitized from the original object, there must be some redundant data points in the data set, which will lead to overlapping or forming some bad triangles. And usually the data set obtained from laser scanner is highly dense and has a lot of noise; directly implementing this kind of data to form meshed surface not only consumes too much time but also results in forming uneven meshed surface because the influence of noise on the facet is big referring to the sample density.

In contouring-tracing approaches (Hoppe et al. 1992, Curless and Levoy 1996, Bernardini et al.1997, Boissonnat and Cazals 2000), the surface constructed does not necessarily interpolate the cloud points; instead, it traverses between the points. A typical example is that of (Hoppe et al. 1992). There are several steps in this method:

(1) A tangent plane is set up by fitting the neighbor point of each sample point.

(2) An orientation algorithm is used to guarantee that the normal of tangent plane propagate consistently through the smooth, low curvature region.

(3) A signed distance function is defined and computed from the sample points.

(4) A zero-set of the singed distance function is used to extract a triangular iso-surface.

In this category of approaches, for the surface reconstructed approximates rather than interpolates the sample points, it limits its applications only to computer graphic and virtual reality. While in CAD/CAM and rapid manufacturing, the accuracy of the reconstructed model usual is more preferable.

2.2 The Proposed Hybrid Digitization Method

The surface reconstruction approach adopted in our hybrid digitisation method is a region-growing method for well-distributed scattered data, which is obtained by a filtering process firstly implemented on the laser scanned cloud data. The filtering process not only thins the cloud data set according to a user input tolerance bound, which greatly reduces the computation time during triangulation, but also improves the reconstruction surface by reducing the overlapping and uneven facets. This is because it cuts off most of the redundant data and reduces the sampling density, which reduces the influence of noise on the rebuilt surface.

Due to the problem that there are usually some loss of important feature information (edge or holes) during the data capturing process (some sharp edges may not be correctly captured) and data thinning process (during the data thinning, some edge or boundary information may be filtered), a probe-path planning algorithm is designed for the CMM for re-acquisition of these feature points.

The overall procedure in our hybrid digitization method includes three steps:

(1) Data thinning: Patch surface fitting of the image data, curvature estimation of the surface patches, filtering the cloud data according to the property of the most curved surface patch.

(2) 3D modeling: modeling the filtered image data by a region-based triangulation method and obtaining the key features of the 3D model (holes, sharp edges and boundaries) at the same time.

(3) Feature points re-digitization probe-path planning: designing an algorithm to generate the probe path for a CMM to recapture the key features of the target object according to the information obtained in the 3D modeling process.

CHAPTER 3

CLOUD DATA THINNING

In this chapter, we extend the voxel binning method (Sun et al. 2001) to compress the cloud data acquired from the target object surface by a laser scanner that is merged from different scanning directions, overlapped at some areas and unstructured. The outline of the voxel binning method is presented as follows:

(1) Compute bounding box for the whole cloud data set and offset the bounding box by a point coincidence tolerance (*T*):

$$\begin{pmatrix} x_{\min} \\ x_{\max} \\ y_{\min} \\ y_{\max} \\ z_{\min} \\ z_{\max} \end{pmatrix} = \begin{pmatrix} x_{\min} \\ x_{\max} \\ y_{\min} \\ y_{\max} \\ z_{\min} \\ z_{\max} \end{pmatrix} + \begin{pmatrix} -T \\ T \\ -T \\ T \\ -T \\ T \end{pmatrix}$$

(2) Sample *N* points from the most curved area of the cloud data and use a least squares fitting method to interpolate a local parabolic quadric surface patch to the *N* points.

(3) Calculate the maximum second partial derivatives from the obtained quadric function coefficients, which are used to calculate the maximum triangle edge length (Ω) for the mesh process.

(4) Determine the maximum triangle edge length space Ω, from the derivative values and ε, the desired error tolerance between the original cloud data and formed triangulated surface.

(5) Compute the bin size (B), with the respect to ε.

(6) Use voxel binning method to thin cloud data with the bin size B .

The thinned data set is then used to construct the final triangular polyhedral mesh. The details of the key steps for obtaining the bin size are presented in the following sections.

3.1 Surface Fitting Error Analysis

To judge the accuracy for a surface representation to the underlying data points, the geometric definition of the surface error can be formulated as (see Fig. 3.1): given a parametric C^2 surface patch S , which interpolates the given cloud points and an arbitrary triangle **T** with its vertices on the parametric surface, determine the maximal distance between the surface and triangle within the same parametric bound, if the distance is smaller than a user-specified tolerance ε shown in Eq. (3.1), the patch is said to be satisfactory.

$$\sup_{(u,v)\in T} \left\| S(u,v) - T(u,v) \right\| \le \varepsilon \tag{3.1}$$

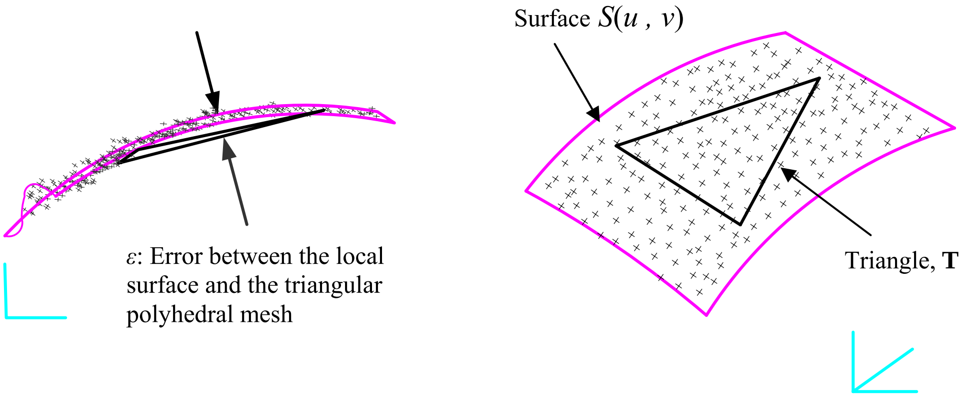

ε: Error between the local surface and the triangular polyhedral mesh
Surface $S(u , v)$
Triangle, **T**

Figure 3.1. Error between the surface representation and underlying points

3.2 Maximum Edge Length Calculation

The maximum partial derivatives of $S(u, v)$ are used to conjunct with the surface fitting error tolerance ε and then to determine the maximum triangle edge length Ω. As defined, for surface $S(u, v)$ interpolating to point set $\mathbf{V}$ and an arbitrary linear triangle $\mathbf{T}$ with vertices $(A, B, C) \in \mathbf{V}$, the deviation between the triangular facet and the surface patch satisfies,

$$\sup_{(u,v)\in T} \left\| S(u,v) - T(u,v) \right\| \le \frac{1}{8}\Omega^2 (M_1 + 2M_2 + M_3) \tag{3.2}$$

Where

$$M_1 = \sup_{(u,v)\in T} \left\| \frac{\partial^2 S(u,v)}{\partial u^2} \right\| \tag{3.3}$$

$$M_2 = \sup_{(u,v)\in T} \left\| \frac{\partial^2 S(u,v)}{\partial u \partial v} \right\| \tag{3.4}$$

$$M_3 = \sup_{(u,v)\in T} \left\| \frac{\partial^2 S(u,v)}{\partial v^2} \right\| \tag{3.5}$$

Combining equations (3.1) to (3.5), the relation between the maximum edge length Ω and the surface fitting tolerance ε is obtained as follows:

$$\Omega = 2\sqrt{\frac{2\varepsilon}{\left(M_1 + 2M_2 + M_3\right)}} \tag{3.6}$$

3.3 Least Square Quadric Surface Fitting

The maximum values of the second partial derivatives M_1, M_2 and M_3, are estimated from a second order quadric surface equation given by:

$$\vec{S}(u,v) = \vec{a}u^2 + \vec{b}uv + \vec{c}v^2 \tag{3.7}$$

The above surface approximation equation is set up by interpolating to a group of neighbour points **P** sampled in the locality of the central point P_i. A small set ranged from 24 to 32 points was experimentally found to be an effective trade-off between computation time and sufficient sampled data which can gain a reasonable interpolation to the underlying surface form.

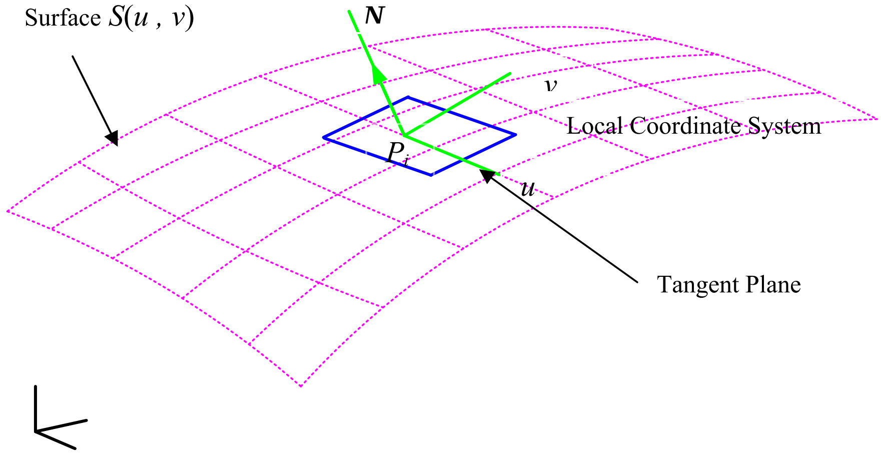


Figure 3.2. Illustration of quadric surface interpolation

As shown in Fig. 3.2, the local coordinate system employed for the quadric surface is defined as that P_i and the **S** axis aligned with the local surface normal ***N*** and the surface normal is obtained from an initial planar fit to the same local set of cloud data points. By definition, the gradient at the local origin, with respect to the parameters u and v, is zero and is, hence, a minimum. The direction of u and v axes can be assumed arbitrarily in the plane perpendicular to ***N***. The coefficients $\left(\vec{a},\vec{b},\vec{c}\right)$ of function ***S*** can then be determined from a least squared fit to local data (Ferrie, et al. 1993),

$$X = [A^T A]^{-1} A^T B \tag{3.8}$$

Where the function parameters are collected in vector X,

$$X = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \dfrac{\partial^2 S}{\partial u^2} \\ \dfrac{\partial^2 S}{\partial u \partial v} \\ \dfrac{\partial^2 S}{\partial v^2} \end{bmatrix} \tag{3.9}$$

and the vectors A and B are given by,

$$A = \begin{bmatrix} u_1^2 & u_1 v_1 & v_1^2 \\ u_2^2 & u_2 v_2 & v_2^2 \\ \vdots & \vdots & \vdots \\ u_n^2 & u_n v_n & v_n^2 \end{bmatrix} \tag{3.10}$$

$$B = \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_n \end{bmatrix} \tag{3.11}$$

3.4 The Voxel Bin Thinning Method

The process of voxel binning can be described as follows:

(1) The entire volume is subdivided so that a cloud data set occupies into smaller cubic volumes called voxels.

(2) For each bin, a single data point closest to the centre is selected to represent all other data points in the cubic volume. The remaining points in the cube are deleted.

As a result of above data reduction process, a new uniform set of points distributes across the object surface. Meshing these points forms a triangular faceted model. The maximum triangle edge length can then be estimated. In the following section, we show how to determine the maximum bin size based on the maximum triangle edge length.

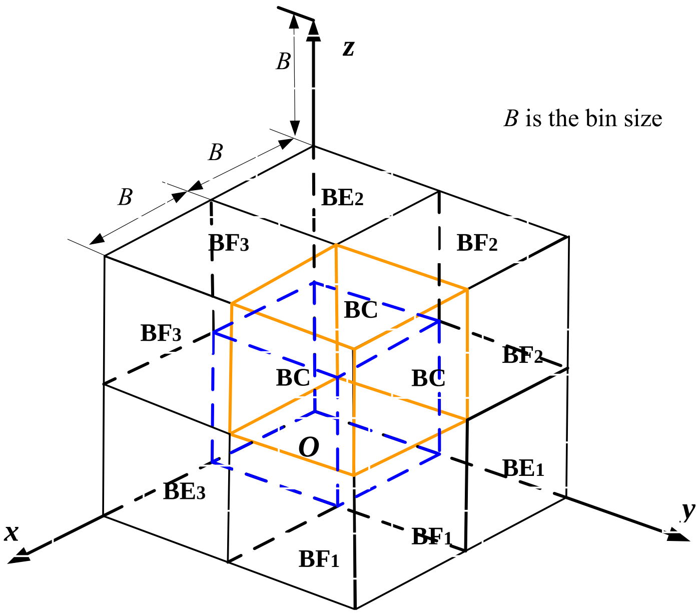


Figure 3.3. Illustration of 26 adjoining bins

As seen in Fig. 3.3, each bin has 26 adjoining bins, each containing a single point. The dimension of each bin is *B*, which has 8 vertices, 12 edges, and 6 faces. With reference to the diagram, there are 8 bins, e.g. the bin (in blue), which has one point contact with the central bin **BC** (in orange). There are a further 12 bins, **BE,** that each has contact along one edge with **BC** and another 6 bins, **BF**, that contact across a full bin face with **BC**. The initial cloud data is very dense, so an assumption could be made that the surface passes through two neighbouring bins in one of the following three possibilities:

(1) It passes through bins which share a common face (BF1-3).

(2) A triangle edge passes obliquely through two neighbouring cells (i.e., at 45°) that share a common edge (BE1-3).

(3) A triangle edge passes through two bins sharing one point (BP1).

Based on the diagram and the above assumption, the maximum point to point distance in the neighbouring cells which is the maximum triangle edge length $\Omega_{\max}$ is given by,

$$\Omega_{\max} = l_{\max} = \sqrt{3}B \tag{3.12}$$

A triangle edge length (Ω) should be less than $\Omega_{\max}$. Therefore, B should satisfy the following:

$$B \leq \frac{\sqrt{3}}{3}\Omega \tag{3.13}$$

Combining Eqs. (3.6) and (3.13), we have,

$$B \leq \frac{2}{3}\sqrt{\frac{6\varepsilon}{(M_1 + 2M_2 + M_3)}} \tag{3.14}$$

Where M_1, M_2 and M_3 are the maximum partial derivative values of the locally interpolated surface. Sun et al. (2001) proposed to select several (*n*) areas on the screen interactively and the corresponding $B_i (i = 1,2,\dots,n)$ is calculated. The final bin size applied to thin the cloud data is chosen as $Min(B_i)$.

After the bin size B is obtained, the cloud data volume is subdivided into a set of small bin with the respect to *B*. Only one point is retained in each bin if the bin has points in it, which is the nearest to the centre of the bin. After the data thinning procedure, the remaining data set is more uniformly distributed and called a simplified data set.

3.5 Implementation

The overall procedure of the cloud data thinning process includes three steps:

(1) Surface fitting of the sampled patches.

(2) Curvature estimation of the surface patches.

(3) Filtering the cloud data according to the property of the surface patch with the maximum curvature.

Compared with the method of Sun et al. (2001), we add the curvature estimation algorithm in step (2). In this way, it can guarantee that the highest curved area is identified automatically. After patch surface fitting of each point of the data set, a curvature estimation algorithm is applied to find the most curved area that can be used to calculate the bin size. The whole process of surface fitting in step one is illustrated as followed:

- Find a point P_i from the cloud data set.
- Find *N* nearest neighbour points of P_i.
- Take P_i as origin and approximate the set of *N* points using the least square surface fitting method.
- Set local coordinate which take P_i as origin and the normal of tangent plane as one axis.
- Transform the *N* neighbor points to the local coordinate.
- Do patch surface fitting of the *N* neighbor points.

3.5.1 Tangent plane estimation

Now we present the process of tangent plane estimation. The problem is how to find a plane (defined in Eq. 3.15) passing through the point $P_0(x_0, y_0, z_0)$ to fit the *N* neighbour points $\mathbf{P}\{P_n(x_n, y_n, z_n)\}$ of P_0.

$$(z - z_0) = a(x - x_0) + b(y - y_0) \tag{3.15}$$

Rearrange the Eq. (3.15) in the following form,

$$AX = B \tag{3.16}$$

Then according the least square surface fitting theory, we obtain the solution,

$$X = \left[A^T A\right]^{-1} A^T B \tag{3.17}$$

Where,

$$A = \begin{pmatrix} x_1 - x_0 & y_1 - y_0 \\ x_2 - x_0 & y_2 - y_0 \\ \vdots & \vdots \\ x_n - x_0 & y_n - y_0 \end{pmatrix} \tag{3.18}$$

$$B = \begin{pmatrix} z_1 - z_0 \\ z_2 - z_0 \\ \vdots \\ z_n - z_0 \end{pmatrix} \tag{3.19}$$

$$X = \begin{pmatrix} a \\ b \end{pmatrix} \tag{3.20}$$

And then the unit normal vector for the tangent plane is calculated as

$$\vec{n} = \frac{1}{\sqrt{a^2 + b^2 + 1}} \begin{bmatrix} -a & -b & 1 \end{bmatrix} \tag{3.21}$$

3.5.2 Surface patch fitting

For local surface fitting, we select N neighbouring points from the cloud data that are nearest to the given point P_i. N is chosen by a value between 25 to32. From the Eqs. (3.7)-(3.11), we obtain the local surface patch fitting equation.

3.6 Some Examples

In this section, two simulated examples are presented to illustrate the efficacy of the thinning algorithm. The examples are based on simulated data sets in which the original cloud data are generated by mathematical equations, so that the theoretical curvature can be obtained accurately and comparison can be made directly.

3.6.1 Example 1

In this case study, a hemisphere is selected by taking the advantage of its known geometry so that the curvature of approximated surface patch can be compared with the theoretical one easily. As shown in Fig. 3.4, a hemisphere with a radius of r is firstly discretized into a number of sampled points as (note random error δ is incorporated to simulate noise in the point cloud):

$$x = r\cos(\phi)\sin(\theta) + \delta \qquad \phi = [0, \pi/2]$$

$$y = r\cos(\phi)\sin(\theta) + \delta \qquad \theta = [0, 2\pi]$$

$$z = r\sin(\phi)$$

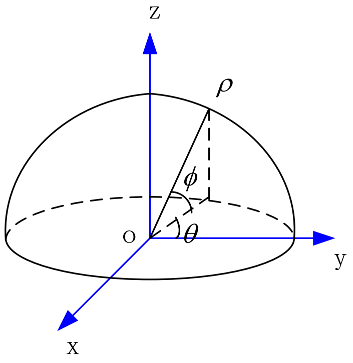


Figure 3.4. Illustration of the model of a hemisphere

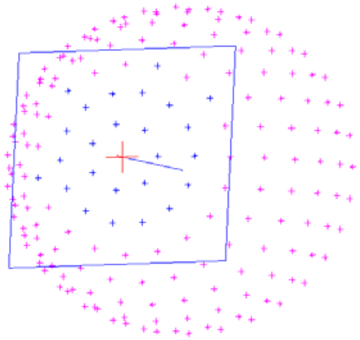


Figure 3.5. Sampling of a hemisphere

We use a sampling tolerance of 0.005 mm, $\Delta\phi$ = 11.46° and r = 1 mm to obtain the point cloud of the hemisphere. There are 188 points generated, and the original sampling points (pink) are shown in Fig. 3.5. The points in blue are the neighbouring points of the red point. The plane is the estimated tangent plane and the blue line is the normal of the tangent plane. We used two methods to select the neighbouring points for a given point, minimum distance method (find *N* nearest points to the selected point and take the selected point as origin to do coordinate transformation) and improved minimum distance method (find *N* nearest points to the selected point and take the point nearest to the centre of gravity of the *N* neighbouring points as origin to implement coordinate transformation). The error of surface fitting is calculated by,

$$Error = \sum_{i=1}^{n} \left| T_{iz} - S_i(T_{ix}, T_{iy}) \right| / N \tag{3.22}$$

Where n=1~N**,** T_{iz} is the Z value in the local coordinate for the i^{th} neighbouring point and S_i is the corresponding Z value of the approximated surface for the given point.

The result of surface fitting by 25 neighbouring points (see Fig. 3.6) for the points lying from near the centre to near the boundary was shown in Fig. 3.7a and 3.7b, which are the corresponding results using nearest distance and improved nearest distance rules to select neighbouring points. As we can see from the picture, first, the surface fitting error using the improved minimum distance criteria is smaller than the one using the minimum distance criteria. Second, the surface fitting error for the point near the centre is smaller than the surface fitting error for the point near the boundary.

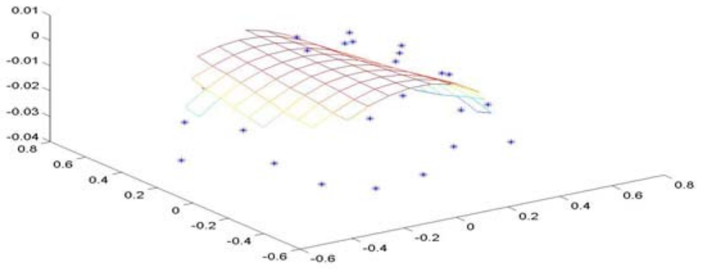
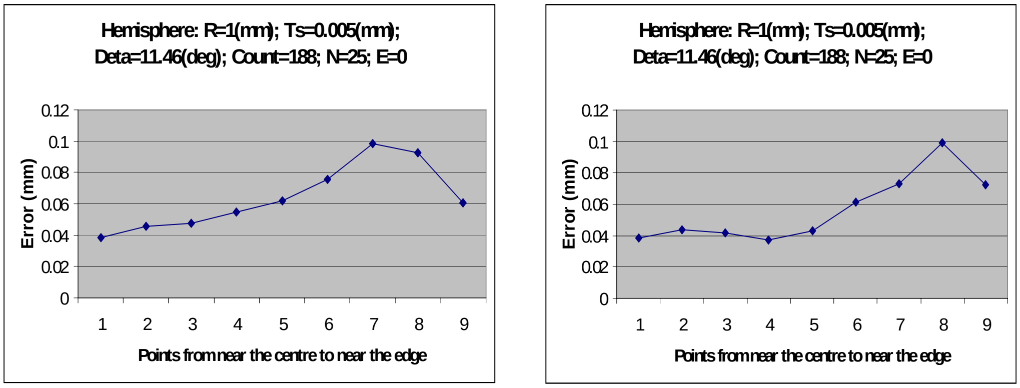


Figure 3.6. Surface fitting by 25 neighbouring points



(a) Fitting error for using nearest distance criteria to find neighbor point

(b) Fitting error for using improved nearest distance criteria to find neighbor point

Figure 3.7 Comparison of fitting error according to different criteria

The results for curvature estimation for the same data set with the change of the number of neighbouring points and the input sampling error are shown in Figs. 3.8 and 3.9 respectively. As seen in Fig. 3.8, choosing 25 neighbouring points can gain a better approximated surface so that the curvature of the fitted surface (series 2) by choosing 25 neighbouring points is closer to the theoretical value than that of approximated surface by choosing 15 and 35 neighbouring points (series 1 and series 3 respectively). And as for noise standing analysis, we can see the input sampling error has little effect on the surface fitting results (Fig. 3.9) in which series 2 represents the curvature of the fitted surface with random input error less than 0.025 mm and series 1 represents that of the fitted surface of the same data set with zero sampling error.

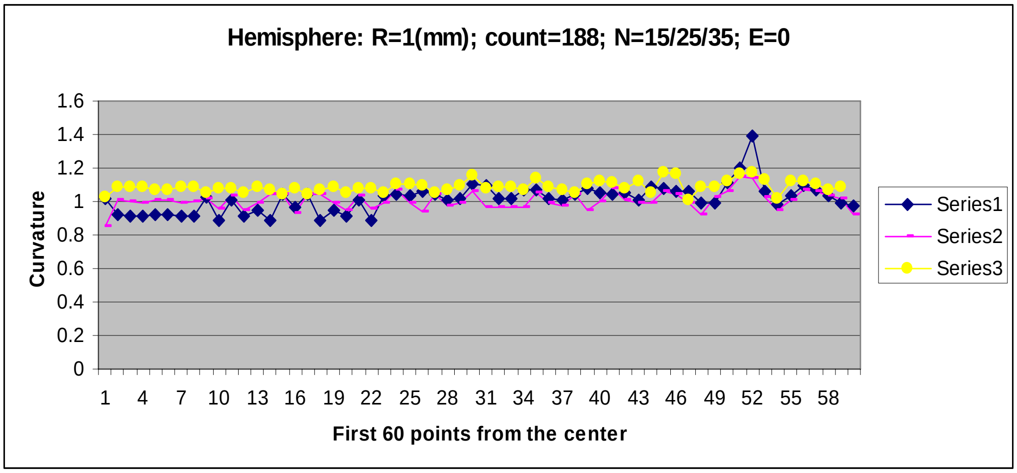


Figure 3.8. Curvature estimation with different number of neighbouring points

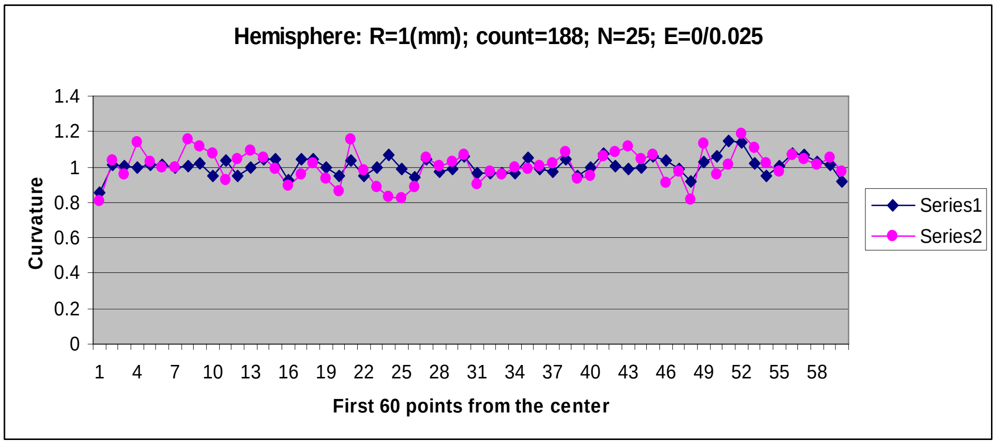
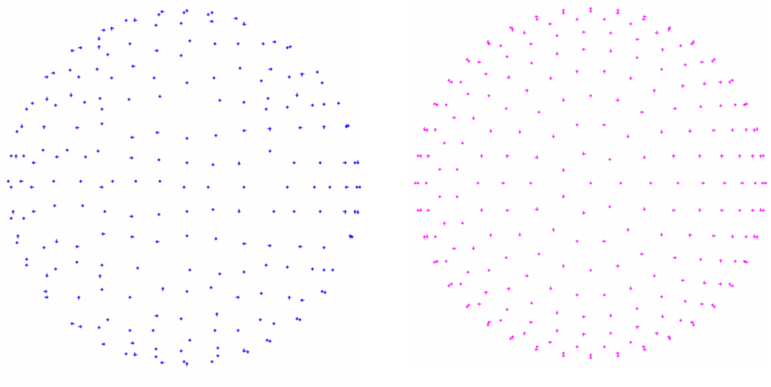


Figure 3.9 Curvature estimation analyses with change of sampling noise

As we can see from above figures and analysis, using a local parabolic surface to interpolate a set of 25 neighbouring points can achieve a fairly good fitting quality when the location of points is in fixed grid format (uniformly sampling) and it has good feature to stand off noise. Using improved minimum distance criteria to find neighbouring points can better surface fitting results.

The result for data thinning of a hemisphere with a radius of 8 mm is shown in Fig. 3.10. First, we sample the hemisphere with a tolerance of 0.005 mm. There are totally 1,376 points generated. Using a thinning tolerance of 0.025mm, a bin size of 1.2 mm is obtained. The data set after thinning has 223 points (see Fig. 3.10a). For comparison, we also directly discretize the hemisphere using a tolerance of 0.025 mm, which generates 292 points (see Fig. 3.10b). There is a fairly good agreement between these two set of data, which suggests that the algorithm for bin size estimation produces fairly good results.



(a) (b)

Figure 3.10. Validation of the algorithm for bin size estimation

3.6.2 Example 2

In the second case study, a cone is used (see Fig. 3.11), whose discrete expression is based on the following basic form: $x = x$, $y = y$, and $z = h - {h\sqrt{x^2 + y^2}}\big/{r}$, where r is the radius and h is the height.

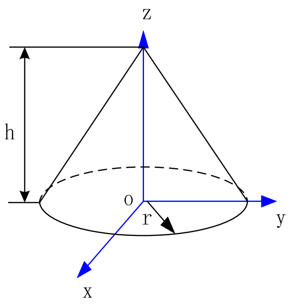


Figure 3.11. Illustration of the model of a cone

We use sampling increments Δx, Δy of 0.05 to uniformly sample x and y to obtain the cloud data of the cone with r = 1 and h = 1. There are altogether 1,247 points obtained as shown in Fig. 3.12.

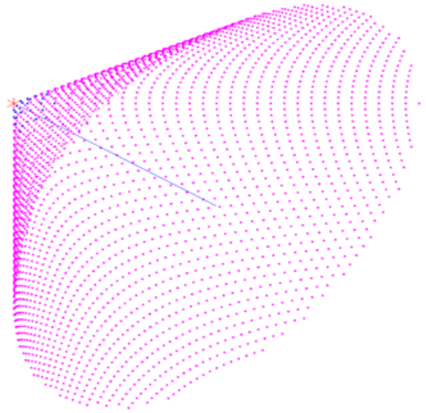


Figure 3.12. Uniformly sampling data points of cone

The curvature estimation result is shown in Fig. 3.13. The maximum curvature of the cloud data set equals to15.25, locating at the point represented by the red star seen in Fig.3.12. The blue points represent the neighbouring points of the red point that is used for fitting with a quadric surface. The bin size calculated for each point with a tolerance of 0.02 is shown in Fig. 3.14. The minimum bin size of the data set equals to 0.103 that locates at the same point with the curvature maximum.

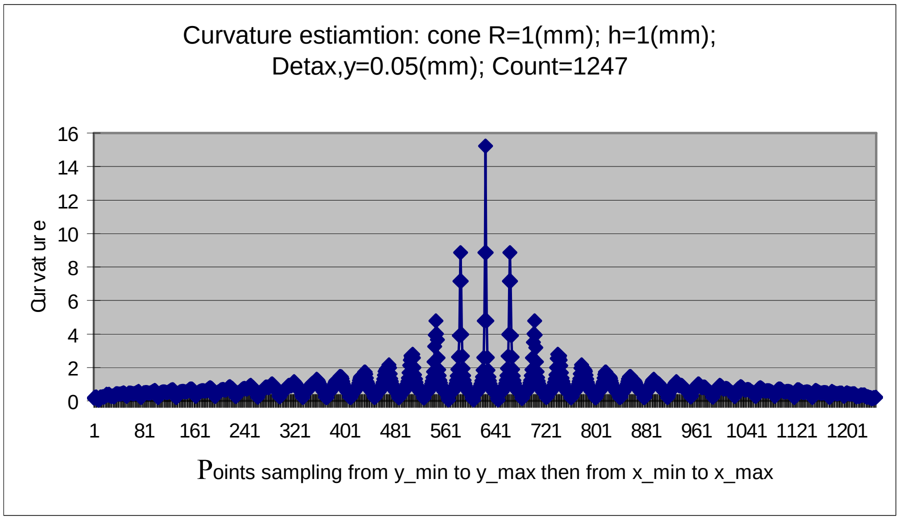


Figure 3.13. Results of curvature estimation for the cone

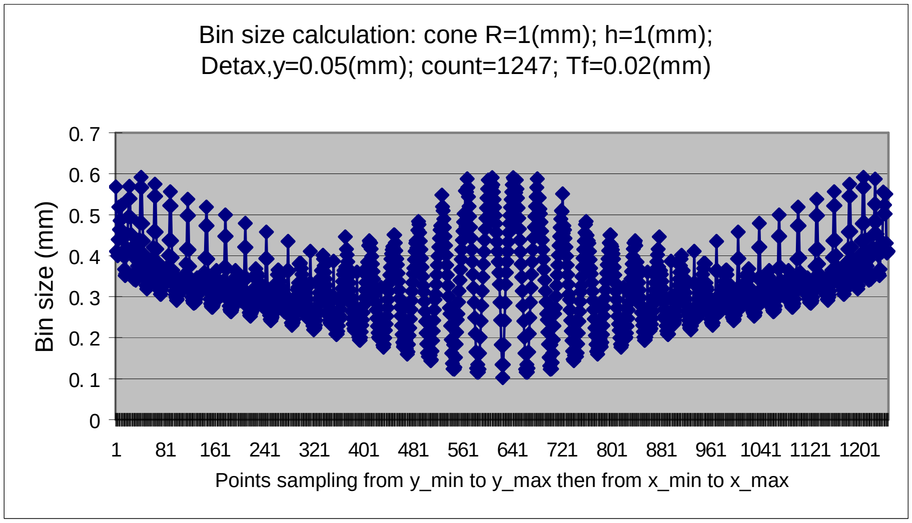


Figure 3.14. Results of bin size calculation for the cone

In summary, these two case studies show that the results from the surface fitting algorithm, curvature estimation algorithm, and bin size estimation algorithm have a high degree of agreement with the accurate results obtained mathematically.

CHAPTER 4

TRIANGULATION

After data thinning, we get a nearly uniformly distributed data set. A triangulation process is then applied to represent the object. Furthermore, the important features such as edges are to be extracted from the cloud data set so that the subsequent probe-path planning process can be performed based on this information as a reference to capture the real edge.

The triangulation method adopted here is based on a region-growing method developed by Sun et al. [2001] and Shi [2001]. In this method, the patch boundaries, sharp edges formed by two patches, and interior holes can be successfully identified.

The triangulation algorithm begins with a randomly selected point to form a seed triangle by connecting two other neighbouring points. Based on a set of heuristic rules, a new point (suitable point) is sorted to add a new triangle to expand the mesh by connecting the current edge of the seed triangle to the suitable point. The mesh grows by interrogating the geometric and topological information in the current triangle and sorting suitable point to form a new triangle. The mesh thus expands outwards from the seed triangle to cover the whole surface.

In this process, each boundary edge of the triangulated area is used to search for an external suitable point to form a new triangle unless it encounters the boundary of the cloud data, i.e., sharp edge or a hole. The triangulation process is repeated until all points are meshed. In this way, a complex surface can be well triangulated and edges and holes can be successfully identified at the same time.

4.1 Basic Definitions and Data Structures

Some terms used in the triangulation algorithm are defined as follows:

a) Seed triangle, starting point, ending point and normal vector

A seed triangle is defined by three initially selected neighbouring points. Each edge of the triangle is defined as an edge vector directed from one vertex (*starting point*) to the other vertex (*ending point*). The edges of triangle follow a counter-clockwise direction and each triangle has a *normal vector* pointing outside the surface according to the right-hand rule.

b) Loop

In the meshing process, the loop edges of the meshed area follow the counter-clockwise direction if it is an *external loop* and while the loop edges follow a clockwise direction if it is an *internal loop* (interior hole).

c) Triangular facet

A triangle obtained from the meshing process is not only a geometric triangle but also a *triangular facet*. The information of vertices, edges and adjacent of the facet are stored during meshing.

The data structures of the point of the cloud data and triangulated facet are defined as follows:

a) The data structure of a point

```
typedef struct SpointStruct
{
        double pt[3];  /*the coordinate  x, y, z of the cloud data point*/
        double  cntr[3]; /* the center of the bin in which the point resides in */
        int       next;    /* 0 <= next <= count */
        int       prev;     /* 0 <= prev <= count */
        int      where;  /* if the point is on boundary, or external, or internal */
        double nml[3];  /* initial value [0 0 0] */
        int      patch_bndry;   /* initial value = NO */
}Spoint;
```

b) The data structure of a triangulated facet

```
typedef struct SfacetStruct
{
        double v0_pt[3], v1_pt[3], v2_pt[3];
        int id;
}Sfacet;
```

4.2 Rules for Forming the Seed Triangle and Sorting Suitable Point

In this section, the heuristic rules with which the first triangle is built and the whole triangulated mesh process are described.

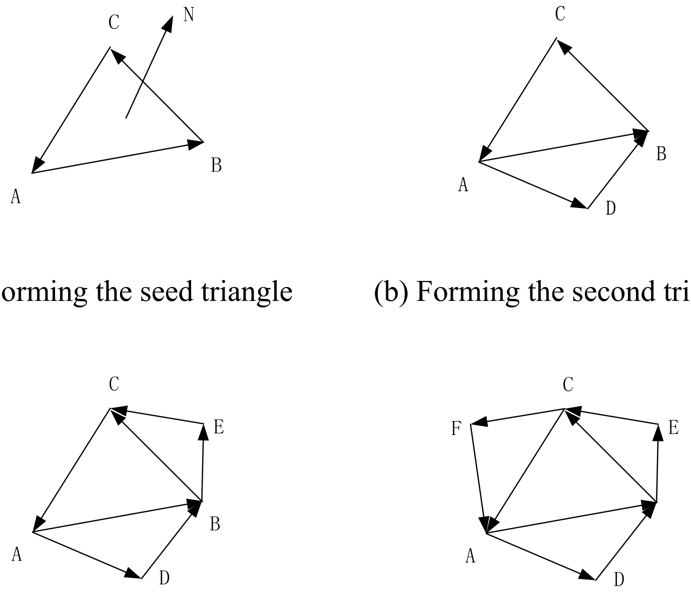
4.2.1 Forming the seed triangle and further triangulation

An upside of triangulated meshed surface is defined according to the right hand rule and the seed triangle is set up in counter-clockwise direction. Because the cloud data can be quite dense and the point lying in the front and back may overlap, it is not easy to select three neighbouring points manually on the screen. So to keep the seed triangle well shaped and no points inside it, we firstly randomly select one point manually by zooming in the point cloud, then sort the other two points in the cloud data automatically to form the seed triangle according to the following rules:

(1) Searching in the cloud data to find the nearest point to the selected point by the distance in E space.

(2) Searching in the cloud data to find the third point to form the seed triangle, which has the nearest distance to the centre of the line joined by the first and second point and formed seed triangle must have a minimum angle $\theta \geq 10^\circ \sim 12^\circ$.

As seen in Fig. 4.1a, the sequence of the three points of forming the seed triangle is **A→ B→ C**. The edges of the triangle are defined as vectors that follow the counter-clockwise direction and the normal of the triangle points to the upside of the surface patch. Every edge vector has a starting and ending point. For instance, edge $\overrightarrow{AB}$ has starting point **A** and the ending point **B**.



(a) Forming the seed triangle (b) Forming the second triangle

(c) Forming the third triangle (d) Forming the fourth triangle

Figure 4.1. Forming the seed triangle and further triangulation

To expand the mesh, each boundary edge is used to search for a point in the whole data set to form a new triangle. New boundary edges are formed for the meshed area following the counter-clockwise direction. As shown in Fig. 4.1, a detailed triangulation process is described as follows:

(1) As shown in Fig. 4.1b, to form a new triangle, edge $\overrightarrow{AB}$ is used to find an external point **D** according to a set of rules described in section 4.2.2. Then the triangle ΔADB is formed, which follows the direction of **A→D→B** and the new loop, $\overrightarrow{AD}\ \overrightarrow{DB}\ \overrightarrow{BC}\ \overrightarrow{CA}$ follows the counter-clockwise direction too.

(2) To continued, the vector edge $\overrightarrow{BC}$ is then used to find another external point **E** to form a new triangle following the direction of **B→E→C**, shown in Fig. 4.1c. Thus the obtained loop, $\overrightarrow{AD}\ \overrightarrow{DB}\ \ \overrightarrow{BE}\ \overrightarrow{EC}\ \overrightarrow{CA}$, follows the counter-clockwise direction.

(3) In Fig. 4.1d, edge $\overrightarrow{CA}$ is used to find a point **F** to form a new triangle that follows the direction of **C→F→A**. The external loop $\overrightarrow{AD}\ \ \overrightarrow{DB}\ \ \overrightarrow{BE}\ \overrightarrow{EC}\ \overrightarrow{CF}\ \overrightarrow{FA}$ still follows the counter-clockwise direction.

In this way, the triangulation mesh grows and spreads on the surface.

4.2.2 Rules for suitable point sorting

In the triangulation process, only one point satisfying all the rules described below is chosen to form a new triangle. We name this point as *suitable point* and the loop edge currently used in the search as *searching edge*.

During the triangulation process, the newly identified point (*suitable point*) for forming a new triangle must satisfy the following constraints:

(1) Search scope

After data thinning, the data points are almost evenly arranged and every bin retains at most one point (some bins may not contain any points). Theoretically, except for encountering the object boundary, sharp edges or holes, every loop edge has a suitable point in the neighbourhood to form a new triangle. Here we define the neighbourhood as the 26 neighbouring bins of the end of the searching edge. The suitable point must be within the 26 neighbouring bins of either end point of the searching edge. In other words, either end point of the searching edge can be used to find a suitable point within the neighbouring 26 bins to form a newborn triangle unless the searching edge lies on the boundary, sharp edges or holes. Thus, the searching space is limited within the 26 neighbouring bins of either end point of the searching edge.

(2) Minimum triangular facet angle

To prevent nearly co-linear vertices from forming long and narrow triangular facets (bad facet), the smallest angle in each triangular facet must be larger than the minimum angle θ , experimentally, $\theta = 10^\circ \sim 12^\circ$.

(3) Maximum angle between two neighbouring facets

On a smooth surface patch, the data points left are smoothly and evenly arranged, the angle between any two neighbouring triangular facets must be within a specific scope. Normally, a newborn triangle may join several triangles within the meshed area. The angle between the normal of the newborn triangle and any joined triangle must be less than a specific value. Experimentally, when the specific value is chosen as 90°, a well triangulated mesh will be achieved for most free-form surfaces without sharp corners. For the cases that the triangulated mesh can’t be well formed, there are usually two problems in it, overlapping or failed to find the edges. One solution is to change the specific value. Enlarging the value will eliminate overlapping and a relative smaller value will help to find edges. This heuristic rule keeps the facet mesh growing smoothly and also could automatically identify the sharp edge of two patches.

(4) Closest to the middle of the edge

There may be more than one point satisfying all constraints mentioned above. But only one point should be selected to form the new triangle facet. Experimentally, choosing the one closest to the middle of the searching edge can achieve an evenly distributed triangular mesh.

4.3 The Algorithm

In every step of forming a triangle, only one point satisfying the constraints above is selected to form a new triangle. This point is called the *suitable point* and the edge used in the searching is called the *searching edge*. The location of a suitable point may fall in the following situation:

a) There is no suitable point. This means the mesh reach the boundary of the image (internal holes or external boundary).

b) The suitable point **D** is at the outside of the loop of meshed area, as shown in Fig. 4.1b, which is the most common situation in triangulation.

c) The suitable point is on the loop of meshed area and next to the searching edge. There are two situations:

(1) As shown in Fig. 4.2a, the suitable point next to the end point of the searching edge. The original external loop of is …→**A**→**B**→**C**→**D**→…, and there is no direct link between **A** and **C**. While point **C** is found to be the suitable point for the searching edge $\overrightarrow{AB}$, a new triangle ΔACB is formed. The resulted new external loop of meshed area becomes …→*A*→*C*→*D*→…, thus filling the gap between **A** and **C** in the original loop list.

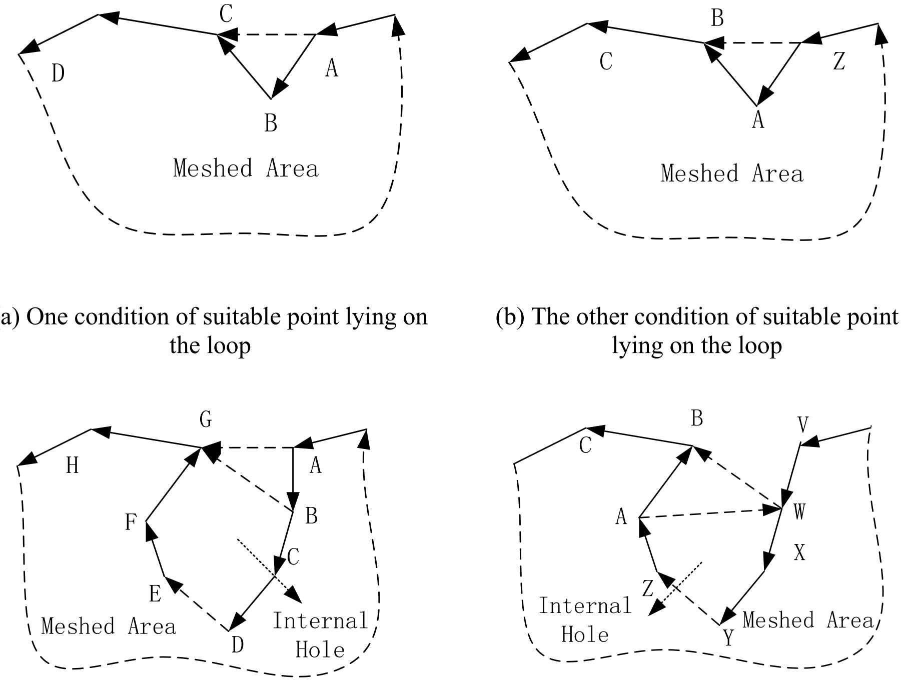
(2) Similarly, as seen in Fig. 4.2b, the suitable point is next to the starting point of the searching edge in the current loop of meshed area and the gap between **B** and **Z** in the original loop list is filled up.

d) The suitable point is on the loop, but not next to the searching edge in the current external loop. There are three situations:

(1) **G** to $\overrightarrow{AB}$, as seen in Fig. 4.2c. Now the current external edge loop is…→**A**→**B**→…→**F**→**G**→**H**→… and **G** is found to be the suitable point to searching edge $\overrightarrow{AB}$ .A new triangle ΔAGB is formed. So …→**A**→**G**→**H** becomes the counter-clockwise external loop. While, loop **B**→**C**→**D**→…→**E**→**F**→**G**→**B** falls into a clockwise direction, which is separated from the external loop and is called an internal loop. Actually, this is the interior hole off the surface patch.

(2) **W** to $\overrightarrow{AB}$, as shown in Fig. 4.2d. For this case, the original external is…→**V**→**W**→**X**→…→**A**→**B**→**C**→… and **W** is found to be the suitable point of searching edge $\overrightarrow{AB}$. A new triangle ΔAWB is formed. The vector $\overrightarrow{AW}$ forms a closed clockwise loop of **X**→**Y**→…→**Z**→**A**→**W**→**X**. This internal loop is thus separated from the external loop resulting in a hole. And new external loop becomes …→**V**→**W**→**B**→**C** →…. searching edge $\overrightarrow{AB}$ .A new triangle ΔAGB is formed. So …→**A**→**G**→**H** becomes the counter-clockwise external loop. While, loop **B**→**C**→**D**→…→**E**→**F**→**G**→**B** falls into a clockwise direction, which is separated from the external loop and is called an internal loop. Actually, this is the interior hole off the surface patch.

(3) **W** to $\overrightarrow{AB}$, as shown in Fig. 4.2d. For this case, the original external is…→**V**→**W**→**X**→…→**A**→**B**→**C**→… and **W** is found to be the suitable



(a) One condition of suitable point lying on the loop

(b) The other condition of suitable point lying on the loop

(c) One condition of suitable point lying on the interior hole

(d) The other condition of suitable point lying on the interior hole

Figure 4.2. The suitable point on the loop of meshed area

point of searching edge $\overrightarrow{AB}$. A new triangle ΔAWB is formed. The vector $\overrightarrow{AW}$ forms a closed clockwise loop of **X**→**Y**→…→**Z**→**A**→**W**→**X**. This internal loop is thus separated from the external loop resulting in a hole, and the new external loop becomes …→**V**→**W**→**B**→**C** →…. .

4.4 Case Study

The triangulation algorithm and a visualisation algorithm are implemented using VC++ and the OpenGL. Four cases are presented here to illustrate the results of data thinning introduced in Chapter 3 and triangulation presented in this chapter. The first one is a simulated case, and the other three are case studies of actual objects, whose cloud data points are obtained using the laser scanner, VIVID 900 system.

4.4.1 Case 1

In the first case, the surface patch used is defined by the following expression:

$$x = x$$

$$y = y \qquad a = h/r^2$$

$$z = a\sqrt{x^2 - y^2 + r^2} \qquad r, h \text{ is randomly input by user}$$

The uniformly discrete model with r of 2 mm, h of 3 mm and increment Δx , Δy of 0.04 mm is shown in Fig. 4.3. There are totally 7,837 points generated.

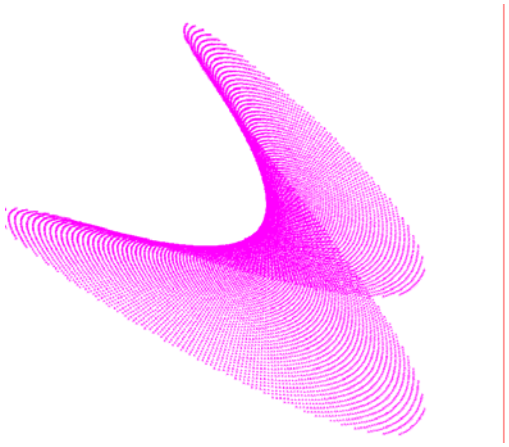
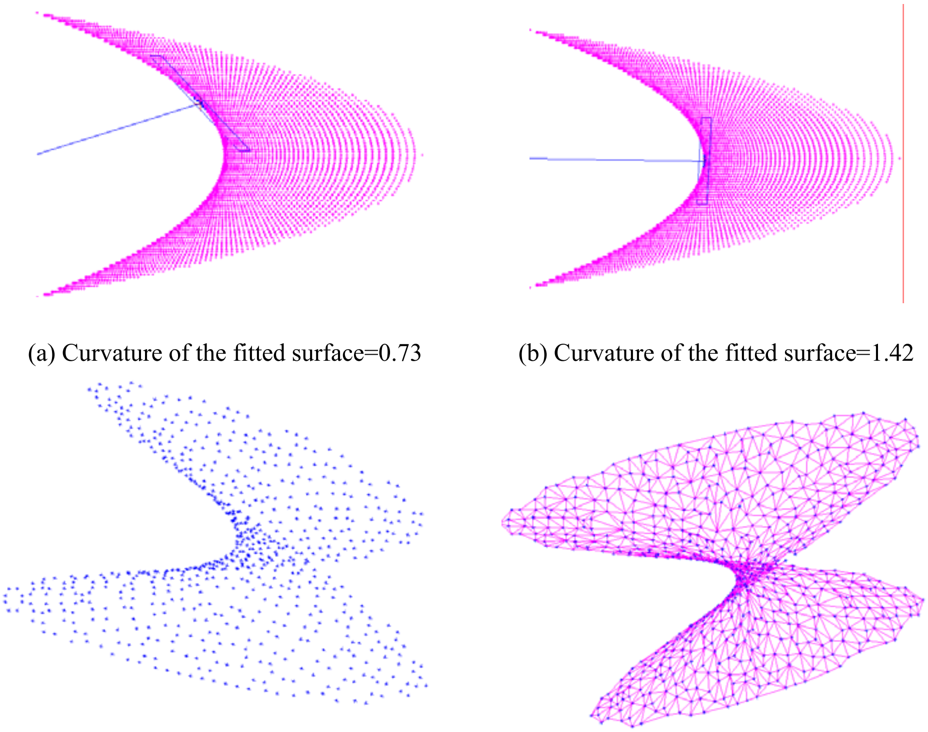


Figure 4.3. Discrete data set of the simulated model

We randomly selected a point from the screen, and then conducted patch fitting and curvature estimation at several locations. The results of two locations are shown Fig. 4.4a and b respectively. The blue plane is the estimated tangent plane of the selected point and the blue line is the normal of the tangent plane. The point with the maximum curvature of 1.42 was found. The bin size was obtained as 0.21 mm. Data thinning was conducted accordingly. There are 707 points left, as seen in Fig.d

4.4c. After that, we applied the algorithm presented in this chapter to the thinned set and obtained the meshed model as shown in Fig. 4.4d.



(a) Curvature of the fitted surface=0.73

(b) Curvature of the fitted surface=1.42

(c) The cloud data after filtering

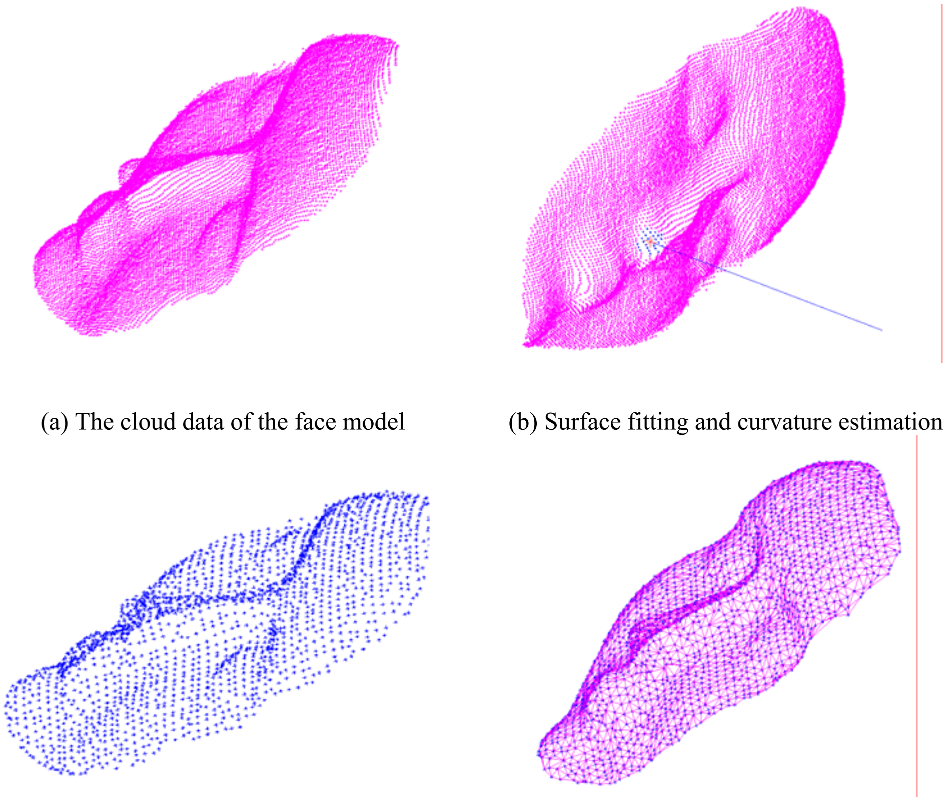
(d) The triangulated surface model

Figure 4.4. Case study 1

4.2.2 Case 2

The cloud data of the second case is gained by scanning a face model using the laser scanner, which occupies a volume of 45mm×64mm×15mm. There are totally 13,258 points generated, as seen in Fig. 4.5a.

Surface fitting and curvature estimation were conducted at several selected positions and it was found that the most curved region is at the nose area as shown in Fig. 4.5b. Using a filtering tolerance of 0.6 mm, the bin size of 1.2 mm was obtained. After thinning, 1,947 points are remaining as shown in Fig. 4.5c. The final meshed model after triangulation consists of 3,773 facets, shown in Fig. 4.5d.



(a) The cloud data of the face model

(b) Surface fitting and curvature estimation

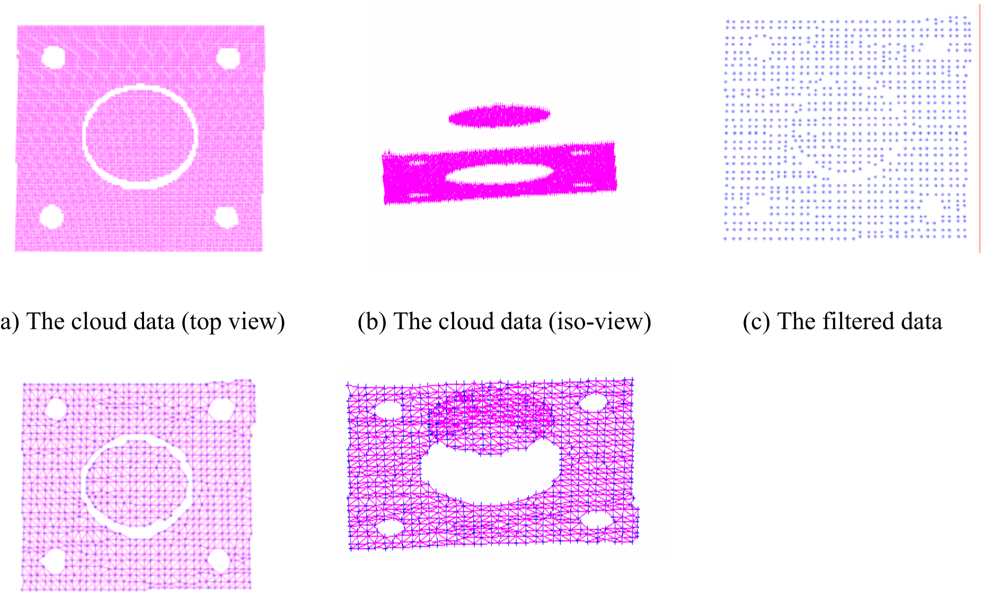
(c) The cloud data after filtering

(d) The triangulated surface model

Figure 4.5. Case study 2

4.4.3 Case 3

The model of the third case is a block with five holes at the top surface. The dimension of the object is 88 mm × 95 mm × 41mm. The radius of the big hole in the top surface of the block is 21 mm and the radius of the other four small holes in the top surface is 5mm. A scanning right from the top was conducted and 11,521 points were obtained as shown in Fig. 4.6a and b respectively. We then used a bin size of 3 mm to filter the data set and there are 978 points left, seen in Fig. 4.6c. Finally, a triangular meshed model with 1,386 facets was obtained as shown in Fig. 4.6d and e. It can be seen that the five holes have been successfully identified and meshed.



(a) The cloud data (top view)

(b) The cloud data (iso-view)

(c) The filtered data

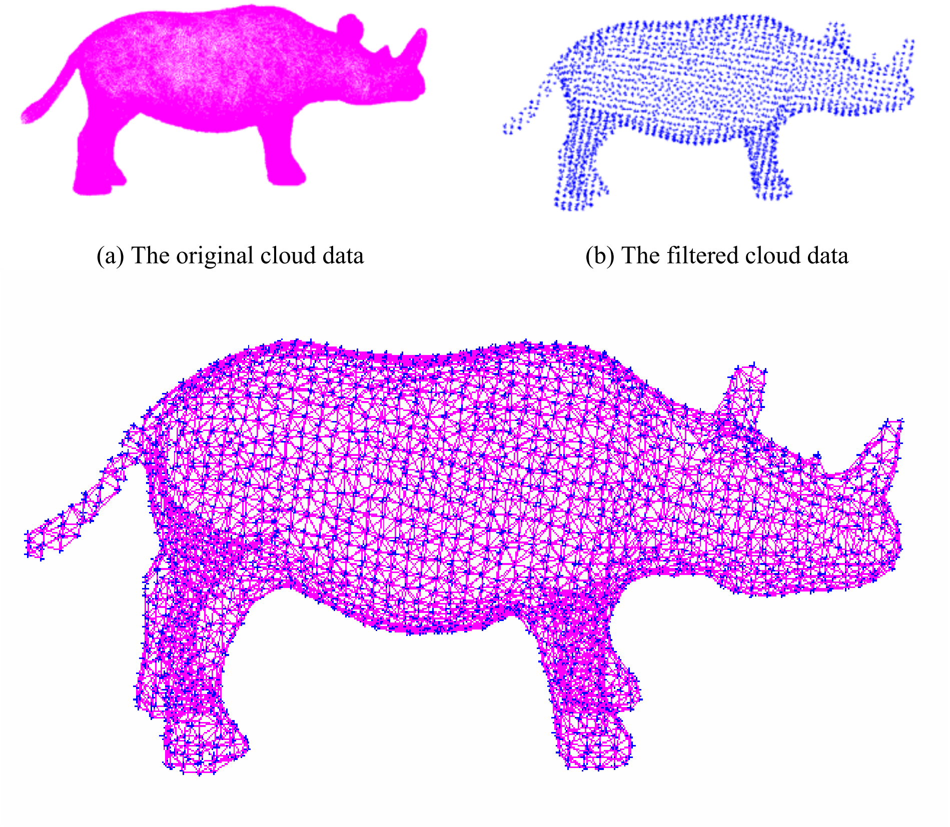
(d) The meshed model (top view)

(e) The meshed model (iso-view)

Figure 4.6. Case study 3

4.4.4 Case 4

The cloud data of case 4 is obtained by scanning a toy of rhino, which occupies a volume of 65 mm×130 mm× 35mm. The original data set consists of 57,835 points as shown in Fig. 4.7a. We used a bin size of 2.5 mm to filter the cloud data and there are 2,538 points remained as shown in Fig 4.7b. The triangulation process generated 4,766 facets as shown in Fig. 4.7c. This case study shows that the data thinning and triangulation algorithms can handle complex cases effectively.



(a) The original cloud data

(b) The filtered cloud data

(c) The meshed model

Figure 4.7. Case study 4

CHAPTER 5

PROBE PATH PLANNING

As we have discussed in the Chapter 2, laser scanners are commonly used in RE for data acquistion due to its fast data capturing speed. However, there is no guarantee that the important features such as holes and edges are accurately captured. While, the CMM has more accuracy than the laser scanner, the main disadvantage of the CMM is its low data capturing speed.

To resolve the above problem and combine the advantages of both the laser scanner and the CMM, a hybrid digitization method is developed in this chapter that first uses the cloud data captured from the laser scanner to form a triangular meshed model and then uses the meshed model as a reference to generate the probe paths for the CMM to re-capture important features such as holes and edges on the object.

To put the necessary information obtained from the point cloud of the scanned object into use in the CMMs, transformation is required to be done on the two coordinates. In order to obtain the transformation matrix between the two different reference systems, a dimensionally known base-plate with distinct easily identified four corner points from the cloud data is first scanned and then manually digitized on the CMM. Then the object is scanned and digitized together with its base-plate.

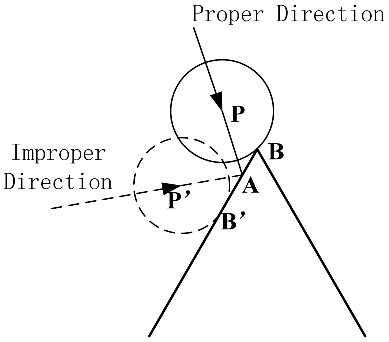


Figure 5.1. Proper and improper probe direction for capturing the edge

In this method, the designing of the probe-path on a CMM is very important. As shown in Fig. 5.1, **A** is a reference edge point obtained from the meshed model and **B** is the real edge point. As the touch-probe moves to **A** along the probe-path **PA**, it will touch the real edge point. On the contrary, if it approaches **A** along **P’A**, it will touch point B’, instead of A.

5.1 Problem Definition

As shown in Fig. 5.2, we assume that two planes, P1 and P2, form an edge and the doted line represents the reference edge in the meshed model, which is not the real edge but zigzags around the real edge in a predefined tolerance. The reference edge point **A** may lie in the plane P1 and P2 or in the space formed by these two planes.

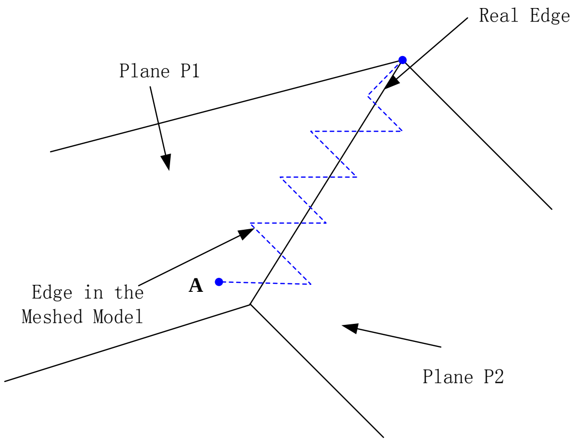


Figure 5.2 Illustration of real edges and edges in the meshed model

Furthermore, the reference edge point has all the information obtained in the triangulated process, e.g., neighbouring vertices, normal vectors, etc. An example is shown in Fig. 5.3, point **A** is a reference edge point that is shared by facets **BAC, BMA** and **CAN** belonging to the up patch (in blue), which have an average normal of **N**1 and facets **FAE, EMA** and **FAF** belonging to the down patch (in black), which have an average normal of **N**2. The sequenced data structure for a reference edge point, in this case **A**, is given below:

```
typedef struct SedgeStruct
{       double pt[3]; /* the 3-d coordinate of edge point, for example A*/
        double upnml[3]; /*the normal of the up facet that A belongs to, N1*/
        double dwnml[3]; /*the normal of the down facet that A belongs to,
                              N2*/
        double prbdrn[3]; / *the probe-path that is supposed to generated */
}Sedge;
```

The problem can be defined as to find the real edge information by using the reference edge of the mesh model: given a meshed model and the information of all the reference edge points, generate a proper probe-path, corresponding to every reference edge point, which ensures the probe of the CMM can digitize the real edge.

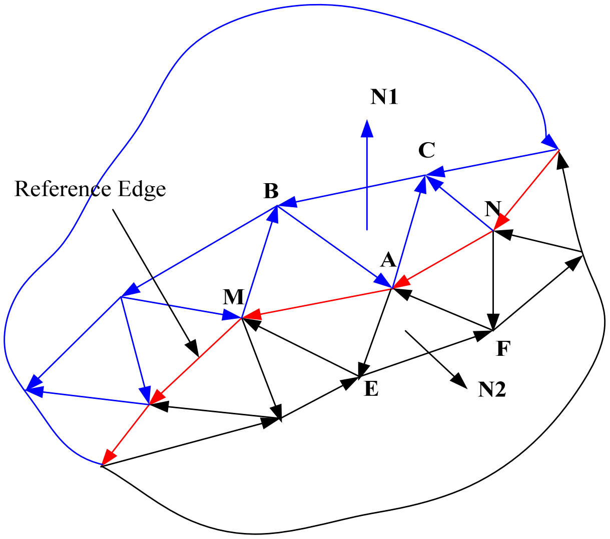


Figure 5.3. Illustration of reference edge in the meshed model

5.2 Probe-path Generation

As we have discussed above, the reference edge points may lie on the two planes that form the real edge or the space between the two planes. We will discuss how to find the probe path for each situation in the following section. But firstly, we set up a local coordinate system for each edge point from the reference edge information. As shown in Fig. 5.4, **A’** is the real edge point, **N**1’ and **N**2’ represent the normal of the surface elements that form the real edge. **BA** represents the facet **BAC** (as shown in Fig. 5.3) and **EA** represent the facet **EAF**. **A** is the reference edge point. **N1** and **N2** represent respectively the average normal vectors of the facets belonging to the up and down patch that share the vertex **A**. And the angle formed by **N1** and **N2** nearly equals to the angle formed by **N1’** and **N2’**, for the laser scanned data is within a small tolerance to the object, the normal of formed triangular facet is nearly the same as the real normal of object surface. Based on the above description, we set up a local coordinate for each edge point. It takes the real edge point, for example **A’,** as the origin. **N0**, the average of **N1’** and **N2’**, is y-axis. The x-axis is perpendicular to **N0**. The probe is represented in the **XA’Y** plane by a circle. A probe path is to be generated to make the probe in contact with the real edge point **A’** while approaching to the reference edge point **A**.

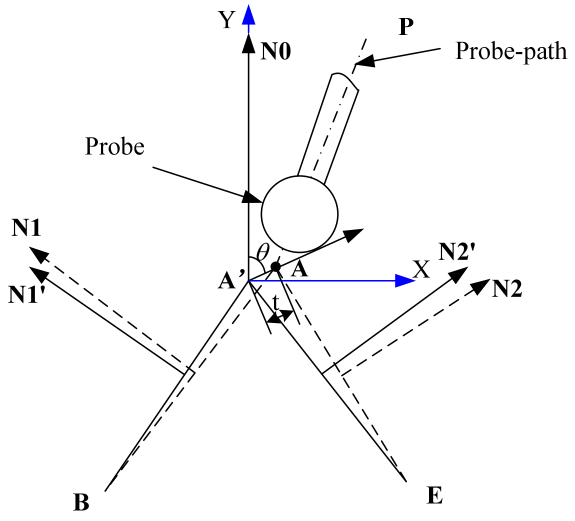


Figure 5.4. Illustration of the local coordinate of edge points

5.2.1 Probe-path generation with fixed reference edge point position

After setting the local coordinate for every edge point, we first discuss how to choose the probe path when the reference edge point lies in one of the surface element at A, shown in Fig. 5.5 (a, b, c), where **n1** and **n2** are the normal of the surface elements, **BE** and **BF**, which form the real edge point **B**; **n0** is the average of n1 and n2; the circles having the centre at **O**, **O’**, **P** and **P’** represent the touch-probe.

As shown in Fig. 5.5a, the circle **O** is tangent with the surface element **BE** at the real edge point **B**, the direction **OA** is the critical direction and the range of directions between **OA** and **n0** is the probe-path along which the probe can make contact with the real edge point **B** when approaching the reference edge point A from the left side of **n0**. We can see that the probe approaching along probe-path **P'A** will touch false edge point B' instead of the real edge point **B**.

a	b
c	d

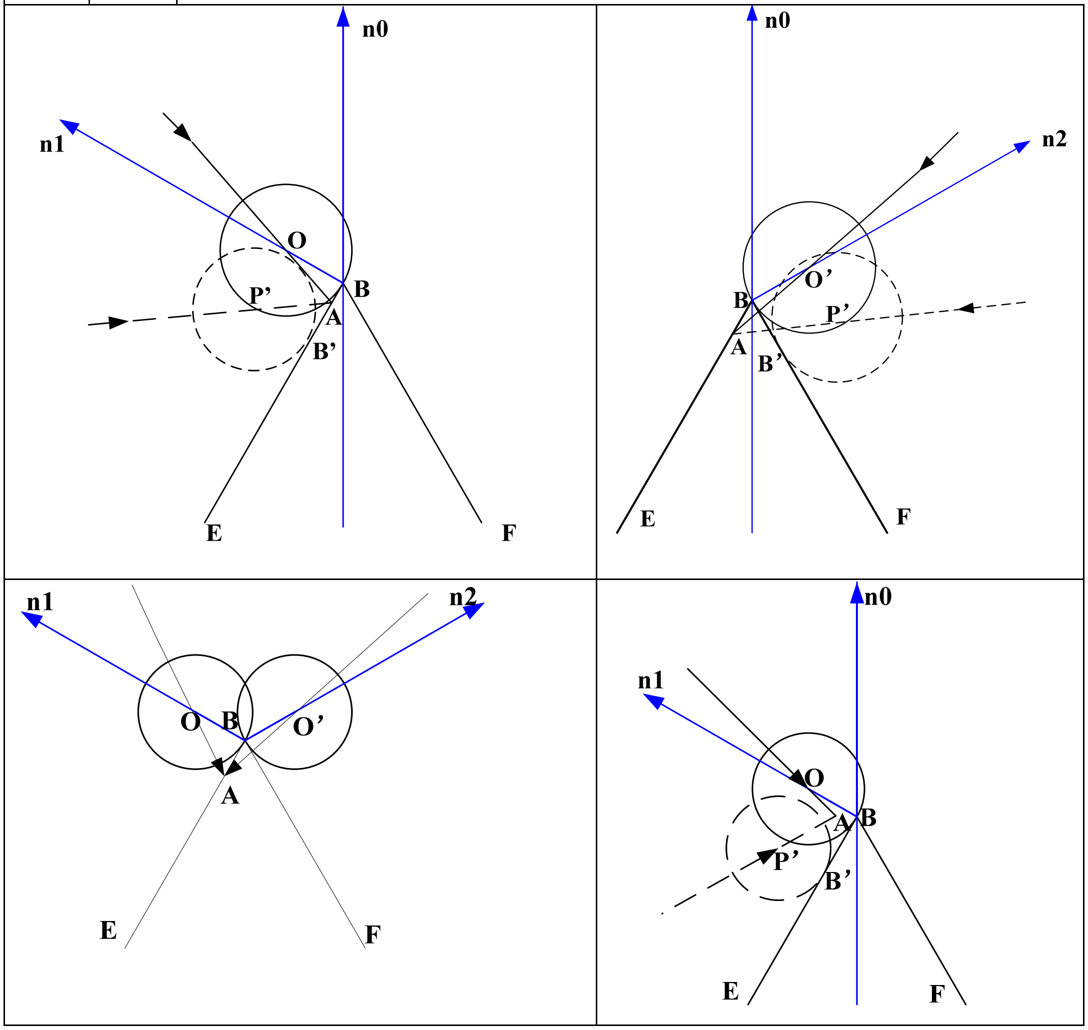


Figure 5.5. Range of the proper probe direction

Similarly, Fig 5.5b shows the probe-path generated along which the probe moves from the right side of **n0**. And as shown in Fig. 5.5c, the probe-path between

OA and **O'A** indicates the total range of directions, along which the probe can touch the true edge point **B**.

Fig. 5.5d shows the probe-path generated when the reference edge point lies in the space between the surface elements. The process of probe-path generation is the same as the situation when the reference edge point lies in the surface elements.

In this section, we generalize a method to design the probe-path assuming the actual position is known where the reference edge point lies in. Firstly, draw two circles with the radius of the touch-probe that are tangent with the two surface elements respectively at the real edge point **B**, seen in Fig. 5.6a. Secondly, draw two lines linking the centre points of the two circles and the reference edge point **A,** as shown in Fig. 5.6b. Finally, we find the probe-path that lies between the two lines.

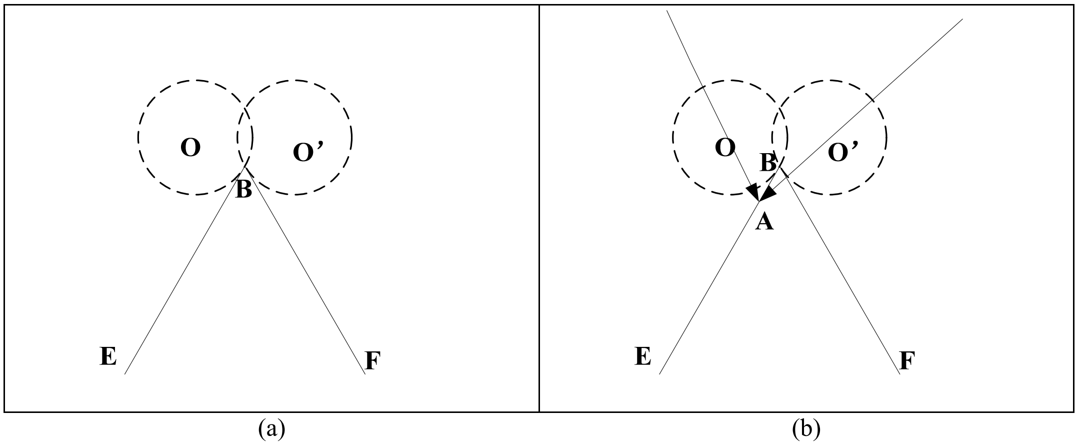


Figure 5.6. Illustration of probe-path generation

5.2.2 Probe-path generation when the position of reference edge point is unknown

In the previous section, we have discussed that given a fixed reference edge point we can find the probe-path along which the probe can find the real edge point. In practice, the reference edge points may deviate around the real edge in a small range based on the tolerance of laser and its position cannot be predicted. So, in this section we will discuss if there exists a subset of probe-path, along which the probe can touch the real edge points no matter where the reference edge point lies in. We will also discuss the influence of positional error of the reference edge point on the probe-path generation.

The relative position of the reference edge point according to the real edge point can be defined by two parameters, *t* and θ (as shown in Fig 5.4). *t* represents the distance between the reference edge point **A** and the real edge point **A'** and θ represents the angle between line **A'A** and line **N0**. In the practice, the error can be controlled in a range $t \le \delta$ (δ is the largest error), according to the accuracy of the laser scanner. θ can vary between 0 and 2π. In the following, we will discuss the influence of these two parameters on probe path generation.

1) The distance *t* is constant and angle θ varies

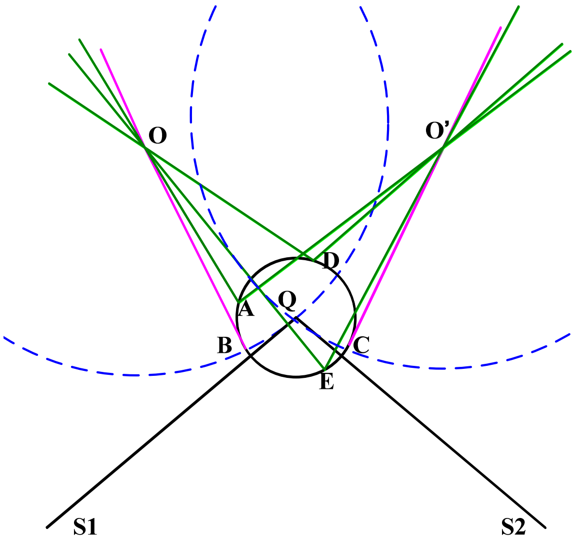


Figure 5.7. The influence of θ on the probe-path

As shown in Fig. 5.7, **S1** and **S2** represent two surface elements that form the real edge **Q**; the black circle centred at **Q** represents the reference edge points, for example **A**, **D**, and **E**, having the distance *t* from **Q**; the dotted blue arcs represent the circles centred at **O** and **O'** with the radius of the touch-probe, which is tangent with **S1** and **S2** at **Q**. We have discussed in the section 5.2.1 that the lines connecting the reference edge point with **O** and **O'**, for example **OA** and **O'A**, **OD** and **O'D**, **OE** and **O'E**, are the range of probe-path, along which the probe can make contact with **Q** while approaching **A**, **D** and **E**. We draw lines **OB** and **O'C**, which are tangent with circle **Q** at **B** and **C**.

We can see probe-path between **OB** and **O'C** are critical, which are included in all the probe-path no matter where the reference edge point lies.

2) The effect of distance *t*

As shown in Fig. 5.8, **Q** is the real edge point formed by two surface element **S1** and **S2**; **O** and **O'** represent the centre of the probe stylus ball, which are tangent with surface **S1** and **S2** at real edge point **Q**; **OB** and **O'C** are the critical probe-path of the reference edge points having the distance *t*1 with the real edge point; **OA** and **O'D** are the critical probe-path of the reference edge points having the distance *t*2 with the real edge point; *t*1<*t*2.

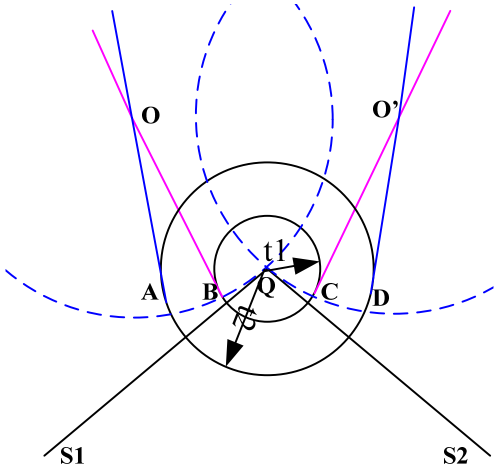


Figure 5.8. The influence of *t* on the probe-path

From the geometry, we can see that the further the reference point from the real edge point the smaller is the angle between the two critical probe-paths. And the smaller range of probe-path is always included in the larger ones.

From the above discussion, we conclude that given δ , the accuracy of the laser scanner, we can generate the probe-path range that is between the two critical probe-paths. Then along the probe-path, the probe can make contact with the real edge no matter where the reference edge point lies

5.2.3 Probe-path generation algorithm

The purpose of probe-path planning is to design a range of probe path according to the edge information of the meshed model, along which the probe can make contact of the real edge. First, we set up a local coordinate according to information of each reference edge point. Then we calculate the range of probe-path with knowing the accuracy of the laser scanner *t* and the radius of the probe *r*.

As shown in Fig. 5.9, we consider only the left side due to the symmetry. **S1** represents one surface element. **Q** is the true edge point. The circle centred at **Q** represents the reference edge points, having the distance *t* from the real edge point. **O** is the centre of the probe top as it is tangent with **S1** at **Q**. We draw line **OP** tangent with the circle **Q** at **P**.

Based on the discussions presented above, **OP** is the critical probe direction. The probe can touch the real edge point **Q**, moving towards **P** along any direction between **OP** and vertical line **AP** from the left side of the vertical line **BQ**.

Let $\angle S_1QS_2 = \alpha$; PQ = *t*; $OQ = r$ (radius of probe); $\angle OPA = \phi$. Then from the geometry we can obtain the following:

$$\angle PQO = a\cos(t/r) \tag{5.4}$$

$$\angle POQ = \pi/2 - \angle PQO \tag{5.5}$$

$$\angle PAQ = \pi/2 - \alpha/2 \tag{5.6}$$

$$\phi = \angle OPA = \angle PAQ - \angle POQ = \angle PQO - \alpha/2 \tag{5.7}$$

ϕ is the range of probe-path approaching from the left of vertical line

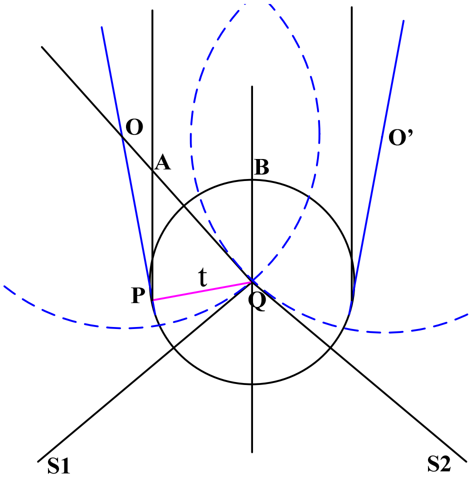


Figure 5.9. Probe-path generation

5.2.4 Probe-path validation

As shown in Fig. 5.10, we just consider the left side for the symmetry. S1 represents the top surface element. Q is the real edge point; T represents a random reference edge point position: $t \le \delta$ (δ is the largest error) and $0 \le \theta \le 2\pi$; OT is an input probe-path, represented by $\phi = \angle OAQ$ and O represents the centre of the touch-probe, having a radius *r*.

Firstly, from O we draw a line perpendicularly to S1, which intersect with the circle representing the touch-probe at M. Then, from M we draw a line parallel to OA, which intersect with S1 at P. From T we draw a parallel line intersecting a vertical line drawn from L. Assume OL equals *h*, which is the touch-probe retreat distance.

As shown in Fig. 5.10, if the S1 has infinite length, the profile of touch probe, represented by circle O must make contact of it at point P. If $P_x < 0$, which means the probe make contact with the surface, but not the edge, the probe-path OT is a failed generated probe direction. Otherwise, the probe will digitize the edge and the OT is a well generated probe-path.

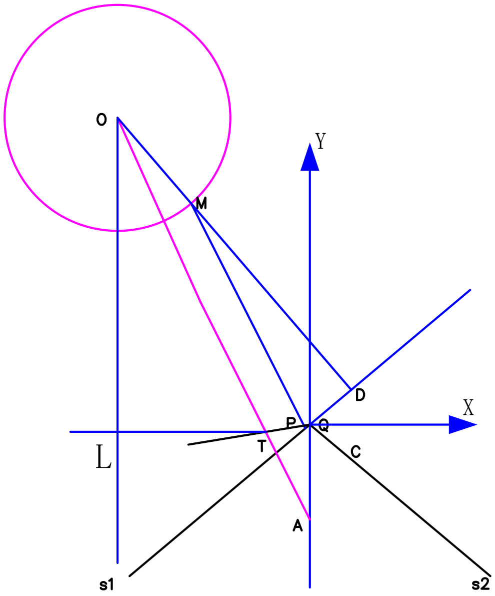


Figure 5.10. probe-path validation

Then from the geometry we can obtain the following relationship:

$$t_x = t\cos\theta \\ t_y = t\sin\theta \tag{5.8}$$

$$o_x = t_x - h\tan\phi \\ o_y = t_y + h \tag{5.9}$$

$$M(M_x, M_y) = \begin{bmatrix} (x-o_x)^2 + (y-o_y)^2 = r^2 \\ (y-o_y) = \tan(\pi - \alpha/2)(x-o_x) \end{bmatrix} \tag{5.10}$$

Solving Eq. (5.10), we have,

$$M_x = \frac{r}{\sqrt{1+\tan^2(\alpha/2)}} + o_x \\ M_y = \frac{-r\tan(\alpha/2)}{\sqrt{1+\tan^2(\alpha/2)}} + o_y \tag{5.11}$$

The equation of the line MP is as follows:

$$(y - M_y) = (x - M_x)\tan(\pi / 2 + \phi)$$

$$P(P_x, P_y) = \left[\begin{array}{l} y - M_y \ = -(x - M_x) / \tan\phi \\ y = x\tan(\pi - \alpha / 2) \end{array} \right) \tag{5.12}$$

Solve Eq. (5.12), we have,

$$P_x = \frac{M_x / \tan\phi + M_y}{1 / \tan(\alpha / 2) + 1 / \tan\varphi} \tag{5.13}$$

Algorithm

Input: *up and down surface element normal n1, n2 of each edge point, largest error δ, Touch-probe radios r, and a probe-path, represented as ϕ*

Output: *the number of false identified edge point according to the input probe-path*

```
Procedure: int number=0
           Calculate the angle between n1 and n2
           $\alpha$ Equals PI-angle_n1n2
           Calculate Px, from equations (5.8)-(5.13)
           If (Px<0) Number=number+1
           Else pedge-id=0;
```

5.3 Case Study

5.3.1 Case1

As shown in Fig. 5.11, **ADB** and **ACB** represent the two cones, which have radius R and height H. The probe path is defined as the angle between y-axis and the probe approaching direction, i.e., **BP** or **BP**'.

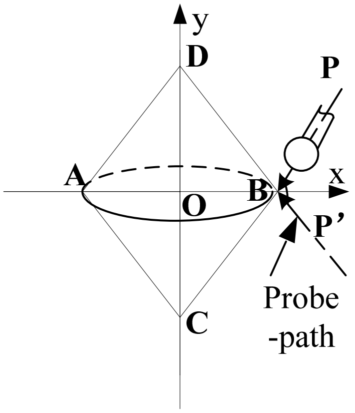


Figure 5.11. Illustration of two-cone intersection

The radius of the probe stylus ball is 5 mm and the largest scanning error is 2×10^{-1} mm. We then generate different probe-path direction range, among which the probe can make contact with the edge. Principally each reference edge point has a probe-path range, but for the edge is formed by the intersection of two cones, the range of the each probe-path is same. So we use probe-path angle to represent all the probe-path range for all the edge point in each condition, for example H=150, R=100 the probe-path angle equals $72.1° \sim 107.9°$. And the probe-path range can be calculated by the probe-path angle from the geometric relationship shown in Fig. 5.11, with the assumption that each largest probe-path range lies in the plane perpendicular to the referent edge element. The results according to different H and R are shown in Table 5.1.

Table 5.1. The probe-path for edge re-digitization in case 1

H	R	Probe-path angle range
150	100	72.1° ~ 107.9°
120	100	66.2° ~ 113.8°
100	100	58.7° ~ 122.3°
80	100	55.2° ~ 124.6°
60	100	47.5° ~ 132.5°

5.3.2 Case 2

As seen in Fig. 5.12, **ADB** and **ACB** represent a hemisphere and a cone, which have radius R and height H respectively. The radius of the probe stylus ball and the tolerance of the laser scanner are the same as in case study 1.

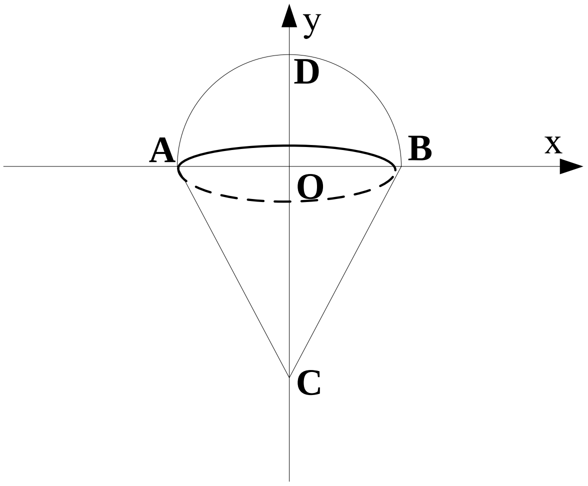


Figure 5.12. Illustration of hemisphere-cone intersection

We then generate different probe-path direction range according to different H and R as shown in Table 5.2.

Table 5.2. The probe-path for edge re-digitization in case 2

H	R	Probe-path angle range
20	100	105.25° ~ 153.45°
50	100	105° ~ 137.5°
70	100	105.7° ~ 129.3°
100	100	112.5° ~ 120.3°

5.3.3 Discussion

The probe-path generated in the above two case is calculated by the equation 5.4-5.7 with acknowledgement of the reference edge information, the assumed laser scanner tolerance and probe radius. Validation by graphical simulation has verified the result of each case study.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

In this thesis, a hybrid digitization method is presented, which combines the advantages of both the laser scanner and the CMM. The method begins with a filtering algorithm that thins the 3D scanned cloud data within a user controlled tolerance. Then a triangulation method is applied to construct the surface from the thinned data so that edges and holes can be identified. Taking the edge/hole information obtained as a reference, the probe-path is generated for CMMs to re-capture the real edges and/or holes.

The main contributions of this research are in two aspects. Firstly, curvature is estimated after patch surface fitting. It is capable of finding the maximum curvatures of the surface patches, according to which the bin size is calculated. Secondly, an algorithm is designed to generate the probe-path for a CMM to recapture the key features of the target object according to the information obtained in the triangulation process.

Further challenging issues are as follows:

Develop a data transformation program to transfer the reference model from the scanner's frame to the CMM frame. Develop validation by practical case study to verify the results.

REFERENCES

Amenta N, Bern M. Surface reconstrution by voronoi filtering. In Proceedings of the ACM 14th Annual Symposium on Comutational Geometry, New York; June 1998. p. 39-48.

Amenta N, Choi S, Koluri R. The power crust. In 6th ACM Symposium on Solid Modelling ands Applications, Ann Arbor, MI; June 2001.

Boissonat J-D. Pepresnting 2D and 3D shapes with the delaunay triangulation. In 7th IEEE International Conference on Pattern Recognition, Montreal, Canada; July/August 1984. p. 745-8.

Bernardini F, Bajaj C, Chen J, Schikore D. Triangulation-based object reconstruction methods. Proceedings of SCG'97 (ACM Symposium on Computational Geometry); 1997. 481-484.

Boissonat J-D, Cazals F. Smooth surface reconstruction via natural neighbour interpolation of distance function. Proceedings of SCG'00 (ACM Symposium on Computational Geometry); 2000, 223-232.

Choi B. K., Shin H. Y., Yoon Y. I. and Lee J. W. 'Triangulation of scattered data in 3D space' Computer-Aided Design Vol. 20, No. 5, 1988, pp. 239-248.

Cignoni P. Montani C. and Scopigno R. 'DeWall: A fast divide and conquer Delaunay triangulation algorithm in Ed' Computer-Aided Design, Vol. 30, No.5, 1998, pp. 333-341.

Chen, X. and Schmitt, F., Surface modelling of range data by constrained triangulation. CAD, 1994, 26(8), pp. 632-645

Chivate P. N. and Jablokow A. G. 'Solid-model generation from measured point data' computer-Aided Design, Vol. 25, No. 9, 1993, pp. 587-600.

Curless B, Levoy M. A volumetric method for building complex models from range images. Comput Graphsic (Proc SIGGRAPH) 1996; 303-12.

Dey TK, Gisesen J, Hudson J. Delauney based shape reconstruction from large data. Symposium on Parallel and Large-Data Visualization and Graphics 2001; 146:19-27.

Eck M, Derose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. Computer Graphics (Proceedings of SIGGRAPH), 1995, p. 82-90.

Fang T. P. and Piegl L. ‘Delaunay triangulation in three dimensions’ IEEE Computer Graphics and Applications, Vol. 15, No. 5, 1995, pp. 62-69.

Hamann, B., Curvature approximation for triangulated surfaces. Computing Supplementum, 1993, 8, pp. 139-153

Hinker P, Hansen C. Geometric optimization. IEEE Visualization’ 93 Proceedings, Los Aldmitos, CA: IEEE Compater Society Press, 1993. p 189-95.

Hoppe H, De Rose T, Duchamp T, McDonald J, Stuetzle W. Mesh optimization. ACM Coputer Graphics, SIGGRAPH 1993:19-26

Hoppe H. Progressive meshes. Computer Graphics (Proceedings of SIGGRAPH), 1996, p. 99-108.

Hinker P, Hansen C. Geometric optimization. IEEE Visualization ’93 Proceedings, Los Aldmitos, CA: IEEE Computer Society Press, 1993 p. 189-95.

Jun, Y., Raja, H. Y. and Park, S., Geometric feature recognition for reverse engineering using neural networks. International Journal of Advanced Manufacturing Technology, 2001, 17, pp. 462-470

Kobbelt, L., Discrete fairing and variational subdivision for free-form surface design. Visual computer, 2000, 16(3-4), pp. 142-158

Lawson C. L. ‘Software for C1 surface interpolation’ Mathematical Software III Academic Press, 1977, pp.161-194.

Lee, K. H. and Woo, H., Direct integration of reverse engineering and rapid prototyping. Computers & Industrial Engineering, 2000, 38 (1), pp.21-38.

Lucier B. Wavelet and image compression. In: Lyche T, Schumaker L, editors. Mathematical Methods in Computer Aided design 11, Academic Press, 1992. p. 391-400.

Menq, C. and Chen, F. L., Curve and surface approximation from CMM measurement data. Computers & Industrial Engineering, 1996, 30(2), pp. 211-225

Milroy, J., Bradley, C. and Vickers, G. W., Segmentation of a wrap-around model using an active contour. CAD, 1997, 29, pp. 299-320

Piegl LA, Richard AM. Tessellating trimmed NURBS surfaces. Computer-Aided Design 1995;27(1):16-26.

Peng, Q. J. and Loftus, M., A new approach to reverse engineering based on vision information. International Journal of Machine Tools and Manufacture, 1998, 38(8), pp. 881-899

Stollnitz EJ, De Rose T, Salesin DH. Wavelets for computer graphics. IEEE Computer Graphics and App 1995; May:76-84.

Samet H. Applications of Spatial Data Structure. Reading, MA: Addison-Wesley, 1990.

Shephard MS, Georges MK. Automatic three-dimensional mesh generation by the finite Octree technique. Int J Numer Methods Eng 1991;32:709–49.

Shi M. Surface reconstruction using an edged-based growing method. M. Eng. Thesis, National university of Singapore, 2001.

Soucy M, Laurendeau D. Multiresolution surface modeling based on hierarchical triangulation. Computer Vision and Image Understanding 1996; 63(1):1-14

Sun W, Bradley C. Zhang Y.F. and Loh H.T. ‘Cloud data modeling employing a unified, non-redundant triangular mesh’ Computer-Aided Design, Vol. 33, No. 2, 2001, pp.183-193.

Sheng X, Hirsch BE. Triangulation of trimmed surfaces in parametric space. Computer-Aided Design 1992;24(8):437-44.

Tnamminen M, Samet H. Efficient Octree conversion by connectivity labeling. Computer Graphsics 1984; 18(1):43-51.

Ueng, W. D., Lai, J. Y. and Doong, J. L., Sweep-surface reconstruction from three dimensional measured data. CAD, 1998, 30(10), pp. 791-805

Varady, T., Martin, R. R., and Cox, J., Reverse engineering of geometric models-an introduction. CAD, 1997, 29 (4), pp. 255-68.

Weir D. J., Milroy M. J., Bradley C. and Vickers G. W. ‘Reverse engineering physical models employing wrap-around B-spline surfaces and quadrics’ Proceedings of the Institution of Mechanical Engineers - Part B, Vol. 210, No. B2, 1996, pp. 147-157.

Williams L. Pyramidal parametric. Computer Graphics (SIGGRAPH’ 83); 17 (3):1-11.

Woo H., Wang E., Wang Semyung, Lee Kwan H., A new segmentation method for point cloud data. International Journal of Machine Tools & Manufacturing 2002, 42, 167-178.

Yang, M. and Lee, E., Segmentation of measured point data using a parametric quadric surface approximation. CAD, 1999, 31, pp. 449-457.

Yan, X. and Gu, P., A review of rapid prototyping technologies and systems. CAD, 1996, 28(4), pp. 307-318.