
3D ANALYSIS OF TOOTH SURFACES TO AID ACCURATE BRACE PLACEMENT

SHEN YIJIANG

(M.ENG, NUS)

A THESIS SUMBITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND ELECTROINC ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Contents

Abstract

Orthodontics is one of the specialized fields of dentistry, which is concerned with the growth, and development of the dentition and course, the treatment of irregularities that can occur. Orthodontists are interested in evaluating geometric parameters to describe teeth and malocclusions occurring in teeth. Traditionally, orthodontists use plaster models to study these parameters; they use such tools as hand caliper-and-ruler measurements to manually measure sizes, shapes and distances. Tooth brackets are often used to correct misalignments and malocclusions. The decision of selecting a tooth bracket for a specific tooth has been an empirical activity of the orthodontists. Traditional diagnoses require tedious work, and the results are not always satisfactory.

Computer vision techniques together with 3D scanning and visualization tools enable the orthodontists to evaluate and compute geometric measurements and also to decide the best-fit tooth bracket easily and more accurately. This thesis describes work that applies 3D computer vision techniques for the surface matching of tooth bracket surfaces and tooth surfaces from 3D scanning of tooth models and tooth bracket surfaces, 3D visualization of tooth models, manual segmentation of tooth surfaces, and finally a technique of matching the tooth bracket surfaces and tooth surfaces. These works will help the orthodontists to choose a precise and even customized tooth bracket to fit a specific tooth surface.

CHAPTER 1

INTRODUCTION

Orthodontics is a branch of dentistry concerned with correcting and preventing irregularities of the teeth and poor occlusion. The goal of orthodontic treatment is to reposition the teeth into a proper bite (occlusion) while maintaining or improving a person’s appearance. The practice of orthodontics requires professional skill in the design, application and control of corrective appliances (fixed and removable) to bring teeth, lips and jaws into proper alignment and achieve facial balance. Orthodontists often use tooth brackets to help align irregular teeth. An important consideration is therefore the matching of tooth brackets to tooth surfaces. This consideration requires surface analysis of tooth bracket surface and tooth surface.

To aid the orthodontists in the treatment and diagnosis of misalignment and malocclusion, the surface patches of tooth bracket and tooth surface have to be analyzed. The work presented in this thesis has two main objectives. The first object is to develop a suite of tools and programs to automatically analyze the plaster models taken from a patient. These proposed computer-vision based tools and programs will eventually be incorporated into a larger system capable of complete tooth diagnosis and description. The other objective is to use the extracted tooth surface and tooth bracket surface to compute similarity measurements [26] in order to find a best fit of the tooth brackets to the tooth surfaces and subsequently to help in designing customized tooth brackets and other orthodontics devices. Current orthodontics devices depend on coarse models that seldom take into account differences in shape geometry of tooth surfaces found in people belonging to different ethic groups for

---

example, and orthodontists currently depend on their experiences in their diagnoses and treatments.

1.1 The Role of Computer Vision in Orthodontics

Orthodontists routinely diagnose malocclusion and plan treatment based on information gathered from clinical examination and evaluation of records. Of the records taken, photographic representation of the patients' face, the cephalogram and the plaster model are essential aids in diagnosis and treatment planning. Cephalogram is the most common radiographic view used for facial analysis derived from the relative geometry between identified landmarks on the X-ray images. The plaster dental-moulds are taken directly from the patients' mouth. Plaster models are widely used by dentists and clinics in day-to-day diagnosis of orthodontic problems and are invariably the first step in realizing treatment. Orthodontists usually use tooth brackets in the treatment of misalignment and malocclusion. There are several commercial available sets of tooth brackets, and the selection of a tooth bracket to put on a patient's tooth is an empirical activity of the orthodontists. This activity results in inherent error because of lack of complete information of the tooth bracket and tooth surface.

In the early years of computer vision, the shape information of three-dimensional objects was obtained using camera images that are two-dimensional projections of three-dimensional objects. There have been a few attempts at automating the tasks related to orthodontic treatment evaluation. These include using wax-wafer alternatives to plaster moulds [35], detecting interstices on wax-wafer imprints [36], and detection of cups and other important surface features, again on wax-wafer imprints [37]. A substantial amount of work has addressed issues related to segmentation [37,38,39], which is an orthodontics problem. Computer modeling techniques for describing the tooth surface have been suggested in [40,41]. Finite element methods for discussing the mechanical properties of tooth brackets have been discussed in [42,43]. Because of the lack of depth information about the objects in the scene, the proposed approaches suffer from difficulties especially when there are such problems as significant lighting variations, complex shape of the objects, etc. In recent years, due to the advances in three-dimensional scanning technology and various shape recovery algorithms, digitized three-dimension surface data have become widely available.

To aid orthodontists in deciding which tooth bracket is best fit to a specific tooth surface, surface analysis of tooth bracket surface and tooth surface has to be conducted. A suitable surface representation of the tooth bracket surface and tooth surface should be applied and later on surface matching can be carried out. The main objective of the work described in this thesis is to design a system capable of producing customized tooth brackets from a three-dimensional mould taken from a patient’s jaw. The methodology suggested can be easily ported to a clinical setting eliminating the need for extensive background support from technical personal. The computer vision based technique, described in this thesis has good accuracy, which is limited by the resolution of the acquisition device, the laser scanners.

Towards the achieving the main objective of the work, tools related to the visualization of tooth models, segmentation of tooth surface from a tooth model, and the visualization of tooth bracket surfaces, have been developed. .

1.2 Previous Work

The key point in the matching of tooth bracket surface to tooth surface, is to find a good representation of the surfaces and then the surface matching can be conducted. Applications of surface matching can be classified into two categories. The first category is *surface registration* [26]. Surface registration can be roughly partitioned into three issues: choice of transformation, elaboration of surface representation and similarity criterion, and matching and global optimization. The first issue concerns the assumptions made about the nature of relationships between the two modalities. The second issue determines what type of information that needs to be extracted from the 3D surface, which typically characterize their local or global shape, and how we organize this representation of the surface, which will lead to improve efficiency and robustness in the last stage. The last issue pertains to how we exploit this information to estimate transformation which best aligns local primitives in a globally consistent manner or which maximizes a measure of the similarity in global shape of two surfaces. The registration of 3D surfaces is dealt extensively in machine vision and medical imaging literature as industrial inspection, surface modeling and mesh watermarking [26]. The second category is object recognition with the goal of locating and/ or recognizing an object in a cluttered scene. Robot navigation is one of the application examples in this category.

A considerable amount or research has been conducted on comparing 3D free-from surfaces. The approaches used to solve the problem can be classified into two categories according to methodology. Approaches in the first category try to create some form of representation for input surfaces and transform the problem of comparing input surfaces to the simplified problem for comparing their representations. These approaches are used most often in model-based object recognition. In contrast, approaches in the second category work on the input surface data directly without creating any representation. One data set is aligned to the other by looking for the best rigid transformation. These approaches are most used in surface registration.

In our work, two kinds of laser scanners are used. One is the *Cyberware Laser Scanner*; the scanner scans the model and gives out the triangular mesh objects. The other scanner in the Mechanical Engineering Lab provides explicit 3D points from which a 3D model can be constructed. In [3], Partial Differential Equation parameterization and neural network Self Organizing Maps parameterization were developed for the parameterization stage. The Gradient Descent Algorithm and Random Surface Error Correction were developed and implemented for the surface fitting stage.

Many local representations are primitive based. In [9], model surfaces are approximated by linear primitives such as points, lines and planes. The recognition is carried out by attempting to locate the objects through a hypothesis-and-test process. In [5], super segments and splashes are proposed to represent 3D curves and surface patches with significant structural changes. A splash is a local Gaussian map describing the distribution of surface normals along a geodesic circle. Since a splash

----

can be represented as a 3D curve, it is approximated by multiple line fitting with differing tolerances. In [4], a three-point-based representation is proposed to register 3D surfaces and recognize objects in clustered scenes. On the scene object, three points are selected with the requirement that (1) their curvature values can be reliably computed; (2) they are not umbilical points; and (3) the points are spatially separated as much as possible. In [4], a curved or polyhedral 3D object is represented by a mesh that has nearly uniform distribution with known connectivity among mesh nodes. A shape similarity metric is defined based on the L_2 distance between the local curvature distributions over the mesh representations of the two objects.

One major approach to surface matching is based on matching individual surface points in order to match complete surfaces. Two surfaces are said to be similar when many points from the surfaces are similar. By matching points, we are breaking the problem of surface matching to many smaller problems. Stein and Medioni [5] recognized 3D objects by matching points using structuring indexing and their “splash” representation. Similarly, Chua and Jarvis [6] match points to align surfaces using principal curvatures. In [7] and [8], spin-image is used to compare the similarity of two surfaces. Spin-images are simply transformations of the surface data; they are created by projecting 3D points to 2D images, spin-images do not impose a parametric representation on the data, so they are able to represent surfaces of general shape. Instead of looking for primitives and feature points at some part of the object surface with significant structure changes, a Spin-image is created for every point of the object surface as a 2D description of the local shape at that point. Given an oriented point on the surface and its neighborhood of a certain size, the normal vector and tangent plane are computed at that point. Then the shape of the neighborhood is

----

described by the relative positions of the vertices in the neighborhood to the central vertex using the distances to the normal and tangent plane. A Spin-image is a 2D histogram of those distances. Good recognition results in complex scenes using Spin-Images are reported in [10]. However, Spin-images are not well understood at a mathematical level and they discard one dimension information of the underlying surfaces, namely, Spin-images do not preserve the continuity of the surfaces.

Among 3D surface registration algorithms, Iterative Closet Point (ICP) plays an important role. In [14], the ICP shape matching algorithm is proposed. ICP handles the full 6-degree of freedom, and it is independent of shape representation. It does not require preprocessing of 3D point data, such as smoothing, as long as the number of statistical outliers is near zero. Although this approach guarantees finding the local minimum of the registration error, it requires good initial estimate of the transformation in order to find the global minimum. Another limitation of this approach is that it cannot handle two surfaces, that only partially overlap. A heuristic method was proposed in [16] to overcome partially overlapping difficulty. A K-D tree structure was also used in [16] to accelerate the process of finding the closet point. Unlike the ICP approach, an algorithm is proposed in [17] to increase the accuracy of registration by minimizing the distance from the scene surface to the nearest tangent plane approximating the model surface. In order to reduce computation complexity, control points are selected for registration instead of using the entire data set of the model surface. However, this may not work well on surfaces with no control points selected on some of their parts that have significant structure changes. Moreover, this approach also requires a good initial estimate of the transformation. In [23], surfaces are approximated by constructing a hierarchy of Delaunay triangulations at different

----

resolution levels. In summary, in order for the surface registration algorithms to work well, a good initial estimate of the transformation is usually required.

1.3 Problem Definition in Orthodontics Work

In our orthodontics experiments, the tooth models are scanned using the *CyberWare Laser Scanner*. The tooth surface is then segmented from the tooth models. The set of tooth brackets is scanned using *MAHR OMS 400 Multi-Sensor Coordinate Measuring Machine* and tooth bracket surfaces are extracted. The surface patches are represented by triangular meshes in the 3D space.

We construct the Harmonic Maps of the tooth surfaces and tooth bracket surfaces, which are then used to generate the Harmonic Shape Images of the surfaces. The Harmonic Shape Images of the tooth bracket surface and tooth surface are compared to find the best fit.

1. 4 Thesis Overview

Remain chapters of the thesis are summarized as follows.

Chapter 2 provides a brief introduction to the orthodontics work. Chapter 3 describes the visualization of the tooth models and tooth bracket surfaces and describes in detail the 3D acquisition system used to digitize the dental plaster cast. Chapter 4 describes in detail the generation of Harmonic Maps and Harmonic Shape Images. Chapter 5 describes the matching of Harmonic Shape Images, the resampling of Harmonic

Shape Images, the relocating of the resampling points. Chapter 6 describes the how the Harmonic Shape Images are applied in the matching of the tooth surfaces and the tooth bracket surfaces. The results of the matching are discussed. Chapter 7 concludes this thesis by summarizing our contributions and describing possible future research in this area.

CHAPTER 2

BACKGROUND ON ORTHODONTICS

One major consideration of orthodontics is the use of special devices, also called appliances, to move teeth or adjust the underlying bone. Dental braces are used to straighten crooked teeth, align upper and lower jaws, and improve the aesthetics of smiles and faces. Teeth can be moved by a number of various removable appliances or by fixed braces, depending on the kind of problem that was originally present.

Fixed braces usually include metal bands that are cemented to the molars, and metal brackets that are directly bonded or glued to the enamel of front teeth (incisors and bicuspids). Fixed braces, as the name suggests, are not removable by the patients. A stainless steel arch wire is used to connect the bands and the brackets in each arch (one for the upper teeth and one for the lower teeth).

2.1 Basic Dental Terminology

Here is a brief description of the often-used terms.

Mandible:	The lower jaw; the inferior maxilla.
Maxillary:	Pertaining to the upper teeth.
Malocclusion:	Poor positioning or inappropriate contact between the teeth on closure.
Buccal:	Pertaining or directed toward the cheek.

Bracket: A metal or ceramic part that is glued onto a tooth and serves as a means of fastening the arch wire.

Braces: Orthodontics appliances used to correct dental irregularities; consists of many brace-pads (brackets), and a supporting arch wire.

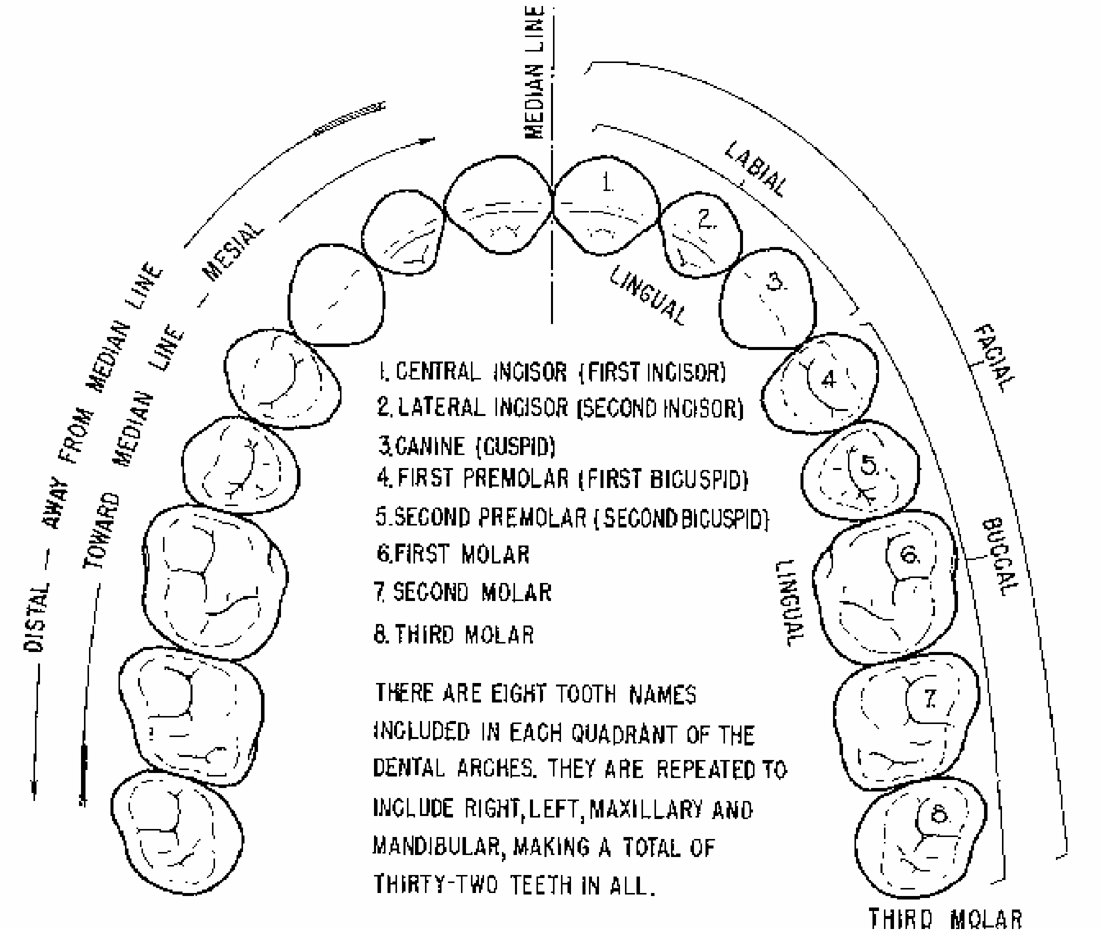

MEDIAN LINE
LABIAL
LINGUAL
FACIAL
BUCCAL
LINGUAL
MESIAL
DISTAL — AWAY FROM MEDIAN LINE
TOWARD MEDIAN LINE
1.
2.
3.
4
5.
6.
7.
8.
1. CENTRAL INCISOR (FIRST INCISOR)
2. LATERAL INCISOR (SECOND INCISOR)
3. CANINE (CUSPID)
4. FIRST PREMOLAR (FIRST BICUSPID)
5. SECOND PREMOLAR (SECOND BICUSPID)
6. FIRST MOLAR
7. SECOND MOLAR
8. THIRD MOLAR
THERE ARE EIGHT TOOTH NAMES INCLUDED IN EACH QUADRANT OF THE DENTAL ARCHES. THEY ARE REPEATED TO INCLUDE RIGHT, LEFT, MAXILLARY AND MANDIBULAR, MAKING A TOTAL OF THIRTY-TWO TEETH IN ALL.
THIRD MOLAR

Fig 2.1 Arrangement and surface of teeth

2.2 Bracket Design and Placement Issues

When an orthodontic force is applied to a tooth over a period of time, the tooth moves owing to resorption (dissolving) of the underlying alveolar bone on the pressurized side and apposition of new bone tissue on the opposite side. This is the theory behind making use of a host of orthodontics appliances to correct tooth alignment and malocclusion problems. Fixed appliances remain the most popular choice of an orthodontic appliance because of their effectiveness and precision in tooth movement.

This thesis discusses the surface analysis of the bracket surface that actually sits on a tooth’s lateral (or buccal) surface. Most orthodontists prescribe a “standard” bracket to a patient that does not always take into account the shape surface of an individual tooth. The methodology applied in the thesis makes it simpler for the orthodontists in their diagnoses and treatment. Bracket placement is normally done on the intersection of the *Long Axis of the Clinical Crown (LACC)* and the *Mid-Transverse Plane* (also called the *Andrews Plane*). The *LACC* is a longitudinal line and is easily marked------it divides a single tooth sagittally into two sections, left and right. The *Clinical Crown* refers to the portion of dental crown that is visible above the gums. The *Mid-Transverse Plane* divides this *Clinical Crown* into transversely into two sections, upper and lower.

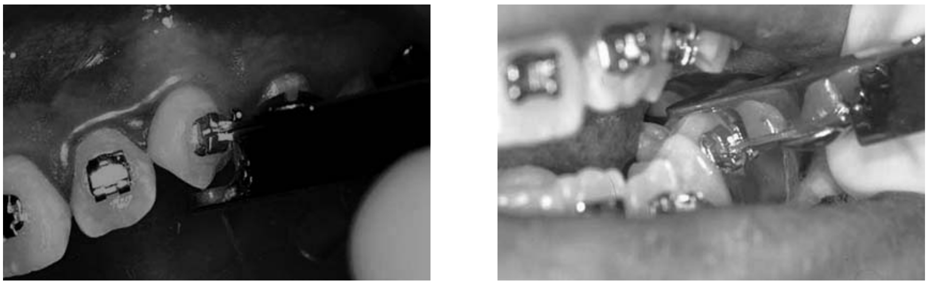


Fig 2.2 Positioning of tooth brackets

2.3 Overview of the Solution to the Surface Matching Problem

The purpose of this study is to develop a set of tools and software programs to help the orthodontists in several ways as the visualization of 3D scenes, and selection of best-fit tooth bracket to the tooth surface. The key point of the problem lies in 3D free form surfaces matching. Difficulties of matching 3D free-form surfaces include the following: Topology, Resolution, Connectivity, Pose and Occlusion. The two surfaces to be matched may have different topologies. The topology issue is difficult to address when trying to conduct global matching between two surfaces. Generally speaking, the resolutions of different digitized surfaces are different. The resolution problem makes it difficult to establish correspondences between two surfaces, which in turn, results in the difficulty of comparing the two surfaces. Even if the resolution of the two sampled surfaces is the same, in general, the sampling vertices on one surface are not exactly the same as that on the others. For arbitrary triangular meshes, the connectivities among vertices are arbitrary. Even if two surfaces have same number of vertices, they may still have different connectivities among vertices. This is in contrast to images. An image has a regular *m* by *n* matrix structure. The connectivities are the same for all pixels (pixels on the boundary have the same connectivity pattern as well). When conduct template matching, the correspondences between two images can be naturally established. It has been mentioned that there is no prior knowledge about the positions of the two surfaces in 3D space. Therefore, unlike conduct template matching of images, there is no natural coordinate system for aligning two surfaces. Although an exhaustive search strategy could be used to find the transformation in the six-dimensional space, it is computationally prohibitive without a good initial estimate of the transformation. Either self-occlusion or occlusion due to other objects is a common phenomenon in real scenes. When comparing two images, if occlusion is present in one image, then some robust techniques maybe used to discount the corresponding part in another image, so that only the non-occluded parts of the two images are taking into account in template matching. Here, it is important to notice that the occlusion does not change any of the remaining part of the images. Therefore, the comparison result of the two images will not be affected by occlusion as long as the occluded part can be correctly detected and discounted. In contrast to comparing 2D images, matching 3D free-form surfaces is far more complicated when occlusion is present in the scene. Model-based matching is a common framework for solving the 3D surface-matching problem. Although a considerable amount of work has been done in developing representations for 3D free-form surfaces, the problem of developing occlusion-robust representation is still open. Occlusion is not encountered in our work, because the tooth surfaces and tooth bracket surfaces are all intact without occlusion after we scan the tooth models and the tooth bracket surface, and extract tooth surfaces from the tooth model.

In [26], the surface-matching problem is investigated using a mathematical tool called *harmonic maps*. Harmonic maps are used for studying the mapping between different metric manifolds from an energy minimization point of view. A surface representation called *harmonic shape images* [26] is generated to represent and match 3D free-form surfaces. The basic of harmonic shape images is to map a 3D surface patch (the definition of surface patch is defined in Chapter 4) with disc topology to a 2D domain and encode the shape information of the surface patch into the 2D image. This simplifies the surface-matching problem to a 2D image-matching problem. Harmonic shape images, which are well defined mathematically, have the following advantages: (I) preserve both the shape and continuity of the underlying surfaces; (II) robust to occlusion; (III) independent of any specific sampling scheme.

The work described in this thesis involves the following:

- Segmentation of tooth surface.
- Scanning of tooth models and tooth bracket surfaces to obtain 3D representation.
- Construct the Harmonic Maps of the tooth surfaces and tooth bracket surface.
- Construct Harmonic Shape Images of the surface patches.
- Carry out surface matching by comparing the Harmonic Shape Images, and computing similarity measurements.

2.4 Manual Segmentation of Tooth Surface from Tooth models

In order to compare the similarity of the tooth surface and the tooth bracket surface, individual tooth surface is manually segmented. There are two major steps in the manual segmentation of tooth surface:

1. Surfaces patches containing an individual tooth surface are extracted from the tooth model using OpenGL Selection mode. When the left mouse button is pressed, the surrounding area of the clicked point is selected.
2. Extract the tooth surface from the tooth surface patch obtained in step 1. This extraction deals with some mathematical computation.

2.4.1 On OpenGL

OpenGL is a low-level graphics library specification. OpenGL makes available to the programmers a small set of geometric primitives------- points, lines, polygons, images and bitmaps. OpenGL provides a set of commands that allow the specification of geometric objects in two or three dimensions, using the provided primitives, together with commands that control how these objects are rendered into the frame buffer.

The OpenGL API was designed for use with the C and C++ programming languages, but there are also bindings for a number of other programming languages such as Java, Tcl, Ada and FORTRAN. The OpenGL specification is operating system and windowing independent. It relies on windowing system for window management, event handling, color map generation, etc.

OpenGL is a software interface to graphics hardware. This interface consists of about 120 distinct commands, which you use to specify the objects and operations needed to produce interactive three-dimensional applications. OpenGL has a built in selection mechanism that allows users to select then modify objects from the screen and manipulate them. More details of OpenGL can be referred to *OpenGL Programming Guide or the "Red Book"*[27].

2.4.2 Extraction of Surface Patches Containing Individual Tooth Surface

Our application should allow the user to identify objects on the screen and then to move, modify, delete or otherwise manipulate those objects. Since objects drawn on the screen typically undergo multiple rotations, translations, and perspective transformations, it is difficult to determine which object a user is selecting in a 3-

dimensional scene. OpenGL provides a selection mechanism that automatically indicates which objects are drawn inside a specific region of the window.

Typically in our case, when we are trying to use the OpenGL selection mechanism to extract the surface patches from the tooth model, first we draw our scene into the frame buffer and then enter selection mode and redraw the scene. Once in the selection mode, however, the contents of the frame buffer don’t change until we exit selection mode. When exiting, OpenGL returns a list of primitives (in our case, triangles) that would intersect the viewing volume. Each primitive (triangle in our case) that intersects the viewing volume causes a selection *hit*. The list of triangles is actually returned as an array of integer-valued *names* and related data—the *hit records*—that correspond to the current contents of the *name stack*. In our selection application, each triangle of the tooth model is named with an integer number from 1 to n, where n is the number of triangles in the tooth model. Then we construct the *name stack* by loading names onto it as we issue triangle-drawing commands while in selection mode. Thus, when the list of names is returned, we can use it to determine which triangle might have been selected on screen. Fig 2.3 shows us one surface patch containing an individual tooth surface. The surface patch is in red color for better visualization. Fig 2.4 shows the surface patch extracted from the tooth model. With the surface patch available, we can go on to manually segment the individual tooth surface.

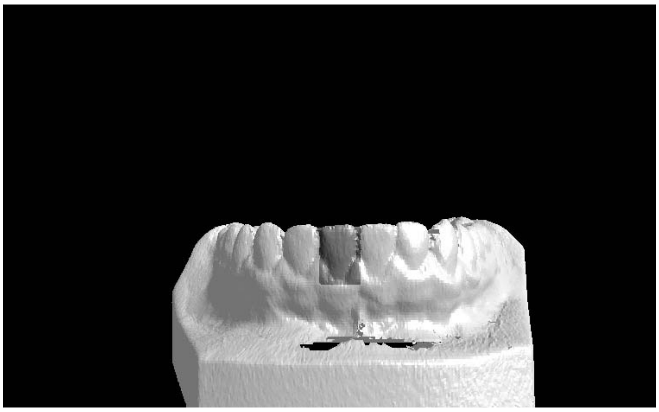


Fig 2.3 Extraction of surface patch containing an individual tooth surface

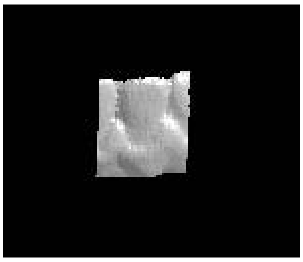


Fig 2.4 The surface patch extracted from the tooth model

2.4.3 Manual Segmentation of Tooth Surface

There are two steps in manually segmenting the tooth surface from the tooth surface patch. Firstly, numerous points are selected along the edges of the tooth surface; the point selection process is also in the OpenGL selection mode and the selected points are saved in the name stack. Because in our selection application, only graphic primitives can be selected and saved in the name stack, in our case, the triangles, the center points of the selected triangles in the name stack are saved as the edge points of the tooth surface. Fig 2.5 shows how the edge points are selected in our application.

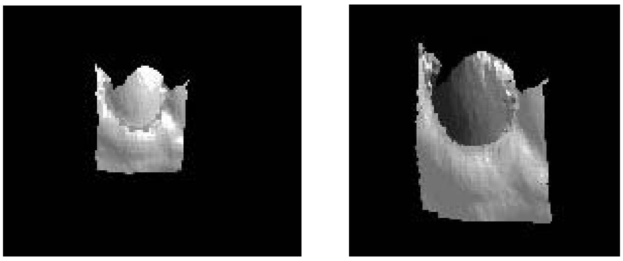


Fig 2.5 points selection along the edge of the tooth surface

Secondly, the triangles that are contained inside the edges of the tooth surface are extracted and saved as the individual tooth surface. Fig 2.6 shows us the segmented tooth surface.

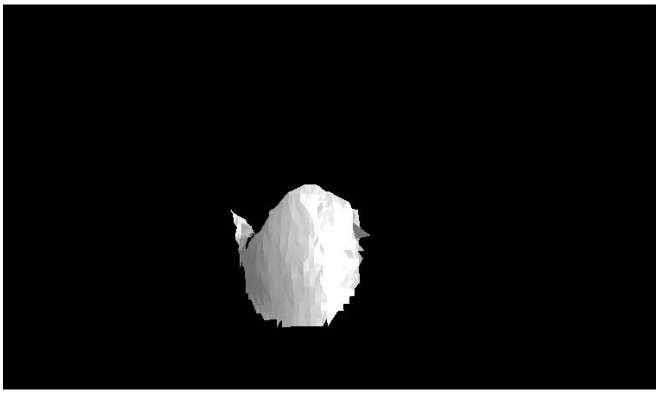


Fig 2.6 Segmented tooth surface

CHPATER 3

VISUALIZATION OF TOOTH MODELS AND TOOTH BRACKET SURFACE

Visualization of tooth models and tooth bracket surfaces is of great importance in helping the orthodontists with their diagnoses and treatment. In this work, tooth models and tooth brackets are scanned using laser scanners, which enables good visualization results. We use OpenGL as the main interface to visualize the 3D objects, some details about OpenGL are briefed in 2.4.1.

3.1 3D Data Acquisition System

We use Cyberware 3D scanner Model *3030 HIREZ* as the 3D data acquisition system and the motion platform Model *MM* for the scanning of tooth models and Mahr OMS 400 Multi-Sensor Coordinate Measuring Machine for the scanning of tooth brackets. Using active range finding technique, this Cyberware 3D data acquisition system is capable of giving 3D scans with high resolution and accuracy. The specifications of this system are found to be acceptable for use in the study. This section briefly describes the principle behind the 3D scanner Model *3030 HIREZ* and gives the specification of the 3D data acquisition system. The last part touches on the data format of the digitized 3D data.

3.1.1 Cyberware 3D Digitizing System

Cyberware Model *3030 HIREZ* is a rugged, self-contained optical range-finding scanner whose high sensitivity accommodates varying lighting conditions and surface properties. Together with the Cyberware motion platform Model *MM* which can translate and/or rotate the object to enable the scanner to capture different viewpoints, the 3D scanner can capture the shape of the entire object. The scanning process and the movement of the motion platform are performed entirely under software control

> Model *3030 HIREZ* operates on the principle of triangulation to obtain range images. Triangular meshes are then created from these images for surface rendering. When the object is scanned in different *orientations*, *registration* is required to merge the data obtained for the different orientations. The scanning of an object in different orientations is necessary because the motion platform does not allow six degrees of freedom. It offers only translation and rotation around one axis. Typically, cylindrical and translational scans are taken from the object. To capture the top and underside surfaces of the object, the object has to be re-orientated on its side to “expose” these surfaces. Subsequently, another set of cylindrical and translational scans are taken again. To match the object data from the two different orientations, registration is performed. The creation of the triangle meshes and registration of the range images form an area of active research.

3.1.2 Active Optical Triangulation

Active optical triangulation is one of the most common methods for acquiring range data. The basic principle is simple. The scanner has an illuminant which projects a pattern of light on the object. This pattern of light will be observed by a sensor that is off axis to the direction of light, as shown in Fig 3.1. Knowing the positions of the illuminant and the sensor, the intersection of the projected light direction and the sensor viewing direction gives the "depth" (distance of object from the scanner) value.

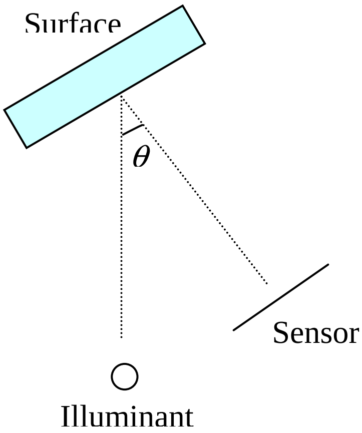


Fig 3.1. Optical triangulation geometry. The angle θ is the triangulation angle.

In the Cyberware Model *3030 HIREZ* scanner, the illuminant is a stripe of laser light. This laser stripe is created by spreading the laser beam using a cylindrical lens. The light that is reflected off the object will be captured by a 3D charged-coupled device (CCD) matrix. The accuracy of the range data depends on the proper interpretation of the imaged light reflections. The problem is locating the center of the sensor, which should map to the center of the illuminant, on the imaged data. Typically, statistical parameters such as mean, median or peak of the imaged light has been used as representative of the center. Once a proper interpretation is chosen the CCD image will show the depth data as imaged by the laser stripe. A typical CCD image is shown in Fig 3.2. Combining multiple frames of the CCD images, as the object is moved through the laser stripe, gives the full range image.

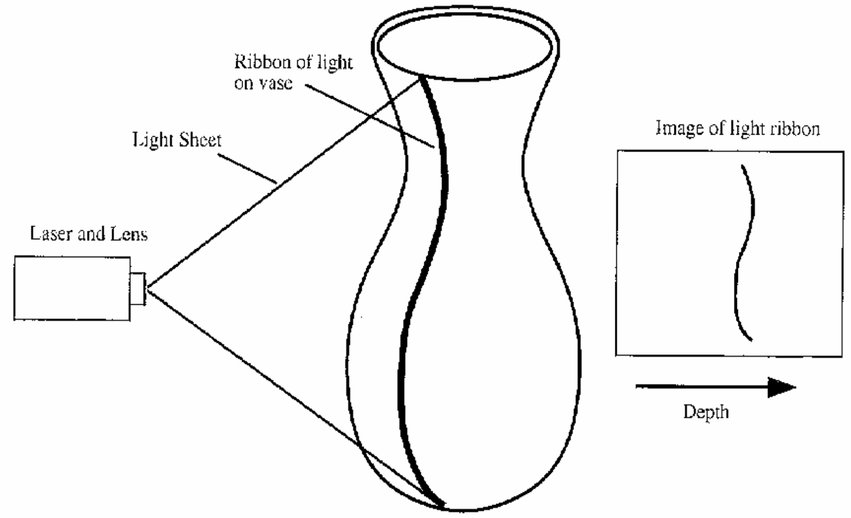


Fig 3.2. A typical CCD image.

3.1.3 Specifications of the Scanner System

The Cyberware 3D data acquisition system consists of two hardware components: the scanner and the motion platform. The object of interest, the dental cast, is small enough to fit in the field of view of scanner. The specification that is of interest is the spatial resolution. The scanner is capable of digitizing the shape of an object in (*x*, *y*, *z*) co-ordinates. The spatial resolution is given as follows:

x	:	0.5mm to 2mm, depends on platform speed
y	:	0.313mm
z	:	0.05 to 0.2 mm, depends on surface quality

The spatial resolution given for the *x*-axis can actually be decreased to obtain a finer resolution. The *x*-axis is along the translational direction of the motion platform and hence resolution is determined by the platform speed. By taking rotational scans, the *x*-axis resolution can be reduced as illustrated in Fig 3.3. The magenta line shown in Fig. 3.3 indicates the laser stripe during the first scanning process. After the first scan, the rectangle object is rotated and scanned the second time. The green line denotes the laser stripe during the second scan. The results of the two scans show that the x-axis resolution is reduced but not uniform. The resolution enables a reasonable digitization of the dental plaster cast.

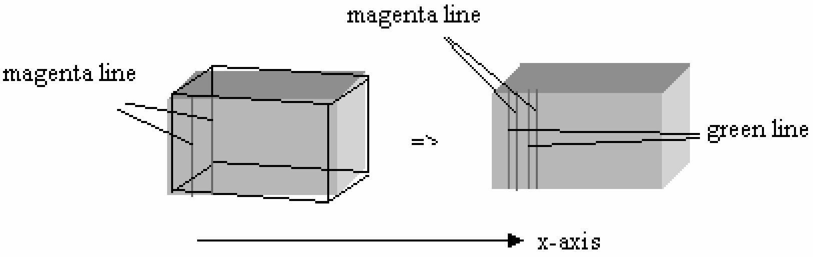
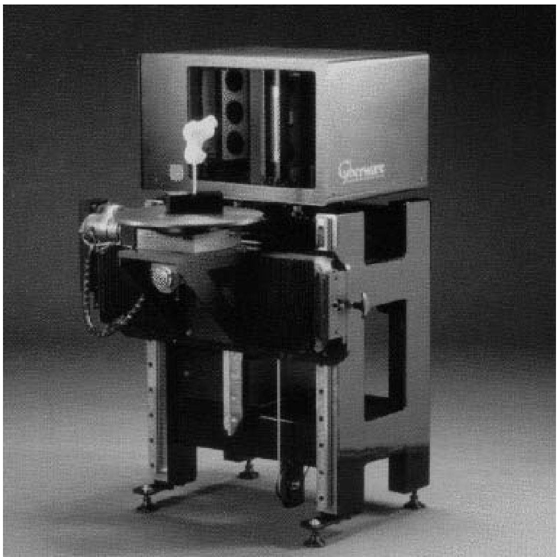


Fig 3.3. ***x*-axis resolution reduction through rotational scans.**

Together with the motion platform model *MM* which is capable of translational and rotational movements around one axis, the scanner system is able to scan the shape of the entire dental cast. The 3D (x, y, z) data is transferred to the Silicon Graphics IRIS



----

workstation via an Ethernet link. Fig 3.4 shows a picture of the Cyberware 3D data acquisition system.

Fig 3.4. Cyberware 3D data acquisition system.

<u>3.1.4 3D Data Format</u>

After digitization, the *raw* output of the scanner are data *points* defined in terms of the Cartesian coordinates (*x*, *y*, *z*). The concept of triangular meshes where the surface of the object is represented by many smaller triangular meshes (3 points to form a plane, i.e. surface) is used for surface rendering in the Cyberware system. Hence the Cybeware system gives the scanned output data in terms of the (*x*, *y*, *z*) points set and the set of triangular meshes.

odel *3030 HIREZ*   Cyberware Motion Platform Model *MM*

The shape of the object is defined by the Cartesian points which form the raw data. The surface information in terms of the triangular meshes will only be useful for processing purpose if their normal vectors (vectors perpendicular to the triangular meshes) are computed. A normal vector of a triangular mesh basically indicates the direction that the mesh is pointing to. In this study, the points form the crucial information and techniques were designed to work on them.

In our work, we scanned the dental casts using the Cyberware 3D data acquisition system. But the tooth brackets are too small to be scanned by the Cyberware 3D data acquisition system; we used the OMS 400 Multi-Sensor Coordinate Measuring Machine to scan the tooth brackets.

3.1.2 Mahr OMS 400 Multi-Sensor Coordinate Measuring Machine

Mahr OMS 400 Multi-Sensor Coordinate Measuring Machine (OMS 400) is a lab grade multi-sensor system with quality lab standard, high precision and rapid measure speed. OMS 400 integrates with three measuring sensors: optical, laser and touch probe. Together with NT software, OMS 400 can provide 3D assessment of all kinds of parts and applications. The scanning process and the movement of the motion platform are performed entirely under software control.

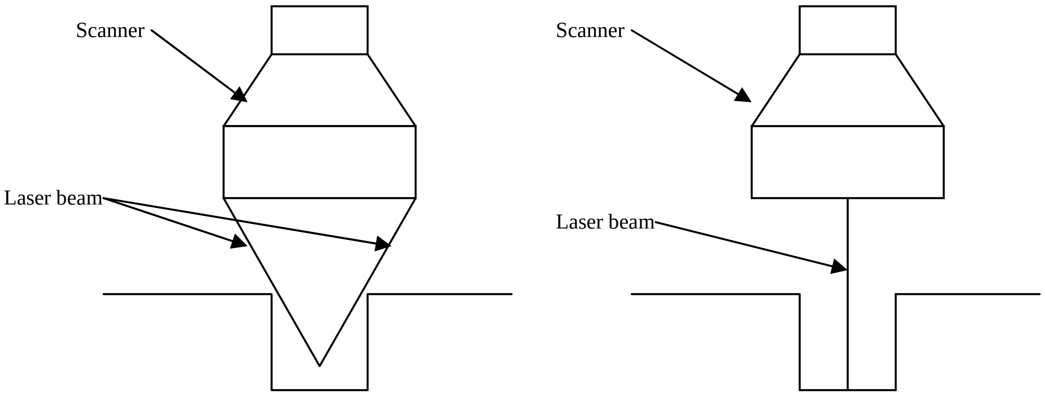


Fig 3.5 (a) Conventional Laser (b) MAHR Laser

The MAHR LASER system and the optical sensor have a common optical path guaranteeing offset free measurements between the two sensors. The MAHR LASER operates with a programmable intensity control which adapts the laser to the various material surfaces and enables measurements even on polished glass, ceramic or metal surfaces where conventional triangular laser systems fail. Fig 3.5 shows a conventional laser and a MAHR laser. Fig 3.6 shows a picture of the MAHR OMS 400 Multi-Sensor Coordinate Measuring Machine.

Fig 3.6 MAHR OMS 400 Multi-Sensor Coordinate Measuring Machine

In our work, the object of interest, the tooth bracket, is small enough to fit in the field of the view of the scanner. After digitization, the *raw* output of the scanner are data *points* defined in terms of the Cartesian coordinates *(x, y, z)*. Unlike the Cyberware system, which can give out the data points and the triangular meshes of the surface, the OMS system only gives out the data points.

3.2 Visualization of Tooth Models and Tooth Bracket Surfaces

3.2.1 Visualization of Tooth Models

We developed a *VRML* based environment of visualization and selection. The environment is also capable of the functions to allow a better view of the tooth models: rotation, zoom (in and out), and translation if necessary. Fig 3.6 and Fig 3.7

visualize a tooth model in high resolution and low resolution. The difference between high resolution and low resolution is that the tooth model scanned in high resolution has more triangles, which gives more detailed quality.

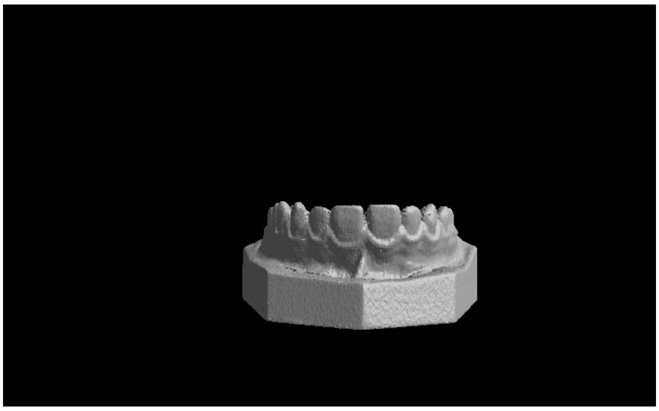


Fig 3.6 High-resolution tooth model

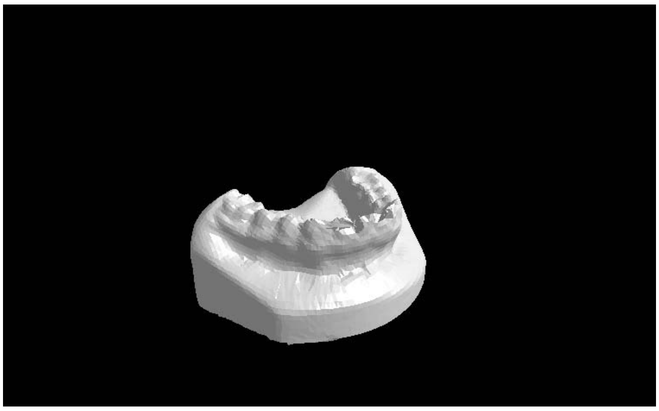


Fig 3.7 Low-resolution tooth model

3.2.2 Visualization of Tooth Bracket Surfaces

The laser scanner in 3.1.2 can only scan individual points on the surface in lines from left to right and from top to bottom. The output file of the scanning gives us the x, y, z coordinates of the points on the tooth bracket surface. Fig 3.8 shows a typical tooth bracket surface.

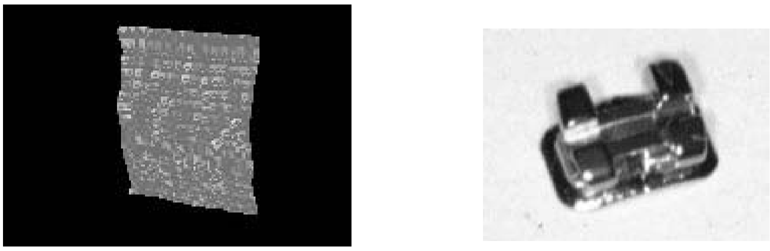


Fig 3.8 A typical tooth bracket surface and a tooth bracket

In this chapter, the 3D data acquisition system is described. The visualization of tooth models and tooth bracket surfaces is also described. In next chapter, we will go down to the construction of the Harmonic Map of the surface patches (in our case, tooth surface patch and tooth bracket surface patch) and later on to the similarity comparison of tooth surface and tooth bracket surface.

CHAPTER 4

GENERATION OF HARMONIC SHAPE IMAGES

In this thesis, harmonic maps are used to conduct surface matching between a single tooth brackets surface and a single tooth surface. The idea of using harmonic maps to conduct surface matching is partly inspired by the work in the computer graphics field done by Zhang Hebert [26]. In order to compare the similarity of the two surfaces, Harmonic Shape Images of the two surfaces, are created using harmonic maps. In this chapter, we discuss the background, and the core steps of the generation process, interior mapping, boundary mapping, and different schemes for approximating the curvature at each surface vertex, will be discussed next in detail in the following sections.

4.1 Harmonic Maps

A map $u : M \to N$, between two compact *Riemannian manifolds,* is a harmonic map if it is a critical point for the energy functional

$$\int_M |du|^2 d\mu_M \, .$$

The norm of the differential $|du|$ is given by the metric on *M* and *N* and $d\mu_M$ is the measure on *M*. Typically the class of the allowable maps lies in a fixed homotopy class of maps. The Euler-Lagrange differential equation for the energy functional is a non-linear elliptic partial differential equation. For example, when *M* is the circle, then the Euler-Lagrange equation is the same as the geodesic equation. Hence, *u* is a

----

closed geodesic if *u* is harmonic. The map from the circle to the equator of the standard 2-sphere is a harmonic map, and so are the maps that take the circle and map it around the equator *n* times, for any integer *n*. Note that these all lie in the same homotopy class. A higher dimensional example is a meromorphic function on a compact Riemann surface, which is a harmonic map to the Riemann sphere. A harmonic map may not always exist in a hotomopy class, and if it does it may not be unique.

A harmonic map between Riemannian fields can be viewed as a generalization of a geodesic when the domain dimension is one, or of a harmonic function when the range is a Euclidean space. The theory of harmonic maps studies the maps between two manifolds from an energy point of view. Formally, let *(M,g)* and *(N,h)* be two smooth manifolds of dimensions *m* and *n* respectively, and let ϕ : *(M,g)*$\rightarrow$*(N,h)* be a smooth map. Let $\left(x^i\right)$, $i = 1,\ldots, m$ and $\left(y^\partial\right)$ $\partial = 1,\ldots, n$ be local coordinates around x and $\phi(x)$, respectively. Take $\left(x^i\right)$ and $\left(y^\partial\right)$ of *M* and *N* at corresponding points under the map ϕ whose tangent vectors of the coordinate curves are $\partial/\partial x^i$ and $\partial/\partial y^\alpha$, respectively. Then the energy density of ϕ is defined as

$$\ell(\phi) \quad = \frac{1}{2}\sum_{i,j=1}^{m} g^{ij} \sum_{\alpha,\beta=1}^{n} \frac{\partial\phi^\alpha \partial\phi^\beta}{\partial x^i \partial x^j} h_{\alpha\beta}(\phi) \qquad \textbf{(}4.1.1)$$

In the equation above, g_{ij} and $h_{\alpha\beta}$ are the components of the metric tensors in the local coordinates on *M* and *N* respectively. The energy of ϕ in local coordinates is given by the number

$$E(\phi) \quad = \sum e(\phi) v_g$$

----

if ϕ is of class C^2, $E(\phi) < \infty$, and ϕ is an extremum of the energy, then ϕ is called a harmonic map and satisfies the corresponding Euler lagrange equations. In the special case in which *M* is a surface *D* of disc topology and *N* is a convex region *P* in E^2, the following problem has a unique solution.

4.2 Interior Mapping

The theory of harmonic maps has been briefly reviewed in section 4.1. It is clear that the solution to harmonic maps is the solution to a partial differential equation. Our work involves discrete tooth surfaces and tooth bracket surfaces, it is clear that the solution to harmonic maps of surface patch is the solution to a partial differential equation. Because the computation cost of a solution to a partial differential equation is so high, it would be more appropriate and practical that some approximations be made to compute the harmonic maps.

Let $D(v, R)$ be a 3D surface patch (the definition of 3D surface patch can be found in [26]) with central vertex *v* and radius *R* measured by surface distance. The computation of surface distance is a non trivial and will be discussed in the following sections. Let *P* be a unit disc in a two-dimensional plane. Let ∂D and ∂P be the boundary of *D* and *P*, respectively. Let v_i, $i = 1,.....n$, be the interior vertices of *D*. The interior mapping ϕ maps $v_{i,}$ $i = 1,.....n$, onto the interior of the unit disc *P* with a given boundary mapping *b*: $\partial D \to \partial P$, ϕ is obtained by minimizing the following energy functional.

$$E(\phi)=\frac{1}{2}\sum_{\{i,j\}\in Edges(D)} k_{ij}\left\|\phi_i-\phi_j\right\|^2 \qquad (4.2.1)$$

In (3.2.1), for the simplicity of notation, ϕ_i and ϕ_j are used to denote $\phi(v_i)$ and $\phi(v_j)$ which are the images of the vertices v_i and v_j on *P* under mapping ϕ. The values of ϕ_i and ϕ_j define the mapping ϕ and k_{ij} serve as spring constants which will be discussed shortly.

An instance of the function $E(\phi)$ can be interpreted as the energy of a spring system by associating every edge in *D* with a spring. Then the mapping problem from *D* to *P* can be considered as adjusting those springs when flattening them down onto *P*. If the energy of *D* is zero, then the energy increases when the mesh is flattened down onto *P* because all the springs are deformed. Different ways of adjusting the spring lengths correspond to different mappings ϕ. The best ϕ minimizes the energy functional $E(\phi)$.

The minimum of the energy functional $\mathrm{E}(\phi)$ can be found by solving a sparse linear least-square system for the values $\phi(i)$. Taking the partial derivative of $\mathrm{E}(\phi)$ with respect to $\phi(i)$, $i=1,.....n$ and make it equal to zero yield the following equations:

$$\frac{\partial E(\phi)}{\partial \phi(i)}=k_{ij}\left(\phi(i)-\phi(j)\right)+k_{ik}\left(\phi(i)-\phi(k)\right)+k_{il}\left(\phi(i)-\phi(l)\right)+....,$$

$$=\sum_{i,j\}\in 1-Ringof\, i\}} k_{ij}\left(\phi(i)-\phi(j)\right)=0, i=1,...,n \qquad (4.2.2)$$

The equation (4.2.2) can be rewritten as:

$$A_{nxn}X_{nx2}=b_{nx2} \qquad (4.2.3)$$

In equation (4.2.3), $X_{nx2} = [\phi(1),.....\phi(n)]^T, \phi(i) = [\phi(x_i),\phi(y_i)]$. X_{nx2} denotes the unknown coordinates of the interior vertices of D to P under mapping ϕ. Because A_{nxn} contains the connectivity information of the surface patch $D(v,R)$, it is a sparse matrix which has the following structure.

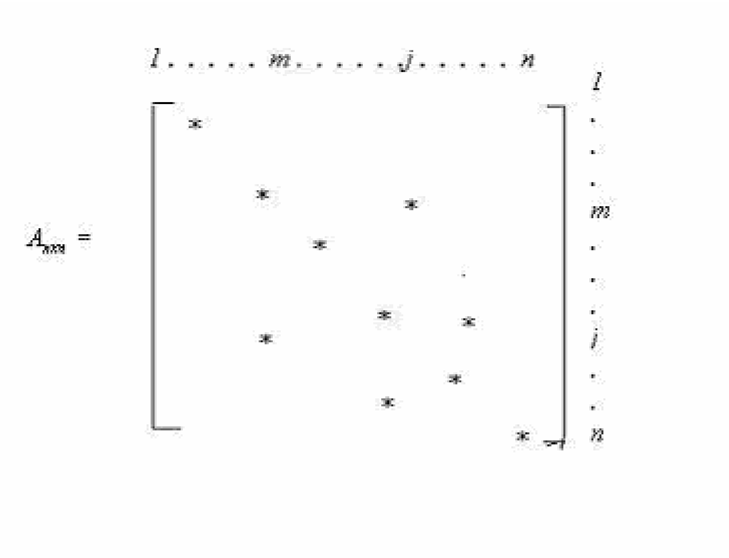


Fig 4.1 Structure of the Matrix $A_{n \times n}$

If D is to be considered as a bi-directional graph (will be discussed shortly), A_{nxn} can be interpreted as the adjacency matrix of D. All the diagonal entries of A_{nxn} are non zero. For an arbitrary row i in A_{nxn}, if vertex v_i is connected to vertices v_j and v_m, then only the *jth* and *mth* entries in row i are non zero. Similarly, the *ith* entries in row j and row m are also non zero.

The boundary conditions are accommodated in the matrix b_{nx2}. In X_{nx2}, if a vertex v_i is connected to boundary vertices, then its corresponding entry i in b_{nx2} is weighted by the coordinates of those boundary vertices. Otherwise the entries in b_{nx2} are zero.

A detail example of the entries in b_{nx2}, A_{nxn} and X_{nx2} will be given in the computation process of the harmonic map of an individual tooth surface shortly in the following sections.

The computation of the above equations requires the construction of D into a bi-directional graph. The adjacency list structure is selected to construct the graph. Matrix A_{nxn} is constructed according to this adjacency list.

There are different ways of defining the spring constants k_{ij} in equation (4.2.1). One way to define k_{ij} is as in equation (4.2.4)[26]

$$k_{ij} = ctg\theta\left(e_{mi}, e_{mj}\right) + ctg\theta\left(e_{li}, e_{lj}\right) \qquad (4.2.4)$$

in which $\theta\left(e_{mi}, e_{mj}\right)$ and $\theta\left(e_{li}, e_{lj}\right)$ are defined in Fig 4.2.

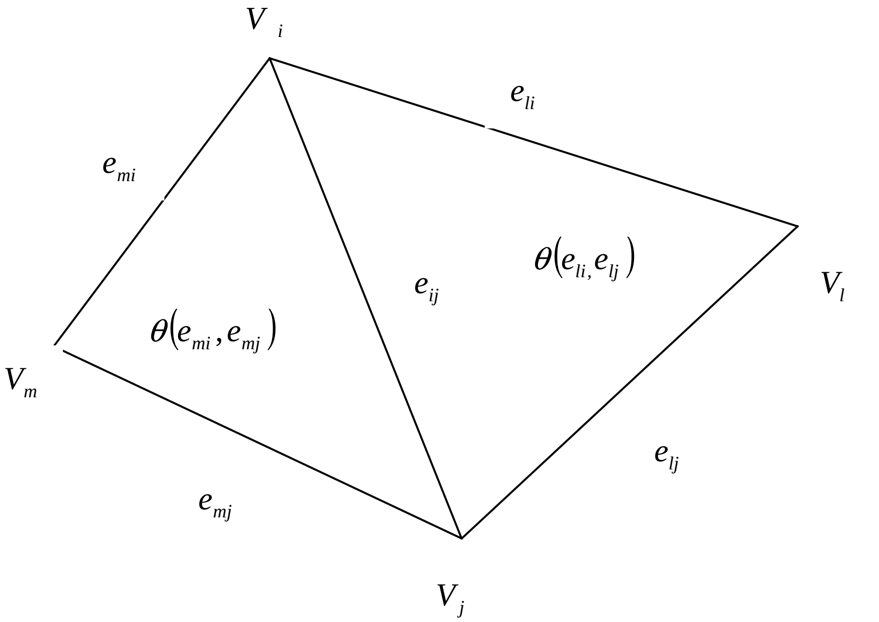


Fig 4.2 Definition of spring constants using angles

The intuition behind this definition is that long edges subtending to big angles are given relatively small spring constants compared with short edges that subtend to small angles. This definition of the spring constants also means that long edges will remain long in the harmonic map image while short edges will remain short. From the discussion above, it can be seen that the energy functional (4.2.1) tries to preserve the ratio of edges lengths on the original surface patch $D(v, R)$ by defining the spring constants as in (4.2.4). Given a specific sampling of the surface patch (such as the *WRL* file format which we use to represent the tooth surface patch and the tooth bracket surface patch), the ratio of edge lengths is closely related to the shape of the surface patch. Thus, by preserving the ratio of edge lengths, the interior mapping ϕ preserves the shape of the surface match when mapping it down to unit disc *P*. Inevitably, there will be a distortion when mapping $D(v, R)$ on to *P*. Best mapping ϕ minimizes this distortion.

In order to preserve the ratio of edge lengths, another way to define the spring constants is to use the inverse of the edge length as shown in equation (4.2.5).

$$k_{ij} = \frac{1}{\left\| e_{ij} \right\|} \qquad (4.2.5)$$

As in equation (4.2.4), the springs associated with long edges will have smaller spring constants compared with the springs associated with short edges.

In [28], it has been shown that there is little difference between the harmonic shape images resulted from the two approaches to defining the spring constants and no evidence has been found that one approach is better than the other. In our project, the inverse of edge lengths (4.2.5) is used because of its simplicity and convenience to work in programming.

4.3 Boundary Mapping

As discussed in section 4.2, the unit disc *P* is selected to be the target domain of the harmonic mapping. In this section, the construction of boundary mapping, which maps the boundary of $D(v, R)$ onto the boundary of *P* is discussed.

The construction of boundary mapping is illustrated in Fig.4.3. The vectors and vertices in this figure should be defined first here. *O* is the central vertex of the surface patch $D(O, R)$ and $O^{'}$ is the central vertex of the unit disc *P*. $v_{i,}, i = 1,....5$ are the boundary vertices of the surface patch $D(O, R)$. For some boundary vertices, for example, $v_i, i - 1,...4$, the distance between any of them and the central vertex *O* is equal to R; these vertices are called *radius boundary vertices.* For other vertices, such as v_5, the distance between any of them and the central vertex *O* is less than R: these vertices are called *occluded boundary vertices. Radius boundary vertices* are determined by the radius of the surface patch, while *occluded boundary vertices* are determined either by self-occlusion or by occlusion by other objects. In our practice, there is clearly no occlusion by other objects, so *occluded boundary vertices* are determined by self-occlusion. The vector $v_{\substack{r \\ i}}$ from the central vertex *O* to a *radius boundary vertex*, e.g. $v_i, i - 1,...4$, is called a radius vector, while the vector v_j^o from the central vertex *O* to an *occluded boundary vertex*, e.g. v_5, is called an occlusion vector.

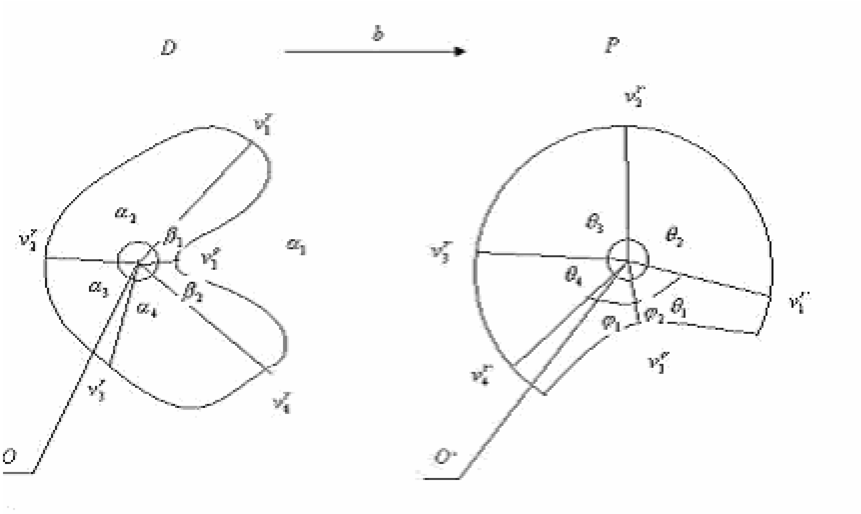


Fig 4.3 Illustration of boundary mapping

The angles in Fig 4.3 are defined as followed. Angles $\alpha_i, i = 1,...4$ are the angles between two adjacent radius boundary vectors $v_i^r O$ and $v_{i+1}^r O$. Angles $\beta_j, j = 1,2$ are the angles between two adjacent occlusion boundary vectors, or one occlusion boundary vector and one adjacent radius boundary vector, in an occlusion range. An occlusion range is an occlusion vector sequence, except for the first vector and the last vector. For example, (v_1^r, v_1^o, v_4^r) in Fig.4.3 is an occlusion vector sequence. The sum of β_j over an occlusion sequence is the angle α_i formed by the first and last radius boundary vectors, for example, the sum of β_j over (v_1^r, v_1^o, v_4^r) is α_1.

The construction of the boundary mapping is made up of two steps. At the first step, the radius boundary vertices are mapped onto the boundary of the unit disc *P*. In Figure 4.3.1, vectors $v_i^r, i = 1,...4$ are mapped to $v_i^{r'}, i = 1,...4$. It is quite clear that once the angles θ_i are determined, the positions of $v_i^{r'}, i = 1,...4$ are easily computed. θ_i is computed as follows:

$$\theta_i = \frac{\alpha^i}{\sum_{k=1}^{n} \alpha_k} 2\pi \tag{4.3.1}$$

At the second step, the occlusion boundary vertices are mapped onto the interior part of the unit disc *P*. In Fig.4.3, vector v_1^o is mapped to $v_1^{o'}$. It can be shown that once the angles φ_j and the radii r_j are determined, the positions of v_j^o are easily computed. φ_j Is computed as follows:

$$\varphi_j = \frac{\beta_j}{\sum_{m=1}^{n} \beta_m} \alpha_i \tag{4.3.2}$$

in which n is the number of angles within the occlusion range, α_i is the angle corresponding to the occlusion range. r_j is defined as follows:

$$r_j = \frac{dist(r_j, O)}{R} \tag{4.3.3}$$

In which $dist(r_j, O)$is the surface distance between the occlusion vertex v_j^o and the central vertex *O*. *R* is the map radius of the surface patch.

The intuition behind the boundary mapping is that the ratio of the boundary edge lengths is preserved as much as possible on the boundary of *P*. This is consistent with the way interior mapping is constructed. Therefore, the harmonic mapping minimizes the shape distortion when mapping the surface patch $D(v, R)$to the unit disc P. In Figure (4.3), angles $\alpha_i, i = 1,...4$ and angles $\beta_j, j = 1,2$ are surface angles. The computation of $\alpha_i, i = 1,...4$ and $\beta_j, j = 1,2$ is non trivial job, so a simplification is conducted in our project to make the computation easier. For any arbitrary boundary vertex, either a radius boundary vertex v_{r_i}, or a occlusion boundary vertex v_j^o, the positions of the corresponding mapping points on the unit disc *P*, $v_i^{r'}$ or $v_j^{o'}$ can be computed as follows:

$$coordx = \frac{distx(v_i, O) \times dist(v_{i,}O)}{Edist(v_i, O) \times R} \qquad (4.3.4)$$

$$coordy = \frac{disty(v_i, O) \times dist(v_i, O)}{Edist(v_i, O) \times R} \qquad (4.3.5)$$

in which $coordx$ and $coordy$ are the *x* and *y* coordinates of the mapping points onto the boundary of the unit disc *P*. $Edist(v_i, O)$ is the Euclidean distance between the boundary vertices of the surface patch $D(v, R)$ and the central vertex O. $distx(v_i, O)$ and $disty(v_i, O)$ is defined as follows:

$$distx(v_i, O) = \left\| v_i.x \;\; - O_{.}x \right\| \qquad (4.3.6)$$

$$disty(v_i, O) = \left\| v_i.y - O.y \right\| \qquad (4.3.7)$$

in which $v_{i.}x$ and $v_{i.}y$ are the *x* and *y* coordinates of the boundary vertices of the surface patch $D(v, R)$ respectively. $O.x$ and $O.y$ are the *x* and *y* coordinates of the central vertex O respectively.

The interior mapping and boundary mapping mentioned in section 4.2 and section 4.3 require the construction of the bi-directional graph of the surface patch $D(v, R)$ and the computation of surface distance between two arbitrary points on the surface patch. We will discuss these in detail in the following sections.

4.4 Bi-directional Graph of the Surface Patch *D(v,R)* and its Adjacency List

It is possible to represent graphs in computer memory with a variety of different of data structures. One strategy is to use an ***Adjacency Matrix***. An adjacency matrix is a two dimensional array in which the row and column headers represent different vertices in the graph. A one-way edge between two arbitrary connect vertices is denoted by a positive value in the corresponding array position. In a weighted graph the value stored in the each array location corresponds to the weight or cost of each particular edge. If an edge is bi-directional, it has two entries in the matrix. One entry represents the (source destination) route while the other handles the (destination source) return route.

Another method for representing graphs is as a more complicated ***Linked List*** structure. Each vertex in the graph is a node in a master linked list. Another linked list emanates from each vertex node and denotes the vertices directly adjacent to a given source vertex. This method, often called an **Adjacency List**, is more space efficient than the adjacency matrix for graphs, which do not have many edges.

In our work, the tooth surface meshes and tooth bracket surface meshes are stored as the *WRL* file format. As mentioned in the previous chapters, the surface meshes are represented by thousands of triangles. It is quite natural for us to construct a weighted bi-directional graph representation of the surface meshes. The vertices of the graph represent the vertices of the triangles in the surface mesh. The length of the corresponding triangle side denotes the weight of each edge. Adjacency list is used to represent the weighted bi-directional graphs of the surface meshes.

In order to represent the tooth surface mesh (or tooth bracket surface mesh), following rules are applied to create the weighted bi-directional graph:

1. Read in the *WRL* file representing the tooth surface
2. Search through the triangles that make up the surface mesh to obtain all the vertex nodes of the graph.
3. For each vertex node v_i of the graph, check the triangles of the surface mesh and find those triangles with one of its vertices being v_i. Check the sides of those triangles to find the adjacent vertex nodes of v_i, thus constructing the adjacency list of v_i.
4. Compute the lengths of the sides of those triangles and set them to be the weighted value of the corresponding edges.
5. After the construction of the weighted bi-directional graph, the surface distance of two arbitrary vertices on a given surface mesh is computed.

4.5 The Computation of Surface Distance of Two Arbitrary Vertices on a Given Surface Mesh

As mentioned before, the construction of boundary mapping and interior mapping requires the computation of surface distance of two arbitrary vertices on a given surface mesh. In our work, upon the construction of the weighted bi-directional graph, we can compute the central vertex node of the surface mesh. It can be shown that the construction of the boundary mapping requires the computation of the surface distance from the boundary vertex nodes to the central vertex node.

In our work, we conducted the surface distance computation in two different ways. If the resolution of the surface mesh is high enough to tolerate the error of Dijkstra Algorithm, the algorithm is applied in the construction of the harmonic maps, which gives good results [29,30]. For better accuracy, we applied another method ***z-coordinate*** **projection method**, which also gives satisfactory results.

Dijkstra Algorithm (named after its discover, E.W. Dijkstra) solves the problem of finding the shortest path from a point in a graph (the source) to a destination. It can be found that the shortest paths from a given source to all points in a graph in the same time, and hence this problem is sometimes called the *single-source shortest paths* problem.

4.5.1 *Z-coordinate* projection method

If the resolution of the surface mesh is not high enough, the computation of surface distance using Dijkstra Algorithm will not be so accurate. The *Z-coordinate* projection method overcomes the problem.

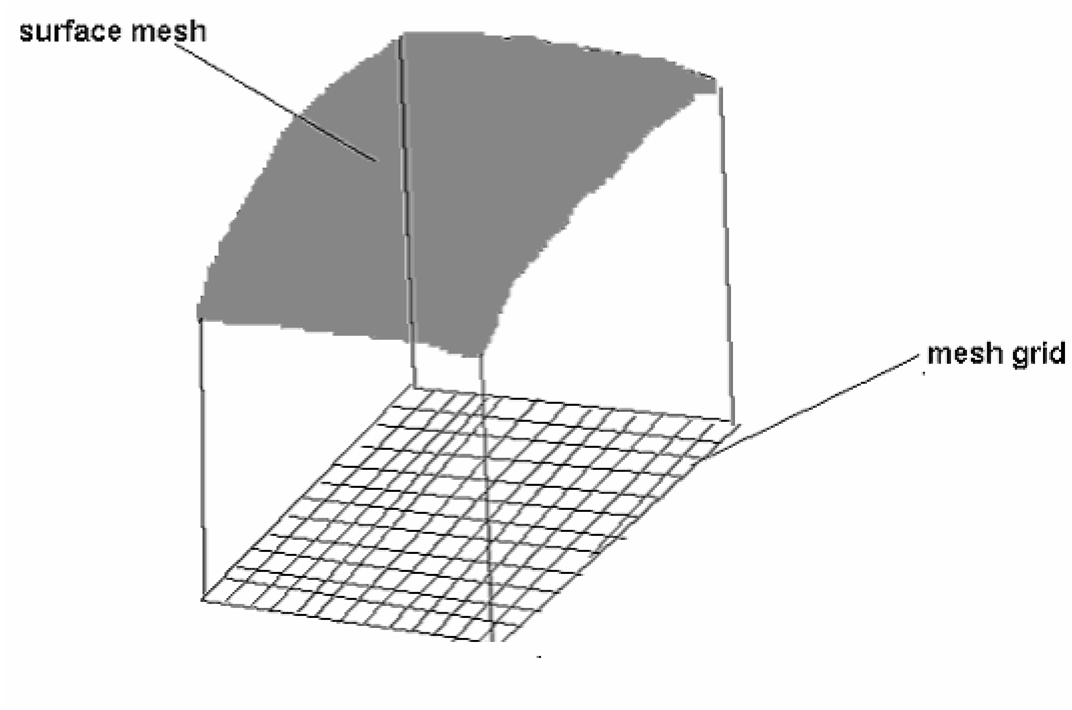


Fig 4.4 Mesh grid of the projection of surface mesh on a plane

As shown in Fig 4.4, the first step is to construct a mesh grid of the surface mesh's projection onto a plane. The mesh grid is constructed as follows: for an arbitrary grid point on the mesh grid, v, with v_x, v_y, v_z denotes the *x, y, z* coordinates respectively, the *z* coordinate preserves the curvature information of the surface mesh. The dimension of the mesh grid can be computed as:

$$x\_dimension = (\max x - \min x) \div resolution\_x$$

$$y\_dimension = (\max y - \min y) \div resolution\_y$$

Where $\max x$, $\max y$, $\min x$, $\min y$ denotes the max *x*, *y* coordinates and min *x*, *y* coordinates of the vertices on the surface mesh respectively. $resolution\_x$ and $resolution\_y$ denotes the resolution on the *x*, and *y* coordinates. For tooth bracket surface mesh, $resolution\_x$ is set to 0.1 and $resolution\_y$ is set to 0.2. For grid point v , suppose that v is the *ith* and *jth* point on the *x* and *y* coordinates respectively:

$$v_x = \min x + i * resolutionx$$

$$v_y = \min y + j * resolutiony$$

As for the computation of v_z, we can project v back to the surface mesh, based on the availability of v_x and v_y, and we can either choose the *z* coordinates of the nearest vertex on the surface mesh to projection or we can use the weighted value of the *z* coordinates of the *k* nearest points to the projection.

As for the surface distance from two arbitrary vertices on the surface mesh, i.e. m, n, Fig 4.5 gives the method to compute the surface distance.

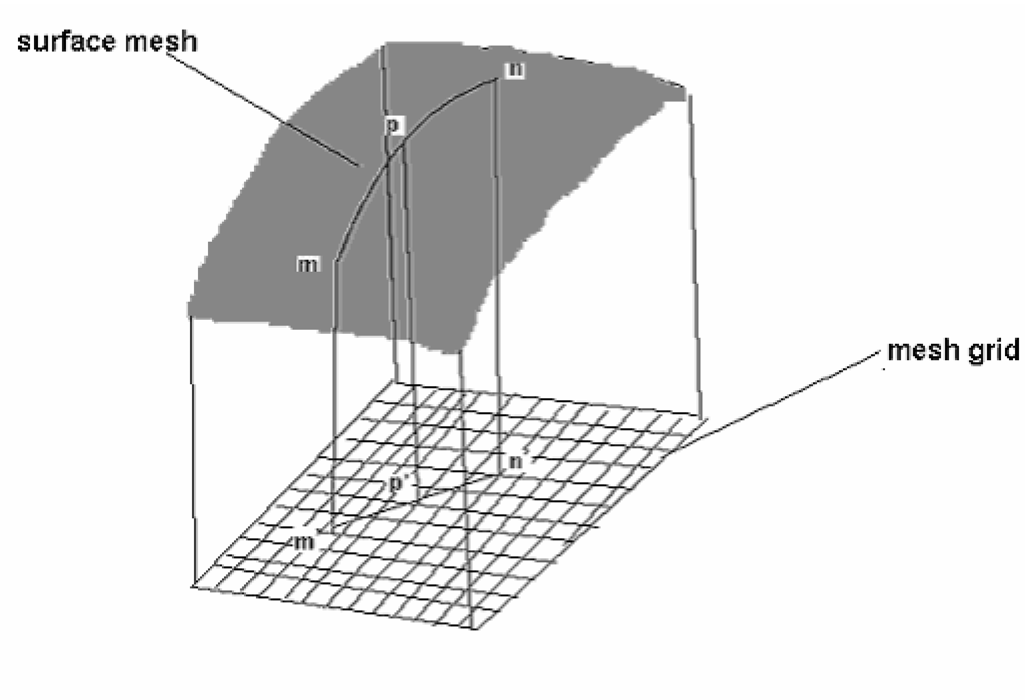


Fig 4.5 Computing the surface distance

In Fig 4.5, m , n are two arbitrary vertices on the surface mesh, m' and n' are the projection points of m and n onto the mesh grid respectively. p' is the center point of the line between m' and n'. The coordinates of p' can be easily obtained from the coordinates of m' and n'. p is the corresponding vertex of p' on the surface mesh.

The coordinates of p can be computed as mentioned above. So the surface distance between m and n can be approximated as:

$$Surfacedist(m,n) = Surfacedist(m,p) + Surfacedist(p,n)$$

If $Surfacedist(m,p)$ and $Surfacedist(p,n)$ are small enough, they can be replaced by the line distances between m , p and n , p respectively.

In our implementation, we divided the distance between m' and n' by 100. The surface distance computed is more accurate than the surface distance computed by Dijkstra Algorithm.

4.6 The Generation of Harmonic Shape Images

In earlier sections, we discussed how to define a surface patch and to obtain its harmonic image. In this section, we explain how to construct the *harmonic shape image* from its harmonic image.

As explained before, an important property of harmonic image is the one-to-one correspondence between the vertices on the surface patch $D(v,R)$ and those on its harmonic image $HI(D(v,R))$. This means that, all distribution functions defined on a surface patch can be recorded accordingly on its harmonic image. In other words, harmonic images provide a general framework for representing distribution functions defined on surfaces, e.g. surface normal, shape, color, texture. To describe the shape of free-form surfaces, shape information, e.g., curvature is stored at each vertex of the harmonic image $HI(D(v,R))$. The resultant image is called a *harmonic shape image*,

----

which is denoted as $HSI(D(v,R))$. If texture information is stored on $HI(D(v,R))$, the resultant image is called a *harmonic texture image*.

Curvature is a shape descriptor of surfaces. However, there are two difficulties of computing curvature. First we deal with discrete surfaces. The other is that computing curvature involves computing the second order derivatives, which are sensitive to noise. One way to address these issues is that smooth surface can be fit to the triangular mesh locally so that the curvature can be computed. This approach can be computationally expensive because of non-linear minimization. Another approach is to approximate the curvature distribution function using *discrete curvature*. In the following sub-sections, three different approaches for computing discrete curvature are discussed.

4.6.1 Simplex Angle

The first discrete curvature, *Simplex Angle* proposed in [31], is used to describe the shape at each vertex in a mesh with the appropriate regularities. A topology constraint on the mesh requires that each vertex has three neighboring vertices. Another requirement is that the projection of each vertex onto the place formed by its three neighboring vertices should coincide with the center of the triangle formed by those three vertices. Suppose P is a vertex on a given mesh M and P_1, P_2, P_3 are its neighboring vertices. The Simplex Angle at P is defined in Fig 4.6.

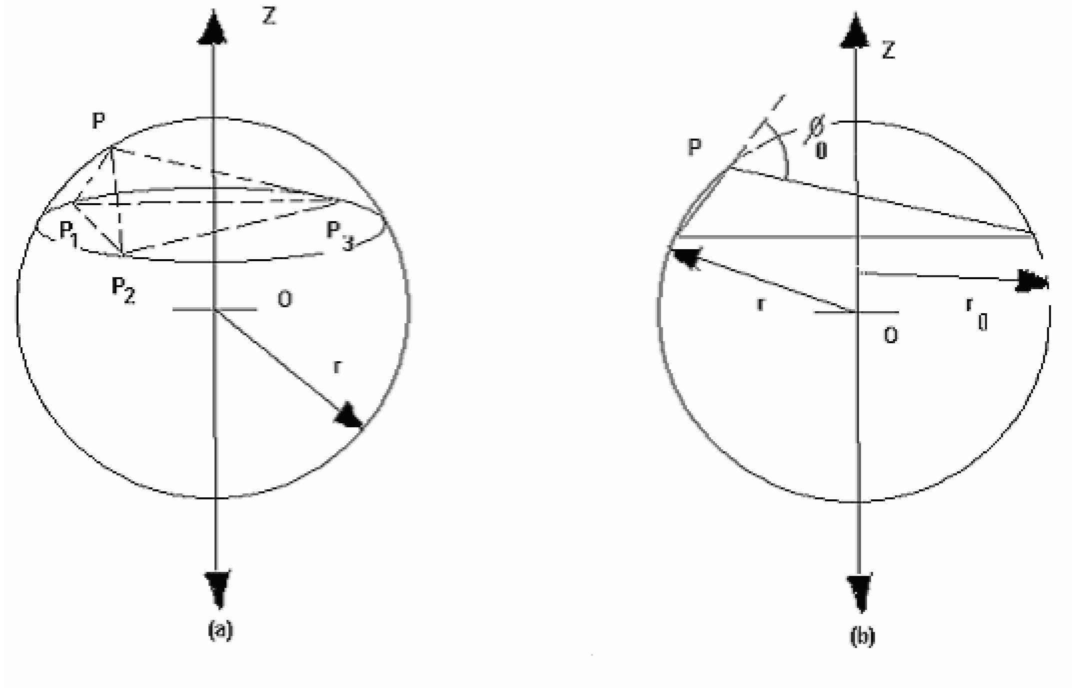


Fig 4.6 Definition of Simplex Angle

In Fig 4.6 (a), O is the center of the sphere circumscribed to the tetrahedron (P, P_1, P_2, P_3). Z is the line passing through the center O and through the center of the circle circumscribed to the tetrahedron (P, P_1, P_2, P_3). Consider the cross section of the surface by the plane $\prod$ containing Z and P. The intersection of $\prod$ with the tetrahedron is a triangle. One vertex of the triangle is P, and the edge opposite to P is in the plane formed by (P_1, P_2, P_3)(Fig.4.6 (b)). The angle ϕ_0, between the two edges of the triangle intersecting at P is defined to be the Simplex Angle at P.

One issue that should be noted when computing Simplex Angle is that, for arbitrary triangular meshes, the connectivities among the vertices are arbitrary, e.g., a vertex can have any number of neighboring vertices. However, in order to compute Simplex Angle, each vertex must have three neighboring vertices. In order to address the connectivity issue, it is necessary to compute Simplex Angle for the dual mesh of the given mesh. The dual mesh is obtained by taking the center of each triangle on the given mesh and connecting the centers of neighboring triangles as shown in Fig 4.7. The Simplex Angle for each vertex on the dual mesh angle can be computed as long as it has three neighboring vertices. In fact, it is the Simplex Angle for each face on the original mesh. The Simplex Angle at each vertex of the original mesh can be taken as the average of the Simplex Angles of the faces intersecting at the vertex.

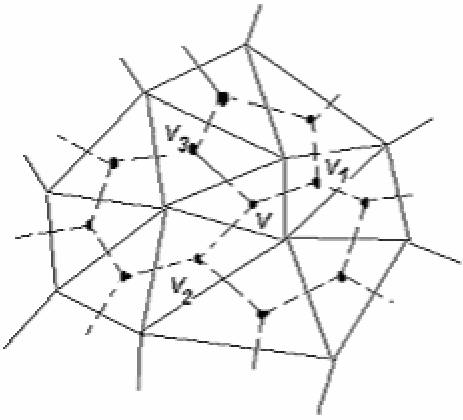


Fig 4.7. The duel mesh of a given triangular mesh. The solid lines represent the original triangular mesh. The dashed lines represent the duel mesh.

4.6.2 Complete Angle

The second method for computing discrete curvature is to use the *complete angles* [32] at each vertex of the triangular mesh (Fig 4.8). The complete angle at a vertex is defined to be

----

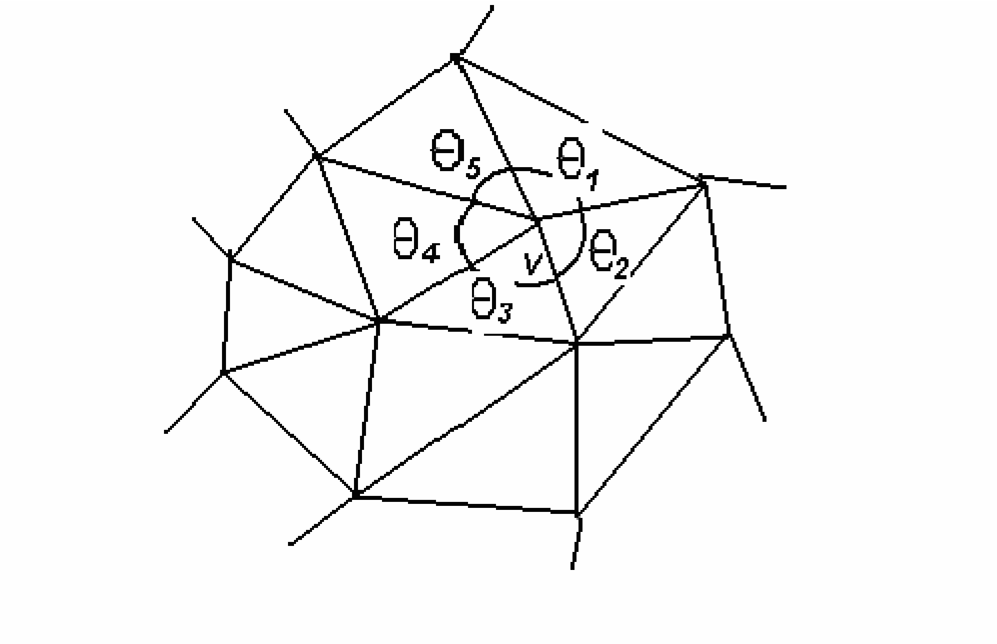


Fig 4.8. Definition of discrete curvature using complete angles

$$\theta = \sum_{i=1}^{m} \theta_i \tag{4.6.1}$$

in which m is the number of neighboring vertices around v. The discrete curvature c is defined to be

$$c = 2\pi - \theta \tag{4.6.2}$$

The definition in Fig.4.8 does not give the sign of the discrete curvature at v, i.e., it describes how much the surface is curved at v, but does not indicate whether the surface is locally convex or concave. Similar to computing the Simplex Angle at v, the sign of c can be determined by taking the average of the Simplex Angles of the faces intersecting at v.

Attention should be paid to the boundary vertices when computing the discrete curvature using Fig.4.8. Because the angles are not complete for the boundary vertices, their curvature values should be labeled invalid.

4.6.3 Weighted Dot Product of Normals

The third method for computing discrete curvature is to use area-weighted dot product of normals as illustrated in Fig 4.9. This discrete curvature is defined for each face of a given triangular mesh. For a face f with surface normal N, its local neighborhood with a certain size is defined first. Then for each triangle in this neighborhood, compute the dot product of its normal N_i with N and weight the dot product with its area A_i. The curvature of f is defined to be

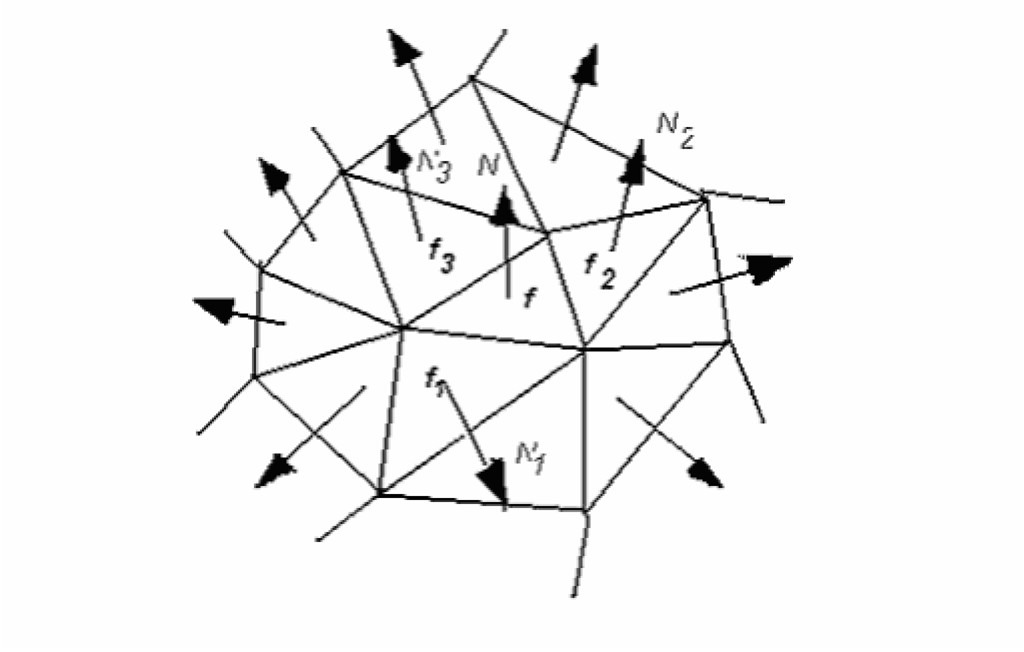


Fig 4.9. Illustration of defining discrete curvature using area-weighted dot product of normals.

$$c = \frac{\sum_{i=1}^{m} A_i (N_i \bullet N)}{\sum_{i=1}^{m} A_i} \qquad (4.6.3)$$

This definition can be interpreted using the concept of normal curvature. By taking the weighted average of the dot product of normals, the definition in equation (4.6.3) measures the change of the surface normal in all directions. Using the area of each triangle as the weight takes into account the sampling resolution. Similar to the definition in 4.6.2, this definition does not give the sign of the curvature. The sign can be determined using the same method as that used to compute the Simplex Angle. After the curvature is computed for each face on a given mesh, the vertex curvature values can be computed by taking the average of the values of the faces intersecting at that vertex. Again, triangle area can be used to weight the average.

Using the above three definitions for discrete curvature, the Harmonic Shape Images of the surface mesh can be computed. In [26], it can be seen that all three definitions reflect same shape features. Similar to the result found in [26] and [33], the Simplex Angle is better than the Complete Angle at differentiating shapes with small variation. In our implementation, because we deal with surface meshes with boundaries, we use weighted dot product of normals to construct the harmonic shape image.

4.7 Complexity Analysis

Suppose there are n_i triangles formed by n vertices on the input surface D. ∂ is the ratio between the boundary vertices and n. n_e is the number of edges in D. n_r denotes the number of resampling vertices in 2-D domain P. In general, n_r is about the same as n. Table 4.1 lists the main functions in generating Harmonic Shape Images along with their computation complexities.

Function	Complexity
ConstructEdgeListonSurface	$O(n_e \log(n_e))$
ConstructBoundaryMapping	$O(n_e) + O(\alpha n \log(\alpha n))$
ConstrucInteriorMapping	$O((1-\alpha)n \log((1-\alpha)n))$
ResampleInputSurface	$O(n_r)$
ComputeShapeInvariant	$O(n_r)$

Table4.1: Functions and their computation complexity

CHAPTER 5

MATCHING HARMONIC SHAPE IMAGES

From the generation process of Harmonic Shape Images, it can be seen that they are no different from $2D$ images except that they are not in *m*-by-*n*-pixel format. Both horizontal and vertical scanning of ***HSIs*** can result in such a format. This implementation issue will be discussed later in this chapter. For now, Harmonic Shape Images are considered to be the same as general $2D$ images. The comparison of two Harmonic Shape Images can be performed using the normalized correlation (5.1.1):

$$R(HSI1, HSI2) = \frac{N\sum P_i Q_i - \sum P_i \sum Q_i}{\sqrt{\left(N\sum P_i{}^2 - \left(\sum P_i\right)^2\right)} \times \sqrt{\left(N\sum Q_i{}^2 - \left(\sum Q_i\right)^2\right)}} \tag{5.1.1}$$

in equation (5.1.1), $HSI1$ and $HSI2$ are two Harmonic Shape Images, p_i and q_i are corresponding pixels in $HS1$ and $HS2$, respectively. N is the number of correspondences.

It has been discussed in Chapter 4 that there may be a planar rotation difference between two Harmonic Shape Images due to different choices of the starting vertex of the boundary mapping. This rotation difference needs to be found by the matching process. This means that the correlation coefficient between two surface patches is a function of the rotation angles θ between their Harmonic Shape Images:

$$R(HS1, HS2) = \max_{\theta}\left(R(\theta)\right) \tag{5.1.2}$$

The most discrete approach work is as follows: Given two surface patches $S1$and $S2$, compute their Harmonic Images $HS1$, $HS2$ respectively. Compute the Harmonic Shape Images $HSI1$ of $HS1$ and fix it. Then rotate $HS2$ starting at 0 degree, incrementally increase the angle until the angles reaches 2π. At each rotation angle, compute the Harmonic Shape Image HSI_2^r from HS_2^r and compute the normalized correlation coefficient $R(\theta)$ between $HSI1$ and HSI_2^r. The maximum $R(\theta)$ is the correlation coefficient between $S1$ and $S2$.

As discussed in Chapter 4, surface continuity is preserved in Harmonic Shape Images. This property makes the correspondence problem trivial, which means that no extra computation is needed in order to establish the correspondence between two surface patches after their Harmonic Shape Images are matched. The reason is as follow: the correlation-based image matching establishes the correspondence between two Harmonic Shape Images naturally. Because each vertex on the Harmonic Shape Images has its correspondence on the original surface patch, the correspondences between the two Harmonic Shape Images are also the correspondences between the vertices on the two surface patches.

5.1 Shape Similarity Measure

The comparison of two Harmonic Shape Images has been discussed in the previous section. Suppose that we have another surface patch $S3$ and we compare it to $S1$, and we get the correlation coefficient $R(HSI1, HSI3)$. Now we want to know which surface patch, $S2$ or $S3$ is more similar to $S1$ in shape. The answer is easy since we

---

will choose the surface patch that has a greater correlation coefficient with $S1$. However, if we have a large number of surface patches and we need to find out which one is most similar to a specified surface patch, using a simple threshold would not be an adequate answer. In this case, we would need a more sophisticated measurement in order to determine how similar the most promising surface patch is to the specified patch. This measurement is called a shape similarity measure. It is defined using the normalized correlation coefficient R between two Harmonic Shape Images.

$$C(S1,S2)=\ln\frac{1+R(HSI1,HSI2)}{1-R(HSI1,HSI2)} \qquad (5.2.1)$$

The above similarity measure is a heuristic loss function that will return a high value for two highly correlated Harmonic Shape Images. The change of variables, a standard statistical technical [34] performed by the hyperbolic arctangent function on the right hand side of equation (5.2.1), transforms the correlation coefficient into a distribution that has better statistics properties, namely, the variance of the distribution is independence of R. In this case, the variance of the transformed correlation coefficient becomes $1/(N-3)$, in which N is the number of correspondence in the two Harmonic Shape Images [34].

In the experiment to explain how to use the shape similarity measure [26], it can be seen that we can choose the surface patch with the greatest shape similarity measure. However, this does not tell us how good this match is or how well it is distinguished from the other matches. Therefore, instead of using a simple threshold, a statistical method [34] is used to automatically detect the best match and determine how good the match is. According to this method, good matches correspond to the outliers of a given histogram.

5.2 Resampling Harmonic Shape Images

As an implementation issue, it has been mentioned in Section 5.1 that a horizontal and vertical scanning of Harmonic Shape Images need to be done in order to request them using the usual m-by-n-pixel format. This step is called resampling Harmonic Shape Images.

A unit grid is created and overlaid on the Harmonic Shape Images. Then the curvature value for each of the points on the unit grid is determined by interpolating the Harmonic Shape Images. For an arbitrary point $u(i, j)$ on the grid, its curvature value $c(i, j)$ is interpolated using:

$$c(i, j) = \alpha c(v_0) + \beta c(v_1) + \gamma c(v_2) \qquad (5.3.1)$$

in which v_0, v_1, v_2 are the vertices of the triangle in the Harmonic Shape Image that $u(i, j)$ falls in. α, β and γ are the barycentric coordinates of $u(i, j)$ in the triangle (v_0, v_1, v_2) and they satisfy the constraint:

$$\alpha + \beta + \gamma = 1, \alpha.\beta.\gamma \geq 0 \qquad (5.3.2)$$

5.2.1 Resampling Resolution

There are few issues that need to be discussed with respect to the resampling process. In order to not lose the shape information of the original surface, the resolution of the resampling grid should not be lower than that of the triangular mesh. Suppose that the size of the resampled Harmonic Shape Image is $N \times N$, N is determined according to the following:

$$N = \left\lfloor \sqrt{n_v} \right\rfloor \qquad (5.3.3)$$

in which n_v is the number of vertices on the triangular mesh and the function $ceil(x)$ means to obtain the nearest integer which is not greater than $\sqrt{n_v}$. When there are two surface patches, with N_m and N_v vertices, respectively to be compared, N is selected to be the larger one between N_m and N_v.

5.2.2 Locating Resampling Points

The second issue is how to efficiently locate the resampling points on the Harmonic Shape Image. As discussed earlier in this section, the curvature value of a resampling point is interpolated using its barycentric coordinates of the triangle in which it falls. So there is the issue of locating the triangle in which the resampling point falls.

The criterion for locating the right triangle is to compute the barycentric coordinates for the resampling vertex in each triangle on the surface patch. If the barycentric coordinates satisfy equation (5.3.2), then the resampling vertex falls into that triangle. The barycentric coordinates can be obtained by solving the following equations. Let $(x_i, y_i), i = 0,1,2$ denote the coordinates of the resampling vertex u. Then the following equations hold:

$$\begin{aligned} \alpha x_0 + \beta x_1 + \gamma x_2 &= x \\ \alpha y_0 + \beta y_1 + \gamma y_2 &= y \\ \alpha + \beta + \gamma &= 1 \end{aligned} \qquad (5.3.4)$$

equation (5.3.4) can be written in matrix form as equation (5.3.5).

$$\begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix} \begin{vmatrix} \alpha \\ \beta \\ \gamma \end{vmatrix} = \begin{vmatrix} x \\ y \\ 1 \end{vmatrix} \tag{5.3.5}$$

The unknown vector $[\alpha\beta\gamma]^T$ can be solved as follows:

$$\begin{vmatrix} \alpha \\ \beta \\ \gamma \end{vmatrix} = \begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ 1 & 1 & 1 \end{vmatrix}^{-1} \begin{vmatrix} x \\ y \\ 1 \end{vmatrix} \tag{5.3.6}$$

According to the way the interior mapping is constructed, there should not be any degenerated triangles. This means that the matrix in (5.3.6) always has full rank.

CHAPTER 6

MATCHING TOOTH BRACKET SURFACES TO TOOTH SURFACES

The concept of Harmonic Shape Images, how to generate Harmonic Maps, Harmonic Shape Images and how to match Harmonic Shape Images, has been discussed in previous chapters. In this Chapter, we will apply the Harmonic Shape Images to the tooth surfaces and tooth bracket surfaces, and try to find a best match of the tooth bracket surface out of a set of tooth bracket to one specified tooth surface.

6.1 The construction of Harmonic Shape Images of the tooth surface and tooth bracket surface

The procedure of the generation of the Harmonic Shape Images of the tooth surfaces and tooth bracket surfaces is as follows:

1. Construct the bi-directional graph of the specific surface.
2. Use Dijkstra or z-coordinate projection methods to compute the surface distance of the points on the surface mesh.
3. Compute the Harmonic Mapping of the surface.
4. Compute the Harmonic Shape Image of the surface using the distribution functions.

Fig 6.1 shows an arbitrary tooth surface patch. Fig 6.2 shows the Harmonic Map of the tooth surface patch in Fig 6.1. Fig 6.3 shows an arbitrary tooth bracket surface. Fig 6.4 shows the Harmonic Map of the tooth bracket surfaces in Fig 6.3.

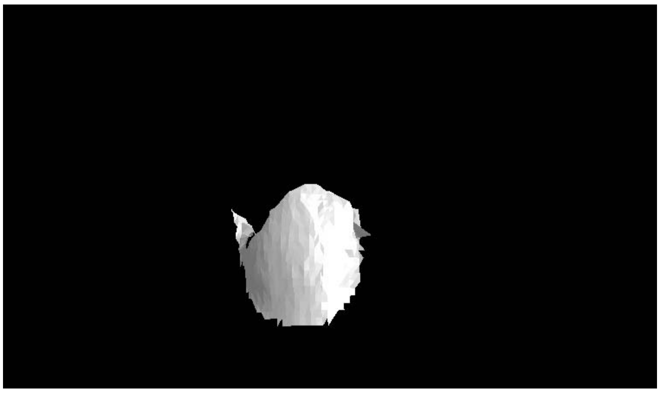


Fig 6.1 An arbitrary tooth surface

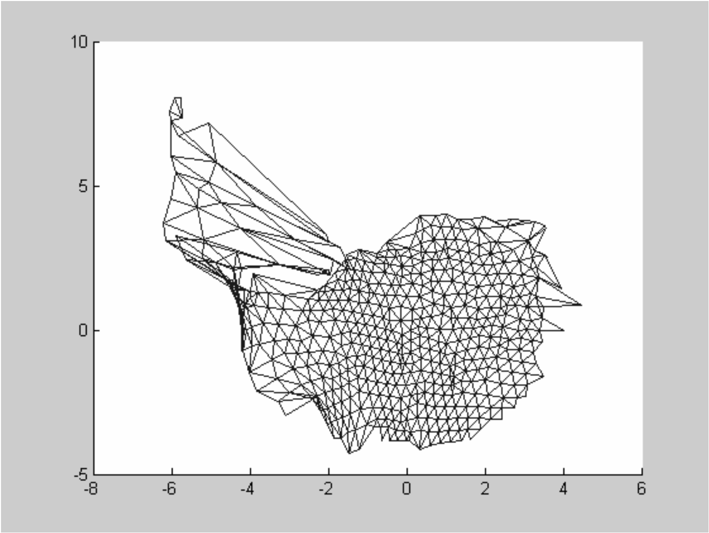


Fig 6.2 Harmonic Map of the tooth surface

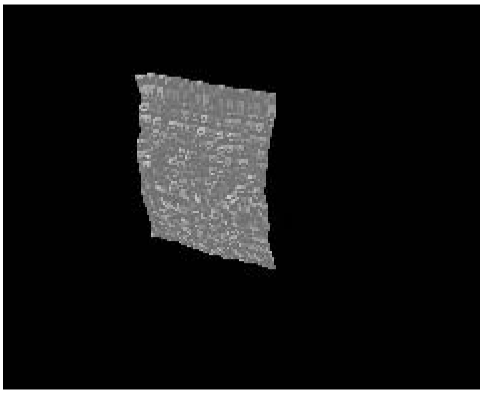


Fig 6.3 An arbitrary tooth bracket surface

.

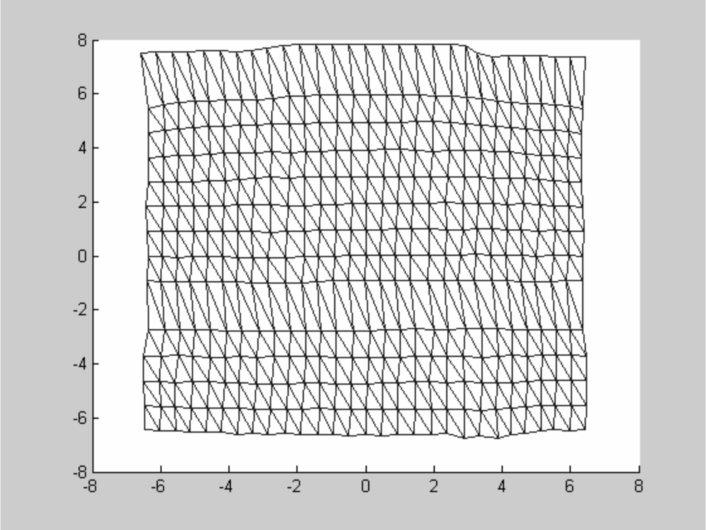


Fig 6.4 Harmonic Map of the tooth bracket surface

6.2 Matching Tooth Surfaces and Tooth Bracket Surfaces

The matching procedure of tooth surfaces and tooth brackets surfaces is outlined as follows: the Harmonic Shape Image of every tooth bracket surface is computed and stored. Then every tooth bracket surface is compared to the tooth surface by comparing their Harmonic Shape Images, and the shape similarity value is computed.

In [26], four properties of the Harmonic Shape Images have been analyzed. The four properties are discriminability, stability, robustness to resolution, and robust to occlusion. In our project, there is no occlusion problem encountered considering the scanning of tooth models and tooth bracket surfaces. One of the important properties of shape representation is its ability to discriminate surfaces of different shapes. This is referred as the discriminability of the representation. Stability is another important property of Harmonic Shape Images. Unlike discriminability which measures the capability of discriminating different shapes, stability measures the capability of identifying similar shapes. It has been discussed that Harmonic Shape Images do not depend on any specific sampling strategy, e.g., uniform sampling. For a given surface, as long as the sampling rate is high enough such that the shape of the surface can be sufficiently represented, its Harmonic Shape Image is also accurate enough for surface matching. It should be noted that the comparison of Harmonic Shape Images does not require that the two surface patches have the same sampling frequency. In practice, it is rare for discrete surfaces to have exactly the same sampling frequency. In our project, the robustness to resolution is especially important because the tooth models and tooth brackets are scanned using different laser scanners. One important property for surface representation is its robustness to occlusion, i.e., correct matching result should still be obtained even when the surfaces being compared are not complete. In our project, the tooth surface patches and tooth bracket surface patches are all complete, so we will not encounter occlusion problems.

In this chapter, twenty tooth bracket surface patches are presented and the Harmonic Shape Images of each tooth bracket surface are compared with the Harmonic Shape

----

Images of one specific tooth surface. The values of the shape similarity measure are computed and listed in Table 6.1.

Table 6.1 shows that the shape similarity measure of tooth bracket surface patch 12 is 2.9833, greater than other tooth bracket surface patches. Because the tooth bracket surface patches are similar in their shape, the values of the shape similarity measure do not differ much from one another. It can be seen that due to the scanning error of the tooth brackets, there are some considerations that the result may have errors.

Tooth bracket number	**1**	**2**	**3**	**4**	**5**
Shape similarity measurement	**2.3112**	**2.358**	**2.2604**	**2.4171**	**2.5026**
Tooth bracket number	**6**	**7**	**8**	**9**	**10**
Shape similarity measurement	**2.5345**	**2.737**	**2.6501**	**2.8249**	**2.7605**
Tooth bracket number	**11**	**12**	**13**	**14**	**15**
Shape similarity measurement	**2.9064**	**2.9833**	**2.8476**	**2.8351**	**2.6633**
Tooth bracket number	**16**	**17**	**18**	**19**	**20**
Shape similarity measurement	**2.6507**	**2.2099**	**2.1651**	**1.8743**	**1.9004**

Table 6.1 values of the shape similarity measurements

CHAPTER 7

CONCLUSION

In conclusion, the work presented in this thesis was an attempt to study the various ways in which the practice of orthodontics could benefit from the advancement of computer vision. It was seen that orthodontists largely use conventional techniques for routine diagnoses and treatment. Dentists seldom are concerned about shapes, sizes and measurements of teeth, and other geometric parameters. This work has produced a surface matching strategy together with a complete 3D visualization of the dental plaster cast that will help the orthodontists in deciding which tooth bracket should be put on to the surface of an individual tooth.

In this study, our main focus was on the problem of tooth brackets. In the practice of orthodontics, fixed appliances like tooth brackets are a common means to achieve appropriate movements to align and re-position teeth. However, due to lack of complete information about the tooth bracket surfaces, the selection of tooth bracket to put onto a tooth surface is an empirical activity of the orthodontists. We proposed a set of tools that can help the orthodontists in extracting the tooth surface interested from a dental plaster cast and use the tooth surface to compare with the tooth bracket surface.

The work presented in this thesis is not complete due to lack of tooth brackets. Future work may include: matching tooth brackets to the teeth of different ethical groups, which will help the orthodontists a lot in their dental practice, designing tooth brackets according to the tooth surface shape of the patient, which means an individual patient may have his unique tooth brackets.

References:

[1]. C. Dorai, A.Jain, COSMOS – a representation scheme for 3D free-form objects, *IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence,* 19(10): pp. 1115-1130, 1997.

[2]. J. Foley, A. VanDam, S. Feiner and J. Hughes, *Computer Graphics: Principles and Practice,* Addison Wesley, New York, 1990.

[3]. J. Barhak and A. Fischer, *Parameterization Reconstruction from 3D Scattered Points Based on Neural Network and PDE Techniques*, IEEE Translations on Visualization and Computer Graphics, VOL 7, No.1, January-March 2001.

[4]. Heung-Yeung Shum, Martial Hebert, Katsushi Ikeuchi, *On Shape Similarity,* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '96), June, 1996, pp. 526 - 531.

[5]. F. Stein and G. Medioni, *Structuring Indexing: efficient 3D object recognition,* IEEE Tans. Pattern Analysis and Machine Intelligence, VOL. 14, No.2, pp, 125-145,1992.

[6]. C. Chua and R. Jarvis, *3D Free-form Surface Registration and Object Recognition*, Int'l Jour. Computer Vision, VOL 17, No.1, pp, 77-99, 1996.

[7]. Andrew E. Johnson and Martial Hebert, *Surface matching for object recognition in complex three-dimensional scenes*, Image and Vision Computing, **16**, pp. 635-651, 1998.

[8]. Andrew E. Johnson and Martial Hebert, *Using Spin-Images for efficient multiple model recognition in cluttered 3-D scenes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(5), pp. 433-449,1999.

[9]. O.D. Faugeras and M. Hebert, *The Representation, Recognition and Locating of 3D Objects,* int'l J. of Robotics Research, VOL 5, No. 3, pp 27-52, Fall 1986.

----

[10]. A. Johnson, *Spin-images: a representation for 3D surface matching,* Ph.D Thesis, CMU-RI-TR-97-47, Robotics Institute, Carnegie Mellon University, 1997.

[11]. Andrew E. Johnson and Martial Hebert, *Recognizing objects by matching oriented points,* Proc. Computer Vision and Pattern Recognition (CVPR '97), pp. 684-689, 1997.

[12]. Andrew E. Johnson and Martial Hebert, *Efficient multiple model recognition in cluttered 3-D scenes,* Proc. Computer Vision and Pattern Recognition (CVPR '98), pp. 671-677, 1998.

[13]. Andrew E. Johnson and Martial Hebert, *Control of polygonal mesh resolution for 3-D computer vision,* Graphical Models and Image Processing, 60, pp 261-285, 1998.

[14]. P. J. Besl and N. D. Mckay, *A method for registration of 3D shapes,* IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence, 14 (2): pp, 239-256, 1992.

[15]. P. J. Besl, *Triangles as a primary representation,* Object Representation in Computer Vision, M. Hebert, J. Ponce, T. Boult and A. Gross, eds.. pp. 191-206, Berlin, Springer-Verlag, 1995.

[16]. Z. Zhang, *Iterative point matching for registration of free-form curves and surfaces,* IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence, 13 (2); pp, 119-152, 1994.

[17]. Y. Chen and G. Medioni, *Surface description of complex objects from range images,* Image Vision Computing, 10 (3): 145-155, 1992.

[18] K. S. Arun, T. S. Huang, and S. D. Blostein, *Least square fitting of two 3D point sets,* IEEE Transaction Pattern on Pattern Analysis and Machine Intelligence, VOL, PAMI-9, No.5, pp, 698-700, 1987.

----

[19]. P. J. Besl, *Geometrical modeling and computer vision,* Proc. IEEE, VOL 76, No. 8, pp. 936-958, 1998.

[20]. P. J. Besl and R. C. Jain, *Three dimensional object recognition,* ACM Comput. Surveys, VOL 17, No.1, 75-145, 1985.

[21]. S. Z. Li, *Inexact matching of 3D surface,* VSSP-TR-3-90, Univ. of Surrey, England, 1990.

[22]. D. W. Murrey, *Model based recognition using 3D shape alone,* Comput. Vision Graphics Image Processing, VOL. 40, pp. 250-266, 1987.

[23]. R. Bergevin, D. Laurendeau, and D. Poussart, *Estimating the 3D rigid transformation between two range views of a complex object,* 11th, IARP, Int'l Conf. Patt. Recog., pp. 478-782, the Hague, 1992.

[24] Y. Xin, *Geometry of Harmonic Maps,* Birkhauser, 1996.

[25] J. Eeles and I. H. Sampson, *Harmonic Mappings of Riemannian manifolds,* Amer. J. Math., 86: 109-160, 1964.

[26] D. Zhang and M. Hebert, *Harmonic Maps and Their Applications in Surface Matching,* IEEE Conference on Computer Vision and Pattern Recognition (CVPR '99), Vol. 2, 1999.

[27] Jackie Neider, Tom Davis and Mason Woo, *OpenGL Programming Guide or the " Red Book",* Addison-Wesley Publishing Company, 1994.

[28] M. ECK, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle, *Multi-resolution Analysis of Arbitrary Meshes*, *Technical Report,* 95-01-02, The Computer Science Department, University of Washington, January, 1995.

[29] Takashi Kanai, Hiromasa Suzuki, *Approximate Shortest Path on Polyhedral Surface and Its Applications,* Computer-Aided Design, Volume 33, Number 11, pp.801-811, September 2001.

----

[30] Takashi Kanai, Hiromasa Suzuki, Fumihiko Kimura, *Three-dimensional Geometric Metamorphosis Based on Harmonic Maps,* The Visual Computer, Vol. 14, Issue 4, pp.166-176, 1998.

[31] H. Delllingette, M. Hebert and K. Ikeuchi, *A spherical representation for the recognition of the curved objects,* Proc. Computer Graphics (SIGGRAPH’96), August 1996.

[32] A. Aho, J. E. Hopcraft and J. D. Ullman, *Data Structure and Algorithms,* Addison-Wesley, Reading, Mass., 1983.

[33] H. Dellingette, M. Hebert and K. Ikeuchi, *Representation and recognition of free-form surfaces,* Technical Report, CMU-CS-92-214, Computer Science Department, Carnegie Mellon University, November 1992.

[34].J. Deovre, *Probabilities and statistics for engineering and sciences,* Brooks/Cole, Belmont, CA, 1987.

[35]. D.Laurendeau, *A computer vision technique for the acquisition and processing of 3D profiles of dental imprints: An application in Orthodontics.* IEEE Trans. on Medical Imaging, vol. 10, no. 3, sep. 1991.

[36]. Jean Cote, D. Laurendeau and D. Poussart, *A multi-operator approach for the segmentation of 3D images of dental imprints*, Proc. Vision Interface, Calgary, June 3-7, 1991, pp. 189-196.

[37]. F. Methot, M. Mokhtari, D. Laurendeau and D. Poussart, *Senior fusion of 2D and 3D data for the processing of images of dental imprints,* Proc. SPIE – The Intl. Soc. for Opt. Engg., Vol. 2055, pp. 546-556, 1993.

[38]. M. Mokhtari and D. Laurendeau, *Feature detection of 3D images on dental imprints,* Proc. IEEE Workshop on Biomed. Image. Anal., pp. 287-296, 1994.

----

[39]. C. Revol and M. Jourlin, *A new minimum variance region growing algorithm for image segmentation*, Pattern Recognition Letters, Vol. 18, iss. 3, pp. 249-258.

[40]. M. Alcaniz, S. Albalat, V. Colomer and C. Monserrat, *Digital dental surface registration with laser scanner for orthodontics set-up planning,* Proc. SPIE – The Intl. Soc. for Opt. Engg., Vol. 3032, pp. 506-512, 1997.

[41]. M. N. Ahmed, S. M. Yamany, E. E. Hemayed, and A. A. Farag, *3D model building of the jaw impression*, Proc. SPIE – The Intl. Soc. for Opt. Engg., vol. 3023, pp. 87-97, 1997.

[42]. B. Kralj, J. Middleton, M. L. Jones, and P. F. Hubsch, *3D numerical modeling of orthodontic brackets,* Computer Methods in Biomechanics and Biomed. Engg., Swansea, UK, Sep. 1994, pp. 371-381.

[43]. D. Paulus, K. H. Kunzelmann, S. Kuppers, H. Neimann, and M. Wolf, *Automatic CNC program generation from range data,* Proc. of Intl. Conf. on Recent Advances in Mechatronics, ICRAM, pp. 230-237, vol. 1, 1995.

ACKNOWLEDGEMENTS

There are many people who helped me in my thesis. First and foremost, I am most indebted to my thesis advisor, Dr. Ashraf Kassim. Dr. Ashraf guided me through the work. His broad and in-depth knowledge of computer vision and his enthusiasm to exploring unknown research fields have great influence on me. I am always grateful to Dr. Ashraf for his knowledge and his guidance.

I am grateful to Dr. Ong Sim Heng. I developed most of the thesis work in the team of Vision and Image Processing Lab. Dr. Ong gave me great help in my research work and guided me through my work. Dr. Ong is smart, funny, easy going; excellent qualifications for any advisor. I also had a great time when discussing in the team, I would like to thank the team members, Toshi Kondo, Xiao Gaoyu, Yangmei.

I would like to acknowledge Dr. Kelvin Foong. Dr. Foong is the supervisor from the Faculty of Dentistry. He gave me a lot of help from the orthodontist’s point of view, gave me guidance on how the research work should be done. And also, he helped me a lot in getting the tooth brackets.

I would like to thank Mr. Neo Ken Soon. He is the technician of the Mechanical Engineering Lab. He helped me in scanning the tooth bracket surfaces. He gave me great support in training me how to use the scanner.

I would also like to thank all my friends. They helped me in one-way or the other of my research.

Last but not least, with all my heart, I would like to thank my parents, and my brother, for their understanding and consistent support.