3D ORTHODONTICS VISUALIZATION

ZHANG HUA

(B.Eng., National University of Singapore)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2005

Acknowledgements

I would like to extend my sincerest gratitude and deepest appreciation to the following people for their help and support throughout the course of study.

A/P Ong Sim Heng, my research supervisor and constant source of support, encouragement and inspiration.

A/P Kelvin Foong, my research co-supervisor, for his support and guidance, especially his expertise in orthodontics.

Mr. Francis Hoon, laboratory officer, for his constant support.

Mr. Hiew Litt Teen, research scholar, for his help in developing the visualization software.

All those who have helped me in one way or another during the study.

Table of Contents

Summary

The rapid advances in computer visualization techniques have been of great benefit to clinical practice. With rapid advances in computer vision technology, there has been great demand in developing computer visualization systems to facilitate the tedious, manual flow of orthodontics treatment planning. However, orthodontic treatment nowadays is still primarily based on the orthodontist's expertise. A suitable computer simulation tool is yet to be developed to fulfill the required capabilities in orthodontics simulation.

Orthodontics refers to the speciality and practice of preventing and correcting irregularities of the craniofacial complex, particularly the growth of teeth. Conventionally, orthodontists conducts treatment planning based on their expertise, with the help of various information sources, such as study models and the orthopantomogram. Orthodontics treatment is a long-term process which requires frequent visits with orthodontists, during which the treatment plan is gradually developed.

In this study, we developed an orthodontics visualization system. Our objective is to replace the conventional assessment of malocclusions with 3D computer simulations. Various medical imaging techniques are employed in this study, including plaster study model and orthopantomogram. Automatic crown segmentation followed by individual crown extraction allows orthodontists to freely re-arrange individual crowns to achieve satisfactory occlusion status. Moreover, the system is able to estimate 3D root shape by deforming generic tooth model with reference to the actual orthopantomogram, which is not available by conventional orthodontics practices. By merging the estimated root shape with the crown information from the study model, the system is able to present full set of 3D tooth (with roots) without using CT scans, which is a unique feature compared to earlier attempts in orthodontics visualization.

Occlusion analysis enables orthodontists to visualize the occlusal status of a pair of study models. With the ability in tracing affine transformations, the system is also able to simulate the occlusal status of a re-arranged dentition.

Other data sources, such as 3D facial model and the lateral cephalogram, are also bought in to provide orthodontists with more references in treatment planning. With the ability of incorporating different medical imaging techniques, the system allows orthodontists to assess the case in different perspectives.

This study is part of an entire research project whose objective is to develop a visualization system that is capable of presenting a virtual human head to facilitate orthodontics and craniofacial treatment planning. Preliminary results have shown that this visualization system developed in this study is able to provide adequate degree of accuracy in measurements. The system has the potential in replacing the manual process in orthodontics treatment planning with powerful computer simulations and thus changing the flow of orthodontics treatment.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Orthodontics

Dentistry refers to the practice of preventing and treating diseases of the teeth, gums, and tissues of the mouth. Unlike other human tissue, such as human skin, that continuously grows and self-rejuvenates, dental structures generally cannot repair themselves and require regular care to retain their health and vitality. If not treated, dental health problems can lead to complications in other parts of the body. Thorough and timely dental care is not only important for maintaining healthy teeth and gums, it is essential to overall human health.

Orthodontics is the branch of dentistry that is concerned with the study of the growth of the craniofacial complex (Chateau, 1975). The aim of orthodontic treatment is to improve occlusal function by the correction of tooth position irregularities, and as a consequence the creation of better dental health and improved aesthetics.

The irregular position of teeth has been a problem for some individuals since the beginning of time. Attempts to correct this disorder go back to at least 1000 BC. However, since very few people in those days had a complete set of teeth, little attention was given to how teeth bite together until the mid-eighteenth-century, when the first texts which systematically described orthodontics appeared, in which Dr.

Norman Kingsley first derived the use of extraoral force to correct protruding teeth. Advancements were made starting from 1890s, when artificial teeth were used and the theory of how teeth should bite together was developed. Dr. Edward Angle, "The Father of Modern Orthodontics", was one of the first to emphasize occlusion[1] in natural dentition. His interest in creating proper occlusion in natural teeth created the specialty of orthodontics. With the fundamental concepts established by Angle, orthodontics began to evolve into the treatment of malocclusion[2], instead of just straightening of teeth.

Cephalometric radiographs[3] (cephalograms) were first introduced to orthodontics in 1940s. With this technology, which allows orthodontists to see how the bones of the face contributed to malocclusion, it was discovered that the growth of bones in growing individuals could be altered to prevent malocclusion. As for individuals whose bones have stopped growing, malocclusions became correctable in 1970s, when techniques of oral surgery appeared.

1.2 Conventional Orthodontics Treatment

In orthodontics, occlusion is defined as the contact of the dentures of both arches resulting from the action of the muscles. The detection and correction of

[1] the alignment and spacing of upper and lower teeth when biting down
[2] poor positioning of teeth
[3] an X-ray of the head that shows whether teeth are aligned properly

malocclusions and other dental abnormalities is a significant area of work in orthodontic diagnosis and treatment.

Conventionally, orthodontists rely on two important sources to assess orthodontics information in diagnosis and treatment planning, namely the study model (Figure 1.1) and orthopantomogram[4] (Figure 1.2).

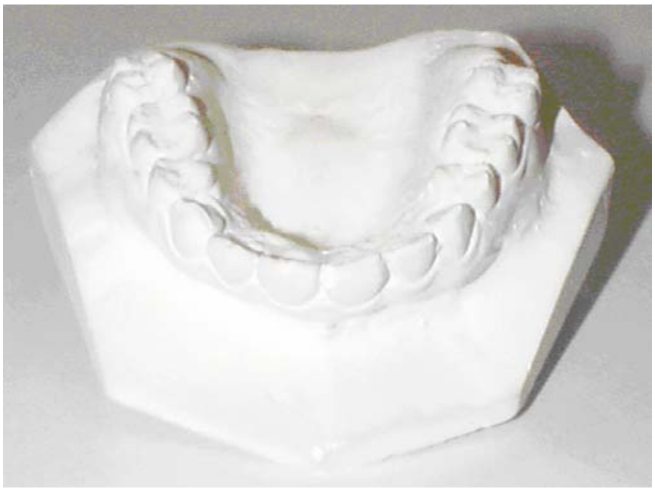


Figure 1.1: Study model.

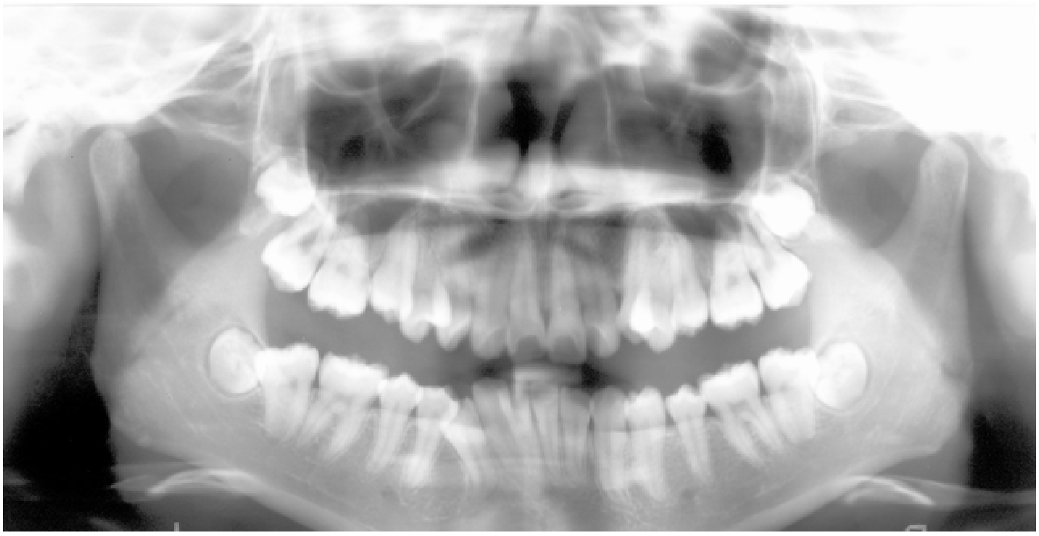


Figure 1.2: Example of orthopantomogram.

[4] a panoramic display of the teeth produced by the technique based on tomography

Modern orthodontics uses a combination of extraoral forces to align teeth as well as growth modification, surgery and extractions to accomplish three goals:

- Create the best occlusal relationship
- Create acceptable facial esthetics
- Create a stable occlusal result

In order to correct malocclusions, orthodontists apply forces to specific teeth. When an orthodontic force is applied to a tooth over a long period of time, it moves owing to resorption of the alveolar bone on the pressurized side and apposition of new bone on the opposite side. Alveolar bone is the tissue situated inside the gum and where teeth are fixed by their roots. Forces are applied by using special appliances as shown in the Figure 1.3.

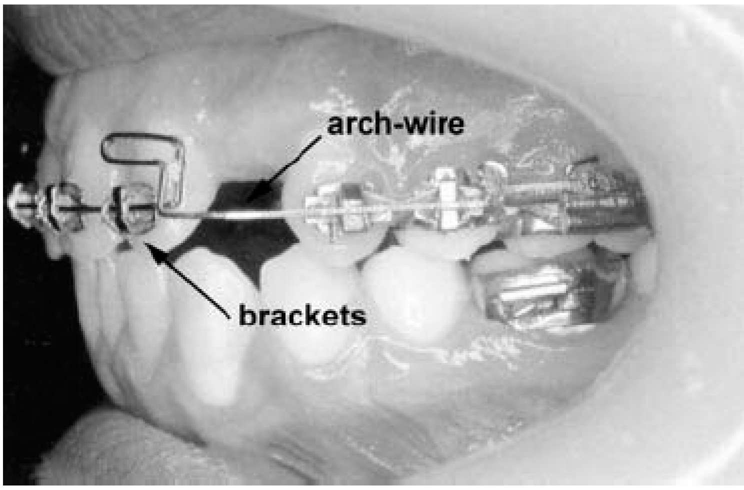


Figure 1.3: Appliances used for teeth movements during orthodontics treatments.

The conventional flow of orthodontics treatment takes a patient a minimum of 4 phases to the orthodontist to complete the treatment, namely:

- Initial orthodontics examination - The initial examination with the orthodontist is a chance to get acquainted and to address any orthodontic concerns or questions that a patient may have. It also gives the orthodontist the chance to make a preliminary evaluation, discuss whether treatment is necessary, when is the best time to initiate treatment, what treatment options are available, time frames and cost.

- Records appointment - This initial orthodontic exam is then followed by a records appointment. Diagnostic information such as photos, models, and specific X-rays are taken on each patient. A treatment plan including available options is completed after a thorough review by the orthodontist.
- Treatment consultation - The consultation appointment (3rd visit) is where the treatment plan is presented and the financial options are reviewed. Treatment may be initiated following consent for treatment.
- Orthodontic treatment - Orthodontic treatments are regular visits at intervals between 3 to 6 weeks until completion of the treatment. The orthodontist uses these visits to evaluate the progress of treatment, and alter the treatment plan if necessary.

Before the introduction of the lateral cephalogram, dental-facial orthopedic treatment planning was based primarily on observations of facial form and dental relationships. With the help of the cephalogram, orthodontists became more aware of the underlying jaw disproportion in the etiology of malocclusion. Also, analyses and treatment methods designed to address both the dental and facial skeletal components of orthodontic problems became more universal. Nowadays, orthodontists and craniofacial surgeons mainly rely on the lateral cephalogram (Figure 1.4) and study model to acquire necessary clinical information in diagnosis and treatment planning. However, it is the orthodontist’s expertise that plays the major role in the treatment process.

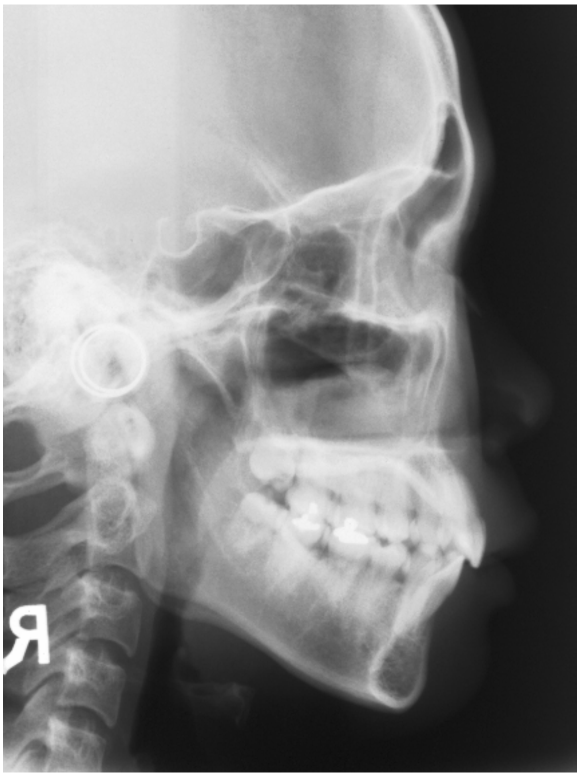


Figure 1.4: Example of lateral cephalogram.

1.3 Computer-Assisted Orthodontics Treatment

Research on 3D medical imaging techniques has been reported frequently since the 1970s [1]. Many researchers have tried to apply engineering techniques, especially image processing and computer vision techniques, to offer help to orthodontists who rely mainly on their expertise in the orthodontics treatment process.

Over the years, computer-aided systems have been introduced to orthodontic diagnosis and treatment planning. The computer-assisted system reported in [2] is capable of handling the classification of skeletal and dental malocclusion from a limited number of measurements. Requiring no human intervention, the measurements from the system can be fed into an expert system which classifies the severity of the malocclusions in three broad categories, i.e.,

- no orthodontics treatment required.

- elective orthodontics treatment required.
- orthodontics treatment required.

An expert system was developed by Yamamoto *et al.* [3] to monitor and measure 3D movement of teeth during orthodontics treatments. In this system, 3D data is obtained by taking study models of the dentition through different stages of the treatment. Tooth movement is estimated by computer-aided registration between teeth profiles extracted from study models. This technique provides a useful tool for orthodontists to optimize a treatment procedure as compared to conventional treatment, where the orthodontists must re-adjust the force and bends of a brace in response to tooth movement.

A computer-aided orthodontic treatment process (the SureSmile process [4]) was recently introduced. In this process, 3D dentition information is obtained directly from patients' mouths using a hand-held scanner. Treatment planning is done using software, and a customized arch is built using robots. It is claimed that minimum teeth movement is achieved by using this process. As a representative of recent development in orthodontics visualization, SureSmile will be described in more detail in Chapter Four.

We have noticed that most of the reported research, some of which are already commercialized, lacks the ability of presenting tooth root information in a virtual 3D environment. As root information is vital in some clinical analysis, such as biting force analysis, this limitation greatly restricts the practicability of such systems. Few systems are indeed able to provide 3D root information. However these systems rely on computerized tomography (CT) images, which is radiologically and also economically expensive.

Advances in computer technology have also brought a positive impact to conventional dental-facial orthopedic treatment. An example is the use of computerized imaging technology in the virtual craniofacial patient project carried out by the Craniofacial Virtual Reality Laboratory and the Integrated Media Systems Center at the University of Southern California [5]. The work here was targeted at building a virtual craniofacial patient from CT data, digital teeth models, and human jaw motion tracking, which would enable researchers to develop methods for surgery simulation and treatment planning, among others.

Much effort has been devoted to integrate data acquired through different imaging techniques, such as the cephalogram, study model, etc. The first step in such integration is data registration. Hajeer *et al.* [6] reviewed a number of 3D techniques that have been utilized to register and analyze the face in three dimensions, and discussed each system's merits and disadvantages. They developed a relatively new 3D imaging system called C3D to capture the 3D geometry of the face. Landmark identification on 3D facial models is facilitated by a software-based facial analysis tool. This system is very useful in studying facial soft tissue changes following orthognathic[5] surgery and other types of facial surgery, as well as assessing facial soft tissue growth and development of the craniofacial complex.

An extension to conventional cephalogram, a Video-Cephalometric system was reported by Sarver [7]. In such a system, a patient' profile images are gathered, integrated and calibrated with the cephalogram. Treatment options are then simulated

[5] normal relationships of the jaws

on the system, where the patient’s facial appearance changes are calculated according to different treatment options and presented. This allows the patient and the surgeon to visualize the facial effect of orthodontic treatments.

As for progress in computer-aided simulation and visualization, Noguchi and Goto [8] have developed a simulation model of an orthognathic patient from the integration of the morphological data of the teeth and face, frontal and lateral cephalograms, and jaw templates by using commercial software (Surfacer, Imageware Inc., Ann Arbor, MI, USA). The objective is to simulate a 3D model without using CT data. However, the system lacks the ability to generate tooth root information, which limits its application areas.

The above mentioned systems have shown their values in respective clinical areas. However, they do have their own limitations. The most prominent one is again about the radiologically invasive CT scan. These systems either lack the capability of presenting 3D information or require CT scans in order to achieve that. Since CT scan is not generally indicated for majority of patients, either due to radiological or economical reasons, these systems are far from practical.

1.4 Objective

The use of various imaging techniques, such as X-rays and study models, are instrumental to the treatment planning process in orthodontics. They give information on the patients’ dentition and as well as their skeletal structure. However, these investigations are 2D and static. They lack the ability to provide 3D information of the teeth and bone and most importantly they lack the ability for users (orthodontists) to manipulate and predict.

The ultimate objective of this research, taken up by our project team, is to develop a visualization system, which is capable of presenting a virtual human head to help doctors in planning of relevant clinical treatments, such as orthodontic treatments or craniofacial surgeries. By taking in data generated from various medical imaging techniques, such as plaster imprint, computerized tomography (CT), magnetic resonance imaging (MRI), panoramic and cephalometric X-ray, the visualization system helps doctors in predicting the outcome of possible treatment plans and hence to decide which one should be taken in practice.

This thesis describes the research work that we have done, including mesh deformation, individual crown extraction, root deformation and occlusion analysis. Each of these topics has its own specific objectives, which will be described in detail in the overview section of the respective chapter.

1.5 Organization of Thesis

This thesis consists of five chapters. Chapter 1 provides general introduction to orthodontics visualization. Chapter 2 introduces related medical imaging techniques in detail. Chapter 3 covers research topics related to 3D facial models, cephalograms. Chapter 4 presents a 3D orthodontics visualization system in detail, which is the major contribution of this research. Chapter 5 focuses on techniques in occlusion analysis. Chapter 6 concludes this thesis and proposes possible extensions of this research.

Chapter 4, where the major research contributions are presented in detail, has its own introductory section consisting of the literature review for the particular research area. In Chapters 3, 4 and 5, in-depth analysis of algorithms and techniques employed are presented, followed by experimental results and critical discussion.

Chapter 2

Data Acquisition

2.1 Overview

Data acquisition refers to the process of obtaining useful data from a physical object. In the case of computer-aided systems, data acquisition is concerned with the digitization of a physical object, which can be the targeted object itself, or an intermediate object, such as 3D imprints and photographic films of the targeted object.

In this research, data acquisition refers to optical data acquisition. 2D data sources, i.e., various kinds of X-ray films, are digitized using ordinary scanners. As for 3D data sources, there are two categories of imaging techniques:

- Passive techniques: without using a special light source, passive techniques rely on the targeted object's reflection under normal lighting condition. Multiple cameras are usually necessary. Typical examples of passive techniques include photometric stereo, shape-from-shading and stereo vision.
- Active techniques: by using a special light source, usually a laser beam for better control, active imaging techniques measure parameters such as time-of-traversal and distortion of projected patterns, hence reconstruct an accurate shape of the targeted object.

In terms of accuracy, researches have shown that both 2D and 3D digitization techniques are capable of delivering the degree of accuracy that is needed for clinical applications. Hajeer has summarized the application of 3D imaging in orthodontics [6]. According to Kusnoto [9], the surface laser scanner has great research potential because of its accuracy and ease of use. The reliability comparison between digital models and plaster models [10] confirms that the magnitude difference is not clinically relevant.

In the following sections, we shall discuss in detail various imaging and digitization techniques that are employed in this research.

2.2 3D Facial Model

Surface mesh data of the patient’s head is of importance in orthodontics treatment planning. With the patient’s actual face as reference and appropriate computer-aided simulation, orthodontists are able to predict the esthetical changes to the patient after possible treatment plans.

A passive 3D imaging system, the Geometrix FaceVision 600, is used to provide 3D facial models in this research. The system consists of six high-speed IEEE-1394 cameras mounted in a rigid frame containing a complete optimized lighting system. The cameras are arranged in three pairs, each of which captures a 3D image from one view of the subject's face (front, right, and left). The image capturing process is completed in less than a second, after which the system’s built-in software automatically reconstructs a seamless textured 3D surface based on the three sets of captured images. The result is a finished ear-to-ear 3D facial model (Figure 2.1). Facial biometrics such as pupil positions and nose shape are also determined. The 600

Series' automated 3D processing software eliminates extensive manual clean-up needed in other 3D facial imaging system.

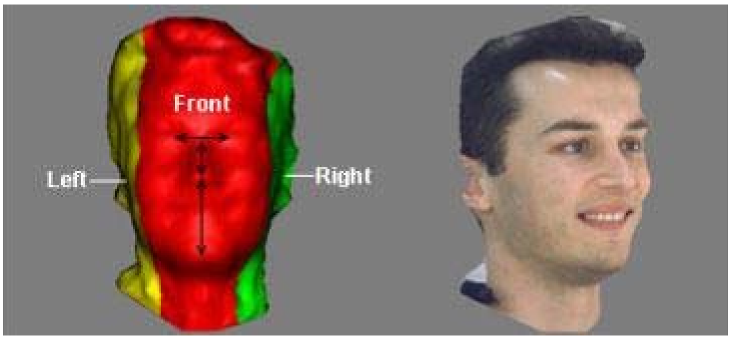


Figure 2.1: Reconstruction of a 3D facial model.

2.3 Cephalogram

The lateral cephalogram (Figure 2.2) is a tomography technique that gives the side-view of a human face, obtained so that accurate measurement of facial landmarks can be performed. A proper cephalogram should provide the diagnostician excellent bony detail in addition to an adequate soft tissue profile. Clear visualization of the Lambodial suture (the suture extending up from the floor of the skull at approximately a 45 degree angle posteriorly towards the occipital bone) is a must and a benchmark for all cephalograms.

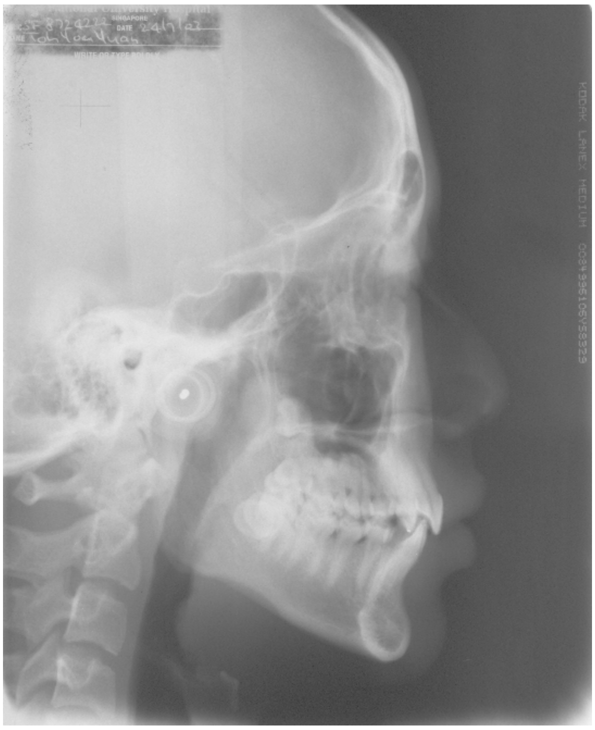


Figure 2.2: Example of lateral cephalogram.

The cephalogram is particularly useful for oral surgeons, orthodontists and prosthodontists who need it prior to planning surgery and tooth manipulation.

The digitization of the cephalograms is done with an Agfa Arcus II scanner at a resolution of 300dpi.

2.4 Study Model

Dental study models, made of plaster, are the 3D imprints taken from patients' dentition (Figure 2.3). Study models in orthodontics are taken to record the initial (baseline) malocclusion of the patient. They may be used to allow a more accurate assessment of the malocclusion and to facilitate measurement.

In order to achieve a high degree of accuracy, we digitized the study models using an active digitization system - Cyberware's 3030RGB/HIREZ/MM (Figure 1.1).

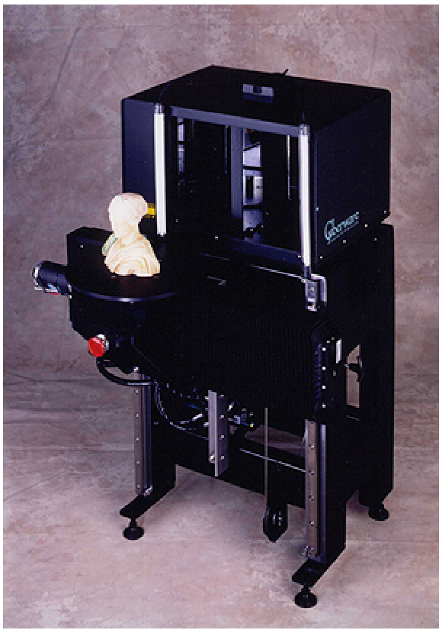


Figure 2.3: Cyberware's 3030RGB/HIREZ/MM laser scanner.

The Model 3030RGB/HIREZ scan head is an implementation of active 3D digitizing technology. The sub-system incorporates a self-contained optical range-finding system, whose dynamic range accommodates varying lighting conditions and surface properties. Entirely software controlled, the scan head requires no user adjustments in normal use.

In operation, the 3030RGB/HIREZ shines a low-intensity laser on an object to create a lighted profile. A video sensor captures this profile from two viewpoints. The system can digitize thousands of these profiles in a few seconds to capture the shape of the entire object. By combining the scanner with an automatic turntable mechanism the system is able to perform a reliable and highly accurate digitization of study models. In terms of accuracy, the system configured for this study is able to give a resolution of 0.15 to 0.10 mm horizontally, 0.30 mm vertically and 0.05 to 0.20 mm along the depth direction.

To obtain a full view of a study model, the scanning process requires a model to be positioned on the turntable at four different poses, each of which contributes 12 scanning shot with 30 degrees of angle offset in between. The typical scanning time of one study model, including subsequent merging and registration steps on the attached graphics workstation, is about 60 minutes. The resulting 3D model is a complete, hole-free mesh of the study model. The 3D mesh data is converted to vertices and face indices and saved to VRML format.

2.5 Orthopantomogram (OPG)

The orthopantomogram (Figure 2.5) is a panoramic display of the teeth produced by the technique based on tomography, i.e., only objects in the focal trough are in focus. The OPG machine is specifically designed to produce panoramic tomographic X-rays of the teeth, jaws and temporomandibular joints. A panoramic X-ray unit rotates in a path about a vertical axis and creates a vertically oriented 3D zone of focus, or focal trough. The X-rays are converted to light images of the patient’s dental arch structure by a stationary intensifying screen. A fiber optic minifying lens reduces the size of the image from the screen while proportionately increasing the light intensity of the image, thus making it possible to provide useable film images at reduced X-ray exposures due to non-linear film exposure versus optical density characteristics.

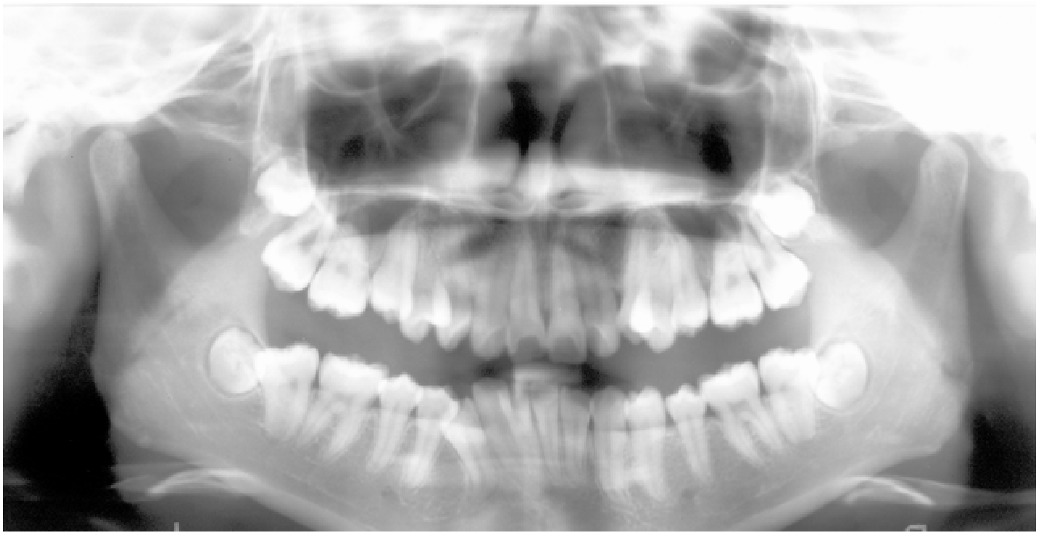


Figure 2.4: Example of orthopantomogram.

The images provide an overview of the state of the dentition as well as information regarding the bones of the upper and lower jaw (the mandible and maxilla); the sinuses in the upper jaw and the joints between the jaw and the skull (the temporomandibular joints).

The OPG is usually indicated for a pre-treatment assessment of the patient’s dentition. It provides valuable information on the total number of teeth, their stage of eruption, and the exclusion of any pathological features. Positions of buried or erupted teeth are hard to assess with a single OPG. The measurement is usually done with the help of an X-ray taken in another view. The paralleling technique is then used to determine the positions of the root and crown of the teeth.

An OPG is usually indicated prior to completion of treatment to check the root positions of the teeth. The aim is to achieve general root parallelism in all the teeth to improve the stability of the treatment outcome.

The OPG system employed in this research is Panoramic Corporation’s PC-1000 (Figure 2.6).

Figure 2.5: PC-1000 OPG system from Panoramic Corporation.

Digitization of orthopantomogram is done with an Agfa Arcus II scanner with a resolution of 300dpi.

Chapter 3

3D Facial Model and Cephalogram

3.1 Introduction

The 3D facial model and lateral cephalogram are both capable of acquiring information of a human face. Compared to computerized tomography (CT), 3D facial model is radiologically harmless and cheaper. However, it only gives surface information but not the more valuable information hidden under the human skin. On the other hand, the lateral cephalogram is also able to acquire information of the entire craniofacial complex. However, it delivers only 2D information, which makes it difficult to adopt it into a 3D visualization system.

The ultimate objective of this research is to develop a visualization system which is capable of presenting a virtual human head by taking in data through various imaging techniques including facial models and study models. However, there is no direct relationship between facial models and study models. In this chapter, we present a procedure to make use of lateral cephalograms as an intermediate reference to perform the registration between facial models and study models.

This part of the research work has been carried out by various members of our project team. The automatic detection method of a significant clinical feature (the facial profile) has been developed to enable the detection of it on both the facial model and the cephalogram. Based on the detection result, a registration method has also been developed. In this chapter, we shall discuss three related topics concerned about these two imaging techniques in this research:

- Facial model deformation.
- Registration of facial model and cephalogram.
- Pre-treatment and post-treatment comparison.

3.2 Facial Model Deformation

3.2.1 Overview

Deformation of 3D models refers to changing the shape of the models. As a high-level operation, deformation usually focuses on the entire geometric structure that represents the target object, therefore making it ideal at the application level as opposed to lower-level operations, such as vertex operations.

The incentive to develop a free-form deformation algorithm in this research is to obtain a general tool for refining some of the imperfect 3D models prior to analysis. Restricted by the imaging techniques, loss of some sharp features of human faces is visible from facial models. These features include nose tips, nose bases and lips.

Figure 3.1 shows an example of an imperfect facial model where the facial profile does not strictly conform to the one on the lateral cephalogram, especially at the nose tip region. The facial profile detected from the cephalogram has a higher nose tip compared to the one detected from the facial model.

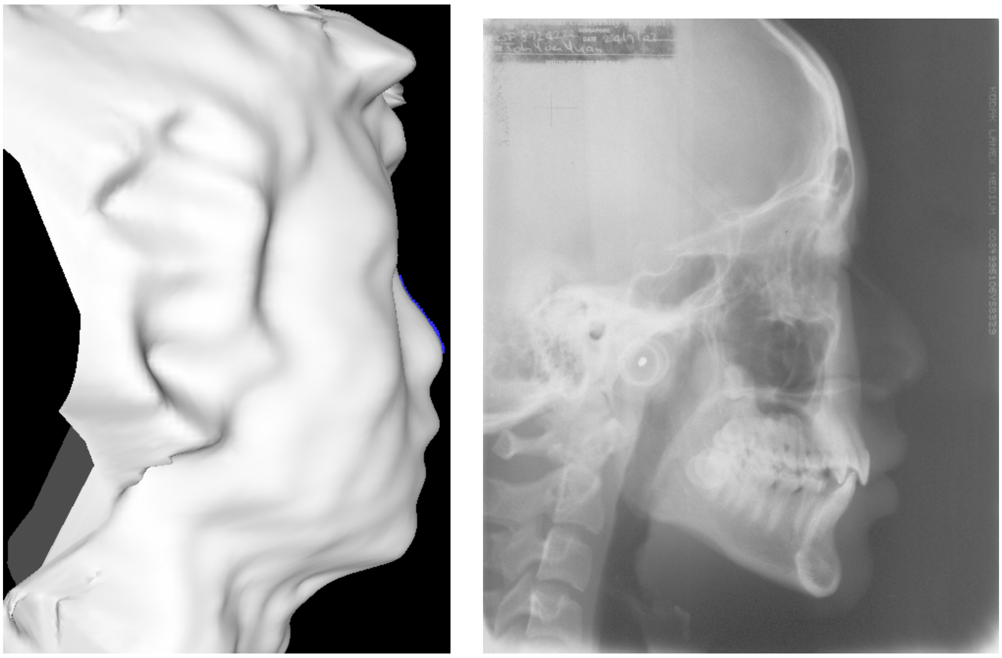


Figure 3.1: Difference in facial profile obtained from facial model and from lateral cephalogram.

The current usage of this algorithm is to refine the 3D facial models given by the FaceVision 600 system, which have visible defects at some part of the images (particularly the nose tip of a human face). It is expected that the developed algorithm will be very useful in more advanced aspects of the research project, e.g., in simulating deformation of soft tissues induced by treatment to nearby hard tissues.

3D shape deformation has been a popular research area since 1990s. The main objective is to obtain fast, efficient and intuitive free-form deformation. Borrel and Rappoport [11] proposed a method called simple constrained deformation (Scodef), which provides user-controlled deformation by picking control points. Yoshizawa [12] extended the idea by proposing the idea of virtual control points. Kobbelt and Bareuther [13] took a different approach by looking at the mesh connectivity. There are two clearly distinguishable ways of realizing controllable free-form deformation:

- Deformation by spatial distance to control point(s).
- Deformation by graph distance to control point(s).

In order to explain the above approaches, we shall look at 3D geometry in some detail. Generally speaking, there are two ways of representing 3D objects, namely volumetric model and mesh model. As suggested by their names, the volumetric model describes 3D objects as solid ones, whereas the mesh model only describes 3D objects in terms of their surfaces (with an empty interior). In this research, we are only concerned with the mesh model stored in the VRML (Virtual Reality Markup Language) format.

In the VRML format, a 3D mesh is described in terms of a vertex list, a face list and optionally a texture coordinates list. The vertex list stores the 3D coordinates of all the vertices contained in a 3D object. The face list describes how these vertices are connected to form polygons that make up the surface of the 3D object. Although general polygons are supported by the VRML format, triangles are used in most cases. It should be noted that polygons in the face list are represented in terms of lists of vertex indices, which directly refer to the vertex list. In other words, the face list itself contains only vertex connectivity information but no spatial information at all. By totally separating the spatial and connectivity information, the VRML format is ideal for most mesh operations, which often manipulate either spatial or connectivity information but not both.

Figure 3.2 illustrates the VRML format by referring to a simple object (a box without lid). The vertex list simply lists all the vertices present in the model, each of which is represented by three numbers as their x, y, z coordinates (Figure 3.3).

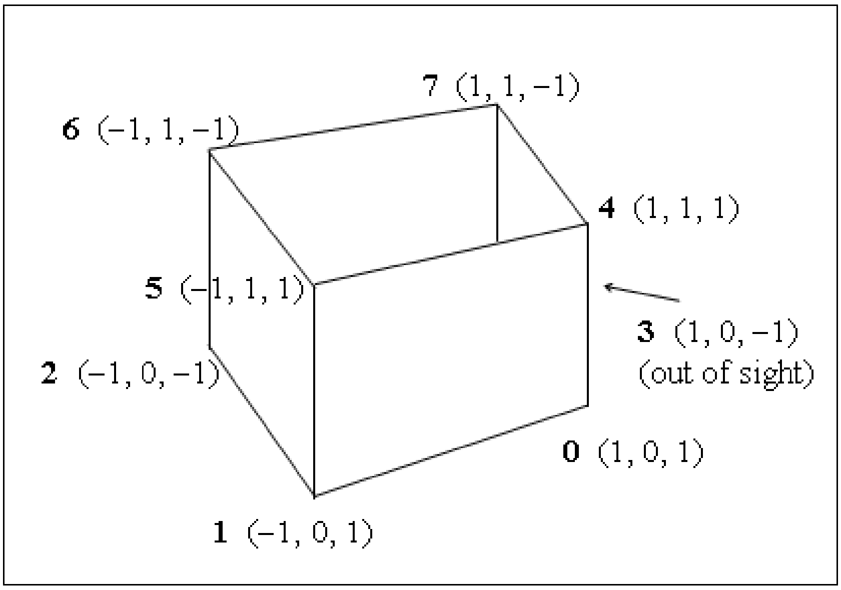
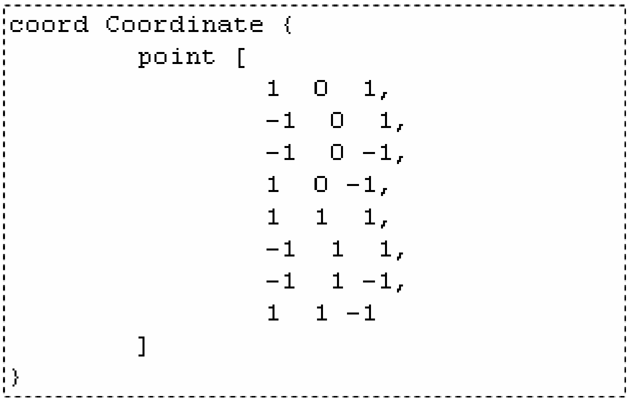


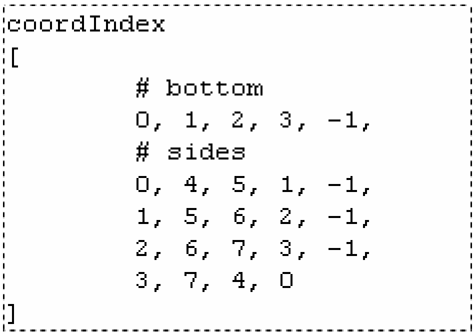
Figure 3.2: A box without lid in the VRML format.



```
coord Coordinate {
        point [
                1  0  1,
                -1  0  1,
                -1  0 -1,
                1  0 -1,
                1  1  1,
                -1  1  1,
                -1  1 -1,
                1  1 -1
        ]
}
```

Figure 3.3: A vertex list in a VRML file.

The face list (Figure 3.4) contains five quadrangles, each of which is represented by four vertices. As explained previously, these vertices are presented in terms of indices that point to the vertex list. The number “-1” is attached to the end of each quadrangle as a delimiter. Character “#” is used as the commenting character in the VRML format. Any line starting with “#” would be ignored during language parsing.



```
coordIndex
[
        # bottom
        0, 1, 2, 3, -1,
        # sides
        0, 4, 5, 1, -1,
        1, 5, 6, 2, -1,
        2, 6, 7, 3, -1,
        3, 7, 4, 0
]
```

Figure 3.4: A face list in a VRML format.

With reference to mesh deformation, both approaches to free-form deformation work by applying a spatial translation to some (or in some cases, all) vertices in the vertex list, i.e., changing the 3D coordinates of these vertices. Both approaches are similar as they require single or multiple (user-defined) control point(s). The difference between the two approaches lies in the way they determine the amount of translation applied to each vertex in the vertex list.

As the first approach (deformation by spatial distance to control points) suggests, the amount of translation of a particular vertex should be a function of the spatial distance between this vertex and the control point, or a function of the distances between this vertex and all control points in a multiple control points situation.

Generally speaking, the translation vector used in this approach gives a stronger response to vertices closer to the control point(s). Therefore it is able to produce a deformation similar to the one induced by gravity. The closer a part of the mesh is from the control point(s), the higher amount of deformation it gets. Figure 3.5 shows a deformation result of a plane by using the above approach.

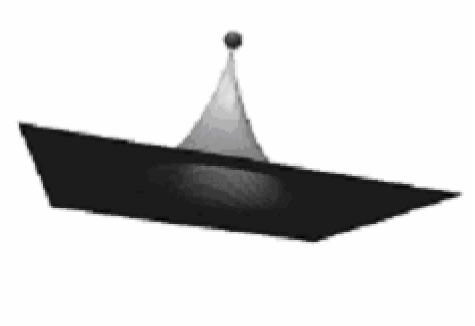


Figure 3.5: Deformation by spatial distance to control points.

The second approach uses a rather different idea. In order to determine the amount of translation of a particular vertex, the algorithm considers the graph distance (instead of the spatial distance) between this vertex and the control point. As the translation vector function is usually a Gaussian-like bump function, deformation by graph distance tends to give an effect similar to lifting a fishing net by one of its knots; those knots directly connected to the lifted knot are most affected, followed by knots that are not directly connected, but one knot away from the lifted knot, and so on. Figure 3.6 gives an example of deformation by graph distance.

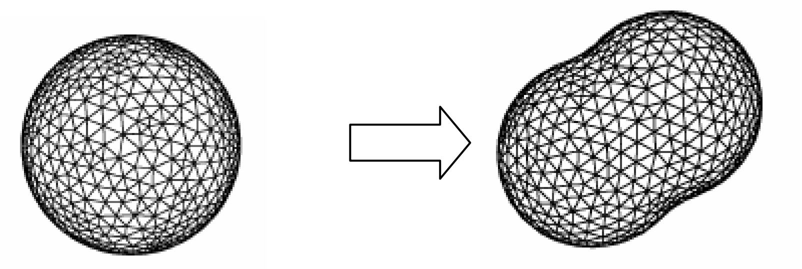


Figure 3.6: Deformation by graph distance.

3.2.2 Algorithm

In order to provide general usage, we decided to develop algorithms based on both approaches, namely:

- Deformation by spatial distance to control point.
- Deformation by graph distance to control point.

We start by taking the basic idea of each approach, followed by looking at the variable parameters and deciding whether each parameter should be fixed or made customizable by the end user. The objective is to develop an easy-to-use yet powerful deformation tool specifically for applications in orthodontics visualization. In the following sections, we shall look at the algorithms in detail.

3.2.2.1 Deformation by Spatial Distance to Control Point

The basic idea of deformation by spatial distance to control point is presented here. Given a mesh and a control point C, let us translate a vertex V in the mesh to a new position V' defined by

$$V' = V + D(C,V)$$

with the translation vector D given by

$$D(C,V) = \frac{\gamma}{\sigma}(V - C)W(C,V)$$

$$\sigma = W(C,V_{\min}),$$

where W is generally a bump-shape function. Specifically, W can take the following Gaussian form,

$$W(C,V) = \exp(-\frac{|C - V|^{\alpha}}{2\varepsilon^{2}})$$

where $V_{\min}$ is the original vertex in the mesh that is closest to C. The shape of the deformation is determined by the parameters α and ε. Parameter γ plays a major role in controlling the direction of deformation (Figure 3.7):

- $\gamma > 0$: the mesh is repelled by the control point.
- $\gamma < 0$: the mesh is attracted by the control point.

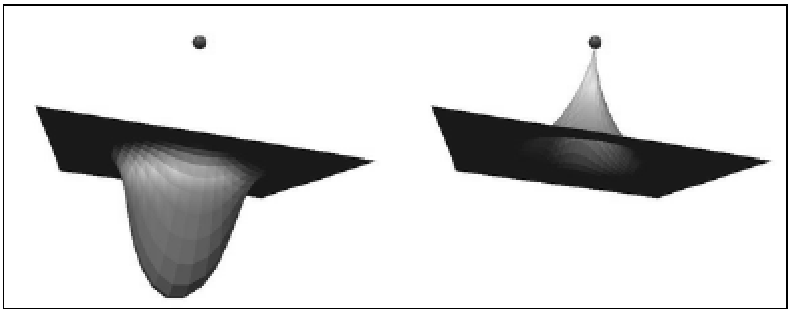


Figure 3.7: Role of γ.

The above idea works well in simulating the deformation of meshes that are uniformly tensioned. However, if we were to apply exactly the same algorithm in this research, the major drawback would be its inability to handle the deformation of human soft tissue, which often sustains different amounts of tension force in different directions. For example, if a small piece of wrinkled human skin is deformed by underlying bone movement, it tends to extend more in an anisotropic manner where the skin has different amount of extension in different directions.

We decide to modify the basic idea to suit our application. Different priorities (weights) are introduced to the three coordinates (x, y, z) in calculating the spatial distance between the vertex of interest and the control point, such that it is possible to perform a direction-sensitive deformation (e.g., the wrinkled skin example previously mentioned). Given a mesh and a control point $C(x_{cen}, y_{cen}, z_{cen})$, the translation of a vertex $V(x_v, y_v, z_v)$ to its new position V' is defined by

$$V' = V + D(C,V)$$

with the translation vector D given by

$$D(C,V) = \lambda\boldsymbol{\delta}\frac{1}{\sqrt{(x_v - x_{cen})^a + (y_v - y_{cen})^b + (z_v - z_{cen})^c}}$$

where $\lambda, \boldsymbol{\delta}, a, b, c$ are parameters.

By taking this approach, the end user can effectively give different weights to the x, y, z coordinates by setting different values for parameters a, b, c to perform a anisotropy deformation. λ here is a scalar controlling the amount of deformation, and vector $\boldsymbol{\delta}$ designates the direction of deformation.

We also set up criteria to define the region-of-influence of the deformation.

$$t_{x,l} < x_v - x_{cen} < t_{x,u}$$

$$t_{y,l} < y_v - y_{cen} < t_{y,u}$$

$$t_{z,l} < z_v - z_{cen} < t_{z,u}$$

$$\left| D(C,V) \right| < t_d$$

where $t_{lx}, t_{ly}, t_{lz}, t_{hx}, t_{hy}, t_{hz}, t_d$ are parameters that determine the region of influence with respect to the control point. Such an arrangement can be very useful in certain circumstances. For example, in order to deform the nose tip of a facial model, one can use the nose tip as the control point and define a region of influence as shown in Figure 3.8 to prevent other regions of the facial model from being deformed.

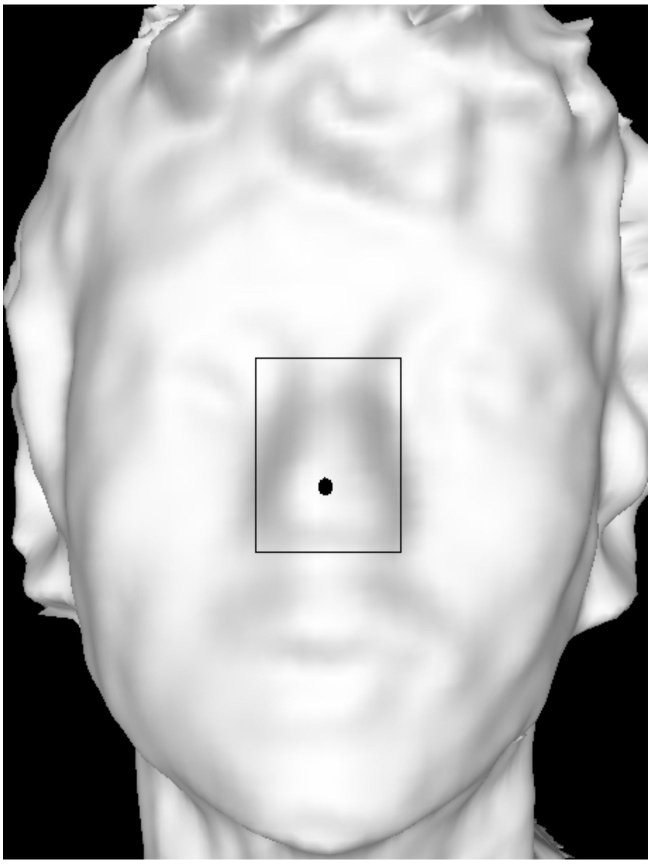


Figure 3.8: Defining a region-of-influence.

3.2.2.2 Deformation by Graph Distance to Control Point

The surface mesh of a 3D object can be considered as a finite graph in graph theory. The graph distance $d(u,v)$ between two vertices u and v of a finite graph is the minimum length of the paths connecting them. If no such path exists (i.e., if the vertices lie in different connected components), then the distance is set equal to ∞. Figure 3.9 illustrates the definition of graph distance.

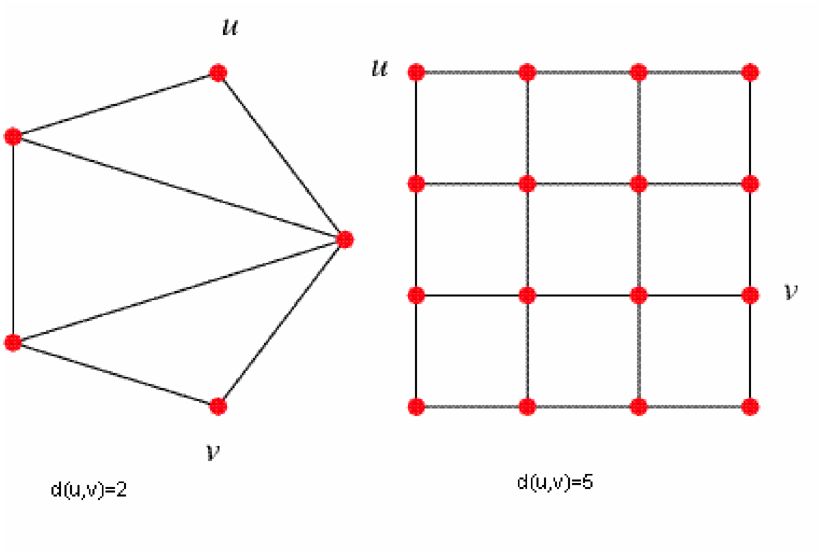


Figure 3.9: Graph distance.

Given a mesh and a control point C on the mesh (i.e., the control point itself is a vertex in the mesh), the translation of a vertex V on the mesh to a new position V' is given by

$$V' = V + W(d(C,V))$$

where $d(C,V)$ denotes the graph distance between C and V and W is a general vector function used to control the shape of deformation.

The basic version of deformation by graph distance generally works well. However, since we are dealing with high-resolution triangulated models in this research, selecting the control point (a vertex on the mesh) becomes a problem. Either text-based selection or graphical selection is difficult to implement and impractical for the user. Therefore, we decide to use a slight variation, where a control point is replaced by a control face. Compared to a sizeless vertex, it is much easier to ask the user to designate a face on a 3D object.

The new formulation is shown as follows. Given a mesh and a collection of control points $C_1, C_2, ..., C_n$ on the mesh, the translation of a vertex V on the mesh to a new position V' is given by

$$V' = V + W(D(C,V))$$

where

$$D(C,V) = \min\{d(C_1,V), d(C_2,V), ..., d(C_n,V)\}$$

and $d(C_i,V)$ denotes the graph distance between C_i and V . Figure 3.10 illustrates the idea of multiple control points.

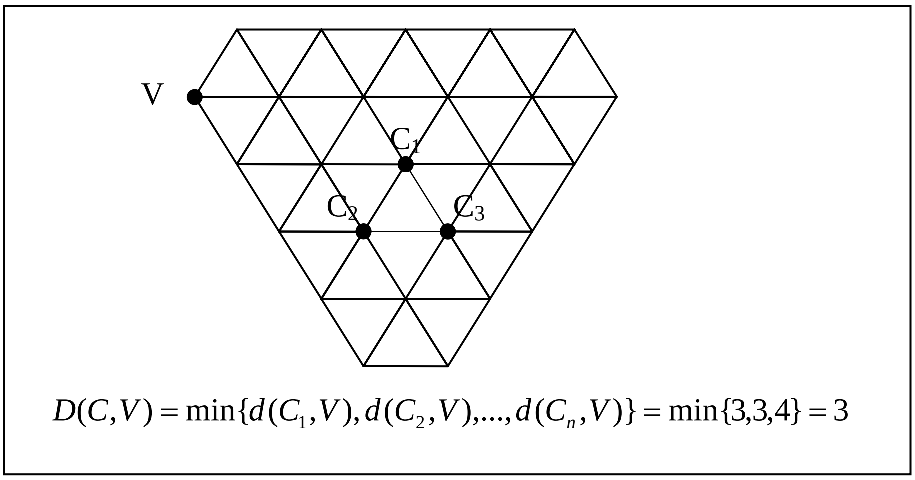

V
C1
C2
C3
$D(C,V) = \min\{d(C_1,V), d(C_2,V), ..., d(C_n,V)\} = \min\{3,3,4\} = 3$

Figure 3.10: Multiple control points in deformation by graph distance.

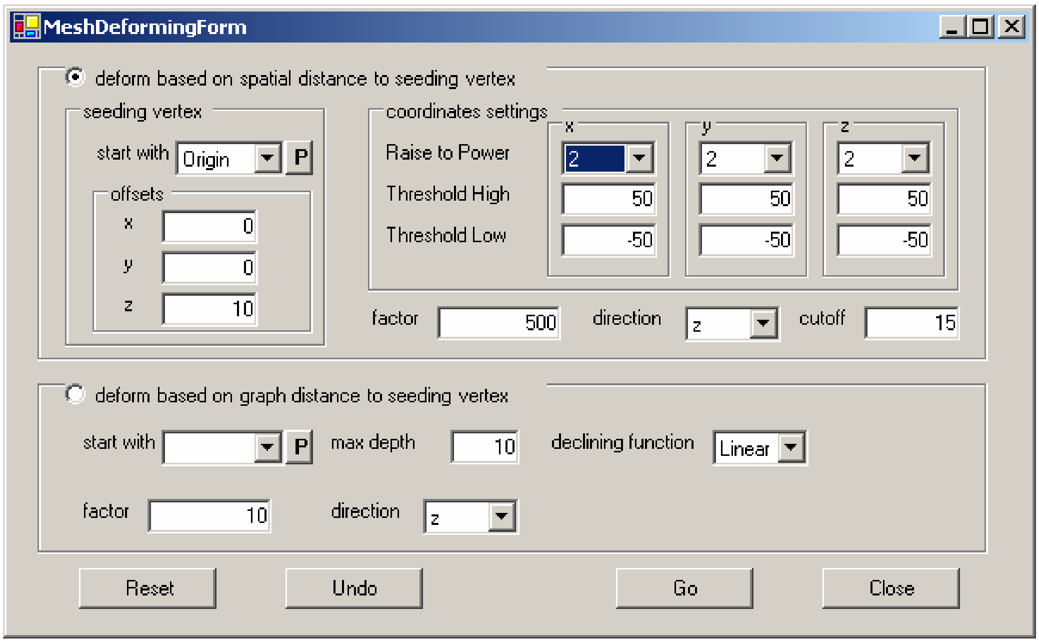
Similar to the first approach, we set up some criteria to define the region of influence.

$$D(V,C) < t_d$$

where t_d is user-definable parameter controling the maximum-depth that the deformation is allowed to propagate to. We also provide several options of W , the vector function that controls the shape of deformation. The user can choose a linear or Gaussian implementation of W .

3.2.3 Implementation

The mesh deformation algorithm is integrated as part of the application developed in Microsoft Visual Studio .NET. A screenshot of the related dialogue is shown in Figure 3.11.



● deform based on spatial distance to seeding vertex

seeding vertex

start with Origin P

offsets

x 0

y 0

z 10

coordinates settings

	x	y	z
Raise to Power	2	2	2
Threshold High	50	50	50
Threshold Low	-50	-50	-50

factor 500 direction z cutoff 15

○ deform based on graph distance to seeding vertex

start with P max depth 10 declining function Linear

factor 10 direction z

Figure 3.11: Mesh deformation dialogue.

The procedure of deforming a mesh using the implementation is shown in Figure 3.12.

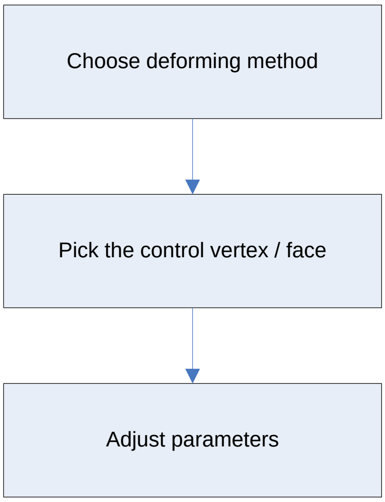


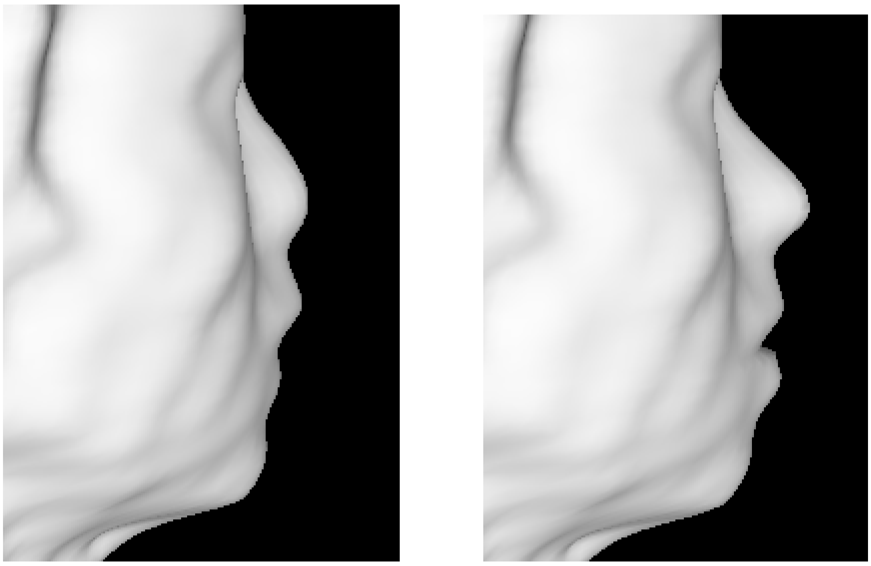
Figure 3.12: The procedure of deforming a mesh

The implementation also supports an unlimited undo operation; thus allowing users to experiment with different parameters freely before obtaining a satisfactory deformation.

3.2.4 Results

The experimental results show that the deformation algorithm works as expected. Users are able to perform free-form 3D deformation on any mesh models. Comparing the proposed deformation approaches, we can conclude that they differ in terms of the determination of spatial translation: the first approach relates the amount of spatial translation to the spatial distance between the vertex of interest and the control point, whereas the second approach uses graph distance instead of spatial distance to determine the amount of spatial translation. In deforming a regular (constant edge length) and relatively flat meshes, the two approaches give similar results.

Figure 3.13 shows a facial model before and after a series of 3D deformation, the objective of which is to modify the facial model, especially the region near the nose tip, such that its facial profile meets the one detected from the corresponding cephalogram. The deformation is done by using the first approach, i.e., deformation by spatial distance to control points.



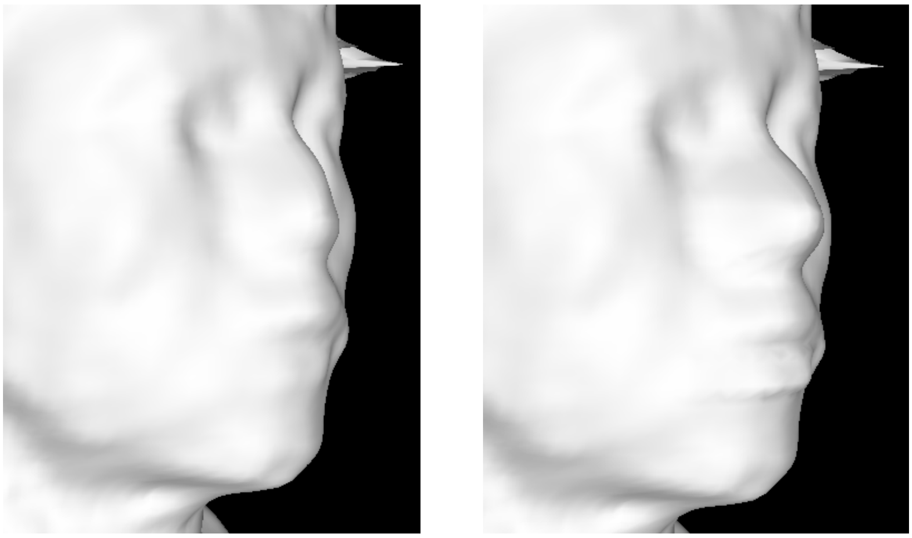


Figure 3.13: Facial model before and after deformation.

3.3 Registration of Facial Model and Cephalogram

3.3.1 Overview

The objective of registering a 3D facial model and a cephalogram is to establish a spatial relationship between a 3D facial model and study models by using the cephalogram as the intermediate reference. However, such multi-modality registration is not a trivial task. This part of the research is a continuation of previous work, where the registration algorithm that was implemented had limitations.

The 3D geometry discussion in this thesis is based on the following axis definition. (Figure 3.14)

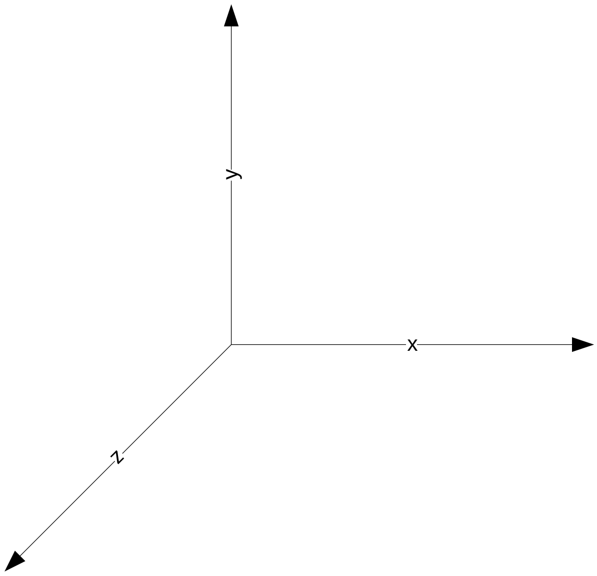


Figure 3.14: Axis definition

3.3.2 Existing Algorithm

An early attempt in developing the algorithm of registration of facial model and cephalogram was proposed by Zhao [14] in 2003. A brief description of the algorithm is given below.

The algorithm consists of three major steps.

1. Detection of facial profile on the lateral cephalogram – This step mainly focuses on the analysis of the cephalogram, which undergoes the 2D image processing techniques of median filtering, logarithm filtering and Canny filtering before a facial profile is detected (Figure 3.15).

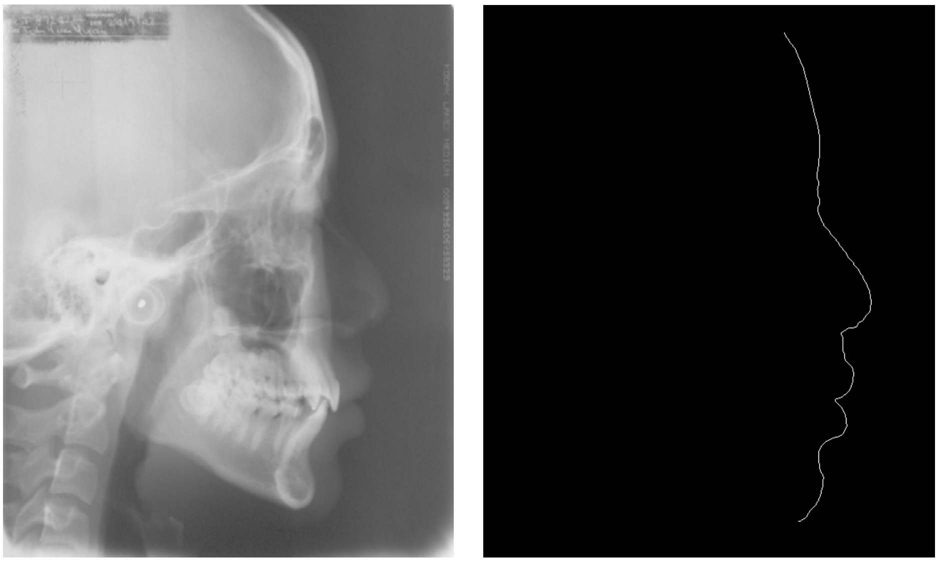


Figure 3.15: Detected facial profile from the cephalogram.

2. Detection of the partial facial profile on the facial model – With an objective similar to step I, the 3D facial model is analyzed in order to detect the facial profile in the form of a series of discrete points on the facial model (Figure 3.16). The steps of the algorithm are:
    a. Pre-Rotation – The facial model is rotated around the y axis by 90 degrees.
    b. Horizontal slicing – The vertically central portion of the 3D facial model, which has a y coordinate between 0 and 26, is sliced along the y axis into evenly-separated fine slices.
    c. Analysis of slices – Considered as a 2D image, each slice is analyzed to find out the point that corresponds to the nose ridge.
    d. Post-rotation – The facial model is rotated around the y axis again, such that the detected nose ridge points lie on plane $z = 0$.

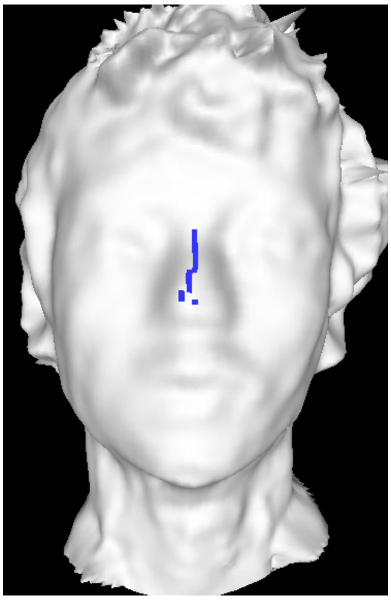


Figure 3.16: Detected feature points on a facial model.

3. Registration of facial model and cephalogram – Correspondence between the facial model and the cephalogram is established by comparing the detected facial profiles, one from the cephalogram and the other from the facial model. Finally, the cephalogram is mapped onto plane $z = 0$ with the calculated size and offset parameters.

3.3.3 Problem and Analysis

The results show that the existing algorithm gives fairly remarkable results. Both the facial profiles are correctly detected. However, the final registration result is not satisfactory. Although the detected facial profiles are successfully registered, the mapped cephalogram does not conform to the center (the plane of symmetry) of the head, which is easily visible from various viewing angles of the model (Figure 3.17).

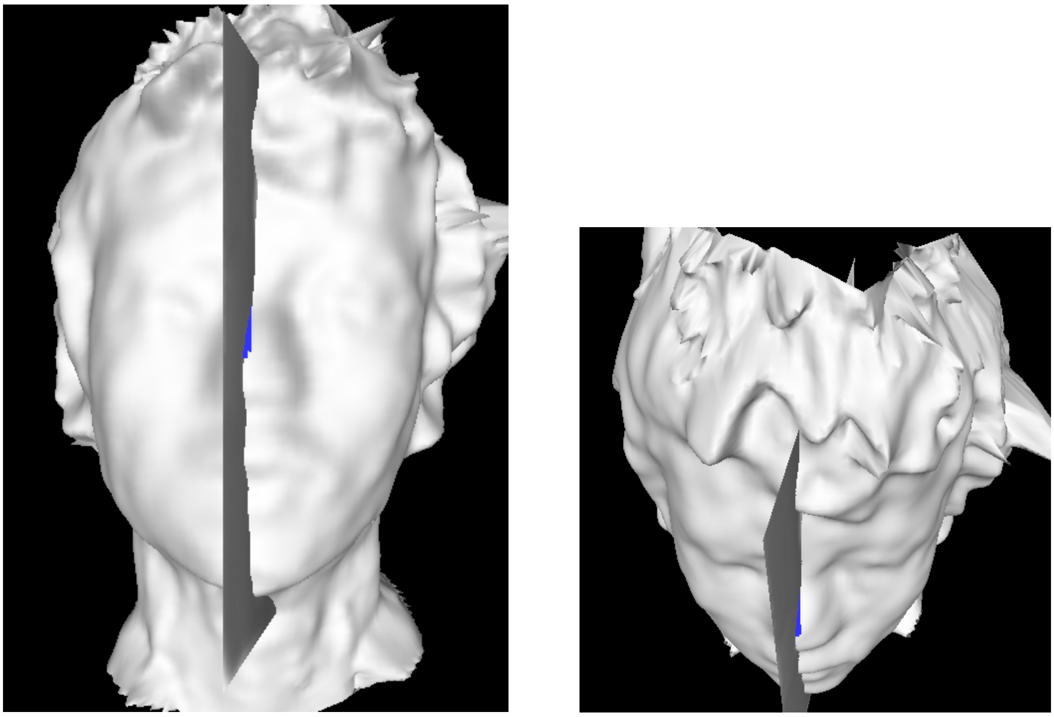


Figure 3.17: Registration result given by the existing algorithm.

In order to investigate the reason behind the unsatisfactory result, we first look at the assumptions made by the above algorithm.

1. The initial facial model has to be positioned correctly in the virtual environment. More specifically, the facial model must:
   - Appear as a straight front view when viewing from the default view port, i.e., stationing on the positive z axis and looking into the $-ve$ direction of the z axis.
   - Be stationed in a height such that the y coordinate of the nose is between 0 mm and 26 mm.
2. The plane of symmetry of the facial model (the plane that passes through the mid-line of the face and divides the entire human head into mirrored halves) must pass through the origin in the virtual environment.

Assumption I ensures that the facial profile from the facial model would be detected correctly. From the test result, we can conclude that this assumption is implicitly enforced by the data-acquisition system, Geometrix's FaceVision 600.

Assumption II is an implicit assumption that is not discoverable unless the algorithm is carefully analyzed. Since the cephalogram is the side view of a human head, the only correct way of mapping it into the 3D environment is to map it to the symmetrical plane of the facial model. However, we notice that through the entire algorithm, the facial model undergoes no translation but only rotation around the y axis, which essentially means that the distance between the origin and the symmetrical plane of the head is not affected. The cephalogram is then mapped onto plane $z = 0$, i.e. the cephalogram passes through the origin at the end of the registration process. To conclude, unless the plane of symmetry of the facial model passes through the origin in the initial position, the algorithm will not be able to register the facial model and the cephalogram correctly.

The following figures (Figure 3.18 – 3.20) illustrate this assumption. All the three figures show top views of the facial model, which appears in a symbolic form in the figures. Figure 3.18 shows the situation where the facial model is rotated to a position where the cephalogram can be correctly mapped. Figure 3.19 shows the necessary initial condition in order to achieve the result in Figure 3.18. Figure 3.20 shows an undesirable initial condition under which the algorithm fails to do the registration. The dashed line in Figure 3.19 and Figure 3.20 represents the plane of symmetry of the facial model.

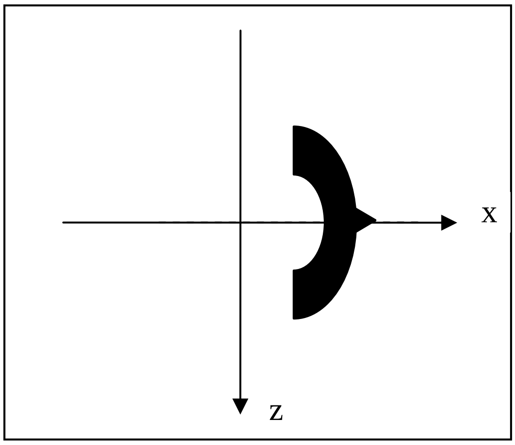


Figure 3.18: Correct result after rotation.

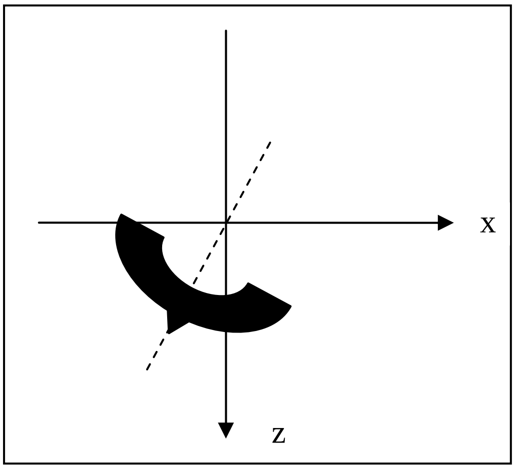


Figure 3.19: Desirable initial condition.

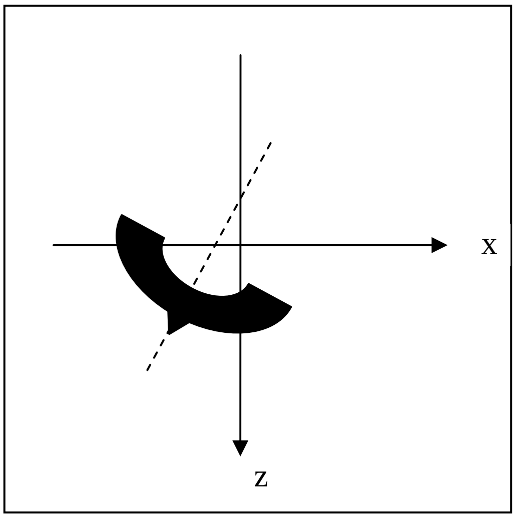


Figure 3.20: Undesirable initial condition.

After examining the output data from FaceVision 600, we confirm that it is the violation of assumption II that causes the unsatisfactory result given by the algorithm.

3.3.4 Solutions

The straightforward idea is to satisfy the problem-causing assumption by translating the facial model such that the symmetrical plane of the human head passes through the origin. However, determining the symmetrical plane of a 3D model itself is a difficult problem, not to mention that our facial model does not provide a complete view of the human head.

We decide to design a more problem-specific solution to the problem. A pre-calibration procedure was introduced to solve the problem. The objective of this pre-calibration procedure is to make the problem-causing assumption valid prior to applying the registration algorithm.

The calibration procedure contains two steps. Firstly, certain feature points are determined on a facial model, based on which a translation vector is calculated; by applying this translation vector to the facial model, the assumption required by the existing algorithm is satisfied.

3.3.4.1 Determination of Facial Landmarks

We propose two methods in determining facial landmarks from a facial model. The first method is a fully automatic method which requires no user interaction. The second method, where user interaction is required, is designed to handle models with high degree of ambiguity.

We decide to use the eye centers on the facial model as the landmarks. The reason is that the eyes are one of the most prominent landmarks on a human face. Moreover, the patients that we are mainly concerned about are patients undergoing craniofacial surgeries, most of whom have symmetrical eyes.

The automatic detection of eye centers is conducted on the 2D texture image (Figure 3.21). This image is provided by FaceVision 600 together with every facial model captured, and is a color image containing the texture information of a facial model. Compared to a binary 3D image (a facial model), it is much easier to detect the eye centers from such a true-color 2D image.

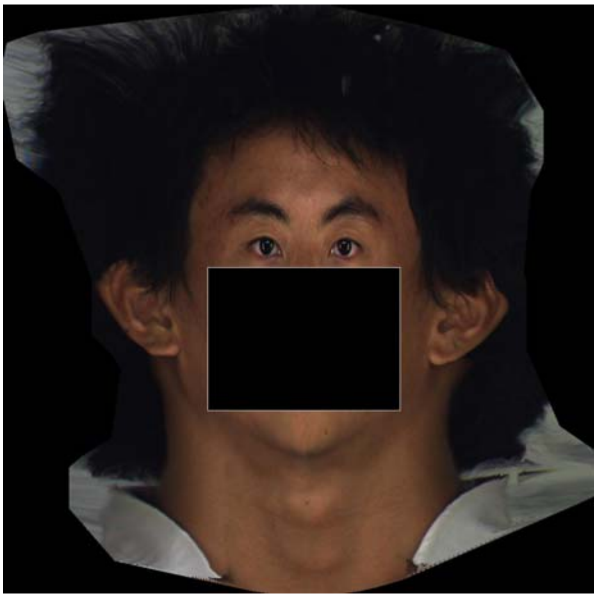
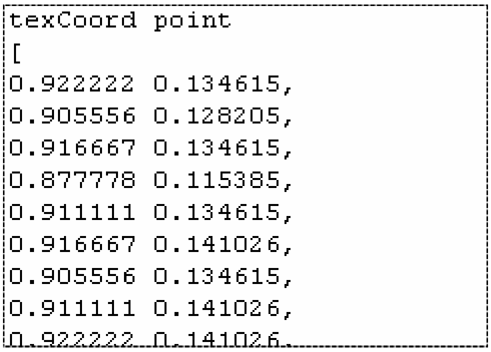


Figure 3.21: 2D texture image
(with part of the face covered to ensure anonymity).

In order to make use of the texture image, a one-to-one relationship between vertices and texture coordinates is defined in the corresponding VRML file. In addition to the vertex list and face list presented a VRML file (refer to Section 3.2.1), a texture coordinate list is also defined in a VRML file which contains texture information. Figure 3.22 shows the beginning portion of a texture coordinate list. By strictly following the order of the vertex list, we obtain a texture coordinate list that stores the 2D coordinates of texture points corresponding to the vertices in the vertex list.



```
texCoord point
[
0.922222 0.134615,
0.905556 0.128205,
0.916667 0.134615,
0.877778 0.115385,
0.911111 0.134615,
0.916667 0.141026,
0.905556 0.134615,
0.911111 0.141026,
0.922222 0.141026,
```

Figure 3.22: A texture coordinate list in a VRML file.

The following algorithm was designed to detect the eye centers from a facial model,

I. Detect the centers of the eyes from a 2D texture image:

a. Define a region of interests which contains primarily the face of the patient by trimming the original texture image and taking the center portion.

b. Convert the image from true color to grayscale by eliminating the hue and saturation information while retaining the luminance.

c. Threshold the image using Otsu’s method.

d. Invert the image.

e. Perform closing operation with a circular structural element (diameter =5).

f. Perform erosion operation with a circular structural element (diameter =6).

g. Perform connected component labeling and feature measurement to each components.

h. Determine the pair of components corresponding to the eyes by using these features: area, ratio between the longest and shortest axis, relative centroid location.

II. Determine the 3D coordinates of vertices with their texture coordinates closest to the detected centers of the eyes.

The second method of determining facial landmarks involves user interaction. After enhancing the displayed model by texture-mapping, we created an environment in which the users can accurately designate the facial landmarks. (Figure 3.23)

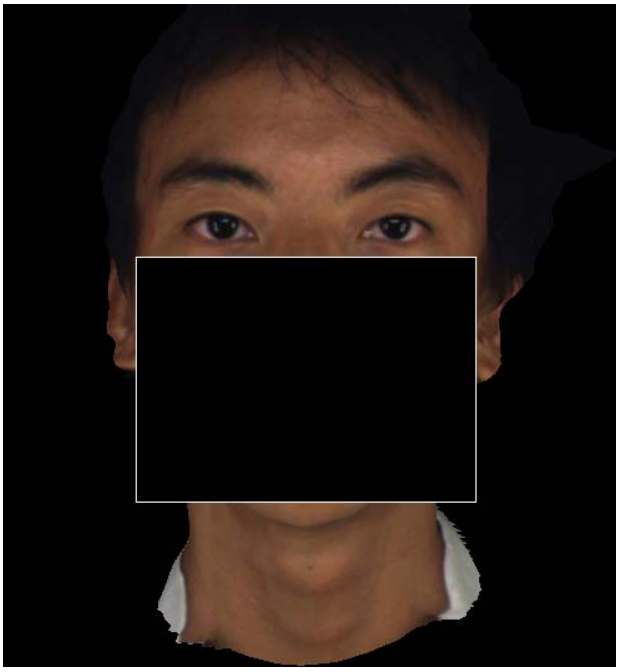


Figure 3.23: Facial model with texture-mapping

It should be noted that in the second method, users can designate any two symmetrical points on the facial model as facial landmarks that are not limited to the centers of the eyes.

3.3.4.2 Model Translation

Once the facial landmarks are obtained through automatic detection or user interaction, the symmetrical plane is determined, based on which the facial model is translated such that the plane of symmetry passes through the origin.

It should be noted that this solution also assumes that the facial model has the correct pose in the 3D environment, i.e. it is not tilted in the z direction, which is part of the assumptions in the existing algorithm. This assumption effectively makes the 3D analysis a 2D one.

Given a facial model and two facial landmarks, $L(x_l, y_l, z_l)$ and $R(x_r, y_r, z_r)$, because of the assumption, we know that $y_l = y_r$ and the projection of the symmetrical plane of the facial model is a straight line in plane $y = y_l$, which is also the perpendicular bisector P of line LR (Figure 3.24).

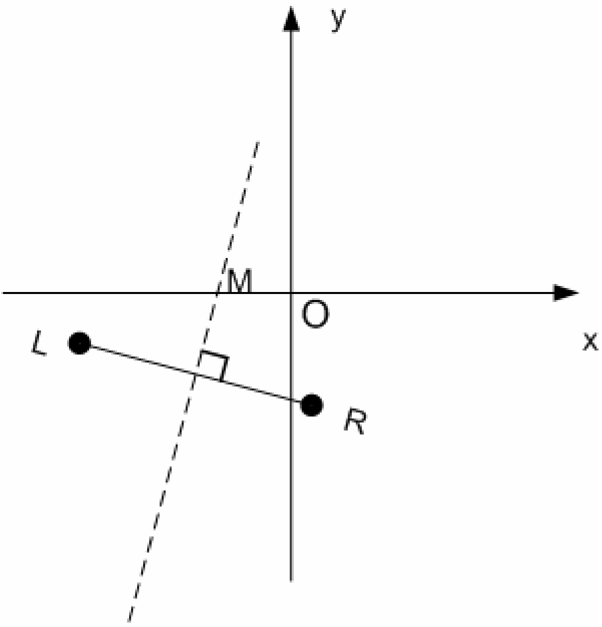


Figure 3.24: Geometric analysis in the solution (top view).

Since we are restricted by the existing algorithm, we cannot simply translate the model such that the midpoint of LR passes through the origin O, which would potentially affect the polar coordinates analysis of the registration algorithm and hence affect subsequent operations. Under such a constraint, only translation in the x direction is permitted. Therefore, the objective here is to translate the facial model in the x direction such that P passes through the origin.

Since P passes through the midpoint of LR, $(\frac{x_l + x_r}{2}, \frac{z_l + z_r}{2})$, the slope of P is

$$-\frac{1}{m}$$

where m is the slope of LR

Therefore, the slope of P is

$$-\frac{1}{\frac{z_r - z_l}{x_r - x_l}}, \text{ or } \frac{x_l - x_r}{z_r - z_l}$$

The equation of P can be written as

$$z = \frac{x_l - x_r}{z_r - z_l}(x - \frac{x_l + x_r}{2}) + \frac{z_l + z_r}{2}$$

Substituting $z = 0$, we obtained the x coordinate of the intersection point M

$$x = -\frac{\frac{z_l + z_r}{2}}{\frac{x_l - x_r}{z_r - z_l}} + \frac{x_l + x_r}{2} = \frac{(z_l + z_r)(z_r - z_l)}{2(x_l - x_r)} + \frac{x_l + x_r}{2}$$

The following translation vector

$$V = (-\frac{(z_l + z_r)(z_r - z_l)}{2(x_l - x_r)} - \frac{x_l + x_r}{2}, 0, 0)$$

is applied to the facial model, after which the assumption will be satisfied, i.e. the plane of symmetry of the facial model passes through the origin:

Figure 3.25 shows the result after the translation, where the necessary assumption has been satisfied.

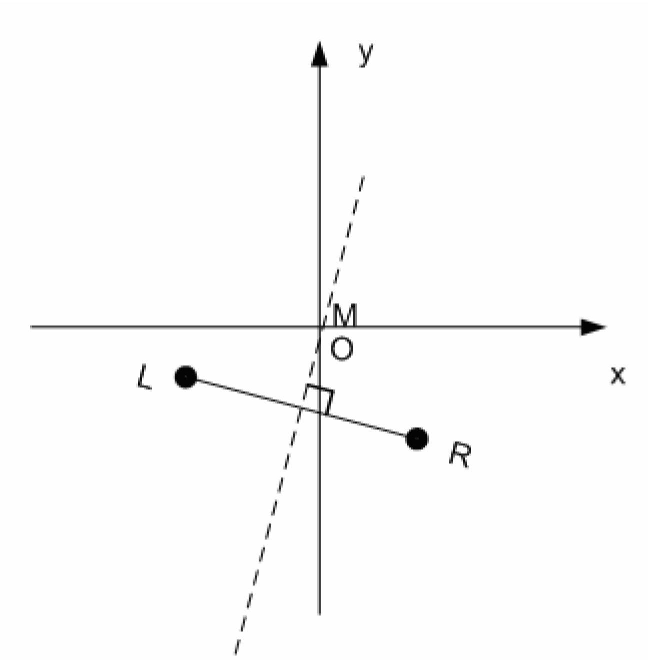


Figure 3.25: Geometric analysis after translation (top view).

3.3.5 Results

3.3.5.1 Automatic Landmark Detection

Experimental results indicate that the automatic landmark detection algorithm works in most cases. Out of 110 test images,

- In 103 images, the centers of the eyes were correctly detected.
- In 7 images, the detection was incorrect due to the following causes:
  - The patient's eyes were narrowed (one image).
  - The patient's head was not positioned properly, i.e., at the center of the image when the facial model was taken (three images).
  - The patient's head was not kept stationary when the facial model was taken (three images).

The high success rate makes the automatic detection method an ideal choice in handling normal cases where the facial model is correctly taken. Manual determination of facial landmarks is only required in handling highly ambiguous images, such as narrowed eyes.

3.3.5.2 Model Translation

The suggested solution gives satisfactory results. After applying the pre-calibration process prior to the existing algorithm, the registration result is greatly improved, where the cephalogram is perfectly mapped to the center of the facial model. Figure 3.26 shows the result.

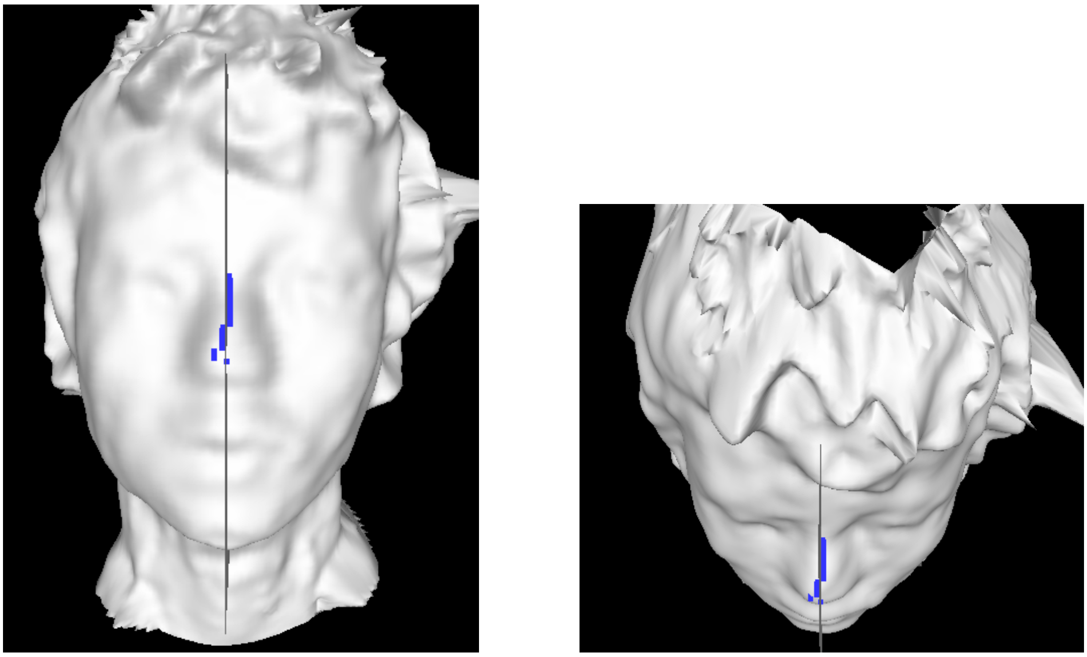


Figure 3.26: Results obtained by applying calibration solution.

In order to investigate the quality of result in a more direct manner, we have conducted the following validation procedure. The 3D registration result is sliced horizontally at the height of the nose tip, resulting the following cross sectional plot (Figure 3.27 and Figure 3.28).

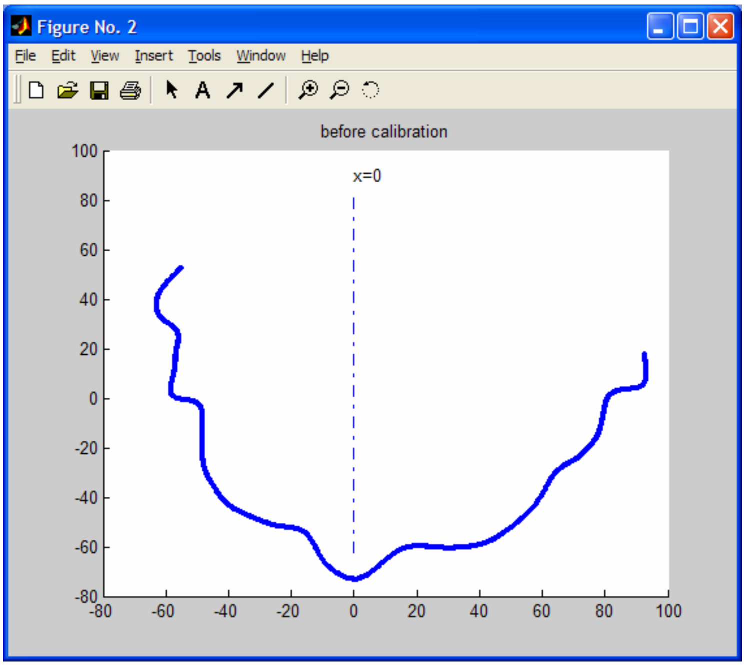


Figure 3.27: Cross sectional plot at the nose tip (before calibration).

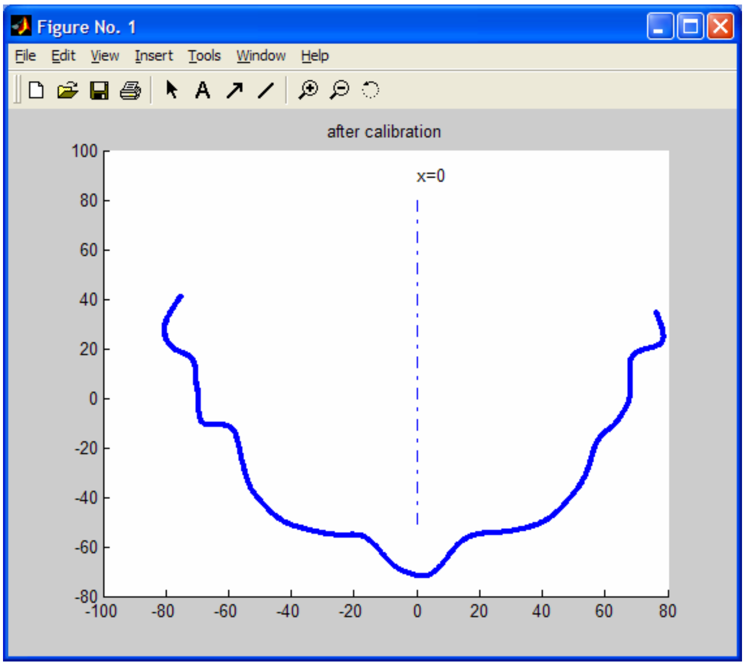


Figure 3.28: Cross sectional plot at the nose tip (after calibration).

From the plot, it is obvious that the improved algorithm (after applying the calibration solution mentioned previously) gives a much better registration result than the previous algorithm. The facial model is translated and rotated to a correct orientation to fit the cephalogram, which is shown as a dashed line in the figures.

Flipping the right half of the facial model to the left with respect to the cephalogram makes the improvement more obvious. (Figure 3.29 and Figure 3.30)

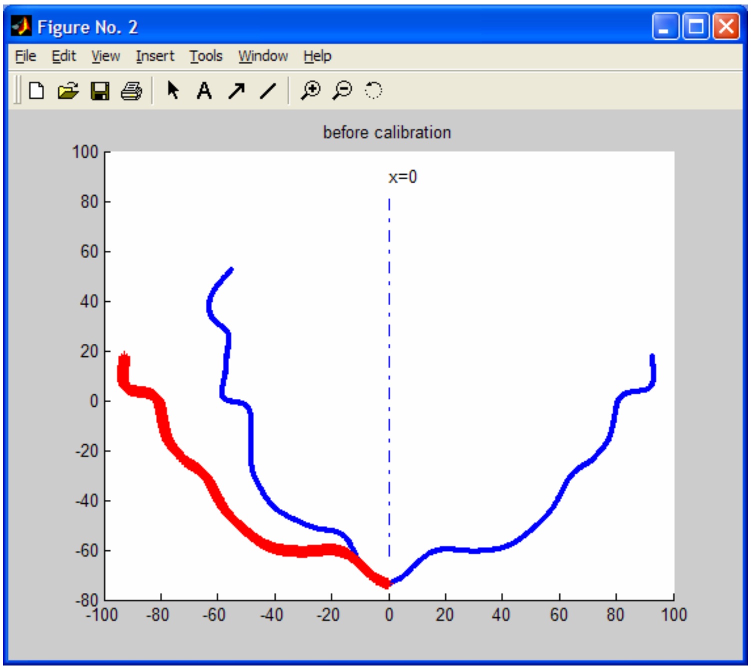


Figure 3.29: Cross sectional plot at the nose tip (before calibration, with flipping).

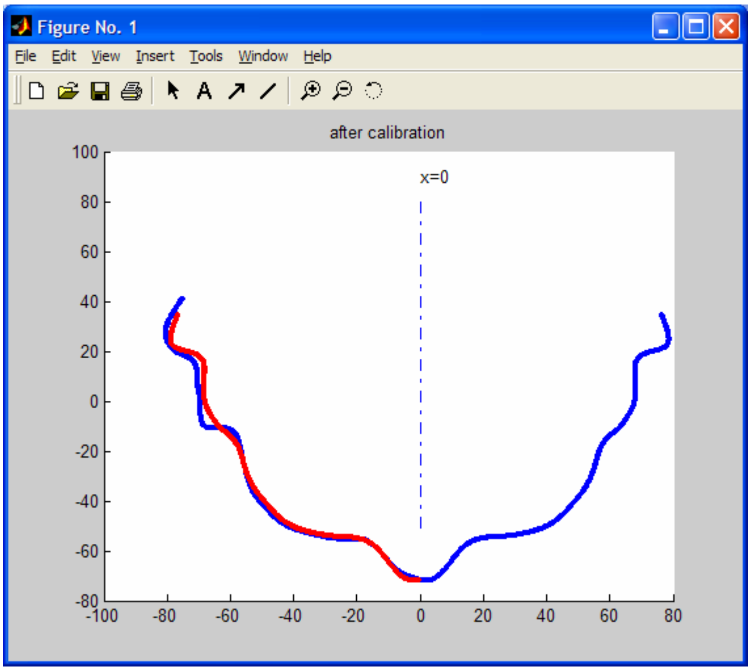


Figure 3.30: Cross sectional plot at the nose tip (after calibration, with flipping).

Quantitatively, the degree of symmetry of the rotated facial model about the cephalogram can be evaluated by integrating the cross sectional plot and comparing the result obtained from both sides of the cephalogram. Figure 3.31 illustrates the method used in gathering the data. A threshold value for y is chosen to define the region to be integrated. The cross sectional plot is then divided into two halves by the midline $x = 0$, each of which is integrated using trapezoid method. The degree of symmetry of the rotated facial model is then evaluated by comparing the integration results of the two halves.

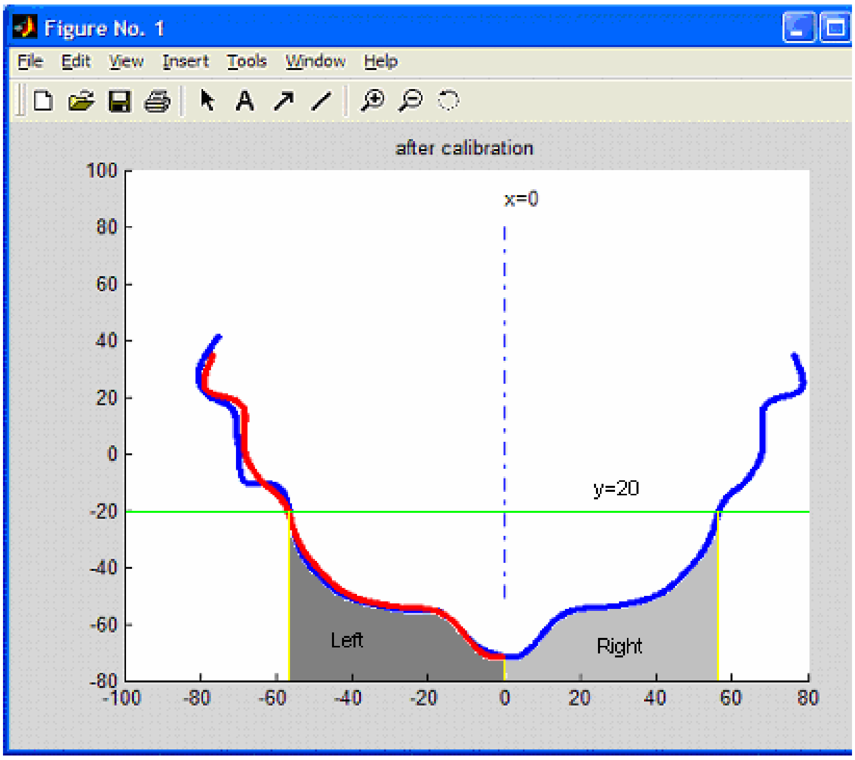


Figure 3.31: Evaluating the symmetry.

The result is shown in Table 3.1:

Table 3.1: Integration of cross sectional plot of registered facial model

y Threshold (mm)	Area (without pre-calibration)			Area (with pre-calibration)		
	Left	Right	Difference	Left	Right	Difference
-70	407.0	351.6	7.3%	273.0	307.8	6.0%
-60	819.0	1450.9	27.8%	848.4	746.0	6.4%
-50	1696.7	3149.6	30.0%	2464.8	2253.4	4.5%
-40	2318.1	3476.1	20.0%	2851.6	2650.5	3.7%
-30	2486.9	3649.7	19.0%	2994.5	2833.4	2.8%
-20	2521.4	3909.9	21.6%	3041.5	2898.4	2.4%
-10	2521.1	3976.8	22.4%	3179.2	2995.1	3.0%
0	2533.1	3985.7	22.3%	3191.1	3021.4	2.7%
10	2538.7	3943.2	21.7%	3187.5	3021.6	2.7%
20	2551.0	3951.6	21.5%	3068.5	2911.3	2.6%

Obviously, the rotation result with pre-calibration has a much higher degree of symmetry than the result without pre-calibration.

3.4 Pre/Post Treatment Comparison

3.4.1 Overview

This section of describes the visualization of the study model to reflect the effect of orthodontics treatment. Two sets of study models and lateral cephalogram are taken before and after treatment; after which the study models are registered in 3D space using the respective cephalogram as the reference. Clinically useful data, such as tooth movement and jaw movement, can be measured after the study models are registered precisely in the 3D space.

The procedure of registering pre-treatment and post-treatment study models is illustrated in Figure 3.32. Both study models are first registered respectively with their corresponding cephalogram; at the same time the cephalograms also undergo a 2D registration process. Finally, registration between the study models is completed by combining the above three sets of registration results. By using the cephalogram as a bridge, we have avoided the complicated process of registering the 3D study models directly. The resulting algorithm is expected to be simple yet reliable.

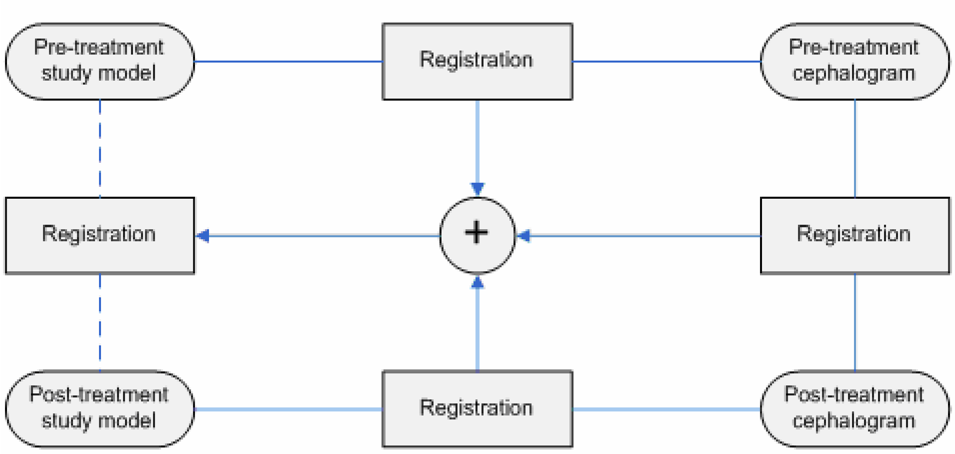

Pre-treatment study model
Registration
Pre-treatment cephalogram
Registration
+
Registration
Post-treatment study model
Registration
Post-treatment cephalogram

Figure 3.32: Registration of pre-treatment and post-preatment study models.

3.4.2 Algorithm

The registration algorithm consists of the following parts:

- Registration of a pair of study models (pre-treatment and post-treatment) and their corresponding cephalograms, respectively.
- Registration of pre-treatment and post-treatment cephalograms.
- Combining of registration results.

3.4.2.1 Registration of Study Model and Cephalogram

Registration of study model and cephalogram requires minimum user interaction. The cephalogram appears as a perfect side-view of the study model. In order to establish the correspondence of a study model and its cephalogram, we firstly transform the study model into its standard position as defined in the previous section. It is easy to see that in such case, a cephalogram should be placed in parallel to plane $x = 0$ to achieve a correct registration. Under this constraint, we need only two sets of point correspondence to complete the registration.

The cephalogram is first resized to actual size through manual calibration to eliminate the effect of magnification factor during imaging and scanning. After that, the cephalogram is positioned at plane $x = 0$ with its centroid lying at the origin. Once the cephalogram is correctly positioned, we can easily convert any user-select feature points on the cephalogram to 3D coordinates.

In order to perform the registration, the user needs to indicate precisely two prominent feature points, the incisor tip and the second molar, on both the study model and the cephalogram. It should be noted that the choice of feature points is unrestricted. Users can choose any prominent feature points under their preferences and the registration quality would not be affected.

The two pairs of corresponding points are

- Incisor tip on study model and cephalogram, $v_{it}(x_{it}, y_{it}, z_{it})$ and $v'_{it}(0, y'_{it}, z'_{it})$.
- Second molar, $v_{sm}(x_{sm}, y_{sm}, z_{sm})$ and $v'_{sm}(0, y'_{sm}, z'_{sm})$.

The study model is first rotated about the x axis by θ to meet the angle of the occlusal plane defined in the cephalogram, where

$$\theta = \tan^{-1}(\frac{y'_{sm} - y'_{it}}{z'_{sm} - z'_{it}}) - \tan^{-1}(\frac{y_{sm} - y_{it}}{z_{sm} - z_{it}})$$

Defining $v''_{it}(x''_{it}, y''_{it}, z''_{it})$ and $v''_{sm}(x''_{sm}, y''_{sm}, z''_{sm})$ as the points after rotating v_{it} and v_{sm} about the x axis by θ, the registration is completed by further translating the study model by vector (T_x, T_y, T_z)

$$T_x = \frac{(x'_{it} - x''_{it}) + (x'_{sm} - x''_{sm})}{2}$$

$$T_y = \frac{(y'_{it} - y''_{it}) + (y'_{sm} - y''_{sm})}{2}$$

$$T_z = 0$$

3.4.2.2 Registration of Pre-Treatment and Post-Treatment cephalogram

Although orthodontic treatment might have brought significant change to the craniofacial complex, the pre-treatment and post-treatment cephalogram can still be easily registered with reference to other clinical features that are not affected by the treatment, such as the spine and skull.

This step is completed manually by the user. The registration result, in terms of the 2D transformation matrix, is prepared for the final step.

3.4.2.3 Combining registration results

At this point we have all the necessary data to establish the correspondence between the pre-treatment and post-treatment study models. By simply applying the transformation matrix obtained in the previous step to one of the study models, we obtain a study model which can be directly superimposed onto the other one.

Once the pre-treatment and post-treatment models are successfully registered, measurements such as tooth movement can be easily made.

3.4.3 Results

The experimental results of the registration process are satisfactory, as illustrated in the following figures.

Figure 3.33 shows the registration result of an upper study model and its corresponding cephalogram. We can see that the cephalogram is precisely aligned with the study model such that it passes through the center of the study model (between two first incisors) with the projected boundary of crowns matching the one on the cephalogram.

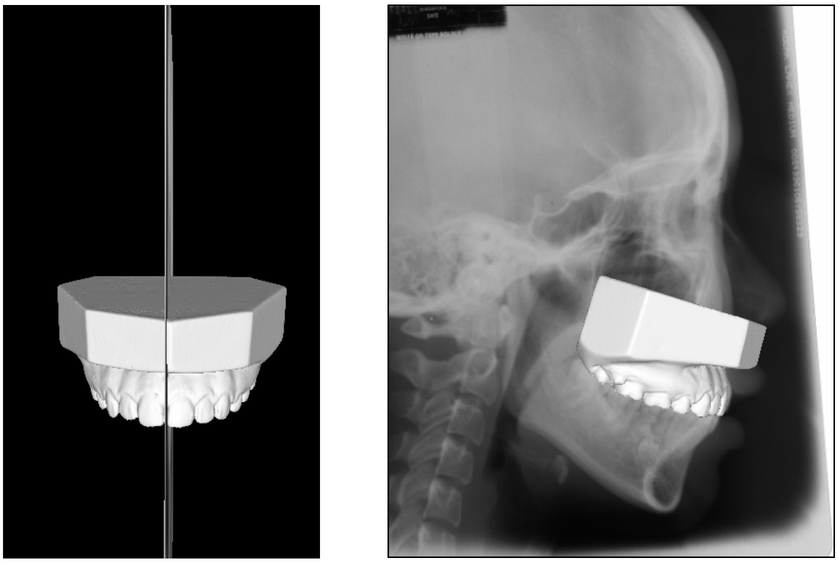


Figure 3.33: Registration result of a study model and its cephalogram.

Figure 3.34 illustrates the registration result of pre-treatment and post-treatment cephalograms, where the two cephalograms are rendered in different colors (blue and yellow) in order to distinguish them.

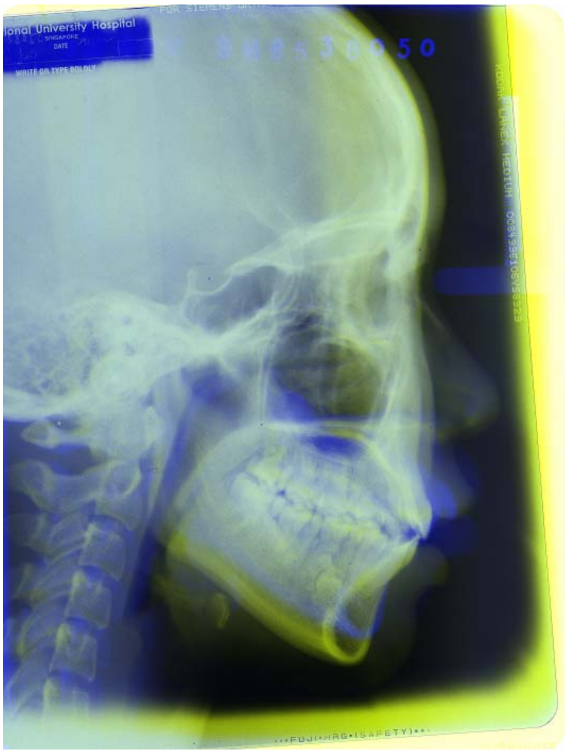


Figure 3.34: Registered cephalogram

Figure 3.35 shows two superimposed study models, representing pre-treatment status and post-treatment status, respectively.

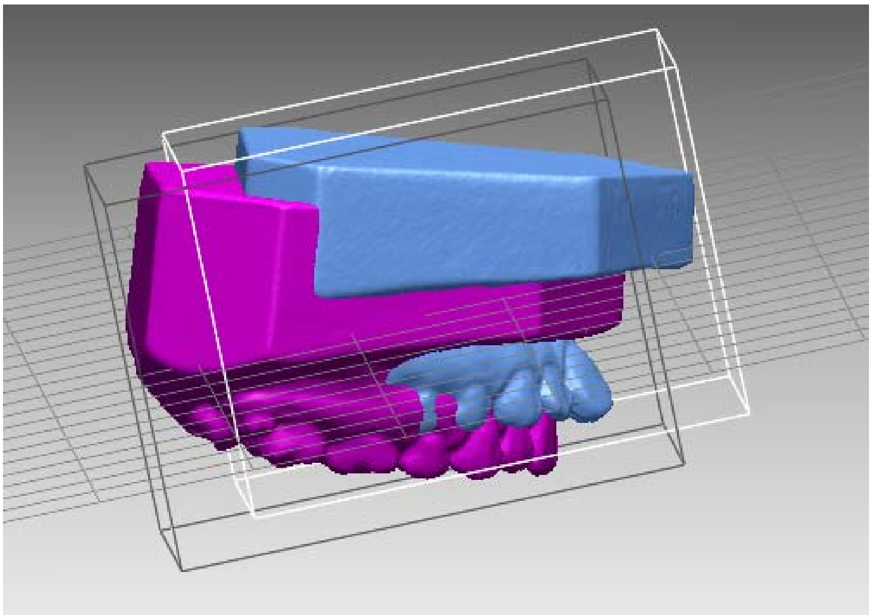


Figure 3.35: Superimposed study models.

Chapter 4

Study Model and Orthopantomogram

4.1 Introduction

Computer-aided orthodontics has become a popular research topic in recent years. Computer vision techniques have been applied to 3D orthodontics imaging [6] [15]. The prediction of various treatment effects using software was reported in [16]-[19]. Computerized techniques have also been employed in statistical studies [20] as well as in surgery analysis [21]. However, due to the lack of suitable computer-aided tools, orthodontic treatment planning is still largely based on the clinical expertise of orthodontists. Beer [22] presented the general problems of developing an orthodontic treatment planning system, which included dentition modeling and simulation of treatment effects.

Currently, orthodontic treatments depend to a large extent on the judgment of the clinician. Appliances are continuously adjusted in order to correct teeth trajectories during the treatment.

The process of computer-aided orthodontic treatment can usually be divided into three steps:

- Data acquisition.
- Data manipulation.

- Visualization of treatment effects.

Data acquisition is concerned with the process whereby physical orthodontics data is scanned, transferred to the computer, and ultimately modeled in the software. The device commonly used in this application is the 3D surface laser scanner, which not only provides information about the length and width but also the depth of objects. Kusnoto [9] suggested that the surface laser scanner has great research potential because of its accuracy and ease of use. The reliability comparison between digital models and plaster models [10] confirms that the magnitude error is clinically negligible.

Data manipulation is the major step in orthodontics visualization It is where the different input data undergoes processing, and finally is integrated to form a complete data set. Typical data manipulation techniques include:

- Segmentation - input data is logically divided into discrete objects, as in:
  - extracting individual crowns from a study model.
  - separating muscles and soft-tissues in a CT scan.
- Registration –correspondence is established between different sets of input data or between input data and physical space, as in:
  - registration of crowns obtained from a study model and roots generated from deforming teeth with reference to the cephalogram.
  - registration of a 3D facial model and the corresponding cephalogram.
- Integration – integrating different input data in one virtual environment, as in
  - integration of 2D and 3D data.
  - integration of surface mesh data and volumetric data.

In the next step, visualization of treatment effects is enabled via software simulation. Orthodontists would be able to decide the orthodontic treatment plan with the help of various simulation tools, including measurement of diagnosis parameters, tooth movements, and eventually bio-mechanic simulation of different orthodontic treatment plans.

In general, the goal of a computer-aided orthodontic treatment planning system is to provide a reliable way of visualizing the treatment effect before it is clinically applied, which can potentially reduce the pain induced by traditional orthodontic planning before the treatment.

Alcaniz [23] first applied the concept of computer-aided orthodontics treatment planning in 1996. As part of the so-called MAGALLANES diagnosis system, the treatment planning system was designed to replace manual measurement methods, which used costly plaster models, by computer measurement methods and simulation of tooth movement. Being an early attempt, the above-mentioned system had presented the feasibility of computer-aided diagnosis, visualization and simulation tools in the planning of orthodontic treatment.

A computer-aided orthodontic treatment process (SureSmile® process [4]) was recently introduced. In this process, 3D dentition information is obtained directly from the patient’s mouth using a hand-held scanner. Treatment planning is done using software and a customized arch is built using robots. It is claimed that minimum tooth movement is achieved by using this process. Table 4.1 compares conventional orthodontics treatment and the SureSmile® process in three major aspects.

Table 4.1: Comparison between conventional orthodontics treatment and the SureSmile® Process

	Conventional Orthodontics Treatment	The SureSmile® Process

Diagnosis	Diagnosis relies on study model and orthopantomogram	3D data is captured by OraScanner^{TM}, a handheld light scanner
Treatment Planning	Treatment plan is developed over time by viewing progress at frequent monitoring visits	Treatment planning is done via computer simulation, which potentially moves the teeth in a more direct manner and hence reduces the treatment time
Fabrication of the Arch Wire	Manually bent by orthodontists with the presence of the patients	Automatically prepared by arch bending robotics

Figure 4.1 illustrates the SureSmile® process.

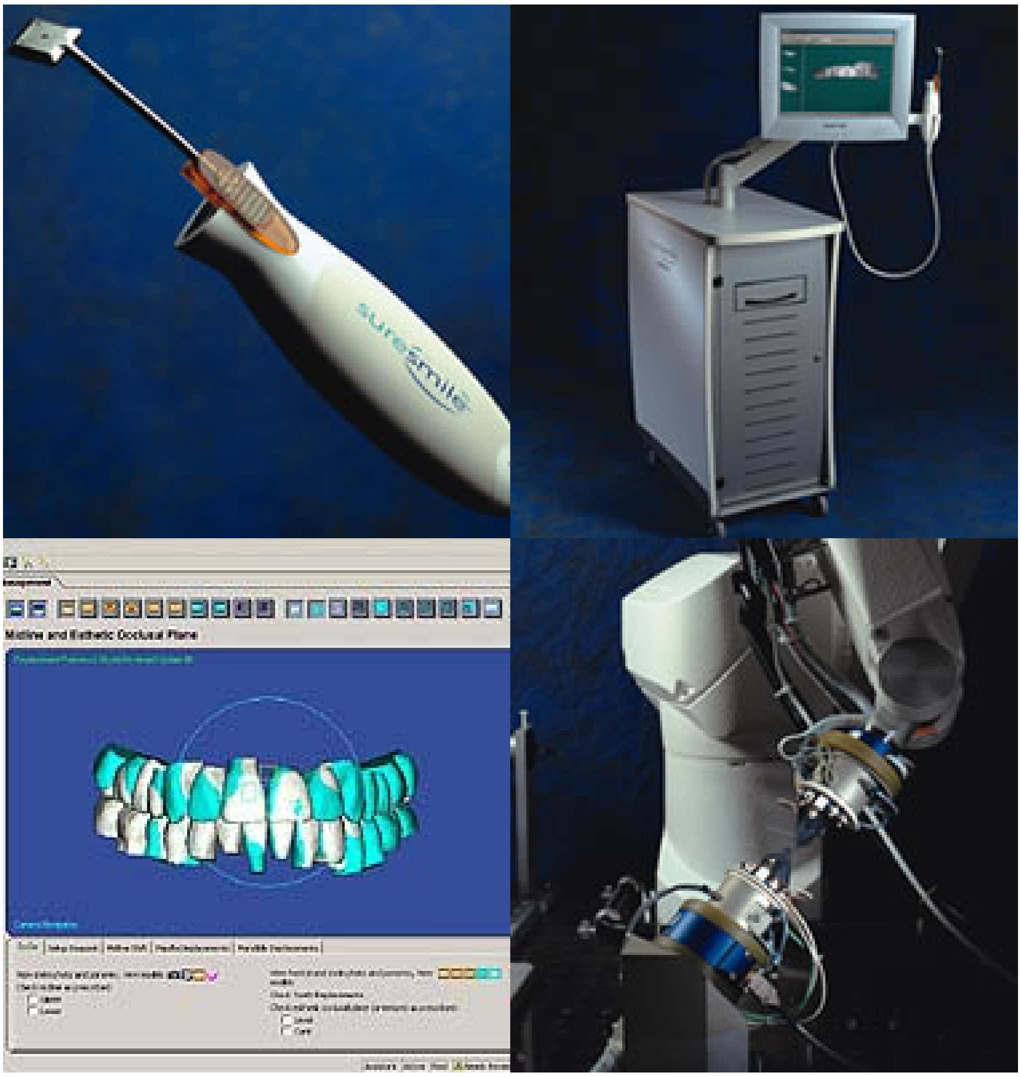


Figure 4.1 : the SureSmile® process: the scanner, the mobile scanner station, the simulation software, the arch-bending robotics

As a representative of computer-aided orthodontics process, the SureSmile® process demonstrates the feasibility of replacing conventional orthodontics treatment planning with a computer simulation process. However, it has some limitations. First of all, the SureSmile® process relies on handheld scanners to acquire 3D information, the reliability of which has not been verified. The other limitation, which not only applies to the SureSmile® process but also to other systems in computer-aided orthodontics treatment, is that the treatment planning is done without the consideration of actual root information. This is an important omission that limits their clinical reliability.

In this study, we present an orthodontics visualization system that is able to provide 3D visualization of complete teeth with only study models and the orthopantomogram as the data source. The ability to provide root information without the use of radiologically invasive computer tomogram (CT) scans is the obvious advantage as compared to earlier approaches.

Hassan *et al.* [34] proposed an approach to reconstruct 3D models of volumetric human jaws. The reconstruction procedure takes a model from a limited tooth library and deforms it with reference to multiple X-ray views of the tooth of interests. Compared to this approach, our approach has two advantages:

- We obtain the crown information obtained from the study model instead of using the deformed tooth; therefore the crown result is more reliable.
- Our method requires only one X-ray view (OPG).

Our orthodontics visualization system consists of the following components

- Crown preparation from study models.
- Root deformation using orthopantomogram (OPG).

- Merging of deformed root and actual crown.

Each of the above components is described in detail in the following sections.

4.2 Crown Preparation

Tooth (crown) segmentation from dental study models is an important step in many semi-automated computer vision systems that deal with individual crowns. Many crown-level analyses, such as the measurement of orthodontics parameters [24]-[26], simulation of tooth rearrangement for correction of malocclusion [27], and tooth pose estimation [28], are only applicable provided that the individual crowns have been extracted from the study models.

However, 3D crown segmentation has remained difficult in computer vision since teeth come in different shapes, sizes as well as poses. Early attempts on crown segmentation were mainly based on detecting tooth interstices (contact areas) along the U-shaped axis of study models. The problem with these attempts is that because they only make use of the plan-view range image of the study model, practical limitations such as small incisor interstices and the deep fissures on the occlusal surface of the molars caused the algorithms to fail.

4.2.1 Algorithm

We decide to investigate one of the latest attempts of accomplishing the task of crown segmentation and adopt it to our study.

An algorithm for tooth segmentation from study models was recently proposed by Kondo *et al.* [29]. The proposed algorithm consists of several steps. Firstly the dental arch, represented by a fourth-order polynomial, is detected by a two-step curve fitting procedure. A panoramic range image is then generated by using the detected dental arch as a reference surface. Finally, tooth interstices are detected based on both the plan-view range image and the panoramic range image. Experimental results show that this algorithm is highly accurate in the segmentation of dental models with various types of malocclusions. Figure 4.2 shows the segmentation result of a study model.

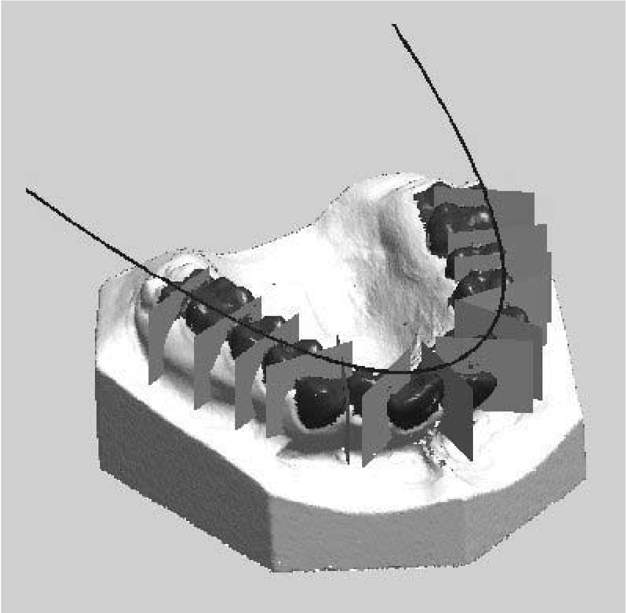


Figure 4.2: Crown segmentation result.

However, the existing implementation of the algorithm is not directly applicable to our study because of the following reasons:

- The original implementation of this algorithm takes approximately 60 minutes to complete a full segmentation of one study model on a 2.4GHz PC. Such a time consuming process is undesirable.
- The segmentation result is presented in terms of tooth interstitial planes as well as the outer (or buccal) gum margin curves, which in turn are detected from the panoramic range image by analyzing the surface discontinuities. However, the algorithm still lacks the ability of logically extracting individual tooth.

- The entire segmentation process is automatic without user interaction, which makes it difficult to fine-tune the segmentation result if necessary.

We have successfully adopted the algorithm into this study by enhancing it. In the following sections, we shall look at the way these limitations are resolved.

4.2.2 Detection of Tooth Interstices and Gum Margins

The original tooth segmentation process was fully implemented in Matlab. The high-level-language nature of Matlab makes it slow in dealing with highly computationally intensive tasks. As a result, the segmentation process takes a long time (nearly 1 hour), most of which is spent on calculating the panoramic image (Figure 4.3). Moreover, the original algorithm does not generate the inner gum margin line, which is essential in extracting individual teeth from the study model.

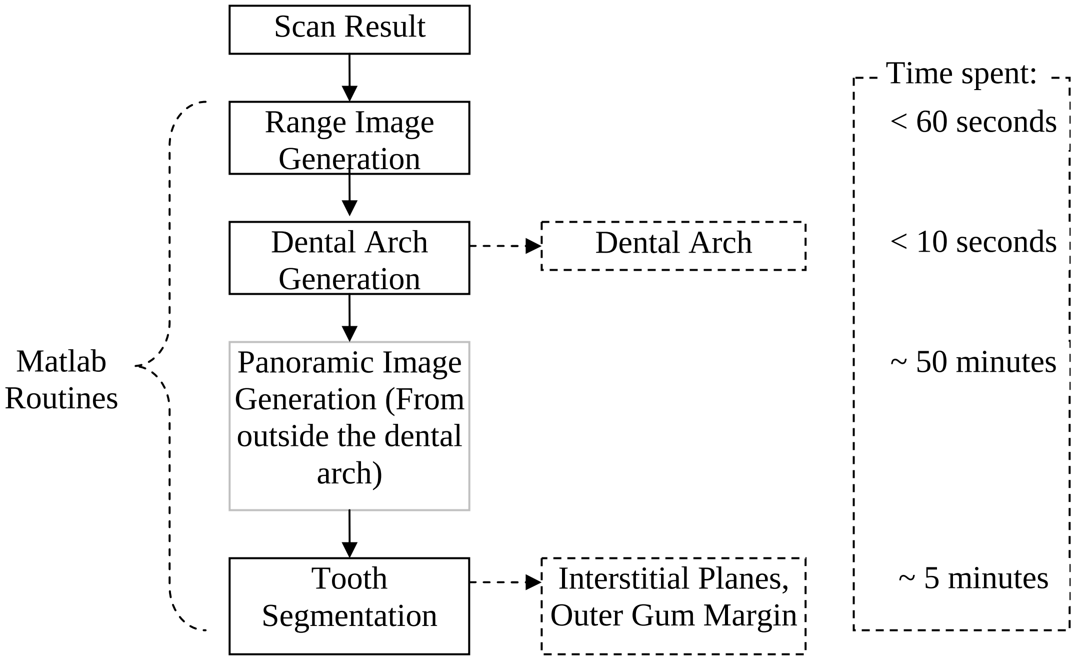


Figure 4.3: Flowchart of the original implementation.

We decide to migrate part of the implementation from Matlab to Java, a cross-platform programming language provided by Sun Microsystems. The well-developed compiler makes programs written in Java outperform Matlab by a considerable margin. Figure 4.4 shows the flow of the enhanced implementation of the algorithm.

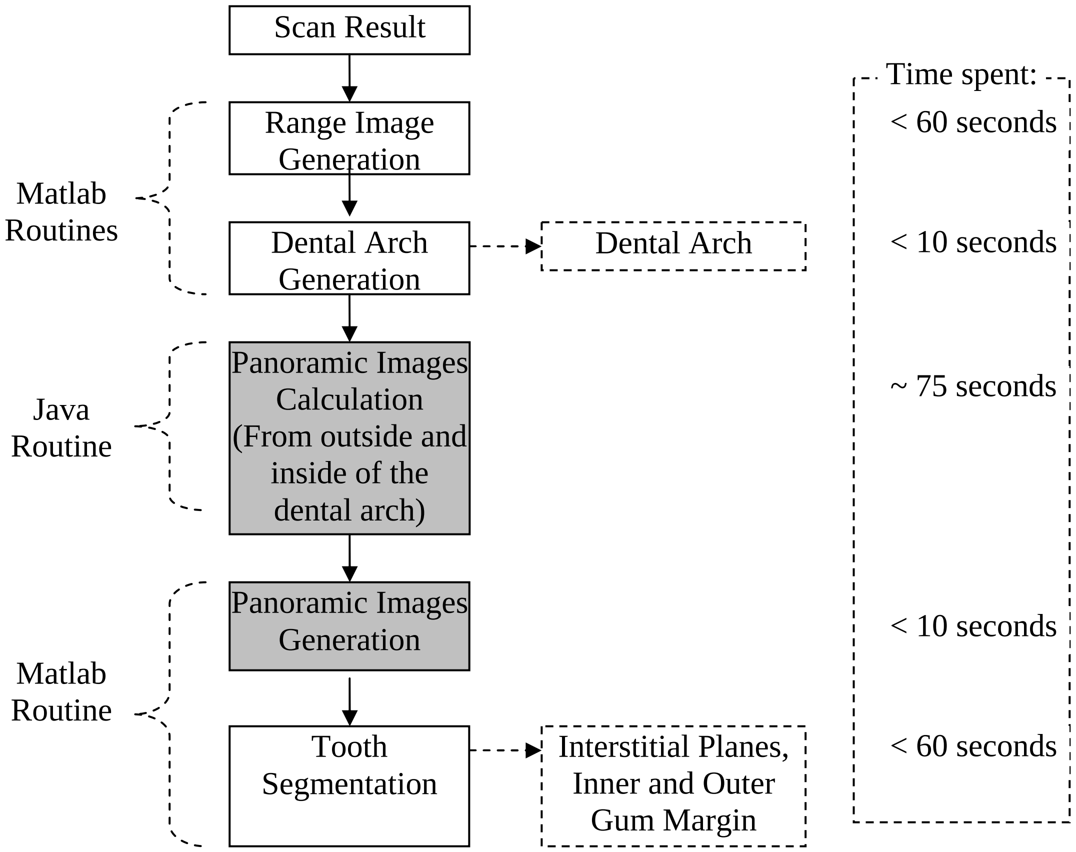


Figure 4.4: Flowchart of the enhanced implementation of the algorithm (changes highlighted).

Experimental results show that the speed performance of the enhanced implementation is much better. Under an identical hardware configuration (a 2.4GHz PC), the total time of generating panoramic images from the 3D object, which is the most time-consuming step in the original implementation, was reduced from 60 minutes to 85 seconds, i.e., a 4200% performance boost!

By generating two panoramic images (from outside and inside of the dental arch), we have also made the necessary modification in the enhanced implementation such that it has the ability to generate both the inner and outer gum margin line. The correspondence between the inner and outer gum margin points that form the gum margin lines can be easily established because these points are generated along the same sampling of the dental arch. This characteristic is particularly useful when we utilize the detection result to extract individual teeth as explained later.

Each of the detected tooth interstices is a rectangle in 3D space, which is represented and stored by its 4 vertices. The detected gum margin first appears as a series of discrete points in the 2D panoramic image (Figure 4.5). At the end of the detection, the correspondence of the panoramic image pixels and the surface of the study model in 3D space is determined and hence used to transform the gum margin from 2D to 3D space.

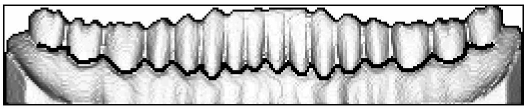


Figure 4.5: Detected outer gum margin on a panoramic Image.

Figure 4.6 shows the detected tooth interstices from a study model. By linking the corresponding inner and outer gum margin points with straight lines, we obtained a series of line segments that are used later in the tooth extraction process (Figure 4.7).

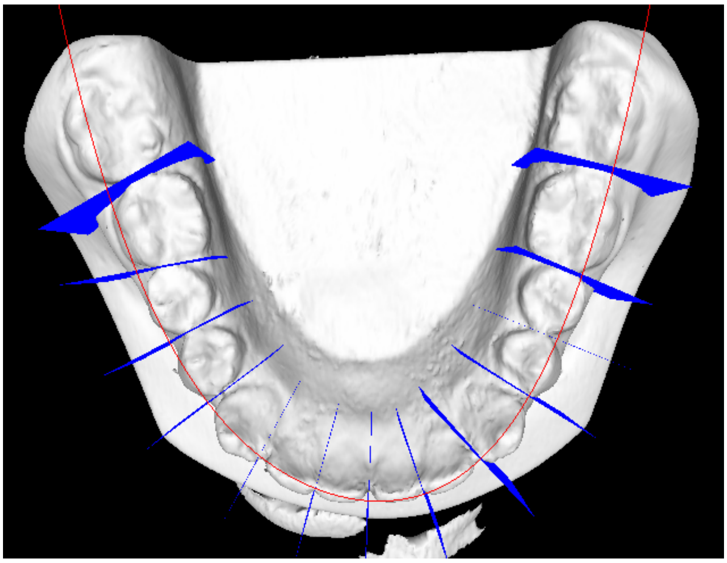


Figure 4.6: Detected tooth interstices.

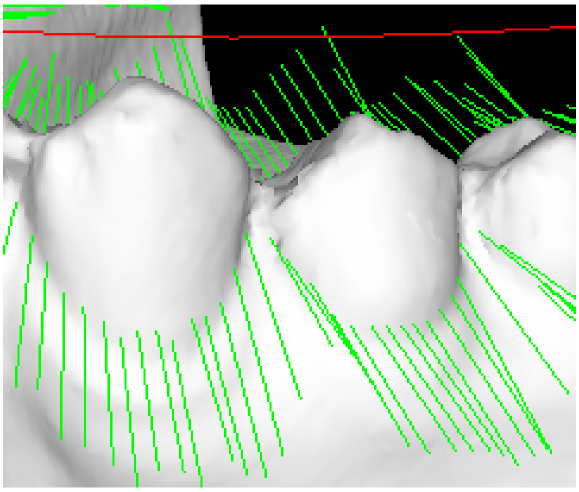


Figure 4.7: Detected outer gum margin.

4.2.3 Manual Adjustment of Tooth Interstices and Gum Margins

Although the algorithm for tooth interstice and gum margin detection gives satisfactory result in most cases, user interaction may still be desirable for fine tuning. Several software utilities were designed to facilitate such fine adjustments.

Adjustment of a tooth interstice is straightforward. Since each tooth interstice is defined as a rectangle completely described by its 4 vertices, we can control the position of a tooth interstice by moving its vertices. However, to preserve the shape of an interstice as a planar quadrilateral and to make the adjustment even easier, we decided to let the user adjust a tooth interstice by translating a pair of adjacent vertices together, i.e., translating an edge. Such an arrangement will allow the users to redefine the tooth interstices freely in 3D space, including translating and rotating the tooth interstices.

The adjustment of a tooth interstice proceeds in the following manner:

1. Select a tooth interstice from the interstice list.
2. Select an edge of interest from the edge list.

3. Translate the edge; three degrees of freedom are given, i.e., the edge can be translated freely in 3D space but not rotated.
4. Changes to the tooth interstice will be reflected.
5. Go back to Step 2 if further adjustment is needed.

Figure 4.8 shows a tooth interstice before and after adjustment.

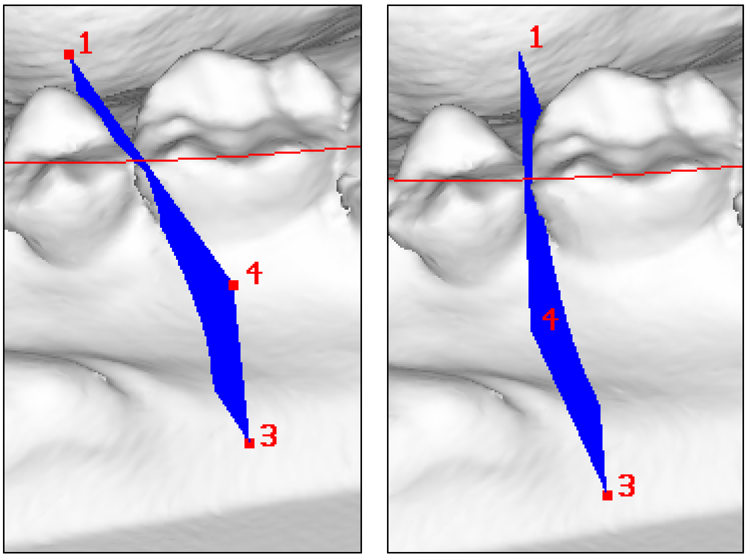


Figure 4.8: Manually adjusting a tooth Interstice.

The adjustment of the gum margins is more difficult to design due the following factors:

- A gum margin line might be poorly defined physically and hence difficult to detect in the image.
- The detected gum margins are represented by a large number of discrete 3D points. It is impractical for the user to adjust each of these points.

After experimenting with different options, we decide to implement the following:

- Performing the adjustment in a 2D plane, i.e., on the panoramic images. Due to lack of practical 3D pointing devices, it is difficult to position or even designate a point precisely in 3D. Performing the adjustment in 2D would greatly ease this effort.

- Making use of a general curve representation, the Bezier curve, to facilitate the adjustment of gum margin of a single tooth. Compared to adjusting each one of the gum margin point, we need only four points to represent the gum margin line for a tooth.

A brief description to Bezier curves [30] is given here. Consider four control points: p_0, p_1, p_2 and p_3. The most straightforward form of a curve represented by these four points is the cubic polynomial curve $p(u)$ that passes through them:

$$p_0 = p(0)$$
$$p_3 = p(1)$$

However, rather than using the last two control points, p_2 and p_3, for interpolation, we use them to approximate the tangents at $u = 0$ and $u = 1$ (Figure 4.9).

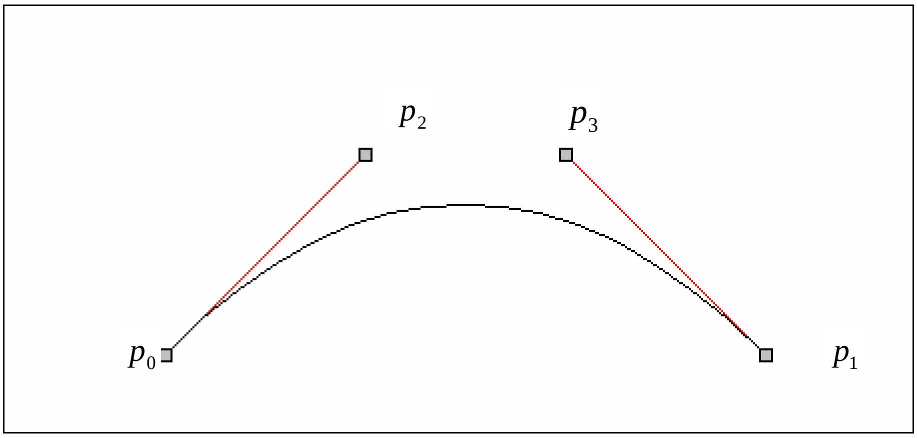


Figure 4.9: A Bezier curve

By using Bezier curves, we have greatly reduced the effort to perform necessary modification to the detected gum margin. Figure 4.10 shows effect of adjustment of gum margin on a panoramic image using a Bezier curve, where the red line represents the original gum margin line, blue points represent 4 control points, and the green line represents the new gum margin line. In this interface,

- Users can freely adjust any of the 4 control points, after which the curve would be automatically re-drawn
- Changes would be reflected to 3D space automatically

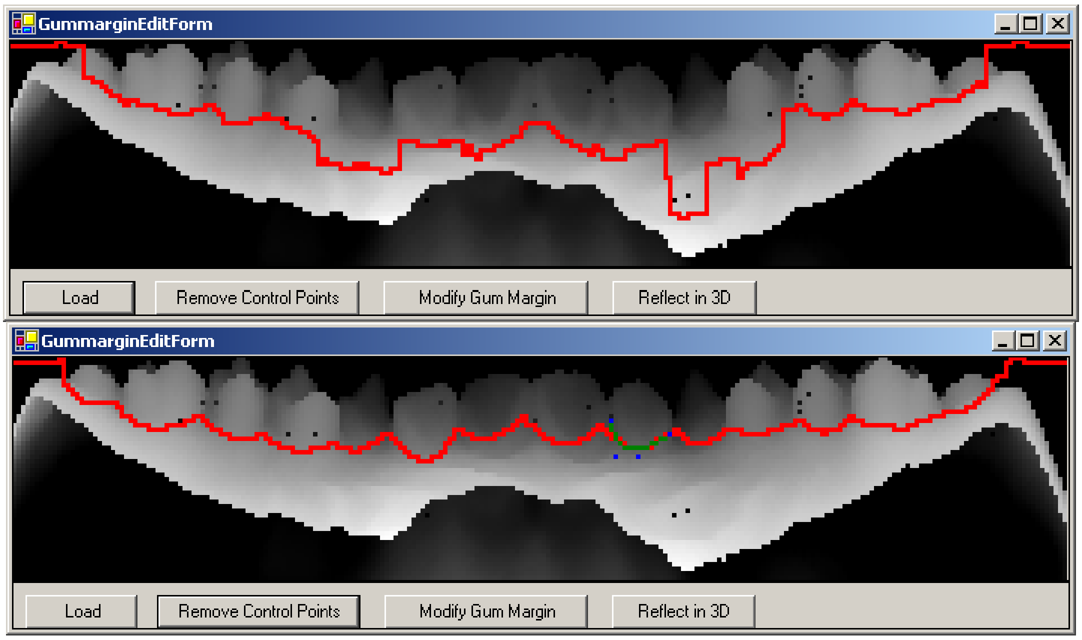
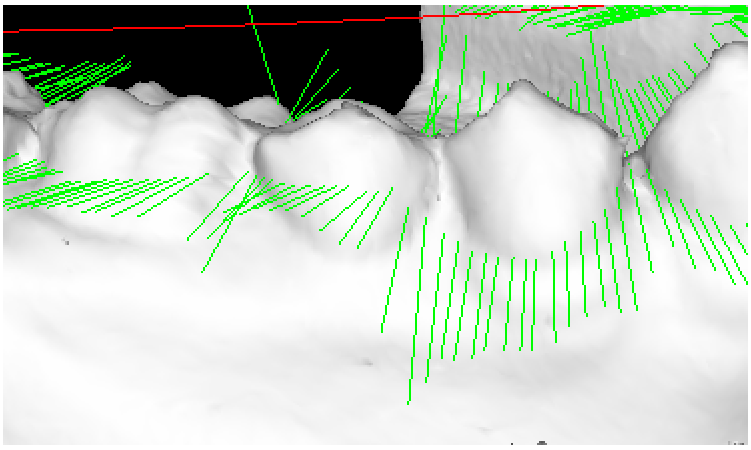


Figure 4.10: Gum margin adjustment using Bezier curve.

Figure 4.11 shows the effect in 3D of gum margin adjustment using Bezier curve.



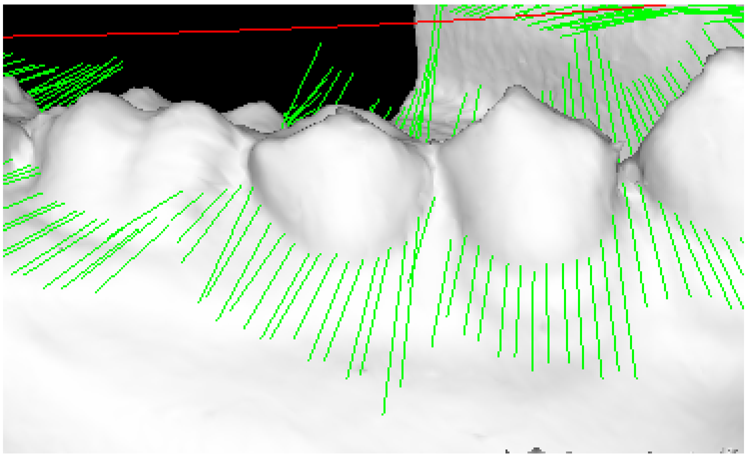


Figure 4.11: 3D effect of gum margin adjustment using Bezier curve

4.2.4 Extraction of Individual Tooth

Based on the segmentation result of a study model (in terms of tooth interstices and gum margins), each individual crown would then be logically extracted from a dental study model. This remains a difficult task due to the irregularity of the segmentation reference, in particular, the gum margin being represented by two series of discrete points corresponding to the inner and outer gum margins.

The digitized model is firstly oriented to a standard pose (Figure 4.12) such that the detected dental arch is parallel to the $x-y$ plane in the Cartesian coordinate system and the crowns are facing to the positive z direction (Refer to Figure 3.14 for axis definition)

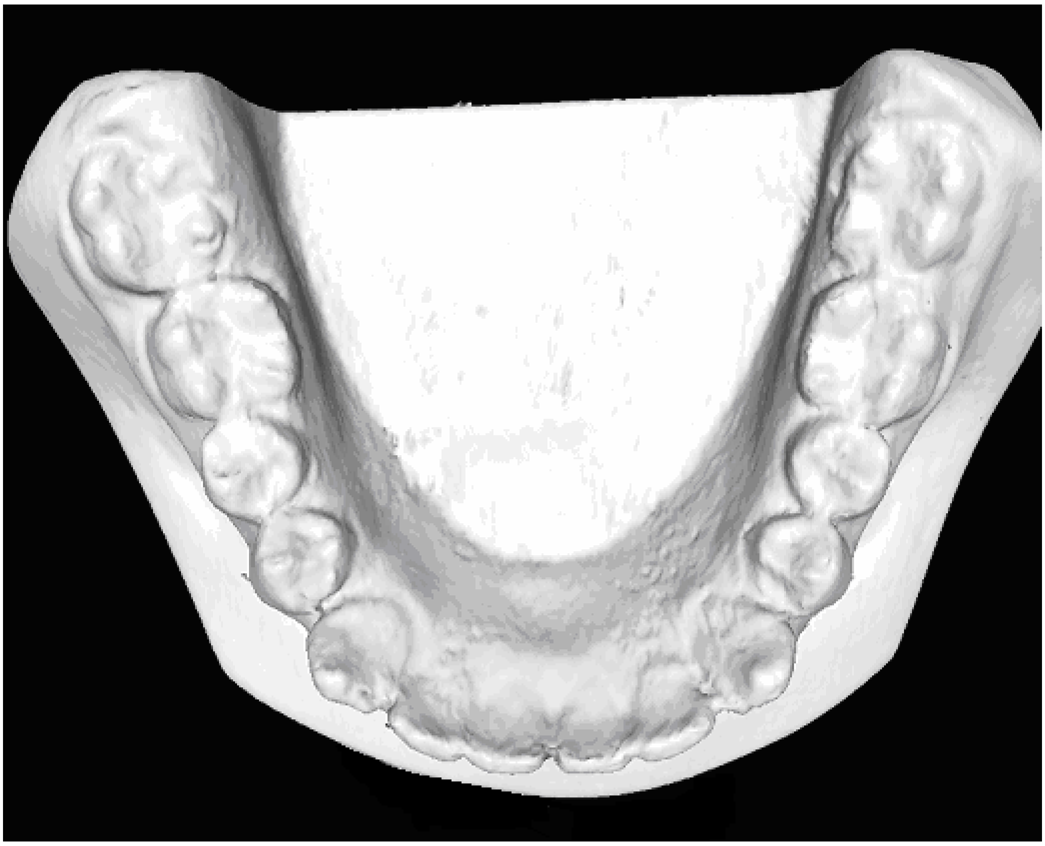


Figure 4.12: Study model before extracting tooth.

The algorithm of extracting individual crowns is a four-stage filtering algorithm:

- Stage 1: Vertices with z coordinates less than the minimum z coordinate of the gum margin curve are removed.
- Stage 2: Vertices below the gum margin curves are removed.
- Stage 3: Vertices are grouped to form individual teeth based on:
  - Distance to the dental arch
  - Relative locations with respect to interstitial planes
- Stage 4: Vertices not assigned in Stage 3 are removed.

Stage 1 is a pre-filtering process that reduces the amount of data to be processed. After removing these superfluous vertices, the face list is reconstructed by discarding faces referencing the unwanted vertices. Around 80% of vertices and faces are removed, which greatly improves the efficiency of the algorithm. Figure 4.13 shows a study model after Stage 1.

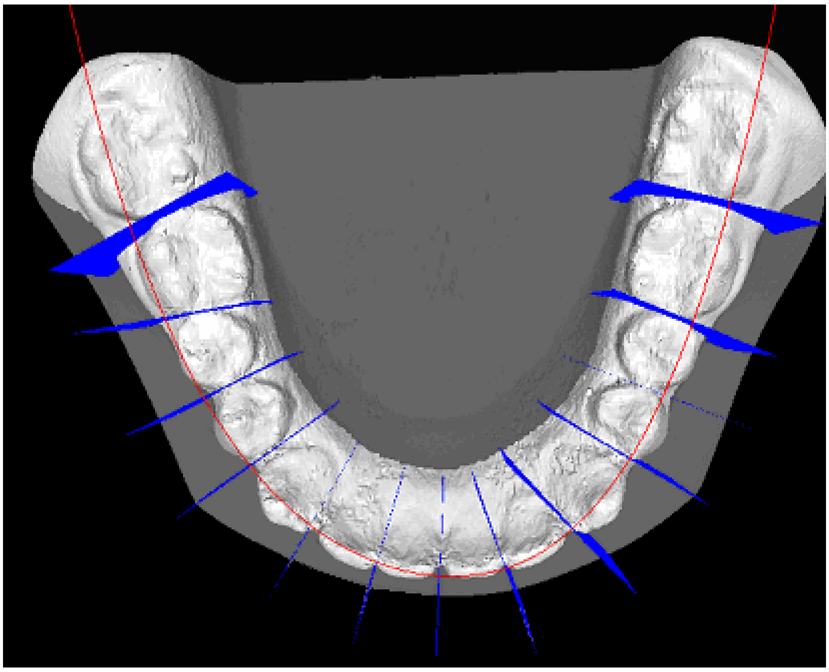


Figure 4.13: Result after Stage 1 (removed portion represented in dimmer parts).

The curved surface that separates the crowns and the rest of the model is incompletely described by two series of discrete points in 3D space, which were converted from the gum margin points detected from two panoramic images. It is not a trivial task to segment a 3D model under such circumstance. Stage 2 goes through the following steps to decide whether or not a vertex should be removed:

1. Resolve the correspondence of the inner and outer gum margin points.
2. Join the corresponding pairs of inner and outer gum margin points to form line segments - gum margin lines.
3. Project all gum margin segments to plane $z = 0$.
4. Pick a vertex v from the vertex list.
5. Let v' be the projection of v on plane $z = 0$.
6. Determine the (projected) gum margin line G', which is the closest to v on the projection.

7. Determine whether v should be discarded by comparing its z coordinate with a threshold value:
    - If v' lies on G', the threshold value is determined by linearly interpolating the z coordinates of both end points of G.
    - If v' does not lie on G', the threshold value takes the z coordinate of the ending point of G which is closer to v.
8. Repeat Step 4 and Step 7 until the entire vertex list is traversed.

Figure 4.14 shows the cross-section geometric relationship used in Step 7.

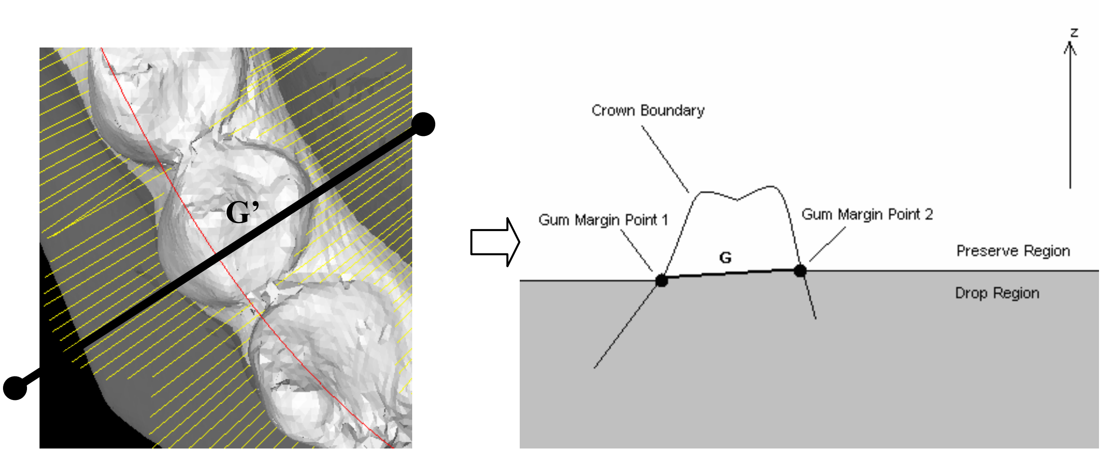


Figure 4.14: Cross section geometric in tooth extraction Stage 2.

Experimental result shows that around 50% percent of the remaining after Stage 1. The result is shown in Figure 4.15.

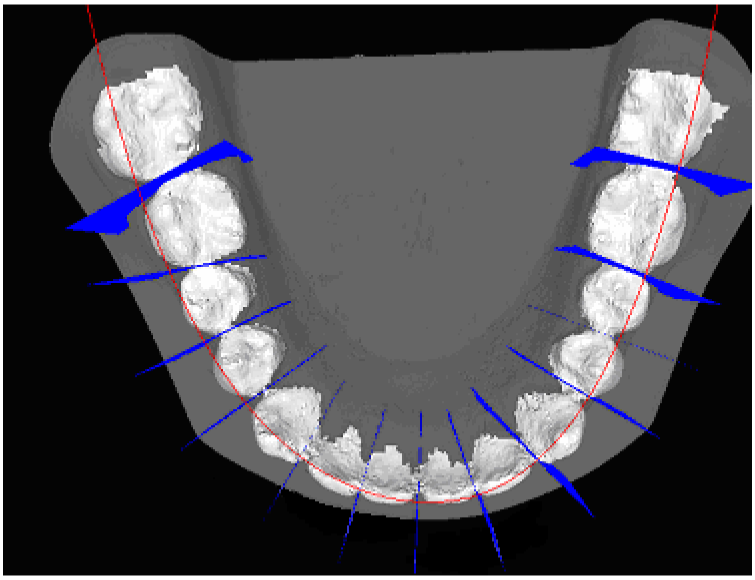


Figure 4.15: Result after Stage 2 (removed portion represented in dimmer parts).

After performing Stage 2, the majority of the remaining vertices belong to the crowns. In Stage 3, these vertices are grouped together to form individual crowns, which is done by considering the following features:

- Distance to the dental arch.
- Relative location with respect to the interstitial planes.

The algorithm first tries to extract the first tooth and the last one (counted along the dental arch, from left to right), the result of which is used to extract the other teeth. The detected dental arch plays a crucial role here by providing a reliable estimate of tooth position with respect to the interstitial planes.

The first tooth is extracted in the following manner:

1. The intersection point P of the dental arch and the first interstitial plane is determined.
2. The dental arch point Q which appears 5 samples before P is taken as the representative point of the first tooth, i.e., the vertices that consist of the first tooth should be on the same side as Q is with respect to the first interstitial plane (Figure 4.16).

3. For any given vertex v from the vertex list, v is considered to be part of the first tooth if these conditions are satisfied:
    - The minimum distance between v and all dental arch sampling points before P is less than 20mm.
    - v stays on the same side of the first interstitial plane as Q .
4. Repeat Step 3 until the entire vertex list is traversed

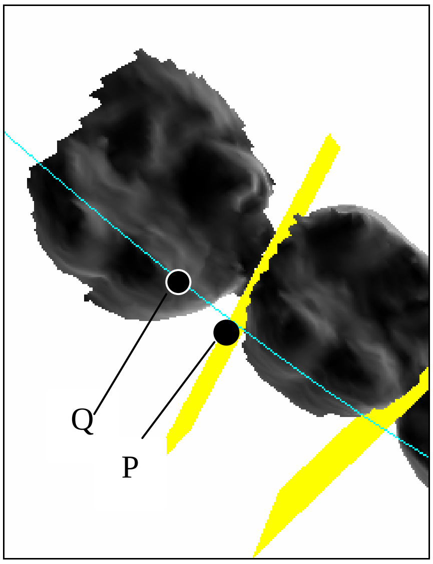


Figure 4.16: Choosing of reference point to extract the first tooth.

The same idea is repeated in extracting the last tooth. The rest of the teeth are extracted in a similar manner. The only difference is that the extraction considers two interstitial planes instead of one. Figure 4.17 shows the result after Stage 3.

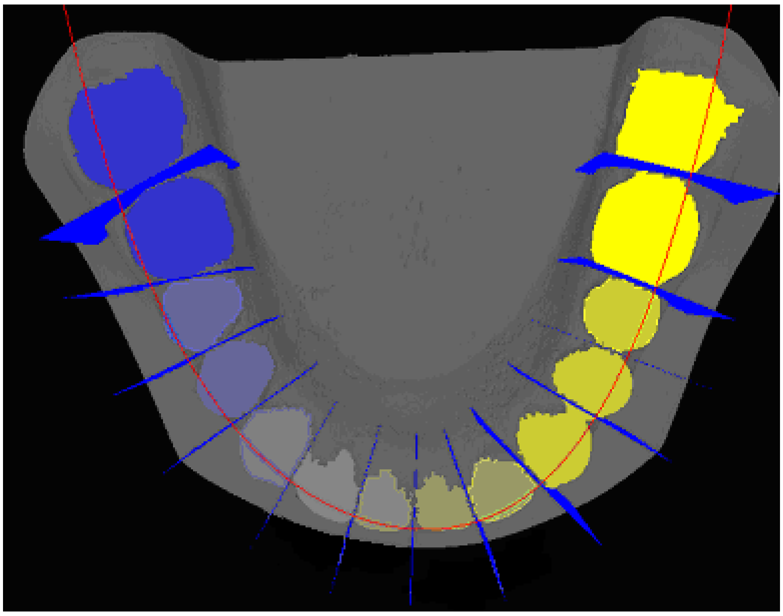


Figure 4.17: Result after Stage 3
(Removed portion represented in dimmer parts; different teeth are shown in different colors).

Stage 4 removes any ungrouped vertices in Stage 3, which is an essential step in some models that has a lump at the center (non-tooth part). This is done by traversing through the remaining vertex list and removing those vertices that are not assigned to any crown.

4.2.5 Results

Experimental results show that the crown segmentation and extraction algorithm generally work as expected. The implemented algorithm successfully exacts individual crowns from every test model. However, the extraction quality is still dependant on the model quality, especially the definition of the gum margin lines.

Figure 4.18 shows the crown extraction result on a high quality model, where both the inner and outer gum margin lines are well defined. The extraction result is highly satisfactory in such cases.

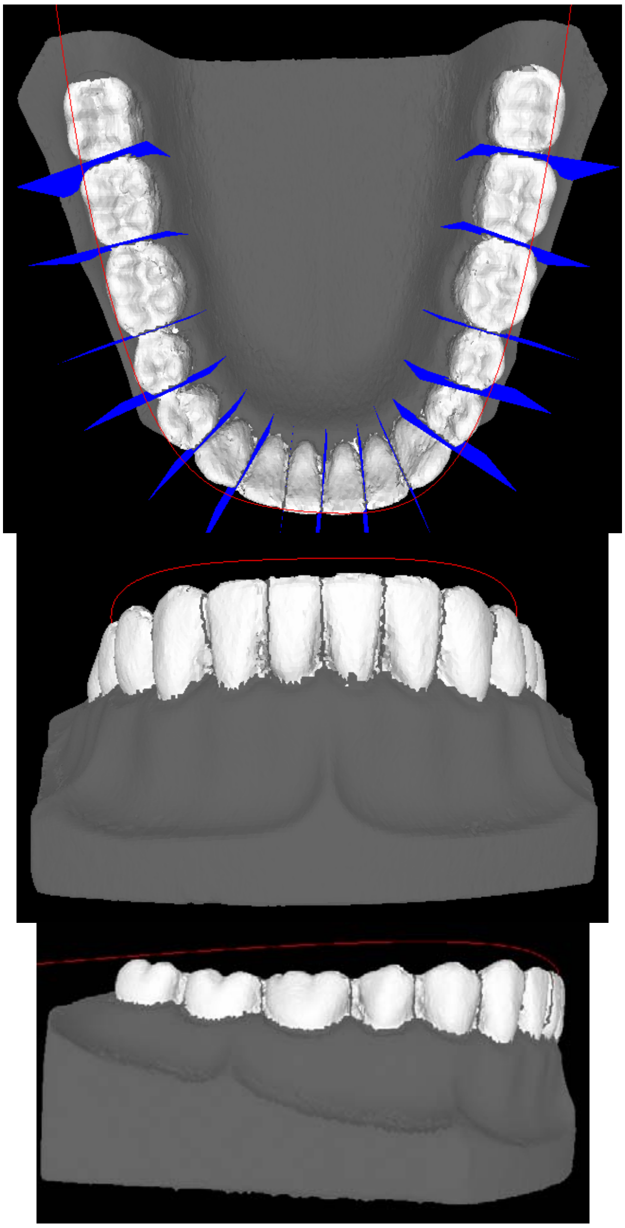


Figure 4.18: Crown segmentation result 1.

Figure 4.19 shows the crown extraction result of a lower quality model, where part of the gum margin lines are hardly distinguishable to the human eye. Without manual adjustment on the gum margin lines, the extraction result is unacceptable. Figure 4.20 shows the detection result of the same model after manual adjustment on the gum margin lines. The detection result is greatly improved except for the presence of an unexpected line fragment at the left first incisor, which is caused by the information lost during panoramic image generation due to poor model quality.

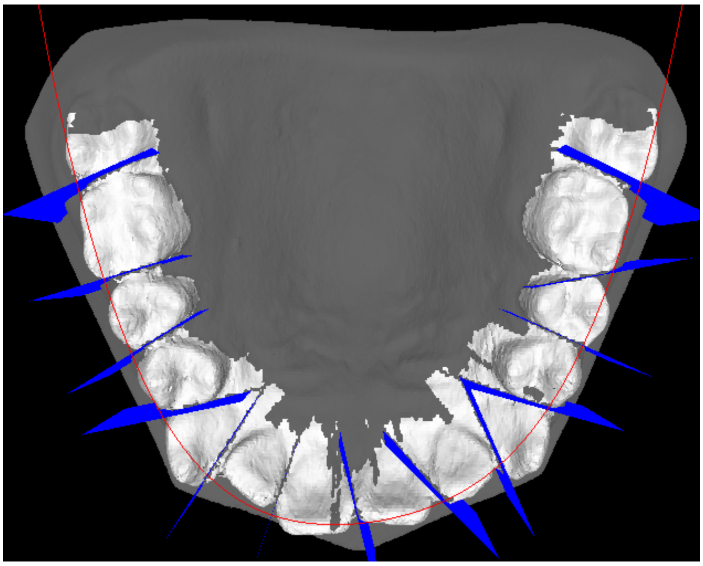


Figure 4.19: Crown segmentation result 2.

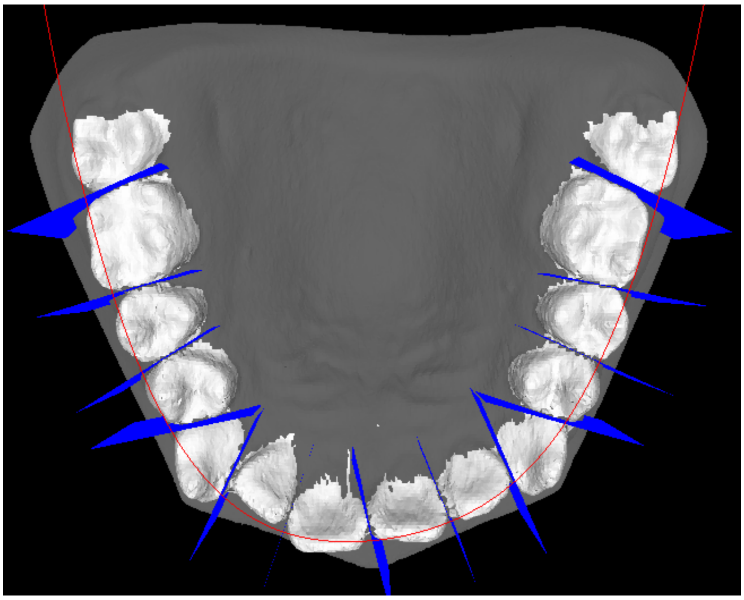


Figure 4.20: Crown segmentation result 3.

In general, the crown segmentation and extraction algorithm gives satisfactory result. Manual adjustment on tooth interstices and gum margin lines is an effective tool in handling minor model problems such as poorly defined gum margin lines. However, unsatisfactory results occur when the model quality is extremely poor which leads to an incomplete panoramic image during segmentation. Figure 4.21 shows such a case, where one of the incisors is poorly extracted due to the incomplete panoramic image.

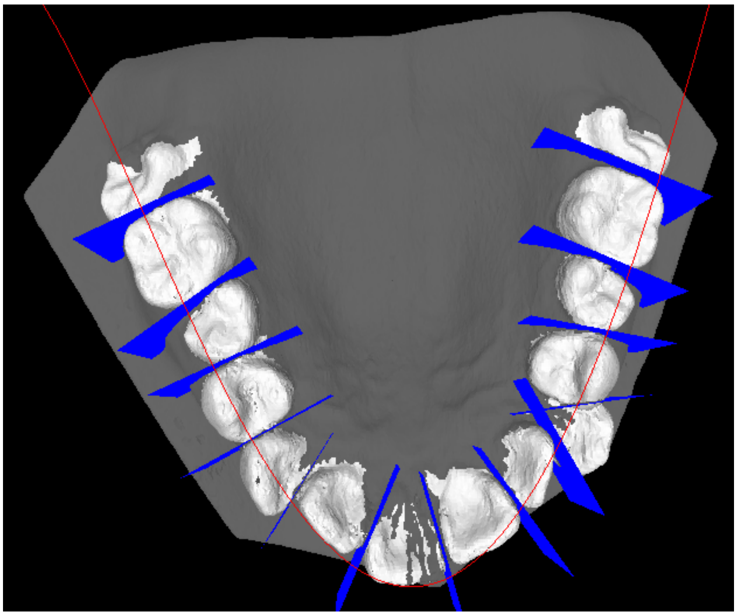


Figure 4.21: Unsatisfactory extraction result

4.3 Root Deformation

A dental study model is an imprint of dentition that is taken directly from patients. Therefore it provides only crown information. However, to simulate and decide on an orthodontic treatment plan, it is necessary for orthodontists to gather information of the entire dentition (both crowns and roots) of a particular patient.

There are two types of medical imaging techniques that are capable of showing root information, namely the orthopantomogram (OPG) and computerized tomography (CT). Compared to the OPG, CT scanning is able to give accurate cross-section information of teeth. However, the standard fan-beam CT technique is not a routine procedure for most patients as it produces substantially more radiation (as compared to the OPG), and is still a very expensive procedure. The OPG, although cheap, is only capable of presenting 2D data, which limits its usefulness in orthodontics analysis. Being a tomography technique, OPG also provides better resolution as compared to CT.

4.3.1 Algorithm

As far as this study is concerned, we are interested in presenting 3D roots using only dental study models and the OPG. This objective is achieved by deforming a set of 3D generic tooth models with the actual OPG of specific patients. The prototype algorithm, proposed by Enciso and Lewis [31], makes use of “a patient-specific radiograph and a 3D geometric prior model to produce a ‘best-fit’ patient specific 3D model of the whole tooth using radial basis functions.” (Figure 4.22).

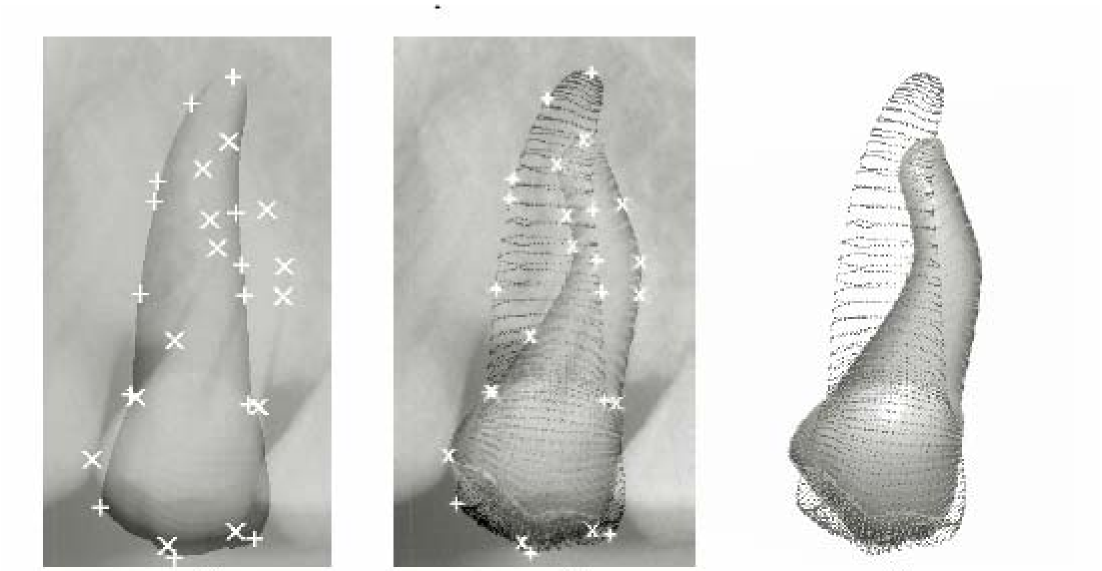


Figure 4.22: Tooth deformation using thin-plate spline transformation.

The user has to indicate two sets of matching landmarks from both the 2D and 3D models. The algorithm computes the thin-plate coefficients from these landmarks for the *x* and (separately) *y* displacements and then warps the 2D image and the 3D mesh (Figure 2c) in the *x* and *y* directions (with depth information kept constant) using the thin-plate interpolant. Thin-plate splines minimize a bending energy:

$$\int\int (f_{xx}^2 + 2 f_{xy}^2 + f_{yy}^2) dxdy$$

while interpolating values of a set of 2D points named landmarks. The thin-plate spline can be represented and solved using radial basis functions:

$$f(x,y) = a_1 + a_x x + a_y y + \sum w_i \bullet U(|(x_i - y_i) - (x,y)|)$$

with

$$U(r) = r^2 \ln(r)$$

The application of the algorithm was to generate "best-fit" 3D presentation of the entire tooth with the OPG only. However, we extend the application to a higher level by performing segmentation on a deformed tooth and extract the 3D root only, which will be then merged with the corresponding 3D crown obtained from the study model. The merging of the deformed root and actual crown is covered in the next section.

4.3.2 Implementation

The deforming procedure is essentially a manual one. A 2D radiograph (in this case the OPG) and a 3D generic tooth model are presented by the visualization system. The user first needs to orient the 3D model to a standard position such that from the view point is approximately the same as the position of the camera where the OPG was taken (Figure 4.23).

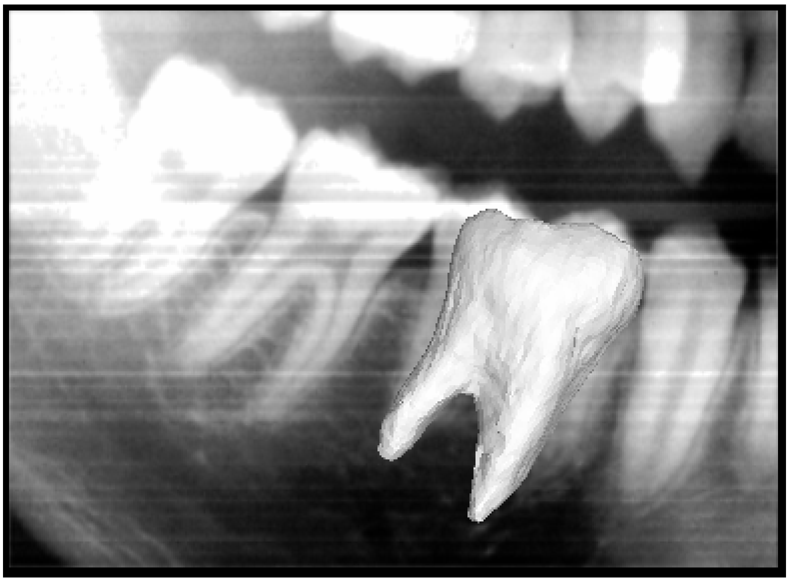


Figure 4.23: Tooth deformation – rotation and translation.

The user is asked to interactively indicate corresponding landmarks of their choice on both the OPG and the view of the 3D model (Figure 4.24). In order to achieve a unique solution to the thin-spline interpolation, a minimum of 14 landmarks (both on the OPG and on the 3D model) are required.

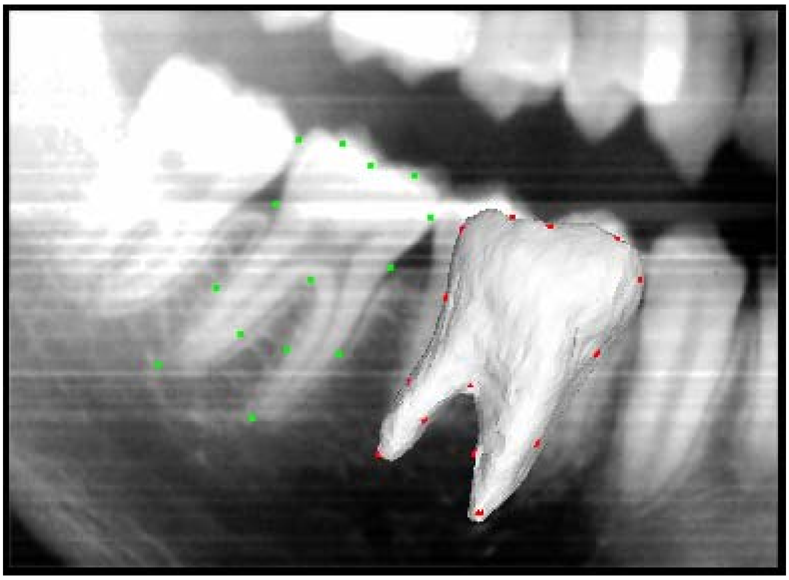


Figure 4.24: Tooth deformation – selection of landmarks.

From the selected sets of landmarks, the system calculates the transformation matrix automatically by solving the thin-plate spline interpolation to minimize the bending energy. The transformation matrix is then applied to the entire 3D model, the result of which is shown as the transformed 3D model (Figure 4.25). It should be noted that the thin-plate spline is the 2D analog of the cubic spline in 1D. The applied transformation matrix does not cause any change to the depth information of the 3D model, which is rational since the 2D OPG does not provide any depth information.

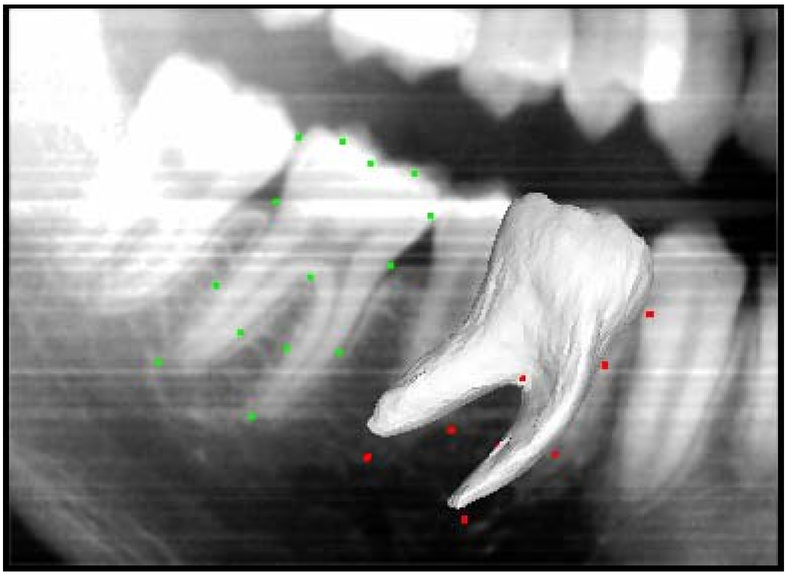


Figure 4.25 Tooth deformation – deformed model.

The user’s expertise plays a major role in this deforming process. Both repositioning (rotating the 3D model to a correct pose) and picking points require orthodontic expertise in order to achieve a good deforming result. It should be also noted that this deforming process comes with limitations. We are trying to deform 3D objects with their 2D projection (the OPG), which does not give complete information about the shape. For example, an OPG is not able to provide information about the tilti angle of the root around the dental arch (Figure 4.26)

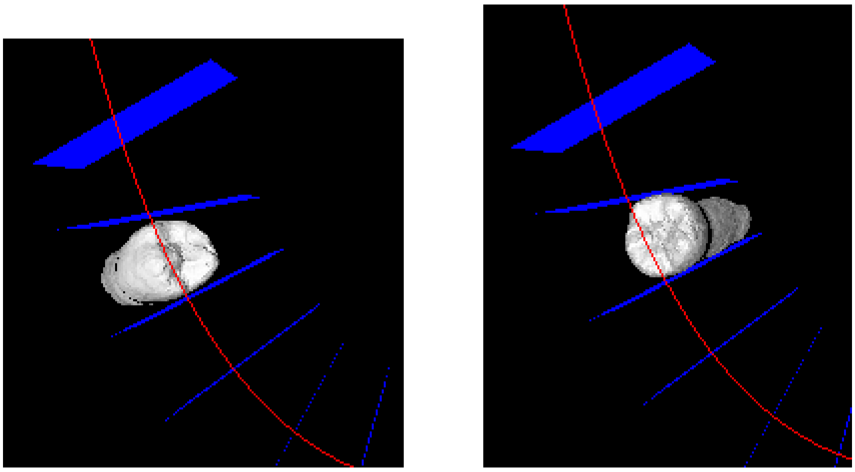
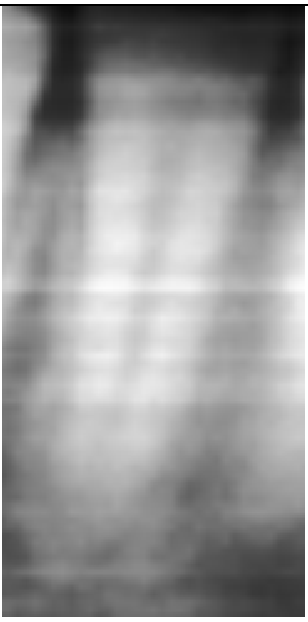
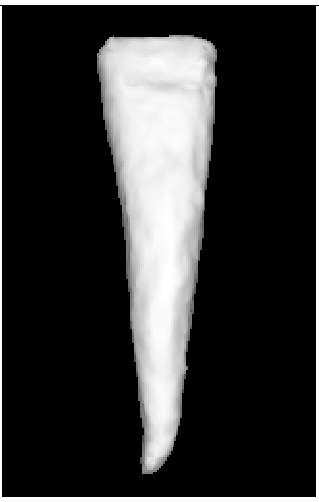
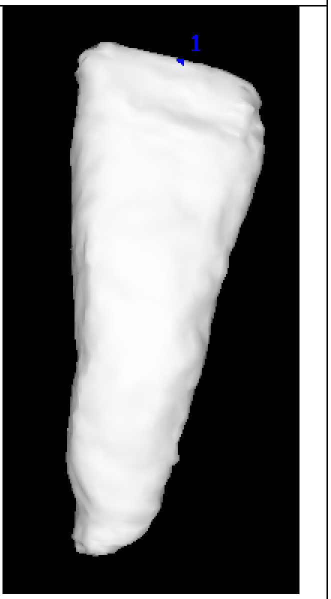
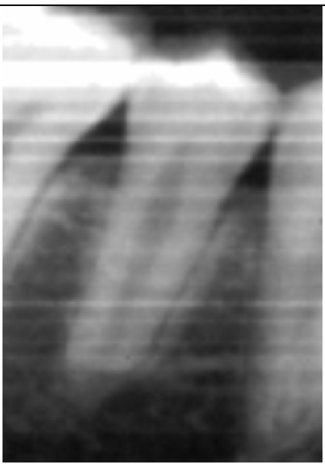
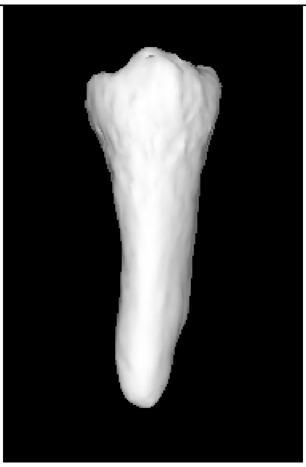
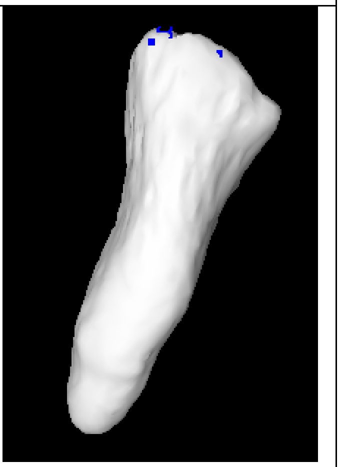
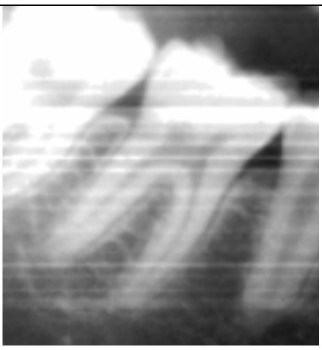
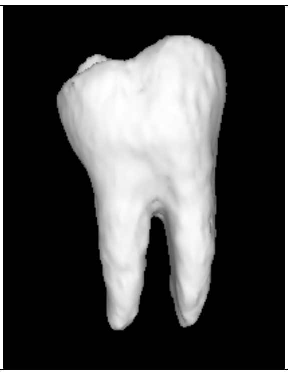
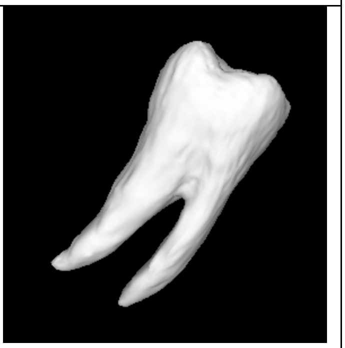


Figure 4.26: Tilting of root around the dental arch.

4.3.3 Results

Experimental results of the root deformation procedure are presented in Table 4.2.

Table 4.2: Tooth deformation results.

Model Type	OPG	Before Deformation	After Deformation
Incisor			
Pre-Molar			
Molar			

In conclusion, we have the following comments:

- The deformation procedure successfully deformed every 3D tooth model according to a patient's orthopantomogram and generated a 3D estimate of the patient's tooth.
- The transformation parameters are calculated based on a number of manually selected feature points on both the 3D tooth and the orthopantomogram. Therefore accurately selecting these feature points plays a major role in the deformation procedure and hence has a direct impact on the result obtained. Poorly selected feature points might result in unsatisfactory deformation result, such as lumps on the deformed model.

4.4 Merging of Deformed Root and Actual Crown

As described previously, we have two different kinds of 3D representation of the teeth, namely the actual segmented crowns from the digitized study model and generic tooth models deformed with the OPG. The former representation, being an imprint directly taken from patients' dentition, is of high accuracy and reliability as compared to the latter, which is a 3D estimate of the tooth shape with its 2D projection as the reference. However, the first representation has only the crown information available whereas the second is a complete representation of entire teeth.

Our objective here is to extract the root information from the deformed teeth and merge it with the actual crown (Figure 4.27) such that we have a 3D representation that is both complete and accurate.

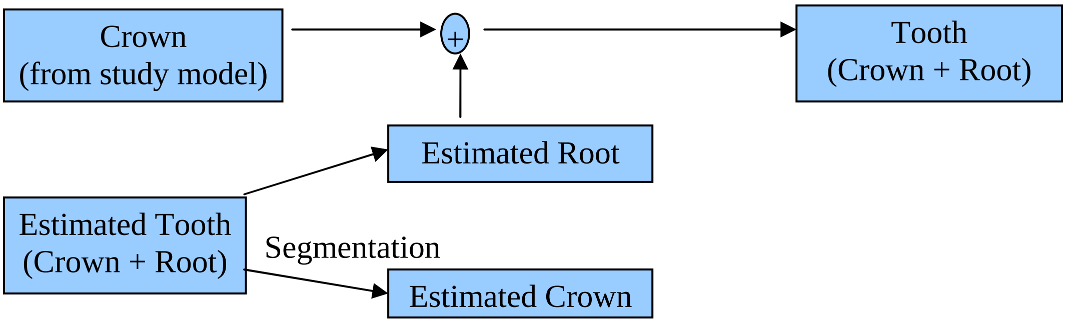


Figure 4.27: Accurate 3D tooth shape.

4.4.1 Procedures

The merging of deformed roots and actual crowns is done in several steps:

1. Repositioning of individual deformed teeth to a standard pose using the longest-axis algorithm.
2. Segmenting individual teeth into crowns and roots.
3. Logically combining individual roots and the corresponding actual crowns into teeth.
4. Smoothing the merged teeth with multi-quadratic radial basis functions to close the gaps between crowns and roots (optional).

4.4.1.1 Repositioning

The deformed teeth must first be positioned to the standard orientation before segmentation can be done. In the standard orientation, a tooth should have its root pointing in the negative z direction. The orientation along the z axis is the one such that the derivative of the dental arch at the origin is along the x axis.

The repositioning is achieved by the longest-axis. The axis of the deformed tooth is located by the following manner:

1. Calculate the centroid of the deformed tooth by averaging all vertices present in the model.

2. Locate the vertex in the model that is furthest to the centroid. Denote the distance between them as d.
3. Locate the vertices whose distance to the centroid is at least $0.9 * d$.
4. Compute the average location of the vertices found in Step 3.
5. The axis of the deformed tooth is obtained by joining the centroid and the vertex calculated in Step 4.

The rationality of the above algorithm is that, since the crown part of the tooth is physically larger than the root and consists of more vertices, the centroid of the tooth is nearer to the crown than the root. Therefore, the root tips of the tooth will be furthest to the centroid. By setting a threshold and averaging vertices that are furtherest to the centroid (the root tips), the axis can be located correctly.

Figure 4.28 shows a second molar in its standard orientation.

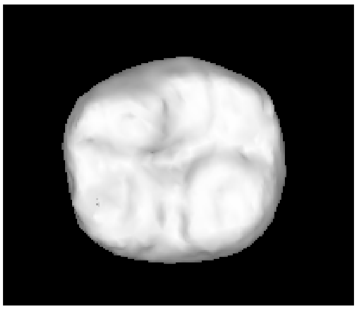


Figure 4.28: A second molar in its standard orientation

After rotating the deformed tooth to its standard position, the tooth must be further rotated by a tilting angle θ about the z axis such that it has the same orientation as the original crown. The value of θ is determined when tooth segmentation was performed. Figure 4.29 illustrates the above concept by taking a canine as an example.

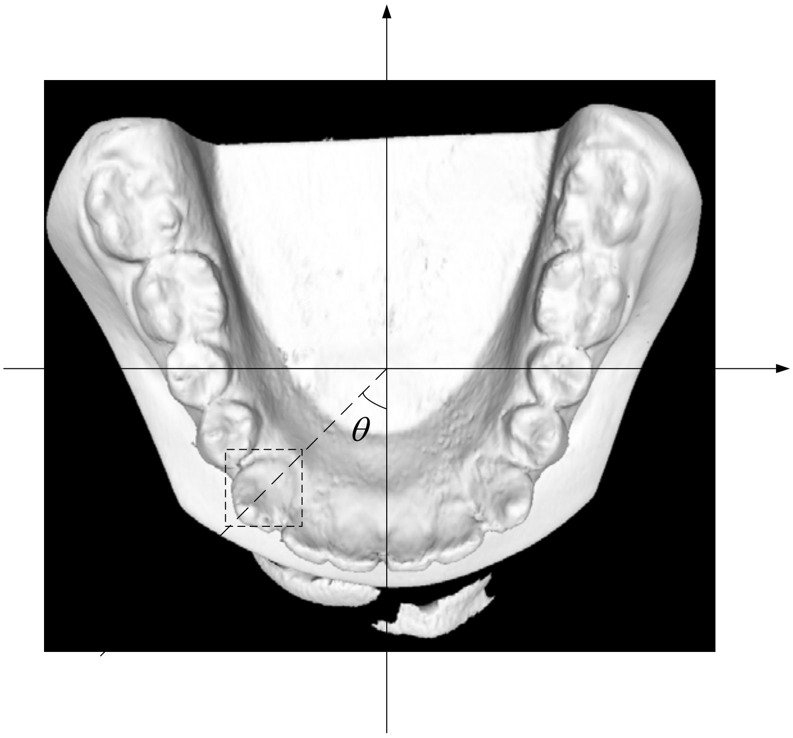


Figure 4.29: The tilting angle θ of a canine

4.4.1.2 Segmentation

The segmentation of a tooth into the crown and root is a difficult problem as the gum margin is not of regular shape. In this study, the segmentation is approximated by slicing a tooth (in standard pose) with a plane parallel to the $x-y$ plane, such that the crown portion has approximately the same size in the z direction as the actual crown. In order to reduce the impact of the irregular edge of the actual crown (caused by segmentation with the detected gum margin), instead of using the maximum occupied distance along the z direction as the slicing criteria, we use the projected distance (along the z direction) between the centroid to the highest vertex (the one with largest z coordinate) slicing criteria. The actual implementation makes use of a binary search procedure to speed up the search for the optimum slicing plane.

4.4.1.3 Merging

The deformed teeth are positioned on the digitized study model once segmentation is done. The deformed teeth are first rotated by 90 degrees along the x direction to suit the dental arch (of the study model), which is parallel to $x-y$ plane. Individual teeth are translated to the position of the corresponding crowns such that the centroids of the segmented crowns match the centroids of the actual crowns. The desired orientation angles of the deformed teeth along the z direction are determined using the derivative of the detected dental arch. Finally, the crown portions of the deformed teeth are removed leaving only the roots, which will be logically combined with actual crowns to form complete teeth.

4.4.1.4 Smoothing

The difference in the nature of data sources results in an undesirable characteristic of the teeth generated from the merging process. The crown and the root of a particular merged tooth are two discrete 3D meshes with no common vertices or faces in most cases. A gap between the crown and the root is clearly visible on the surface of a merged tooth.

In order to improve the display quality, we developed an algorithm to close such gaps. However, this process results in slight changes to the shapes of actual crowns, which is an unnecessary degradation to the reliability of crown shapes. Therefore it should be considered as an optional .

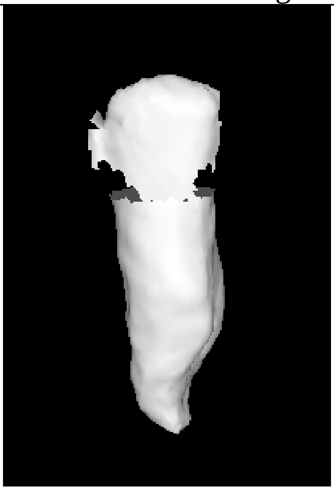
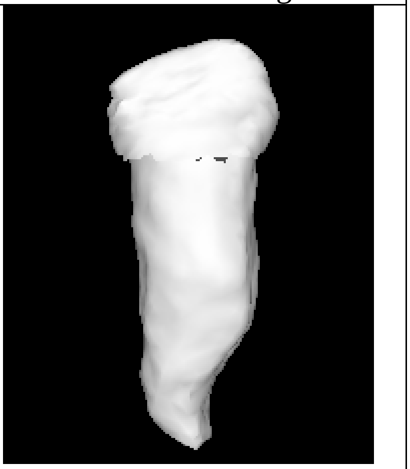
Davis et al. [32] studied various alternatives in the gap-filling algorithm. However, in our application the gaps between crowns and roots are not complex enough to warrant such a mathematically expensive process. Instead the gap-filling process is approximated using a multi-quadratic radial basis function with the deformed crowns (segmented from deformed teeth) as the base geometry.

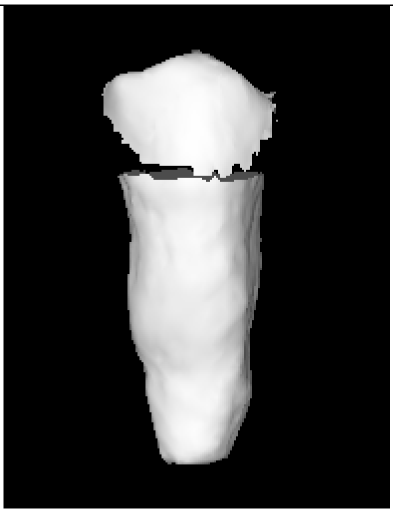
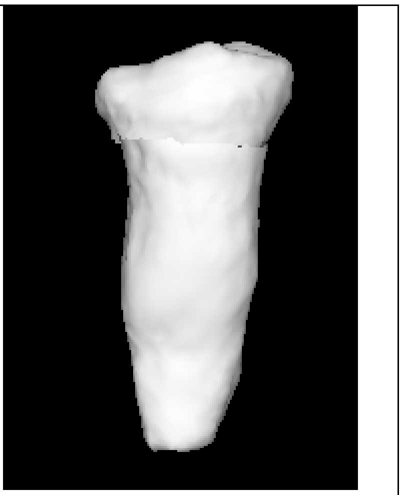
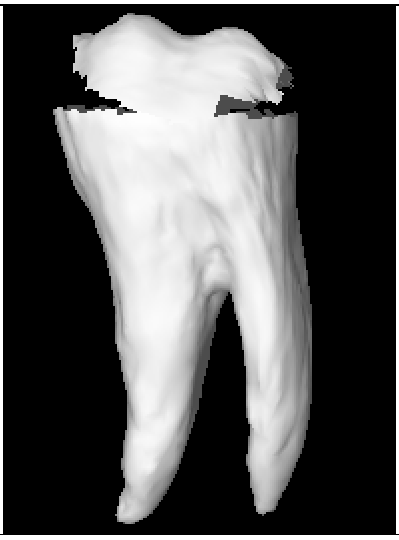
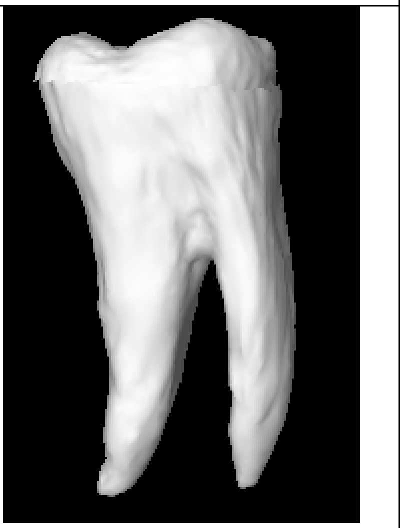
After applying the gap-filling algorithm, it is observed that the gap between the crown and the root is closed, resulting in a smooth and complete tooth. However, the change in the crown shape is also observed. This change, although slight, brings in some degradation to the accuracy of the system.

4.4.2 Results

Table 4.3 shows the re-generated 3D teeth by merging deformed roots and actual crowns.

Table 4.3: Results of merging deformed roots and actual crowns.

Model Type	Re-generated 3D tooth	
	Without smoothing	With smoothing
Incisor		

Pre-molar		
Molar		

We can draw the following conclusion:

- The repositioning algorithm gives satisfactory results. All tooth models, both single-rooted and multiple-rooted, are successfully rotated to their desirable positions.
- The segmentation algorithm successfully segments all tooth models into crowns and roots. However, a gap between the merged models is inevitable since the boundary of the actual crown model, extracted from a study model, does not necessarily meet the boundary of the segmented root, which is sliced using a plane.

- The proposed smoothing (gap-closing) algorithm works best for molar models in terms of closing the gap and minimizing the distortion to the actual crowns. For incisors and pre-molars, a mismatch between the model boundaries is still visible, and the shape of the actual crown is significantly deformed. However, even if a more sophisticated gap-closing algorithm is studied and developed, distortion to the actual crown model is inevitable. In order to preserve the model reliability as much as possible, we suggest that the smoothing operation should only be treated as an optional step to improve the visualization effects.

4.4.3 Limitations

After the merging of the actual crown and deformed root, the result is greatly dependant on the reliability of the deformed root. A perfect merging result is only achievable provided that the shapes of the crown and the root match well, i.e., both parts are from the same physical tooth. However, this could not be guaranteed as we are indeed merging a highly reliable crown model (from the study model) and a less reliable root model (from deformation with reference to the OPG). The 2D nature of the OPG makes the deformation result less reliable, especially along the depth direction, and hence reduces the reliability of merging.

In terms of appearance, the deformed root does have high similarities to the OPG, and so does the merged tooth. However, we cannot draw a conclusion as to whether the change in crown shape is clinically irrelevant until the procedure is applied in real orthodontics environment. A careful clinical study is required before the practical application of the suggested procedure.

Chapter 5

Occlusion Analysis

5.1 Overview

In orthodontics, occlusion is defined as the contact of the dentures of both arches resulting from the action of the muscles. The detection and correction of malocclusions and other dental abnormalities is a significant area of work in orthodontic diagnosis and treatment.

The most popular way of assessing malocclusions in orthodontics is to use a pair of upper and lower study models. Being an accurate 3D imprint of a patient's dentition, a pair of study models can precisely reproduce the occlusion status of the patient. Figure 5.1 shows study models of a well aligned case, while a malocclusion case is shown in Figure 5.2.

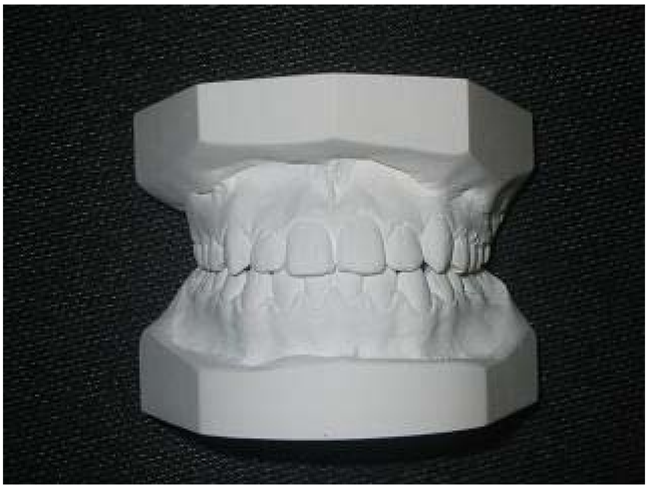


Figure 5.1: A well aligned case

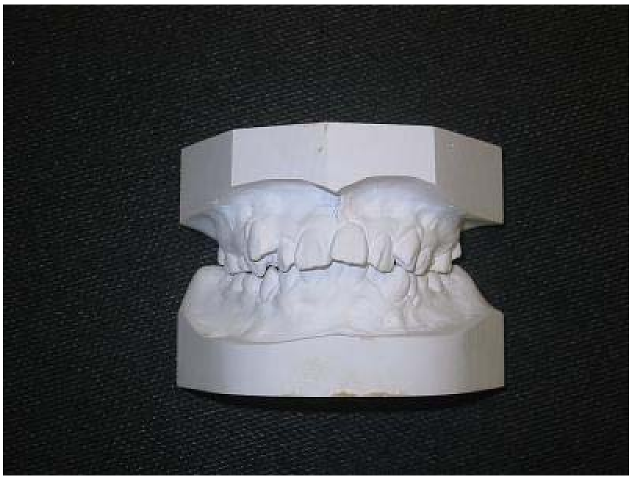


Figure 5.2: A malocclusion case.

However, as the imprint is taken separately for lower and upper halves of the dentition, a pair of study models by themselves does not provide any information about how these two models should bite.

In orthodontics, the occlusion information is assessed in a procedure named “bite registration”, where the patient is required to bite on a piece of softened wax, which precisely records the relative biting position of the patient’s dentition. By using the wax as a guide, a pair of study models is carefully positioned such that they bite exactly as the patient’s dentition does, after which the pair of study models are trimmed using a polishing machine. The resulting pair of study models has three pairs of reference planes (Figure 5.3) that are coplanar pairs only if the study models are presenting the correct occlusal status of the patient. It should be noted that the reference planes are perpendicular to the base (the largest plane) of the study model, which can also be used as a reference.

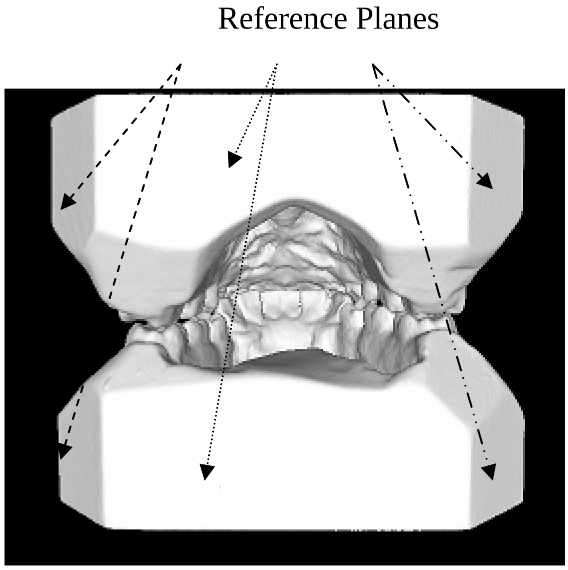


Figure 5.3: Reference planes.

By using these planes, orthodontists can easily reproduce the occlusal status of a patient by

1. Placing the study models such that the teeth are facing each other.
2. Checking the coplanar-status of the reference planes by using a plane surface, such as a tabletop.
3. Pushing the study models towards each other until physical contact is achieved between them.

Occlusal analysis is important in orthodontics treatment planning. Therefore it is desired to simulate the above bite-registration process in the orthodontics visualization system. In the following sections, we shall discuss this issue in detail.

5.2 Algorithm

The designed algorithm aims to simulate the procedure that is used to manually merge a pair of study models. It is a routine task in computer graphics to place the study models in a desired orientation so as to check the coplanar-status of the reference planes.

However, it is difficult to simulate the physical process whereby the two study models are pushed towards each other in a fixed direction until physical contact is made. The problem of collision detection between moving objects has been a popular research area as it is a fundamental process of simulating the physical world. It has been studied in many different communities including robotics, computer graphics, computer-aided design, and computational geometry. There are several algorithms for collision detection. Examples include Lin-Canny closest features algorithm, V-Clip, Collide, OBB-tree and KDS.

Despite of the presence of these algorithms, accurate and efficient, yet simple to implement, collision detection remains an open question. We decided not to follow any of the existing collision detection algorithms. The amount of transformation is restricted in our case, in which only translation in one dimension is considered and no rotation is allowed. In such a situation, using a range image in the direction of movement would suit the purpose well.

The designed algorithm includes the following steps:

- Repositioning – The study models are rotated and translated to a standard position, where their occlusal plane faces each other and the three pairs of reference planes are coplanar, respectively
- Range image generation – The study models have their 3D information projected to a 2D depth map, where the depth direction is defined as the direction along which the study models are being merged

- Merging – The study models are translated towards each other until physical contact is made. The amount of translation is directly determined from the range images previously generated

5.2.1 Repositioning

Different study models appear in different spots and poses in the 3D space when they are fed into the visualization system. The objective of repositioning is to transform a study model into a standard position and pose in preparation for further processing. The transformation includes:

- Rotation to standard orientation
- Translation to standard position (requires a pair of matching study models)

The standard orientation of a study model is defined in the following table.

Table 5.1: Standard Orientation of A Study Model

Criteria	Largest Surface	Second Largest Surface	Occlusal Plane
Lower Cast	Parallel to plane $y = 0$	Parallel to plane $z = 0$	Facing up (positive direction of the y axis)
Upper Cast	Parallel to plane $y = 0$	Parallel to plane $z = 0$	Facing down (negative direction of the y axis)

In reality, the largest surface of a study model is the one opposite the occlusal plane and the second largest surface is the middle one of the reference planes. Figure 5.4 shows two study models in their standard pose.

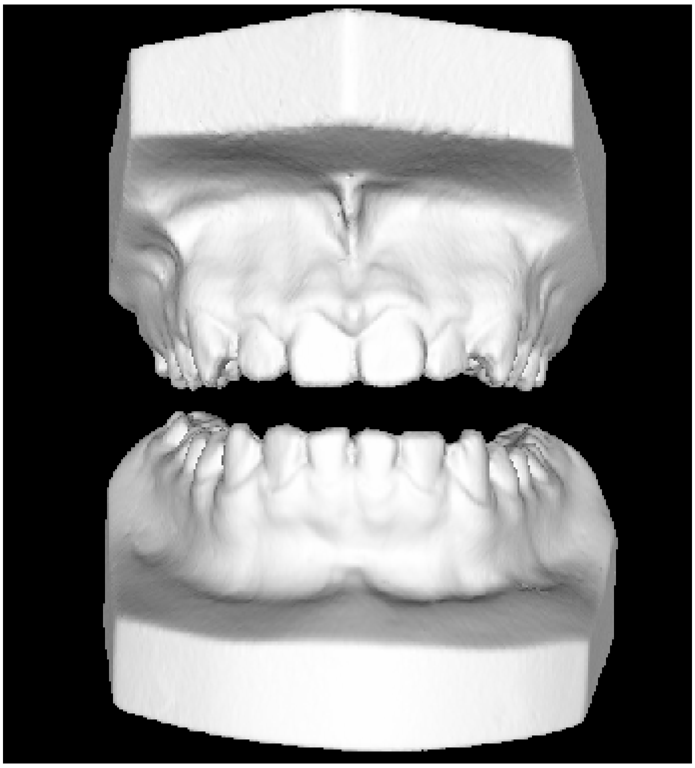


Figure 5.4: Study models in their standard pose.

The algorithm of rotating a study model to its standard pose consists of the following steps:

1. Calculate the normal of all the faces in the model.
2. Calculated the distribution of the face normals.
3. The face normal that occurs most frequently is taken as the normal of the largest surface.
4. The model is rotated about x and z axes such that the largest surface is parallel to plane $y = 0$ and faces down (negative direction of the y axis).
5. Project all faces to plane $y = 0$.
6. Calculate the normal of the projected faces.
7. Calculate the distribution of the face normals.
8. The face normal that occurs most frequently is taken as the normal of the second largest surface.

9. The model is rotated about y axis such that the second largest surface is parallel to plane $z = 0$.
10. If the model is a upper cast, rotate the model about z axis by 180 degrees.

After rotating a matched pair of study models to their standard orientation, the algorithm continues to translate the models to a standard position defined in Table 5.2,

Table 5.2: Standard Position of A Study Model

Criteria	Centroid	Reference Planes
Lower Cast	At the origin $(0,0,0)$	-
Upper Cast	-	Coplanar with respective reference planes of the lower cast

The algorithm of translating a pair of study models to the standard position consists of the following steps:

1. Calculate the centroid of the lower cast by averaging all the vertices.
2. Translate the lower cast such that its centroid lies at the origin.
3. Filter the face list of the lower cast to find all faces parallel to plane $z = 0$.
4. Calculate the minima of z coordinates of the filtered face list, $z_{\min}$.
5. Reduce the size of the face list further by setting the higher threshold of z coordinates to $z_{\min} + \varepsilon$, where ε is a small positive number.
6. Calculate the mean z coordinate of the filtered face list, z_{mean}.
7. Calculate the maxima and minima of x coordinate of the filtered face list, $x_{\max}$ and $x_{\min}$.
8. Calculate the mean of $x_{\max}$ and $x_{\min}$, x_{mean}.
9. Repeat Step 1 to 8 in order to find out the value of x_{mean} and z_{mean} for the upper cast, denoted as x'_{mean} and z'_{mean}.

10. Translate the upper cast by 3D translation vector $(x_{mean} - x'_{mean}, 0, z_{mean} - z'_{mean})$.

In reality, the straight line $\begin{cases} x = x_{mean} \\ z = z_{mean} \end{cases}$ represents the midline of the middle reference plane (Figure 5.5). Matching the midlines of both study models explicitly ensures the coplanarity of all the reference planes. Figure 5.5 also shows the result after repositioning.

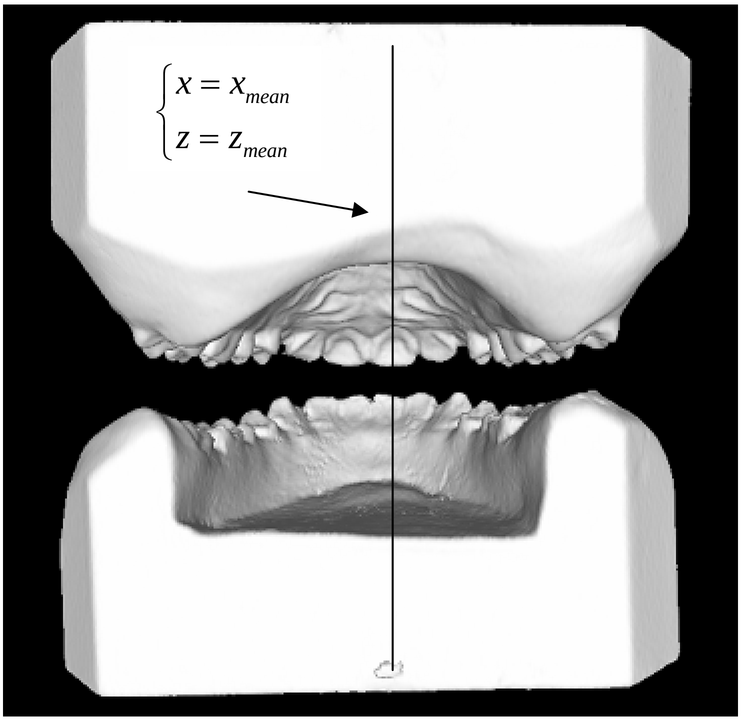


Figure 5.5: Result after repositioning.

5.2.2 Range Image Generation

The previous step, repositioning, has simulated the physical process of putting two study models together and checking their coplanarity. The remaining step is to move one of the model towards the other until physical contact is made. In this step, only translation in the y direction is allowed. Therefore, the objective is to calculate the amount of translation in the y direction which will result in physical contact between models. This is achieved by generating the range images of both models along the y direction and comparing the range images.

A range image is a 2D representation of a 3D object where the height and width information is preserved and the depth information is linearly converted to pixel intensity. Since we are concerned about translation in one direction, analysis using range images can reduce the complexity of the problem significantly.

In order to achieve high accuracy in the range image generation, we also make use of the technique of further triangulation to increase the effective resolution of the 3D model. The following flowchart illustrates the procedure of generating the range image of a lower cast (Figure 5.6). Firstly, a set of parameters of the range image (such as the boundary, resolution of the range image and the default value of pixels) is determined for a pair of study models. In order to maintain the consistency, the range image is initialized with a default value that bears the physical indication of the minimum height presented in the model. The face list is traversed through and the related pixel of the range image is updated accordingly. A high resolution for the range image is enforced by a further triangulation mechanism, i.e., sub-dividing one triangle into four smaller triangles (by linking the mid-points of the edges) and repeat this procedure when necessary.

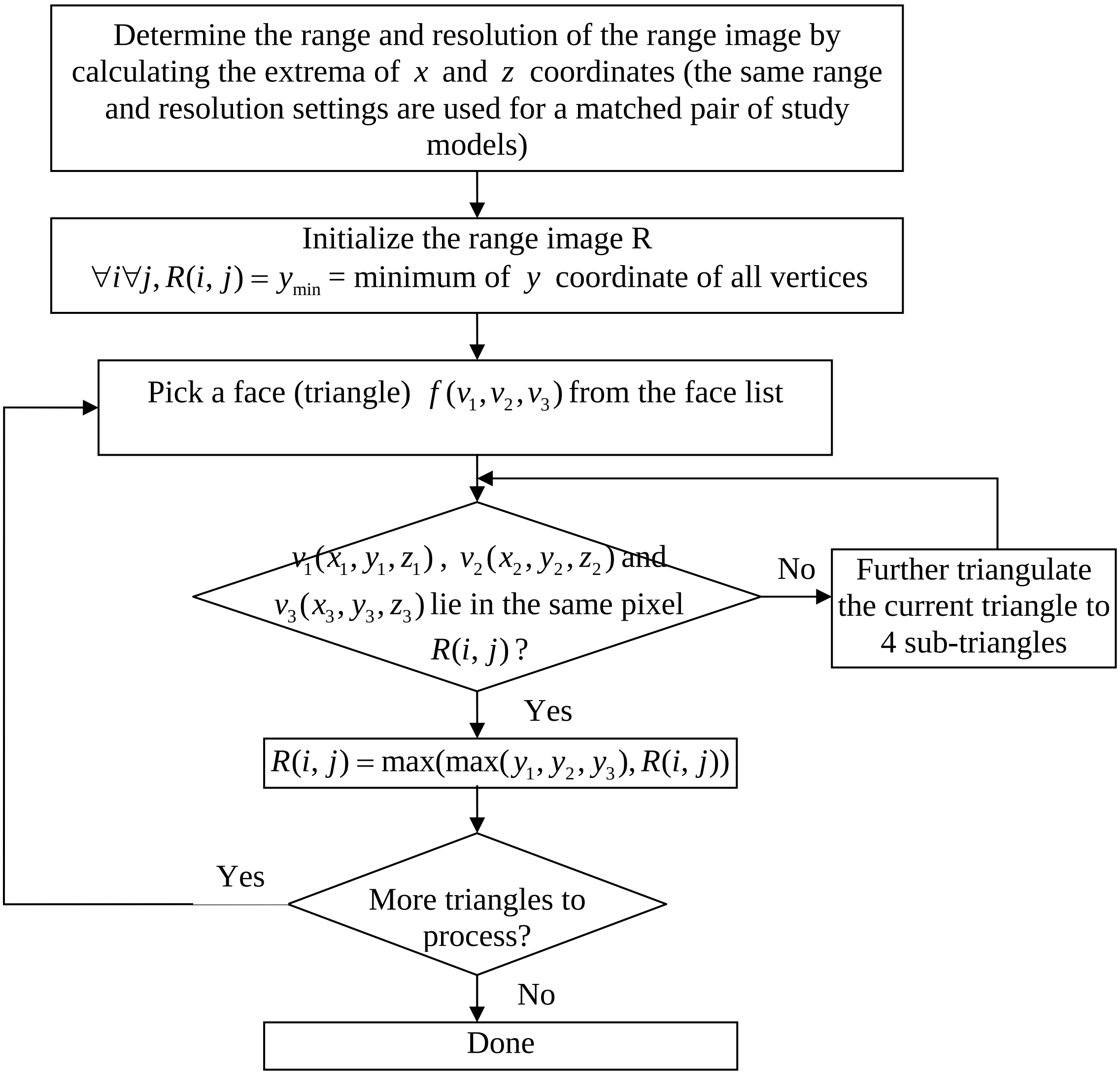

Determine the range and resolution of the range image by calculating the extrema of x and z coordinates (the same range and resolution settings are used for a matched pair of study models)
Initialize the range image R
$\forall i \forall j, R(i, j) = y_{\min}$ = minimum of y coordinate of all vertices
Pick a face (triangle) $f(v_1, v_2, v_3)$ from the face list
$v_1(x_1, y_1, z_1)$, $v_2(x_2, y_2, z_2)$ and $v_3(x_3, y_3, z_3)$ lie in the same pixel $R(i, j)$?
No
Further triangulate the current triangle to 4 sub-triangles
Yes
$R(i, j) = \max(\max(y_1, y_2, y_3), R(i, j))$
Yes
More triangles to process?
No
Done

Figure 5.6: Generating the range image of a lower cast.

Generated range images contain raw data representing height information of models. The raw data can be linearly converted to grey levels to make the range images displayable. Figure 5.7 shows a range image of a lower cast. It is easy to see that the range image actually contains the height information of the model. Therefore, the brighter regions in the figure correspond to higher regions of the model while the darker regions correspond to the lower regions.

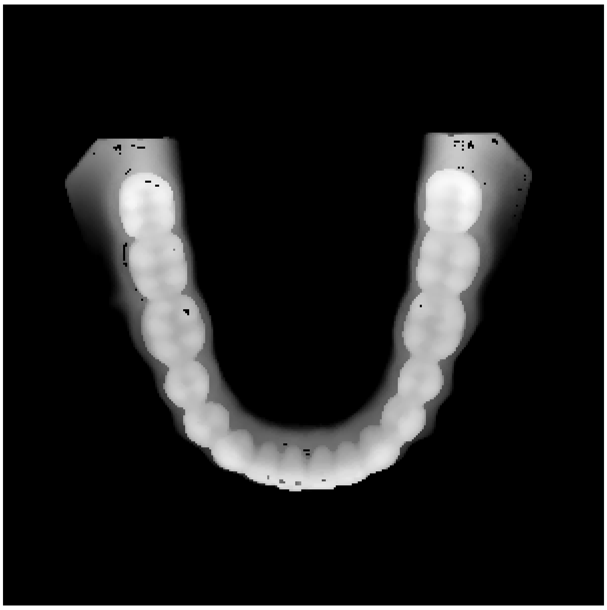


Figure 5.7: Range image of a lower model.

5.2.3 Merging

Once a pair of study models has been converted to range images with the same set of parameters, we have sufficient information to translate the models towards each other and make a merged model. By simply subtracting the lower range image (the range image generated from the lower cast) from the upper range image (the range image generated from the upper cast), we obtain a new range image that directly represents the height deviation of the models. The deviation range image is traversed through to determine the minimum height deviation before merging, which also corresponds to the amount of translation needed to make the minimum height deviation to zero, i.e., to create physical contact between the models. Figure 5.8 shows a merged pair of study models

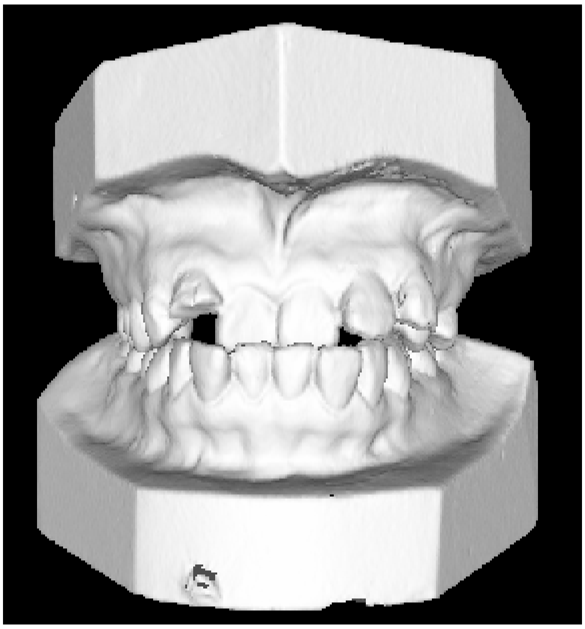


Figure 5.8: Merged models.

5.3 Refinement

The above algorithm provides a good simulation of manual process of merging study models. Experimental result on high-quality models is satisfactory. However, for old or poor quality models, the merging result is less accurate due to imperfect conditions on reference planes, i.e., the reference planes are not strictly coplanar when a correct bite is achieved.

In reality, the correctness of merging can be enforced by a simple concept. A correctly merged pair of study models should be "stable", i.e. the occlusal status of lower and upper teeth prevents the models from sliding apart from each other. As an improvement to the existing algorithm, this concept is integrated into the algorithm.

If we were to determine the exact amount of translation and rotation needed to achieve the best bite, we would have to perform a complete search over the entire domain to determine the global optimum. Such a search takes too much time, although range images generation has been carried out to reduce the complexity of the problem. In the algorithm, the concept of “stable bite” is interpreted as the follows: under a correct occlusal status, any slight transformation (rotation about the y axis and translation in $x-z$ plane) to the upper model causes the height deviation map to have a larger minimum, i.e., such a transformation would force the models to separate more from each other to avoid space collision. We also limit the search domain to be a discrete one by setting step-length in the searching of optimum translation and rotation values,

The amount of translation is considered to be multiples of the unit size of a pixel of the range image. Typically this value is less than 0.2 mm, smaller than that is proven to be clinically irrelevant. Such a presumption allows us to generate multiple height deviation maps by setting block offsets to one of the range image and hence avoid the time-consuming process of generating the range image. The step-length in searching for the optimum rotation amount is user-definable from 0.1 to 1 degree.

The above measures greatly reduce the search time. Experimental result shows that the entire merging process can be finished in few minutes. The improvement on the merging result is significant

5.4 Results

Experiment on the above mentioned occlusion analysis procedure was conducted on a set of high quality study models. Both the lower and upper models were digitized as per normal, followed by a merging using the occlusion analysis procedure. In order to assess the accuracy of the merging, we measured and compared the mean distance between the bases of the models (while the models are merged) from both the physical models and the visualization system.

The measurement on the physical models was obtained by using a caliper with an accuracy of 0.02 mm. The measurement in the virtual environment was obtained by manually picking three points from each base plane and averaging their *y* coordinates. (Figure 5.9)

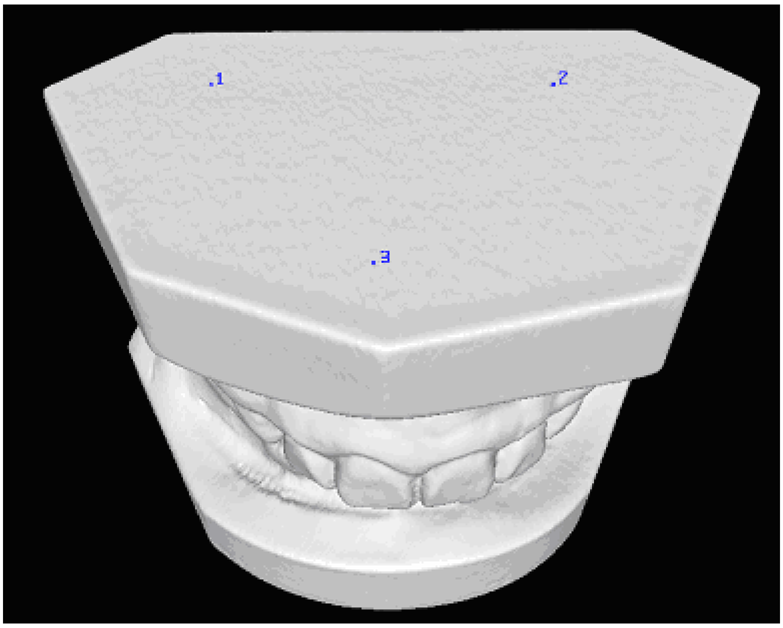


Figure 5.9: Measurement in the virtual environment.

Table 5.3 shows the experimental results.

Table 5.3: Merging result.

Model	Merging Type	Running Time (seconds)	Virtual Distance (mm)	Physical Distance (mm)
Model 1	Fast	35	73.21	72.58
	Accurate	1643	72.70	
Model 2	Fast	29	65.73	64.60
	Accurate	1119	64.70	

The proposed algorithm gives satisfactory result. The faster (without improvement on accuracy) procedure completes the merging within one minute and gives fairly accurate result. If more accuracy is required, the slower (with improvement on accuracy) procedure is able to further merge the models and delivers an error of 0.12 to 0.15 mm, which is more than sufficient in clinical purpose which requires about 0.3

mm in terms of accuracy. Figure 5.10 shows the merging result by using both procedures.

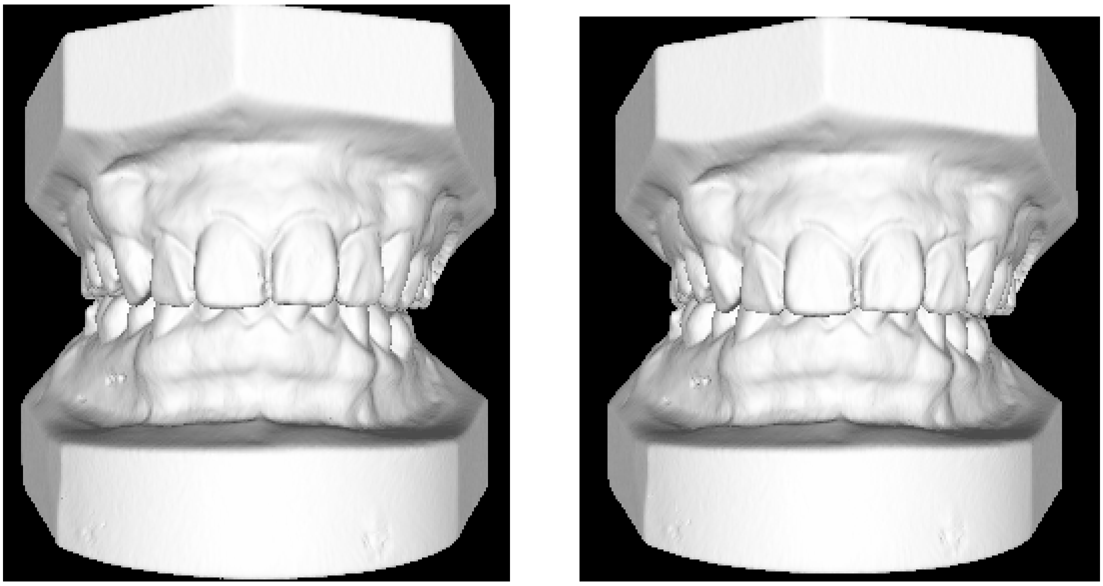


Figure 5.10: Merging results of both procedures on Model 1
Left – Fast, Right – Accurate

Chapter 6

Conclusions and Future Work

Orthodontics refers to the specialty and practice of preventing and correcting irregularities of the craniofacial complex, particularly the growth of teeth. Conventionally, orthodontists conducts treatment planning based on their expertise, with the help of various information sources, such as study models and the orthopantomogram. Orthodontics treatment is a long-term process which requires frequent visits with orthodontists, during which the treatment plan is gradually developed. With rapid advances in computer vision technology, there has been great demand in developing computer visualization systems to facilitate the tedious, manual flow of orthodontics treatment planning.

In this study, we developed such an orthodontics visualization system. Conventional assessment of malocclusions using study models and the orthopantomogram is replaced by 3D computer simulations. Automatic crown segmentation allows orthodontists to freely re-arrange individual crowns to achieve satisfactory occlusion status. Moreover, the system is able to estimate 3D root shape by deforming generic tooth model with reference to the actual orthopantomogram. By merging the estimated root shape with the crown information from the study model, the system is able to present full set of 3D tooth (with roots) without using CT scans, which is a unique feature among similar attempts in orthodontics visualization.

Occlusion analysis enables orthodontists to visualize the occlusal status of a pair of study models. With the help of transformation tracing mechanism, the system is also able to simulate the occlusal status of a re-arranged dentition.

Besides replacing the manual assessment of study models and orthopantomograms with computer visualization process, we also bring in other data sources, such as 3D facial model and the lateral cephalogram, to provide orthodontists with more information in treatment planning. With the ability of incorporate the above medical imaging techniques, the system allows orthodontists to assess the case in different perspectives.

Preliminary results have shown that this visualization system is able to provide adequate degree of accuracy in measurements. The system has the potential in facilitating orthodontist in orthodontics routines. Moreover, it has the potential in replacing the manual process in orthodontics treatment planning with powerful computer simulations and thus changing the flow of orthodontics treatment. However, a careful clinical study is needed before this visualization system can be put in practical applications.

This study is part of an entire project, the ultimate objective of which is to develop a visualization system which is capable of presenting a virtual human head to help doctors in planning of relevant clinical treatments, such as orthodontics treatments or craniofacial surgeries. By incorporating various data sources, such as study model, CT, MRI, panoramic and cephalometric X-ray, the visualization system helps doctors in predicting the outcome of possible treatment plans and hence to decide which one should be taken in practice.

The system still offers many opportunities in further improvement and possible extensions, such as:

- Reducing the amount of user-interaction in the system by replacing it with reliable automatic routines
- Improving the accuracy, efficiency and robustness of existing algorithms

As part of the entire research project, this study has covered most of the topics related to study models, orthopantomogram, facial models and cephalogram. However, in order to present a virtual human head, more data need to be acquired from other imaging source, such as computerize tomogram (CT) and magnetic resonance imaging (MRI), which is also a possible extension to this study.

References

1. G. T. Herman, “A survey of 3D medical image technologies,” *IEEE Engineering in Medicine and Biology*, vol. 9, no. 4, pp. 15-17, Dec. 1990.
2. S. M. Yamany, A. A Farag and N. A. Mohamed, “Orthodontics measurements using computer vision,” *Proc. of 20th Annual Intl. Con. of IEEE in Med. and Bio. Soc.*, vol. 20, no. 2, pp. 536-539, Oct. 1998.
3. K. Yamamoto, H. Morikawa, A. Tomochika, S. Hayashi, S. Nakamura and T. Mikami, “Three-dimensional measurement of dental cast profiles and its applications to orthodontics,” *Intl. Con. of IEEE in Med. and Bio. Soc.*, vol. 12, no. 5, pp. 2052-2053, 1990.
4. J. Mah, R. Sachdeva, “Computer-assisted orthodontic treatment： The SureSmile process”, *Am J Orthod Dentofacial Orthop*, vol. 120, pp. 85-87, 2001
5. R. Enciso, A. Memon, U. Neumann and J. Mah, “Three-dimensional visualization of the craniofacial patient: volume segmentation, data integration and animation”, *Orthodontics & Craniofacial Research*, vol. 6, no. s1, pp. 66-71, 2003
6. M.Y. Hajeer; D.R. Millett, A.F. Ayoub and J. P. Siebert, “Applications of 3D imaging in orthodontics: part I”, *Journal Of Orthodontics*, vol. 31, no. 1, pp. 62-70, 2004
7. D. M. Sarver, “Video cephalometric diagnosis (VCD): a new concept in treatment planning”, *Am J Orthod Dentofacial Orthop*, vol. 110, no. 2, pp. 128-136, 1996

8. N. Noguchi; M. Goto, “Computer simulation system for orthognathic surgery”, *Orthodontics & Craniofacial Research*, vol. 6, no. s1, pp. 176-178, 2003

9. B. Kusnoto, “Reliability of a 3D surface laser scanner for orthodontic applications,” *American Journal of Orthodontics and Dentofacial Orthopedics*, vol. 122, no. 4, pp. 342-348, 2002

10. M. Santoro, S. Galkin, M. Teredesai, O. F. Nicolay, T. J. Cangialosi, “Comparison of measurements made on digital and plaster models,” *Am J Orthod Dentofacial Orthop*, Jul, vol. 124, no. 1, pp. 101-105, 2003

11. P. Borrel and A. Rappoport, “Simple constrained deformations for geometric modeling and interactive design,” *ACM Transactions on Graphics*, vol. 13, no.2, pp. 137-155, 1994.

12. S. Yoshizawa , A. G. Belyaev , H. P. Seidel, “A Simple Approach to Interactive Free-Form Shape Deformations,” *Proceedings of the 10th Pacific Conference on Computer Graphics and Applications,* p.471, October 09-11, 2002

13. L. Kobbelt, T. Bareuther, and H. P. Seidel, “Multi-resolution shape deformations for meshes with dynamic vertex connectivity,” *Computer Graphics Forum Eurographics*, vol. 19, no. 3, pp. 249-260, 2000.

14. B. P. Zhao, “Registration of multimodal dental and facial images”, 2004

15. J. D. Decker, “TECHNO BYTES - Image editing,” *American Journal Of Orthodontics*, vol. 125, no. 2, pp. 215-219, 2004

16. T. Gedrange; C. Bourauel; C. Köbel and W. Harzer, “Simulation of bone strain by orthodontic implants using the finite element method,” *Biomedizinische Technik. Biomedical Engineering*, vol. 48, no. 10, pp. 287-290, 2003

17. J. Zhou; S. Luo; X. Yu and N. Huang, “Research and establishment of the computerized orthodontic and orthopedic prediction system,” *West China Journal Of Stomatology*, vol. 19, no. 1, pp. 49-51, 2001

18. L. Keilig; K. Piesche; A. Jäger; C. Bourauel, “Applications of surface-surface matching algorithms for determination of orthodontic tooth movements,” *Computer Methods In Biomechanics And Biomedical Engineering*, vol. 6, no. 5-6, pp. 353-359, 2003

19. R. J. Cousley; E. Grant and J. D. Kindelan, “The validity of computerized orthognathic predictions,” *Journal Of Orthodontics*, vol. 30, no. 2, pp. 149-154, 2003

20. G. T. McIntyre and D. T. Millett, “Crown-root shape of the permanent maxillary central incisor,” *The Angle Orthodontist*, vol. 73, no. 6, pp. 710-715, 2003

21. T. Stamm, U. Meyer, N. Meier, U. Ehmer and U. Joos, “Public domain computer-aided surgery (CAS) in orthodontic and maxillofacial surgery,” *Journal Of Orofacial Orthopedics*, vol. 63, no. 1, pp. 62-75, 2002

22. A. C. Beers; W. Choi; E. Pavlovskaia, “Computer-assisted treatment planning and analysis,” *Orthodontics & Craniofacial Research*, vol. 6, pp. 117-125, 2003

23. M. Alcăniz, F. Chinesta, C. Monserrat, V. Grau. and A. Ram´on, “An advanced system for the simulation and planning of orthodontic treatments,” *4th Int. Conf. of Visualization in Biomedical Computing*, Hamburg, 1996.

24. D. Laurendeau, L. Guimond, and D. Poussart, “A computer-vision technique for the acquisition and processing of 3-D profiles of dental imprints: an

application in orthodontics," *IEEE Trans. Med. Imag.*, vol. 10, no. 3, pp. 453-461, 1991.

25. J. Cote, D. Laurendeau, and D. Poussart, "A multi-operator approach for the segmentation of 3-D images of dental imprints," *Advances in machine vision, Strategies and Applications*, *World Scientific Publishing*, pp. 343-360, 1992.
26. M. Mokhtari and D. Laurendeau, "Feature detection on 3-D images of dental imprints," *Proceedings of the IEEE Workshop on Biomedical Image Analysis*, pp. 287-296, Jun. 1994.
27. J.H. Chuah, S.H. Ong, T. Kondo, K.W.C. Foong, and T.F. Yong, "3D space analysis of dental models," *SPIE International Symposium on Medical Imaging 2001*, *San Diego*, Feb. 2001.
28. V. Mok, S.H. Ong, K.W.C. Foong, and T. Kondo, "Pose estimation of teeth through crownshape," *SPIE International Symposium on Medical Imaging 2001*, San Diego, Feb. 2002.
29. T. Kondo, S.H. Ong, K.W.C. Foong, "Tooth Segmentation of Dental Study Models Using Range Images," *MedImg*, vol. 23, no. 3, pp. 350-362, 2004
30. E. Angel, "Interactive Computer Graphics with OpenGL," *Addison-Wesley*, 1997.
31. R. Enciso, J. Mah, "3D Tooth Shape from Radiographs Using Thin-Plate Splines," *Studies in Health Technology and Informatics*, vol. 94, pp. 62-64, 2003
32. J. Davis, S. R. Marschner, M. Garr and M. Levoy, "Filling holes in complex surfaces using volumetric diffusion," *First International Symposium on 3D Data Processing, Visualization, and Transmission*, *Padua, Italy,* June 19-21, 2002

33. M. B. Dillencourt, "Polyhedra of Small Orders and Their Hamiltonian Properties," *Tech. Rep. 92-91, Info. and Comput. Sci. Dept., Univ. Calif. Irvine*, 1992

34. H. Hassan, A. El-Baz, A. G. Farman and M. Miller, “A 3D Model of The Human Jaw,” *Proced. of Computer Assisted Radiology and Surgery (CARS’05)*, Berlin, Germany, June, 2005

Appendix

Advanced Techniques in 3D Visualization

A.1 Overview

An orthodontics visualization system is different from a general-purpose 3D application in at least the following aspects,

3D scene complexity – In order to achieve a high degree of accuracy, 3D orthodontics models are usually of very fine resolution and contains extremely large amount of polygons. General-purpose 3D applications are simply unable to handle such models. For example, loading a digitized study model in commercial software, such as 3D Studio Max, causes the computer system to hang.

Complicated affine transformations – An orthodontics visualization system has to provide interfaces for handling complicated affine transformations to 3D models in order to simulate treatment effects. Capabilities such as performing arbitrary transformation on individual models in a scene become basic requirements. Tracking a sequence of affine transformation is particularly useful as it allows transformations to be easily replicable.

In the following sections, we shall discuss several advanced computer vision techniques that have been implemented in the visualization system to enhance system functionalities and user experience.

A.2 Fast Rendering by Triangle Stripping

A.2.1 Theory

Smooth and responsive rendering plays a vital role in user experience over a visualization system. No one would be interested in using a visualization system which takes a few seconds to refresh a scene.

We first define a commonly used unit. FPS, standing for frames per second, describes on average how many frames can a system renders a scene within one second. FPS is directly related to user experience in using a visualization system. The higher the FPS, the smoother a user feels the system to be in rendering motion scenes. It is commonly accepted that a minimum of 25 FPS is needed to achieve satisfactory effects.

Taking a high resolution study model for example, such a model usually consists of more than 100,000 vertices and 300,000 triangles. Experimental result shows that an FPS of 25 can be achieved on a 2.4GHz PC. However, the system has to handle far more complicated scene than barely one study model. For example, in order to perform occlusal analysis, the system has to render two such models where the FPS drops to 12.

Given a fixed hardware platform, the speed at which a system renders frames depends on the amount of data it needs to process in each frame. Since we are only concerned about triangulated models, it is easy to see that the speed of rendering meshes depends on the number of triangles need to be processed. Going one step further, the speed depends on the number of vertices (that form the triangles) need to be processed. Figure A.1 shows the procedure of rendering a scene. Scene data is first transferred from system memory to the CPU where the triangles are streamlined into vertices and transferred to the 3D acceleration card, which performs affine transformation and rasterization. Finally, the rasterized scene is sent to a monitor to be displayed.

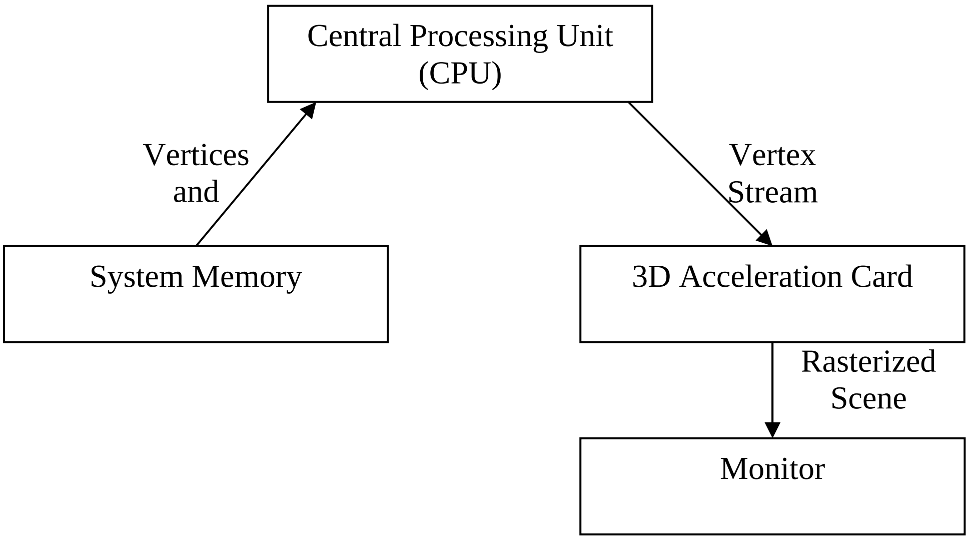


Figure A.1: Rendering a 3D scene.

Nowadays the CPU and the 3D acceleration chip are extremely fast as compared to the link between them, which has become the major bottlenet in the above rendering procedure.

It is certainly impossible to reduce the number of triangles in a model without affecting the model quality. However, we can render the triangles in a more efficient way to reduce the vertices that have to be processed. Triangle stripping is one of the most popular way of improving the efficiency of rendering triangulated meshes.

Triangle stripping exploits the facts that in a triangulated mesh, most triangles shares vertices with their adjacent triangles. Triangle strip primitives take advantage of triangle adjacencies and describe a sequence of adjacent triangles by using a list of points where three consecutive points represent one triangle. The following figure (Figure A.2) illustrates the idea of triangle stripping in the most simplified case.

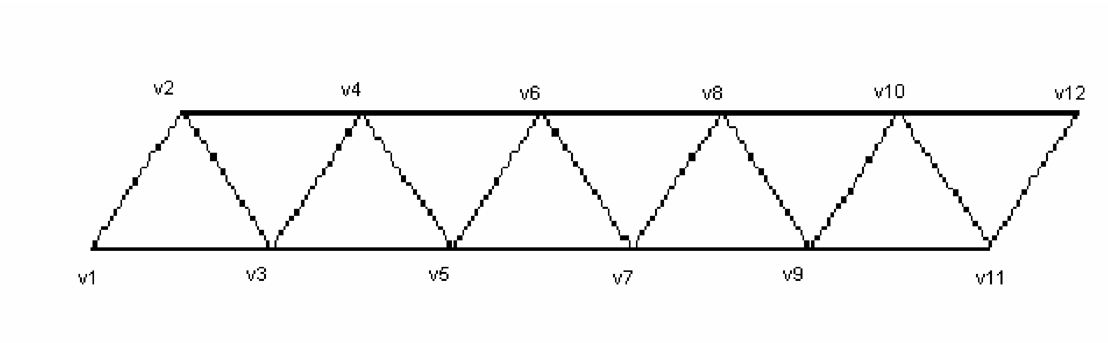


Figure A.2: Triangle stripping in a most simplified case.

For simplicity, the right-hand rule is ignored in triangle representations. This is the perfect case to use triangle stripping. There are 10 triangles in the scene, forming a complete strip of triangles. Under the conventional procedure, the vertex stream that is passing from the CPU to the 3D acceleration card contains $3 \times 10 = 30$ vertices,

$$\begin{gathered}(v1, v2, v3), (v2, v3, v4), (v3, v4, v5), (v4, v5, v6), (v5, v6, v7), \\ (v6, v7, v8), (v7, v8, v9), (v8, v9, v10), (v9, v10, v11), (v10, v11, v12)\end{gathered}$$

However, if triangle stripping is used, only $10 + 2 = 12$ vertices need to be passed through the narrow link between the CPU and the 3D acceleration card,

$$v1, v2, v3, v4, v5, v6, v7, v8, v9, v10, v11, v12$$

yet the rendering result is identical. Moreover, triangle stripping is ideal to work with the pipeline rendering mechanism, which all 3D acceleration cards implement in one way or another.

In general, it takes a stream of length $N + 2$ vertices to describe a triangle strip containing N triangles, while the conventional way takes $3N$ vertices to describe an identical scene. It is easy to see that in order to achieve maximum performance boost, it is desirable to describe a scene with as few triangle strips as possible, each of which containing as many triangles as possible. When the length of a triangle strip approaches infinity, it takes on average one vertex to pass through the link between the CPU and the 3D acceleration card to render one triangle, making it 300% as efficient as the conventional rendering procedure.

Given a triangulated mesh, the best way of rendering it using triangle strips is to construct one single triangle strip which passes through all the triangles in the mesh, which is equivalent to finding a Hamiltonian path in the bi-directional graph in the mesh. However, Dillencourt [33] shows that finding such a path is NP-difficult. Fortunately, it is not necessary to find the optimum way of constructing triangle strips to make use of them. The above analysis shows that even using very short triangle strips of length no more than a few triangles is able to exploit the fundamental concept and makes the rendering more efficient.

A.2.2 Implementation and Results

We have implemented a triangle stripifier (the algorithm that constructs triangle strips from triangles) in the visualization system to facilitate fast rendering, especially in highly complicated scenes. During the designing of the stripifier, we tried to achieve a balance between strip length and the stripifier's running time.

Experimental result shows that the implemented stripifier works very well in accelerating the rendering speed of the system. Stripifying is successfully carried out on every mesh, breaking the mesh into triangle strips of average length of 9.8 to 43.6 triangles. The running time is controlled within an acceptable level where the stripifier is able to process about 75,000 triangles per second on a 2.4GHz PC, making the processing time of a high quality model at about 5 seconds. The impact over the rendering speed is significant; the visualization system is able to deliver at least twice as much FPS after stripifying models.

Details experimental results, such as model size (number of vertices/triangles), stripifying time, average strip length, performance impact (in FPS) are tabulated in table A.2.

Table A.1: Experimental results of stripifier.

Scene	Scene Complexity		Stripifying Time (sec)	Average Strip Length	Performance Impact (avg. FPS)	
	#Vertices	#Triangles			Before	After
Facial Shape	20728	41303	0.4	10.9	133	320
Study Model	151398	302574	5.3	9.8	16	32
4X over-sampled facial model	330727	660848	22.4	43.6	9	25

With the aid of the stripifier, the visualization system is now able to deliver smooth and speedy rendering even again highly complicated scenes.

A.3 Affine Transformations in Homogeneous Coordinates

A.3.1 Theory

In computer graphics, an affine transformation is a function that takes a point and maps it into another point [31]. Commonly used affine transformations include the following,

- Translation – Displacing of points to new positions defined by a displacement vector
- Rotation – Difficult to define in Cartesian coordinate system, but in polar 2D coordinate system, rotation around the origin can be taken as a change to the angle coordinate. Similar idea is extended to other coordinate systems of high dimensions.

- Scaling – Non-rigid-body transformation (i.e. the shape of the target is altered during transformation) around a fixed point that makes an object bigger or smaller. Scaling can be uniform or non-uniform.

More complicated affine transformations, such as shear, can be considered as properly concatenation of basic transformations.

It is often preferable to work with a 4D space in computer graphics, although systems are designed to dealing with 3D objects in 3D space. The homogeneous coordinate representation appends another dimension to the normal 3D representation of points or vectors. The conversion of point representations between Cartesian coordinate system and homogeneous coordinate system takes the following form,

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \Leftrightarrow \begin{bmatrix} kx_0 \\ ky_0 \\ kz_0 \\ k \end{bmatrix}$$

where k is a scalar.

Discussions of the analytical impact of using homogeneous coordinate system is certainly beyond the scope of this thesis and would not be presented here. In a nutshell, one can conclude that by using homogeneous coordinate representation, conventional affine transformations can all take a new form as matrix multiplications, which is ideal to be evaluated by sophisticated pipeline-designed digital signal processors found in most graphic systems. In homogeneous coordinate system, affine transformations can be represented in terms of the following matrix multiplications,

$$\mathbf{v} = A\mathbf{u}$$

where $\mathbf{u}$, $\mathbf{v}$ stands for the homogeneous representation of the point (or vector) before and after the transformation respective, and the transformation matrix is of the form

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

The 12 values can be set to any scalar value; hence the transformation is said to have 12 degrees of freedom.

As an example, the following discussion illustrates the conversion of translation in Cartesian coordinate system to matrix multiplication in homogeneous coordinate system.

In Cartesian coordinate system, translation of a point $\mathbf{p}$ to $\mathbf{p}'$ by a distance vector $\mathbf{d}$ takes the following form,

$$\mathbf{p'} = \mathbf{p} + \mathbf{d}$$

In homogeneous coordinate system,

$$\mathbf{p} = \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}, \mathbf{p}' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}, \mathbf{d} = \begin{bmatrix} \alpha_x \\ \alpha_y \\ \alpha_z \\ 0 \end{bmatrix}$$

where

$$\begin{cases} x' = x + \alpha_x \\ y' = y + \alpha_y \\ z' = z + \alpha_z \end{cases}$$

However, the transformation can also be written in matrix multiplication form

$$\mathbf{p}' = \mathbf{Tp}$$

where

$$\mathbf{T}=\begin{bmatrix}1&0&0&\alpha_x\\0&1&0&\alpha_y\\0&0&1&\alpha_z\\0&0&0&1\end{bmatrix}$$

since

$$\mathbf{Tp}=\begin{bmatrix}1&0&0&\alpha_x\\0&1&0&\alpha_y\\0&0&1&\alpha_z\\0&0&0&1\end{bmatrix}\begin{bmatrix}x\\y\\z\\1\end{bmatrix}=\begin{bmatrix}x+\alpha_x\\y+\alpha_y\\z+\alpha_z\\1\end{bmatrix}=\mathbf{p}'$$

Similarly, other affine transformations, such as scaling, rotation and shear can also be written as matrix multiplications in homogeneous coordinate system as shown in the following table (Table A.3)

Table A.2: Transformation Matrixes of Affine Transformations.

Affine Transformation	Transformation Matrix
Rotation	$\mathbf{R}_x=\mathbf{R}_x(\theta)=\begin{bmatrix}1&0&0&0\\0&\cos\theta&-\sin\theta&0\\0&\sin\theta&\cos\theta&0\\0&0&0&1\end{bmatrix}$ $\mathbf{R}_y=\mathbf{R}_y(\theta)=\begin{bmatrix}\cos\theta&0&\sin\theta&0\\0&1&0&0\\-\sin\theta&0&\cos\theta&0\\0&0&0&1\end{bmatrix}$ $\mathbf{R}_x=\mathbf{R}_x(\theta)=\begin{bmatrix}\cos\theta&-\sin\theta&0&0\\\sin\theta&\cos\theta&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$
Scaling	$\mathbf{S}=\mathbf{S}(\beta_x,\beta_y,\beta_z)=\begin{bmatrix}\beta_x&0&0&0\\0&\beta_y&0&0\\0&0&\beta_z&0\\0&0&0&1\end{bmatrix}$
Shear	$\mathbf{H}_{x\sim y}(\theta)=\begin{bmatrix}1&\cot\theta&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}$, etc.

The matrix-multiplication representation of affine transformations not only allows the transformation to be evaluated efficiently using sophisticated hardware, it also provides a way of tracing concatenation of arbitrary affine transformations and evaluating them in a more efficient way.

Suppose that three successive transformations are carried out on a point $\mathbf{p}$ to create a new point $\mathbf{q}$,

$$\mathbf{q} = (\mathbf{C}(\mathbf{B}(\mathbf{Ap})))$$

where $\mathbf{C}, \mathbf{B}, \mathbf{A}$ are transformations matrixes (Figure A.3).

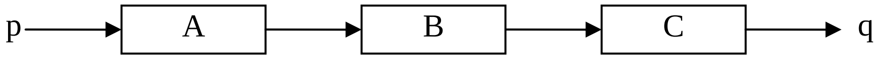


Figure A.3: Step-by-Step Transformations.

Because the matrix multiplication is associative, we can also write the above transformation as

$$\mathbf{q} = \mathbf{Mp}$$

where

$$\mathbf{M} = \mathbf{CBA}$$

and obtain the same result (Figure A.4).

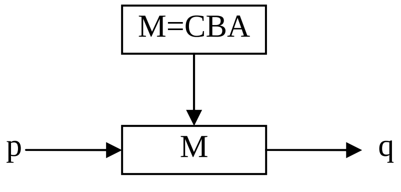


Figure A.4: Transformations in one step.

If the above transformation is applied to limited number of points, such an arrangement creates unnecessary overhead (in evaluating $\mathbf{M}$) and brings negative impact to the performance compared to calculating the transformations directly.

However, if the transformation is applied to an entire object which contains considerably large number of vertices, the overhead becomes insignificant and the new arrangement, which requires only one step of intensive calculation, is clearly preferred than the conventional way where three steps of such calculation are needed.

Moreover, theoretical analysis implies that transformation matrix **M** itself is sufficient to represent the entire concatenation of affine transformations. The same idea can be extended arbitrarily further; a series of affine transformations can always be represented by a single transformation matrix, which is evaluated by taking the product of all the transformation matrixes of each single transformation.

The ability of handling arbitrarily complicated affine transformations is essential in the orthodontics visualization system. The simulation of treatment effects basically relies all on various affine transformations which mainly include translations and rotations. Apart from rendering with affine transformations, the ability of tracing and replicating certain transformations is also needed. In the next section, we shall look at the benefits of implementing affine transformations in homogeneous coordinate system.

A.3.2 Implementation and Results

An implementation of affine transformations in homogeneous coordinate system is included in the orthodontics visualization system. In order to comply with the modular design concept, all related objects and mechanisms are encapsulated into a single class, which accepts conventional commands (such as rotation and translation), handles internally the commands using homogeneous coordinate system, and finally communicates with the rendering protocol (OpenGL) directly using transformation matrixes.

Such an arrangement not only delivers great efficiency in handling complicated affine transformations, it also supports tracing and replicating of transformations as shown in the following example.

Suppose we want to view a scene of biting crowns which requires two essential steps,

- Occlusal analysis – in order to merge lower and upper study models
- Tooth segmentation – in order to extract 3D tooth from study models

However, the above steps cannot be performed one after another since occlusal analysis transforms the study models, yet tooth segmentation can only be carried out when the study model is in a fixed position and orientation. In order to resolve the conflicts, we can trace the transformation during the occlusal analysis and replicate it on segmented teeth (Figure A.5).

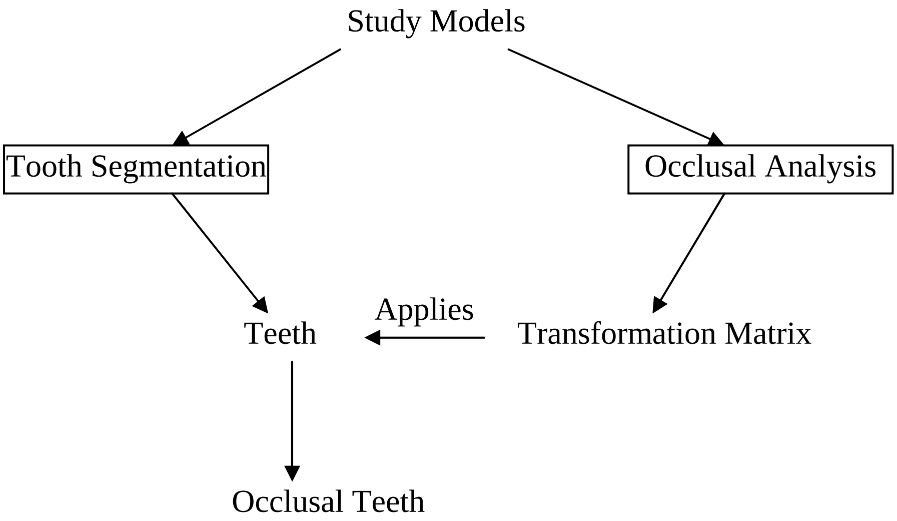


Figure A.5: Steps of obtaining occlusal teeth.

Conforming to our expectations, a scene of occlusal teeth is successfully created (Figure A.6).

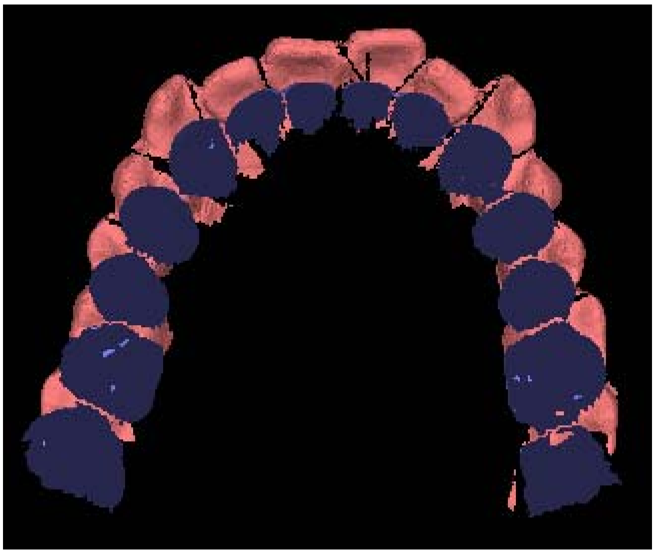


Figure A.6: A set of occlusal teeth.