

Simulating Craniofacial Growth

by

Kevin Michael Coughlan

B.Sc. (Computer Science) Université d'Ottawa, 1992

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
MASTER OF SCIENCE**

in

**THE FACULTY OF GRADUATE STUDIES
COMPUTER SCIENCE**

We accept this thesis as conforming
to the required standard

.....

.....

THE UNIVERSITY OF BRITISH COLUMBIA

April 1997

© Kevin Michael Coughlan, 1997

In presenting this thesis in partial fulfillment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

Computer Science

The University of British Columbia

2075 Wesbrook Place

Vancouver, Canada

V6T 1Z1

Date: May 4, 1997

Abstract

Current methods for facial reconstruction are tedious and time-consuming, and require forensic artists with years of practical experience. Furthermore, the complexity of the reconstruction problem greatly increases when time-related factors come into play, such as those that occur in missing children scenarios. This thesis describes a software system for simulating the growth of the craniofacial skeleton. It is a first step towards our goal of a complete software package for three-dimensional craniofacial reconstruction. There is a tremendous amount of data on craniofacial growth in the form of studies that collect frontal and lateral cephalograms, which can be used to generate three-dimensional coordinates of landmarks on the craniofacial skeleton at various ages. We define a simplified model of bone growth that uses these landmarks to drive the growth of the rest of the craniofacial skeleton. The inputs to our growth model include a triangular mesh acquired from the bone to be grown (e.g. skull, mandible), a set of vertices on the mesh identified as landmarks, the coordinates of these landmarks through time, and vertex weights which are a measure of the influence exerted by landmarks on the rest of the vertices. The output is a triangular mesh, “grown” either forwards or backwards in time to a specified age. An expert in craniofacial growth assigns these vertex weights by using a specialized tool called *Krayola*. We also provide a tool for automatically generating a first approximation for the vertex weights of a new mesh given the weights previously assigned to a mesh of similar bone type (e.g. skull, mandible). Validation of our growth model is an outstanding issue; we lack three-dimensional data (e.g. from CT scans) for an individual through time, with which we would compare the output of our software. For now, we must be content with the expert opinion of our colleagues in the Department of Dentistry’s craniofacial reconstruction group, who are quite pleased with our results so far.

Table of Contents

Abstract	ii
Table of Contents	iv
List of Figures	vii
Acknowledgments	ix
Chapter 1 Introduction	1
1.1 Overview of the Skull Growth Project	1
1.1.1 Applications to Forensic Anthropology	2
1.1.2 Craniofacial Growth Prediction	3
1.2 Thesis Objectives	4
1.3 Outline	5
Chapter 2 Background	7
2.1 Of Tissues and Cells	8
2.2 Overview of Craniofacial Growth and Development	9
2.2.1 Remodelling	10
2.2.2 Displacement	11
2.2.3 Remodelling and Displacement Combinations	12
2.3 Cephalometry and its Contribution	12
2.4 Previous Work	14
Chapter 3 Our Approach	15
3.1 Overview of the Software	15
3.2 Inputs	16
3.2.1 Triangular mesh	16

3.2.2	Starting age, target age and number of increments	17
3.2.3	Set of landmarks	19
3.2.4	Coordinates of landmarks through time	19
3.2.5	Influence of set of landmarks	20
3.3	The Simplified Growth Model	20
Chapter 4	Algorithms	25
4.1	The Growth Algorithms	25
4.1.1	Growth Along Normals Only	26
4.1.2	Growth Using Shift and Expansion Vectors	31
4.1.3	The Problem With Iteration	34
4.2	Automating the Generation of Weights.	35
Chapter 5	Implementation Details.	37
5.1	Geomview	37
5.1.1	External module - Landmark Picker.	39
5.1.2	External module - Krayola	40
5.2	Automatic generation of 1st guess at weights.	43
5.2.1	strip	44
5.2.2	procruste	45
5.2.3	apply	46
5.2.4	skullgrowth.	46
5.3	LEDA	46
5.4	File Formats.	47
5.4.1	Open Inventor.	47
5.4.2	OOGL and OFF	47
5.4.3	The Specifications File.	48
Chapter 6	Results and Analysis	51
6.1	The Growth Algorithm	51

6.1.1 Simple Geometry	51
6.1.2 Complex Geometries	53
6.2 Approximating the Weights	53
6.3 Validating the Method.....	57
Chapter 7 Conclusions	59
7.1 Summary	59
7.2 Future Research.....	62
7.3 Conclusion.....	63
Bibliography	65
Appendix A Estimating Coordinates of Landmarks Through Time	69

List of Figures

1.	Craniofacial skeleton vs. skull	8
2.	Newborn vs. adult craniofacial skeleton	9
3.	Incorrect conceptualization of the remodelling process	10
4.	Remodelling and the mandible	11
5.	Remodelling and displacement combinations yielding similar results	12
6.	Bolton Orientator with lateral and frontal tracings in position	14
7.	Simplified view of the workflow involved in using our system	16
8.	Triangular mesh of a skull, with landmarks.	18
9.	Direction of shift and expansion components	21
10.	Applying the shift and expansion terms.	22
11.	Effect of landmark influence on the growth of a circle	23
12.	Flowcharts for the growth algorithms	26
13.	Two valid ways for triangulating a quadrilateral.	27
14.	Computing the surface normal at a vertex.	28
15.	Area of influence of a landmark	29
16.	Computing the expansion growth vector of a landmark	30
17.	Overlapping, truncated Gaussian functions.	31
18.	Computing new coordinates of non-landmark vertices.	31
19.	The effect of iteration of our growth algorithm.	34
20.	Detailed workflow involved in using our system	38
21.	Snapshot of the <i>Krayola</i> environment	41
22.	Abbreviated program listing of <i>transferWeights</i>	44

23. Output of the strip program	45
24. Template for the specifications file	48
25. Example of a valid specifications file	49
26. Applying our growth algorithm to a sphere.	52
27. Applying our growth algorithm to a mandible	54
28. Applying our growth algorithm to a skull	55
29. Approximating weights of a new mesh	56
30. The problem of surface inter-penetration	63
31. Craniofacial landmarks on a lateral head film.	70

Acknowledgments

“R’gardez, les enfants! Un diplômé! [...] Ah ben. Un mémoire.”

À quelques reprises, j’ai été tenté de simplement dire que *“le chien l’auront probablement mangé.”* Maintenant que l’aventure tire à sa fin, je dois avouer que l’expérience en a valu le coup! Vive moi! Sentiment d’extase, de soulagement. Ouf! Mais comme dirait McGraw: *“Respirons par le nez!”* Me voilà rendu à la toute fin, les remerciements. Et j’en ai tellement...

Tout d’abord à mon directeur de recherche, mon guide, mon mentor, Dave Forsey. Quand je perdais le nord, tu me poussais gentiment dans la bonne direction. Merci, Dave, pour ton enthousiasme! Peter Cahoon, mon co-directeur de dernière minute, qui a le don de mettre toute chose en perspective, et qui n’hésite pas de le faire. Geerling Langenbach, ma personne-ressource en tout ce qui concerne le côté biologique de ma recherche, qui a pris le temps de lire la première ébauche de mon mémoire, l’a massacrée et ainsi rendu la version finale tellement mieux. Alan Hannam, qui a accepté d’être le deuxième membre de mon jury, et qui m’a donné une infusion de confiance en soi. Merci, les grands!

Mam, Dad, John, Stacie, je vous accorde votre propre paragraphe. C’est grâce à vous que je suis ici, que je suis qui je suis. Vous n’avez jamais perdu confiance en moi! Mon amour pour vous est sans limite, sans exception, sans fin. Et juste pour toi, Stacie, j’inclus ici Tux. *Envoie donc. Qu’en penses-tu, Dad? Tabarouette.*

Ian, you were integral to this thesis, not only as student reader, but for life’s little intangibles: a true friend indeed. I can’t thank you enough. Nancy et Stéphane, vous ne vous êtes jamais plaints de mes heures bizarres, ou de ma chambre en désordre: des colocs sans égal!

Merci pour les éclairs et pour Cocotte! Paul L., mon gouru en infographie, qui a toujours su répondre à mes questions, même les niaiseuses. Joel, ton enthousiasme était contagieux: il n'y a pas de doute, tu sais où tu t'en vas. Will, you always listened to my hare-brained ideas, poked and prodded at them, were a veritable fount of ideas and information. Many thanks to Paul F., sounding board "extraordinaire", who always had time for brainstorming and NetRunner. To Scott A., pool player without peer, who forced me to finish my thesis by luring me to video game development. To Marcelooooo ("caipirinha com cachaça!") and Rob W., 414 wouldn't have been the same without you. To Kori, who somehow managed to finagle what amounted to an office for two, and kept things hopping with Wednesday Pit nights. Jean-Luc, il va falloir prendre une bière pour fêter ça! Comme d'habitude... To Chris R., who on my first day declared for all to know that I "passed the test!". Believe it, Bill. Bill, Bill, Bill. So much I could say. And Gwen. Ah, the Pit nights. And the Paperboys. To Christian, for making sure that nothing untoward happened to me on the long way back from Arts County Fair (hic!). Stéphane pis François, qui n'ont pas hésité à me réveiller à 02h00 pour être photographe, santé! To Gene, for the rides back home after late night shifts. To the members of the CS basketball team, for putting up with my antics on the court. To Jim, for keeping the pop flowing. To Kelly, Alain, Jason, Stan, Bob, Rob S., and the rest of Imager, for teaching me so much about graphics, and about sticking to your guns. And all my thanks to those I've inadvertently omitted from this list (you know who you are!): I'll buy you a beer if you read this.

Finally, I'd like to thank Eric Borm, who introduced me to the wonderful sport of Ultimate, and in so doing to the one who has become the light of my life. Alison, les mots ne sauraient exprimer mon amour pour toi. Tu es, sans aucun doute, la fleur de mon coeur. Merci.

Kevin Michael Coughlan

The University of British Columbia, April 1997

1.1 Overview of the Skull Growth Project

Many fields of academic studies are showing growing interest in the development of graphical tools to aid both in research and in commercial applications. An example is the ongoing collaboration between a group of researchers at the University of British Columbia (UBC) and the Royal Canadian Mounted Police (RCMP). The UBC group is composed of members of the Media and Graphics Interdisciplinary Center (MAGIC) and of the Craniofacial Biology Laboratory of the Faculty of Dentistry. They are currently developing a system of methods for interactively modeling craniofacial reconstructions. Current scientific methods used in craniofacial biology will be combined with methods used by forensic artists to produce 3D panels of life-like images for subject identification.

The UBC group first met with the forensic division of the RCMP in June 1992. MAGIC was represented by Dr. Peter Cahoon and Dr. Kelly Booth, the Faculty of Dentistry by Dr. Alan Han-

nam, Dr. David Sweet and Dr. Geerling Langenbach, and the RCMP by Inspector Herb Leroy, Dr. Brian Yamashita and Officer Cameron Pye. The main theme of the meeting was the issue of forensic craniofacial reconstruction. Could graphical tools be used to aid in reconstructing the face of an individual, given only the craniofacial skeleton? The RCMP group was also particularly interested in the question of craniofacial growth prediction, and its application to the identification of missing children. In short, could we use a computer to simulate the aging of the craniofacial skeleton? If so, this could then be used to create pictures of missing children, artificially aged along user-specified criteria.

1.1.1 Applications to Forensic Anthropology

Forensic anthropology is the subdiscipline of physical anthropology which specializes in the fields of osteology and skeletal identification, as they pertain to courts of judicature and to public discussion. Coroners, medical examiners and forensic pathologists are routinely involved in the investigation and solving of unexplained deaths, and deal primarily in analyzing evidence in the form of remains. Sometimes, due to circumstances, the flesh remains cannot provide enough material for conclusive studies. Therefore, the hardier skeleton becomes the only avenue of information, and forensic anthropologists are called upon for their greater osteological expertise.

The first aim of forensic facial reconstruction is to aid in individual identification. Through this technique, the facial appearance of an individual is reconstructed from the bony remains of the skull. Existing tools and methods demand the services of forensic artists who are highly specialized in facial reconstruction. Furthermore, there are a wide range of equally probable faces that could reasonably be generated from a given skull, but investigators can usually only afford to produce a limited number. Alternative methods are needed, methods that could both speed up and facilitate facial reconstruction, as well as provide more conclusive information.

Forensic anthropology can also be applied to archeology. Research into human evolution throughout the ages, into the development and growth of civilizations, and the particular traits that characterized each one in different parts of the world has evolved around the recovery and analysis of ancient human skeletons and artifacts. Through the study of a variety of these artifacts, researchers can determine the way of life of a people: what tools were used, what clothes were worn, which musical instruments were played, even what diseases prevailed. However, the difficulty increases when efforts are made to draw the actual portrait or physiognomy of the people. Craniofacial reconstruction using skeletal remains provides a means to answer this question. In fact, it could become a medium through which scientists strive to determine and understand the effect of customs, way of life, habitat, religious beliefs, social phenomena, to name but a few, on the appearance and behaviour of past peoples. Past Forward Ltd. [20], based in England, is doing very interesting work in this regard for the Jorvik Viking Centre.

1.1.2 Craniofacial Growth Prediction

The problem of facial reconstruction is further complicated when time is factored into the reconstruction process. In particular, we are interested in providing assistance in the case of missing children. Current forensic approaches require a specialized artist to combine the facial features of the parents and child using only photographs, to predict the appearance of a child several years after disappearance. This method relies heavily upon the experience and talent of the artist in question. While the results can be quite convincing, it is very time-consuming process. Alternative, more interactive procedures could be of great benefit to the artist, especially if automation could reduce the more tedious aspects of facial reconstruction.

1.2 Thesis Objectives

The goal of this project is to provide a software system that can be used by a skilled practitioner to simulate the growth of the craniofacial skeleton. This project is part of a larger software chain, leading towards the ultimate goal of a software package for three-dimensional craniofacial reconstruction. As such, it is essential that we produce software that is both complete by itself (i.e. it works) and extensible (i.e. we can later include a much more complicated growth model without having to rewrite the entire system).

It should be clear that there are a tremendous number of biological factors that affect the growth of the craniofacial skeleton. Attempting to include all of these factors into our growth model is beyond the scope and intent of this project. In these initial stages of development, we concentrate on providing a simplified model of growth. Even though it is vastly simplified when compared to the biological model, our mathematical model will at least be *loosely* based on actual biological processes.

An example usage of our system is as follows. We start with a triangular mesh, acquired by performing a laser range scan of the skull of a 5 year old child. Using an off-line method such as the one described in Appendix A, we compute the coordinates through time of a small set of landmarks on the surface of the skull. Using custom-built software tools, we assign weights to each vertex which correspond to the amount of influence exerted on the vertex by each landmark. Given all these inputs, our system can construct a triangular mesh that represents the skull at any desired age, thus simulating its growth.

1.3 Outline

Chapter 2 briefly introduces the reader to the essentials of human craniofacial growth, from a biological point of view. The related field of cephalometry is described, along with the importance of craniofacial landmarks. Also, a brief mention is made on related works and publications.

Chapter 3 gives an overview of our growth simulation package and describes in detail the inputs and outputs of our system. This chapter also introduces the simplified growth models that are used to simulate bone growth.

Chapter 4 explores in further detail the algorithms behind the growth simulation. Also, we introduce and describe an algorithm used for semi-automatically assigning to vertices the amount of influence exerted by craniofacial landmarks on the rest of the craniofacial skeleton.

Chapter 5 looks at the implementation details of our software package, from file formats to third-party software libraries. We describe two custom-built tools for operations on triangular meshes: *Landmark Picker* simplifies the selection of craniofacial landmarks, while *Krayola* is used for assigning system-specific values to vertices of the mesh.

Chapter 6 examines the results of applying our growth algorithms to simple geometries such as spheres, and to more complex geometries such as skulls and mandibles. The problems associated with validation of the growth model are discussed.

Finally, chapter 7 summarizes the work that has been done in the context of this research project, looks at some open issues involved in the use of our system, and outlines avenues for future research.

Craniofacial growth is an extremely complex process. There are a great many functions packed into the relatively small space of the craniofacial region — sight, smell, speech, breathing and food intake, to name but a few. Every one of these functions has its own set of influences on bone growth which is in itself already complicated.

Enlow and Hans [10] provide an excellent introduction to the process of craniofacial growth and development. The purpose of their book is “to outline in *abridged format* the enormous range of morphogenic information dealing with key craniofacial growth concepts.”¹ In particular, it is well suited for researchers from other academic fields who wish to better understand the biological concepts that underlay craniofacial growth. The following chapter borrows heavily from the above-mentioned textbook, and also from Enlow [9] and Ranly [22], to describe the essentials of the craniofacial growth process.

1. page ix in Enlow and Hans [10].

A note on terminology. The reader might be confused by our use of *craniofacial skeleton* vs. *skull*. For our purposes, the craniofacial skeleton is composed of two major bone structures: the mandible and the skull.

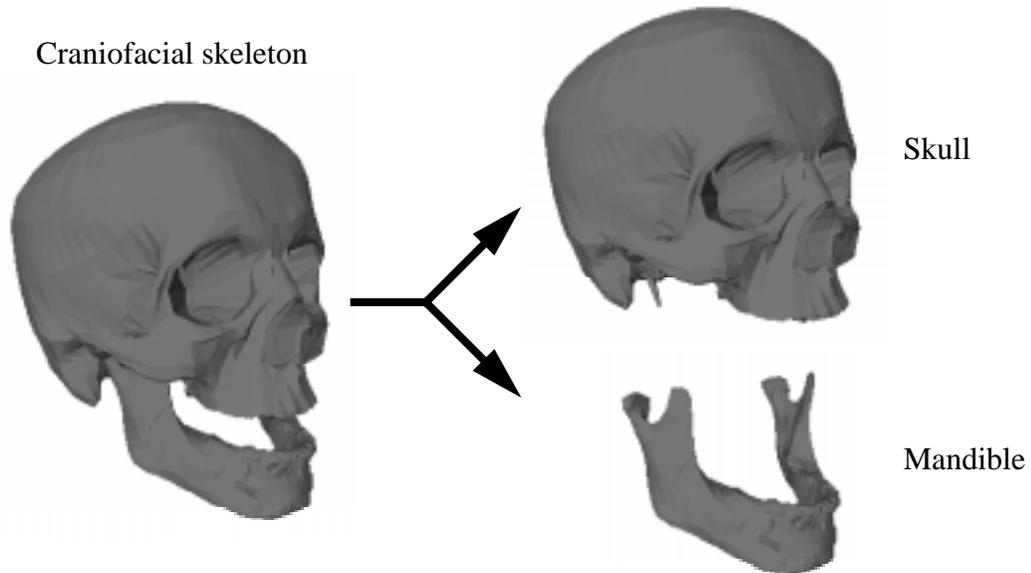


FIGURE 1. Craniofacial skeleton vs. skull

2.1 Of Tissues and Cells

The growth and development of a living structure is called *morphogenesis*. Morphogenesis is the formation and differentiation of tissues and organs, a biological process having underlying control at the cellular and tissue level. Differentiation is the sum of the processes whereby apparently indifferent cells, tissues and structures attain their adult form and function. Bone is “simply” a specialized type of tissue, but no craniofacial component is developmentally self-contained and self-regulated. Growth of a component is not an isolated event unrelated to other parts: it is the composite change of all components. As such, our task of providing a mathematical model for the growth and development of the craniofacial skeleton seems all the more daunting.

A note on terminology. The complex operations described above rely on carefully regulated “growth”. Growth is a general term implying simply that something changes in magnitude, but which does not necessarily account for how it happens. To try to understand how it works, and what actually happens, the more descriptive and explanatory “development” comes into play.

2.2 Overview of Craniofacial Growth and Development

Enlow and Hans define two basic kinds of growth movement that guide the facial growth process: *remodelling* and *displacement*. Remodelling is the process by which a bone’s shape changes through time. A bone does not appear in a prenatal infant in its final shape, and cannot simply grow by new additions keeping the same form. Rather, some areas of the bone grow faster or to a greater extent than others (Figure 2). Thus the term *remodelling*. Displacement is the process by which contiguous bones push away from each other as they grow, to allow for enlargement of the separate bones. While this may seem intuitive, it is important to realize that growth of a particular bone is not an isolated event. It is affected throughout its growth by all surrounding components.

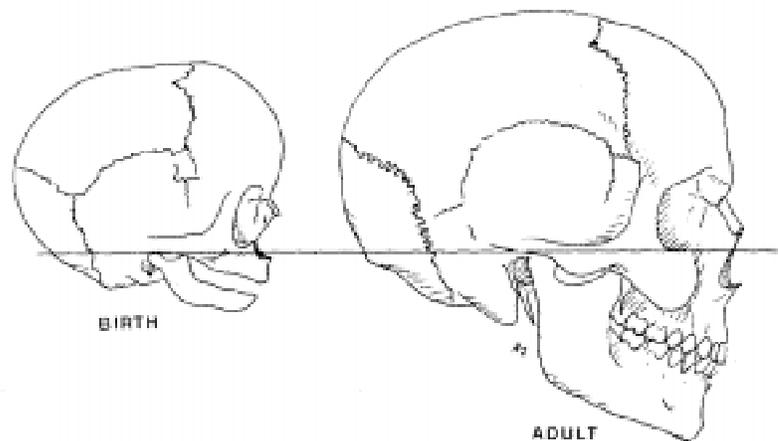


FIGURE 2. In the newborn the face is roughly 1/8 the size of the cranium, whereas in the adult the ratio is closer to 1/2.
Adapted from Figure 1-15, page 12, in Jacobson and Caufield [16].

2.2.1 Remodelling

A lay person might assume that bone grows “simply by generalized, uniform deposition of new bone (+) on all outside surfaces”¹ (see Figure 3). Such a growth process would not allow for the complex morphology of craniofacial bone, and for the complex interrelationships between the growth of bones, muscles and other soft tissues. Growth is a three-dimensional process: any given bone grows differentially, that is, it grows in some directions much more than others and at varying regional rates. Bones grow by virtue of addition, or “apposition”, of new bone tissue on one side of a surface and by removal, or “resorption”, of bone tissue on the other. If the rate of apposition is higher than the rate of resorption, then the bone’s surface increases in size and thickness. A bone’s inside and outside surfaces are covered by fields of growth, which are either appositionary or repository. This combination of apposition and resorption is responsible both for enlargement of the bone and for relocation of the bone in space.

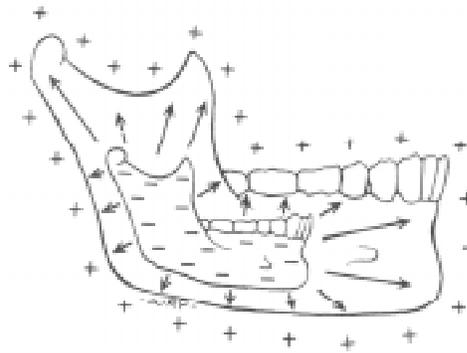


FIGURE 3. Incorrect conceptualization of the remodelling process.
Adapted from Figure 2-1, page 19, in Enlow and Hans [10].

Relocation is a key concept in the remodelling process. Consider the example in Figure 4, where bone is represented as a stack of chips. As growth occurs, bone tissue is added to the right

1. pages 18-19 in Enlow and Hans [10].

and removed from the left. Notice that the black chip's position in space does not change. In essence, the relative position of the black chip within the stack changes — it is being “relocated”. Notice how the apposition and deposition patterns (greatly simplified in this example) are responsible for both the enlargement and relocation of the mandible bone.

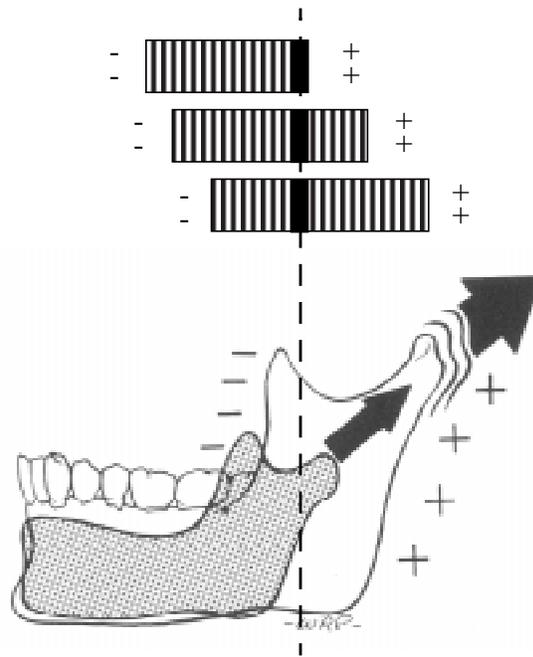


FIGURE 4. Remodelling and the mandible.
Adapted from Figure 2-3, page 20, in Enlow and Hans [10].

2.2.2 Displacement

As mentioned earlier, remodelling of a bone does not happen in isolation. The growing bone includes contacts with other bones which are also enlarging. Displacement occurs as these bones push away from each other. The whole bone is moved by mechanical means, as opposed to the relocation of the bone surface during remodelling. Consider the analogy of two balloons that are in contact with each other. As the balloons are inflated, they are displaced as they compete for the same space.

2.2.3 Remodelling and Displacement Combinations

Both remodelling and displacement are responsible for the growth movement of bones. The difficulty lies in determining which combinations of these two components occur for a particular section on the surface of a bone. Consider Figure 5, where bones *X* and *Y* are in contact with each other. All of the growth combinations depicted would have an overall similar effect. Chapter 3 in Enlow and Hans [10] contains an analysis of which of the many hypothetical combinations actually take place in the various regions of the craniofacial skeleton. For now, we will content ourselves with being aware of these combinations.

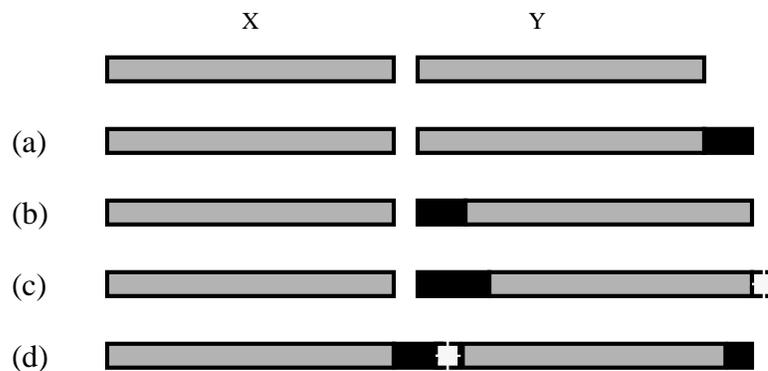


FIGURE 5. Remodelling and displacement combinations yielding similar results.
 (a) apposition on right side of *Y*; (b) apposition on left side of *Y*;
 (c) apposition on left side of *Y*, with resorption on right side of *Y*;
 (d) apposition on right side of *X* and *Y*, resorption on left side of *Y*.

2.3 Cephalometry and its Contribution

Cephalometry is the science of measuring the head, usually through the use of standardized lateral and frontal head radiographs, or cephalograms. Broadbent's seminal work in this field [3] describes a means for producing the lateral and frontal head films, with the patient in a fixed and reproducible position. There are numerous studies on morphological changes of the craniofacial

skeleton, of which Behrents [2] does an exhaustive review. Of particular interest are those that include the collection of frontal and lateral cephalograms. Hunter *et al.* [14] give a listing and description of longitudinal craniofacial growth record sets currently extant in North America. The largest of these is the Broadbent Bolton Growth Study, where cephalograms were collected on over 5000 children. A subset of these comprise a comprehensive longitudinal growth study of men and women of the ages 1 to 18 years, and have become known as the Bolton Standards.

In order to provide measurements of the craniofacial skeleton, there exists a set of agreed-upon points of reference, or landmarks. The number of landmarks that have been named and defined is quite large. Martin and Saller [19] list 69 landmarks on the craniofacial skeleton itself, while Behrents [2] identifies 87 landmarks from cephalograms. Admittedly, many of these landmarks are rarely used. Rogers [23] defines a smaller set of 30 landmarks, which contains those most frequently used in practice.

With proper calibration, frontal and lateral cephalograms can be used to obtain three-dimensional coordinates of certain landmarks. Grayson *et al.* [13] propose a standard, reliable set of such landmarks, and describe how to derive the three-dimensional coordinates of a landmark using the two-dimensional coordinates obtained from lateral and frontal cephalograms. Their technique is essentially the same as the one described by Broadbent *et al.* [4], where an “Orientator” is overlain on the frontal and lateral cephalograms (see Figure 6). This is used to establish a correlation between the two cephalograms, as they were related at the time the films were taken. This way, the points in one cephalogram that correspond to a point in the other can be located.

Subramanyan and Dean [27] describe custom-built visualization software for the analysis of frontal and lateral cephalograms, which facilitates the collection of three-dimensional landmark data. At the time of this writing, they were close to making available their collection of three-dimensional landmark data from the nearly 2000 “Bolton Standards” cephalograms.

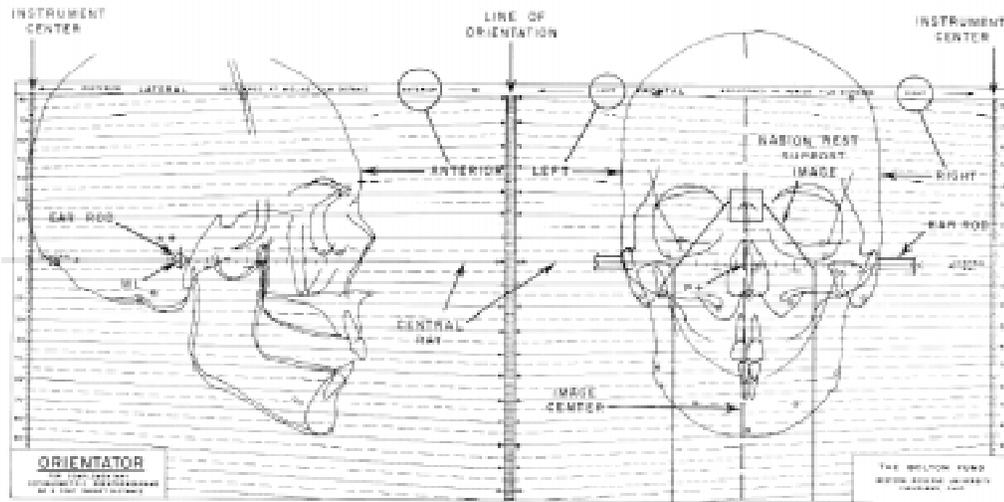


FIGURE 6. Bolton Orientator with lateral and frontal tracings in position. Adapted from Figure 3-5, page 32, in Broadbent *et al.* [4].

2.4 Previous Work

While there are an abundance of works that explore in detail the biological processes that dictate craniofacial growth and development, attempts to create computer graphical models of these processes are scarce. Cahoon and Hannam [5] simulate the “growth” backwards in time of an adult human mandible to that of a six year old child. The mandible was reconstructed from *ct* scans. Data on mandibular growth from the Saksena studies [24] [25] was used to drive a simple, three-dimensional free-form deformations algorithm. Landmarks on the mandible acted as control points for the deformations. The main drawback is that their technique is very unwieldy, which makes it difficult to properly control the deformation.

3.1 Overview of the Software

The purpose of the software is to simulate the aging of a craniofacial skeleton, either forwards or backwards in time. In theory, the inputs and outputs are simple: provide the system with a polygonal model of a craniofacial skeleton, a starting age and a target age. The output is a polygonal model of the craniofacial skeleton, aged to the specified target age. However, this would require an extremely complex, fully functional biological model of bone growth. In practice, the inputs to the system are quite a bit more complex. At least initially, it requires additional knowledge on craniofacial growth and development that can only be provided by an expert in the field, because reliable craniofacial growth data only exists for the landmarks. For these reasons, our system is initially targeted to users with prior knowledge of craniofacial growth and development, such as forensic artists and researchers in craniofacial reconstruction. Figure 7 provides a general overview of the steps involved in using our software.

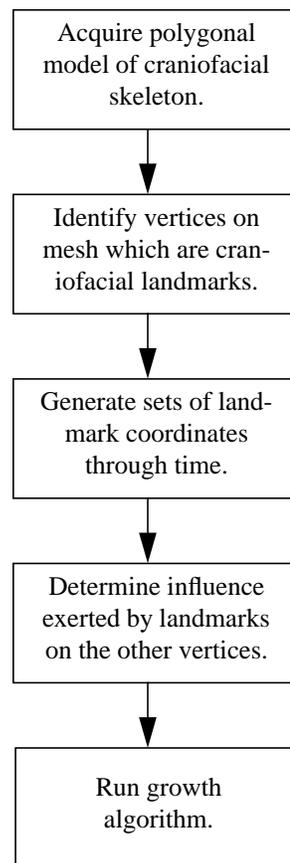


FIGURE 7. Simplified view of the workflow involved in using our system.

3.2 Inputs

Inputs to our software are provided through a specifications file, the format of which is detailed in Section 5.4.3. Each of the following sections examines one of these inputs in detail, and describes how they can be acquired by users of our system.

3.2.1 Triangular mesh

Our system requires a polygonal model of the bone to be grown (e.g. skull, mandible), specifically a triangular mesh defined in the Open Inventor 3D Interchange File Format [29]. In the case of forensic reconstruction where skeletal remains are available, a typical first step would be

to perform a laser range scan of the bone, for example with a Cyberware 3D Color Digitizer™. Other data sources include the extraction of the skull from CT scans, as described by Koch *et al* [17].

Currently, our system computes growth for a single bone. In other words, our biological model does not include the effect of displacement as defined in Section 2.2.2. Therefore, the user can choose between two approaches: either generate a triangular mesh for the entire craniofacial skeleton, or generate triangular meshes for both the skull and the mandible. In the latter case, the growth of the two meshes must be computed separately, i.e. as two completely separate operations. The two resulting meshes could then be merged by using off-the-shelf three-dimensional modelling software such as *Alias*.

Another issue is the number of triangles in the mesh. Laser range scans can typically produce meshes containing hundreds of thousands of vertices. Obviously, there is a trade-off between precision and usability. Overly complex meshes can require large amounts of storage space, and incur significant computational costs when performing both the growth of the mesh and when interactively viewing the results. There is an increasingly large body of literature on the subject of mesh optimization and decimation, which includes work by Eck *et al.* [8], Schroeder *et al.* [26] and Turk [28]. The meshes that were used for testing our system have on the order of 3000 vertices and 6000 triangles, an example of which is shown in Figure 8.

3.2.2 Starting age, target age and number of increments

The “starting age” refers to the age of the bone from which the triangular mesh was generated. In the case of reconstruction from skeletal remains, this will be the best estimate from an expert in the field of osteology.

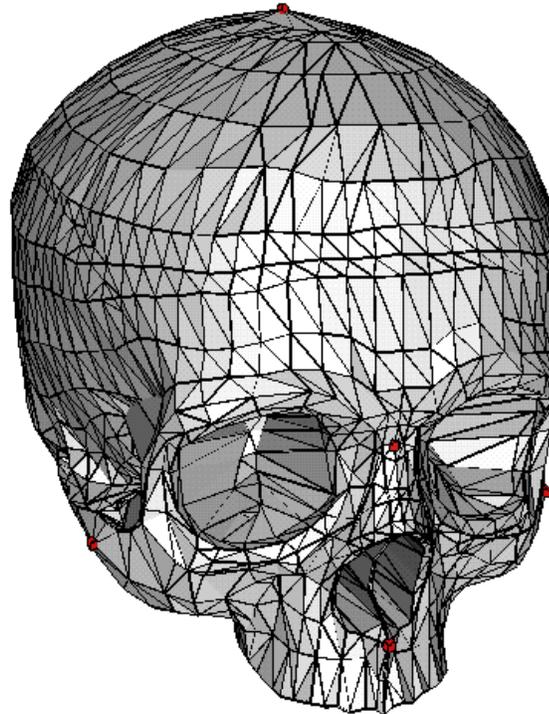


FIGURE 8. Triangular mesh of a skull, with landmarks displayed as red cubes.

The “target age” refers to the age to which the bone will be grown. It can be either greater than or lesser than the start age, depending on the situation. As an example of the former, consider the case of a search for a missing child where source data exists for the child at age t (e.g. cephalograms, CT scans). The target age would then be $t+n$, where n is the amount of time elapsed since the source data was acquired. Alternatively, consider a case where skeletal remains are discovered, and determined to be of age t . Further assume that a child went missing n years ago in the area of the discovery of the remains. The target age would then be $t-n$.

The “number of increments” allows for the generation of meshes at intermediate stages of growth from the “start age” to the “target age”. As an example, consider the following inputs:

start age of 7, target age of 15, with 4 increments. The output would then be four triangular meshes representing the bone at ages 9, 11, 13 and 15. This feature can be used to produce animations of the growth of the craniofacial skeleton.

3.2.3 Set of landmarks

The key additional information is a set of landmarks on the input mesh, corresponding to craniofacial landmarks such as those described in Section 2.3. The landmarks must be vertices on the triangular mesh, i.e. not some arbitrary point on the surface. Thus, the person who generates the mesh must ensure that there is suitable refinement in the appropriate areas of the mesh. Figure 8 shows an example of such a set of landmarks.

3.2.4 Coordinates of landmarks through time

Let's consider a growth forwards in time. The coordinates of the landmarks in the initial triangular mesh are assumed to correspond to the coordinates at the starting age. Then, there has to be at least one more set of coordinates for the landmarks, corresponding to an age greater or equal to the target age. If necessary, our system will generate a set of coordinates for the landmarks exactly at the target age, by using simple linear interpolation. For greater control on the growth simulation, the user can add any number of additional sets of landmark coordinates at any point in time between the start age and target age.

There does not need to exist a one-to-one correspondence between the number of sets of coordinates and the number of growth increments. For instance, in the example given in Section 3.2.2 we could specify the sets of landmark coordinates for age 8, 11, and 17. Our system would then use linear interpolation to generate the missing sets of landmark coordinates, namely for the ages of 9, 13 and 15.

Admittedly, the generation of these sets of landmark coordinates at various ages is a time-consuming process, and currently can only be done off-line. Appendix A describes a technique for computing the three-dimensional coordinates of landmarks through time for a specific craniofacial skeleton, by using data from two-dimensional frontal and lateral cephalograms that can be found in the Saksena studies [24] [25]. The main reason behind our approach of using craniofacial landmarks to drive the growth of the rest of the skeleton is that growth data for landmarks is readily available. In contrast, there currently does not exist three-dimensional growth data for the entire craniofacial skeleton of individuals (e.g. extracted from CT scans of the same person at a series of different ages).

3.2.5 Influence of set of landmarks

The coordinates through time of the landmarks are insufficient to generate the new meshes because the number of landmarks is so small compared to the number of vertices in the triangular mesh (typically less than 1% ratio of landmarks to vertices). As such, we require an expert in craniofacial growth and development to provide an estimation of the influence that each landmark has on each of the non-landmark vertices.

3.3 The Simplified Growth Model

As mentioned in the previous chapter, facial growth can be broken down into two components: remodelling and displacement. It would therefore make sense to divide our mathematical model into these two components. However, we concentrate in this thesis project on the remodelling aspect as it pertains to a single bone, chiefly because of the type of input data that we currently have (see Section 3.2), and also due to time constraints on the project.

Consider the naive interpretation of bone growth, where bone grows simply by uniform deposition of bone tissue on outside surfaces, i.e. bone grows by outward *expansion* of all surfaces. A

mathematical model for this growth conceptualization is relatively straightforward — for each point on the surface, perform the following:

- (1) Determine the direction of growth as being perpendicular to the surface at that point.
- (2) Assign a magnitude of growth at each point.

This grossly oversimplified model was used in the early stages of software development, in order to properly test the underlying software foundation and as a proof-of-concept of our system. It is still available as an option in the current release, more as a curiosity than anything. It also served as a stepping stone towards our finalized model, explained below.

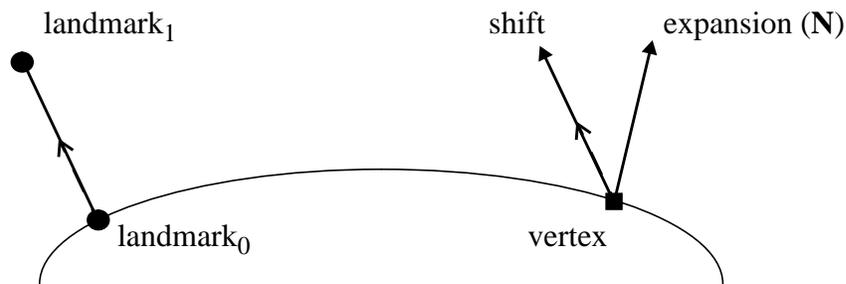


FIGURE 9. Direction of *shift* and *expansion* components.

A better model of the remodelling process uses the *relocation* concept. Consider the example in Figure 4 of Section 2.2.1. If we examine only the contour of the mandible, we can realistically break down the mandible's growth into two components: first, the mandible has increased in size, which we'll call the *expansion* term. Second, the mandible has been relocated to the upper-right, which we'll call the *shift* term. In other words, we can approximate the growth of the mandible by successively applying the *expansion* term and the *shift* term to each point on the surface. Figure 9 gives a graphical view of the two growth components for a given vertex, assuming that the vertex is influenced by a single landmark. For the *expansion* component, the vertex moves in

the direction of the surface normal defined at its position. For the *shift* component, the vertex moves in a direction parallel to the direction of growth of the landmark. If the vertex is influenced by more than one landmark, its growth is the sum of the shift and expansion components computed for each landmark. Figure 10 gives a simple example of applying our growth model to a square. Note that the *expansion* and *shift* terms can vary from point to point, i.e. there doesn't necessarily have to be a uniform expansion or uniform shift of the points.

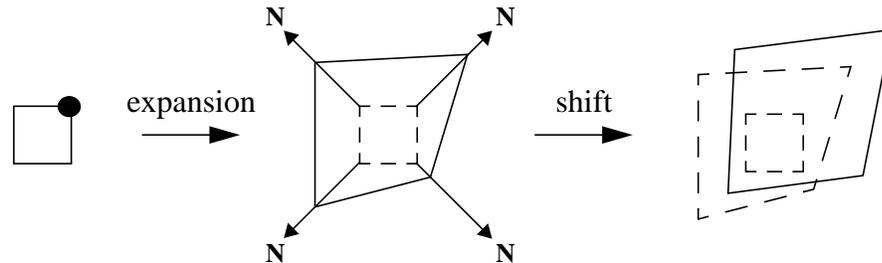


FIGURE 10. Applying the *shift* and *expansion* terms.
The filled circle identifies a landmark.

The reader may have noticed that there are a great many different combinations of *shift* and *expansion* terms that will produce the same shape, just as there many different combinations of remodelling and displacement that can produce the same growth in bones (see Section 2.2.3). This is where the experts in craniofacial growth prediction (e.g. forensic artists) make their appearance. For each vertex that is not a landmark, the expert assigns the amount of influence exerted by the landmarks. Each landmark's influence is given as two values in the range [0..1], one for the shift term and one for the expansion term. Subsequent chapters will go into detail as to how these influences are assigned to the vertices. For now, let us consider an example in two dimensions, where the object that we want to grow is a circle, and for which we have a single

landmark. Figure 9 shows the growth that results from varying the amount of influence that the landmark exerts on the points of the circle, where the circle represents the location of bone.

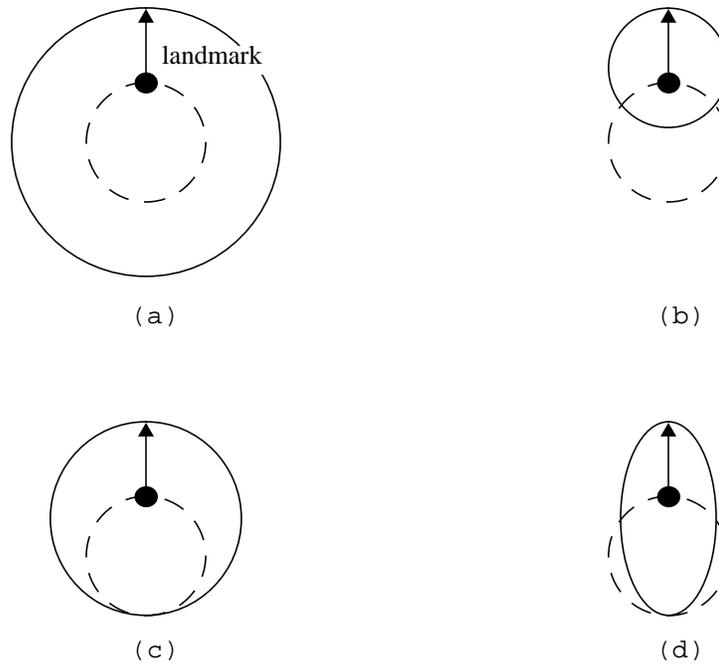


FIGURE 11. Effect of landmark influence on the growth of a circle.
(a) full influence on all vertices for *expansion*;
(b) full influence on all vertices for *shift*;
(c) full influence at landmark for *expansion*, tapering down to zero;
(d) full influence at landmark for *shift*, tapering down to zero.

4.1 The Growth Algorithms

The following explains the algorithms used for computing the growth of a bone, along with a description of issues both solved and outstanding at each step. Two different methods were used: growth along normals only, and growth using *shift* and *expansion* terms. They differ significantly in the way they compute the influence of landmarks on vertices, and compute the new coordinates of vertices. The flowcharts for these two methods are shown in Figure 12. Growth along normals only corresponds to the naive model of bone growth as described in Section 3.3, where bone grows only by adding bony tissue to the outside of all surfaces. Growth using *shift* and *expansion* terms is our attempt to include the relocation concept of bone growth.

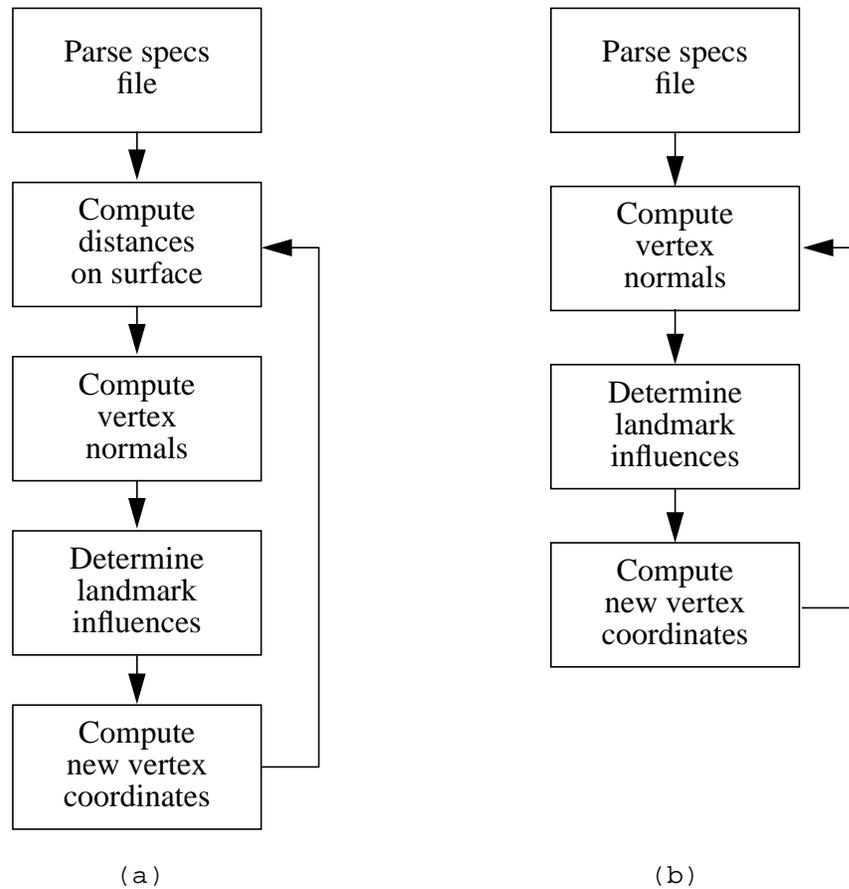


FIGURE 12. Flowcharts for the growth algorithms.
 (a) growth along normals only;
 (b) growth using *shift* and *expansion* terms.

4.1.1 Growth Along Normals Only

Step 1: Parse the specifications file

The specifications file contains user-defined information specific to the particular growth problem at hand. This includes all the input data as specified in Section 3.2, and the names to assign to the output files. The main internal data structure of the software is a parameterized graph structure. Given the file that defines the triangular mesh for the bone to be grown, we create a

graph where the nodes correspond to vertices in the mesh, and where the edges correspond to triangle edges in the mesh. The nodes that correspond to landmark vertices are marked as such. Section 5.4.3 contains a detailed description of the contents of the specifications file.

Step 2: Compute approximate distance on surface between landmarks and vertices

For each landmark vertex, we need to compute the surface distance to all the other vertices. This will later be used in step 4 to compute the amount of influence exerted by landmarks on vertices. The surface distance is approximated by performing Dijkstra’s single-source shortest-paths algorithm [7] on each landmark, where the edge weights are the Euclidean distance between the vertices of the edge. This algorithm finds the shortest path from a given source vertex (i.e. a landmark) to every other vertex. The length of these paths is used as an approximation of the surface distance.

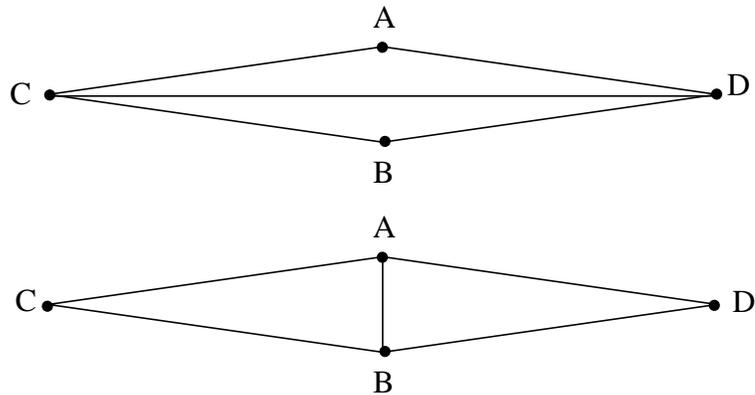


FIGURE 13. Two valid ways for triangulating a quadrilateral.
For approximating the surface distance, the bottom one is best.

This approximation for surface distance will vary depending on the triangulation of the mesh. For example, consider the two possible triangulations shown in Figure 13. In the topmost triangulation, the distance between C and D is exact, but the distance between A and B will be less

well approximated, since it will be computed as the distance from A to (C or D) to B . It should be clear that the approximation works much better with the bottommost triangulation. Here, the distance between A and B is exact. The distance between C and D , approximated by the distance from C to (A or B) to D , is relatively good. Performing a Delaunay triangulation of the mesh would ensure that the bottommost triangulation is chosen, because this procedure maximizes the minimum angle in the mesh. For now, the generation and optimization of the triangular mesh is left to those constructing the input meshes.

It would also be possible to use a better approximation of surface distance, using algorithms such as those given by Chen and Han [6]. However, the surface distance is not needed in the model of growth where *shift* and *expansion* terms are used, since the influence of the landmarks is assigned directly by the user. Therefore, we decided that it was not worth the time and effort needed to implement more robust algorithms for surface distance.

Step 3: Compute vertex normals

We compute the surface normal at a vertex by first summing the normal vectors of all faces adjacent to the vertex, and then normalizing this sum.

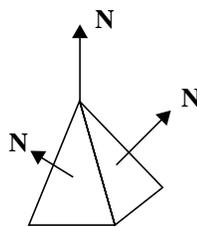


FIGURE 14. Computing the surface normal at a vertex.

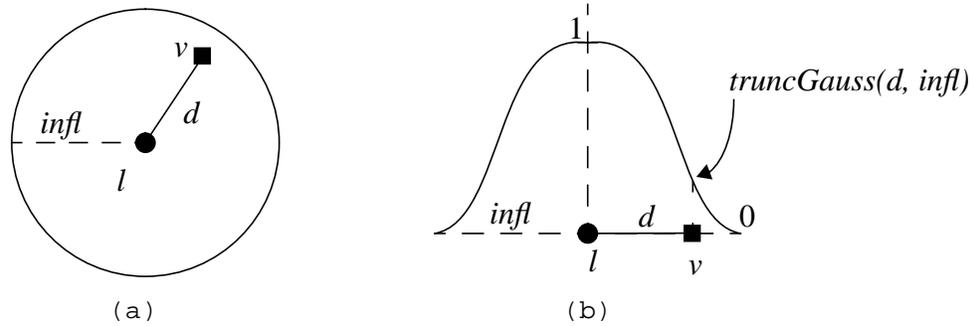


FIGURE 15. Vertex v lies within the area of influence of landmark l , as shown in (a). To determine the landmark's influence, use a truncated Gaussian function, as in (b).

Step 4: Determine influence of each landmark on individual vertices

One of the key elements in the growth algorithm is determining, for a particular vertex, which landmarks influence its growth and to what extent. Here, we assign to each landmark a *radius of influence*. If the distance d between a vertex v and a landmark l is less than the radius of influence of l , then we say that v lies within the area of influence of l . Note that the distance between v and l is computed in step 2. Finally, a truncated Gaussian function centered on the landmark is used to determine the amount of influence that l has on v , as defined in Equation 1. The value of the function at the landmark is 1, and the value on the perimeter of the area of influence (i.e. at the radius of influence) is 0.

$$\text{truncGauss}(d, \text{infl}) = \frac{e^{-\left(\frac{d}{\text{infl}}\right)^2} - \frac{1}{e}}{1 - \frac{1}{e}} \quad (\text{EQ 1})$$

Step 5: Compute new coordinates for all vertices

First, a note on how we quantify the expansion growth of each landmark. Consider a specific landmark l , where $l(t)$ denotes the position in space of l at time t , and where Δl is the vector from $l(t)$ to $l(t+\Delta t)$. The expansion growth vector $\mathbf{g}_e(l)$ is approximated by projecting Δl onto the surface normal at $l(t)$.

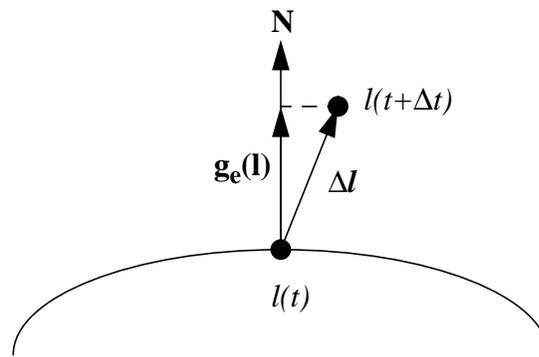


FIGURE 16. Computing the expansion growth vector $\mathbf{g}_e(l)$ of a landmark.

If a vertex lies within the area of influence of more than one landmark, then the influences of the landmarks are summed (Figure 17). Landmarks have no influence on other landmarks, since the coordinates through time of all the landmarks are given as input to the algorithm (see Section 3.2.4). The pseudo-code in Figure 18 describes how the new coordinates of a non-landmark vertex are computed. \mathbf{N}_v is the surface normal at vertex v , as computed in step 3.

Step 6: Repeat from step 2

Remember that one of the input data is the “number of increments”, i.e. the number of meshes to generate (see Section 3.2). Thus, we repeat steps 2 through 5 for “number of increments”.

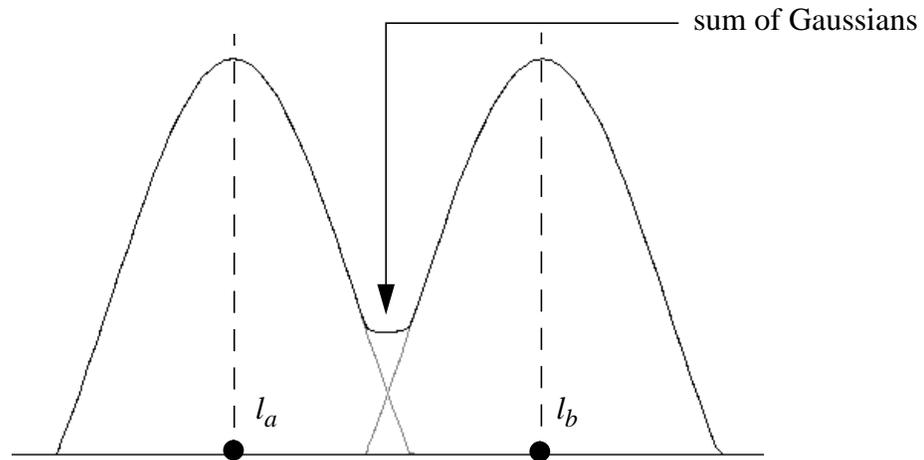


FIGURE 17. Overlapping, truncated Gaussian functions.

```

forall_vertices( v, graph ), not landmarks {
    sumInfluence = 0.0;
    forall_landmarks( l, graph ) {
        if( distance(v, l) <= l.influence ) {
            sumInfluence += truncGauss(v, l) * ||g_e(l)||;
        }
    }
    v.newcoords = v.oldcoords + sumInfluence * N_v;
}

```

FIGURE 18. Pseudo-code for computing new coordinates of non-landmark vertices.

4.1.2 Growth Using Shift and Expansion Vectors

This model uses on the relocation concept for bone growth, as described in Section 2.2.1. The modifications to the previous method are twofold: firstly, to include a means by which vertices are “pulled” in the direction of growth of the landmarks — this is the *shift* term. Secondly, instead of Gaussian functions, the new method uses more directly an expert’s knowledge in determining both the area of influence of landmarks and the amount of influence exerted on specific vertices.

Step 1: Parse the specifications file

Same as in Step 1 of Section 4.1.1.

Step 2: Compute vertex normals

Same as in Step 2 of Section 4.1.1.

Step 3: Determine influence of each landmark on individual vertices

In order to properly describe the amount of influence that landmarks have on a specific vertex, we must assign to the vertex two values for each landmark: the *expansion weight* and the *shift weight* for that landmark. These values are in the range $[0..1]$, where 0 indicates that the landmark has no influence on the vertex. For the remainder of this thesis, the term “weight of a vertex” refers to the set of tuples (expansion weight and shift weight) assigned to the vertex, one for each landmark.

These weights must be assigned to the vertices of the mesh prior to the actual computation of growth. This process can be quite time-consuming: for example, in a typical mesh with 5 landmarks and 5000 vertices, a total of 50000 values must be assigned! To reduce the effort, a specialized software package called *Krayola* was created that is used to interactively “paint” the vertices of a triangulated mesh, where the colours correspond to the amount of influence of the landmarks. Painting is done in two separate layers, one for the shift weights and one for the expansion weights. In essence, the expert uses *Krayola* to define the area of influence of the landmarks. See Section 5.1.2 for more details on *Krayola*.

Selecting appropriate values for the vertex weights is not an easy task. Therefore, we provide an extra mechanism for “tweaking” the global influence of a landmark. Each landmark has a coefficient of expansion K_e and a coefficient of shift K_s . A value of 1 is the default. Lowering these coefficients has a dampening effect on the influence of the landmark.

Step 4: Compute new coordinates for all vertices

The growth of a vertex is computed by summing two vectors: the *expansion vector* and the *shift vector*. The expansion vector corresponds to growth in the direction of the surface normal at the vertex (the vertex normals are computed in step 2). The magnitude of the expansion vector is basically the same as in the previous method, the main difference being that instead of using a truncated Gaussian function, we use the expansion weight as defined in step 3. The shift vector of the vertex is parallel to the actual growth vector of the landmark (see Figure 9 in the previous chapter). If the vertex is influenced by more than one landmark, then the sum of the shift and expansion vectors for all the landmarks is used. The decision to use the sum of the effect of landmarks rather than a scheme based on weighted averages was somewhat arbitrarily based on our perception of the way someone would use our system. The correspondence between modifications to vertex weights in *Krayola* and resulting changes in the output mesh is thus more intuitive. This is especially important because our growth model relies heavily on the ability of an expert to properly assign the vertex weights.

The following equation describes how to compute the new coordinates of a vertex:

$$v_{new} = v_{old} + \sum_l (N_v \cdot \|g_e(l)\| \cdot K_e(l) \cdot E(v, l) + \Delta l \cdot K_s(l) \cdot S(v, l)) \quad (\text{EQ 2})$$

Where:

- v_{new} are the new (x, y, z) coordinates of v
- v_{old} are the old (x, y, z) coordinates of v
- N_v is the normalized surface normal at v
- $g_e(l)$ is the expansion growth vector as defined in Figure 16
- $E(v, l)$ is vertex v 's expansion weight, for landmark l
- $S(v, l)$ is vertex v 's shift weight, for landmark l
- $K_e(l)$ is the coefficient of expansion of landmark l
- $K_s(l)$ is the coefficient of shift of landmark l
- Δl is the growth vector of l and defined in Figure 16

4.1.3 The Problem With Iteration

An outstanding issue to be addressed in future work is the effect of iteration on the growth algorithm. Given that the vertex normals are recomputed at each iteration, the growth of the bone is intrinsically linked to the number of increments between the starting age and the target age, i.e. the number of triangular meshes generated by the software. Consider the situation depicted in Figure 19. The starting mesh is a simple segment with three vertices. There is one landmark, whose starting position is at l_0 and whose end position is at l_1 . Only vertex v is influenced by the landmark; it grows half as much as the landmark, and only along the surface normal at v (both of our growth algorithms support this type of growth). Clearly, the mesh that results from only one iteration is different than the one that results from four iterations.

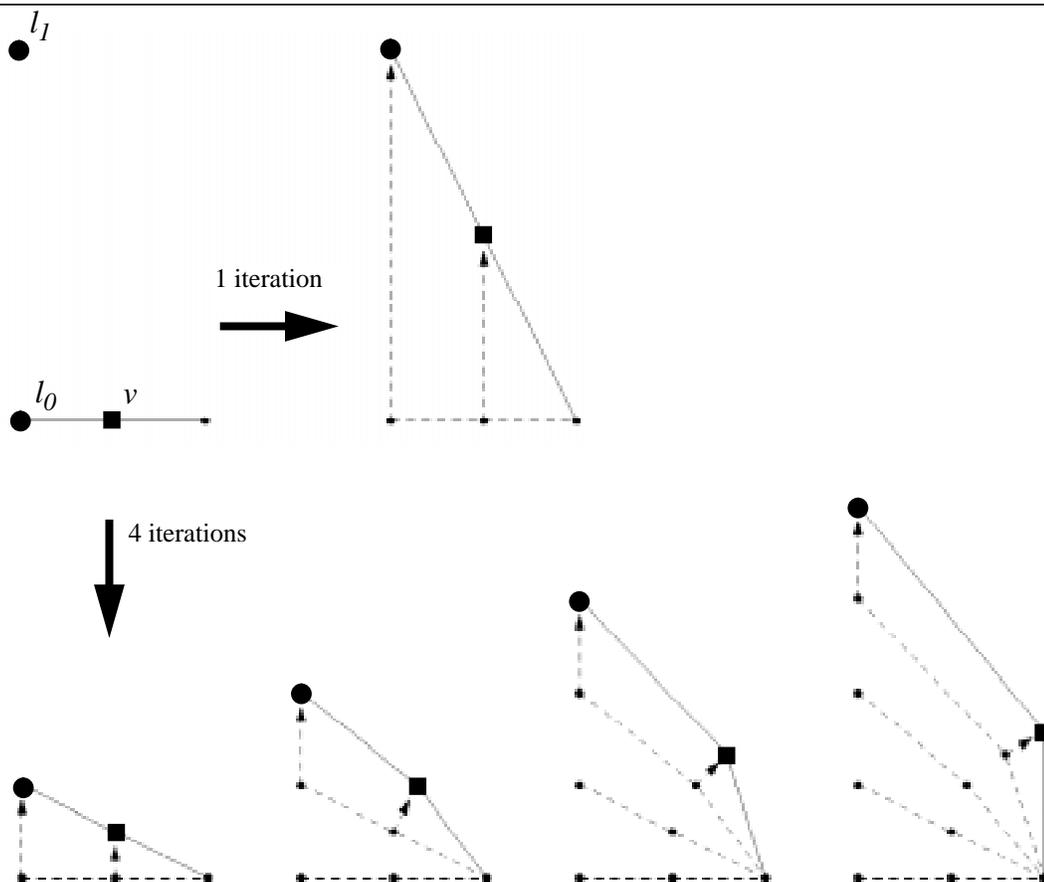


FIGURE 19. The effect of iteration of our growth algorithm. The landmark starts at l_0 and grows to l_1 . Vertex v grows half as much as the landmark.

4.2 Automating the Generation of Weights

Assigning weights to vertices by painting them, while better than directly editing the text file that contains the weights, is still a very tedious process. Furthermore, without automatic assistance the procedure has to be repeated for every new mesh, even if the mesh is of the same bone type (e.g. skull, mandible) as that of a previously painted mesh. This is due to the fact that very rarely will two meshes contain exactly the same vertices, in exactly the same order, especially if they were generated completely separately, or if the meshes correspond to bones of different individuals. A useful test of the system would be to compare the growth results for a bone at various levels of detail, i.e. by adding or removing vertices in critical areas such as the condyle and mandibular ramus in the skull. An analysis could determine whether or not the increased precision gained by adding vertices is sufficient to offset the added complexity and computation time.

transferWeights is a stand-alone software program that semi-automatically generates the weights for the vertices of a mesh, given the weights for the vertices of a second mesh. The procedure is justified by the fact that the growth patterns of a bone (i.e. the fields of growth) do not vary greatly from person to person, barring fairly substantial abnormalities.

Step 1: Align the two meshes

The source mesh (for which we have vertex weights) and the target mesh (for which we want to generate vertex weights) must first be aligned as closely as possible. Only the landmark vertices are considered, since they are the only vertices for which there is a reliable correspondence between the two meshes. Furthermore, the method only works under the restriction that both meshes have the same number of landmarks, and that there exists a one-to-one correspondence between the landmarks of each mesh. The algorithm described by Jang *et al.* [15] is used to compute a matrix for translation and rotation that most closely maps the first set of points (i.e. the landmarks on the target mesh) to the second set of points (i.e. the landmarks on the mesh for which

the weights are known). This matrix is then applied to all the vertices of the first mesh, effectively placing the meshes “one of top of the other”.

Step 2: Compute the geometric center of the target mesh

The geometric center of the target mesh is computed by first summing the coordinates of all the vertices of the mesh, and then dividing by the number of vertices in the mesh.

Step 3: Align the two meshes

For each vertex in the target mesh, a ray is cast from the geometric center of the target mesh through the vertex. A ray-triangle intersection algorithm is used sequentially on the triangles of the source mesh in order to determine which of these triangles is intersected by the ray. The weight at the intersection point is computed by interpolating between the weights of the vertices of the intersected triangle. This weight is then assigned to the vertex in the target mesh through which the ray was cast. This procedure is repeated for each vertex in the target mesh.

This algorithm for approximating vertex weights blatantly ignores problem areas. For instance, it is possible for a ray to intersect more than one triangle in the source mesh, since the meshes are usually concave. Currently, only the first “hit” is registered. A better approach might be to only consider the intersected polygon in the source mesh that minimizes the distance to the vertex in the target mesh. However, it should be clear that any automatic generation of the weights is bound to be imperfect, and will require some manual adjustment of the weights by using the *Krayola* vertex painting program (described in Section 5.1.2). For now, our qualitative examination of the meshes generated by *transferWeights* indicates that it is “good enough”.

Section 5.2 contains a more detailed look at some of the algorithms used in the individual components of *transferWeights*.

The flowchart in Figure 20 contains all the steps involved in using our software system.

5.1 Geomview

From the Geomview manual [11], “Geomview is an interactive program for viewing and manipulating geometric objects, written by staff members of the Geometry Center” at the University of Minnesota. “It can be used as a standalone viewer for static objects or as a display engine for other programs which produce dynamically changing geometry. It allows for standard direct manipulation of 3D objects (rotation, translation, scaling, etc.) and for control of appearances such as lighting, materials, and shading.” Geomview particularly shines in its support of *external modules*. Again, from the Geomview manual:

“An external module is a program that interacts with Geomview. When Geomview invokes an external module, it creates pipes connected to the module’s standard input and output. [...] Geomview interprets anything that

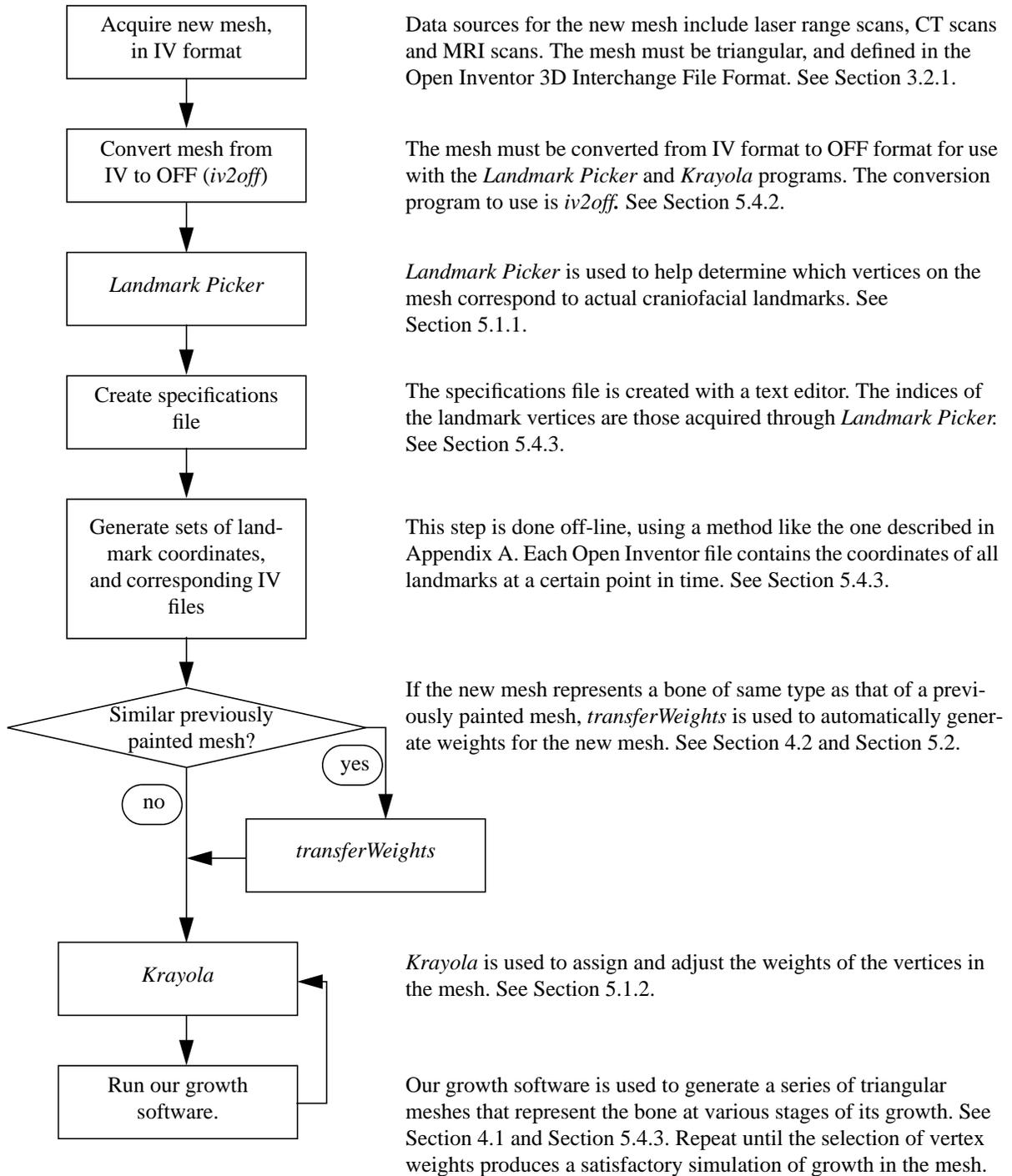


FIGURE 20. Detailed workflow involved in using our system.

the module writes to its standard output as a Geomview Command Language (gcl) command. Likewise, if the external module requests any data from Geomview, Geomview writes that data to the module's standard input. Thus all a module has to do in order to communicate with Geomview is write commands to standard output and (optionally) receive data on standard input.

Two external modules to Geomview were created specifically for use in our project, to provide better user interaction with the growth software:

- *Landmark Picker* allows the user to interactively select vertices on a polygonal mesh. It is used to identify the vertices that most closely correspond to actual craniofacial landmarks.
- *Krayola* is used to “paint” weights onto the vertices, i.e. to assign to a vertex the amount of influence that a particular landmark has on its growth.

5.1.1 External module - *Landmark Picker*

One of the first tasks is for the user to determine which vertices on the triangular mesh correspond to craniofacial landmarks. To facilitate this process, the *Landmark Picker* external module was created. The steps involved in using this module are described as follows:

Step 1: Start up Geomview, usually by typing **geomview** at the UNIX command prompt.

Step 2: In Geomview, load the OFF file which contains the triangular mesh (see Section 5.4.2 for information on the OFF file format).

Step 3: Use the **Appearance** control panel to customize Geomview's environment. Turning on *Edges* and selecting *Smooth Shading* greatly improves the quality of the image.

Step 4: Invoke the *Landmark Picker* external module.

Step 5: Pick the landmarks, as explained below.

In order to tentatively pick a landmark, the user right-clicks on the appropriate vertex. This vertex is overlain with a purple cube, to aid in gauging whether or not the vertex is

the one that most closely corresponds to an actual craniofacial landmark. Right-clicking on a different vertex moves the purple cube to the newly selected vertex. When satisfied with the selection of a vertex, the user right-clicks on the purple cube, which marks the vertex as being a landmark. The cube becomes green and is permanently attached to the landmark vertex, and the index of the vertex is printed to the console window. Other landmarks can then be identified.

Step 6: Modify the specifications file.

A text editor is used to copy to the specifications file the indices of the landmark vertices identified in the previous step (the indices displayed in the console window). See Section 5.4.3 for a detailed look at the specifications file.

5.1.2 External module - *Krayola*

Krayola is a modified version of *Crayola*, an external polygon painting module that is included in the Geomview distribution. The unmodified version is used to interactively add and modify colours of an OOGL object (see Section 5.4.2 for a description of OOGL objects). The modified version, *Krayola*, is used to paint “weights” on vertices. These weights are a series of tuples which indicate the amount of influence that a landmark has on a vertex; each vertex receives a separate tuple for each landmark. The file that is generated by *Krayola* is used when computing growth using shift and expansion vectors. The following modifications were made to the control panel of *Crayola* (see Figure 21).

- In *Crayola*, the user chooses a colour by manipulating a colour wheel and sliders for intensity and opacity. In *Krayola*, colours that lie on the white-red axis only are used, specifically by using the **weight** slider. The weight slider’s values are in the range [0..1], with 0 corresponding to white, and 1 to red. The colours are used strictly as a means for assigning *shift* and *expansion* weights to vertices, with 0 corresponding to zero influence. For the remainder of this section, “colour” refers to the value chosen in the **weight** slider. Note that the

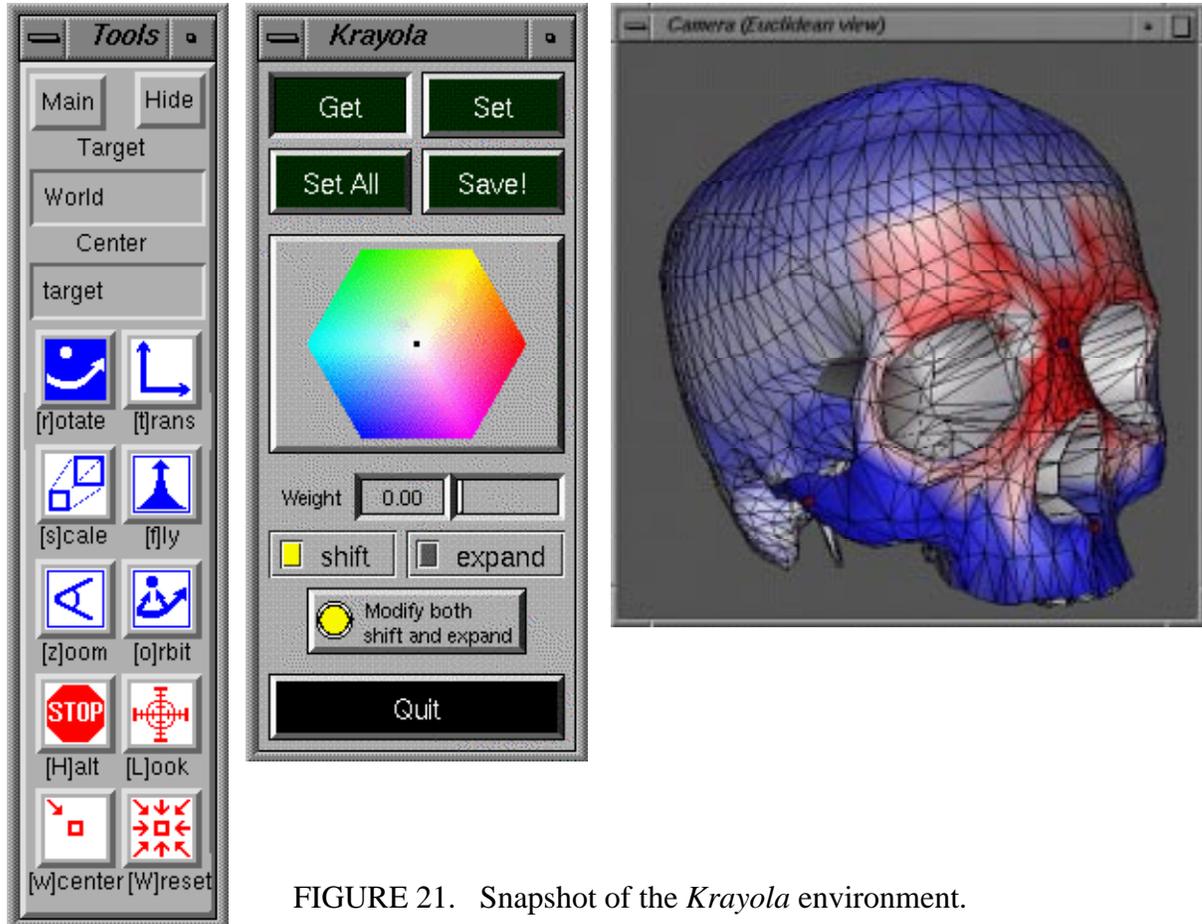


FIGURE 21. Snapshot of the *Krayola* environment.

colour wheel is still present, but only as a visual cue for the chosen colour. The sliders for intensity and opacity have been removed.

- Two radio buttons were added: **shift** and **expand**. These allow the user to choose between viewing the colours corresponding to the *shift* term or those corresponding to the *expansion* term.
- A toggle button labelled **Modify both shift and expand** has been added. When it is “on”, any changes made to the colour of the vertices affect both the *shift* and *expand* terms. When it is “off”, changes only affect one of the terms, as determined by the **shift** and **expand** radio buttons.
- The **Undo** and **Eliminate Color** buttons have been removed.

The following steps are involved in using *Krayola* to generate the weights file.

Step 1: Create a specifications file (see Section 5.4.3 for details on this file).

Two fields of the specifications file are used: `WEIGHTS_SHIFTandEXPAND` (*sic*), which contains the name of the weights file to be created or modified, and `LANDMARKS_INDICES`, which contains the vertex indices of the landmarks as defined in *Landmark Picker*. The other fields are ignored. Note that the specifications file must be named “crayola.specs” (it’s hardwired - don’t ask).

Step 2: Start up Geomview, usually done by typing **geomview** at the UNIX command prompt.

Step 3: In Geomview, load the OFF file which contains the triangular mesh (see Section 5.4.2 for information on the OFF file format).

Step 4: Use the **Appearance** control panel to customize Geomview’s environment. Turning on *Edges* and selecting *Smooth Shading* greatly improves the quality of the image.

Step 5: Invoke the *Krayola* external module. Note that in order to “kick-start” the display of colours, the user should start by selecting a landmark vertex.

Cubes centered on vertices are used as visual cues, to identify the landmarks. The currently selected landmark’s cube is blue, while the other ones are red. To select a landmark, the user right-clicks on the appropriate cube. All vertices that are influenced by the selected landmark (i.e. those vertices that have non-zero values for either the *shift* or *expansion* term) are coloured using different saturations of red. Vertices that are not influenced by the selected landmark, but are influenced by other landmarks are coloured using different saturations of blue. Admittedly, this visual representation does a poor job of displaying vertices that are affected by more than one landmark, but it works reasonably well in practice.

Step 6: Colour the vertices.

The **Set** and **Get** buttons in the control panel represent modes of operation. When the **Get** button is turned on, right-clicking on a vertex sets the current colour to be the colour of that vertex. To modify the colour of a vertex, the user first clicks on the **Set** button, and uses the **Weight** slider to select a colour. The user then right-clicks on the intended vertex. Alternatively, the user can right-click inside a triangle, which assigns the current colour to the three vertices of that triangle. This colour represents the influence that only the currently selected landmark exerts on the vertex or vertices. To select a different landmark, the user right-clicks on the landmark.

Step 7: Save the weights file.

When satisfied with the colour of the vertices, the user clicks on the **Save** button. For instance, if the `WEIGHTS_SHIFTandEXPAND` parameter in the specifications file is “skull.weights”, then the file “skull.weights.new” is created. Colours will automatically be loaded from the new weights file at the next session.

5.2 Automatic generation of 1st guess at weights

transferWeights is a shell script used to facilitate the process of generating a first approximation for the weights of a triangular mesh, given the weights of another mesh (see Section 4.2 for situations where this is useful). It takes two command-line parameters: the specifications file of the mesh for which we know the weights, and the specifications file of the mesh for which we want to approximate the weights. Figure 22 contains the abbreviated program listing for *transferWeights*. Four stand-alone programs are invoked sequentially: *strip*, *procruste*, *apply*, and *skullgrowth*. These are described below.

```
# targIV is the name of the BASE_OBJECT .iv file, extracted from the
specifications file. targWEIGHTS is the name of the WEIGHTS_SHIFTandEXPAND
weightsfile, extracted from same. The extractions are done with awk files.

targIV=`nawk -f getIVfile.awk < $2`
targWEIGHTS=`nawk -f getWEIGHTSfile.awk < $2`

# create tmp00000.iv, which contains targIV * matrix,
# where matrix = the transformation matrix computed by procruste.
# Note that strip extracts from the two specifications file the 3D coords
# of the landmarks, and outputs them in the format expected by procruste.

strip $2 $1 | procruste - m | apply $targIV tmp00000.iv

# modify the 2nd specsfile, so that the BASE_OBJECT is tmp00000.iv
# awk -f makeTmpSpecsfile.awk < $2 > tmp00000.specsfile

# generate the new weights file.

skullgrowth $1 weights tmp00000.iv $targWEIGHTS
```

FIGURE 22. Abbreviated program listing of *transferWeights*.

5.2.1 *strip*

Usage: *strip* <org.specs> <targ.specs>

This program extracts the coordinates of the landmarks from two triangular meshes defined in the Open Inventor file format. Both meshes must contain the same number of landmarks (these are extracted from their specifications file). The coordinates are written to *stdout* in the format shown in Figure 23.

```
<num vertices>
<x1 y1 z1>
...
<xn yn zn>
<X1 Y1 Z1>
...
<Xn Yn Zn>
```

FIGURE 23. Output of the *strip* program

5.2.2 *procruste*¹

Usage: *procruste* - *m*.

This program reads, from *stdin*, two sets of 3D points with a one-to-one correspondance between the points, and finds the translation and rotation matrix that best matches the corresponding data points. In other words, it calculates the best fit of two similar clouds of 3D data points. The transformation matrix is output on *stdout*.

Briefly, the algorithm works as follows. The best least squares translation is computed from the centers of mass of the two sets of data points. The rotation matrix is computed using a linearized iterative least squares algorithm. This algorithm converges on the rotation values given that suitable initial estimates are chosen. These initial estimates are computed by the solution of the “orthogonal Procrustes problem” (see Golub and Van Loan [12]). A minimum of three points is necessary to perform these calculations. Refer to Jang *et al.* [15] for a complete description of this algorithm.

1. Written by Stan Jang, at the Univeristy of British Columbia.

5.2.3 *apply*

Usage: *apply* <target.iv> <rotated.iv>

This program reads a transformation matrix from *stdin* and applies it to all the vertices in <target.iv>, which is a triangular mesh stored in the Open Inventor file format. The resulting mesh is stored in <rotated.iv>.

5.2.4 *skullgrowth*

Usage: *skullgrowth* <org.specs> *weights* <rotated.iv> <target.weights>

Section 4.2 gives a general overview of the algorithm used to approximate the weights file. Remember that by “weight” of a vertex, we mean the array of *shift* and *expansion* terms which measure the influence of landmarks on the vertex. The *skullgrowth* program, when given the *weights* parameter, performs the actual approximation. For each vertex in the target mesh, a ray is cast from the center of the mesh through the vertex. The ray-triangle intersection algorithm described by Badouel [1] is used to determine which triangle in the original mesh is intersected by the ray. Finally, the weight of the vertex is computed by interpolating between the weights of the vertices of the intersected triangle.

5.3 LEDA

One of the cornerstones of software development for this project is LEDA (Library of Efficient Data types and Algorithms), developed at the Max-Planck-Institut für Informatik in Saarbrücken, Germany. From the LEDA web page [18], “LEDA provides a sizable collection of data types and algorithms [...] efficient implementations for each of the data types”. It is implemented by a C++ class library. LEDA can be used freely for academic research and development.

Of particular use to us is the data type *Graph*. It can be parameterized, which means that additional, user-defined, data can be assigned to the graph's nodes and edges. Specifically, a parameterized graph is the internal data structure that we use to store the triangular meshes. LEDA further provides iteration macros such as “for all nodes v of a graph G do”, which greatly simplifies operations on the meshes.

5.4 File Formats

5.4.1 Open Inventor

The triangular meshes that are used as input to our system must be defined in the Open Inventor 3D Interchange File Format [29]. A further restriction is that meshes must be defined as *indexed face sets*. The meshes generated by our system are also defined using this file format.

5.4.2 OOGL and OFF

An OOGL object is an object that can be loaded into *Geomview*; it stands for “Object Oriented Graphics Library”. An OFF file (named for “Object File Format”) defines a specific type of OOGL object which is well-suited for describing triangular meshes. For a more detailed look at OOGL and OFF objects and their syntax, please refer to the *Geomview* manual [11].

In order to use *Geomview* and the accompanying *Landmark Picker* and *Krayola* external modules, the triangular mesh which serves as initial input to our system must first be converted from an Open Inventor file to an OFF file. We provide a file format converter called **iv2off** for this purpose, the usage of which is **iv2off <file.iv> <file.off>**. This converter only works for triangular meshes with no coincident vertices.

5.4.3 The Specifications File

The specifications file contains all the input necessary to our system. Figure 24 shows its syntax. Any line beginning with the “#” character denotes a comment, and is ignored. In the template (not in the actual specifications file), terms that can appear 0 or more times are shown in curly braces “{...}”, lines that can appear 1 or more times start with the “+” character, and optional lines start with the “\$” character. Figure 25 shows an example of a valid specifications file. The following list describes the expected tokens:

- **BASE_OBJECT**: the first parameter is the Open Inventor filename of the mesh for the bone to be grown. The second parameter is the age of the bone.
- **OUTPUT_BASENAME**: the prefix of the filenames of the generated meshes. The age of the generated mesh and the “.iv” suffix will be added to complete the filename.
- **WEIGHTS_SHIFTandEXPAND**: the name of the file generated by *Krayola*.
- **GROWTH_PARAMETERS**: the starting age, the target age, and the number of increments, as described in Section 3.2.

```

# Comments, comments, of course

BASE_OBJECT          <filename.for.skull.iv> <age of skull>
OUTPUT_BASENAME      <basefilename.for.growth.output>
$ WEIGHTS_SHIFTandEXPAND <filename.for.weights (from Krayola)>
GROWTH_PARAMETERS    <ageBegin> <ageEnd> <steps>
LANDMARKS_INDICES    <index> { <index> }
GROWTH_TYPE          <type> { <Ks> <Ke> }
$ LANDMARKS_INFLUENCES <influence> { <influence> }
                      (same number as
                      for LANDMARKS_INDICES
+ LANDMARKS_COORDS    <landmark.coords.iv> <age>

```

FIGURE 24. Template for the specifications file.

-
- **LANDMARKS_INDICES**: the indices of the landmark vertices in the starting triangular mesh. These are acquired by using *Landmark Picker*.
 - **GROWTH_TYPE**: depending on which growth algorithm to use, the first parameter is either “shift_and_expand” or “normals_only”. For the former, this is followed by K_s and K_e pairs, one for each landmark (see Section 4.1.2).
 - **LANDMARKS_INFLUENCES**: if the growth type is “normals_only”, then these parameters are the radius of influence of each landmark.
 - **LANDMARKS_COORDS**: the first parameter is the file which contains the positions of the landmarks at a certain age. This age is defined by the second parameter. Note that there can be an arbitrary number of these lines, each one giving the coordinates of the landmarks at a different point in time. Each file must be defined in the Open Inventor format, specifically as a *point set* [29].
-

```
# Vive l'Acadie! (A comment like any other.)

BASE_OBJECT           Data/skull.iv 5
OUTPUT_BASENAME       Output/skull.out
WEIGHTS_SHIFTandEXPAND Data/skull.weights
GROWTH_PARAMETERS     5 15 4
LANDMARKS_INDICES     1454 55 33 210 1657
#GROWTH_TYPE          normals_only
GROWTH_TYPE           shift_and_expand 0.50 0.75 0.50 0.75 0.50 0.75
0.50 0.75 0.50 0.75
LANDMARKS_COORDS      Data/skull.lm.10.iv 10
LANDMARKS_COORDS      Data/skull.lm.15.iv.15
```

FIGURE 25. Example of a valid specifications file

6.1 The Growth Algorithm

6.1.1 Simple Geometry

Figure 26 shows the results of applying our growth algorithm to a sphere, using shift and expansion vectors as defined in Section 4.1.2. There is one landmark, situated at the north pole of the sphere. Figure 26a is provided to give a general idea of the assignment of weights to the vertices. The weights are displayed using varying saturation levels of red, where full red denotes maximum influence and white denotes zero influence. We see that the maximum influence is at the landmark, and that it uniformly tapers down to zero at the other pole of the sphere. Figure 26a also shows the position of the landmark at the next time interval — the landmark grows straight up.

Figure 26b shows the result of only using the shift weights (i.e. the expansion weights are all zero). Notice how all the vertices grow in the same direction as the landmark. Figure 26c shows the result of only using the expansion weights. Notice how each vertex grows in the direction of

the surface normal *at that vertex*, not at the landmark. Figure 26d shows the result of using both the shift and expansion weights. In order to make this figure an averaging of Figure 26c and Figure 26b, the coefficients of expansion (K_e) and of shift (K_s) were changed from 1.0 to 0.5.

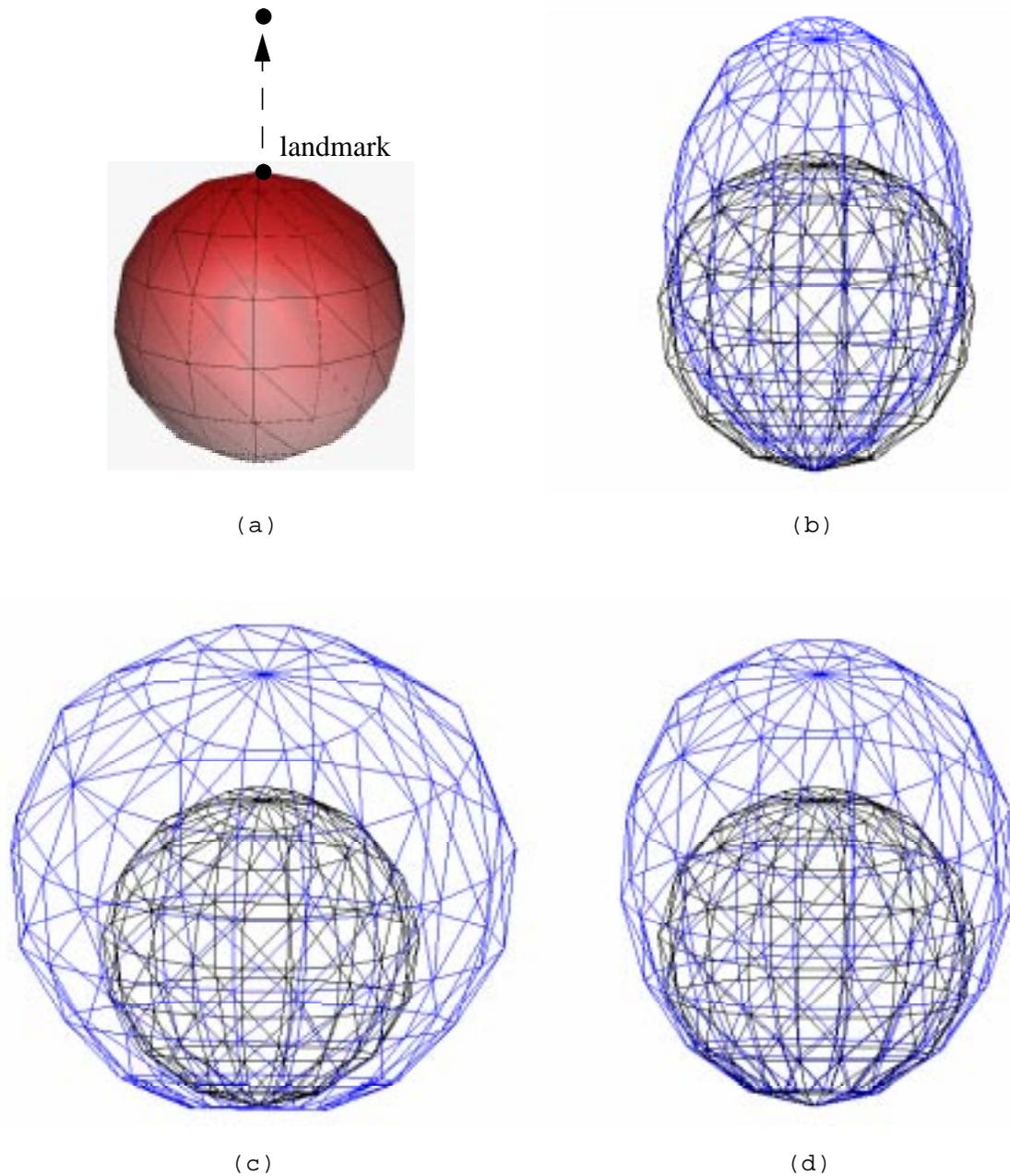


FIGURE 26. Applying our growth algorithm to a sphere. (a) assignment of vertex weights. (b) shift only. (c) expansion only. (d) shift and expansion.

6.1.2 Complex Geometries

Figure 27 shows the result of applying our growth model to a mandible. Figure 27a shows the position of two of the three landmarks used in this example, and the corresponding assignment of vertex weights. The landmark on the tip of the jaw corresponds to *Menton*, and the other visible landmark to the right is *Gonion*. The hidden landmark corresponds to the left *Gonion*. Figure 27b contains the initial triangular mesh, which was acquired from CT scans. Applying our growth algorithm to this mesh yields the mesh displayed in Figure 27c. Finally, Figure 27d provides a good idea of the growth by overlapping wireframe models of the initial and resulting mesh.

Figure 28 shows the result of applying our growth model to a skull. Figure 28a shows the position of all five landmarks, and the corresponding assignment of vertex weights. The two landmarks on the side of the skull are *Zygoma*, the landmark on top of the skull is *Apex*, the landmark on the bridge of the nose is *Nasion*, and the landmark on the bottom of the nose is *Anterior Nasal Spine*. Figure 28b contains the initial triangular mesh, which was acquired from CT scans. Applying our growth algorithm to this mesh yields the mesh displayed in Figure 28c. Figure 28d contains overlapping wireframe models of the initial and resulting mesh.

6.2 Approximating the Weights

Figure 29 shows the results of using *transferWeights* to approximate the vertex weights for a new mesh, given the weights of a similar source mesh. The source mesh, shown in Figure 29a and Figure 29c, is a triangulated sphere which contains 151 vertices and 224 triangles. The target mesh, shown in Figure 29b and Figure 29d, was generated by squashing and warping a sphere, and contains 637 vertices and 992 polygons. Both meshes have four landmarks in roughly the same positions: one at each of the poles, and two along the equator. Vertex weights for the source mesh were assigned in *Krayola*. Figure 29a and Figure 29b show the expansion weights of the vertices for the source and target meshes, while Figure 29c and Figure 29d show the shift weights.

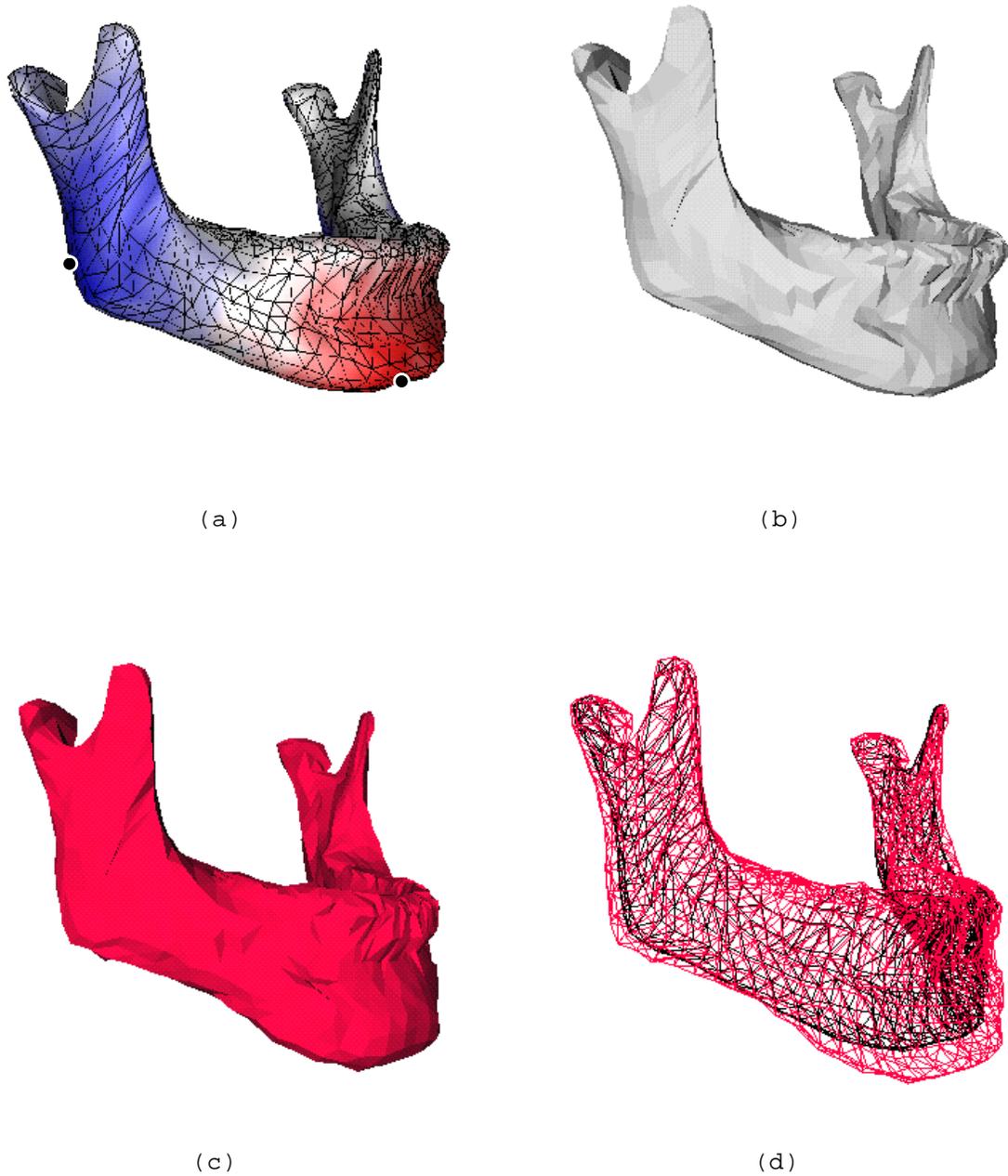


FIGURE 27. Applying our growth algorithm to a mandible. (a) assignment of vertex weights. (b) initial mesh. (c) resulting mesh. (d) initial and resulting meshes, overlapped.

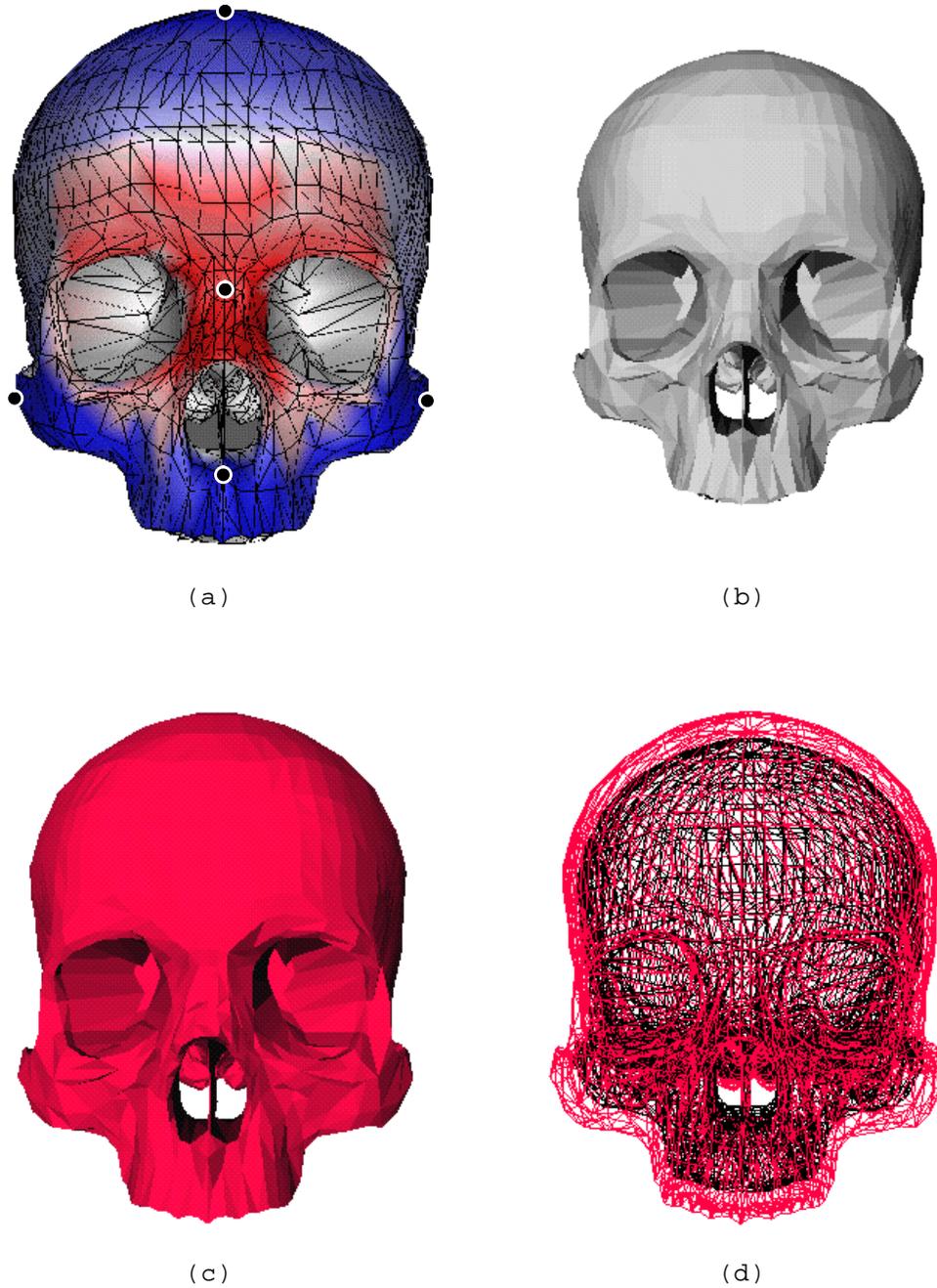


FIGURE 28. Applying our growth algorithm to a skull.
(a) assignment of vertex weights. (b) initial mesh. (c) resulting mesh. (d) initial and resulting meshes, overlapped.

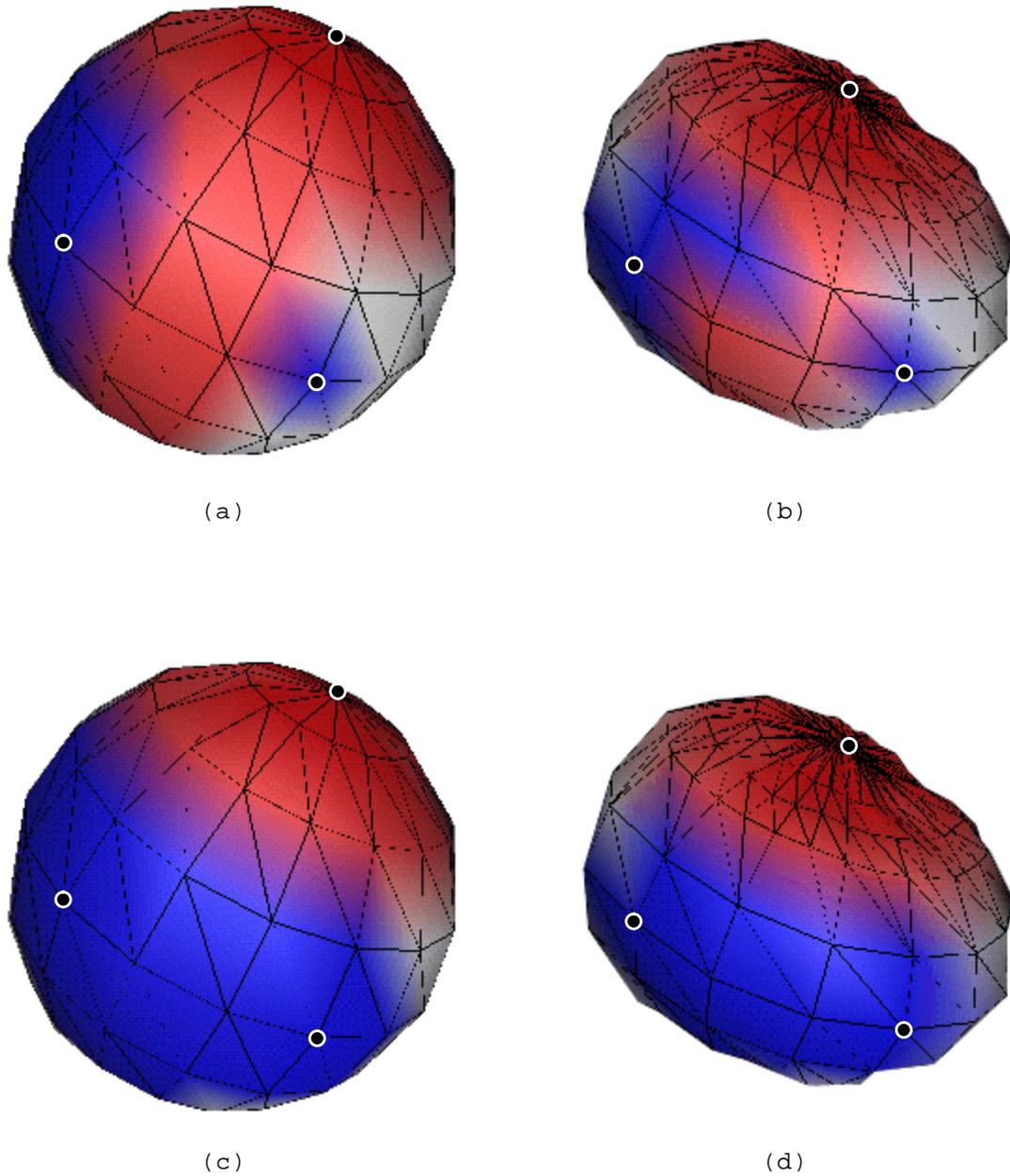


FIGURE 29. Approximating weights of a new mesh.
Expansion weights of source mesh (a) and target mesh (b).
Shift weights of source mesh (c) and target mesh (d).
Visible landmarks are highlighted with a filled circle.

6.3 Validating the Method

Validating our growth model is a tricky proposition. Currently, there is not enough hard data on craniofacial growth with which to compare the output of our software. It is relatively easy to get three-dimensional data for the craniofacial skeleton of an individual (e.g. from MRI and CT scans). However, what is lacking is data for that same individual *through time*, i.e. scans of the same person at various ages.

In the ideal case, we would be able to analyze our growth model as follows:

- (1) input data: triangular meshes for the craniofacial skeleton of an individual, at age t (call it I , for input) and at age $t+\Delta t$ (call it R , for reference).
- (2) output data: the resulting mesh at age $t+\Delta t$ after applying our growth algorithm to mesh I (call it O , for output).
- (3) the analysis: comparison of R and O .

This last step raises an interesting question: how do you compare two polygonal meshes? What is a reasonable measure for “closeness”? The problem is further complicated by the fact that the two meshes will not necessarily contain the same number of vertices and triangles. Given the current lack of data with which to compare our growth model, our research into the problem of comparing meshes hasn’t yet gone past the “brainstorming” stage.

Lastly, it should be remembered that one of the purposes of this project is to provide a mesh of an aged craniofacial skeleton, on which other software can perform facial reconstruction. Research is ongoing here at the University of British Columbia on methods for reconstructing the facial feature from a craniofacial skeleton. Comparing the results of that process with pictures of actual people could very well be the “true” validation of our growth model. For now, we must be content with the expert opinion of Dr. Geerling Langenbach and Dr. Alan Hannam, colleagues in the Department of Dentistry’s craniofacial reconstruction group at the University of British

Columbia. They are quite pleased with our results so far; our software gives them a tremendous amount of control over the process by which the growth of a mesh is computed, which is one of their requirements. Although there are disadvantages to having so much control (namely the time necessary for assigning weights to vertices), assigning weights to other bones of a given type (e.g. mandible) is relatively straightforward once the weights are generated for any particular bone of that type.

7.1 Summary

Facial reconstruction is required when police investigators attempt to identify an individual from skeletal remains. However, current methods for reconstruction of facial features are tedious and time-consuming, and require forensic specialists with years of practical experience. Furthermore, the complexity of the reconstruction problem greatly increases when time-related factors come into play, such as those that occur in missing children scenarios. Our research group is particularly interested in exploring the uses of craniofacial growth prediction to help in the identification of missing children.

This thesis lays the groundwork for using a computer to simulate the growth of the craniofacial skeleton. Our method requires a great deal of input from an expert in the field of craniofacial growth and development. As such, the software is initially targeted towards users such as forensic artists and medical researchers.

There exists a tremendous amount of data on craniofacial growth in the form of studies that collect frontal and lateral cephalograms. More importantly, this data can be used to produce the three-dimensional coordinates of certain points of reference (landmarks) on the craniofacial skeleton, at various ages. The main thrust of this thesis was in creating a model of bone growth that would use these known landmarks to drive the growth of the rest of the craniofacial skeleton. The problem was complicated by the fact that the number of identifiable landmarks is very small when considered against the entire surface of the bone. For example, in a typical triangular mesh representing a skull, the ratio of landmarks to vertices was less than 1%.

The input data to our software system is composed of five parts: a triangular mesh acquired from the bone to be grown (e.g. skull, mandible), typically from a laser range scan; the age of this bone and the target age to which growth will be simulated; a set of landmarks on the mesh; the coordinates of those landmarks through time; and a measure of the influence that the landmarks have on the rest of the vertices of the mesh. We have described two growth models that use these inputs for simulating bone growth. They mainly differ in the way that a landmark's influence is determined, and in the method used for computing the new coordinates of a vertex. Our growth software produces triangular meshes as output.

The first model of growth makes two extreme simplifications. First, each vertex only grows in the direction of the surface normal at the vertex, which simulates the naive model of bone growth where bone grows simply by addition of bony tissue to the surface. Second, each landmark's area of influence (i.e. which vertices are affected by the landmark's growth) is defined by a truncated Gaussian function centered around the landmark. This model was quickly seen to be insufficiently accurate, even from a qualitative point of view.

In the second model of growth, there are two components that determine the direction of growth of a vertex: the *expansion vector* and the *shift vector*. For the expansion component, the

vertex moves in the direction of the surface normal defined at its position. For the shift component, the vertex moves in a direction parallel to the direction of growth of one or more landmarks. The expert knowledge of the user is required to determine which landmarks influence the growth of a particular vertex. For each non-landmark vertex, the user must assign the amount of influence exerted by the landmarks. Each landmark's influence is given as two values, one for the shift term and one for the expansion term. The collection of these values is called the “weight” of the vertex.

To facilitate the assignment of the vertex weights, we provide a specialized software tool called *Krayola*. *Krayola* is an external module for use with *Geomview*, a software package provided by the Geometry Center at the University of Minnesota. *Krayola* is used to interactively “paint” the vertices of a triangulated mesh, where the colours correspond to the amount of influence of the landmarks. Painting is done in two separate layers, one for the shift weights and one for the expansion weights.

One of the keys in making our software a viable tool is the ability to free the user from the tedium of having to assign shift and expansion weights for every new triangular mesh. For this purpose, we provide a means for automatically generating a first approximation of the weights of the new mesh, given the weights of previously painted mesh of the same bone type (e.g. skull, mandible). The two meshes can have a different number of vertices and edges, which makes the technique useful for levels-of-detail analysis of a particular mesh. The only restriction is that both meshes have the same number of landmark vertices.

Validation of our growth models is an outstanding issue. Currently, what is lacking is three-dimensional data (from CT or MRI scans) for an individual *through time*, i.e. scans of the same person at various ages. For now, we must be content with the expert opinion of our colleagues in the Department of Dentistry's craniofacial reconstruction group, who are satisfied with our results so far.

7.2 Future Research

There are many open issues that demand more thorough investigation:

- Validating the growth model

While the preliminary results are encouraging, the algorithm currently being used in the growth model requires much more rigorous testing and analysis. What are “reasonable” values for the shift and expansion weights for a particular bone, and how is “reasonable” defined? Section 6.3 suggests a method for performing the validation, but it requires data for the skull of the same individual at different ages, which is currently not available. So far, tests of our software have only used on the order of 5 landmarks. It seems obvious that the accuracy of the simulation would increase with the number of landmarks used. The collection of three-dimensional landmarks for the Bolton standards that have been promised by Subramanyan and Dean [27] would be of great help in this regard.

- Improving the growth model

One of the goals of this project was for the growth model to be at least loosely based on the actual biological processes that dictate bone growth. A avenue worthy of further research would be to attempt to make our model more biologically accurate. For example, the current model is applied to a single bone at a time. A nice improvement would be the ability to grow both the skull and the mandible at the same time, by incorporating into our model a means for two bones to affect each other’s growth.

Another aspect of our growth model that needs to be addressed is the ability to detect and prevent surface-surface penetration. Consider Figure 30. As vertices *A* and *B* grow (and expand in the direction of the surface normals), they will intersect, causing unwanted artifacts. This situation is not currently being detected.

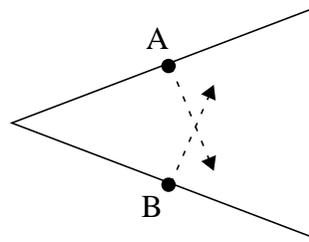


FIGURE 30. The problem of surface inter-penetration.

- Merging the growth algorithm and *Krayola*

It is not always obvious what effect modifications to the weights of vertices will have on the growth of a mesh. Currently, the user must go through a series of iterations: (1) modify vertex weights by using *Krayola*; (2) compute the growth of the mesh using these weights; (3) repeat the previous steps until the selection of weights produces satisfactory growth in the mesh. In a truly interactive environment, it would be possible for the user to immediately see the effect of changing vertex weights. For example, the user would make modifications to the weights in *Krayola*, press a button, and immediately see in a separate window the resulting growth of the mesh. Given the current trends of increasing computing power, this scenario seems feasible.

7.3 Conclusion

We re-emphasize the fact that this thesis represents only a small part in the development of a complete software package for craniofacial reconstruction. The main goals were met, in that we have created a software system that simulates the growth of the craniofacial skeleton and whose inputs and outputs are clearly defined and in a readily available format: the Open Inventor 3D Interchange File Format. Ongoing work by Katrina Archer and Dr. David Forsey at the University of British Columbia is exploring how to reconstruct the facial features of an individual using only the craniofacial skeleton. That research project will be combined with our growth system to produce aged pictures of missing children.

Bibliography

- [1] Didier Badouel. "An Efficient Ray-Polygon Intersection." *Graphics Gems*, edited by Andrew S. Glassner, Academic Press Professional, 1990, pp. 390-93.
- [2] Rolf G. Behrents. *Growth in the Aging Craniofacial Skeleton*. Monograph 17, Craniofacial Growth Series, Ann Arbor: Center for Human Growth and Development, University of Michigan, 1985.
- [3] B. Holly Broadbent. "A new x-ray technique and its application to orthodontia." *Angle Orthod.*, Vol. 1, 1931, pp. 45-69.
- [4] B. Holly Broadbent, Sr., B. Holly Broadbent, Jr., and William H. Golden. *Bolton Standards of Dentofacial Developmental Growth*, Saint-Louis, Missouri, The C. V. Mosby Company, 1975.
- [5] Peter Cahoon and Alan Hannam. "An Interactive Modelling Environment for Craniofacial Reconstruction." *Proceedings of SPIE Symposium on Visual Data Exploration and Analysis* (San Jose, California, February 7-8, 1994), Vol. 2178, pp. 206-215.
- [6] Jindong Chen and Yijie Han. "Shortest Paths on a Polyhedron." *Proceedings of the 6th Annual ACM Symposium on Computational Geometry*, 1990, pp. 360-69.
- [7] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. The MIT Press and McGraw-Hill Book Company, 1990.
- [8] Matthias Eck et al. "Multiresolution Analysis of Arbitrary Meshes." *Proceedings of SIGGRAPH 96* (Los Angeles, California, August 6-11, 1995), pp. 173-82.
- [9] Donald H. Enlow, *Facial Growth*, 3rd Edition, Philadelphia, Pennsylvania, W. B. Saunders Company, 1990.
- [10] Donald H. Enlow and Mark G. Hans. *Essentials of Facial Growth*. Philadelphia, Pennsylvania, W. B. Saunders Company, 1996.
- [11] Geomview manual, <http://www.geom.umn.edu/software/geomview/manual.html>, current as of April 19, 1997.

- [12] G. H. Golub and C. H. Van Loan. *Matrix Computations*. Baltimore, The John Hopkins University Press, 1983, pp. 425-426.
- [13] Barry Grayson et al. "The three-dimensional cephalogram: Theory, technique, and clinical applications." *Am J Orthod Dentofac Orthop*, Vol. 94, October 1988, pp. 327-37.
- [14] W. Stuart Hunter, Sheldon Baumrind and Robert E. Moyers. "An inventory of United States and Canadian growth record sets: Preliminary report." *Am J Orthod Dentofac Orthop*, Vol. 103, No. 6, 1993, pp. 545-55.
- [15] Stan Jang et al. "Three-Dimensional Analysis of Scoliosis Surgery Using Stereo Photogrammetry." *Proceedings of SPIE Symposium on Electronic Imaging Science & Technology* (San Jose, California, February 6-10, 1994).
- [16] Alex Jacobson and Page W. Caufield. *Introduction to Radiographic Cephalometry*. Philadelphia, Pennsylvania, Lea & Febiger, 1985.
- [17] R. M. Koch et al. "Simulating Facial Surgery Using Finite Element Models." *Proceedings of SIGGRAPH 96* (New Orleans, Louisiana, August 4-9, 1996), pp. 421-28.
- [18] The LEDA Library, <http://www.mpi-sb.mpg.de/LEDA/leda.html>, current as of April 19, 1997.
- [19] Rudolf Martin and Karl Saller. *Lehrbuch der Anthropologie in Systematische Darstellung*. Stuttgart, Gustav Fischer Verlag, part 3 1956, part 8 1959.
- [20] Past Forward Ltd., <http://www.pastforward.co.uk/>, current as of April 19, 1997.
- [21] Thomas Rakosi. *An Atlas and Manual of Cephalometric Radiography*. Philadelphia, Pennsylvania, Lea & Febiger, 1982.
- [22] Don M. Ranly. *A Synopsis of Craniofacial Growth*. New York, New York, Appleton-Century-Crofts, 1980.
- [23] Spencer L. Rogers. *The Human Skull - Its Mechanics, Measurements, and Variations*. Springfield, Illinois, Charles C. Thomas, 1984.
- [24] Sudha Srikrishen Saksena. *A Clinical Atlas of Roentgenographic Measurements in norma frontalis*. New York, Alan R. Liss, Inc., 1990.
- [25] Sudha Srikrishen Saksena et al. *A Clinical Atlas of Roentgenocephalometry in norma lateralis*. New York, Alan R. Liss, Inc., 1987.

- [26] William Schroeder, Jonathan Zarge, and William Lorensen. "Decimation of triangle meshes." *Proceedings of SIGGRAPH 92* (Chicago, Illinois, July 26-31, 1992), pp. 65-70.
- [27] Krishna Subramanyan and David Dean. "Scanned Bi-orthogonal Radiographs as a Source for 3D Cephalometric Data." *SPIE* Vol. 2710, 1996, pp. 717-24.
- [28] Greg Turk. "Re-tiling Polygonal Surfaces." *Proceedings of SIGGRAPH 92* (Chicago, Illinois, July 26-31, 1992), pp. 55-64.
- [29] Josie Wernecke. *The Inventor Mentor*. Addison-Wesley Publishing Company, 1994.

Estimating Coordinates of Landmarks Through Time

This appendix describes a technique for computing the three-dimensional coordinates of landmarks through time for a specific craniofacial skeleton. For the most part, it was taken from personal communications with Geerling Langenbach of the Craniofacial Biology Laboratory of the Faculty of Dentistry at the University of British Columbia. It is assumed that the reader already has a good grasp of the terminology of craniofacial growth and reconstruction.

Human craniofacial growth is often studied using cephalograms, which are X-rays of the head. Certain points on the craniofacial skeleton are identified as landmarks, and are used to describe the shape and growth of the craniofacial skeleton, by measuring the linear distances between these landmarks. Enormous databases (cross sectional as well as longitudinal) of these measurements are available (refer to the report by Hunter *et al.* [14]). Mostly a combination of two X-rays are taken — a lateral and frontal view — which gives some insight into the three-dimensional shape of the skull. However, these databases are examined two-dimension-

symphyseal area). Two landmarks have a non-midline position: Gonion (on the lateral X-ray, it's point on the curvature of the mandible angle located by bisecting the angle by lines tangent to the posterior ramus and lower border of the mandible, and on the frontal X-ray it's the most lateral point on the mandible angle), and Zygoma (on the frontal X-ray, it's the most lateral point on the zygoma bone). For four landmarks (Menton, Apex, Gonion and Zygoma) the orientation of the skull can influence the chosen point on the skull. Normally the X-rays are taken with the Frankfort Horizontal (the line connecting the upper border of the external auditory meatus with the lower border of the orbit) of the patient's head parallel to the floor. All these landmarks, except Zygoma, are easy to locate on both films.

To transform the two-dimensional linear distances between the above mentioned landmarks into three-dimensional data, the following method has been used. Starting with the lateral film, the distance between Nasion and Menton (*Na-Me*) is called the reference line, and this is assumed to be of average distance. All the other distances are scaled to this reference, and their deviation from the average is determined (in Standard Deviations). From the *Na-Me* line, the *ANS* position is determined by first drawing two circles whose centers are located at *Na* and *Me*, and whose respective radii are of length *Na-ANS* and *Me-ANS*. Then, the most reasonable of the two intersections is taken to be the position of the *ANS*. The positions of Apex and Gonion are obtained in similar ways. All these points, except for Gonion, are assumed to be exactly in the midline, so their z-coordinate is set to zero. The z-coordinates of Gonion and Zygoma are obtained using the frontal film measurements. For both of these points it is assumed that the face is symmetrical, and their position is determined by the same method using respectively the *Na-Go* & *Go-Go*, and *Na-Zy* & *Zy-Zy* lengths. For *Go* there is actually a third measurement (*Me-Go*), but this length is less accurate. With this method, the *Zy* landmark will only obtain a z-coordinate. As this landmark is not accurately locatable in

the lateral films, it will mainly be used to provide some measurement of the width of the upper face.

When any other than the average craniofacial skeleton has to be grown in time, the Nasion-Menton line again is used as the reference, and that line will be assumed to be of average length. Every other linear distance will be scaled to this average. For each of these distances, the distance from its own average (in Standard Deviations) is computed. The new lengths can now be computed for the target age. Finally, the same procedure as described above is used to obtain the three-dimensional coordinates of the landmarks in the new craniofacial skeleton.