Generating Anatomical Substructures for Physically-Based Facial Animation

OLUSOLA OLUMIDE AINA

A thesis submitted in partial fulfilment of the requirements
of Bournemouth University for the degree of
Doctor of Philosophy

November 2011

Bournemouth University

ABSTRACT

Physically-based facial animation techniques are capable of producing realistic facial deformations, but have failed to find meaningful use outside the academic community because they are notoriously difficult to create, reuse, and art-direct, in comparison to other methods of facial animation. This thesis addresses these shortcomings and presents a series of methods for automatically generating a skull, the superficial musculoaponeurotic system (SMAS – a layer of fascia investing and interlinking the mimic muscle system), and mimic muscles for any given 3D face model. This is done toward (the goal of) a production-viable framework or rig-builder for physically-based facial animation.

This workflow consists of three major steps. First, a generic skull is fitted to a given head model using thin-plate splines computed from the correspondence between landmarks placed on both models. Second, the SMAS is constructed as a variational implicit or radial basis function surface in the interface between the head model and the generic skull fitted to it. Lastly, muscle fibres are generated as boundary-value straightest geodesics, connecting muscle attachment regions defined on the surface of the SMAS. Each step of this workflow is developed with speed, realism and reusability in mind.

CONTENTS

- Boldface letters, e.g. $\mathbf{p}$, indicate multi-component or multidimensional quantities. Normal letters, e.g. p, indicate single-component or scalar quantities.
- square braces, e.g. $\mathbf{p}[i]$, indicate one component of a multi-component or multidimensional quantity.
- subscripts, e.g. p_i or $\mathbf{p}_i$, indicate a single element in a collection of single or multi-component quantities.

LIST OF TABLES

LIST OF FIGURES

Figure 7.14 Left: mutual tangents and convex hulls (both red) of muscle attachment regions (black) constructed on SMAS surface (blue). right: muscle fibres (red) as boundary-value straightest geodesics. Fibres of the procerus muscle not shown. 144

ACKNOWLEDGEMENTS

I thank my supervisors Prof. Jian Jun Zhang and Dr. Ian Stephenson for overseeing this thesis and for their patience with me. Thanks also to Prof. John Vince who cosupervised this work until his retirement, and to the Bournemouth University Media School for funding parts of my research.

In addition, I cannot but thank the following people who have directly contributed to this thesis. Steffen Engel, Eike Anderson and the rest of the guys at `zfx.info` for letting me use their subversion server; the interlibrary loans team – especially Janet Coles for fielding my very many requests for surgery journals and maths texts; My colleagues for listening to me rant and for letting me bounce ideas off them, especially Denis Kravtov for teaching me how to use the Visual Studio debugger, Prof. Peter Comninos, Prof. Alexander Pasko and David Eberly for the various helpful discussions on and offline; Sue Armstrong for proof reading my thesis and paper drafts, and for being a wonderful dance partner (Ceroc is one of the few things that make the equations go away – for a while); Prof. J. T. Kent of the University of Leeds for making John Little's thesis available to me, and for answering my many questions on kriging and thin-plate splines; Dr. Alexander Belyaev of the University of Edinburgh discussing his work on discrete differential geometry and Dr. Bryan Mendelson, of the centre for facial plastic surgery in Melbourne Australia, for his excellent research into the anatomy of the human face and for answering my many questions on the subject; Eugene Ressler for "Sketch" and to Jahirul ("Jay") Amin for the artwork, and last but not the least my brother, Femi, for the sweet IBM T41 Thinkpad on which I implemented most of the work in this thesis and on which I'm writing these words.

To my parents, James Idowu and Olufunmilayo Ayodele ('love you and I miss you both), and my siblings, Yetunde Folashade, Olufemi Olusegun and Olufunke Monisola. There is no me without you.

Thanks everyone!

Thanks be to God.

DECLARATION

Some ideas and figures from this thesis have previously appeared in the following publications:

AINA, O. O. 2009. Generating Anatomical structures for physically based facial animation. part 1: A methodology for skull fitting. *Visual Computer. 25, 5-7*

AINA, O. O. AND ZHANG, J. J. 2010. Automatic muscle generation for physically-based facial animation. ACM SIGGRAPH 2010 Poster.

Oh that my words were now written!
oh that they were printed in a book!
— Job 19:23

1 INTRODUCTION

There are five fundamental techniques of facial animation (FA). They are: key-framing with interpolation, parametric or direct parametrization, performance-based methods, muscle, and pseudo-muscle based Parke and Waters (1996). The most popular of these five techniques is the parametric method. Parametric FA generally involves setting up a system of lever-like controls with which the features and expressions of a computer generated (CG) face or head can be manipulated in the same way a puppeteer attaches wires to the hands and feet of a marionette. In puppetry as well as computer animation the setup process is known as rigging. The difficulty with rigging is that it is an iterative process that is difficult to perfect, and needs to be performed for every model that is to be animated. Nevertheless, parametric FA techniques are the most widely used because rigging is considerably cheaper and easier to implement than the other techniques.

Performance-based FA methods require that a set of markers be strategically placed on a live actor, and a corresponding set of markers on a CG face, so that the facial expressions of the live actor can be tracked and replicated on the synthetic. Although tracking can produce ultra-realistic animation, it may require expensive hardware, and typically generates a torrent of data that can be difficult to manage.

Physically-based FA methods involve modeling the complex response of human facial tissue to mechanical stress, so that unique facial expressions can be produced by considering the various forces exerted by any number of facial muscles. Physically-based methods are therefore capable of producing a wider range of anatomically-correct facial expressions than parametric techniques.

1.1 MOTIVATION

Physically-based facial animation techniques have largely remained an academic curiosity, in part because an extensive understanding of human anatomy, physics, and mathematics is required to develop them. Therefore, only large production houses such as ILM, PDI/Dreamworks, Sony Imageworks and Weta Digital, with sufficient financial resources and manpower, have been able to develop physically-based facial animation systems.[1]. Smaller studios and independent artists, however, rely exclusively on general-purpose features of off-the-shelf 3D software such as Maya, Softimage, 3D Studio MAX, and

1 Cary Phillips of ILM and Dick Walsh of PDI/Dreamworks's were awarded technical achievement awards in 1998 and 2003, by the academy of motion picture artists for muscle-based facial animation systems developed for the movies *Dragonheart* and *Shrek* respectively.

LightWave, none of which support physically-based facial animation, for their facial animation projects.

Another likely reason for the slow uptake of existing physically-based FA methods is their non-reusability: a new system needs to be painstakingly constructed for every CG face. Although parametric facial animation techniques share this limitation, physically-based systems are, in addition, significantly more difficult to construct. This situation is summarized in Table 1.1.

Technique	Easy to set up	Reusable
Parametric	somewhat	(rig) no
Performance	somewhat	(data) yes, (rig) no
Physically-based	no	no

Table 1.1 Facial animation techniques: ease of use versus reusability

A survey of artists identified the following additional limitations of existing physically-based facial animation techniques. These shortcomings, listed below, can rightly be considered as research challenges.

i. Existing physically-based FA techniques are difficult to art-direct compared to other facial animation techniques. The importance of control over visual elements cannot be over-estimated in the CG industry where there is always a specific look that must be achieved.

ii. Most physically-based FA techniques are computationally expensive to run and very difficult to setup. In comparison, most of the simpler facial rigging techniques can be used to achieve comparable results in less time and with less effort. As such, physically-based FA techniques will only become commonplace when they become considerably easier to build and customize and demonstrate significant performance benefits over competing methods. (Current production practices and pipelines are unlikely to be replaced by techniques that offer only marginal improvements.)

iii. Because subjects of character animation projects are often cartoon-like or fantasy characters for which there is no need for realistic facial expressions, the demand for facial animation techniques geared toward producing realistic expressions is often understated. The *Navi* characters in the movie "Avatar" and the "Benjamin Button" character in the movie "The curious tale of Benjamin Button" are just two examples of CG characters that required realistic facial animation.

1.2 THESIS OBJECTIVES AND CONTRIBUTIONS

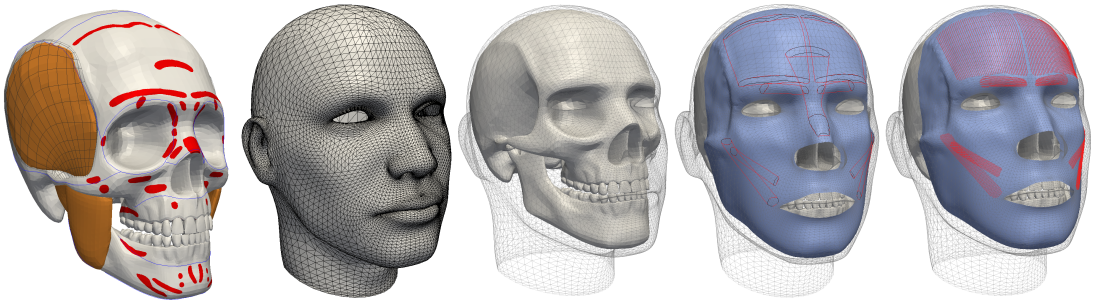
This thesis develops a series of methods for automatically generating the bony and soft-tissue substructures for any given 3D face model, toward the goal of physically-based facial animation. As shown in Figure 1.1, the first part of this workflow involves fitting a generic skull to a given human (or humanoid) head model. This is done in two steps. First, pairs of stationary landmarks and sliding semilandmarks are placed on the skull and the head models. The latter set of landmarks are assigned virtual offsets based on interactively-scaled soft-tissue thicknesses originally measured for forensic purposes. The semilandmarks on the skull model are allowed to slide until the bending energy of the thin-plate spline deformation is minimized, and an initial fitting of the skull to the head is performed using this "optimized" landmark set. In the second step, a more accurate, corrective, refitting is subsequently performed based on landmark normals.

In the second part of the workflow, the SMAS is modeled as a Hermite variational implicit or radial basis function surface, constructed from a set of point constraints and field normals obtained from the given head model and the skull fitted to it. The implicit surface is discretized using the marching triangles method. This discretization process is terminated by a set of discrete curves indicating the orifices and the extent of the face region.

Third, and lastly, the individual fibres that constitute each muscle of facial expression are modeled as boundary-value straightest geodesics on the surface of the (discretized) SMAS. These fibres are generated by interpolating the mutual tangents connecting a pair of muscle attachment regions specified on the SMAS. Boundary-value straightest geodesics are computed by iteratively straightening the straightest discrete curve that can be embedded in a sequence of polygon faces, based on a set of Euclidian-distance heuristics. The development of this boundary-value straightest geodesic algorithm is one of the unique contributions made in this thesis.

This thesis also argues that physically-based facial animation systems, for which automatic rig-builders exist, can be production-viable and makes the case for a facial expression subsystem that is capable of using the anatomical information generated in order to compute facial expressions. This framework lends itself to a layered, modular pattern of development that allows any components to be substituted. For example, it should be possible to use any one of a wide variety of known or yet undiscovered skin-solvers in order to compute facial expressions. Furthermore the entire process:

- Takes the age, gender, body type, and ethnicity of the character being rigged into account during the skull fitting step, by using facial tissue depth data dependent on these factors.
- Conceivably works for a range of humanoid characters, e.g. those without horns. This is because the skull fitting step only requires that a set of landmarks be locatable on an input head model, whose form need not be strictly human. The



(a) (b) (c) (d) (e)

Figure 1.1 (a) Generic skull package (b) typical head model (c) generic skull fitted to head model (d) SMAS-plane showing and mutual tangents and convex hulls of muscle attachment regions (e) muscle fibres as boundary-value straightest geodesics.

workflow also readily permits the arbitrary creation of any number of non-standard or phantom muscles, particularly for fantasy creatures.

- Is designed to require moderate user interaction by default, e.g. the manual placement of a set of landmarks during the skull fitting and SMAS construction step, and avoids the two extremes of allowing no user input (over-automation, or automating-out artistic freedom) or requiring too much user input (no automation). Furthermore all components of the system have default configurations that work well, but can nevertheless be altered – thus unburdening the user, while remaining flexible.

- Employs familiar computer graphics primitives such as textures, and would therefore be easy to incorporate into an existing production pipeline. Good tools must support and enhance tried and tested workflows.

The anatomical substructures generated are validated by comparing them with descriptions and illustrations sourced from anatomy texts and references. However, because this work is done primarily with a view toward physically-based facial animation for movie production, its ultimate validation would be the realism and level of detail of the facial expressions generated using the models. Furthermore, in light of the target application, production-quality facial models used in such applications are assumed.

1.3 OUTLINE

This thesis consists of eight (8) chapters, with the following four (4) logical divisions:

PART 1 – Introductory material

CHAPTER 1 – the current chapter. Reviews facial animation techniques in general, highlights the limitations of existing physically-based facial animation techniques and makes a case for further research into this class of methods.

CHAPTER 2 – Facial Animation: Techniques and Applications. This chapter broadly discusses various facial animation methods and reviews their strengths and weaknesses. However, special attention is paid to physically-based facial animation methods. An in-depth critique of some landmark and recent works identify a lack of anatomical detail, non-reusability and overly-simplistic constitutive relationships as the primary shortcomings of the state of the art. An appraisal of the demands placed on facial animation technologies by several application domains is also made.

PART 2 – Foundations: these chapters detail the various concepts that inform the development of the framework presented in the implementation chapters (Chapters 6 and 7). These concepts are discussed in advance of the implementation so that references to the various foundational concepts need only be made in later chapters. This separation (of foundations from implementation) is done to keep the implementation chapters as simple as possible, and to prevent the confusion of the development of the relevant concepts with their applications.

CHAPTER 3 – Anatomy of the human face. In this chapter, the anatomy of the hard and soft tissues of the human face is presented. In particular, the five layers of soft tissue that comprise the human face are introduced. The SMAS is the third layer of this soft tissue matrix, and is modeled in Chapter 7 as a variational implicit or radial basis function surface.

CHAPTER 4 – Some basic theory and applications of thin-plate splines. This thesis makes repeated use of the thin-plate spline method in computing the true position, normal and principal tangents at (landmark) points on the surface of a mesh (Chapter 6). The method is also used to model the SMAS as an implicit surface (Chapter 7) and in order to fit a generic skull to a given head model and determine the optimal placement of slidable semilandmarks (Chapter 6). Due to its fundamental importance, this chapter gently develops the thin-plate spline method from first principles and illustrates its use with the five applications that are relevant to this thesis.

CHAPTER 5 – Boundary-value straightest geodesics on discrete manifolds. Boundary-value straightest geodesics are a generalization of straight lines to non-planar, discrete manifolds. They are used at each stage in the construction of muscle fibres (Chapter 7). This chapter develops three path tracing and path straightening heuristics for constructing boundary-value straightest geodesics, and also presents an algorithm for approximating the straightest-possible geodesic in the neighborhood of hyperbolic vertices, where no straightest geodesic exists. In addition, the neighborhood or corridor of the approximated path

is exhaustively searched in order to validate the result of the three heuristics, when no straightest geodesic is found. This process is accelerated by the early detection and pruning of subpaths that cannot embed a straightest geodesic.

PART 3 – Implementation chapters.

CHAPTER 6 – A landmark based method for skull fitting. This chapter develops a two-step method for fitting a generic skull to a given head model. In the first step, a primary morph is performed using an optimal set of stationary landmarks and an arbitrary number of optimally-located sliding semilandmarks, the placement of which minimize the bending energy of a 3D thin-plate spline that deforms one set of landmarks to the other. The secondary morph is performed using another 3D thin-plate spline, computed from the correspondence between a subset of landmark normals and principal tangents on either model. This is done in order to improve the initial fit. A simple, interactive technique for scaling experimentally-determined soft tissue thicknesses is also introduced and is used to offset landmarks placed on the head model, based on the gender, body-type, age-range and ethnicity assumed for (or assigned to) the head model.

CHAPTER 7 – Constructing facial muscles and the superficial musculoaponeurotic system (SMAS). In this chapter, a novel approach to constructing muscles of facial expression on the SMAS as boundary-value straightest geodesics is developed. First, the SMAS is constructed as a variational implicit or radial basis function in the interface between a given head and the generic skull fitted to it. This is done using a set of surface and (field) normals sampled from both models. Next, the fibres of the facial mimic muscle system are computed as the mutual tangents of their muscle attachment regions. This process involves generalizing the concept of a convex hull to non-planar discrete manifolds.

PART 4 – Summary and conclusions

CHAPTER 8 – Conclusion: future work, possibilities and perspectives. This chapter summarizes the contributions and discusses the limitations of some of the techniques developed in this thesis. The latter is suggested as future work. However the two potential applications areas that stand to benefit most from this and future developments are highlighted. In particular, this chapter briefly argues that, contrary to popular belief, physically-based facial animation techniques are production-viable.

PART 5 – End matter

APPENDIX A

BIBLIOGRAPHY

of making many books there is no end
— Ecclesiastes 12:12

2

FACIAL ANIMATION: TECHNIQUES AND APPLICATIONS

This chapter surveys and critiques milestone facial animation technologies, as well as the state of the art following the broad classification of Parke and Waters (1996), and reviews physically-based facial animation techniques in considerable detail. However, what follows is not intended to be an exhaustive survey of facial animation, as several detailed surveys have already been published e.g. Parke and Waters (1996), Noh and Neumann (1998), Krinidis *et al.* (2003), Haber and Terzopoulos (2004), Radovan and Pretorius (2006), and Ersotelos and Dong (2007). Accordingly, publications cited herein are intended merely to be representative of a much larger body of work. Furthermore, because the emphasis is on 3D computer facial animation, this chapter does not discuss facial animation in the image space, for example Ravyse (2006) and Ezzat *et al.* (2002), or works devoted to facial modeling such as Blanz and Vetter (1999) and other criteria that facial models must satisfy in order to meet animation requirements. For this reason, the subject of computerized facial reconstruction will not be discussed in this chapter, but is reviewed instead in Chapter 8. In essence, this chapter focuses on the methods of and mechanisms (i.e. rigs) for driving facial animation.

Applications of facial animation technologies are also discussed, in addition to examining current thinking as to what constitutes good facial animation. This chapter concludes by identifying a number of research challenges, and makes the case for more sophisticated facial animation techniques.

2.1 FACIAL ANIMATION TECHNIQUES

2.1.1 *Blend Shape Interpolation*

This is the simplest and oldest of all facial animation techniques. The technique derives from the traditional cartoon animation workflow in which a senior artist draws key poses, and animation assistants draw the transitions between adjacent key poses over a specified number of frames. The method was first adapted to 3D facial animation by Parke (1972). In 3D however, an animator is the "senior artist" and a computer program the animation assistant which interpolates between key poses specified by the artist. The key-poses are almost always variants of a base or neutral mesh. Because the process of interpolation in 3D is in fact a blending or morphing operation, the variant meshes are commonly referred to as blend shapes or morph targets. Blend shapes represent the extremes of the facial expression e.g. smile, wink, lip shape etc. to be performed by a character. All blend shapes share a common topology, so that each vertex of any blend shape has a

unique corresponding vertex on all other blend shapes. An interpolation function can therefore be used to compute the intermediate position v of any vertex at time t, from its corresponding positions v_1 and v_2 on any two chosen blend shapes at times t_1 and t_2 ($t_1 < t_2$). For example, assuming a linear interpolation function,

$$v = (1-s).v_1 + s.v_2 \qquad \text{where} \qquad s = \frac{t-t_1}{t_2-t_1} \quad \text{and} \quad t_1 <= t <= t_2$$

The factors s and $1-s$ are weighting or blending functions, and add up to 1. Higher order or nonlinear blending functions can also be used if control over the rate of approach of both targets is desired. It may also be desirable to interpolate between more than two blend shapes. For example, in order to have an angry-looking character blink while speaking, an interpolation of the angry, blink and lip shape blend shapes must be performed. However, care must be taken when trying to produce some lip shapes and other "rigid" blend shapes. In speech production one or more parts of the vocal apparatus are always required to attain a target. For example, producing the /b/ sound requires that the lips touch but are free to spread or be pursed as in a smile or kiss. Such constraints must be taken to consideration.

In mathematical terms, blend shapes are the basis functions of a possibly infinite-dimensional space $\mathcal{S}$ of all facial expressions. Accordingly, any two blend shapes will only generate those facial expressions in the one-dimensional subspace of $\mathcal{S}$ formed by both basis functions, while three blend shapes will generate facial expressions in a (larger) two-dimensional subspace of $\mathcal{S}$ etc. In theory therefore, an infinite number of blend shapes are required to reproduce the space $\mathcal{S}$ of facial expressions. In practice, $\mathcal{S}$ can be approximated by a very large number of blend shapes.

Nevertheless because blend shape interpolation requires relatively simple per-vertex calculations, it can be readily implemented in modern programmable graphics hardware (Beeson (2004), Lorach (2007)).

2.1.2 *Parametrization*

The obvious problem with blend shape interpolation is the large number of blend shapes that need to be created for each character in an animated sequence. This is especially true of "hero characters" as they need to exhibit a wide range of emotions and perform extensive dialogs. The process of sculpting all the required blend shapes from the base mesh is clearly a tedious undertaking.

One way to simplify the task of blend shape construction is to parameterize the face mesh. Parametrization is based on the assumption that any facial expression can be constructed from a finite set of parameters. Each parameter deforms a region of the mesh and produces a distinct facial action e.g. a wink, or the rounding of the lips. The most popular parametrization in use by artists is the Facial Action Coding System (FACS), developed by Psychologists Ekman and Friesen (1978) for distinguishing facial expressions in their study of human emotions. The FACS deconstructs almost all known

facial expressions in terms of 45 Action Units (AU), some of which are the result of the action of more than 1 muscle. The FACS is an excellent parametrization of the human face, and has become the de-facto parametrization for most artists. The process of parameterizing a model is called *rigging*. Virtually any blend shape can be more readily constructed from a suitably rigged model.

Parameterizations are commonly constructed using a combination of the following components, which, when used together together, constitute a rig:

i. Skeleton: a relatively simple hierarchical system of joints equivalent to the human skeleton[1] used to pose a character. The connections between joints are referred to as bones. The skeleton is bound to a model, so that each joint has a degree of control over several vertices of the mesh. The degree of control of a joint over a vertex, i.e. the vertex weight, normally depends on the distance between the node and the vertex. Weights can however be manually adjusted by painting. Weights range from 0 (joint has no influence on vertex) to 1 (joint has complete control over vertex). As a result, a vertex can be influenced by more than one joint. Skeletons are ideal for jointed or articulated structures such as the jaw or neck, but can also be used for producing movement in other parts of the face such as the lips and forehead. The process of binding a skeleton to a mesh and setting weights is known as *smooth skinning*.

ii. Influence object: non-renderable geometry whose position affects skin points in its vicinity. Influence objects are used to refine smooth-skinned meshes by preventing unwanted deformations or simulating the movement of veins, muscle bulges and tendons that would be difficult to simulate by smooth-skinning. The effect of an influence object can also be triggered by the rotation of a joint.

iii. Lattice: a grid-like structure of control points of arbitrary resolution, for performing free-form deformations. The lattice is bound to a face model, so that changes made to the lattice control points affect mesh vertices in its neighborhood. Lattices can be used to simulate puffed or sunken cheeks.

iv. Cluster deformer: control a given set of control points e.g. mesh vertices with varying degrees of influence. Each control point is weighted to indicate the degree to which they are to be controlled by the transformations (e.g. translations, rotations, scaling) made to the cluster deformer.

Although rigs simplify the task of blend shape construction, the process of smooth-skinning each character with a skeleton, weight painting, and setting up influence objects, lattices and cluster deformers in order to produce all 48 FACS AUs for example is a very complicated and time consuming process.

1 For the rest of this chapter, unless otherwise stated, the words skeleton and bones will refer to their CG equivalents.

2.1.3 *Performance-driven Facial Animation*

> "I can generally do about 30 seconds of quality facial animation in about 4 hours. I did some math and I figure at that rate, it would take me around 164 years to complete the facial animation and our game would ship in the year 2,173. Even if we had 4 animators working around the clock non-stop all year, it would still take almost 10 years to complete just the facial animation."
>
> Ben Cloward (2010)
> Senior Technical Artist, Bioware

Although tools exist for constructing blend shapes, creating realistic facial animation by interpolating blend shapes alone remains a difficult task, even for skilled animators. In fact, "the consistent production of large amounts of flawless facial animation (by keyframing) is thought to be expensive and impractical" (Pighin and Lewis, 2006a). This is partly because the motion path between facial poses is not smooth as assumed in blend shape interpolation, but rather complex and irregular. These irregularities in facial motion are due to nuances of human facial motion without which virtual character animation appears distinctly unreal. As such, the obvious approach to incorporating these subtleties in facial behavior is to copy real facial motion and replicate them on virtual characters. In the absence of a recorded performance, animators commonly resort to using themselves as motion references, and often unwittingly reproduce their facial nuances in the characters that they rig. This unnamed observation was confirmed in a survey of character animators.

The performance-based facial animation method as first demonstrated by Williams (1990), involved tracking the movements of strategically located markers in a video recording of a live actor. These movements were targeted to a CG model, by a set of radial basis functions used as warping kernels. In the 20 years since William's publication, such considerable advances have been made in the acquisition, processing and retargeting of facial motion data that a comprehensive review of the state of the art now spans entire volumes, for example Pighin and Lewis (2006b), Deng and Neumann (2008).

Some of the terms commonly used to describe performance-based facial animation techniques include: marker-based or markerless, depending on the use of markers for feature tracking, and, volume or face-only, if body motion and face motion are captured simultaneously or otherwise. The simultaneous capture of face and body motion of several actors in a volume permits actors to move, interact, and therefore emote better. In comparison, face-only capture systems often restrict a performer to a chair in front of a camera (or cameras). Also, a separate body motion capture session may be required. This two-step process can result in facial expressions that do not correspond to body movements. Dynamic facial textures are more readily extracted from markerless capture systems.

Marker-based volume capture systems are however very complicated and require that marker identification, labeling, tracking, stabilization, and normalization be performed. Frame by frame marker identification and tracking is necessary for recovering motion and distinguishing the markers that belong to each actor. This process can be complicated by noise caused by errors in the computation of marker locations (from camera images), and missing, invisible or occluded markers. Stabilization involves the removing of head rotations and body movements from facial markers. This is necessary because the process of retargeting facial motion requires that the head be stationary. Marker normalization involves compensating for discrepancies in the positions of markers at the start of each capture session. This is accomplished by comparing the initial positions of all markers at the start of each shoot to a single, master set or pose whose marker positions are accepted as correct. In the typical workflow, these steps are automated but considerable manual tweaking is required in the event of errors. Sony Imageworks Havaldar (2006) uses such a system, consisting of 200 infrared video cameras that record lights reflected off markers placed on bodies and faces of up to 6 actors in a 20 x 20 x 16 feet (length, width, height) volume. (Eighty markers were placed on each face.)

The MOVA contour reality capture system Perlman (2006) is a face-only, markerless performance capture system. It requires that the face of a performer be dabbed with off-the-shelf phosphorescent makeup that is not visible under white light. The system consists of two arrays of inexpensive video cameras (up to 44 in number), synchronized to the fluorescent lights flashing at about 90–120 fps (beyond 80 fps the human eye does not perceive flashing). One group of cameras capture the glow of phosphor powder on the face of the performer in the dark while the other group captures normal images. Both image types are acquired from various angles. Randomly distributed phosphor patterns from the "dark frames" are correlated and used to accurately construct an ultra-high resolution three-dimensional shape of the face per frame, while images from the "lit frames" are used to processed into dynamic textures.

The Universal Capture (UC) system Borshukov *et al.* (2003) is another example of a markerless technique, capable of recording the facial motion of a live actor and playing it back under different lighting conditions. In addition to geometry, UC also captures dynamic textures. A marker-dependent version of the UC system for use in real-time scenarios also exists Borshukov *et al.* (2006).

The University of California's (USC) Light Stage 5 facility and pipeline, used to capture facial expressions for the Digital Emily Project Alexander *et al.* (2009), is another notable face-only capture system. This system, however, requires that several small dots be placed on the face of the performer in order to help align various facial scans acquired by a pair of stereo digital still cameras under different lighting gradations and patterns of polarization – actors are required to hold their facial expression for about 3 seconds, while the cameras acquire 15 images under the various lighting conditions. These images are used in order to compute diffuse, specular intensity and normal maps, as well as

high-resolution facial geometry for each expression. This data is used to build blend shapes.

Other state of the art facial capture devices include head-mounted cameras permitting the independent capture of facial expressions during body motion capture sessions, and electrooculograms for measuring and recording eye movements. Head-mounted cameras and electrooculograms were used in the movies "Avatar" and "Beowulf" respectively Duncan (2010), Duncan (2008).

In spite of these advances performance-based facial animation methods possess several shortcomings. For example, the technique:

i. Requires extensive post-processing of data, and although post-processing algorithms have been developed, the process frequently requires manual correction or human intervention. Also, the resulting output of the process may not be aesthetically pleasing, and must therefore be fine tuned by an animator. For example, processing the 37 Digital Emily facial expression scans acquired by the USC Light Stage 5, took 1 artist 10 days, and rig constriction took 3 months Alexander *et al.* (2009).

   Furthermore, the cost of data acquisition, processing hardware, and software makes performance capture very expensive and difficult to set up.

ii. Does not obviate the need for rigs, as a rig is required to retarget captured data. Therefore, rig construction unavoidably adds to the complexity of the process.

iii. Does not automatically guarantee convincing facial animation. This is because the process of retargeting performance data to a rig is in fact a sampling operation, which, if not done properly, will result in facial animation that fails to reproduce important, life-like nuances of the live actor. Furthermore, the selection and placement of markers on the face of the performer also amounts to an additional, preliminary sampling operation (prior to retargeting). This too can contribute to deviations of the resulting facial animation from the performance of the live actor. Furthermore, the quality of the final animation is determined by the sophistication of the rig.

2.1.4 *Physically-based facial animation*

Physically-based facial animation techniques were introduced by Platt and Badler (1981) as part of an effort to develop an efficient and accurate model of the human face for an American Sign Language (ASL) project. Their model consists of a high-level skin mesh driven by muscle fibres grouped into a tension net that propagates the force of muscle contraction to the skin surface. Their work also introduced the FACS to facial animation, and the idea of step-wise contraction of muscles analogous to Euler's method of solving differential equations. They also proposed (but did not implement) several

other important ideas such as: the extraction of action units from video (nine years before facial performance capture was implemented by Williams), the incorporation of an articulated skull into the facial models, muscle sheets that flow over the skull and cartilage, and, the simulation of complex action of the lips, tongue and cheeks. These proposals appear to have set the stage for future research which falls under the three following categories. The accuracy of each technique depends on its closeness to reality as determined by its anatomical correctness and the validity of the physical laws it employs.

Muscle vectors

Waters (1987) muscle model treats facial skin as a flat elastic sheet, and defines mathematical relationships that simulate the response of skin to abstracted linear (vector-like), rectangular and elliptical muscles. These models are very simple and easy to implement. Unfortunately, they are not very accurate as they does not take the geometry and biomechanics of the face into account, i.e. they are based on a flat model of facial skin, and assume that muscles are either linear or planar (whereas real muscles follow the geometry of the face and skull). In spite of these shortcomings, these muscle models proved to be very influential as will be subsequently shown.

Mass-spring systems

In contrast to the muscle vector technique, in mass-spring systems skin is not modeled as a sheet of zero thickness, but rather as a grid of nodes connected by springs to form a lattice representing several layers of soft tissues. The outermost part of this lattice is formed by the face mesh that is to be animated. Displacements applied to any node of the mass-spring system, e.g. as a result of muscle action, generate unbalanced forces that propagate throughout the system in order to attain a new equilibrium.

Mass-spring facial animation models were introduced by Terzopoulos and Waters (1990). The system they developed models the cutaneous, subcutaneous, and muscle layers of the human face with structurally-stable cross-strutted springs, and employed biphasic springs for approximating the behavior of skin. (Human tissue is readily extensible under mild forces but increasingly exhibits resistance to further deformation after a threshold is reached.) They also used constraints to penalize changes in the volume of the lattice's elements in order to simulate the incompressibility of soft tissues, and used energy minimizing splines (or snakes) in order to track facial features in video footage of a human face from which muscle activation levels were subsequently extracted and used to drive the mass-spring system. However, their model does not explicitly incorporate a skull (but rather constrains the bottommost layer of nodes) and represents muscle as linear vectors.

The mass-spring system developed by Kähler *et al.* (2001) (see also Kähler (2003)) also used biphasic springs but nevertheless represents a variation on the earlier approach. They model muscles as a sequence of segments consisting of overlapping ellipsoids

indicating the size and shape of each portion of a muscle. This representation allows muscles to merge and intertwine, especially at the corner of the mouth as real muscles do. Springs run along the middle axis of each muscle and connect merged muscles by a common node. Muscle generation is user-initiated via an interactive muscle editor for sketching the outlines of muscles on a given head mesh. The outlines are automatically subdivided, offset beneath the head mesh and used to create small volumetric cells from which overlapping ellipsoids are derived. The ellipsoids overlay an approximation of a skull which is created by applying a mesh simplification and an offset operation to the head mesh. The approximate skull is only used in muscle construction and is not part of the mass-spring system (skull penetration is handled by constraints). In order to reduce the likelihood of the lattice collapsing or folding over, each node is given a second spring that pushes outwards and resists volume changes. The ellipsoidal muscle units are allowed to bulge on contraction. Unfortunately, this bulging is unwarranted as mimic muscles are thin and sheet-like, and therefore undergo negligible increases in thickness upon contraction.

Zhang *et al.* (2002) (see also Zhang *et al.* (2004)) incorporate a more-realistic skull model (manually fitted using affine functions) in their mass-spring system and abandon biphasic springs in favor of non-linear functions for simulating the stress-strain relationship of real tissue. (Biphasic spring functions are discontinuous and produce sudden jumps and other visual artifacts.) They use edge-springs to prevent the lattice from collapsing but incorporate older ideas such as cross-strutted springs, skull constraints, and incompressibility constraints. They also introduce the novel idea of adaptively upsampling the head mesh and mass-spring according to the complexity of the deformation it experiences. Unfortunately, as no scheme for downsampling the mesh and mass-spring system is provided, the model is likely to become heavy and eventually slow down after considerable use, especially if the mesh experiences large deformations. Furthermore they employ linear, sphincter, and sheet muscle models comparable to those developed by Waters (1987).

Although all the mass-spring systems discussed above run in real-time, a lattice structure consisting of nodes and elastic springs cannot accurately model the behavior of real soft tissues. In fact, Gelder (1998) shows that "assigning the same stiffness to all springs badly fails to simulate a uniform elastic membrane", and that also "an exact simulation is in general not possible".

Furthermore, the elastic response model underlying mass-spring systems is grossly inadequate for problems involving large deformations, and must be augmented with constraints, for example to preserve the volume of the lattice, in order to produce plausible deformations. An ideal model of soft tissue deformation would implicitly encode the required constraints in its physical laws and should not require ad-hoc constraints. However, the most serious shortcoming of current mass-spring systems is their inability to model shear or sliding between various layers of soft tissue. (The anatomy of the face is such that, under the action of muscles, superficial layers of tissue

slide over deeper layers. Simulating such an effect with a mass-spring system would require the duplication of nodes at the sliding interface.) Also, current mass-spring systems are susceptible to element collapse, if the forces on elements exceed the stiffness of the constraints or if the time steps taken are too large.

Finite-element methods

The finite-element method (FEM) was first applied to problems of skin deformation by Larrabee and Galt (1986). Their model consists of 2D elastic membrane divided by a series of nodes attached to springs representing subcutaneous attachments to a nonmobile underlying structure. This model was developed in order to study skin flap advancement and wound closure during surgery. Although their model employed a linear stress-strain relationship, it did not incorporate viscoelastic properties and incorrectly assumed that skin is isotropic with zero tension in its resting state, but nonetheless produced more accurate results than existing techniques at the time.

One of the recommendations for further work made by Larrabee and Galt (1986) was the development of a curved 3D shell-like version of their model. This was done by Deng (1988), whose model comprised of a topmost skin layer, a middle sliding layer and a bottom muscle layer in contrast to the single-layer model of Larrabee and Galt (1986).

These landmark works were followed by several publications, such as Keeve *et al.* (1996), describing applications of the FEM in maxillofacial facial surgery simulation. However, none of these publications addressed the problem of how to generate facial animation using the FEM. The first team of researchers to do so was Koch *et al.* (1998). Their model was constructed for the Visible Human dataset from which a head and skull geometry was extracted. A system of "main springs" was subsequently constructed by projecting the vertices of the facial surface to the skull, followed by a system of strut springs. The stiffness of each spring was computed from the weighted-average of stiffnesses assigned to each distinct tissue type penetrated by the spring while the force of muscle contraction was modeled using linear vectors. Texture maps were used to specify boundary conditions indicating rigid (non-displaceable) and non-rigid (displaceable) as well as stretching and bending tensors indicating the resistance of various parts of the face to deformation. The use of linear muscles and linear elastic springs are the major shortcomings of this model. Also, the constitutive model they used did not take into account the incompressible nature of human soft tissues.

Choe *et al.* (2001) also use simple quadrilateral and sphincter muscle models in their FEM of the human face. In addition, they described an algorithm for estimating muscle actuation parameters from video recordings of a set of markers placed on the head and face of a live actor. The estimated muscle activations were fed into the muscle models and thus used to drive the facial animation. Unfortunately, their FEM does not consider the human skull and assumes a linear elastic skin.

Gladilin *et al.* (2001) abandon the simple model of the facial muscles and instead describe muscles as a field or bundle of fibres running from one one end of the muscle

to the other. This muscle field is constructed from any partial differential equation of mathematical physics capable of modeling flow, for example Laplace's equation, with boundary conditions specifying the shape of the muscle at either end as well as the muscle body. (Muscle shapes were obtained from CT-scan data.) The other restriction imposed on the solution of the PDE is that the resulting field does not generate knots. The force density acting in the direction of fibre tangents is obtained by multiplying by an empirically determined force magnitude with the unit vector field of fibre tangents. The deformation of surrounding soft tissues due to muscle force densities is computed by solving a boundary value problem describing an isotropic homogeneous linear hyperelastic material using the FEM.

Sifakis *et al.* (2005) use B-splines to represent muscles fibre fields in an anatomically-accurate model of the head, neck, and skull extracted from the Visible Human dataset. This data is morphed to fit the laser scan of a living subject based on point correspondences. A quasistatic FEM incorporating material non-linearities was used to simulate the deformation of skin in response to muscle contractions. They also outline an optimization framework for extracting muscle activations of their human subject from a set of facial markers. This is done by computing the muscle activations that minimize the distance between virtual and real marker locations for each frame of video.

Barbarino *et al.* (2008) (see also, Barbarino *et al.* (2009)) extracted the surface form of a human head, a subset of muscles, in addition to the skull and the jaw from MRI images, and, supplement it with anatomically-faithful models of the superficial musculoaponeurotic system (SMAS), parotid gland, retaining ligaments, and the subset of muscles that could not be resolved from the MRI images. The SMAS and the tissues above it are represented as one layer of constant thickness and deep fat is constructed as a filler of all empty space below the SMAS. Adjacent organs are attached by connective tissues. They use a detailed, non-linear material model of biological tissues and experimentally obtained material constants, and validate their FEM by comparing their computed skin deformations with observed results in a variety experiments such as: air filled oral cavity (puffed cheeks), sagging of facial tissue under gravity loads and displacements caused by foreign objects in both cheeks. The level of anatomical detail they incorporated and the constitutive equations they employed makes their model the most sophisticated physically-based model of the human face to date. Unfortunately, the technique is not immediately applicable to CG models for which there is no anatomy to extract from MRI images.

A comparison of the above physically-based facial animation techniques is presented in Table 2.1. It is worth noting that, although this thesis presents a work in progress, the techniques developed (herein) advance certain aspects of the state of the art, by: accurately fitted a generic skull model to a head model (using sliding semilandmarks and landmark normals), automatically generating muscles fibres whose form (straightness) follows their function (contraction) and modeling the SMAS without recourse to medical imaging data.

Type	Author	Use of skull model	Geometry of muscles	Other substructures modeled	Biomechanical model of soft tissues
Geometry-based deformation	Waters (1987)	none used	linear, elliptical, rectangular	outer skin surface only	not taken to account
Mass spring	Terzopoulos and Waters (1990)	none used	linear vectors	cutaneous, subcutaneous	biphasic springs
	Kähler *et al.* (2001)	Yes	chains of ellipsoidal "fibres"	epidermis, subcutaneous tissue and fatty layer	biphasic springs
	Zhang *et al.* (2002)	Yes	linear and sheet	epidermal, dermal, hypodermal layers	nonlinear springs
FEM	Koch *et al.* (1998)	Yes (extracted from VHD)	linear vectors	no distinction between soft tissues made	linear elasticity
	Choe *et al.* (2001)	none used	quadrilateral, elliptical (sphincter)	no distinction made between soft tissues	linear elasticity
	Gladilin *et al.* (2001)	Yes (extracted CT scan data)	field or bundle of fibres	no distinction made between soft tissues	linear elasticity
	Sifakis *et al.* (2005)	Yes (extracted from VHD)	B-spline fibre fields	no distinction made between soft tissues	Hyperelastic
	Barbarino *et al.* (2008)	Yes (extracted MRI scan data)	3D geometry	retaining ligaments, SMAS, deep fat	Hyperelastic
Unknown (WIP)	Aina (2011)	Yes	geodesic fibres	SMAS (at present)	unknown (WIP)

Table 2.1 Comparison of Physically-based facial animation techniques reviewed in Section 2.1.4. (VHD: visible human data, WIP: work in progress.)

2.2 FACIAL ANIMATION REUSE

As highlighted in Table 1.1 facial animation rigs are generally non-reusable and must be recreated for every head model to be animated. As such, the obvious approach to reducing the animator's workload is to reuse the elements of a previous facial animation project or animation. Existing attempts at addressing the problem of reusability fall under the three following categories:

2.2.1 *Expression cloning*

This technique was first introduced by Noh and Neumann (2001), and was designed to copy facial animation from one head model to another irrespective of dimension, topology and the technique by which the animation was created. Expression cloning facilitates the compilation of a library of facial motions and the rapid construction of blend shapes by copying facial poses from other models. However, the variety and quality of such libraries and blend shapes are limited to (or by) the motions and expressions of the input face models. The technique consists of two major steps. First, a source mesh is morphed to a target mesh using a radial basis function (RBF) followed by cylindrical projection of the morphed source mesh to the target. A local coordinate system is then established for each vertex of the deformed source mesh so that the matrix that transforms each vertex to its undeformed state can be computed. In the second step, the target mesh is projected to the source mesh (using an RBF and cylindrical projection) so that the motion vector of the transformed vertex can be interpolated from the motion vectors of the vertices of the source mesh. The estimated motion vectors of the deformed target mesh are transformed to the undeformed target mesh using the matrices computed in step 1. Lastly, in order to preserve the character of the animation between models with considerable differences in geometry, the magnitude of the target motion vectors are adjusted by a scaling factor determined by the bounding box of the polygons sharing a vertex. These two operations must be computed for each frame of animation. The average error of the vertices with motion was 4.07% - 8.56%, while the average errors owing to the size of the models tested were negligible.

2.2.2 *Rig transfer*

In contrast to the expression cloning method where an animation produced by a rig is copied, rig transfer involves copying the rig itself from a source to a target model. Owing to the variety of rigs in existence, no universal rig transfer technique is likely to, or is known to, exist. Therefore, each method of transfer is applicable to only one type of rig. Accordingly, a rig is deemed reusable only if a rig exporter for it exists.

Rig transfer tools were used by the Weta facial team (of 12 people) in order to deliver up to 300 hero-level facial rigs for the *Navi* characters in the movie "Avatar", as the traditional approach of manual rigging was not feasible in a project of such a scale Hellard (2010). Some of the facial animation techniques for which transfer techniques have been developed include:

Linear vector muscles

Bui *et al.* (2003) developed a technique for transferring linear muscle vectors (Waters (1987)) between head models. This method requires that a full set of landmarks be placed on a source head model and a partial set of easy-to-identify landmarks specified on the target head model. Initial guesses of the positions of the remaining, hard to locate, target landmarks are then made and a genetic algorithm used to iteratively adjust their positions. At each step of the iteration, a copy of the source mesh is morphed to the target mesh using the full set of landmarks and an RBF function, and the difference between both meshes is computed at a set of sampling points predetermined on the source model. The iteration continues until the difference is small enough and the final RBF function is used to transfer the start and end positions of each muscle vector from the source model to the target.

Fine-tuning of the computed target muscle vectors is performed using an interactive genetic algorithm that varies the parameters of the muscle vectors at each iteration and presents the user with nine versions of the same facial expression produced with different muscle parameters. The (muscle parameters used to generate the) three facial expressions selected by the user are used by the genetic algorithm to produce a new generation of muscles. This is done until the user is satisfied with the facial expressions.

Standard parametric rigs

Facial rigs commonly used by artists consist of control skeletons, painted weights, lattices and influence objects. Orvalho (2007) developed a series of techniques for transferring each component of a rig from one head model to another regardless of the configuration of the components or the quality of the rig as a whole. As a first step, the source model is deformed to fit the target using an RBF function computed on a set of landmarks placed on either object. Subsequently, dense correspondence is achieved by projecting every point of the warped source model to the closest point on the target model. A new RBF deformation function is computed based on the vertices of the undeformed source mesh and the dense correspondence mesh respectively. Because the skeleton, influence objects and lattices are defined by point-based attributes, they are transferred to the target mesh by the new RBF function. Because the copy of the source mesh and the target are in alignment after dense correspondence, skinning weights are copied from the aligned source mesh to the target. However, minor imperfections in the alignment of

the source mesh result in discontinuities in the weight map. This was solved by applying a smoothing filter to the target map generated.

A similar geometry matching and skin weights cloning utility based on the expression cloning technique was developed by the researchers at the Filmakademie Baden-Württemberg as part of their freely-available facial animation toolset[2].

2.2.3 *Rig building*

Rig building is characterized by the procedural generation of any number of rig components but may involve copying existing resources from generic models. Because rig building embodies rig creation and transfer in a single step, the problem of reusability ceases to exist. Facial animation techniques for which rig builders exist include:

Standard parametric rigs

The "Face Robot"[3], a closed-source commercial system developed by AutoDesk is one example of a rig builder. The Face Robot system generates a standard parametric comprising a simple system of bones and painted vertex weights for given 3D head models.

Bibliowicz (2005) developed a technique for adapting a standard rig (consisting of various bones systems, constraints, and deformers) to an arbitrary laser-scanned head model. Prior to rig construction, a reference head model (called "Murphy") having specific connectivity is conformed to the digitized face in three steps. First, an affine transform is used to conform the reference head to the laser scan such that the silhouette difference between the silhouette of the reference mesh and the outer hull of the point cloud is minimized. Next, local alignment is performed on sections of the reference mesh containing the ears, nose and mouth in order to further minimize the distance between the point samples and the mesh surface. The result of each alignment is blended with the surrounding parts of the mesh. Finally, a global transform is used to further minimize the distance between the point cloud and the prototype mesh. Thereafter, the connectivity of the reference head mesh is used to direct a rig construction script that replicates the standard rig on the conformed reference head model. Because the rig construction script is connectivity-dependent, it only works with the Murphy head model (or head models having the same connectivity).

Mass-spring systems

For each mass-spring system, earlier reviewed (in 2.1.4), there are techniques for transferring each mass-spring system from a pre-rigged generic model of the human head to a range scan of a live-subject. In each technique, a generic model of the human head is

2 http://wiki.animationsinstitut.de/doku.php?id=facialanimationtoolset:fat2
3 http://softimage.wiki.softimage.com/index.php/Category:FaceRobot

conformed to a laser scan of a real-life subject. Thereafter, the conformed model is used instead of the range scan. Range scans are replaced because they are noisy, over sampled, contain holes and do not capture some parts of the facial anatomy such as the eyes, eyelids, inner part of the lips, mouth and teeth. In contrast, the generic head model has well defined features, is efficiently triangulated and incorporates a mass-spring system. All mass-spring transfer techniques however differ in the manner fitting is performed.

For example, the transfer technique developed by Lee *et al.* (1993) (see also, Waters and Terzopoulos (1991), Terzopoulos and Waters (1993), and Lee *et al.* (1995)) aligns the planar version of a generic face mesh to the edges and facial features detected in the laplacian range image of a live subject. Fine-tuning is achieved by allowing the springs in the planar mesh to minimize their deformations, while using the feature points as immobile boundary conditions. Thereafter the conformed planar mesh is transformed to 3D by sampling the range image at the nodes of the planar mesh.

In the "Head shop" technique Kähler *et al.* (2002) (see also Kähler (2003)), the head model is initially fitted to the range scan using a thin-plate spline (TPS) function computed from a sparse set of anthropometric landmarks placed on the source and target surfaces. Because the initial fit does not capture the details of the facial geometry, new landmarks are generated for both surfaces by constructing a "feature mesh" from both sets of (source and target) landmarks. Each face of the target feature mesh is refined by subdivision and projected to the surface of the target mesh to form a new landmark. The same operation is performed on the source mesh. (Both feature meshes are in alignment after the initial morph.) Using the new (larger) set of landmarks, a new TPS deformation function is used to re-fit the source mesh to the target mesh. The feature meshes are refined in order to generate additional landmarks and the process is repeated until the source model is well adapted to the target. Because the landmarks on the skull are related to their counterparts on the skin by an offset vector, the landmarks of the target skull are computed by displacing each target landmark inward by an amount corresponding to the thickness of skin at the landmark. Both sets of skull landmarks (created from the deformed and undeformed head models) are used to create a TPS function that fits the generic skull to the deformed generic head model. Finally, as muscles are specified by a grid painted on the skin, the muscles are constructed as earlier described (in Kähler *et al.* (2001)) after the skull and head models have been generated.

Zhang *et al.* (2005) (see also Zhang *et al.* (2006)) develop a multilayer deformation technique in which affine functions are used to align a generic head model to the laser scan model at a set of anthropometric landmarks. In the next stage of fitting, the surface of the generic mesh is converted to a mass-spring system and subjected to local forces generated by the vertices of the laser scan model with the goal of minimizing the distance between both surfaces. In order to prevent the fitting from being stuck in a local minima, the set of anthropometric landmarks on the target model exerts an additional global force on the generic mesh. The key points defining the linear, sheet and sphincter muscles are transferred to the target head by a series of barycentric interpolation and projection

steps. The skull fitting step involves performing an initial TPS morph and the creation of new landmarks by successive refinement of the feature mesh, similar to the head shop algorithm.

Unfortunately, because models produced by artists have regular, painstakingly optimized topologies, they are not suited to workflows based on geometry substitution.

Position-based dynamics model

The basic building block of the musculoskeletal system developed by Fratarcangeli (2008) is a prototype skull model and a rectangular slab of geometry uniformly subdivided along its length and breadth. Muscle construction is an interactive process whereby a user sketches a closed contour consisting of piecewise curves on visible anatomy. The contour is sampled at regular intervals to match the subdivisions of the muscle slab and an RBF-based algorithm is used to fit the bottom face of the slab to the closed contour. After fitting, the vertices of the muscle slab are constrained to sample points of the contour, so that when the contour's sample points are displaced (simulating muscle contraction) the corresponding vertices of the slab move along with them. Each vertex of the slab is given a unit mass and a set of constraints that resist in-plane compression, tension, traction, shear, twist and bending. The flexibility of a muscle is determined by the strength of its constraints. This technique can be used to construct sheet and sphincter muscles. The result of the interactive muscle construction process is a hierarchy of overlapping muscles called a muscle map in which each muscle is constrained to an underlying structure such that the displacement experienced by a muscle or bone is propagated to the overlying structures. The muscle map and prototype skull are morphed to fit a given head model using an RBF deformation function computed from a set of landmarks placed on prototype skull and muscle map.

The vertices of the head model are treated as unit masses and are connected to each other by a series of stretching, twisting, bending, area and volume preservation constraints. In addition, elastic constraints are used to bind the surface mesh to the underlying structures. An influence map is used to limit skin deformation to the face region. (The influence map is constructed by firing rays normally inward from the vertices of the head mesh toward the underlying structures. A vertex is under the influence of the muscle geometries and jaw if a ray projected from it hits the muscle geometries or jaw or if the vertex is close enough to a vertex whose ray hits a muscle object or jaw.)

Skin simulation is performed using the position-based dynamics (PBD) method Müller *et al.* (2007). Unlike the mass-spring technique, the PBD method does not require the calculation of nodal velocities and accelerations and therefore avoids the problems of overshooting associated with numerical integration. Instead, the position of each node is directly updated. Furthermore with the PBD technique, constraints are easier to handle and can be represented as inequalities while collisions are easy to resolve. In addition, the system can be rolled back in time by using a negative time step.

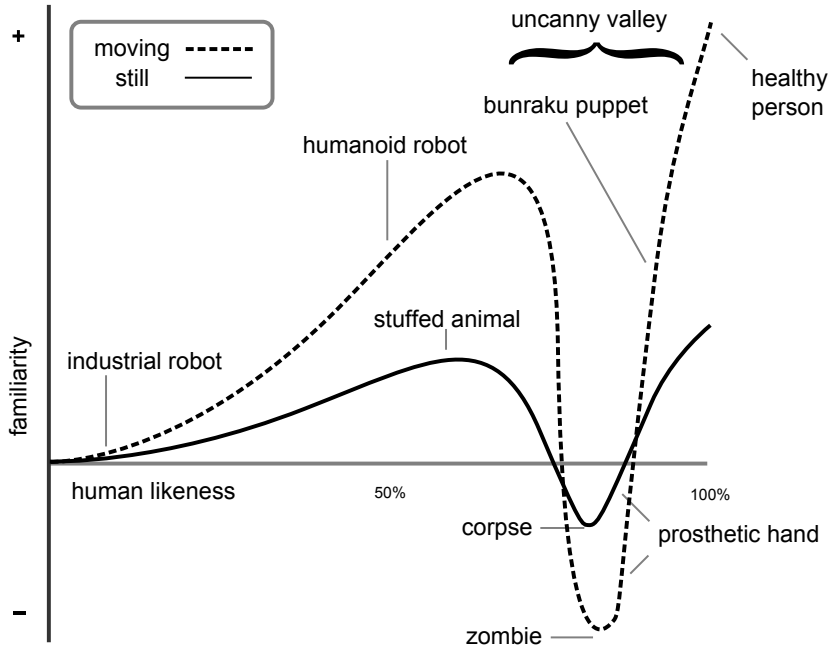


Figure 2.1 Hypothetical plot of the human emotional response to increasing degrees of human likeness of an entity. The plot shows a dip in the level of human comfort triggered by the excessive addition of human-like features. (Source: the Wikimedia Commons.)

2.3 THE UNCANNY VALLEY

Notwithstanding the developments in facial animation technology, synthesizing lifelike representations of the human form still represents an enormous challenge. Ironically, many of the technologies intended to make virtual characters appear more real have contributed to making them sinister-looking or trigger revulsion. Roboticists also encountered a similar problem while trying to build humanlike robots. It was observed that adding detailed humanlike features such as eyes, skin, and hair made the robots more appealing up to a point, after which they appeared unsettling. This law of diminishing returns was summarized by Masahiro Mori in his uncanny valley hypothesis illustrated by a plot of the human psychological response to representations of the human form. The plot, shown in Figure 2.1, shows that the familiarity (or level of comfort or acceptance) of an entity increases with human likeness up to a point, after which additional increases in human likeness results in a marked reduction in familiarity. Mori's chart however suggests that a sense of familiarity can be restored by further increase in human likeness.

At first glance, the dip (or valley) in the plot of familiarity versus human-likeness appears counterintuitive (adding humanlike features to an entity should always make it more humanlike). However, the human expectation of a form scales with the degree to

which it appears human. In other words, it is natural to be critical of entities that appear distinctly human, while overlooking greater flaws in the entities that only look remotely human. This primal response or aversion to the unreal is thought to be necessary for the avoidance of pathogens in infected hosts and corpses Green *et al.* (2008).

Uncanniness can result from flaws in the appearance (e.g. eyes, skin) and/or movement (primary or secondary) of an entity. For example, a representation that looks humanlike but does not move accordingly will be perceived as zombie-like or even lifeless. Furthermore, because to animate literally means "to give life to", an outcome of this sort indicates a failure in animation.

Although some questions have been raised concerning the existence of a valley[4], the hypothesis still offers a good explanation for the apparent paradox experienced by artists and roboticists, and presently informs the development of robots and virtual characters.

On one extreme, Pixar Animation Studios avoids the valley by favoring cartoon-like characters over realistic human representations in its movies. This decision also appears to have helped the company focus on creating characters that emote, instead of replicating the human form.

Alternatively, because the uncanny valley hypothesis as originally described by Mori strictly relates to representations of the human form, another approach is to limit representations to fantasy characters and thereby sidestep the deepest recesses of the valley as was done for the *Navi* characters in the movie "Avatar". Because the *Navi* characters were intended to be non-human (on account of their blue-tinted skin, cat-like noses, eyes and ears), the uncanny valley hypothesis suggests that the characters would not be subjected to the same degree of critical appraisal that would otherwise be accorded if they were intended to look human. However, this is not to suggest that humanoids are trivial to represent and cannot be judged as having imperfections. What is important to note is that a humanoid only needs to look alive (as though it truly exists) in order to be a convincing or acceptable representation; whereas a human character has to look like a living human. (Note that although human likeness encompasses lifelikeness, the opposite is not true.) For reasons earlier discussed and because humans are experts at deciphering non-human traits, a convincing human appearance is harder to achieve than a generic lifelike appearance. (The Gollum character in the movie Lord of the Rings movie is arguably another example of sidestepping the depths of the valley, although the revulsion stimulated by grotesque physical features of the character must be distinguished from that which is engendered by a non-grotesque representation that falls short of attaining the ideal of human likeness.)

The third and most challenging approach to the valley is to attempt to span it by creating humanlike representations of the human form. However, efforts to overcome

4 For example, Geller (2008) presents an image of a flesh-toned robot that looks more appealing than that of an unhealthy-looking human, and uses these images as the basis for a plot of familiarity versus human-likeness that peaks at the robot. However this analysis is flawed because beyond the valley in Mori's graph is a healthy-looking person.

the valley may be unsuccessful or only marginally successful (as in the movies Polar Express, Final Fantasy: the spirits within and Beowulf respectively). Surveys conducted by Lewis (2005) show that while "viewers are able to correctly identify most (but not all) synthetic images in only 1/4 second", there was little consensus and considerable vagueness in the feedback given on the flaws of the digital humans, and that the task of producing plausible virtual humans will at best remain difficult if the shortcomings of existing attempts cannot be precisely identified.

A few studies have been conducted in order to identify the factors that affect the perceived eeriness of a CG character. For example, MacDorman *et al.* (2009) systematically reduced the photorealism of an unanimated (semi-realistic) CG face in order to determine the extent to which various factors affect its eeriness. They found that more-human looking CG faces had more photorealistic textures and level of detail while experiments by Wallraven *et al.* (2005) showed that if motion information is made available, degrading the resolution of facial shape and texture does not significantly diminish cognition. The latter result confirms the conclusions reached by Maddock *et al.* (2005), that "for agents and conversational avatars, movement and behavior are more important than photorealism". Nevertheless further research needs to be done in order to identify specific deficiencies in existing virtual humans. This can be achieved by gradually degrading individual features (or pairs of features) of a realistic CG face in order to estimate the relative contribution of each feature toward achieving photorealism. Note that this experiment differs from that of MacDorman *et al.* (2009) as it is based on a realistic-looking (humanlike) CG character. Knowledge of the relative importance of individual characteristics should help artists decide how to prioritize their character development efforts.

2.4 APPLICATIONS

Although the applications of facial animation technologies span a wide application domain, the impact and rate of adoption of the technologies vary considerably between the application domains. In this section, the typical application domains are examined, in addition to the impact, rate of adoption and primary obstacles to a wider adoption of facial animation technologies within each application area.

2.4.1 *Film*

The entertainment industry and in particular the film industry is the main driver and consumer of facial animation technologies. Facial animation techniques are used to drive virtual characters acting in leading, supporting or background roles. Virtual crowds are easier to create and cheaper to kit and choreograph than a crowd of hundreds or thousands of real human extras.

In visual effects work, facial animation is often used to create digital stunt doubles, or to digitally age or de-age living characters, especially when prosthetic makeup is unconvincing, impractical or unsafe. Some facial changes such as aging are subtractive (aging is associated with tissue loss), and cannot be convincingly modeled by additive processes such as prosthetics. Prosthetics often gives the face a stiff (uncanny) appearance especially when the makeup is unaffected by the contraction of facial muscles. In addition, chemicals in prosthetic makeup can clog pores and irritate skin and the process usually takes several hours to apply and remove and must be reapplied daily. Also, the faces of fantasy creatures can be so different from the human form as not to be realizable by the application of prosthetic makeup to a human face.

The creation of fully realistic animated CG humans is considered to be the holy grail of computer animation. This would make it possible to cast famous actors of yesteryears (e.g. Humphrey Bogart) or younger versions of living actors in modern movies. However, it is possible to misuse the technology by creating realistic digital humans in situations where real humans can easily be used.

2.4.2 *Video games*

Although the earliest games had rudimentary or no facial animation, the current trend is toward an increase in the use and quality of facial animation in games. This trend is evidenced by the growing number of facial animation outsourcing companies[5], and is driven by the increased use of dialog and cinematics or cutaways (also known as cut-scenes). Cinematics are non interactive sequences used to set the stage for and advance storylines in a game. It is not unusual for a modern game to have several hours of cinematic cut scenes. For example, the game "Star Wars: The Old Republic . . . has more dialog than any other game ever made - and is probably the largest voice-over project ever - not only for games but also including movies and TV shows" Cloward (2010).

Because physically-based facial animation techniques are either too slow or hard to direct, the choice of technique is often between blend interpolation and skeleton-based mesh skinning methods. Blend shapes are intuitive to use and generate predicable results, however, the number of blend shapes that must be stored per character exceeds the capacity of current games consoles and graphics cards. On the other hand, mesh skinning requires the storage of only a finite set of vertex weights but is less intuitive and predictable. A combination of both techniques is possible by constraining a bones system to a readily sculptable, low-resolution base mesh from which blend shapes are constructed Watanabe (2010). The set of parameters corresponding to a blend shape is computed from the joint transformations of the bones system constrained to it. The same process can be used to convert facial motion capture data into streams of parameters (or

5 For example, Alter Ego (www.studiopendulum.com/alterego), CaptiveMotion (www.captivemotion.com), Cubicmotion (www.cubicmotion.com), Image Metrics (www.image-metrics.com), and MOVA (www.mova.com).

animation curves). Again, the amount of memory available on game consoles limits the amount of motion capture data that can be used.

Due to resource limitations and the large number of tasks that games must perform in a fraction of a second, in-game facial animation is generally of poorer quality than feature film facial animation. Furthermore, the quality of rigs and other assets is sometimes scaled back in order to maintain interactive frame rates and game play, and ensure that the game has similar performance on all platforms targeted. The poor quality of in-game facial animation is also due to a lack of effective algorithms for automatically generating secondary facial animation (e.g. blinks, random head movements) and performing lip synchronization. Fortunately, increasingly sophisticated graphics and game-play appears to make consumers tolerant of poor facial animation. In feature film production, most of these shortcomings are masked by teams of artists who tweak animation till it is near perfect. Moreover, in feature film production, render times are a lot more generous and there is almost no limit on the amount of computing power that can be brought to bear.

2.4.3 *Medicine and surgery*

The traditional methods of planning maxillofacial surgeries utilize profile x-ray images or solid stereolithographic models of patients. Unfortunately, x-ray images provide a limited view of a patient's anatomy while solid models are slow and costly to build and do not provide any soft-tissue information. In order to overcome these limitations, more recent planning techniques construct patient-specific virtual anatomy and mathematical models of soft and hard tissue behavior from medical imaging data (e.g. Keeve *et al.* (1996), Koch *et al.* (1996), Deuflhard *et al.* (2006), and Koch *et al.* (2002)). These models allow surgeons to predict the post-operative appearance and facial expressions of patients, simulate or experiment with various surgical approaches (and select a strategy that will produce the most pleasant facial appearance), and, help reduce intra-operative decision making and therefore surgery time[6]. The amount of tissue growth that occurs in response to surgical trauma can also be accounted for in order to obtain a better estimation of facial contour months after post-surgical swelling subsides Vandewalle *et al.* (2003). Due to the need for accuracy, surgical applications are exclusively dominated by and require accurate physically-based methods and a degree of interactivity.

2.4.4 *Human-computer interaction (HCI)*

Humans have an innate ability to relate to other faces, and become skilled at doing so long before they learn to read or use other interfaces. Therefore, machines with human-like interfaces (e.g. faces and voices) are rightly thought to be the pinnacle of artificial intelligence and human-computer interaction. By presenting a familiar and universal

6 Examples of facial surgery planning software include: Amira (amira.zib.de) , Axis-Three (www.axisthree.com/professionals/products/face-surgery-simulation) and Surgicase (www.materialise.com/CMF)

interface, such as the human face, such machines require very little or no training to use, and can boost user performance as measured by the proportion of valid responses and the length of time spent interacting with the system Walker *et al.* (1994).

However, the utility of such interfaces depends on factors other than the quality of facial animation. For example, conversational agents and other applications that solicit full (two-way) interaction require advanced speech synthesis, recognition and artificial intelligence beyond what is currently possible in order achieve acceptable levels of service. Moreover, because the goal of HCI is not to give the impression that an agent is human, realistic faces are not essential and could even be counterproductive if they overly raise the expectations of users.

Fortunately, the current of level speech processing technologies is sufficiently advanced for the development of non-interactive (one-way) applications such as electronic guides, virtual newscasters, social agents and low bitrate MPEG4 video-conferencing Pandzic and Forchheimer (2003), Eisert (2003)).

2.5 SUMMARY

Facial animation production for visual effects, games and character animation today is almost entirely based on the parametrization of facial models by rigs, which are either manually keyframed or driven by a live performance. However, as earlier discussed, rigs are often the weakest link in this pipeline, as they determine quality and fidelity of the animation, irrespective of the accuracy of the driver or motion source. Unfortunately, the construction of sufficiently accurate facial rigs is anything but trivial. This has led to research into ways of reusing facial animation. Physically-based facial animation techniques share some of these limitations, and although some reusable techniques have been developed, these techniques are generally not thought to be production-friendly because they are often very slow and not "art-directable". Furthermore, the level of anatomical detail as well as the constitutive and structural models of soft tissue employed in most studies is often overly simplistic, and possibly limits their use in facial surgery applications, where accurate prediction is essential.

The work done by Barbarino *et al.* (2008) (see also, Barbarino *et al.* (2009)) is one of the few exceptions. However, as earlier mentioned, as their technique is reliant on medical imaging data, it is not immediately applicable to CG heads. However, the required internal anatomy can be procedurally generated for CG models. This is the primary objective of this thesis – the generation of anatomical detail in a reusable framework, that is art-directable. This acknowledgement of the significance of the work done by Barbarino *et al.* (2008) however should not suggest that this thesis is an extension or refinement of their research.

I am fearfully and wonderfully made
— Psalm 139:14

3

ANATOMY OF THE HUMAN FACE

This chapter draws extensively from up-to-date research in the field of plastic and reconstructive surgery in order to present a detailed description of the anatomy of the human face beyond what is available in standard references. These descriptions inform the development of algorithms for the automatic generation of detailed anatomical substructures and the more-accurate physically-based model for facial animation.

The anatomy of the human face is presented in an inside-out fashion, starting from the smooth bony substructure known as the skull, because it is from this mass that the face derives its form and to it the overlying soft tissues are attached. The review of the anatomy of soft tissues does not include the vascular (blood supply) system and innervation (distribution of nerves) of the human face, as the function of these tissues is not structural.

3.1 SKELETAL ANATOMY

The skull is the most complex part of the human skeleton. Far from being one bone, it is made up of 29 separate bones that form the braincase or cranium, the facial cavities (orbital, nasal, and oral), and the lower jaw. The bones of the skull meet at joints known as sutures, many of which derive their names from the two bones that form them. The largest of these bones, shown in Figure 3.1, include the:

FRONTAL BONE - forms a variety of structures such as: the forehead, roof of the orbits (eye sockets), top of the nasal aperture, and the floor of the frontal lobes of the brain. Each side of its outer surface is marked by two kinds of protrusions. The first, a rounded elevation known as the frontal eminence, marks the (original) centre of growth (ossification) of this bone. (In the full-term fetus, the frontal bone consists of two separate parts, which eventually fuse.) The second type of protrusion is ridge-like, and runs along the upper part of each orbit, forming the eye brows (more prominent in males). This ridge is variously referred to as the supraorbital ridge, margin or arch.

The lateral side of the external surface of the frontal bone is marked by two (inferior and superior) temporal lines, identifying the attachment of the temporalis muscle (see Section 3.2.4) and its covering fascia respectively. (The temporal lines are more prominent in adults.) The frontal bone articulates with twelve bones including the (suture names given in parenthesis): parietal bones (coronal suture), nasal bones

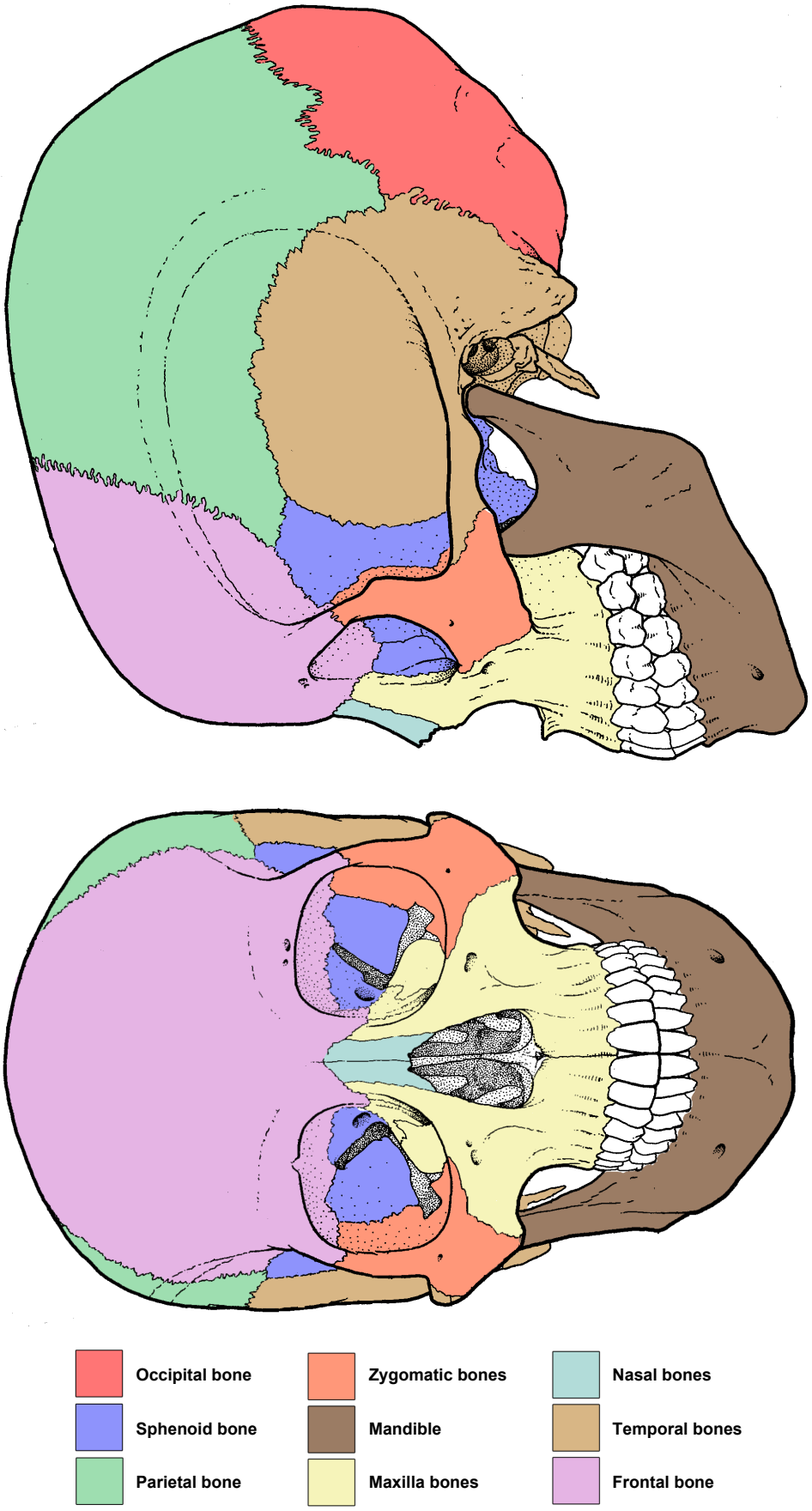


Figure 3.1 Nine major bones of the human skull.

(fronto-nasal suture), maxillae (fronto-maxillary suture), and zygomatic bones (fronto-zygomatic suture).

PARIETAL BONES (paired) - smooth, convex, square-like bones that form the sides and roof of the cranial vault. Each bone has a large, rounded (parietal) eminence that marks the centre of ossification of this bone. The length of its external surface is marked by both temporal lines. Each parietal bone articulates with five bones including: its sister parietal, the occipital bone, the frontal bone, and the temporal bone.

TEMPORAL BONES (paired) - highly irregular in shape, form the transition between the cranial wall and base, house the organs of hearing and form the upper surface of jaw joints. Originating from above the external auditory meatus is a thin projection of bone that forms the posterior half of the zygomatic arch, and unites with the zygomatic bone. The temporal bone articulates with five bones including the: occipital, parietal, mandible, and zygomatic bones.

OCCIPITAL BONE - forms the posterior wall of the brain case, and is the bone surrounding the foramen magnum (Latin: 'great hole' - one of the several apertures at the base of the skull, through which the spinal cord enters and exits the skull vault). Various muscles of the neck and other fascia attach to the convex external surface of this bone, and faintly mark its surface at various nuchal (pertaining to the nape of the neck) lines, including the, highest, superior, inferior nuchal lines (on either side of the midline), and the medial nuchal line. In addition, an external occipital protuberance can be identified on this bone, where the neck meets the skull. (At times, the external occipital protuberance appears to be an extension of the superior and/or the medial nuchal lines.) The occipital bone articulates with six bones including the parietal and temporal bones.

SPHENOID BONE - the most complicated of all the bones of the cranium, situated between the bones of the cranial vault and those of the face. This bone completes the braincase, and forms much of the roof and sides of the orbital wall. The sphenoid articulates with twelve bones which include: both parietal bones, both temporal bones, both zygomatic bones, the frontal bone, and the occipital bone.

NASAL BONES (paired) - small, rectangular bones forming the bony roof of the nose superiorly. The nasal bones articulate with four bones including the frontal bone and maxillae.

MAXILLA BONES (paired) - form the dominant portion of the lower face and cheeks, surrounds the nasal cavity, and contribute to the inferior border and floor of each orbit, and form the medial portion of the infraorbital margin. The maxillae, also hold the upper tooth roots, and form most of the hard palate. Each maxilla articulates with eight bones which include the frontal bone, nasal bone, and zygomatic bone.

ZYGOMATIC BONES (paired) - also termed the malar, form the lateral margin of the orbit and part of the floor of the orbit. The zygomatic bone also forms the anterior part

of the zygomatic process, the lateral portion of the infraorbital margin, and the prominent corners of the cheeks. Each zygomatic articulates with four bones which include the frontal, maxilla, and temporal bones.

MANDIBLE - horseshoe-shaped bone, otherwise referred to as the lower jaw, is the largest and strongest bone of the face, houses the teeth of the dental arcade and provides a surface for the insertion of various muscles of mastication and movement of the lips. Its relatively smooth external surface is marked in the median line by a faint ridge, indicating the symphysis or line of junction of the two pieces of which the bone is composed at birth. This ridge divides below and encloses a triangular mental protuberance or eminence (more pronounced in males), the base of which is depressed in the centre but raised on either side to form the mental tubercle.

3.1.1 *Craniofacial landmarks*

The size and relative prominence of certain skull features provide some indication of gender, age range, and ethnicity (Steele and Bramblett, 1988; Krogman, 1962). These features form convenient baselines or landmarks for taking measurements of and comparing skulls. For this reason, landmarks of the skull are sometimes referred to as cranial osteometric, i.e skull measurement, or craniometric points.

A landmark identifies (a) a point at which three structures meet, (b) maxima of curvature of some local morphogenetic process, or (c) an extremal point (Bookstein, 1991). Wherever facial tissue is relatively thin, the latter two landmark types may be reflected on the face, and examined by palpation (or touch). Landmarks of the face, or cephalometric points as they are known often share the same name as their cranial counterparts, but may not derive from the same anatomical position Farkas (1994).

3.1.2 *Facial Tissue Depth (FTD) Measurements*

Because the size and relative prominence of some skull bones and features provide indication of gender, age range, and ethnicity, as previously stated, a likeness of a person can be constructed from a skull, given sufficient data on facial tissue thicknesses. Such reconstructions are of considerable value to forensic scientists and anthropologists, and is perhaps the singular motivation for the measurement of FTDs. The older methods of measuring FTDs typically involve inserting a soot-covered needle, or a needle with a rubber stopper, or some other sharp object into a cadaver at various facial landmarks. However, because volumetric tissue changes (e.g. shrinkage as a result of drying, or swelling due to embalming) always occur shortly after death, FTD measurements obtained from cadavers using needles are unreliable. In contrast, modern measurement techniques are non-invasive, and can therefore be used on living subjects. Such techniques employ sophisticated technologies such as, X-rays, magnetic resonance imaging (MRI), computed

tomography (CT), or ultrasound. A comprehensive survey of the history and comparison of various FTD measurement techniques and sample data is well documented (Wilkinson, 2004).

3.2 MUSCULAR ANATOMY

Muscles consist of contractile, fibre-like cells collected into bundles known as *fascicles*, groups of which make a muscle belly Williams *et al.* (1989). Muscular forces are produced by the contraction of individual cells and the amount of force produced depends on the number of cells that are active. Furthermore, muscle cells do not contract partially, but rather shorten to the full extent possible, sometimes up to about half their length, upon receiving a nerve impulse Gordon (1989). As such, muscle actions produce contractions and never an increase in length.

However, there are several fundamental differences between the muscles of facial expression (also known as mimic or mimetic muscles) and other skeletal muscles. Unlike skeletal muscles, mimic muscles are thin, sheet-like and therefore do not form the bulk of facial soft tissue. Also, mimic muscles are not attached to the skeleton at both ends but are often attached to the skull at an origin and insert to the soft tissue at the other end. As such, one end of a mimic muscle is almost always fixed to the skull while the other end is free to move. The risorius muscle (see page 34) is one such muscle that does not take its origin from the skull. Such uncharacteristic freedom allows the face to produce a vast array of actions and expressions essential for a range of functions such as speech and the expression of emotion. Muscles are attached to the skull via strong, inextensible and non-contractile parallel bundles of collagen fibres known as tendons. When a wide area of attachment is required, for example to the cranium, tendons exist in sheet-like forms called an *aponeurosis*.

The mimic muscle system consists of about 40 muscles many of which mingle or interdigitate with each other at the fibre level. Mimic muscles are arranged in layers, can exist in (independently acting) pairs (i.e. consist of a left and right half) and often act in concert with other muscles (this is the basis of the action unit and FACS concept). The major mimic muscles are listed below. Note that the masseter and temporalis (Section 3.2.4) are not mimic muscles. The following descriptions are primarily based on Williams *et al.* (1989); Gordon (1989); Goldfinger (1991); Hollinshead (1968); Bentsianov and Blitzer (2004); Patrinely and Anderson (1988), and to a lesser extent on Miller (1991); Hannam and McMillan (1994).

3.2.1 *Muscles of the upper face*

FRONTALIS (paired, see Figure 3.2(a)) - this is the anterior belly of a much larger occipitofrontalis or epicranius muscle covering the cranium. The posterior belly

of the occipitofrontalis is the occipitalis muscle. Both bellies are separated by a broad tendon called the galea aponeurotica which is divided into a superficial and deep plane or sheath. The frontalis muscle is paired and consists of a left and right half. Both halves originate from the deep galea plane and are enveloped by the superficial plane. The lateral margin of the muscles terminates just medial to the superior temporal line of the skull. The frontalis inserts at the skin at the eyebrow level and interdigitates with the orbital portion of the orbicularis oculi. Accordingly, the contraction of the muscle pulls the eyebrows upward and produces a look of worry or surprise.

CORRUGATOR SUPERCILII (paired, see Figure 3.2(b)) - this muscle is paired. Each muscle has its origin on the medial end of the orbital rim of the frontal bone. The muscle is located deep to the frontalis and orbicularis oculi and ascends laterally to blend with the aforementioned muscles at the medial aspect of the eyebrow. The basic action of this muscle is to pull the eyebrows down and bring them closer together, creating strong vertical or oblique skin folds and bulges on the glabella between the medial ends of the eyebrows. The corrugator is often associated with negative emotions such as anger, frowning and confusion.

PROCERUS (unpaired, see Figure 3.2(c)) - also known as the depressor glabellae or pyramidalis nasi, originates from the fascial aponeurosis covering the lower part of the nasal bone and adjoining nasal cartilages. The muscle courses the root of the nose and fans upward to insert in the skin between the eyebrows. The muscle pulls down the skin in the centre of the forehead forming transverse wrinkles in the glabella region and bridge of the nose. This muscle contributes to the expression of anger and disgust.

ORBICULARIS OCULI (paired, see Figure 3.2(d)) - is a broad, flat elliptical muscle, each pair consisting of an outer orbital portion covering the orbital rim, an inner palpebral portion covering the eyelids and a small lacrimal portion (so called on account of its proximity to the lacrimal or tear duct) located deep behind the medial corner of the eye. (The lacrimal portion does not affect facial expressions.) The orbital portion is attached to the medial palpebral ligament at the nasal part of the orbital bone and forms complete loops around the orbital rim and returns uninterrupted to the medial palpebral ligament. Superiorly, the orbital portion blends with the frontalis and corrugator supercilii. Laterally, it overlies the anterior portion of the temporalis fascia. Inferiorly, its fibres overlap or partially blend with the levator labii superioris alaeque nasi, levator labii superioris and zygomaticus major. The palpebral portion starts from the medial palpebral ligament, sweeps across the eyelids and merges with the lateral palpebral ligaments laterally. (The medial and lateral palpebral ligaments anchor the soft tissues around the eye to the bone.) The orbital portion contacts to create a sphincter action that pulls the skin around the eyes medially toward the nasal ridge, partially closing the eye and creating “crows feet” wrinkles

and creating skin folds below the lower eyelid. When strongly contracted, the orbital portion bulges up the cheeks, deepens the nasolabial folds and mildly lowers the brow. When the palpebral portion contracts, the frontal, temporal and malar skin is drawn medially. This action also closes the lids completely by lowering the upper eyelid and raising the lower eyelid. The lower eyelid alone can be raised independent of the upper eyelid. Overall, the entire muscle contributes to the expression of happiness, pain, hostility or confusion.

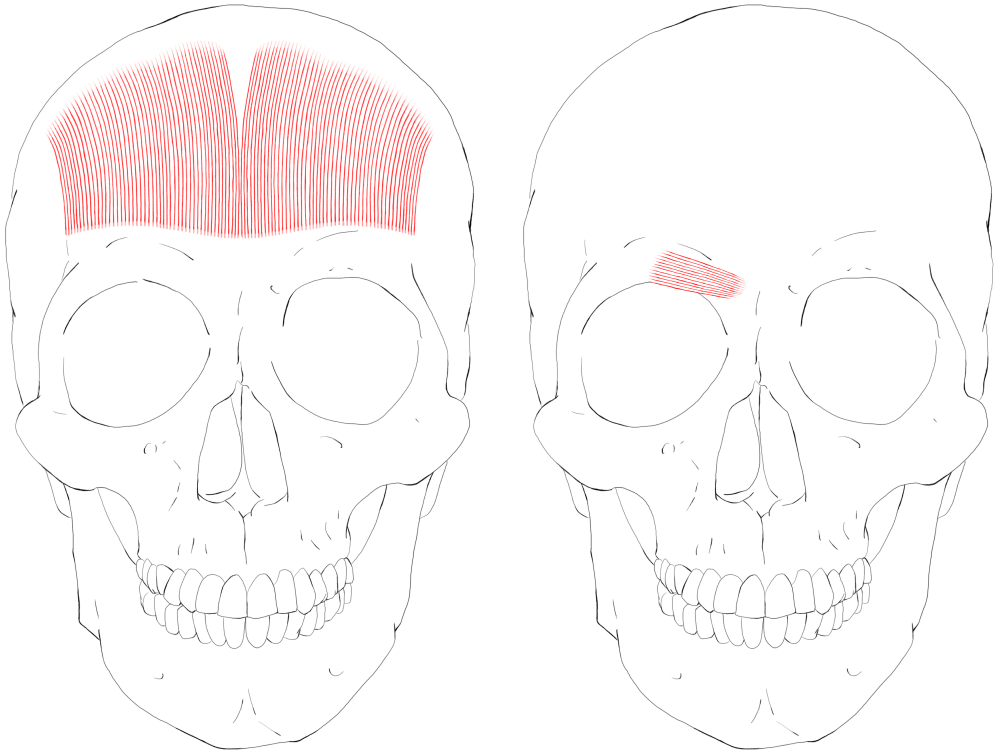
3.2.2 *Muscles of the mid face*

LEVATOR LABII SUPERIORIS ALAEQUE NASI (paired, see Figure 3.3(a)) - literally means "lifter of the upper lip and of the wing of the nose". This muscle takes its origin from the superior prolongation (or frontal process) of the maxilla , just medial to the corner of the eye. This muscle divides into a medial slip that inserts to the uppermost part of the nasolabial furrow, while the lateral portion passes obliquely downward to penetrate the mass of the orbicularis oculi near the philtrum. Contraction of the levator labii superioris alaeque nasi deepens and elevates the upper end of the nasolabial fold, mildly lifts the medial portion of the upper lip (as its name implies) and throws the skin on the side of the nose into folds. The muscle is essential for the expression of disgust.

LEVATOR LABII SUPERIORIS (paired, see Figure 3.3(b)) - is a broad muscle whose origin is along the lower border of the orbit, along the maxilla and zygomatic bone inferior to the orbicularis oculi. The levator labii superioris extends downward and somewhat medially to insert into the nasolabial furrow just lateral to the wing of the nose. Other fibres pass downward beyond the furrow to mingle with the fibres of the orbicularis oris. The contraction of this muscle raises the centre half of the upper lip and nasolabial furrow, producing a typical sneer.

LEVATOR ANGULI ORIS (paired, see Figure 3.3(c)) - sometimes referred to as the caninus, has its origin at the canine fossa, immediately below the infraorbital foramen and runs vertically. This muscle runs more deeply than other levators and passes downward to the corner of the mouth, where it merges with the orbicularis oris. As its name implies, the muscle pulls the corner of the mouth upward.

ZYGOMATICUS MINOR (paired, see Figure 3.3(d)) - arises from the zygomatic bone, below the lateral edge of the orbit and deep to the orbicularis oculi. The muscle descends medially into the middle section of the nasolabial fold. Some of its fibres continue medially to merge with the orbicularis oris. Contraction of this muscle pulls the medial portion of the nasolabial fold and the upper lip upward and outward and exposes the maxillary teeth. This muscle contributes to the expression of a smile or smugness.



(a) Frontalis

(b) Corrugator Supercilli

(c) Procerus

(d) Orbicularis Oculi

Figure 3.2

ZYGOMATICUS MAJOR (paired, see Figure 3.4(a)) - is a long, narrow, muscular band that has its origin at the lateral surface of the zygomatic bone. Its upper and lateral parts are covered by the orbital portion of the orbicularis oculi and the superficial portion of the zygomaticus minor. The muscle runs medially toward the corner of the mouth and merges with a variety of muscles including the levator anguli oris and orbicularis oris. The zygomaticus major pulls the lower face outward and upward, so that the entire cheek bulges and rises. Extreme contractions of the muscle pushes the lower eyelid upward, deepening the wrinkles under the eye and causes "crows feet" wrinkles to develop at the outer corner of the eye. This muscle is integral in the expression of smiles and laughter.

BUCCINATOR (paired, see Figure 3.5(c)) - is a relatively deep, quadrilateral muscle that forms the lateral wall of the oral cavity. It lies deeper than other facial muscles and arises from the alveolar process[1] of the maxilla above and the mandible below. The buccinator fibres converge toward the corner of the mouth, its central fibres cross over each other, so that the upper and lower fibres continue into the upper and lower parts of the orbicularis oris. The muscle pulls the corner of the mouth inward, flattening the cheek against the teeth, thus minimizing the passage of food between the cheek, and prevents the cheek from being unduly distended by air pressure, for example when blowing a wind instrument. For this reason, the buccinator is sometimes referred to as the "trumpeter's muscle".

RISORIUS (paired, see Figure 3.5(a)) - is often poorly developed and differs from other mimic muscles in that it does not arise from a bone, but from the fascia covering the parotid gland (the largest of the salivary glands) which lies in front of the ear over the ramus of the jaw. The muscle runs medially and inferiorly, partly overlying the platysma and inserts into the lateral corner of the mouth. When this muscle contracts, it pulls the corner of the mouth straight back to produce a grinning expression.

INCISIVUS LABII SUPERIORIS (paired, see Figure 3.4(b)) - is a narrow muscular band that takes its origin from a small area in front of the maxilla above the lateral incisor and inserts deep to the peripheral fibres of the orbicularis oris at the corner of the mouth. The muscle contracts to pull the corners of the lip upward and forward.

ORBICULARIS ORIS (unpaired, see Figure 3.4(d)) - so called because it was once assumed that its fibres formed complete ellipses around the oral fissure. Recent findings have however shown that, unlike the orbicularis oculi, the muscle is not a true sphincter (continuous circular muscular ring) and consists of up to eight independent quadrants Williams *et al.* (1989) that jointly allow it to function as such. The orbicularis oris is clearly the most complex of all facial muscles and covers the greater part of the lips, from the nose down to the groove halfway between the lower lip and the bottom of the chin. The fibres of the orbicularis oris run into the

1 The thickened ridge of bone that contains the tooth sockets.

red lip portion of the lips and also blend intimately with the muscles around the mouth which weave and insert into it. The contraction of the whole and various parts of the muscle brings the lips together in order to create an enormous variety of actions such as the closing, withdrawing, puckering and protrusion of the lips.

3.2.3 *Muscles of the lower face and neck*

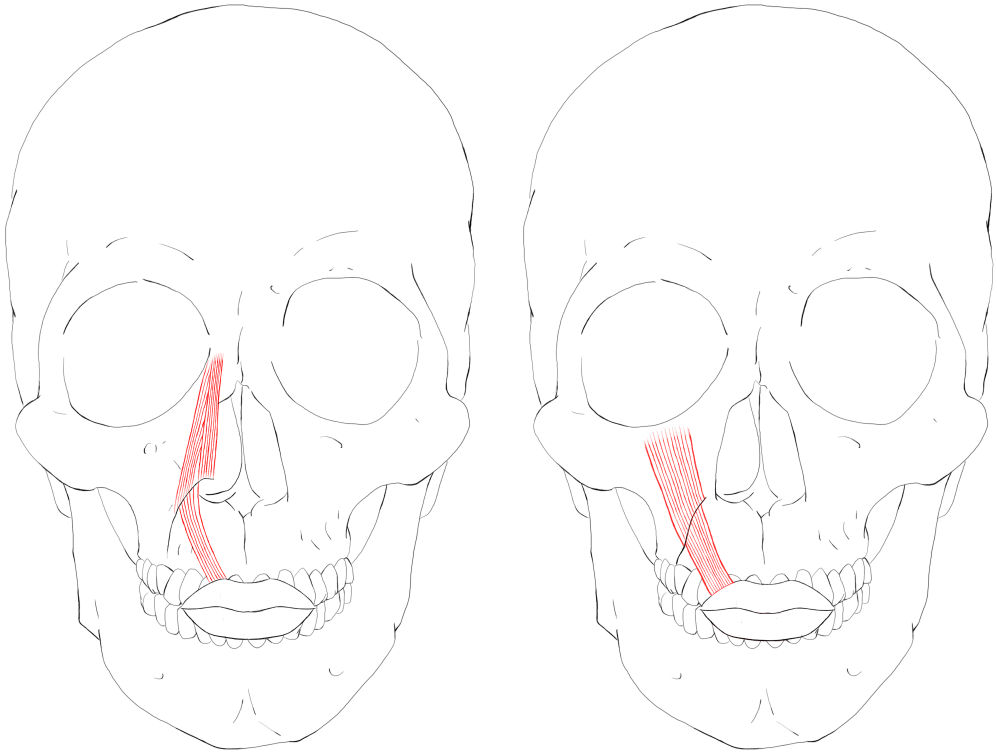
DEPRESSOR LABII INFERIORIS (paired, see Figure 3.5(a)) - also known as the quadratus labii inferioris or quadratus menti, this muscle takes its origin from an oblique line at the front of the mandible, and is partly covered by the fibres of the depressor anguli oris at its origin. The muscle lies deep to the platysma and curves laterally and upwards to blend with the inferior peripheral fibres of the orbicularis oris, and continue to the lower lip. The contraction of this muscle pulls or curves the medial portion of the lower lip downward, exposing the teeth and gums.

INCISIVUS LABII INFERIORIS (paired, see Figure 3.4(c)) - is a narrow muscular band that takes its origin from a small area in front of the mandible below both incisors and also inserts deep to the peripheral fibres of the orbicularis oris at the corner of the mouth. The muscle contracts to pull the corners of the lip downward and forward.

DEPRESSOR ANGULI ORIS (paired, see Figure 3.5(b)) - also known as the triangularis, takes its origin from an oblique line on the outside body of the mandible, inferiolateral to the origin of the depressor labii inferioris, and converges into a narrow area, blending with other muscles such as the orbicularis oris at the corner of the mouth. The muscle contracts to curve the angle of the mouth downward producing the classic look of sadness.

MENTALIS (paired, see Figure 3.5(c)) - is a short, deep lying, conical muscle arising from the mandible at the level of the root of the lower lateral incisor, midway between the lower lip and the bottom of the chin. The muscle passes downward to insert into the skin in front of the chin. The mentalis contracts to pull the chin upward, raises the lower lip and creates numerous wrinkles that dimple the chin.

PLATYSMA (paired, see Figure 3.7) - is a large, thin, wide muscular sheet that rises from the collarbones and upper chest and runs along the lower part of the face and the mandible to attach to the lower lip, where it blends with some of the muscles associated the lower lip, especially the depressor anguli oris and orbicularis oris. The muscle stretches and flattens the lips, especially the lower lip and tenses the neck, so that long, thin, parallel, raised ridges form and pass downward toward the laterally posteriorly from the lower edge of the jaw to the collarbone, widening the neck. This muscle is used to express fear or horror.



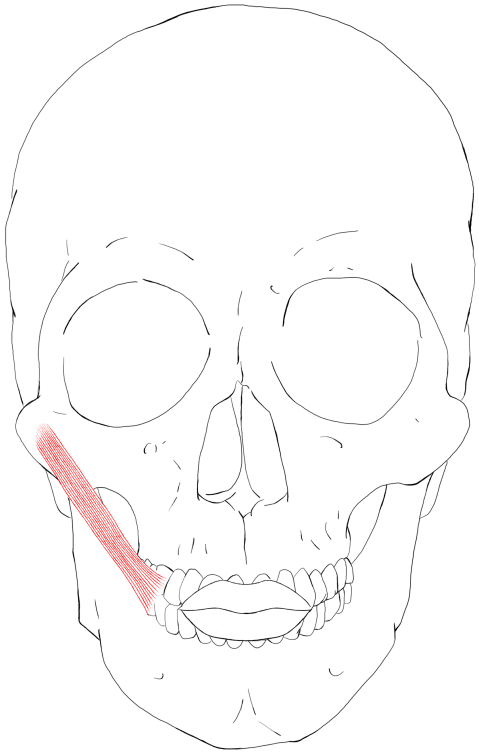
(a) Levator labii superioris alaeque nasi

(b) Levator labii superioris

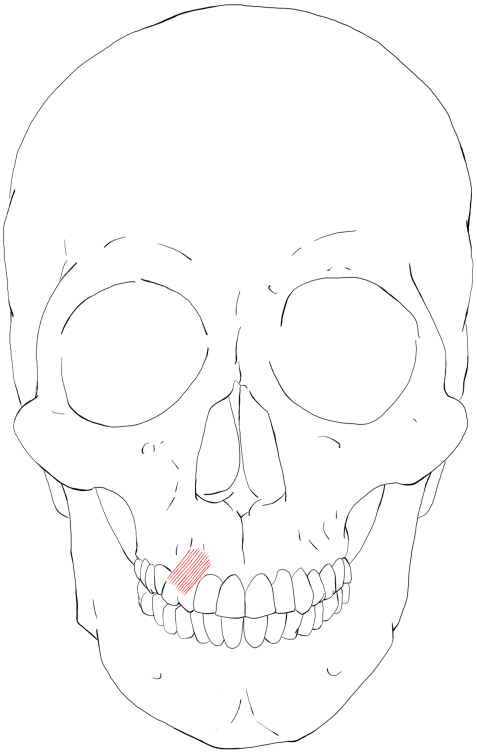
(c) Levator anguli oris

(d) Zygomaticus minor

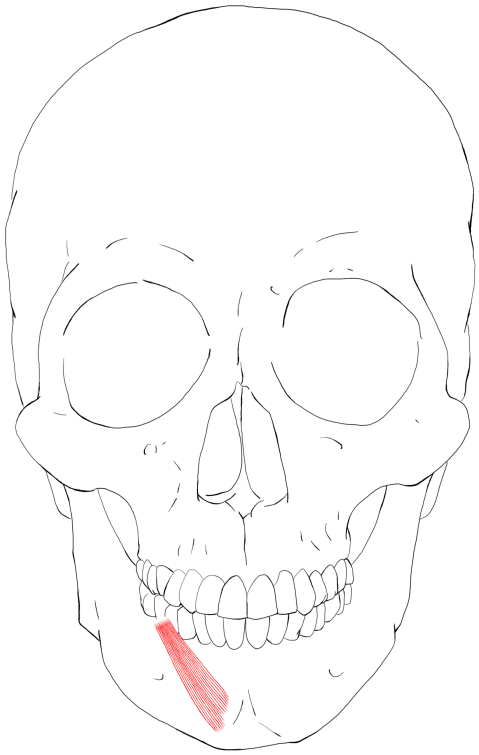
Figure 3.3



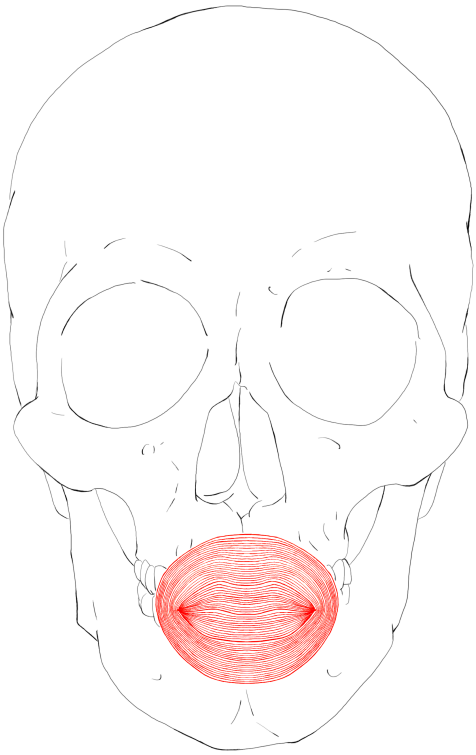
(a) Zygomaticus major



(b) Incisivus labii superioris

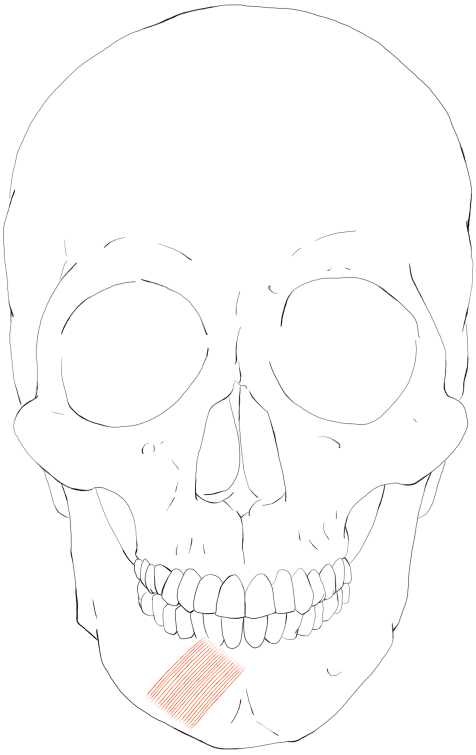


(c) Incisivus labii inferioris

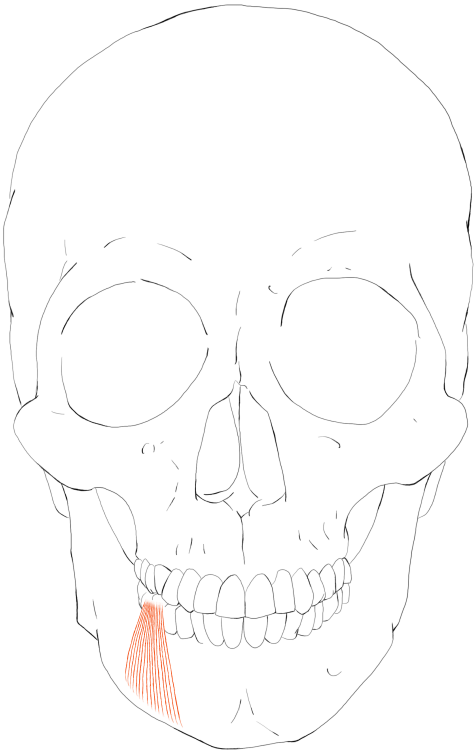


(d) Orbicularis oris muscles

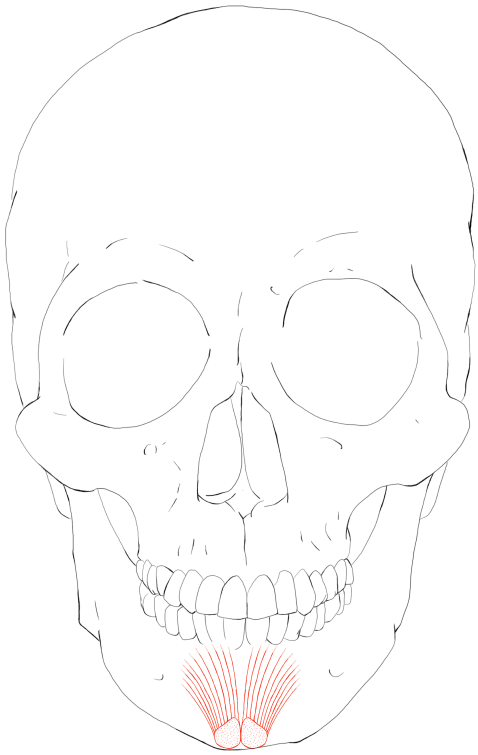
Figure 3.4



(a) Depressor labii inferioris

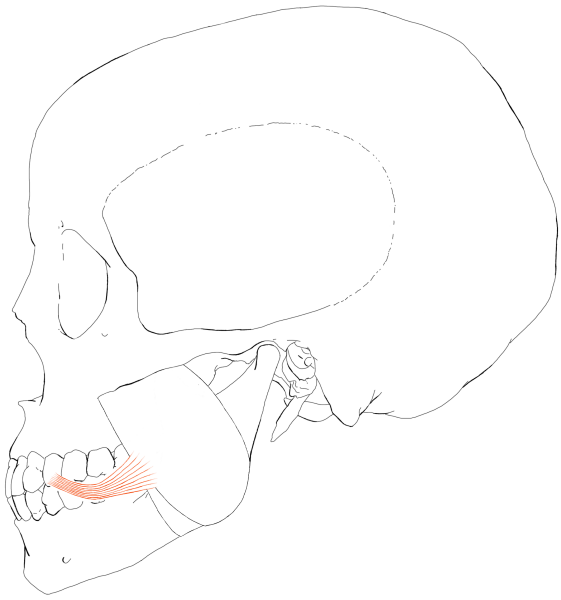


(b) Depressor anguli oris

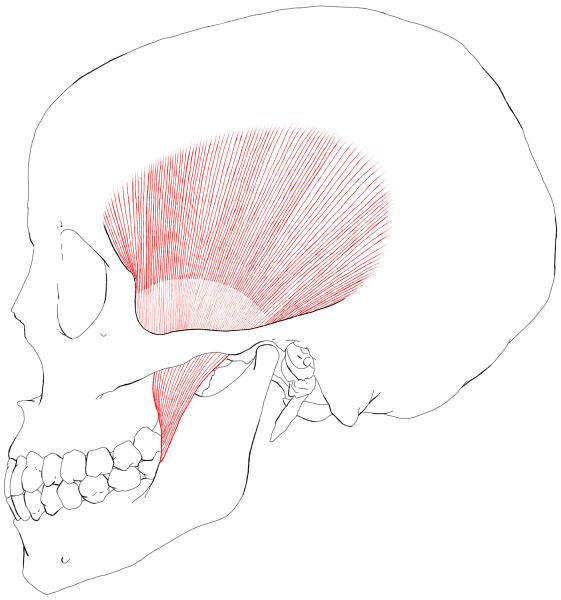


(c) Mentalis

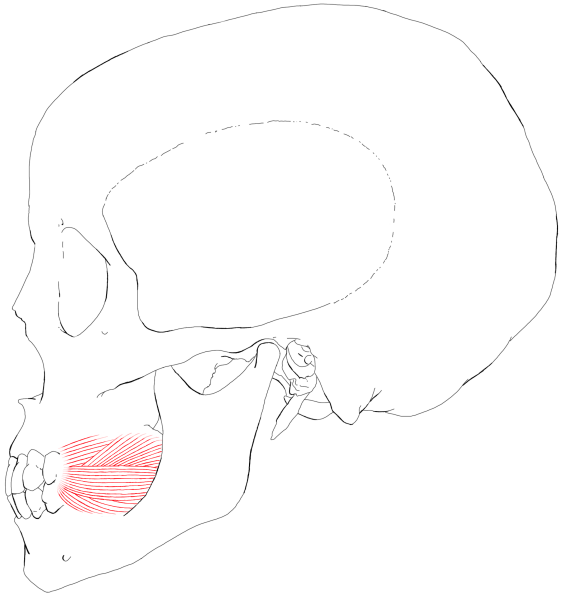
Figure 3.5



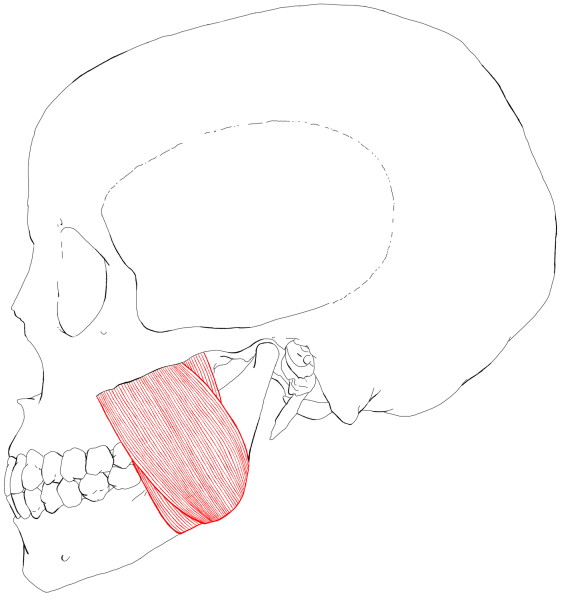
(a) Risorius



(b) Temporalis



(c) Buccinator



(d) Masseter

Figure 3.6

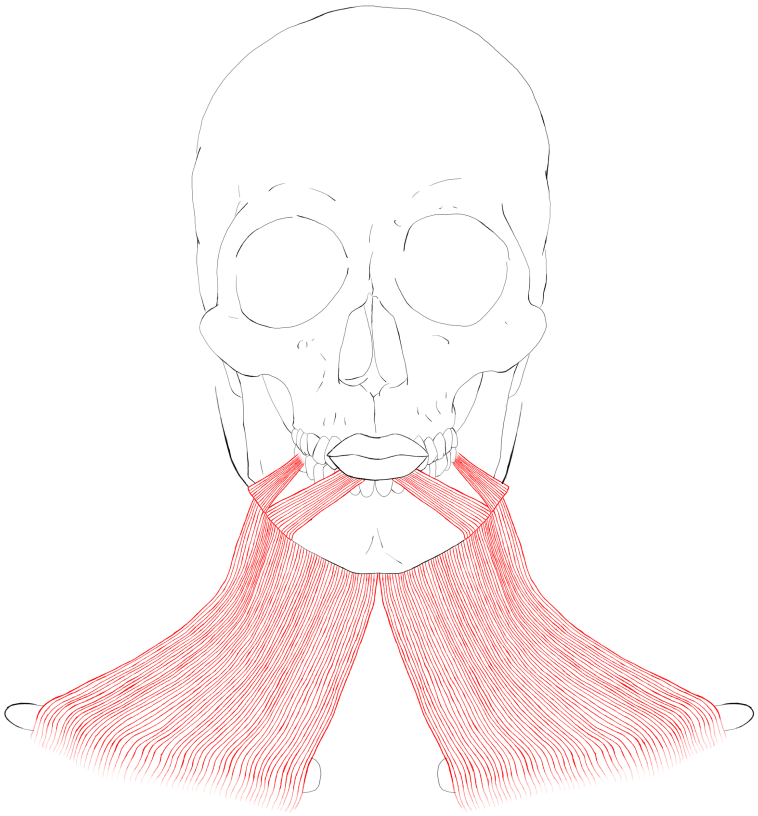


Figure 3.7 The Platysma muscle.

3.2.4 *Masticatory muscles*

TEMPORALIS (paired, see Figure 3.5(b)) - is a relatively-thin, broad, fan-shaped muscle originating just below the inferior temporal line on the side of the cranium. Its collection of long fibres fill the depression on the side of the skull and converge into a flat tendon that inserts into the coronoid process and the ascending border of the ascending ramus of the mandible deep to the zygomatic arch. The muscle is covered by a strong, dense layer of fascia (the deep temporal fascia) that attaches to the skull beyond the margin of the muscle at the superior temporal line. The temporal fascia leaves the surface of the temporalis to attach to the upper edge of the zygomatic arch. The temporalis is seen to bulge when the jaw is closed and sink in when the jaw opens.

MASSETER (paired, see Figure 3.5(d)) - is a thick, quadrilateral muscle, consisting of two overlapping parts. The superficial layer arises from the inferior border of the zygomatic process of the maxilla, while the deep layer originates from the anterior two-thirds of the inferior border of the zygomatic arch. The superficial part of the muscle slants backward, largely covering the deep part as both layers fuse to attach to the lateral surface of the ramus and gonial angle of the mandible, creating a

curved form on the side of the face. The muscle holds the lower jaw tightly against the upper one when clenching or chewing, and bulging the masseter belly and creating several long, thin parallel bundles along the length of the muscle.

3.3 VARIATIONS IN MUSCULAR ANATOMY

> "people usually smile the way they can, not the way they want"
>
> Jacques Zufferey (1992)

Significant variation in the number, size, symmetry, length, orientation and shape of mimic muscles is frequently observed between and within individuals. These variations account for some of the apparent contradictions in the findings reported in various studies and texts. For example, only a few anatomy texts such as Goldfinger (1991) report a malaris muscle continuous with the peripheral fibres of the orbicularis oculi to insert into the cheek fat and the nasolabial fold. Also the risorius muscle is not present in many individuals and is the most variable of all mimic muscles. Other important variations exist and are discussed hereafter. For now, it suffices to state that the above description of the mimic muscle set must be considered an example of the pattern of muscles in the mimic muscle system in which variation is the norm. In general, the variations in the mimic muscle set account for many of the variations observed in the static and dynamic appearance of the face.

3.3.1 *Variation of facial muscles at the angle of the mouth*

According to standard anatomy texts, the levator anguli oris, zygomaticus major, risorius, buccinator and depressor anguli oris (the direct labial tractors) converge at a point lateral to the angle of the mouth and form a palpable, muscular or tendinous node called the *modiolus* (Latin for "the nave of a wheel"). However, surgical dissections performed by Shimada and Gasser (1989) do not support the universal existence of such a muscular node. In a majority of cases, the direct labial tractors were found not to converge to the angle of the mouth but rather above or below it. Based on 147 specimens, three patterns of convergence were identified: lateral to the mouth angle (type A, 43.2%), above the mouth angle (type B, 41.2%) and below the mouth angle[2] (type C, 16.5%). Furthermore, the length of the labial tractors can be inferred from the patterns of convergence. For example, the zygomaticus major is longest in the type C arrangement, such that its contraction will move the corners of the mouth the farthest in the upward and outward direction, producing a uniquely broad ("Mona Lisa") smile. Similarly, other convergence patterns produce different types of smiles. Based on a random sample of 100 subjects, Rubin (1974) identified the three following types of smiles, shown in Figure 3.8:

2 The archetypal arrangement, frequently described in textbooks.

(a) (b) (c)

Figure 3.8 Types of smiles, identified by Rubin (1974). (a) Mona Lisa smile. (b) Canine smile. (c) Full denture smile.

i. 'Mona Lisa' smile (observed in 67% of subjects): due to its length, the influence of the zygomaticus major is stronger and observed sooner as it pulls the corners of the mouth upward and outward, followed by the levators of the upper lips contracting to show upper teeth. According to Shimada and Gasser (1989), the types A and C, which together amount to 58.5% of the patterns of convergence they encountered, "approximate the 67% of the random sample that Rubin (1974) found to exhibit the "Mona Lisa" smile". The discrepancy between both percentages is probably due to the relatively small population sizes on which both findings were based. A repeat of both studies based on much larger sample sizes might produce closer percentages.

ii. 'Canine' smile (observed in 31% of subjects): thought to be produced by the type B arrangement in which the zygomaticus major is relatively short because the point of convergence is above the mouth angle. Also, the levators of the upper lip, especially the levator labii superioris do not terminate at the superior borders of the orbicularis oris, but instead traverse the muscle to attach to the mucocutaneous border in the vermillion. Therefore, the canine muscles predominate i.e. its effects are stronger and observed sooner, and their contraction exposes the canine teeth first, before the corners of the mouth contract secondarily to pull lips upward and outward. A gummy smile results when the contraction of this muscle is severe Benedetto (2005).

iii. 'Full denture' smile (observed in 2% of subjects): is the result of the simultaneous contraction of the upper lip elevators and lower lip depressors around the mouth, and is characterized by the simultaneous separation of the upper and lower lips exposing the maxillary and mandibular teeth. Shimada and Gasser (1989) do not identify the pattern of convergence that gives rise to the full denture smile. This is undoubtedly because of its rarity. Fortunately, this unknown arrangement of muscles, and possibly other types of smiles unidentified by Rubin (1974), can be obtained by simulating smiles using a physically-based facial animation system that supports the repositioning of virtual muscles. In fact, the same experiment can

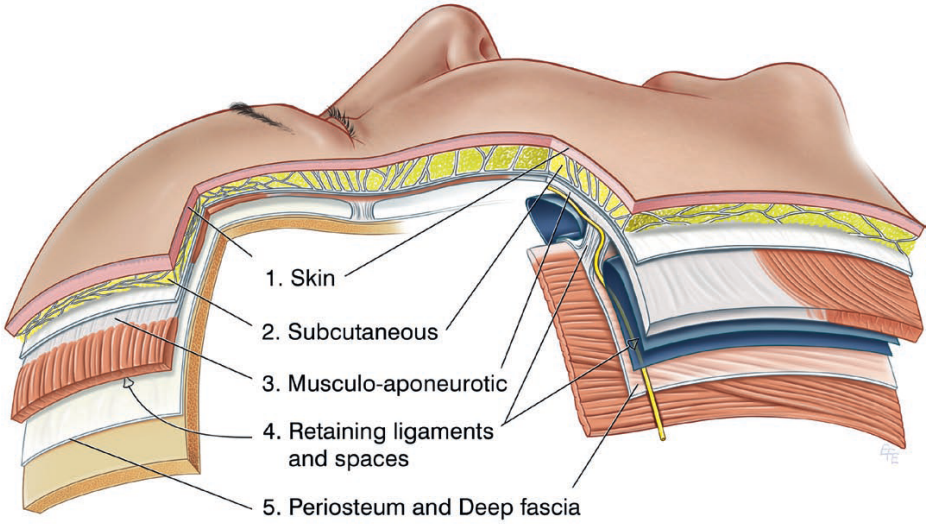


Figure 3.9 The five layers of the human face. (From Mendelson (2009), used with permission of the copyright holder.)

be performed for other facial expressions, with a view to discovering their possible typology.

Variations in the structure of individual mimic muscles are also possible. For example, bifurcated or bifid zygomaticus major muscles have been observed in 34% and 40% of subjects studied by Pessa *et al.* (1998b) and Hu *et al.* (2008) respectively. Bifid zygomaticus major muscles consist of two branches. One bundle of fibres, often inferiorly situated and the wider of the two, that inserts to the corner of the mouth and another bundle that inserts into the lateral cheek, and has dermal attachments. This dermal tethering is thought to produce cheek dimples when the muscle contracts, for example when smiling.

3.4 SOFT TISSUE ANATOMY

The soft tissue of the face is not a homogeneous mass but rather a layering of five functionally and anatomically distinct structures, shown in Figure 3.9. The layers of the soft tissue matrix are readily identified in the scalp and easily recalled by the mnemonic SCALP, where S: skin, C: connective issue, A: aponeurosis, L: loose areolar tissue and P: pericranium. These layers are modified in areas of function but their arrangement is consistent throughout the face Mendelson (2009).

3.4.1 *Layer 1 - skin*

Human skin consists of two layers of tissue: the epidermis and dermis. The epidermis (from the Greek, epi - 'on top of' and derma - 'skin') is the outer of the two and is made

up of cells formed by the division of the single layer of basal cells (at the base) of the epidermis. The new cells migrate toward the surface of the epidermis over a period of about 4-6 weeks, during which they lose their nuclei and start to produce a strong protein called keratin that toughens them. This process is known as keratinization. The cells gradually blend to form squamous or scalelike sheets of flattened cells (mostly dead) whose function is to prevent the entry of external substances to the body and the loss of moisture. The dead cells at the top of the epidermis are regularly shed but are continually replaced by keratinization.

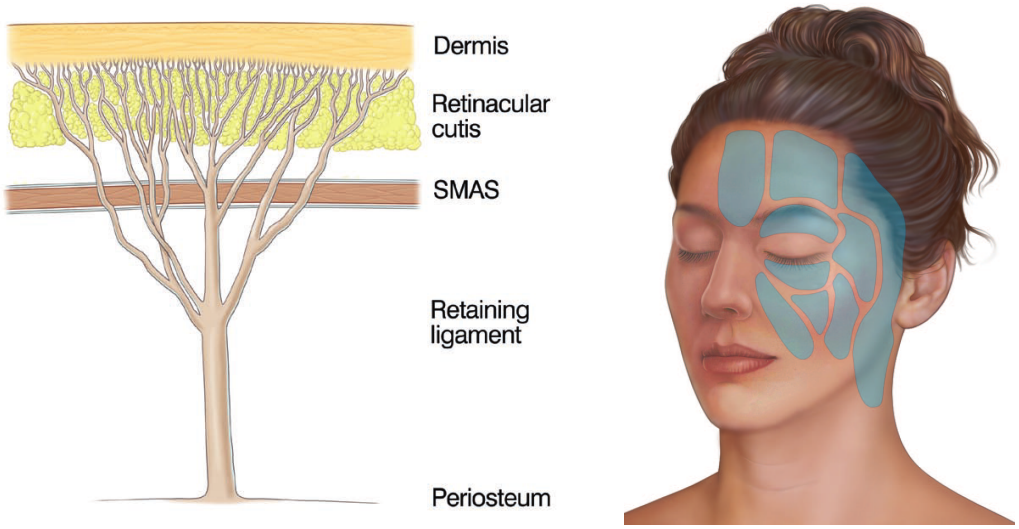
The dermis is the lower layer of the skin and is much thicker than the epidermis. The dermis consists of mostly two protein structures: collagen fibres (about 70% by weight) that provides resistance to stretch and elastin fibres that gives the skin its elastic properties, allowing it to return to its original state when tensed. Collagen fibres prevent elastin fibres from stretching to the point of breaking. Both fibres are bound together by a gelatinous or fluid-like matrix called ground substance that exudes from interfibre space and gives skin its viscoelastic behavior Maurel *et al.* (1998). However, with age, the production of elastin begins to slow and eventually stops (anywhere from puberty to the fourth decade of life). As a result, skin loses its elasticity and (in the absence of sufficient collagen fibres) the stresses of the mimic muscles leave fine-scale wrinkles on the skin.

The thickness of the skin (epidermis and dermis) varies throughout the face and has been found to be thinnest at the eyelid and greatest at the nasal tip Ha *et al.* (2005).

3.4.2 *Layer 2 - subcutaneous*

The subcutaneous layer or hypodermis (from the Greek, hypo - 'under') is made up of adipose (fatty) tissue connected to the dermis by fibrous strands or septae called retinacula cutis. The thickness of the subcutaneous fat and its attachment to the dermis generally varies throughout the face. However, in the scalp, the subcutaneous layer has consistent thickness and fixation to the overlying dermis. Furthermore, because of the tree-like distribution of retinacula cutis fibres, as shown in Figure 3.10(a), the attachment of the subcutaneous layer is stronger at the dermis than at its deep surface Mendelson (2009).

The subcutaneous fat is not a single mass but exists in discrete anatomical units or compartments created by fascial membranes that act as partitioning barriers Rohrich and Pessa (2008). By carefully dissecting cadavers injected with differently coloured dyes, Rohrich and Pessa (2007) identified nine subcutaneous fat compartments in the anterior face as shown in Figure 3.10(b). The subcutaneous layer provides insulation, cushions or protects internal organs and acts as a store of energy.



(a) (b)

Figure 3.10 (a) Tree-like structure of retaining ligaments (From Mendelson (2009) – used with permission of the copyright holder.) (b) Subcutaneous fat compartments of the human face. (From Rohrich and Pessa (2007) – used with permission of Wolters Kluwer Health.)

3.4.3 *Layer 3 - musculoaponeurotic*

The mimic muscle system is enveloped by and linked to a composite fibrofatty layer of tissue, consisting of collagen, elastic fibres and fat cells Har-Shai *et al.* (1997, 1998), called the superficial musculoaponeurotic system (SMAS)[3]. Unlike the more viscous facial skin, the SMAS has delayed stress relaxation, i.e. has a reduced tendency to accommodate itself to applied deformations, and therefore acts as a firmer foundation for the overlying tissues. The SMAS divides the facial fat into two layers. Superficial to it are small fat lobules enclosed by numerous fibrous septa (retinacula cutis) fixing it to the dermis. However, deep to the SMAS, fat is abundant and not divided by fibrous septa.

The SMAS lacks bony insertions, lies in continuity with the superficially situated mimic muscles (platysma, orbicularis oculi and frontalis), and is kept tense by them (Mitz and Peyronie, 1976; Gosain *et al.*, 1993). The corrugator supercilii, procerus, zygomaticus major and minor, levator labii superioris, levator anguli oris, depressor anguli oris and depressor labii inferioris are deep to the SMAS, as they have extensive origins to the facial skeleton Mendelson (2009). Nevertheless these muscles attach to the SMAS at their insertions. The tautness of the SMAS, superiorly by the frontalis and orbicularis muscles and inferiorly by the platysma, allows the SMAS to acts as an amplifier and distributor of facial muscular contractions to the skin. The more the SMAS is tensed, the more readily

3 Pronounced /smas/

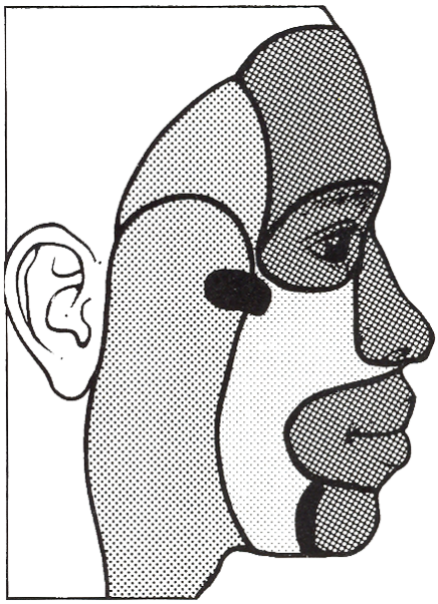


Figure 3.11 Relative strength of attachments of the SMAS to the dermis at various parts of the face. Darker stipples indicate stronger attachments while lighter stipples indicate weaker attachments. (From Keller (1997).)

it transmits the contraction of facial muscles (Mitz and Peyronie, 1976). However, with age the elastic fibres of the SMAS weaken and are less efficient at transmitting muscle contractions to the skin.

The SMAS is an extension of the superficial cervical fascia (i.e. superficial fascia of the neck) into the face Stuzin *et al.* (1992). The SMAS is histologically distinct from, but closely adherent to, the parotid fascia and is discontinuous at the zygoma Gosain *et al.* (1993); Larrabee *et al.* (2008). The superficial temporal fascia or temporal parietal fascia (not to be confused with the deep temporalis fascia which covers the temporalis muscle) is an extension of the SMAS Larrabee *et al.* (1997). The SMAS is very dense and thick overlying the parotid, temporal region and scalp but thinner and less substantive over the masseter, buccal fat pad, nasolabial fold and upper lip Stuzin *et al.* (1992); Pensler *et al.* (1985). In the parotid, zygomatic infraorbital regions and lateral to the nasolabial fold, the SMAS consist of fibrous septa enclosing lobules of fat. However, medial to the nasolabial fold and in the upper lip the SMAS does not contain separate lobules of fat but rather consists of a dense meshwork of intermingled collagen and muscle fibres that reach up to the dermis, conferring a firm connection of facial muscles to the lips Ghassemi *et al.* (2003). Figure 3.11 shows the relative strengths of attachments of the SMAS to the dermis.

3.4.4 *Layer 4 - sub-SMAS or deep plane*

In addition to the deep layer of intrinsic muscles, the interface of the superficial and deep fascias contain the following structures Mendelson (2009, 2008).

Facial spaces

Facial spaces occupy the largest area of the deep plane. Their function is to allow the movement of the overlying tissue due to the contraction of superficial muscles. Facial spaces are often described as consisting of loose connective areolar tissue (i.e. containing loosely organized fibres and significant empty space) or as true clefts in the soft tissue of the face – due to orientation of retinacula cutis fibres. Facial spaces exist in two forms: as spaces provided by bony cavities, for example the oral cavity (the largest of all facial spaces) beneath the lips and nasolabial fold which allows the movement of the midcheek and the muscles of the lips, and the submuscular (preseptal and conjunctival) spaces of the eyelids, within the orbit, which allow the movement of the lids. Soft tissue spaces also exist as a series of voids separated by facial ligaments. Specifically, the prezygomatic space overlies the body of the zygoma and allows the contraction of the orbital portion of the orbicularis oculi, the temporal and premasseter spaces overlie the fascias of the muscles of mastication and facilitate the opening of the jaw and the movement of the mandible relative to the overlying platysma (and vice versa) respectively, while the masticator or buccal space underlies the midcheek medial to the masseter and, like the oral cavity, facilitates movement of the overlying nasolabial segment of the midcheek.

Retaining ligaments

The facial spaces are punctuated by aponeurotic condensations of fibrous tissues, called retaining ligaments, that run from and fix deep facial structures to the overlying dermis. Retaining ligaments also bind adjacent structures and support the layers of facial soft tissue in their normal anatomic positions Stuzin *et al.* (1992), delineate the facial spaces Mendelson and Jacobson (2008) and localize the movement of superficial tissues Mendelson (2001). Retaining ligaments take their origins from the skull (osteocutaneous or dermal-periosteal ligaments) or from the soft tissues of the face (fasciocutaneous or intrafascial ligaments).

Osteocutaneous ligaments, the stronger of the two, support the soft tissues of the face at two locations: at or near the inferior border of the anterior zygomatic arch (zygomatic ligaments) and just above the mandibular border (mandibular ligaments) Furnas (1989). Osteocutaneous ligaments cross the sub-SMAS plane to the underside of the SMAS where they divide into numerous branches in a tree-like fashion and distribute to the dermis to form the retinacula cutis Moss *et al.* (2000).

Fasciocutaneous ligaments are found at three key locations on either side of the face, namely: the anterior cheek lateral to the nasolabial fold (the buccal-maxillary ligaments),

at the middle cheek parallel to the anterior border of the masseter muscle (masseteric-cutaneous ligaments) and the posterior cheek parallel to the angle of the mandible and the lobule of the ear (platysma-auricular ligaments) Raskin and Latrenta (2007).

3.4.5 *Layer 5 - deep fascia*

The deep fascia is the deepest soft tissue layer of the face. Over the bony skeleton, the deep fascia takes the form of the periosteum through which pass the attachments for the deep fascial muscles and facial ligaments. Over the lateral face, the equivalent of the deep fascia is the deep temporal and masseteric fascia, which also provides attachment for retaining ligaments. However, over the bony cavities there is no periosteum and the fifth layer is a lining derived from the cavity Mendelson (2009).

3.5 NASOLABIAL FOLD

The nasolabial fold (NLF) is a crease that runs from the side of the nose to the corner of the mouth on each side of the face. The fold consists of two muscle bundles: a superficial, often poorly-defined bundle of fibres (called fold muscles) running parallel or near to the fold, and a deeper-situated bundle of lip-levator muscle fibres that run across the fold on the way to upper lip vermilion Zufferey (1992). The buccal (cheek) fat lies superior to and is held in place by a combination of both sets of muscles and fibrous septae running through the fat pad Pogrel *et al.* (1998). Even while at "rest" mimic muscles exist in a state of pre-tension or relaxation-contraction called "resting dynamic equilibrium" (RDE) Zufferey (2003). The muscle fibre content of the SMAS enhances its tonus and helps keep the skin smooth and in light tension. A strong, tonic SMAS also "closes the door" against and prevents the deep fat pad from protruding and therefore accentuating the NLF Zufferey (1999).

The knowledge of fold muscles and concept of RDE also explains why:

i. the fold is absent in newborns and young children but, with the loss of baby (cheek) fat, appears at about 25 years of age and becomes a prominent feature by the age 35 Zufferey (1992). This is because the effect of RDE on the fold does not become obvious until baby (cheek) fat is lost. The fold is also retained in death due to rigor mortis.

ii. the fold disappears in the event of facial paralysis. This is because nerve damage completely removes the unconscious muscle tonus (RDE) and facial skin achieves complete relaxation.

iii. botulinum toxins, skillfully administered to the head levator labii superioris alaeque nasi for example, are effective in effacing the upper part of an accentuated fold.

The botulinum toxin blocks neuromuscular transmission and causes muscles and therefore the skin to remain relaxed for extended periods of time.

Zufferey (1992) divides the NLF into four functionally independent segments, two lateral and two medial segments. The medial segments are moved by direct labial tractors (the medial part of the levator labii superioris alaeque nasi, levator superioris oris and zygomaticus minor) while the lateral segments are moved by indirect labial tractors (zygomaticus major, levator anguli oris, buccinator, risorius and depressor anguli oris). The angle between the medial and lateral angles creates three types of folds (concave, convex and straight). Based on a study of 50 subjects, Pessa *et al.* (1998a) found a convex crease in 60% of subjects, and straight and concave crease in 30% and 10% of subjects respectively. Their measurements were also used to classify crease lengths as being short (38%), extended (42%) or continuous (20%) in relation to the corner of the mouth.

Ultimately, the depth and form of the fold depends on the skin excess that results from the conflict between the dynamic and and inert soft tissues of the cheek Zufferey (1992). This conflict between the static and inert tissues grows with progression of the smile and age.

3.6 ANATOMY OF THE AGING FACE

The youthful face is characterized by a uniform rounded fullness Mendelson (1995a) due to an ample distribution of superficial and deep fat that produces a three-dimensional surface characterized by a series of arcs and convexities Coleman and Grover (2006). Furthermore, the young face appears homogenized without demarcation of the cosmetic units. Aging however is associated with fat atrophy (loss) in the (adipose compartments of the) periorbital, forehead, buccal, temporal and perioral areas and hypertrophy (accumulation) submentally in the (adipose compartments of the) jowl, lateral nasolabial fold, labiomental crease and lateral malar areas Donofrio (2000). Both morphologic changes cause skin to sag. In the case of atrophy, facial skin that is emptied of fat hangs due to the relative excess of remaining skin, whereas hypertrophy leads to significant drooping as facial skin is encumbered with extra facial fat. The redistribution of facial fat leads to a vivid demarcation of the cosmetic units as the fat pockets become discernable as separate entities, such that the arcs and convexities of the youthful face are replaced with broken, wavy or convex shapes. Furthermore, deflation of the deeper fat compartments (e.g. of the medial cheek) creates an overall loss of projection of the cheek Rohrich *et al.* (2008); Stuzin (2007).

Developing laxity (or ptosis) of the multilink fibrous support system (connective tissues and retaining ligaments) also contributes to the displacement of soft tissues and loss of structure that characterizes the aging face. The laxity arises from the combination of "age-related atrophy and degeneration of the connective-tissue compounded by the 'wear

and tear' effect from repeated displacement of the soft tissues" as a result of the action of facial muscles Mendelson (2001). Specifically,

i. Attenuation of the masseteric cutaneous ligaments within the cheek leads to an inferior migration of the cheek soft tissues which accumulate at the border of the mandible, to form jowls Stuzin *et al.* (1992). This happens because the anterior portions of the intrafascial system of cheek suspension are relatively weak and exist only at the border of the oral cavity, in a c-shaped pattern Mendelson (2008) around the last places available for skeletal support Mendelson (2009). This combination of weak, sparse intra-cheek suspension and heavier anterior cheek mass ultimately results in inferolateral descent of facial soft tissues Raskin and Latrenta (2007).

   Furthermore, with the attrition of the weaker ligaments, the areas of looser retinacula cutis attachments overlying the facial spaces become distended and appear as bulges in the skin surface Mendelson and Jacobson (2008). However, the stronger ligaments resist distention and create cutaneous grooves on the skin surface, such that the cheek surface "becomes uneven, with bulges alternating between lines of grooves." For example, the distention of the premasseter space contributes to the accumulation of soft tissues behind the unattenuated fixation of the key masseteric ligaments in jowl formation Mendelson *et al.* (2008).

   Similarly, distention of the prezygomatic space, vestibule of the oral cavity and masticator space produce malar (cheek) mounds, accentuated nasolabial folds and labiomandibular folds (or marionette lines) respectively Mendelson (2009).

ii. Over time, the contraction of the medial platysma "eventually pulls the connective tissue away from the depth of the neck concavity" Mendelson (2001) creating a turkey or double chin. Similarly, severe attenuation and lengthening of the orbicularis oculi and laxity of the orbicularis retaining ligament results in the formation of festoons, i.e. sagging of the muscle in the lower lid Furnas (1993). In addition, the distention of the preseptal space of the lower lid and the lengthening of the orbicularis retaining ligament allows the descent of orbital fat into the upper part of the prezygomatic space creating lower lid (or palpebral) bags and palpebromalar grooves (i.e. visible junction between the preseptal part of the lower lid and the cheek) Mendelson *et al.* (2002).

Too often, gravity is wrongly named as the cause of many of the manifestations of facial aging. The truth however is that gravity merely allows us to witness the effects of fat distribution in the aged face. As such, "gravity though always present, is most likely an innocent-bystander but not causative in nature" Donofrio (2000).

Anatomic (or hereditary) and lifestyle variations ultimately determine the type and severity of facial aging. Therefore, the above age-related effects manifest to different degrees in individuals, and are generally accompanied by a decrease in tissue elasticity, loss of collagen and a reduction in facial height due to mandibular and maxillary resorption Coleman and Grover (2006).

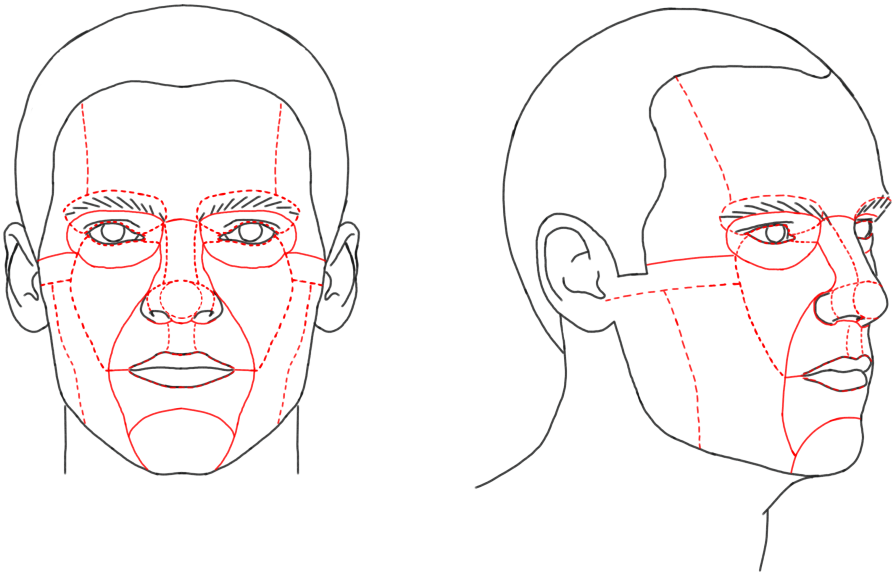


Figure 3.12 Facial aesthetic units and subunits (based on Figures 1 and 2 of Fattahi (2003)).

3.7 FACIAL AESTHETIC UNITS

Based on a study of the regional thickness of skin in the human body Gonzáles-Ulloa (1956) divided the adult human face into 14 aesthetic units (see Figure 3.12), within which skin has fairly uniform thickness, texture, colour and histology. He also showed that superior surgical outcomes can be obtained in facial reconstruction by replacing entire aesthetic units (with grafts[4] or flaps[5] having similar properties) instead of the obvious "affected skin" portions or defects alone Fattahi (2003). This practice allows surgical incisions, and therefore scars, to be hidden within the borders of aesthetic units.

Drawing on the principles of human perception as well as observations about how shapes reflect light and cast shadows, Menick (1987) subdivided the original facial aesthetic units into smaller subunits. All faces have the same number of aesthetic subunits. However, their relative "proportions are unique to each person's appearance" Thompson and Menick (1994).

3.8 SUMMARY

The human body is built stratigraphically, i.e. through the overlapping of several layers of different tissues Ferreira *et al.* (2006). Facial soft tissue for example consists of concentric layers of skin, subcutaneous fat, superficial fascia (SMAS), mimic muscles and deep

4 Donor skin that does not comprise subcutaneous tissue.
5 Donor skin that comprises subcutaneous tissue.

fascia Stuzin *et al.* (1992). The superficial fascia is intimately associated with the mimic muscles and both function as a single anatomic unit, driving the movement of facial skin Stuzin *et al.* (1992). Furthermore, each mimic muscle is accompanied by a related facial space Mendelson (2009) intended to facilitate its movement, while a system of retaining ligaments perform the opposite function by tethering the overlying tissues to deeper structures. The discovery of the structure, function and progression of these tissues explains complex aging changes and has revolutionized the field of facial plastic surgery. Although most of the existing physically-based facial animation systems model the epidermis, dermis, subcutaneous fat layers and the skull, they do not account for structures such as the superficial musculoaponeurotic system, retaining ligaments and facial spaces. Therefore, as earlier stated, their utility in modern, accurate, surgical planning and prediction is severely limited. In contrast, therefore, this work develops a series of techniques for automatically generating two of these vital structures, the mimic muscles and SMAS, in addition to the skull, for any given 3D head model.

And they did beat the gold into thin plates . . . with cunning work.
— Exodus 39:3

4 THEORY AND APPLICATIONS OF THIN-PLATE SPLINES

Given a sequence of spatial data samples, it is often desired to obtain the function from which the samples were taken. There are two approaches to obtaining this function. The functional analysis approach is to consider the (infinite) set or space of all possible functions interpolating the given points and impose restrictions on the space such that the search yields an acceptable function. The idea of a set of "all possible functions" is an extension of the concept of vectors. Accordingly, the review of this point of view will of necessity start with the idea of vector spaces.

The other approach, known as kriging (see Isaaks and Srivastava (1989), Olea (1999)), is stochastic and instead assumes that the data samples are generated by a random process. (Quite often, the measurements taken from natural phenomena.) Although in reality all physical processes are governed by physical laws, this approach makes the reasonable assumption that the process that generates the observations is so complicated that it can be thought of as a random function. The resulting analysis thus involves a study of random functions. Remarkably both kinds of analysis yield related formulae. The equivalence between kriging and thin-plate splines is formally discussed in Kent and Mardia (1994), Wahba (1990), and Matheron (1981).

This chapter presents a simple, intuitive and direct discourse of thin-plate splines, with recourse to only those technicalities that are necessary for the development of the primary arguments. However, for completeness, some technical remarks, lengthy algebra and proofs are given in the footnotes, the appendix and occasional parenthesis. The chapter concludes with an overview of several applications of this technique required in this thesis.

4.1 THIN PLATE SPLINES

4.1.1 *Vector spaces*

A vector space is a set (of vectors[1]) $\mathbf{V}$ for which the operations addition and multiplication by a scalar are defined. Furthermore, given any $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{V}$ and the scalars α, β, the following axioms hold:

1. the addition of any two vectors is another vector i.e. $\mathbf{x} + \mathbf{y} \in V$

1 Here, the term vector is used in a general sense and does not exclusively refer to a directed line segment representing the magnitude and direction of quantity such as force or velocity. However, it will be later shown that similar metrics hold for the generalization.

2. the multiplication of a vector by a scalar produces a vector i.e. $\alpha\mathbf{x} \in V$
3. $(\mathbf{x}+\mathbf{y})+\mathbf{z} = \mathbf{x}+(\mathbf{y}+\mathbf{z})$ (the associativity of addition)
4. $\mathbf{x}+\mathbf{y} = \mathbf{y}+\mathbf{z}$ (commutability of addition)
5. $\mathbf{x}+\mathbf{0} = \mathbf{x}$ (existence of a null or zero element $\mathbf{0}$)
6. $\mathbf{x}+(-\mathbf{x}) = \mathbf{0}$ (existence of the inverse element)
7. $\alpha(\beta\mathbf{x}) = (\alpha\beta)\mathbf{x}$ (associativity with respect to scalar multiplication)
8. $\alpha(\mathbf{x}+\mathbf{y}) = \alpha\mathbf{x}+\alpha\mathbf{y}$ (distributivity with respect to vector addition)
9. $(\alpha+\beta)\mathbf{x} = \alpha\mathbf{x}+\beta\mathbf{x}$ (distributivity with respect to scalar addition)
10. $1\mathbf{x} = \mathbf{x}$ (unitarism)

Interestingly, these axioms hold for (elements of the) real number line and of course Euclidian vectors. Accordingly both sets are vector spaces. In fact, real numbers may be considered one-dimensional Euclidian vectors. By extension, higher-dimensional Euclidian vector spaces also exist.

An important property of vector spaces is the existence of a set of basis vectors $(\mathbf{v_0}, \mathbf{v_1}, \ldots \mathbf{v_n})$, a linear combination of which is sufficient to represent any element $\mathbf{v}$ in the vector space[2], i.e.

$$\mathbf{v} = \alpha_0\mathbf{v_0} + \alpha_1\mathbf{v_1} + \ldots + \alpha_n\mathbf{v_n}$$

Such a set is said to span the vector space. By the same reasoning, using the fourier series representation,

$$f(x) = \sum_{n=1}^{\infty} \alpha_n \sin\frac{n\pi x}{L} + \sum_{m=0}^{\infty} \beta_m \cos\frac{m\pi x}{L}$$

any periodic function $f(x)$ can be expressed as a weighted sum of sine and cosine terms or functions, which can be considered as the basis of an infinite-dimensional vector space (to which $f(x)$ belongs). Further generalizing on the idea of basis functions, it is possible to speak of the vector space of quadratic polynomials in one variable x as having the basis functions x^2, x and 1, a linear combination of which is sufficient to generate a single-variable polynomial ax^2+bx+c. Each point in such a space is a function; and the space in turn is referred to as a function space. To reiterate, a function space is a (vector) space whose elements are functions.

4.1.2 *Normed vector spaces*

A vector space $\mathbf{V}$ is referred to as an inner product space if for any $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbf{V}$ and the real scalar α, there is defined a real scalar $\langle \mathbf{x}, \mathbf{y} \rangle$ for which the following axioms hold:

2 Basis vectors are linearly independent, i.e. none of the basis vectors can be written as a linear combination of the others.

1. $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$ (symmetry)
2. $\langle \mathbf{x}, \mathbf{y} + \mathbf{z} \rangle = \langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle$
3. $\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$
4. $\langle \alpha \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{y}, \mathbf{x} \rangle$
5. $\langle \mathbf{x}, \mathbf{x} \rangle \geqslant 0$ (positivity)
6. $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ if and only if $\mathbf{x} = 0$ (definiteness)

For example, the inner product of the vector space of n-tuples of real numbers having typical elements $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ is the Euclidian, n-dimensional dot or scalar product

$$\langle \mathbf{x}, \mathbf{y} \rangle = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n = \sum_{j=1}^{n} x_j y_j$$

However, the inner-product of a vector space of real-valued functions defined on the interval $[a, b]$ is given as

$$\langle f, g \rangle = \int_a^b f(t)\, g(t) dt \tag{4.1}$$

Dettman (1974) (Example 3.6.2) shows that this equation satisfies the above axioms.

The norm of a vector $\mathbf{x}$ is commonly defined, in relation to the inner product, as

$$\| \mathbf{x} \| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$$

The norm is a measure of the length of a vector. For example, on the real number line, the magnitude $| x |$ measures the length of a number x, while $| x - y |$ measures the distance between x and y. However, when dealing with objects with more dimensions or degrees of freedom, there can be more than one valid notion of length and distance. For example, candidates for the magnitude of a three-dimensional box include: length, width, height, volume, surface area. "Because of this, one abandons the the idea that there should be one notion of 'magnitude' for boxes" Tao (2008). As such a variety of norms can be defined for a vector space of n-dimensional tuples. For example the p-norm:

$$\| \mathbf{x} \| = \left(\sum_{j=1}^{n} | x_j |^p \right)^{\frac{1}{p}}$$

which is a generalization of the familiar 2-norm

$$\| \mathbf{x} \| = \sqrt{\sum_{j=1}^{n} | x_j |^2}$$

Other norms include:

$$\| \mathbf{x} \| = \sum_{j=1}^{n} | x_j | \quad \text{(Manhattan distance)}$$
$$\| \mathbf{x} \| = \max_{j=1}^{n} | x_j |$$

Similarly, there are infinitely many distinct notions of the magnitude of a function or how close two different functions are. Tao (2008) notes that:

> "...this situation may seem chaotic, but it simply reflects the fact that functions have many distinct characteristics - some are tall, some are broad, some are smooth, some are oscillatory, and so forth - and depending on the application at hand, one may need to give more weight to one of these characteristics than to others."

As such, the generalization of p-norm to function spaces is given as

$$\| f \| = \left(\int_a^b | f(t) |^p \, dt \right)^{\frac{1}{p}} \quad \text{where } 1 \leqslant p \leqslant \infty$$

Such a function space over which the above norm is finite is called an L^p space or Lebesgue space. The 2-norm of the fourier expansion is given as

$$\begin{aligned}
\| f \|^2 &= \int_{-L}^{L} \left(\sum_{n=1}^{\infty} \alpha_n \sin\frac{n\pi x}{L} + \sum_{m=0}^{\infty} \beta_m \cos\frac{m\pi x}{L} \right)^2 dx \\
&= \sum_{s=1}^{\infty}\sum_{n=1}^{\infty} \alpha_s \alpha_n \int_{-L}^{L} \sin\frac{n\pi x}{L}\sin\frac{s\pi x}{L}dx + \sum_{s=1}^{\infty}\sum_{m=0}^{\infty} \alpha_s \beta_m \int_{-L}^{L} \sin\frac{s\pi x}{L}\cos\frac{m\pi x}{L}dx + \\
&\quad \sum_{s=0}^{\infty}\sum_{n=1}^{\infty} \alpha_n \beta_s \int_{-L}^{L} \sin\frac{n\pi x}{L}\cos\frac{s\pi x}{L}dx + \sum_{s=0}^{\infty}\sum_{m=0}^{\infty} \beta_m \beta_s \int_{-L}^{L} \cos\frac{m\pi x}{L}\cos\frac{s\pi x}{L}dx \\
&= L\left(\sum_{n=1}^{\infty} \alpha_n^2 + \sum_{m=0}^{\infty} \beta_m^2 \right)
\end{aligned}$$

because

$$\begin{aligned}
\int_{-L}^{L} \sin\frac{n\pi x}{L}\sin\frac{m\pi x}{L}dx &= \begin{cases} L & \text{if} \quad m = n \\ 0 & \text{if} \quad m \neq n \end{cases} \\
\int_{-L}^{L} \cos\frac{n\pi x}{L}\cos\frac{m\pi x}{L}dx &= \begin{cases} L & \text{if} \quad m = n \\ 0 & \text{if} \quad m \neq n \end{cases} \\
\int_{-L}^{L} \sin\frac{n\pi x}{L}\cos\frac{m\pi x}{L}dx &= 0
\end{aligned}$$

In other words, the inner-product of any two basis functions is zero if the basis functions are unique, else the inner-product is a positive number corresponding to the norm of the basis function. Such a set of basis functions is said to form an orthogonal set. Formally, the set of basis functions $\{\phi_n(x)\}$ is said to be orthonormal if

$$\int_a^b \phi_m(x)\phi_n(x)dx = \delta_{n,m} = \begin{cases} 1 & \text{if} \quad m = n \\ 0 & \text{if} \quad m \neq n \end{cases}$$

Clearly, L^p norms reflect the height and width of a function but not its smoothness Tao (2008). In order to capture this information, the Sobolev norm of order p is defined as

$$\| f \|^p = \sum_{0 \leqslant (\alpha = \alpha_1 + \cdots + \alpha_n) \leqslant p} \int_{\mathbb{R}^n} \left| \frac{\partial^{|\alpha|} f}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} \right|^p dx$$

Where the sum is taken over all possible derivatives in $\mathbb{R}^n$ of order up to order p. For example, in one dimension

$$\| f \|^p = \int_{\mathbb{R}} \left(| f |^p + \left| \frac{\partial f}{\partial x} \right|^p + \left| \frac{\partial^2 f}{\partial x^2} \right|^p + \cdots + \left| \frac{\partial^{|p|} f}{\partial x^{|p|}} \right|^p \right) dx$$

A variant of the Sobolev norm is the Beppo-Levi semi-norm (of order q) – so called because it does not penalize (i.e. assigns the value zero to) polynomials of order less than $(q-1)$ – is defined as

$$\| f \|^q = \int_{\mathbb{R}^n} \sum_{\alpha_i + \cdots + \alpha_n = |q|} \binom{|q|}{\alpha_i! \cdots \alpha_n!} \left| \frac{\partial^{|q|} f(x)}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} \right|^2 dx \tag{4.2}$$

Such polynomials are said to be in the null space of the semi-norm. Note that whereas a norm assigns a positive, non-zero length to all non-zero vectors, a semi-norm is allowed to assign zero length to some non-zero vectors.

The inner product corresponding to the Beppo-Levi semi-norm is

$$\langle f , g \rangle = \int_{\mathbb{R}^n} \sum_{\alpha_i + \cdots + \alpha_n = |q|} \binom{|q|}{\alpha_i! \cdots \alpha_n!} \frac{\partial^{|q|} f(x)}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} \frac{\partial^{|q|} \overline{g}(x)}{\partial x_1^{\alpha_1} \cdots \partial x_n^{\alpha_n}} dx \tag{4.3}$$

It is important to note that whereas the Beppo-Levi semi-norm 4.2 admits an inner product, Sobolev norms do not generally have corresponding inner products. (As such, Sobolev norms form Banach spaces – see following section.)

The distance between two functions f and g is given by

$$d(f, g) = \| f - g \|$$

A vector space for which a notion of distance is defined is referred to as a metric space.

4.1.3 *Hilbert Spaces*

In a finite dimensional space, the weighted sum of basis functions, i.e. $\sum_k \alpha_k \, \mathbf{v}_k$, always has a finite value or vector. However, in an infinite-dimensional space it is possible for the series to converge to something other than a vector. For example, consider Newton's method for computing $\sqrt{2}$ by a series of approximations

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} = x_n - \left(\frac{x_n^2 - 2}{2x_n} \right) = \frac{x_n^2 + 2}{2x_n} = \frac{x_n}{2} + \frac{1}{x_n}$$

For example if $f(x) = x^2 - 2$ and $x_0 = 1$ is a first guess. The sequence of rational numbers[3] $(1, \frac{3}{2}, \frac{17}{12}, \ldots)$ is generated, and although the limit $\sqrt{2}$ sequence exists, it is not a rational number. A sequence whose elements become closer to each other (as measured by a distance metric) as the sequence progresses is referred to as a Cauchy sequence. However, the limit of such a sequence can lie outside the space. Such a space is referred to as incomplete as there are elements missing from it (as suggested by the limit of a Cauchy sequence).

An incomplete vector space can be made complete by enlarging the space to include the limit of its Cauchy sequences. For example, the space of all rational numbers can enlarged to a complete metric space of real numbers by the addition of irrational numbers. A complete inner-product space is referred to as a Hilbert space[4] $\mathcal{H}$.

4.1.4 *Reproducing Kernel Hilbert Spaces (RKHS)*

Not all functions in $\mathcal{H}$ are bounded, i.e. have finite values or are pointwise defined. For example the space $\mathcal{L}_2[a, b]$ of square integrable functions defined on the interval $[a, b]$ where the norm

$$\| f \|^2 = \int_a^b f(x)^2 dx$$

is induced by the inner product given in Equation 4.1. One problem with this space is that some functions such as

$$g(x) = \begin{cases} c & \text{if } x = k \\ f(x) & \text{otherwise} \end{cases}$$

have the undesirable property that their values are undefined or infinity at a finite number of zero measure sets, while remaining square integrable. (As a result, two functions $g_1(x)$ and $g_2(x)$ with different values at the point k have the same norm.) Clearly, square integrability of a function is not sufficient to guarantee that the function will be well-behaved or pointwise defined, i.e. can be evaluated at every point in the domain. Therefore additional constraints are required on $\mathcal{H}$ in order to obtain a space of bounded, continuous (one implies the other) functionals, i.e. higher-order functions that evaluate ordinary functions at any given points. However, according to Schechter (2001) (page 29) the only bounded linear functionals in $\mathcal{H}$ are inner products.

Given a Hilbert space $\mathcal{H}$ having the inner-product $\langle x\ ,\ y \rangle$, where $x, y \in \mathcal{H}$, if y is fixed $\langle x\ ,\ y \rangle$ assigns to each x a number. Such an expression is referred to as a functional $\mathcal{F}(x)$. (The norm is another example of a functional.) For any given scalars α_1 and α_2, the functional $\mathcal{F}(x)$ satisfies the following property:

$$\mathcal{F}(\alpha_1 x_1 + \alpha_2 x_2) = \alpha_1 \mathcal{F}(x_1) + \alpha_2 \mathcal{F}(x_2) \qquad \text{(linearity)}$$

3 Can be written as a quotient of two numbers a and b i.e. $\frac{a}{b}$

4 By contrast, a Banach space is a complete normed space. However in a Hilbert space, the norm is defined by its inner product, while the norm of a Banach space is not defined via an inner product. As such, Hilbert spaces may be considered special cases of Banach spaces.

By definition, the kernel of $\mathcal{F}(x)$ is the set of elements y for which $\mathcal{F}(x) = 0$. Clearly, $y = 0$ belongs to this set. However, a non-zero member of this set exists if it is orthogonal to x, i.e. $x \perp y$, so that,

$$\mathcal{F}(x) \;=\; \langle x\,,\, y\rangle \;=\; 0$$

Furthermore, on account of the linearity of $\mathcal{F}$,

$$\mathcal{F}(\;\mathcal{F}(y)\;x - \mathcal{F}(x)\;y\;) = \mathcal{F}(y)\mathcal{F}(x) - \mathcal{F}(x)\mathcal{F}(y) = 0$$

Therefore,

$$\begin{aligned}
\langle\;\mathcal{F}(y)\;x\;-\;\mathcal{F}(x)\;y\;,\;y\rangle \;&=\; 0\\
\mathcal{F}(y)\langle x\,,\,y\rangle\;-\;\mathcal{F}(x)\langle y\,,\,y\rangle \;&=\; 0\\
\mathcal{F}(x)\;=\;\frac{\mathcal{F}(y)}{\langle y\,,\,y\rangle}\langle x\,,\,y\rangle \;&=\; \langle\;x\,,\;\frac{\mathcal{F}(y)}{\|\;y\;\|^2}\;y\rangle
\end{aligned}$$

or

$$\boxed{\mathcal{F}_z(x) \;=\; \langle\;x\,,\,z\;\rangle \;=\; f(x)} \tag{4.4}$$

where

$$z \;=\; \frac{\mathcal{F}(y)}{\|\;y\;\|^2}\;y$$

Equation 4.4 is known as the Riesz representation theorem.

The kernel z is an analog of the dirac delta because it possesses a reproducing property[5]; similarly, z associates each element $x \in \mathcal{H}$ to a number $f(x)$. Therefore z is referred to as a reproducing kernel.[6] Furthermore, there is only one $z \in \mathcal{H}$ with this property. If there were another z_1,

$$\begin{aligned}
\mathcal{F}(x) \;=\; \langle\;x\,,\,z\;\rangle \;&=\; \langle\;x\,,\,z_1\;\rangle\\
\langle\;x\,,\,z\;\rangle\;-\;\langle\;x\,,\,z_1\;\rangle \;&=\; 0\\
\langle\;x\,,\,z - z_1\;\rangle \;&=\; 0
\end{aligned}$$

As this is true for all $x \in \mathcal{H}$, $z - z_1$ would have to be zero, implying that $z = z_1$.

In addition, it is easy to compute $\|\;f\;\|$. From the Cauchy-Schwartz inequality

$$|\;f(x)\;| \leqslant \|\;x\;\|\;\|\;z\;\|$$

By definition, $\|\;f\;\|$ is the smallest possible number that is greater than or equal to $|\;f(x)\;|$. In other words, $\|\;f\;\|$ is the greatest lower bound of $|\;f(x)\;|$.[7] Therefore,

$$\|\;f\;\| = \;sup\;|\;f(x)\;| \leqslant \|\;x\;\|\;\|\;z\;\|$$

5 The relationship $f(t) = \int_a^b f(x)\delta(x-t)dx$ is known as the reproducing property of the dirac delta.

6 The dirac delta function is not a reproducing kernel because $\delta(x)$ is not in $\mathcal{L}_2$. This is because the dirac delta is only defined by its effect on an ordinary function. As such, the product of two or more dirac deltas is meaningless. Therefore the $\mathcal{L}_2$ norm is undefined for it.

7 If $|\;f(x)\;| < \infty$, $\|\;f\;\|$ is the smallest number greater than or equal to $|\;f(x)\;|$.

Clearly, if no restriction is placed on $\| x \|$, $\| f \|$ will always be infinite. Therefore, the norm is evaluated at $\| x \| = 1$, so that

$$\| f \| = \sup_{\|x\| = 1} | f(x) | \leqslant \| z \|$$

If the norm were evaluated at any given value of $\| x \| = n$, for example,

$$\| f \| = \sup_{\|x\| = n} | f(x) | \leqslant n \| z \|$$

the result is simple linear scaling of the norm obtained at $\| x \| = 1$, showing that there is no advantage to evaluating $| f(x) |$ at values other than $\| x \| = 1$. Therefore, the value of $\| f \|$ is the maximum possible value of $| f(x) |$ which, by the above inequality, is $\| z \|$. Formally,

$$\| f \| = \| z \|$$

4.1.5 *Radial Basis Functions*

In two dimensions, the Beppo-Levi semi-norm (equation 4.2) for $q = 2$ is given as

$$\| f \|^2 = \int_{\mathbb{R}^2} \left(\left| \frac{\partial^2 f}{\partial x_1^2} \right|^2 + 2 \left| \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|^2 + \left| \frac{\partial^2 f}{\partial x_2^2} \right|^2 \right) dx_1 \, dx_2 \tag{4.5}$$

This equation coincides with the formula for computing the bending energy of a thin physical plate extended to infinity Rohr (2001) (page 198) – thus the name thin-plate spline[8]. Furthermore, the above semi-norm does not penalize linear polynomials or affine transformations and is therefore rotation, scale and translation invariant.

A termwise application of the Plancherel theorem (equation A.12) and equations A.11 and A.10 to equation 4.5 yields

$$\begin{aligned} \| f \|^2 &= \int_{\mathbb{R}^2} | (2\pi i \, \omega_1)^2 \, \tilde{f}(\omega) |^2 + 2 \, | (2\pi i)^2 \, \omega_1 \, \omega_2 \, \tilde{f}(\omega) |^2 + | (2\pi i \omega_2)^2 \, \tilde{f}(\omega) |^2 \, d\omega \\ &= (2\pi)^4 \int_{\mathbb{R}^2} (\omega_1^4 + 2 \, \omega_1^2 \, \omega_2^2 + \omega_2^4) \, | \tilde{f}(\omega) |^2 \, d\omega \end{aligned}$$

Because $\omega_1^4 + 2 \, \omega_1^2 \, \omega_2^2 + \omega_2^4 = (\omega_1^2 + \omega_2^2)^2 = \| \omega \|^4$, equation 4.5 is equivalent to

$$\| f \|^2 = \langle f , f \rangle = (2\pi)^4 \int_{\mathbb{R}^2} \| \omega \|^4 \, | \tilde{f}(\omega) |^2 \, d\omega = \int_{\mathbb{R}^2} \frac{| \tilde{f}(\omega) |^2}{\tilde{K}(\omega)} d\omega \tag{4.6}$$

where

$$\tilde{K}(\omega) = \frac{1}{(2\pi)^4 \, \| \omega \|^4}$$

Equation 4.6 has the same form (for any m) in n dimensions. (Section A.1.5 repeats the above derivation in three dimensions.) In fact, in general

$$\tilde{K}(\omega) = \frac{1}{(2\pi)^{2|q|} \, \| \omega \|^{2|q|}} \tag{4.7}$$

8 Thin-plate splines can likewise also be defined in three dimensions (and higher), even though such splines do not have physical analogues.

Furthermore, equation 4.6 suggests that the inner product corresponding to such semi-norms is

$$\langle\, f(x)\,,\; g(x)\,\rangle \;=\; \int_{\mathbb{R}^2} \frac{\tilde{f}(\omega)\;\tilde{g}(\omega)}{\tilde{K}(\omega)} d\omega \tag{4.8}$$

And according to the shift theorem A.7, for any $K(x)$

$$K(x-y) \;=\; e^{-2\pi i \omega . x}\;\tilde{K}(\omega)$$

so that from Equation 4.8

$$\langle\, K(x-y)\,,\; f(y)\,\rangle \;=\; \int_{\mathbb{R}^2} \frac{\tilde{K}(\omega)\; e^{-2\pi i \omega y}\; \tilde{f}(\omega)}{\tilde{K}(\omega)} d\omega \;=\; f(x)$$

Showing that the inner product Equation 4.8 satisfies the reproducing property Equation 4.4. Therefore Equation 4.8 defines a reproducing kernel Hilbert space.

The d-dimensional inverse Fourier transform of Equation 4.7 is

$$k(r) = \begin{cases} c_1\, r^{2q-d}\, \log(r) & \text{if } 2q \geqslant d \text{ and } d \text{ is even} \\ \\ c_2\, r^{2q-d} & d \text{ is odd} \end{cases} \tag{4.9}$$

where c_1 and c_2 are constants and $r \;= \| x-y \|$.

The radial basis function Equation 4.9 is valid for only integer values of m. A far more general radial basis function, valid for fractional orders of m can be obtained as follows. From Equation A.14, the norm of a two-dimensional thin-plate spline (Equation 4.5) can be expressed as

$$\begin{aligned} \| f \|^2 \;&=\; \int_{\mathbb{R}^2} \left(\left| \frac{\partial^2 f}{\partial x_1^2} \right|^2 + 2 \left| \frac{\partial^2 f}{\partial x_1^2} \right| \left| \frac{\partial^2 f}{\partial x_2^2} \right| + \left| \frac{\partial^2 f}{\partial x_2^2} \right|^2 \right) dx_1\, dx_2 \\ &=\; \int_{\mathbb{R}^2} \left(\left| \frac{\partial^2 f}{\partial x_1^2} \right| + \left| \frac{\partial^2 f}{\partial x_2^2} \right| \right)^2 dx_1\, dx_2 \end{aligned}$$

Similarly, from Equation A.14, the norm of a three-dimensional thin-plate spline (Equation A.14) can be expressed as

$$\| f \|^2 = \int_{\mathbb{R}^3} \left(\left| \frac{\partial^2 f}{\partial x_1^2} \right| + \left| \frac{\partial^2 f}{\partial x_2^2} \right| + \left| \frac{\partial^2 f}{\partial x_3^2} \right| \right)^2 dx_1\, dx_2\, dx_3$$

And in general, the norm of a d-dimensional thin-plate spline can be expressed as

$$\| f \|^2 = \int_{\mathbb{R}^d} \left(\sum_{i=1}^{d} \left| \frac{\partial^\alpha f}{\partial x_i^\alpha} \right| \right)^2 dx_1 \cdots dx_d \tag{4.10}$$

Furthermore α need not be 2, and can be a rational number, so that $\frac{\partial^\alpha f}{\partial x_i^\alpha}$ is a fractional derivative, in which case Equation 4.10 is referred to as a fractional Laplacian. However, Plancherel's theorem (Equation A.12) still applies because the sum of the fractional derivatives is raised to the power of 2. In fact the Fourier transform of the fractional

differential derivative has the familiar form $(-i\omega_i)^\alpha \tilde{f}(\omega)$ (see Podlubny (1999), sec. 2.9.3), so that by Plancherel's theorem, Equation 4.10 becomes

$$\begin{aligned} \| f \|^2 &= \int_{\mathbb{R}^d} \sum_{i=1}^{d} \left| (-i\omega_i)^\alpha \tilde{f}(\omega) \right|^2 d\omega \\ &= \int_{\mathbb{R}^d} \left(\sum_{i=1}^{d} \omega_i^{2\alpha} \right) |\tilde{f}(\omega)|^2 \, d\omega = \int_{\mathbb{R}^d} \|\omega\|^{2\alpha} |\tilde{f}(\omega)|^2 \, d\omega \end{aligned} \tag{4.11}$$

which is identical to Equation 4.6, so that for the reproducing kernel of a space with the norm Equation 4.10 is given as

$$\tilde{K}(\omega) = \frac{1}{\| \omega \|^{2\alpha}} \tag{4.12}$$

The d-dimensional inverse Fourier transform of this expression is

$$k(r) = \begin{cases} c_3 \, r^{2\alpha} \log(r) & \text{if } \alpha \text{ is an integer} \\ \\ c_4 \, r^{2\alpha} & \alpha \text{ is not an integer} \end{cases} \tag{4.13}$$

for all $\alpha > 0$, where c_3 and c_4 are constants. This radial basis function is identical to the kriging covariance functions (see Kent and Mardia (1994)) and encompasses the form given in Equation 4.9.

Finally, it is worth emphasizing that Equation 4.7 implies that $\|\omega\|^{2m}$ penalizes derivatives of order m. As such, the Gaussian norm

$$e^{-\tau^2\|\omega\|^2} = \sum_{m=0}^{\infty} -1^m \frac{(\tau\|\omega\|)^{2m}}{m!} = 1 - (\tau\|\omega\|)^2 + \frac{(\tau\|\omega\|)^4}{2!} - \frac{(\tau\|\omega\|)^6}{3!} + \cdots$$

penalizes all derivatives, where τ is some scaling factor. As the Gaussian is its own inverse Fourier transform, the RBF corresponding to the above Gaussian norm is

$$k(r) = e^{-r^2/\tau^2} \tag{4.14}$$

4.1.6 *Regularization in Reproducing Kernel Hilbert spaces*

A careful consideration of the motivating problem, described in the introduction to this chapter, shows that it is ill-posed in the sense that it does not admit a unique solution. For example, in the one dimensional case an infinite number of functions interpolate the samples. In order to narrow down on the space of possible interpolants an additional restriction is introduced based on prior knowledge of what a good solution is. The introduction of a prior is known as regularization. The norm of a function can be used as a prior if for example it is assumed that a smooth or small function is a better interpolant.

Therefore, the functional $\mathcal{F}[f]$ assigns a scalar value to any candidate function f based on the norm of the function $\| f \|$ and the degree to which the function interpolates the values $y_1 \ldots y_n$ at a set of data n samples $x_i \ldots x_n$. Such a functional can be written as:

$$\mathcal{F}[f] = \sum_{i=1}^{m} (f(x_i) - y_i)^2 + \| f \|^2 \tag{4.15}$$

The function f^* that best interpolates the data minimizes the functional $\mathcal{F}[f]$, and is the value of f for which the functional or Fréchet derivative $\frac{\delta\mathcal{F}}{\delta f}$ vanishes[9]. By definition

$$\frac{\delta\mathcal{F}}{\delta f} = \lim_{\epsilon\to 0} \frac{\mathcal{F}[\, f^* + \epsilon\tilde{f}\,] - \mathcal{F}[\, f^*\,]}{\epsilon} \tag{4.16}$$

where $\tilde{f}$ is the direction in which the derivative is taken, so that $\frac{\delta\mathcal{F}}{\delta f}$ is the rate of change of $\mathcal{F}$ in the direction of $\tilde{f}$ as $\epsilon \to 0$. Accordingly,

$$\mathcal{F}[f*] = \sum_{i=1}^{m} (\, f^*(x_i) - y_i\,)^2 \;+\; \langle f^*, f^*\rangle$$

and

$$\mathcal{F}[f^* + \epsilon\tilde{f}] = \sum_{i=1}^{n} (\, f^*(x_i) + \epsilon\tilde{f}(x_i) - y_i\,)^2 \;+\; \langle f^* + \epsilon\tilde{f}\,,\; f^* + \epsilon\tilde{f}\rangle$$

Furthermore, because

$$\begin{aligned} \|\, f\,\|^2 &= \langle f, f\rangle \\ \|\, f^* + \epsilon\tilde{f}\,\|^2 &= \langle f^* + \epsilon\tilde{f}\,,\; f^* \;+\; \epsilon\tilde{f}\rangle \\ &= \langle f^*, f^*\rangle \;+\; 2\epsilon\langle f^*, \tilde{f}\rangle \;+\; \epsilon^2\langle\tilde{f}, \tilde{f}\rangle \end{aligned}$$

It follows that

$$\begin{aligned} \frac{\delta\mathcal{F}}{\delta f} &= \lim_{\epsilon\to 0} \frac{2\epsilon\langle f^*, \tilde{f}\rangle \;+\; \epsilon^2\langle\tilde{f}, \tilde{f}\rangle \;+\; \sum_{i=1}^{m} 2\epsilon(f^*(x_i) - y_i)\tilde{f}(x_i) + \epsilon^2 f(x_i)^2}{\epsilon} \\ &= \lim_{\epsilon\to 0} 2\langle f^*, \tilde{f}\rangle \;+\; \epsilon\langle\tilde{f}, \tilde{f}\rangle \;+\; \sum_{i=1}^{m} 2(f^*(x_i) - y_i)\tilde{f}(x_i) + \epsilon f(x_i)^2 \\ &= 2\langle f^*, \tilde{f}\rangle \;+\; 2\sum_{i=1}^{n} (f^*(x_i) - y_i)\tilde{f}(x_i) \end{aligned} \tag{4.17}$$

By Equation 4.4 $\tilde{f} = K_x$ is a reproducing kernel that evaluates f^* at x. That is to say,

$$f^*(x) \;=\; \langle f^*, K_x\rangle \tag{4.18}$$

so that equating $\frac{\delta\mathcal{F}}{\delta f}$ to zero and rearranging yields

$$\boxed{f^*(x) = \sum_{i=1}^{m} \alpha_i K_x(x_i)} \tag{4.19}$$

where

$$\alpha_i = y_i - f^*(x_i)$$

Equation 4.19 is known as the representer theorem and $K_x(x_i)$ or $K(x, x_i)$ the representer of evaluation. The representer associates any function, including itself, with the value of the function at x_j, i.e.

$$\langle\, K(x, x_i)\,,\; K(x, x_j)\,\rangle \;=\; K(x_i, x_j)$$

9 The Fréchet derivative represents differentiation in the direction of a function (as opposed to differentiation in the direction of a vector). Differentials with respect to a function naturally arise in function spaces.

so that in general, for any

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \quad \text{and} \quad g(\mathbf{x}) = \sum_{j=1}^{m} \beta_j K(\mathbf{x}, \mathbf{x}_j)$$

an inner product

$$\begin{aligned} \langle f, g \rangle &= \left\langle \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i), \sum_{j=1}^{m} \beta_j K(\mathbf{x}, \mathbf{x}_j) \right\rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \beta_j \langle K(\mathbf{x}, \mathbf{x}_i), K(\mathbf{x}, \mathbf{x}_j) \rangle \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned} \tag{4.20}$$

can be defined, so that

$$\langle f, f \rangle = \| f \|^2 = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \geqslant 0 \quad \text{for all} \quad i, j \tag{4.21}$$

In matrix form,

$$\| f \|^2 = \boldsymbol{\alpha}^T \mathbf{K} \boldsymbol{\alpha} \tag{4.22}$$

where $\boldsymbol{\alpha}$ is the vector of α_i's and $\mathbf{K}$ the matrix of $K(\mathbf{x}_i, \mathbf{x}_j)$'s. Kernels for which 4.21 is true for every $\mathbf{x}_i$ and $\alpha_i \in \mathbb{R}$ are described as positive semi-definite[10].

As earlier discussed in section 4.1.6, Equation 4.22 is (proportional to) the bending energy of a thin-plate spline in any dimension. However, in general, if $\| f \|$ is a semi-norm (of order q), its null space must be accounted for by adding all polynomial terms of order $q-1$ and less, so that Equation 4.19 becomes

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}) \tag{4.23}$$

As a result, Equation 4.23 now has $n = \binom{d+q-1}{d}$ more terms than equations. Accordingly, n additional equations must be introduced as follows. Note that the polynomial terms are also kernels because (from section 4.1.4) a kernel is any non-zero element for which the linear evaluation functional, shown to be an inner product, is zero. Therefore

$$\begin{aligned} 0 &= \left\langle \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i), \mathbf{p}_j(\mathbf{x}) \right\rangle \\ &= \sum_{i=1}^{m} \alpha_i \langle K(\mathbf{x}, \mathbf{x}_i), \mathbf{p}_j(\mathbf{x}) \rangle \\ &= \sum_{i=1}^{m} \alpha_i \mathbf{p}_j(\mathbf{x}_i) \end{aligned} \tag{4.24}$$

This equation is sometimes referred to as the orthogonality condition. In matrix terms equations 4.23 and 4.24 can be written as

$$\begin{bmatrix} \mathbf{F} \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\mu} \end{bmatrix}$$

10 If $\| f \|$ is semi-norm, the kernel is positive semi-definite (because $\| f \|$ can be zero for a non-zero f). However, if $\| f \|$ is a norm the kernel is (strictly) positive-definite, i.e. $\| f \|$ is positive and non-zero for all non-zero fs.

so that

$$\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\mu} \end{bmatrix} = \begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{F} \\ \mathbf{0} \end{bmatrix}$$

Therefore,

$$\boldsymbol{\alpha} = \mathbf{L}_m^{-1}\mathbf{F} \quad \text{and} \quad \boldsymbol{\alpha}^T = \mathbf{F}(\mathbf{L}_m^{-1})^T$$

where $\mathbf{L}_m^{-1}$ is the m by m upper left submatrix $\begin{bmatrix} \mathbf{K} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix}^{-1}$. Thus Equation 4.22 becomes

$$\| f \|^2 = \mathbf{F}^T (\mathbf{L}_m^{-1})^T \mathbf{K} \mathbf{L}_m^{-1} \mathbf{F} = \mathbf{F}^T (\mathbf{K}\mathbf{L}_m^{-1})^T \mathbf{L}_m^{-1} \mathbf{F} \tag{4.25}$$

It is easy to show that the addition of the null space terms does not change the bending energy as follows:

$$\begin{aligned} \langle f, f \rangle &= \left\langle \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}), \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}) \right\rangle \\ &= \left\langle \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i), \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) \right\rangle + 2 \left\langle \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i), \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}) \right\rangle \\ &\quad + \left\langle \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}), \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}) \right\rangle \\ &= \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \end{aligned}$$

This is because the second term is zero from the orthogonality condition 4.24. The third term is also zero because the semi-norm 4.3 does not penalize the null space. A null space or order one is particularly meaningful, and is often used, because it implies that affine transformations, e.g. rotation and translation, are not penalized because they do not contribute to the bending of the spline. The Gaussian norm, in contrast, penalizes all derivatives.

The equation of the bending energy can alternatively be obtained as follows, without explicit recourse to the reproducing property, noting that from the relation A.19, Equation 4.17 can be expressed as

$$\int_{\mathbb{R}^d} \nabla^4 f^* \, \tilde{f} \, d\mathbf{x} + \int_{\mathbb{R}^d} \sum_{i=1}^{m} (f^*(\mathbf{x}_i) - y_i) \tilde{f}(\mathbf{x}) \, \delta(\mathbf{x} - \mathbf{x}_i) \, d\mathbf{x} = 0$$

where $\delta(\mathbf{x})$ is the d-dimensional dirac delta. Rearranging this expression and dropping first the integrals and then the term f yields

$$\nabla^4 f^* \, \tilde{f} = \sum_{i=1}^{m} \alpha_i \, \tilde{f} \, \delta(\mathbf{x} - \mathbf{x}_i)$$

and subsequently

$$\nabla^4 f = \sum_{i=1}^{m} \alpha_i \, \delta(\mathbf{x} - \mathbf{x}_i) \tag{4.26}$$

where as before, f^* has been replaced by f. From this relationship and Equation A.19, the bending energy can be given as

$$\langle f , f \rangle = \int_{\mathbb{R}^d} f \nabla^4 f \, dx = \int_{\mathbb{R}^d} \sum_{i=1}^{m} \alpha_i \, \delta(x - x_i) \, f \, dx$$

and because of the representer theorem (Equation 4.19)

$$\langle f , f \rangle = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \int_{\mathbb{R}^d} K(x , x_j) \, \delta(x - x_i) \, dx = \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(x_i , x_j)$$

because

$$\int_{\mathbb{R}^d} K(x , x_j) \, \delta(x - x_i) \, dx = K(x_i , x_j)$$

4.1.7 *Regularization with derivative information*

The previous section considered the recovery of a function f based on its values y_i at m sites x_i in a d-dimensional space. This section shows how information about the derivatives g_j of the function at m' number of sites can be used to recover f more accurately. As before, a functional $\mathcal{F}(f)$ assigns to each f a numeric value based on its norm (a measure of smoothness) and the degree to which it interpolates the sites y_i and derivatives g_j at any arbitrary directions t_j. That is to say,

$$\mathcal{F}[f] = \sum_{i=1}^{m} [f(x_i) - y_i]^2 + \sum_{j=1}^{m'} [t_j^T \nabla f(x_j) - g_j]^2 + \| f \|^2 \tag{4.27}$$

where $t_j^T \nabla f(x_j)$ is the derivative of f in the non-canonical direction t_j.

The rest of the analysis proceeds as before. The function f^* that best interpolates the data minimizes $\mathcal{F}[f]$ and is the value of f for which $\frac{\delta \mathcal{F}}{\delta f}$ is zero. However now,

$$\mathcal{F}[f^*] = \sum_{i=1}^{m} [f^*(x_i) - y_i]^2 + \sum_{j=1}^{m'} [t_j^T \nabla f(x_j) - g_j]^2 + \langle f^* , f^* \rangle$$

and

$$\mathcal{F}[f^* + \epsilon \tilde{f}] = \sum_{i=1}^{m} [f^*(x_i) + \epsilon \tilde{f}(x_i) - y_i]^2 + \sum_{j=1}^{m'} [t_j^T \nabla (f^* + \epsilon \tilde{f})(x_j) - g_j]^2 + \langle f^* + \epsilon \tilde{f} , f^* + \epsilon \tilde{f} \rangle$$

where $\| f^* \|^2 = \langle f^* , f^* \rangle$ and $\| f^* + \epsilon \tilde{f} \|^2 = \langle f^* + \epsilon \tilde{f} , f^* + \epsilon \tilde{f} \rangle$.

Because ∇ is a linear operator $\nabla(f^* + \epsilon \tilde{f})(x) = \nabla f^*(x) + \epsilon \nabla \tilde{f}(x)$, and

$$\begin{aligned} [t_j^T \nabla (f^* + \epsilon \tilde{f})(x_j) - g_j]^2 = \; & [t_j^T \nabla f^*(x_j)]^2 + \epsilon^2 [t_j^T \nabla \tilde{f}(x_j)]^2 + g_j^2 \\ & + 2 \epsilon [t_j^T \nabla f^*(x_j)] [t_j^T \nabla \tilde{f}(x_j)] - 2 \epsilon g_j [t_j^T \nabla \tilde{f}(x_j)] \\ & - 2 g_j [t_j^T \nabla f^*(x_j)] \end{aligned}$$

Therefore,

$$\begin{aligned}
\frac{\delta \mathcal{F}}{\delta f} &= \lim_{\epsilon \to 0} \frac{\mathcal{F}[\, f^* + \epsilon \tilde{f}\,] - \mathcal{F}[\, f^*\,]}{\epsilon} \\
&= \lim_{\epsilon \to 0} \frac{2\epsilon \langle\, f^*, \tilde{f}\,\rangle \;+\; \epsilon^2 \langle\, \tilde{f}, \tilde{f}\,\rangle \;+\; \sum_{i=1}^{m} 2\epsilon (f^*(x_i) - y_i)\tilde{f}(x_i) + \epsilon^2 f(x_i)^2}{\epsilon} \\
&\quad + \frac{\sum_{j=1}^{m'} \epsilon^2 \left[\, t_j^T \nabla \tilde{f}(x_j)\,\right]^2 \;+\; 2\,\epsilon \left[\, t_j^T \nabla f^*(x_j)\,\right] \left[\, t_j^T \nabla \tilde{f}(x_j)\,\right] \;-\; 2\,\epsilon\, g_j \left[\, t_j^T \nabla\, \tilde{f}(x_j)\,\right]}{\epsilon} \\
&= \lim_{\epsilon \to 0} 2\,\langle\, f^*, \tilde{f}\,\rangle \;+\; \epsilon\,\langle\, \tilde{f}, \tilde{f}\,\rangle \;+\; \sum_{i=1}^{m} 2\,(\, f^*(x_i) \;-\; y_i\,)\,\tilde{f}(x_i) \;+\; \epsilon\, f(x_i)^2 \\
&\quad + \sum_{j=1}^{m'} \epsilon \left[\, t_j^T \nabla\, \tilde{f}(x_j)\,\right]^2 \;+\; 2 \left[\, t_j^T \nabla f^*(x_j)\,\right] \left[\, t_j^T \nabla \tilde{f}(x_j)\,\right] \;-\; 2\, g_j \left[\, t_j^T \nabla\, \tilde{f}(x_j)\,\right] \\
&= 2\,\langle\, f^*, \tilde{f}\,\rangle \;+\; 2 \sum_{i=1}^{m} [\, f^*(x_i) - y_i\,]\;\tilde{f}(x_i) \;+\; \sum_{k=1}^{m'} 2 \left[\, t_j^T \nabla f^*(x_j)\,\right] \left[\, t_j^T \nabla \tilde{f}(x_j)\,\right] \;-\; 2\, g_j \left[\, t_j^T \nabla\, \tilde{f}(x_j)\,\right] \\
&= 2\,\langle\, f^*, \tilde{f}\rangle \;+\; 2 \sum_{i=1}^{m} [\, f^*(x_i) - y_i\,]\;\tilde{f}(x_i) \;+\; 2 \sum_{j=1}^{m'} \left\{ \left[\, t_j^T \nabla f^*(x_j)\,\right] \;-\; g_j \right\} \left[\, t_j^T \nabla \tilde{f}(x_j)\,\right]
\end{aligned}$$

As in the previous section, $\tilde{f} = K_x$ is a reproducing kernel that evaluates f^* at x, i.e. $f^*(x) = \langle f^*, K_x \rangle$. Furthermore, if $\alpha_i = y_i - f^*(x_i)$ and $\alpha'_i = g_j - t_j^T \nabla f^*(x_j)$, equating $\frac{\delta \mathcal{F}}{\delta f}$ to zero and rearranging yields

$$f^*(x) \;=\; \sum_{i=1}^{m} \alpha_i\; k(\, x\,,\, x_i\,) \;+\; \sum_{j=1}^{m'} \alpha'_j\; t_j^T \nabla \tilde{f}(x_j) \tag{4.28}$$

Where

$$\begin{aligned}
t_j^T \nabla \tilde{f}(x_j) &= \sum_{l=1}^{d} t_j[l]\; \frac{\tilde{f}(\, x_j \;+\; \epsilon\, e[l]\,) \;-\; \tilde{f}(\, x_j\,)}{\epsilon} \\
&= \lim_{\epsilon \to 0} \sum_{l=1}^{d} t_j[l]\; \frac{k(\, x\,,\, x_j \;+\; \epsilon\, e[l]\,) \;-\; k(\, x\,,\, x_j\,)}{\epsilon} \\
&= t_j^T \nabla_{x_j} k(\, x\,,\, x_j\,)
\end{aligned} \tag{4.29}$$

Note that, $\nabla_{x_j} k(\, x\,,\, x_j\,)$ measures the rates of change of $k(\, x\,,\, x_j\,)$ with respect to x_j, while x is held constant.[11] That is to say

$$\nabla_{x_j} k(\, x\,,\, x_j\,) \;=\; \left(\frac{\partial k(\, x\,,\, x_j\,)}{\partial x_j[1]}\,,\, \cdots\,,\, \frac{\partial k(\, x\,,\, x_j\,)}{\partial x_j[d]} \right)^T \tag{4.30}$$

But

$$k(\, x\,,\, x_j\,) \;=\; k(\,\|\, x \;-\; x_j \,\|\,) \;=\; k(\, r\,) \quad \text{where} \quad r \;=\; \sqrt{\sum_{l=1}^{d} \left(\, x[l] \;-\; x_j[l]\,\right)^2}$$

11 $\nabla_{x_j} k(\, x\,,\, x_j\,)$ emphasizes that the terms of the gradient operator are differentials with respect to $x_j[l]$, while $\nabla k(\, x\,,\, x_j\,)$ emphasizes that the terms of the gradient operator are differentials with respect to $x[l]$.

so that by the chain rule

$$\frac{\partial r}{\partial x[l]} = \frac{x[l] - x_j[l]}{r} \quad \text{and} \quad \frac{\partial r}{\partial x_j[l]} = -\frac{x[l] - x_j[l]}{r} \quad \Rightarrow \quad \frac{\partial r}{\partial x[l]} = -\frac{\partial r}{\partial x_j[l]} \tag{4.31}$$

Again by the chain rule,

$$\frac{\partial k(r)}{\partial x[l]} = \frac{\partial k(r)}{\partial r}\frac{\partial r}{\partial x[l]} \tag{4.32}$$

$$\frac{\partial k(r)}{\partial x_j[l]} = \frac{\partial k(r)}{\partial r}\frac{\partial r}{\partial x_j[l]} = -\frac{\partial k(r)}{\partial r}\frac{\partial r}{\partial x[l]} \quad \text{from Equation 4.31} \tag{4.33}$$

Therefore

$$\nabla_{x_j} k(\mathbf{x}, \mathbf{x}_j) = -\left(\frac{\partial k(\mathbf{x}, \mathbf{x}_j)}{\partial x[1]}, \cdots, \frac{\partial k(\mathbf{x}, \mathbf{x}_j)}{\partial x[d]}\right)^T = -\nabla k(\mathbf{x}, \mathbf{x}_j) \tag{4.34}$$

so that, from Equations 4.29 and 4.34, 4.34 becomes

$$f^*(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \, k(\mathbf{x}, \mathbf{x}_i) - \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_j^T \nabla k(\mathbf{x}, \mathbf{x}_j) \tag{4.35}$$

taking the null space of $\| f \|$ into consideration, by adding $n = \binom{d+q-1}{d}$ polynomial terms to f and dropping the asterisk, equation 4.35 becomes

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \, k(\mathbf{x}, \mathbf{x}_i) - \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_j^T \nabla k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{n} \mu_j \, \mathbf{p}_j(\mathbf{x}) \tag{4.36}$$

Differentiating equation 4.36 with respect to each direction vector $\mathbf{t}_j$ produces the following additional m' system of equations

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{t}_1} = \mathbf{t}_1^T \nabla f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \, \mathbf{t}_1^T \nabla k(\mathbf{x}, \mathbf{x}_i) - \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_1^T \mathbf{t}_j^T \nabla^2 k(\mathbf{x}, \mathbf{x}_j) + \sum_{j=1}^{n} \mu_j \, \mathbf{t}_1^T \nabla \mathbf{p}_j(\mathbf{x})$$

$$\cdots$$

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{t}_{m'}} = \mathbf{t}_{m'}^T \nabla f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i \, \mathbf{t}_{m'}^T \nabla k(\mathbf{x}, \mathbf{x}_i) - \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_{m'}^T \mathbf{t}_j^T \nabla^2 k(\mathbf{x}, \mathbf{x}_j) + \sum_{j=1}^{n} \mu_j \, \mathbf{t}_{m'}^T \nabla \mathbf{p}_j(\mathbf{x}) \tag{4.37}$$

Substituting the m values of y for $f(\mathbf{x})$ into 4.36 produces m equations; while substituting the m' gradients in the direction $\mathbf{t}_j$ into $\frac{\partial f(\mathbf{x}_i)}{\partial \mathbf{t}_l}$ in 4.37 produces m' equations. However, as the equation system has $(m + m' + n)$ unknowns, n additional equations are required. As before, these are obtained by imposing orthogonality conditions on 4.35.

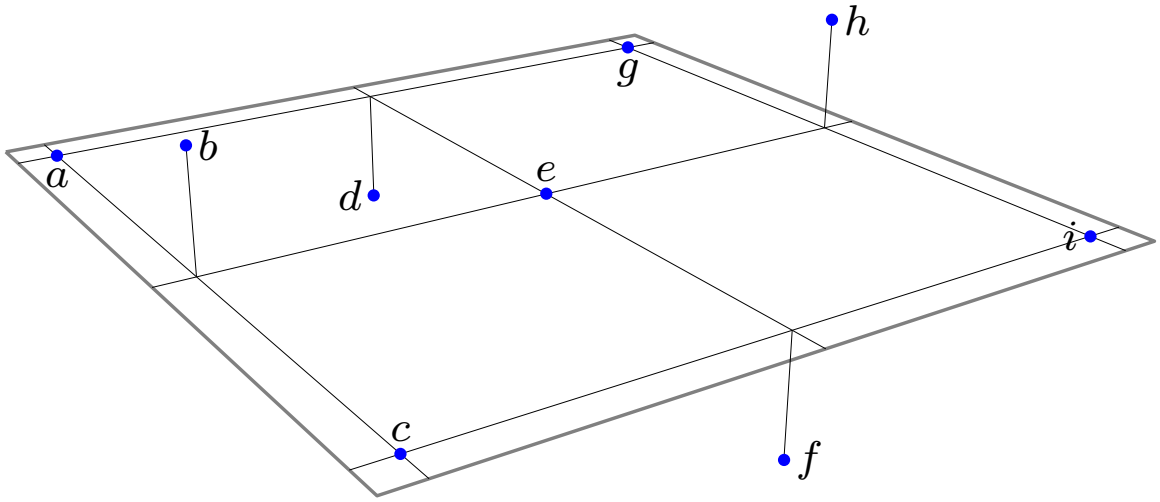


Figure 4.1 A set of nine points, $a = (0.5, 0.5, 0.0)$, $b = (5.0, 0.5, 1.5)$, $c = (9.5, 0.5, 0.0)$, $d = (0.5, 5.0, -1.5)$, $e = (5.0, 5.0, 0.0)$, $f = (9.5, 5.0, -1.5)$, $g = (0.5, 9.5, 0.0)$, $h = (5.0, 9.5, 1.5)$, $i = (9.5, 9.5, 0.0)$.

Also as before, the polynomial terms are orthogonal to the representer of the evaluation. Therefore, for each one of the n polynomial terms $p(\mathbf{x})$,

$$\begin{aligned} 0 &= \left\langle \sum_{i=1}^{m} \alpha_i \, k(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_j^T \nabla k(\mathbf{x}, \mathbf{x}_j) \, , \, \mathbf{p}(\mathbf{x}) \right\rangle \\ &= \sum_{i=1}^{m} \alpha_i \left\langle k(\mathbf{x}, \mathbf{x}_i) \, , \, \mathbf{p}(\mathbf{x}) \right\rangle + \left\langle \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_j^T \nabla k(\mathbf{x}, \mathbf{x}_j) \, , \, \mathbf{p}(\mathbf{x}) \right\rangle \\ &= \sum_{i=1}^{m} \alpha_i \, \mathbf{p}(\mathbf{x}_i) + \sum_{j=1}^{m'} \alpha'_j \, \mathbf{t}_j^T \nabla \mathbf{p}(\mathbf{x}_j) \end{aligned} \tag{4.38}$$

where the last step used the derivative reproducing property

$$\left\langle \mathbf{t}_j^T \nabla k(\mathbf{x}, \mathbf{x}_j) \, , \, \mathbf{p}(\mathbf{x}) \right\rangle = \mathbf{t}_j^T \nabla \mathbf{p}(\mathbf{x}_j)$$

the proof of which is given in appendix A.1.6.

4.2 BASIC APPLICATIONS

Beginning with the motivating problem, described in the introduction to this chapter, this section presents a series of toy examples illustrating five applications of thin plate splines. These techniques are applied in subsequent chapters of this thesis.

4.2.1 *Height-field interpolation*

From Equation 4.23, the two-dimensional thin plate spline

$$f(\mathbf{x}) = \sum_{i=1}^{9} \alpha_i K(\| \mathbf{x} - \mathbf{x}_i \|) + \mu_1 + \mu_2 x_i[0] + \mu_3 x_i[1] \tag{4.39}$$

with a null space or order 1, i.e. consisting of linear polynomial terms, can be used to interpolate nine irregularly spaced points $\mathbf{x}_i = (x_i[0], x_i[1])$, shown in Figure 4.1, having different vertical offsets $f(\mathbf{x}_i)$ from the horizontal plane, where $K(\| \mathbf{x} - \mathbf{x}_i \|)$ is defined in 4.9. Substituting each $\mathbf{x}_i$ into Equation 4.39 results in a system of nine equations. As there are 12 unknowns, the remaining three equations are the following orthogonality conditions

$$\sum_{i=1}^{9} \alpha_i = 0 \quad \sum_{i=1}^{9} \alpha_i x_i[0] = 0 \quad \sum_{i=1}^{9} \alpha_i x_i[1] = 0 \tag{4.40}$$

giving rise to the matrix system

$$\mathbf{Y} = \mathbf{L}\boldsymbol{w} \tag{4.41}$$

so that the weights μ and $\boldsymbol{\alpha}$ can be obtained from

$$\boldsymbol{w} = \mathbf{L}^{-1}\mathbf{Y} \tag{4.42}$$

where

$$\mathbf{L} = \begin{bmatrix} K(\| \mathbf{x}_1 - \mathbf{x}_1 \|) & K(\| \mathbf{x}_1 - \mathbf{x}_2 \|) & \cdots & K(\| \mathbf{x}_1 - \mathbf{x}_9 \|) & 1 & x_1[0] & x_1[1] \\ K(\| \mathbf{x}_2 - \mathbf{x}_1 \|) & K(\| \mathbf{x}_2 - \mathbf{x}_2 \|) & \cdots & K(\| \mathbf{x}_2 - \mathbf{x}_9 \|) & 1 & x_2[0] & x_2[1] \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\| \mathbf{x}_9 - \mathbf{x}_1 \|) & K(\| \mathbf{x}_9 - \mathbf{x}_2 \|) & \cdots & K(\| \mathbf{x}_9 - \mathbf{x}_9 \|) & 1 & x_9[0] & x_9[1] \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1[0] & x_2[0] & \cdots & x_9[0] & 0 & 0 & 0 \\ x_1[1] & x_2[1] & \cdots & x_9[1] & 0 & 0 & 0 \end{bmatrix}$$

$$\mathbf{w} = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_9 & \mu_1 & \mu_2 & \mu_3 \end{bmatrix}^T$$

$$\mathbf{Y} = \begin{bmatrix} f(\mathbf{x}_1) & f(\mathbf{x}_2) & \cdots & f(\mathbf{x}_9) & 0 & 0 & 0 \end{bmatrix}^T$$

As shown in Figure 4.2, the height field smoothly interpolates all nine points.

In addition, it is possible to constrain the gradient of the height field to any value $\mathbf{t}^T \nabla f(\mathbf{x}_i)$ at a given point $\mathbf{x}_i$ and direction $\mathbf{t} = (t[0], t[1])$. From Equation 4.36, the corresponding thin-plate spline is given by

$$f(\mathbf{x}) = \sum_{i=1}^{9} \alpha_i K(\mathbf{r}_i) - \alpha' \mathbf{t}^T \nabla K(\mathbf{r}_5) + \mu_1 + \mu_2 x_i[0] + \mu_3 x_i[1] + \mu_4 x_i[0] x_i[1] + \mu_5 x_i[0]^2 + \mu_6 x_i[1]^2 \tag{4.43}$$

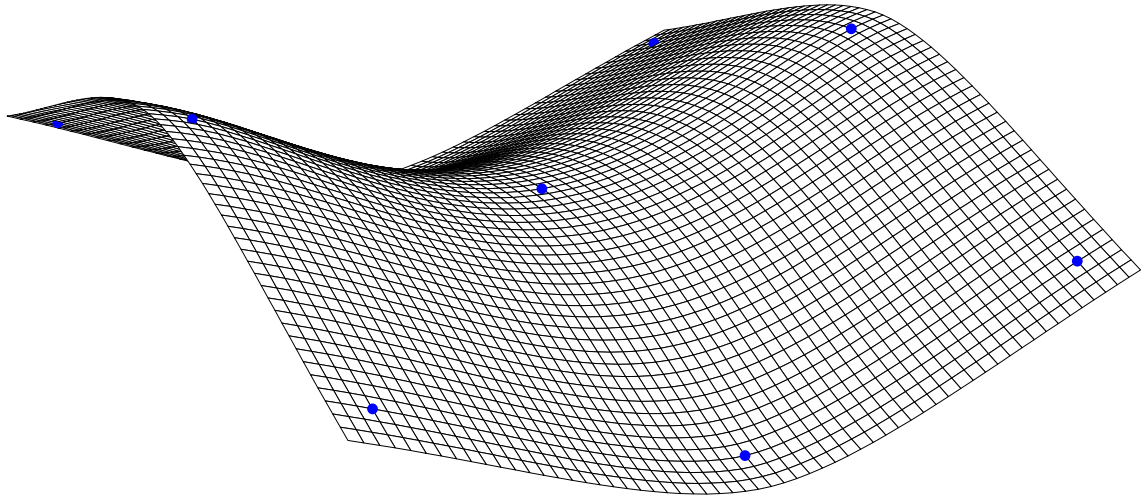


Figure 4.2 Height field interpolating the set of nine points shown in Figure 4.1.

This thin-plate spline has a null space of order 2 and interpolates the set of nine points in Figure 4.2, where $\mathbf{r}_i = \| \mathbf{x}_1 - \mathbf{x}_2 \|$ and $K(\mathbf{r}) = \mathbf{r}^4 \log(\mathbf{r})$. The gradient of the height field in the direction $\mathbf{t} = (1, 1)$ can be specified at the fifth point by differentiating Equation 4.43 with respect to $\mathbf{t}$ in order to obtain

$$\mathbf{t}^T \nabla f(\mathbf{x}) = \sum_{i=1}^{9} \alpha_i \mathbf{t}^T \nabla K(\mathbf{r}_i) - \alpha' \mathbf{t}^T \nabla \left[\mathbf{t}^T \nabla K(\mathbf{r}_5) \right] + 0 + \mu_2 \mathbf{t}^T \nabla x_i[0] + \mu_3 \mathbf{t}^T \nabla x_i[1] + \mu_4 \mathbf{t}^T \nabla (x_i[0] x_i[1]) + \mu_5 \mathbf{t}^T \nabla (x_i[0]^2) + \mu_6 \mathbf{t}^T \nabla (x_i[1]^2) \tag{4.44}$$

From Equation 4.38 the six required orthogonality conditions are

$$\begin{aligned}
\sum_{i=1}^{9} \alpha_i &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[0] + \alpha' \mathbf{t}^T \nabla (x_5[0]) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[1] + \alpha' \mathbf{t}^T \nabla (x_5[1]) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[0] x_i[1] + \alpha' \mathbf{t}^T \nabla (x_5[0] x_5[1]) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[0]^2 + \alpha' \mathbf{t}^T \nabla (x_5[0]^2) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[1]^2 + \alpha' \mathbf{t}^T \nabla (x_5[1]^2) &= 0
\end{aligned} \tag{4.45}$$

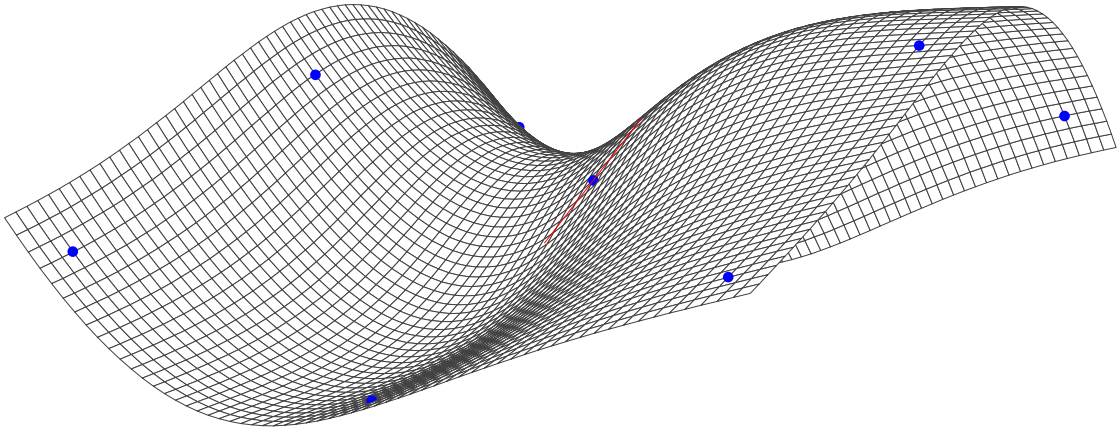


Figure 4.3 Height field interpolating the set of nine points shown in Figure 4.1, with the gradient at point e constrained to 1.0 in the direction $(1, 1)$.

The derivatives in Equations 4.43, 4.44 and 4.45 are typically given by

$$\begin{aligned}
\mathbf{t}^T \nabla K(\mathbf{r}) &= \left(\frac{\partial K(\mathbf{r})}{\partial x[0]}, \frac{\partial K(\mathbf{r})}{\partial x[1]} \right) \cdot (\, t[0], t[1] \,) = t[0] \frac{\partial K(\mathbf{r})}{\partial x[0]} + t[1] \frac{\partial K(\mathbf{r})}{\partial x[1]} \\
\mathbf{t}^T \nabla \left[\mathbf{t}^T \nabla K(\mathbf{r}) \right] &= \mathbf{t}^T \nabla \left(t[0] \frac{\partial K(\mathbf{r})}{\partial x[0]} + t[1] \frac{\partial K(\mathbf{r})}{\partial x[1]} \right) = \nabla \left(\frac{\partial K(\mathbf{r})}{\partial x[0]} + t[1] \frac{\partial K(\mathbf{r})}{\partial x[1]} \right) \cdot \mathbf{t} \\
&= \left(t[0] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]^2} + t[1] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]\, \partial x[1]}, t[0] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]\, \partial x[1]} + t[1] \frac{\partial^2 K(\mathbf{r})}{\partial x[1]^2} \right) \cdot (\, t[0], t[1] \,) \\
&= t[0]^2 \frac{\partial^2 K(\mathbf{r})}{\partial x[0]^2} + 2\, t[0]\, t[1] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]\, \partial x[1]} + t[1]^2 \frac{\partial^2 K(\mathbf{r})}{\partial x[1]^2}
\end{aligned}$$

where $\mathbf{t} = (\, t[0], t[1] \,)$. The partial derivatives of $\mathbf{x}_i$ follow similar steps. The system of Equations 4.43, 4.44 and 4.45 can be solved for the weights

$$\mathbf{w} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_9 & \alpha' & \mu_1 & \cdots & \mu_6 \end{bmatrix}^T$$

by inverting the matrix system 4.41, where $\mathbf{L}$ is given as

$$\begin{bmatrix}
K(\mathbf{x}_1, \mathbf{x}_1) & \cdots & K(\mathbf{x}_1, \mathbf{x}_9) & -\mathbf{t}^T \nabla K(\mathbf{x}_1, \mathbf{x}_5) & 1 & x_1[0] & x_1[1] & x_1[0]\, x_1[1] & x_1[0]^2 & x_1[1]^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
K(\mathbf{x}_9, \mathbf{x}_1) & \cdots & K(\mathbf{x}_9, \mathbf{x}_9) & -\mathbf{t}^T \nabla K(\mathbf{x}_9, \mathbf{x}_5) & 1 & x_1[0] & x_9[1] & x_9[0]\, x_9[1] & x_9[0]^2 & x_9[1]^2 \\
\mathbf{t}^T \nabla K(\mathbf{x}_5, \mathbf{x}_1) & \cdots & \mathbf{t}^T \nabla K(\mathbf{x}_5, \mathbf{x}_9) & 0 & 0 & \mathbf{t}^T \nabla x_5[0] & \mathbf{t}^T \nabla x_5[1] & \mathbf{t}^T \nabla x_5[0]\, x_5[1] & \mathbf{t}^T \nabla x_5[0]^2 & \mathbf{t}^T \nabla x_5[1]^2 \\
1 & \cdots & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
x_1[0] & \cdots & x_9[0] & \mathbf{t}^T \nabla (\, x_5[0] \,) & 0 & 0 & 0 & 0 & 0 & 0 \\
x_1[1] & \cdots & x_9[1] & \mathbf{t}^T \nabla (\, x_5[1] \,) & 0 & 0 & 0 & 0 & 0 & 0 \\
x_1[0]\, x_1[1] & \cdots & x_9[0]\, x_9[1] & \mathbf{t}^T \nabla (\, x_i[0]\, x_i[1] \,) & 0 & 0 & 0 & 0 & 0 & 0 \\
x_1[0]^2 & \cdots & x_9[0]^2 & \mathbf{t}^T \nabla (\, x_5[0]^2 \,) & 0 & 0 & 0 & 0 & 0 & 0 \\
x_1[1]^2 & \cdots & x_9[1]^2 & \mathbf{t}^T \nabla (\, x_5[1]^2 \,) & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

and

$$\mathbf{Y} = \begin{bmatrix} f(\mathbf{x}_1) & f(\mathbf{x}_2) & \cdots & f(\mathbf{x}_9) & \mathbf{t}^T \nabla f(\mathbf{x}_5) & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T$$

The resulting height field is shown in Figure 4.3.

Furthermore, it is possible to impose multiple gradient constraints $\mathbf{t}^T\nabla f(\mathbf{x}_i)$ and $\mathbf{h}^T\nabla f(\mathbf{x}_i)$ at any given point $\mathbf{x}_i$ and directions $\mathbf{t} = (t[0], t[1])$ and $\mathbf{h} = (h[0], h[1])$. In this case the thin plate spline is given by

$$f(\mathbf{x}) = \sum_{i=1}^{9} \alpha_i K(\mathbf{r}_i) - \alpha' \mathbf{t}^T\nabla K(\mathbf{r}_5) - \alpha'' \mathbf{h}^T\nabla K(\mathbf{r}_5) + \mu_1 + \mu_2 x_i[0] + \mu_3 x_i[1] + \mu_4 x_i[0] x_i[1] + \mu_5 x_i[0]^2 + \mu_6 x_i[1]^2 \tag{4.46}$$

assuming as before a null space of order 2 and $K(\mathbf{r}) = \mathbf{r}^4 \log(\mathbf{r})$. However in this case,

$$\mathbf{t}^T\nabla f(\mathbf{x}) = \sum_{i=1}^{9} \alpha_i \mathbf{t}^T\nabla K(\mathbf{r}_i) - \alpha' \mathbf{t}^T\nabla \left[\mathbf{t}^T\nabla K(\mathbf{r}_5)\right] - \alpha'' \mathbf{t}^T\nabla \left[\mathbf{h}^T\nabla K(\mathbf{r}_5)\right] + 0 + \mu_2 \mathbf{t}^T\nabla(x_i[0]) + \mu_3 \mathbf{t}^T\nabla(x[1]) + \mu_4 \mathbf{t}^T\nabla(x[0]x[1]) + \mu_5 \mathbf{t}^T\nabla(x[0]^2) + \mu_6 \mathbf{t}^T\nabla(x[1]^2) \tag{4.47}$$

and

$$\mathbf{h}^T\nabla f(\mathbf{x}) = \sum_{i=1}^{9} \alpha_i \mathbf{h}^T\nabla K(\mathbf{r}_i) - \alpha' \mathbf{h}^T\nabla \left[\mathbf{t}^T\nabla K(\mathbf{r}_5)\right] - \alpha'' \mathbf{h}^T\nabla \left[\mathbf{h}^T\nabla K(\mathbf{r}_5)\right] + 0 + \mu_2 \mathbf{h}^T\nabla(x_i[0]) + \mu_3 \mathbf{h}^T\nabla(x[1]) + \mu_4 \mathbf{h}^T\nabla(x[0]x[1]) + \mu_5 \mathbf{h}^T\nabla(x[0]^2) + \mu_6 \mathbf{h}^T\nabla(x[1]^2) \tag{4.48}$$

while the orthogonality conditions are:

$$\begin{aligned}
\sum_{i=1}^{9} \alpha_i &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[0] + \alpha' \mathbf{t}^T\nabla(x_5[0]) + \alpha'' \mathbf{h}^T\nabla(x_5[0]) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[1] + \alpha' \mathbf{t}^T\nabla(x_5[1]) + \alpha'' \mathbf{h}^T\nabla(x_5[1]) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[0] x_i[1] + \alpha' \mathbf{t}^T\nabla(x_i[0] x_i[1]) + \alpha'' \mathbf{h}^T\nabla(x_i[0] x_i[1]) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[0]^2 + \alpha' \mathbf{t}^T\nabla(x_5[0]^2) + \alpha'' \mathbf{h}^T\nabla(x_5[0]^2) &= 0 \\
\sum_{i=1}^{9} \alpha_i x_i[1]^2 + \alpha' \mathbf{t}^T\nabla(x_5[1]^2) + \alpha'' \mathbf{h}^T\nabla(x_5[1]^2) &= 0
\end{aligned} \tag{4.49}$$

noting that the mixed directional derivative is given as

$$\begin{aligned}
\mathbf{t}^T\nabla \left[\mathbf{h}^T\nabla K(\mathbf{r})\right] &= \mathbf{t}^T\nabla \left(h[0] \frac{\partial K(\mathbf{r})}{\partial x[0]} + h[1] \frac{\partial K(\mathbf{r})}{\partial x[1]} \right) \\
&= \left(h[0] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]^2} + h[1] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]\,\partial x[1]}, h[0] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]\,\partial x[1]} + h[1] \frac{\partial^2 K(\mathbf{r})}{\partial x[1]^2} \right) \cdot (t[0], t[1]) \\
&= h[0]\,t[0] \frac{\partial^2 K(\mathbf{r})}{\partial x[0]^2} + (h[0]\,t[1] + h[1]\,t[0]) \frac{\partial^2 K(\mathbf{r})}{\partial x[0]\,\partial x[1]} + h[1]\,t[1] \frac{\partial^2 K(\mathbf{r})}{\partial x[1]^2}
\end{aligned} \tag{4.50}$$

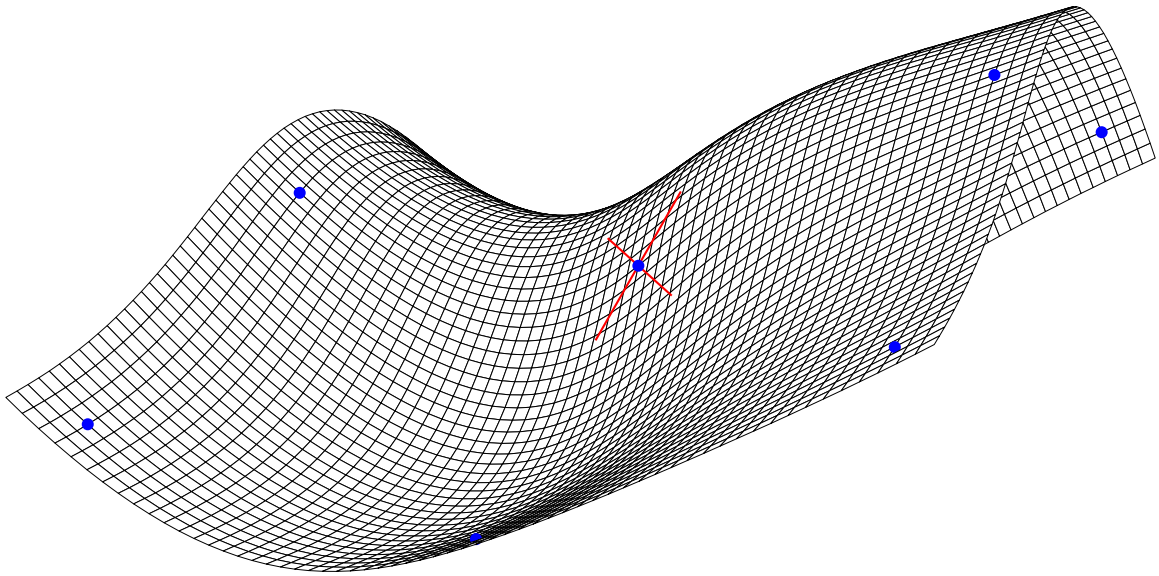


Figure 4.4 Height field interpolating the set of nine points shown in Figure 4.1, with the gradients at point e constrained to 1.0 and 0.25 in the directions $(1,1)$ and $(-1,1)$ respectively.

Equations 4.46, 4.47, 4.48 and 4.49 again can be simultaneously solved for the weights

$$\mathbf{w} = \begin{bmatrix} \alpha_1 & \cdots & \alpha_9 & \alpha' & \alpha'' & \mu_1 & \cdots & \mu_6 \end{bmatrix}^T$$

The resulting height field is shown in Figure 4.4.

4.2.2 *Patch-based differential geometry*

A polygonal mesh is an approximation of a smooth surface, the implicit (or explicit) function of which is often unknown. This smooth surface can be obtained by generalizing the height field interpolation technique, as shown in the next section. From such a representation, differential geometry attributes such as principal tangents and curvatures can be computed at any point. However, if the representation of an entire mesh is too costly to compute, and the differential geometry properties of the mesh are desired at a small number of points, the required attributes can be computed on thin-plate splines fitted in the neighborhood of each point of interest. This is done as follows:

i. Select an N-ring of vertices round the point of interest **P** as shown in Figure 4.5(a) and (b).

ii. Transform the selected vertexes to a local coordinate system, whose origin is **P**, and up direction is the normal of the polygon face on which **P** lies.

iii. Fit a thin plate spline to the transformed ring vertices, as described in Section 4.2.1. The resulting height field, e.g. see Figure 4.6, is known as a monge patch. Further-

more, as illustrated in the previous section, the gradient vector $\left(-\frac{n_i[0]}{n_i[2]}, -\frac{n_i[1]}{n_i[2]}\right)$ in the directions $x[0]$ and $x[1]$ can be brought to bear in patch fitting, where the (transformed) vertex normals of the N-ring of vertices are $\mathbf{n}_i = (n_i[0], n_i[1], n_i[2])$.

iv. Because $\mathbf{P}$ generally does not lie on the monge patch, it must be projected onto the monge patch in order to obtain its true position $\mathbf{P}'$ on the monge patch. This projection step is described subsequently. Note that if $\mathbf{P}$ is a mesh vertex it lies on the monge patch and there is no need for the projection step.

v. The desired differential quantities such as the principal tangents and curvatures can now be computed at the point $\mathbf{P}'$ on the monge patch. However, the surface normal and principal directions computed must be transformed back to the original (world) coordinate system in order to be meaningful.

As shown in Figure 4.5(c), projecting the point $\mathbf{P}$ to the monge patch is done by inflating a hypothetical sphere centred at $\mathbf{P}$ until it touches the monge patch $f(\mathbf{x}) = z$, where z is the displacement or height of the patch at $\mathbf{x}$. The equation of the patch can be expressed in implicit form as $F(\mathbf{x}, z) = f(\mathbf{x}) - z$, implying that $F(\mathbf{x}, z) = 0$ at every point on the patch. The equation of the sphere is given as $R(\mathbf{x}, z) = x[0]^2 + x[1]^2 + z^2$. Inflating the sphere amounts to minimizing $R(\mathbf{x}, z)$ subject to the constraint $F(\mathbf{x}, z)$. This can be done by introducing the Lagrange multiplier t in order to obtain the equation

$$H(\mathbf{x}, z, t) = R(\mathbf{x}, z) + t\,F(\mathbf{x}, z)$$

minimizing H by equating its partial derivatives to zero produces the following system of equations

$$\begin{aligned}
\frac{\partial H}{\partial x[0]} &= 2\,x[0] + t\,\frac{\partial F}{\partial x[0]} = 0 \\
\frac{\partial H}{\partial x[1]} &= 2\,x[1] + t\,\frac{\partial F}{\partial x[1]} = 0 \\
\frac{\partial H}{\partial z} &= 2\,z - t = 0 \\
\frac{\partial H}{\partial z} &= F = 0
\end{aligned}$$

noting that the last equation is the original constraint and that because $t = 2\,z$ the above equations can be simplified as follows

$$\begin{aligned}
x[0] + z\,\frac{\partial F}{\partial x[0]} &= 0 \\
x[1] + z\,\frac{\partial F}{\partial x[1]} &= 0 \\
F &= 0
\end{aligned}$$

This non-linear system can be solved for $\mathbf{x} = (x[0], x[1])$ and z using a multidimensional root solver, noting that the desired projection $\mathbf{P}' = (x[0], x[1], z)$.

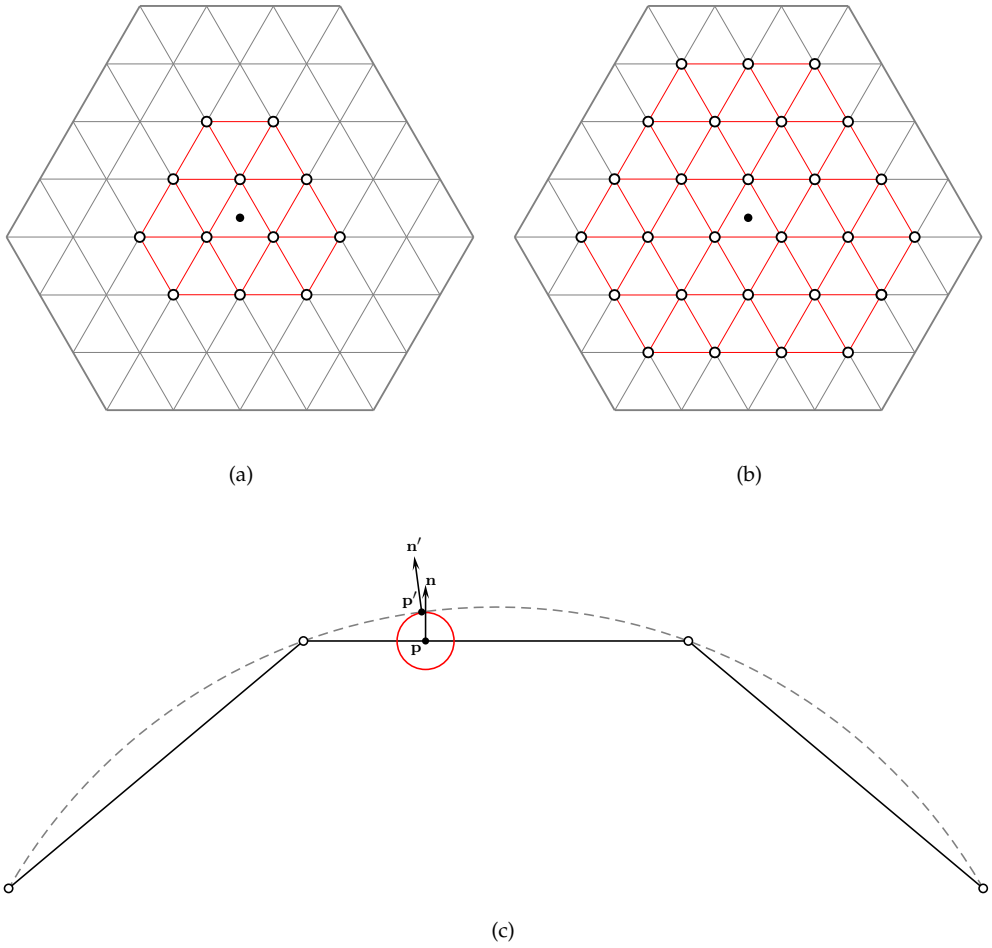


Figure 4.5 (a) Two and (b) three ring neighborhood vertices around a point of interest. (c) Projecting a point of interest $\mathbf{p}$ on a polygonal mesh to the point $\mathbf{p}'$ on thin-plate spline, by inflating an osculating sphere.

Following Smith and Séquin (2003), the principal curvatures and directions at $\mathbf{P}'$ are eigenvalues and eigenvectors of the Weingarten operator

$$W = \mathrm{I}^{-1}\mathrm{II}$$

where I and II are the first and second fundamental forms defined as

$$\mathrm{I} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} \quad \text{and} \quad \mathrm{II} = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$$

where according to Srinark (2008)

$$E = 1 + \left(\frac{\partial f}{\partial x[0]}\right)^2 \qquad F = 1 + \frac{\partial f}{\partial x[0]}\frac{\partial f}{\partial x[1]} \qquad G = 1 + \left(\frac{\partial f}{\partial x[1]}\right)^2$$

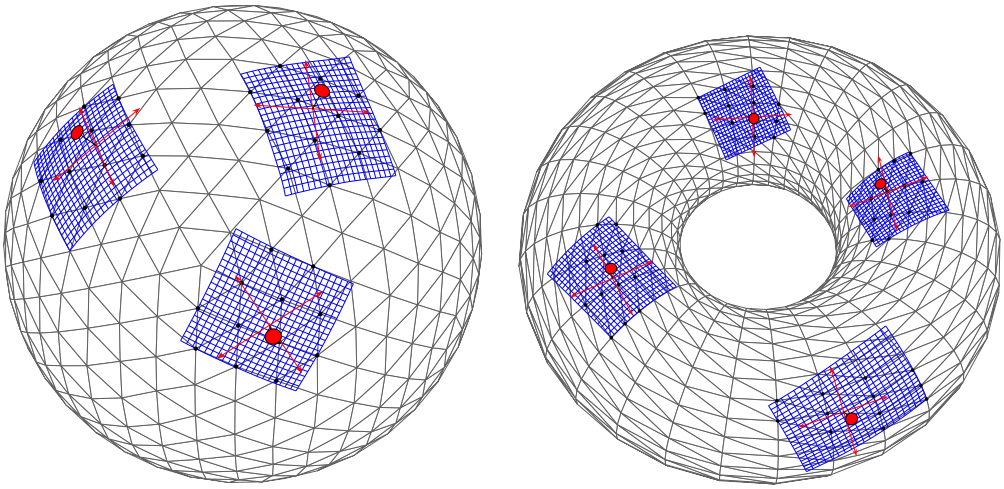


Figure 4.6 Fitting monge patches (blue mini-grids) to the n-ring neighborhood vertices of several points of interest (red pins).

and

$$L = \frac{1}{\sqrt{EG - F^2}} \frac{\partial^2 f}{\partial x[0]^2} \qquad M = \frac{1}{\sqrt{EG - F^2}} \frac{\partial^2 f}{\partial x[0]\partial x[1]} \qquad N = \frac{1}{\sqrt{EG - F^2}} \frac{\partial^2 f}{\partial x[1]^2}$$

Because

$$\begin{aligned} I^{-1} &= \frac{1}{EG - F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix} \\ W &= \frac{1}{EG - F^2} \begin{bmatrix} G & -F \\ -F & E \end{bmatrix} \begin{bmatrix} L & M \\ M & N \end{bmatrix} \\ &= \frac{1}{EG - F^2} \begin{bmatrix} GL - FM & GM - FN \\ EM - FL & EN - FM \end{bmatrix} \end{aligned}$$

the principal curvatures of W are (from Smith and Séquin (2003))

$$\kappa_{1,2} = \frac{(a + d) \pm \sqrt{(a - d)^2 + 4bc}}{2a}$$

while its eigenvectors are

$$v_{1,2} = \begin{bmatrix} \frac{(a-d) \mp \sqrt{(a-d)^2+4bc}}{2a} \\ 1 \end{bmatrix}$$

where

$$a = GL - FM \qquad b = GM - FN \qquad c = EM - FL \qquad d = EN - FM$$

So that the principal tangents are

$$\begin{aligned} t_1 &= \left[v_1[0] \, , \, v_1[1] \, , \left(v_1[0] \frac{\partial f}{\partial x[0]} + v_1[1] \frac{\partial f}{\partial x[1]} \right) \right]^T \\ t_2 &= \left[v_2[0] \, , \, v_2[1] \, , \left(v_2[0] \frac{\partial f}{\partial x[0]} + v_2[1] \frac{\partial f}{\partial x[1]} \right) \right]^T \end{aligned}$$

Unlike the popular method of least-squares polynomial fitting in which the resulting monge patch merely approximates but does not interpolate the selected points (e.g. Cazals and Pouget (2005), Goldfeather and Interrante (2004)), the thin plate spline method exactly interpolates all selected neighboring vertices and is therefore more suited to analyzing noise-free, artist-modeled meshes.[12] As such, the least-squares fitting technique is perhaps best restricted to point clouds and very noisy meshes, for which the earlier assumption that all points lie on a true surface does not hold. Furthermore the method of computing differential geometry quantities using thin-plate splines is not sensitive to mesh topology or restricted to triangulated meshes or restricted to mesh vertices alone, as are some curvature estimation techniques e.g. Meyer *et al.* (2003).

4.2.3 *Implicit surface construction*

In the previous section the explicit equation of a surface $f(\mathbf{x}) = z$ was replaced by the implicit form $F(\mathbf{x}, z) = f(\mathbf{x}) - z$, where $F(\mathbf{x}, z) = 0$ at every point on the patch. The implicit form is of particular interest because the magnitude and sign of $F(\mathbf{x}, z)$ is indicative of the distance of $\mathbf{x}$ from the surface and whether $\mathbf{x}$ is inside or outside the surface. In general terms, $F(\mathbf{x}, z)$ is said to generate a scalar field. For example the implicit equation of a sphere is $F(\mathbf{x}, R) = R^2 - (x[0]^2 + x[1]^2 + x[2]^2) = 0$. Inside the sphere, the field $F(\mathbf{x}, R)$ has positive values, but has negative values outside it. As described in Section 4.2.1, the scalar field generated by an implicit form (possibly unknown), can be reconstructed by fitting a thin plate spline to a set of known field values. This can be done in two ways. In both cases a set of points $\mathbf{x}$ on the implicit surface, having zero field values, is known. However fitting a thin-plate spline on this set of zero field values produces a naive solution or field that is zero everywhere. In the variational implicit surface method formalized by Turk and O'Brien (2002) the points $\mathbf{x}$ are offset normally (i.e. inward or outward) and assigned ad-hoc field values of ± 1, and a non-trivial solution is obtained by solving the system of Equations 4.23 and 4.24, where $f(\mathbf{x})$ is the value of the scalar field at the point $\mathbf{x}$. The field generated by (the equation of) a circle is reconstructed as shown in Figure 4.7(a) using this technique on a set of eight surface points and eight offset points having field values of 0 and -1 respectively.

In the Hermite formulation (see Macêdo *et al.* (2009) Pan *et al.* (2009)) the gradient vector $\left(\frac{\partial f(\mathbf{x}')}{\partial \mathbf{x}[0]}, \cdots, \frac{\partial f(\mathbf{x}')}{\partial \mathbf{x}[d]} \right)$ of the field is additionally specified at a set points $\mathbf{x}'$ that may coincide with $\mathbf{x}$. The Hermite variational implicit surface in $\mathbb{R}^{d+1}$ is subsequently obtained by solving the system of Equations

12 In contrast, because a bivariate polynomial of order n has $m = \frac{(n+1)(n+2)}{2}$ coefficients it can only exactly interpolate m vertices. (If the number of vertices is greater or less than m the polynomial can only approximate it.) Thin plate splines however can interpolate any number of vertices and moreover have been shown to be the smoothest possible interpolant. It is also of interest to note that, in height field problems described above, the null space of a thin plate spline is a bivariate polynomial. One can therefore consider thin-plate splines as a superset of the polynomial fitting technique.

$$f(\mathbf{x}) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i) + \sum_{j=1}^{m'} \sum_{r=0}^{d} \alpha'_{jr} \frac{\partial K(\mathbf{x}, \mathbf{x}_j)}{\partial \mathbf{x}[r]} + \sum_{j=1}^{n} \mu_j \mathbf{p}_j(\mathbf{x}) = 0 \tag{4.51}$$

$$\begin{aligned} \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}[0]} &= \sum_{i=1}^{m} \alpha_i \frac{\partial K(\mathbf{x}, \mathbf{x}_i)}{\partial \mathbf{x}[0]} + \sum_{j=1}^{m'} \sum_{r=0}^{d} \alpha'_{jr} \frac{\partial^2 K(\mathbf{x}, \mathbf{x}_j)}{\partial \mathbf{x}[0]\, \partial \mathbf{x}[r]} + \sum_{j=1}^{n} \mu_j \frac{\partial \mathbf{p}_j(\mathbf{x})}{\partial \mathbf{x}[0]} \\ &\dots \\ \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}[d]} &= \sum_{i=1}^{m} \alpha_i \frac{\partial K(\mathbf{x}, \mathbf{x}_i)}{\partial \mathbf{x}[d]} + \sum_{j=1}^{m'} \sum_{r=1}^{d} \alpha'_{jr} \frac{\partial^2 K(\mathbf{x}, \mathbf{x}_j)}{\partial \mathbf{x}[d]\, \partial \mathbf{x}[r]} + \sum_{j=1}^{n} \mu_j \frac{\partial \mathbf{p}_j(\mathbf{x})}{\partial \mathbf{x}[d]} \end{aligned} \tag{4.52}$$

for m surface surface constraints $\mathbf{x}$ and the $(d+1)$ components of the m' gradient vectors $\frac{\partial f(\mathbf{x}')}{\partial \mathbf{x}}$ at $\mathbf{x}'$ respectively. However, as the equation system has $m + m'(d+1) + n$ unknowns, n additional equations must be supplied by imposing the now familiar orthogonality conditions

$$\sum_{i=1}^{m} \alpha_i \mathbf{p}_j(\mathbf{x}_i) + \sum_{j=1}^{m'} \sum_{r=0}^{d} \alpha'_{jr} \frac{\partial \mathbf{p}_j(\mathbf{x})}{\partial \mathbf{x}[r]} = 0 \tag{4.53}$$

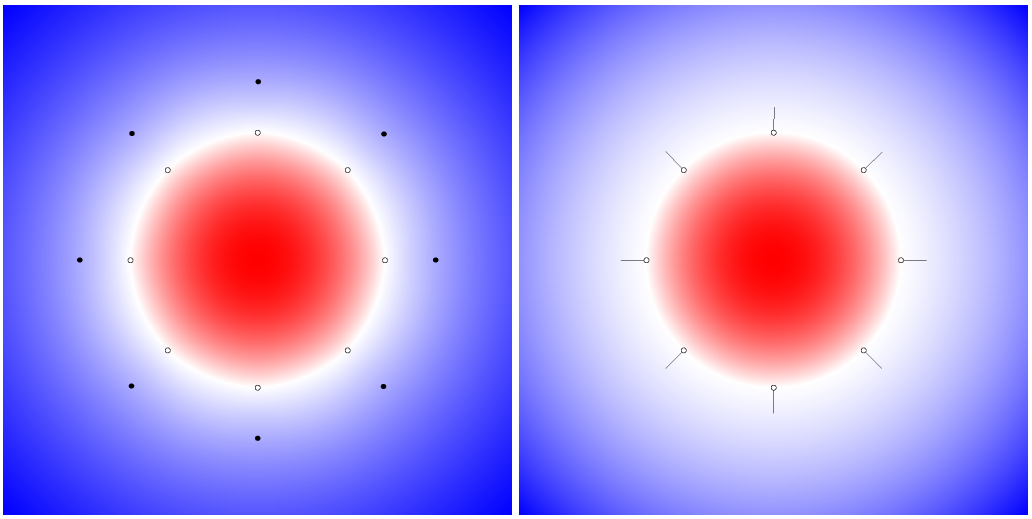
for each of the n null space polynomial terms.

A reconstruction of the field generated by (the equation of) a circle is shown in Figure 4.7(b) by solving the above equation system using the gradient vectors and field values at eight surface points.

As shown in Figure 4.8(a) and (b) both techniques can be used in order to reconstruct scalar fields in higher dimensions. Typically, the objective is to obtain an implicit form for a polyhedral mesh by treating its vertices as surface points and vertex normals as the field gradients. In both cases, the smooth surface approximated by the mesh is the zero level set of the implicit form. Although the variational implicit surface method involves storing and inverting a smaller matrix, it performs poorly when two parts of a surface are very close Macêdo *et al.* (2009). This is due to the ad-hoc offsets in the normal direction.

4.2.4 *Landmark-based deformation*

In Section 4.2.1 thin-plate splines were used to interpolate orthogonal displacements from a horizontal plane. However, the problems to which the method has been applied thus far should not suggest that it is limited to displacements in any particular direction. In general, thin-plate splines can also interpolate the arbitrary displacement $\mathbf{d}_i$ of points $\mathbf{p}_i$ in the horizontal plane. The components $d_i[0]$ and $d_i[1]$ of the displacements are treated like the orthogonal offsets in Section 4.2.1, but with one major difference: two thin-plate splines $f_1(x[0], x[1])$ and $f_2(x[0], x[1])$ are required in order to interpolate the displacement of a plate along either axis, $x[0]$ and $x[1]$. And by extension, in three dimensions, a third thin-plate spline $f_3(x[0], x[1])$ is required in order to interpolate the



(a) (b)

Figure 4.7 Reconstructing the scalar field generated by (the equation of) a circle, given eight surface points having zero field values (hollow dots), using the (a) variational implicit surface technique. Offset points (solid dots) are assigned field values of -1. (b) Hermite formulation, showing the gradient vector and field values at the surface points.

displacement of landmarks along the third axis etc. The lack of a physical analogue for a three dimensional thin plate spline (and higher) is not an argument against its validity.

Returning to the initial problem, and assuming that the set of landmarks are arbitrarily displaced to $(x_1[0]', x_1[1]') \ldots (x_n[0]', x_n[1]')$ in $\mathbb{R}^2$ from their original locations $(x_1[0], x_1[1]) \ldots (x_n[0], x_n[1])$, the thin-plate splines $f_1(x[0], x[1])$ and $f_2(x[0], x[1])$ interpolate the displacements and constitute a multivariate deformation function

$$f : (x[0], x[1]) \rightarrow (x[0]', x[1]') = (x[0] + f_1(x[0], x[1]), x[1] + f_2(x[0], x[1]))$$

However, if, as is often the case, the null space of the thin-plate splines are of order $p \geqslant 1$, i.e. they comprise linear polynomial terms of degree one, $x[0]$ and $x[1]$ can be absorbed into $f_1(x[0], x[1])$ and $f_2(x[0], x[1])$ so that the deformation function becomes

$$(x[0]', x[1]') = (f_1(x[0], x[1]), f_2(x[0], x[1])) \tag{4.54}$$

Therefore the thin plate splines can be used to directly interpolate the target positions of the landmarks, instead of interpolating the displacements they undergo. For example, of the four landmarks $\mathbf{a} = (10, 0)$, $\mathbf{b} = (0, 10)$, $\mathbf{c} = (-10, 0)$ and $\mathbf{d} = (0, -10)$, sampled along the circumference of a circle as shown in Figure 4.9(inset), only the landmarks $\mathbf{a}$ and $\mathbf{b}$ are moved to new positions $(10, 30)$ and $(-30, 20)$ respectively, while $\mathbf{c}$ and $\mathbf{d}$ are kept stationary. Two thin-plate splines are used to interpolate the target positions of

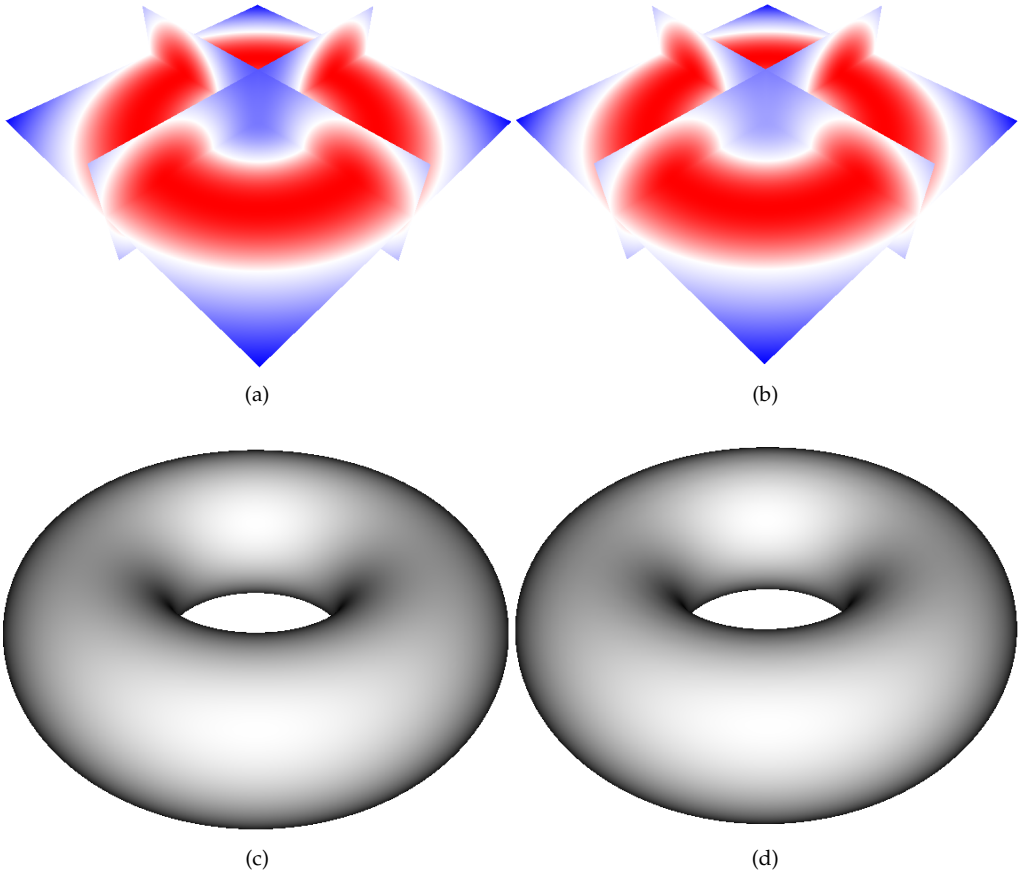


Figure 4.8 Cutting plane through a reconstruction of the scalar field generated by the surface points and normals of a polyhedral torus using the (a) variational implicit surface, and (b) Hermite method. The smooth surfaces (c) and (d), obtained by ray casting, are the zero level sets of the respective scalar fields.

the landmarks in each direction $x[0]$ and $x[1]$, as shown in Section 4.2.1. The resulting deformation is shown in Figure 4.9(main). This method of computing deformations was developed by Bookstein (1989).

Bringing the Hermite information to bear on the deformation is straightforward, and again follows from Section 4.2.1. In fact, thin plate splines can be used to compute deformations that arise from the changes in the gradients and normals at the landmarks, even when the landmarks remain stationary. For example the four stationary landmarks in Figure 4.10 each undergo 45 degree rotations of their normal and gradient vectors. Nevertheless, the parameter m (see Equation 4.9) determines the global properties of the deformation function, as shown in Figure 4.11(a) and (b). Although both deformations satisfy or reproduce the gradient and tangent constraints at each landmark, (a comparison of) the severity of the distortion of the embedding grids suggests that $m = 1.75$ is a better choice for the thin-plate spline parameter. Taken to its logical conclusion, the

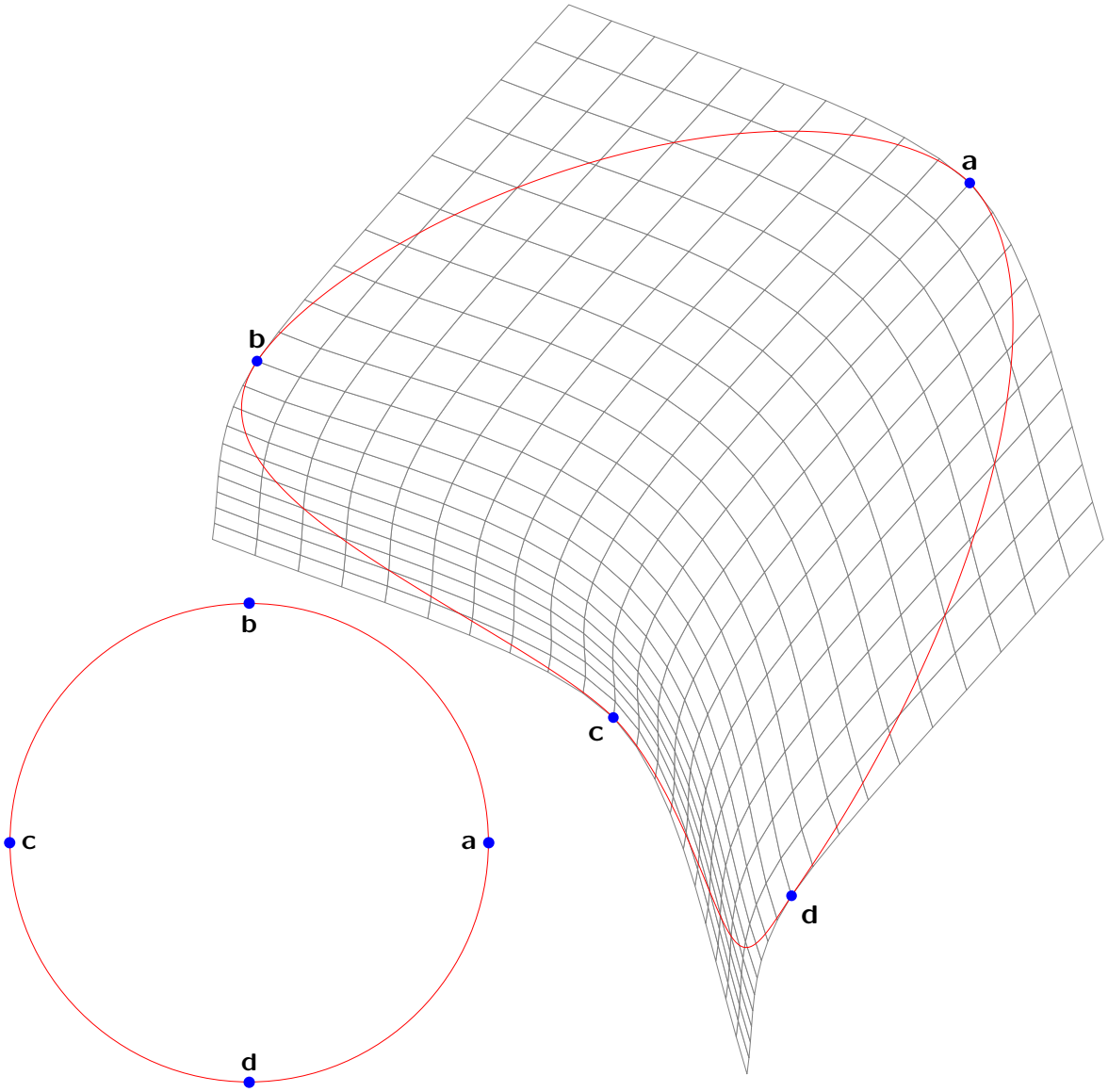


Figure 4.9 Deformation of the circle shown in inset and its embedding grid, due to the relocation of the landmarks $\mathbf{a}$ and $\mathbf{b}$.

ideal parameter would satisfy all the constraints at the landmarks while generating the least-distorted embedding grid. Little (1995) underscores the need for further study on the choice of this parameter, albeit in the kriging framework.

The method is equally applicable to deformations in three dimensions; the Hermite data often consists of two surface tangents and a normal vector, and a third thin-plate spline is required in order to specify the third component of the deformation. For example, the shape shown in Figure 4.12(a), a sphere of unit radius, has 6 landmarks, two of which are placed on its poles while the remaining are equally distributed along its "equator", and each has two surface tangents and a normal vector specified (initial vectors not shown in Figure). Figure 4.12(b) shows the deformation experienced by

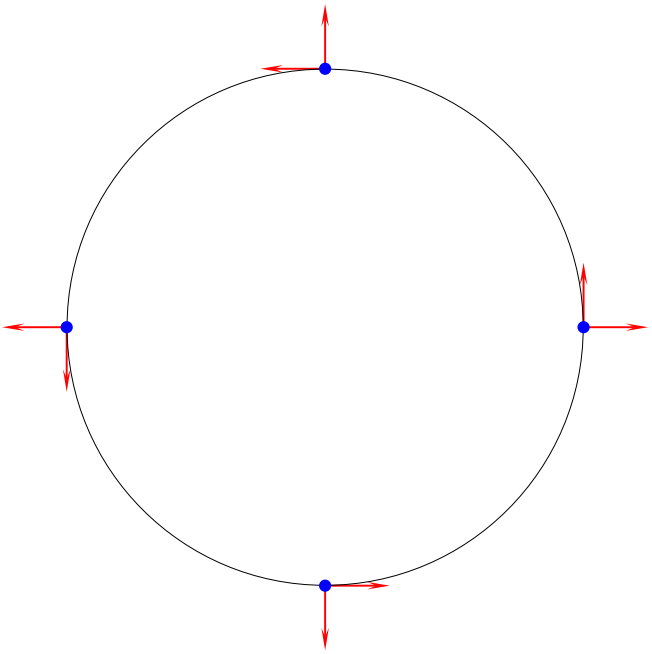


Figure 4.10 Gradient and normals (red) at four landmarks, along the circumference of a circular structure.

the sphere as a result of downward displacement of 0.5 by the landmark at the upper pole, while all other landmarks remain stationary. Figure 4.12(c) shows the deformation experienced by the sphere as a result of a -45 degree rotation of the triad of vectors around the surface tangent along the equator (the other surface tangent points toward the poles). Figure 4.12(d) shows the deformation experienced by the sphere of an additional 45 degree rotation of just one of the triad of vectors (drawn in blue) around the surface normal. This additional twist introduces a ridge on the surface of the sphere.

4.2.5 *Semilandmark-based deformation*

Another question posed by Little (1995) concerns the optimal number and the placement of landmarks. This is especially true of featureless structures commonly characterized by gently curving outlines and surfaces. The naive solution to this problem is to distribute a set of landmarks on the form according to relatively trivial criteria such as uniformity Zelditch *et al.* (2004)(page 396-397). Unfortunately the registration of objects using semilandmarks distributed in this manner can introduce unexpected kinks or artificial features in the deformed object and its embedding grid. In contrast, the optimal placement of semilandmarks will produce no extraneous features or contortions in either object. Furthermore, because the distortion of the structure and its embedding grid are proportional to the bending energy of the thin-plate spline deformers, the optimal arrangement of semilandmarks also minimizes the bending energy of the thin-plate

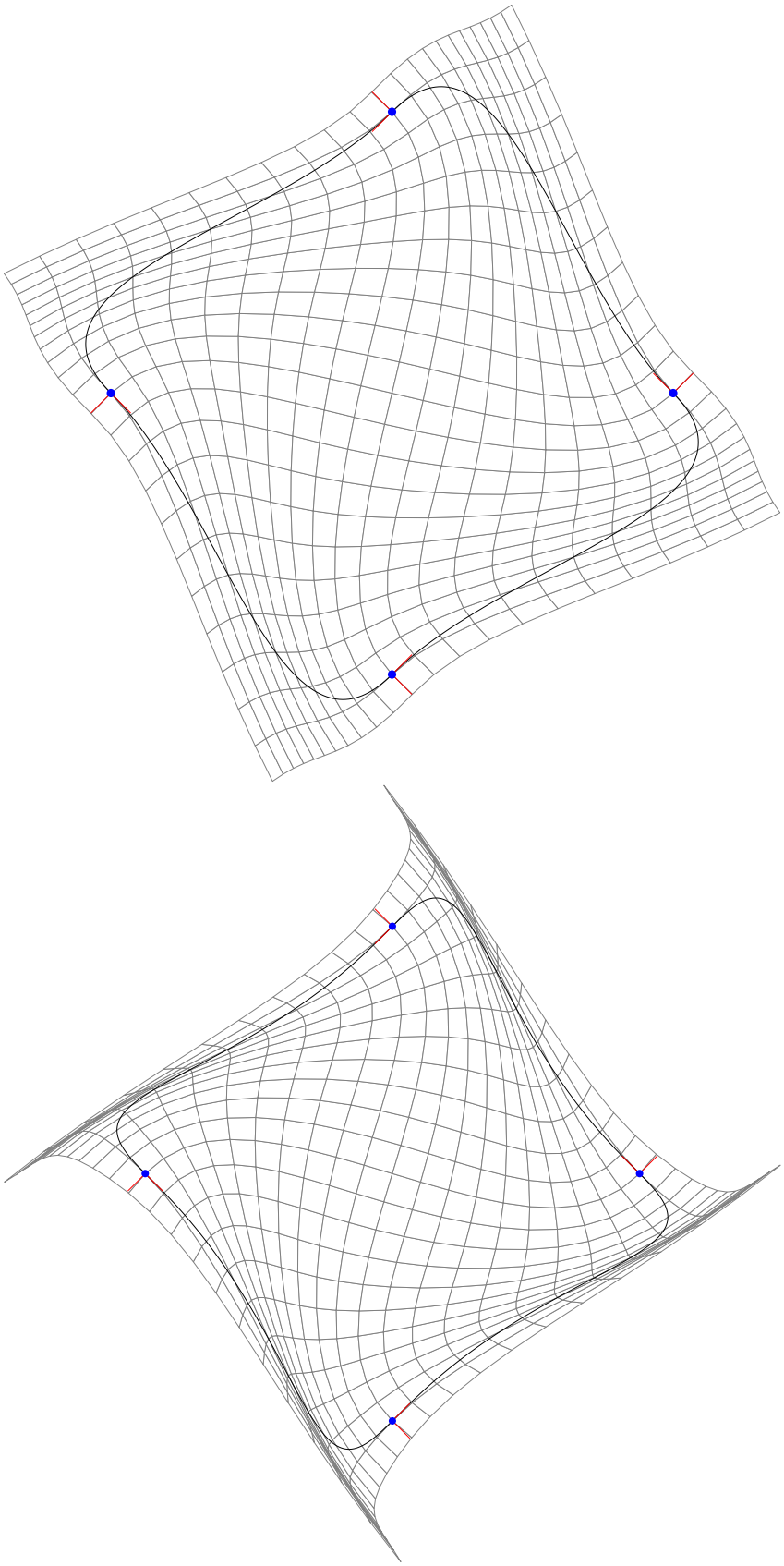


Figure 4.11 Deformation produced by the 45 degree rotation of gradients and normals at the four stationary landmarks in Figure 4.10, using the parameter values (a) above: $m = 1.75$ (b) below: $m = 2$, where $d = 2$ (see Equation 4.9).

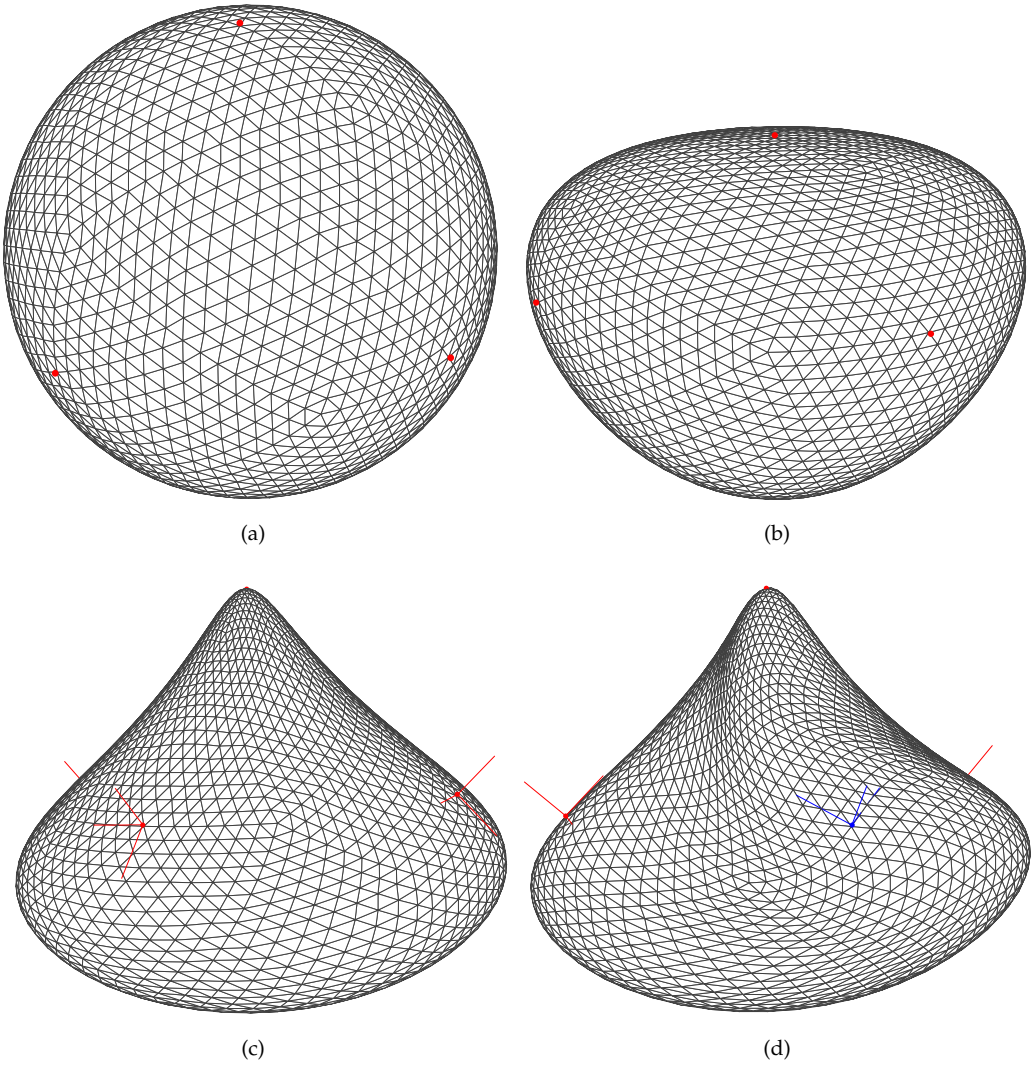


Figure 4.12 Deformations based on three-dimensional thin-plate splines. (a) undeformed, sphere of unit radius, having six landmarks. (b) deformation of sphere based on downward displacement of landmark at the upper pole by 0.5 units. (c) deformation due to -45 degree rotation of triad of vectors at landmarks around the "equator" of the sphere. (d) deformation due to additional twist of one of the triad of vectors (shown in blue).

spline deformers. The kernel of this idea first appeared in Bookstein (1989) but was properly developed for problems in two dimensions and three dimensions in the later publications Bookstein (1997), Gunz *et al.* (2005). In either case the technique involves allowing semilandmarks to slide on the form until the bending energy of the deformer is minimized.

Bookstein appears to drop the middle term $(\mathbf{K}\,\mathbf{L}_m^{-1})^T$ in Equation 4.25 and presents the bending energy of a thin-plate spline as $\mathbf{F}^T\,\mathbf{L}_m^{-1}\,\mathbf{F}$, so that for problems in two dimensions the total bending energy of the deformation, to be minimized, is given as

$$E \;=\; \mathbf{x}[0]^T\,\mathbf{L}_m^{-1}\,\mathbf{x}[0] \;+\; \mathbf{x}[1]^T\,\mathbf{L}_m^{-1}\,\mathbf{x}[1] \tag{4.55}$$

Minimization proceeds iteratively by allowing each semilandmark to slide along the direction of its tangent vectors $\mathbf{u} = (\,u[0]\,,\,u[1]\,)$ to a new position $\mathbf{x}' = \mathbf{x} + t\,\mathbf{u}$, where t is a scalar parameter that determines the direction and extent of sliding. Semilandmarks must however be projected to the curve because the sliding step forces them off the curve. These sliding and projection steps are repeated until the bending energy converges.

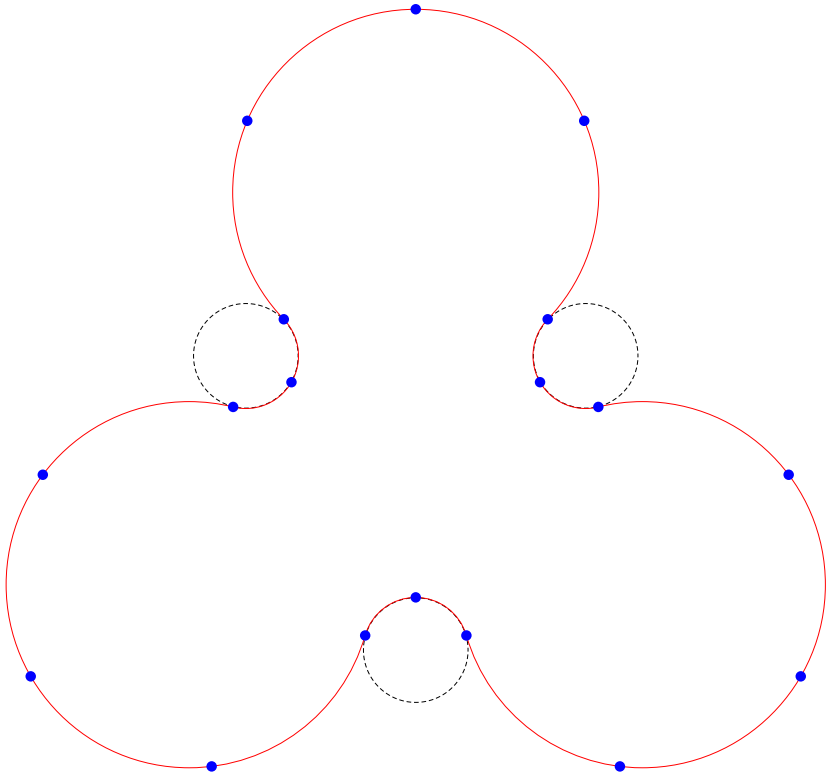
Unfortunately, this method fails if the source object (on which the sliding occurs) curves rapidly, and more sophisticated procedures are required. For example the method fails badly for the pair of (essentially featureless) curves shown in Figure 4.13, because the semilandmarks on the source object (circle) slide so far from it that the projection step effectively rearranges the landmarks. In fact, weighting the sliding parameters t by a factor (less than one, e.g. 0.25) does not prevent this excessive sliding, as illustrated in Figure 4.14. In this case the optimal arrangement of landmarks can be found by the method of Lagrange multipliers. First a cost function

$$J \;=\; E \;+\; \sum_{i=0}^{18} \lambda_i\,(\,x_i[0]^2 \;+\; x_i[1]^2 - R^2\,) \tag{4.56}$$

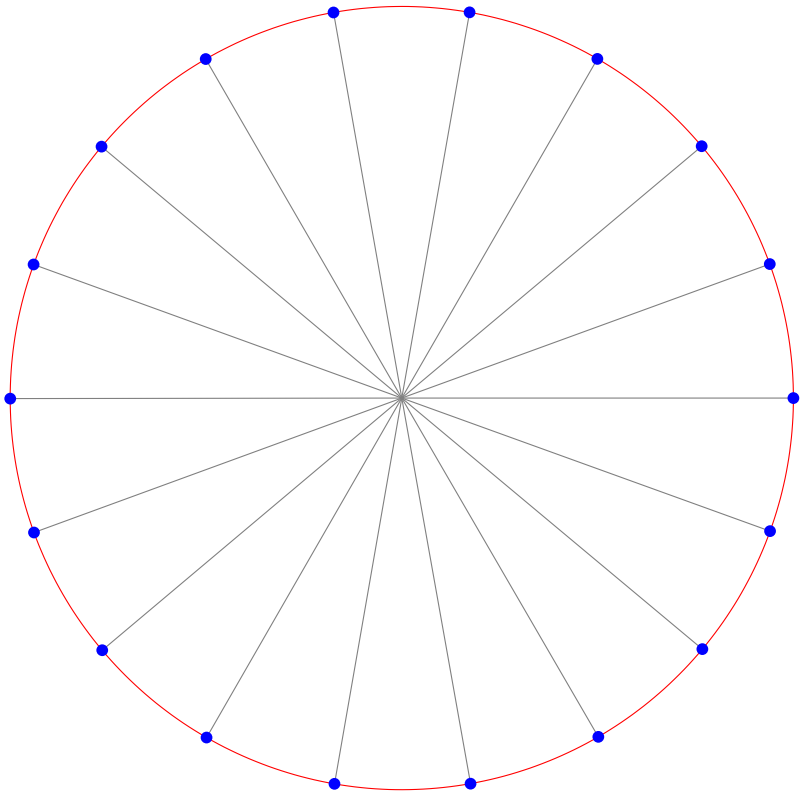
is constructed, where the first term E is from Equation 4.55 and the second term constrains each one of the 18 semilandmarks $\mathbf{x}_i = (\,x_i[0]\,,\,x_i[1]\,)$ to the source shape, a circle having radius R. (The λ_i's are the Lagrange multipliers.) Taking the derivatives of J yields the following system of 54 (18×3) equations

$$\begin{aligned}
\frac{\partial J}{\partial x_i[0]} &= \frac{\partial E}{\partial x_i[0]} \;+\; 2\,\lambda_i\,x_i[0] \\
\frac{\partial J}{\partial x_i[1]} &= \frac{\partial E}{\partial x_i[1]} \;+\; 2\,\lambda_i\,x_i[1] \\
\frac{\partial J}{\partial \lambda_i} &= x_i[0]^2 \;+\; x_i[1]^2 \;-\; R^2
\end{aligned}$$

As in Section 4.2.2 this system of equations is solved for all $\mathbf{x}_i$'s using a multidimensional root solver. The resulting distribution of semilandmarks on the source shape (circle), shown in Figure 4.15(b), is no longer uniform but is now clustered into three symmetric groups, and reassuringly mirrors the distribution of semilandmarks on the target shape (Figure 4.13). Furthermore, registration using the redistributed semilandmarks is accompanied by a 22.5% reduction in bending energy, as evidenced by the smoother embedding grid shown in Figure 4.15(a).

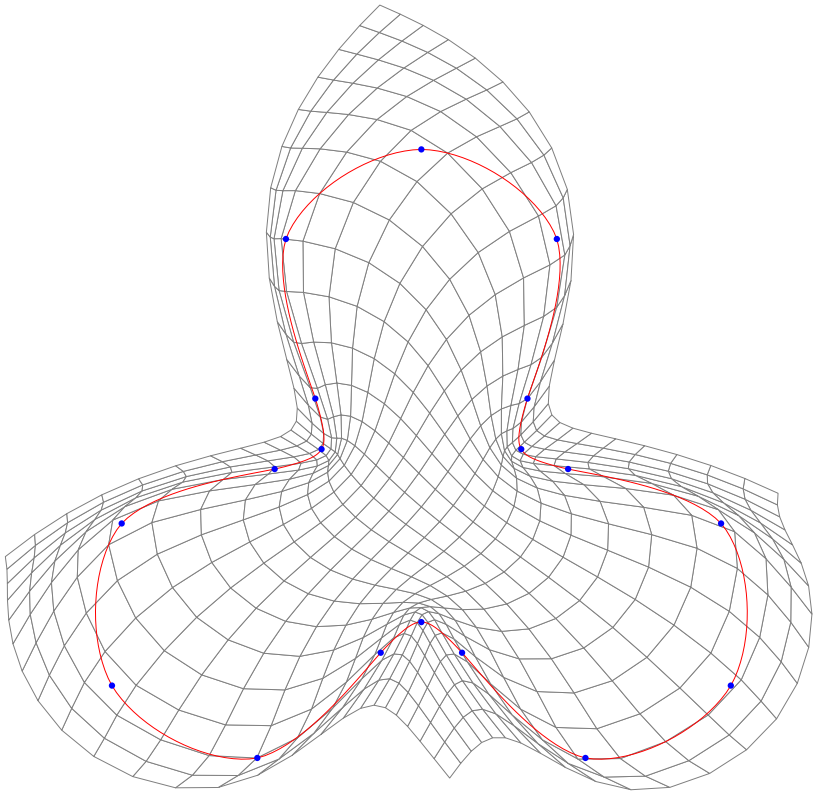


(a)

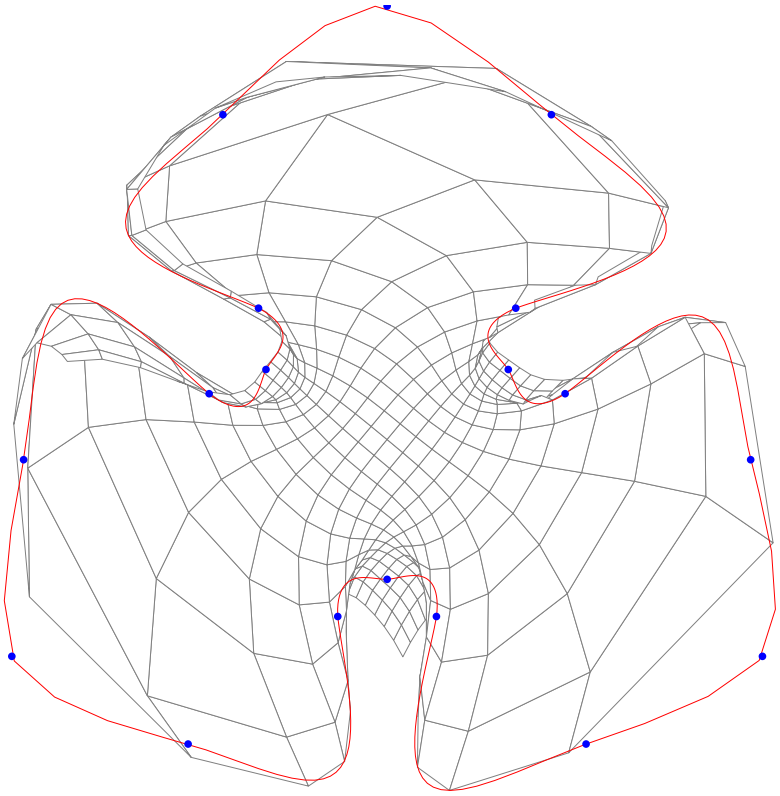


(b)

Figure 4.13 Source and target objects, prior to the sliding semilandmarks. (a) target object, containing stationary landmarks. (b) source object, containing arbitrarily distributed landmarks, allowed to slide.

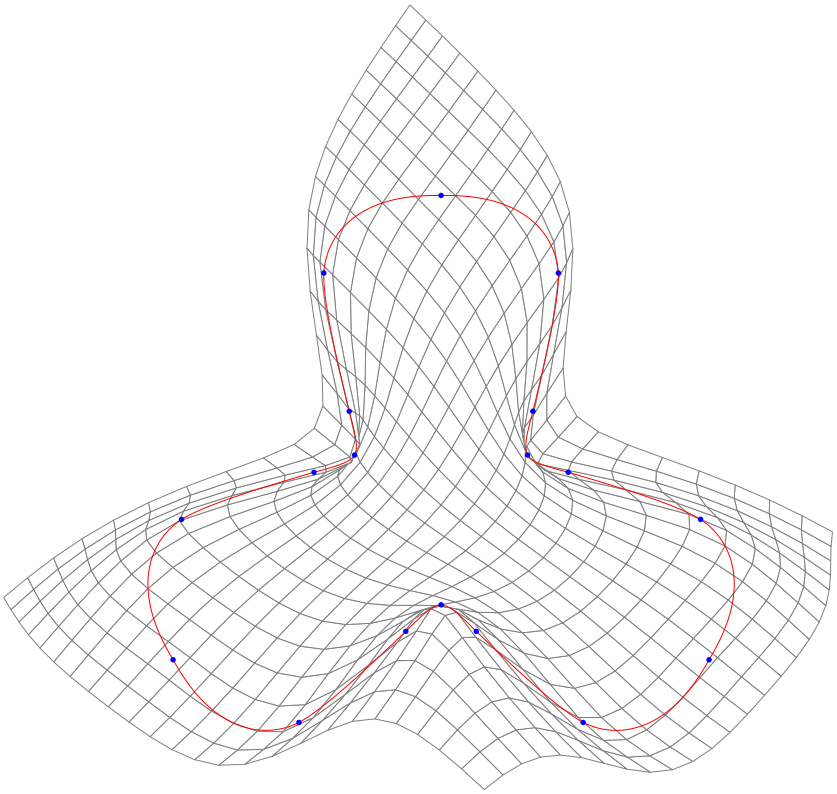


(a)

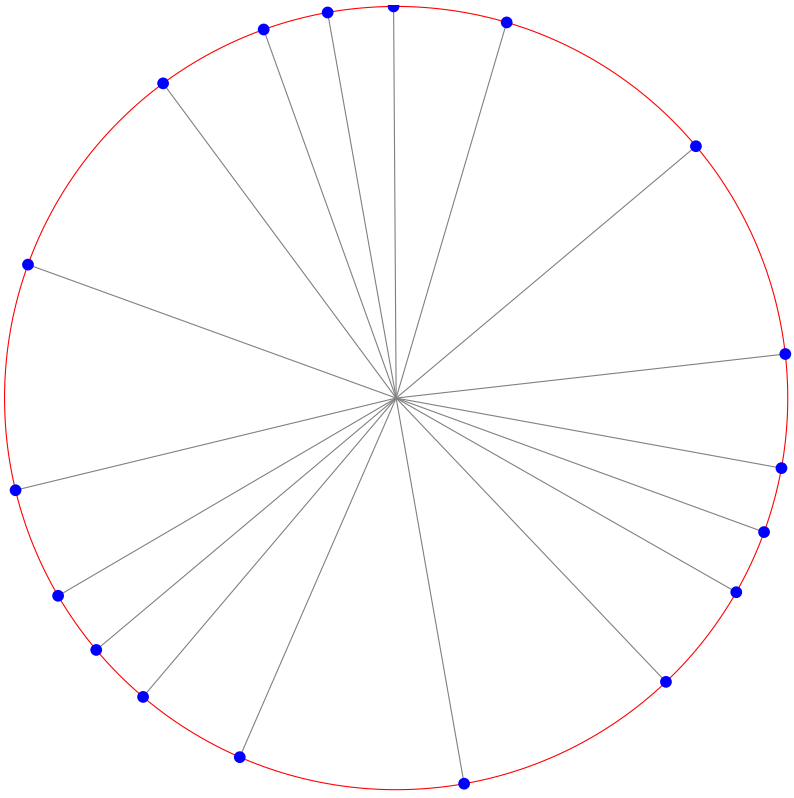


(b)

Figure 4.14 Registration of source object, Figure 4.13(b), with target object, Figure 4.13(a); (a) based on initial position of semilandmarks on source object i.e. prior to sliding. (b) poor registration, after the fifth iteration of the semilandmark sliding algorithm described by Bookstein (1997).



(a)



(b)

Figure 4.15 (a) Registration of source object, with target Figure 4.13(a), after redistributing semilandmarks, as shown in Figure b. (b) redistribution of semilandmarks shown in Figure 4.13(b), based on multidimensional root solving.

4.3 SUMMARY

Thin-plate splines are an important ingredient in many computer animation algorithms, as the catalogue of previous works (in Chapter 2) has already shown. The present work is no exception, as it makes repeated use of the algorithm, in Chapters 6 and 7, in order to: compute the surface properties of model at a set of landmarked points, optimize the placement of landmarks for purposes of registration, fit a generic skull to a head model and build an implicit surface representation of the SMAS from a set of point constraints.

Prepare ye the way of the Lord, make his paths straight — Mark 1:3

5

BOUNDARY-VALUE STRAIGHTEST GEODESICS

Straightest geodesics on polyhedral surfaces were introduced by Polthier and Schmies (1998) (see also Polthier and Schmies (2006)) as discrete curves having the property of being straightest as opposed to being locally shortest – both properties are equivalent for geodesics on smooth surfaces. Straightest geodesics are generalizations of straight lines on discrete manifolds and allow concepts such as the convex hull to be extended to non-Euclidian spaces. Although the definitions of straightest geodesics and shortest geodesics are identical on faces and across edges, they differ at vertices in two important ways. First, whereas straightest geodesics can be extended through spherical vertices, shortest geodesics cannot because they can always be shortened by shifting the path away from the corner vertex, as shown in Figure 5.1 (a). Note that all three paths shown in Figure 5.1 (a) are straightest geodesics, but only the red paths are shortest geodesics. Second, there exists a family of paths or shortest geodesics through a hyperbolic vertex with the same inbound direction, one and only one of which extends the inbound direction as a straightest geodesic. This is because there is a shadow region in the neighborhood of a hyperbolic vertex where two points cannot be joined by a straightest geodesic (see Lemma 2.3, 2.4 and Figure 6 of Polthier and Schmies (1998)). In order to better illustrate this, consider the path svt in Figure 5.1 (b) that passes through a hyperbolic vertex v where s and t are points on the ring of faces that share v. The sum of the left (counterclockwise) and right (clockwise) angles θ_l and θ_r of the path satisfies the inequality $\theta_l + \theta_r > 2\pi$, which in turn implies three possible combinations of values of both angles: *(i)* $\theta_l < \pi$ and $\theta_r > \pi$, *(ii)* $\theta_l > \pi$ and $\theta_r < \pi$ and *(iii)* $\theta_l > \pi$ and $\theta_r > \pi$. In the first two cases the path can be straightened by connecting s and t by a line constructed across the ring of faces flattened around the smaller angle. Unfortunately, in the third case, as both angles are greater than π, a straightest geodesic cannot be constructed because there is no preferred direction in which to flatten the faces. However, the initial value problem is uniquely solvable and involves extending a path, from a specified point, in a given direction along a surface.

This chapter examines the boundary value problem in which a straightest geodesic connecting two given boundary points is desired, and presents three heuristically-motivated algorithms for tracing a path or sequence of polygons that embed a straightest geodesic on a polyhedral surface, in the direction of the straight line between both input points. The motivating application for the development of boundary-value straightest geodesics is in the construction of facial muscle fibres (see Chapter 7). The muscles of facial expression are thin, sheet-like bundles of oriented, contractile fibres, that span the subsurface (superficial musculoaponeurotic system– SMAS) of the human face. Because each muscle

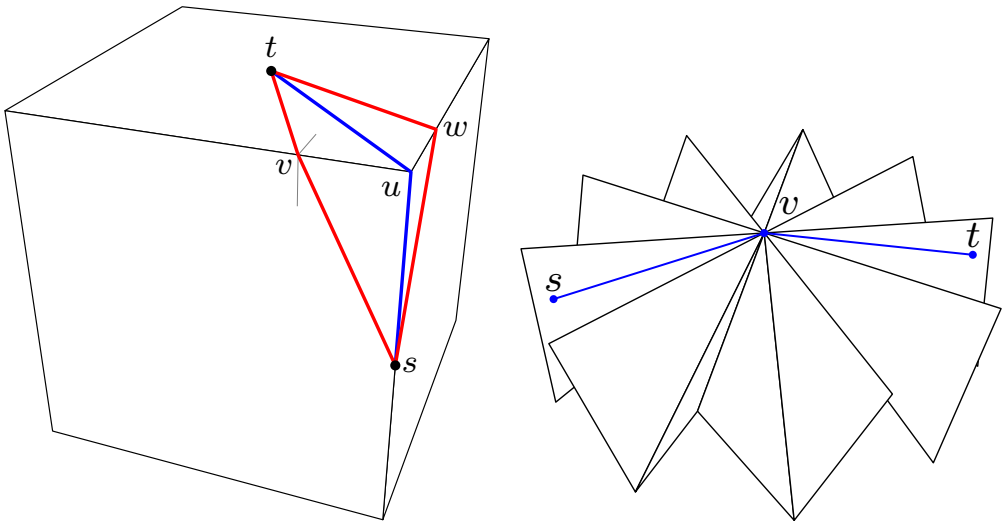


Figure 5.1 (a) Left (based on Figure 6 of Polthier and Schmies (1998)): locally-shortest geodesics (red) cannot be extended through a spherical vertex (b) right (reproduction of Figure 2 of Mitchell *et al.* (1987)): optimal path through a hyperbolic vertex.

fibre contracts towards a straight line connecting its origin (on the skull) and insertion (underneath the skin) it is natural to assume that muscle fibres are laid out as straight as possible and to model them accordingly – as straightest geodesics. In other words, the essential property of straightest geodesics best describes the geometry of muscle fibres.

Although shortest geodesics also possess the property of straightness, Figure 5.1 (a), existing methods of computing shortest geodesics, e.g. the MMP Surazhsky *et al.* (2005) and FMM Kimmel and Sethian (1998), involve propagating waves over large portions of a surface in order to generate a single path. Such surface exploration is costly, wasteful and impractical for the multiple single-source single-destination path computations required by the problem at hand.

Therefore the method presented in this chapter focuses on the less demanding problem of iteratively straightening an initial estimate of the straightest geodesic connecting two given points. This is done by generating sequences of polygonal faces that embed straighter paths connecting the two end points. This approach avoids the expense of propagating distance waves on polyhedral surfaces. Although, in most situations, a path generated using this technique is expected to coincide with the shortest paths, the technique cannot be described as a shortest geodesic method as no explicit attempt is made to minimize the distance between the two given points.

Definition 1 *A sequence or path of polygonal faces embeds a straightest geodesic connecting two points if the line segment joining both points intersects all the inner or shared edges of the path when unfolded, as shown in Figure 5.2.*

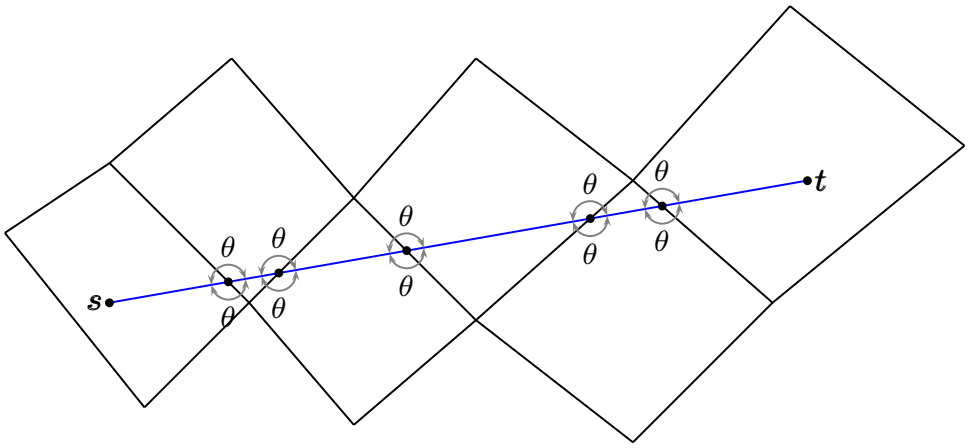


Figure 5.2 The straight line connecting two points s and t.

In other words, a path or sequence of polygonal faces embeds a straightest geodesic connecting two points if one line segment is sufficient to connect both points on a sequence of faces. Such a discrete curve is globally straight because its left and right angles θ are equal at each point including at the inner edges e of the path that embeds it.

Definition 2 *The total signed curvature (TSC) is the sum of all of the angles of the segments of a discrete curve, as shown in Figure 5.3.*

A straightest geodesic has zero TSC.

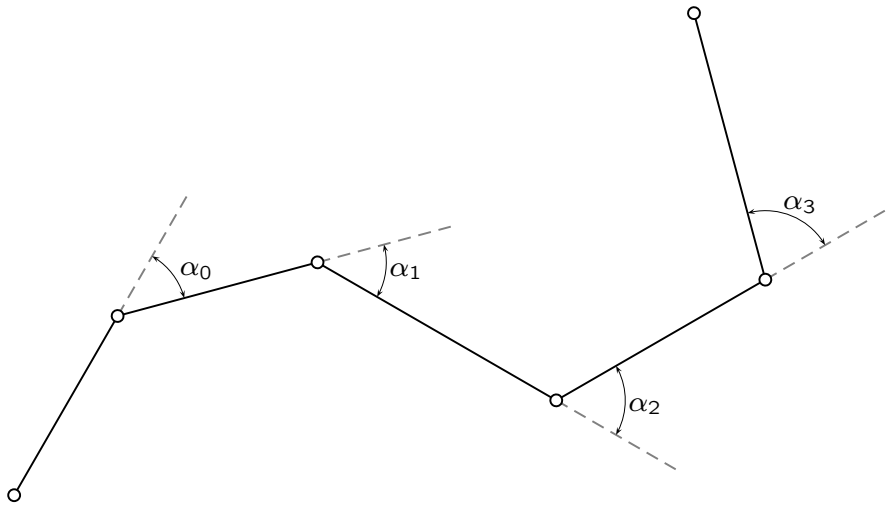


Figure 5.3 Discrete curve consisting of five segments. Total signed curvature is given as $\sum_{i=0}^{3} \alpha_i$.

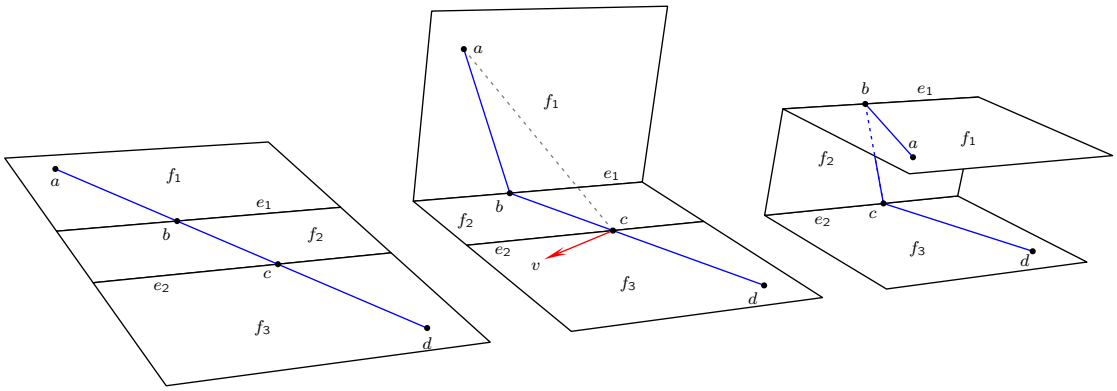


Figure 5.4 Folding a portion of a thick slab edge embedding the straightest geodesic abcd (a) left: unfolded slab (b) middle: partially folded along the edge e_1. (c) right: fully folded slab.

5.1 PREVIOUS WORK: REVIEW AND APPLICATIONS

Much of the previous work undertaken on the subject of straightest geodesics is primarily concerned with local straightest geodesics and the initial value problem of how to advance a straightest geodesic from a given point in a specified direction. For example Polthier and Schmies (1998) use straightest geodesics to perform the parallel translation of tangential vectors along arbitrary curves and generalize the Runge-Kutta method for tracing particles on a polyhedral surface by integrating a given vector field. Polthier and Schmies (1999) later studied the evolution of point source waves on polyhedral surfaces in which the front of each wave is updated by advancing straightest geodesics in equally distributed directions away from its source point, thus creating topological circles that move by a constant distance along the geodesic normal to the front. Lee *et al.* (2001) use this geodesic flow technique in order to model the spread of fire over polygonized objects. Lee *et al.* (2005) and Lee and Lee (2007) use local straightest geodesics in order to improve upon Floater's method for finding the non-negative constraints that produce shape preserving parameterizations for static meshes. Lee *et al.* (2006) however propose a method for computing the straightest geodesic from a source to any destination point on a surface patch of genus 0 by intersecting the patch with an oriented plane passing through source and destination points.

Unfortunately, in general, the intersection of a cutting plane with a polygonal mesh cannot produce a straightest geodesic. To illustrate this, consider the unfolded slab in Figure 5.4 (a) consisting of a sequence of polygonal faces f_1, f_2 and f_3 and embedding the line segment between the points a and d, and which intersects the edges e_1 and e_2 of the unfolded structure at the points b and c respectively such that the path abcd is a straightest geodesic. In order for the cutting plane method to produce this straightest geodesic a single plane would have to pass though the points a,b,c and d when the

faces of the slab are folded back to their original configurations shown in Figure 5.4(c) – implying that $abcd$ will be coplanar whether or not the embedding face is folded. This assertion can be easily tested by progressively restoring or rotating each face of the flattened slab to its original configuration and checking whether $abcd$ remains collinear after each step. Therefore the face f_1, including the embedded point a is rotated along the edge e_1 to its original configuration as shown Figure 5.4(b). Thereafter, the points b, c and d, because they remain collinear, are coplanar with the point a. However, in order for the points to remain coplanar subsequent rotations must occur along an axis vector v which is perpendicular to the plane abc as shown in Figure 5.4(b). Unfortunately, because the next rotation takes place round the edge e_2 the points cannot remain coplanar – unless the unfolded straightest geodesic is perpendicular to all of the edges it traverses and all of the edges of the object are parallel to each other, so that all axes vectors about which the faces are rotated are collinear. An uncapped uniform polygonal cylinder satisfies these conditions because its edges are parallel. Most polyhedral objects however do not. Therefore, in most cases a cutting plane cannot generate a straightest geodesic in the manner described by Lee *et al.* (2006). As such, the problem of boundary-value straightest geodesics is deemed as being unresolved. Nevertheless it is of interest to note that the path $abcd$ in Figure 5.4 remains a straightest geodesic irrespective of the state of unfolding of its faces and the number or degree of arbitrary, shape-preserving, rotations along the edges of the slab.

5.2 PATH TRACING BY VECTOR PROJECTION

Path tracing from a source to a destination point on a polyhedral surface is performed by repeatedly projecting a direction vector $\mathbf{d}$ formed by the straight line from a source or starting point to a target point, both on the surface of the polyhedron as shown in Figure 5.5. The direction vector is projected onto the polygon face f on which the source point lies. The projected vector $\mathbf{p} = \mathbf{n} \times (\mathbf{d} \times \mathbf{n}) = \mathbf{d} - (\mathbf{n}.\mathbf{d})\mathbf{n}$, where $\mathbf{n}$ is the normal of f, is fired outward from the source point and intersection-tested against all the edges of f. This ray intersects only one of the edges of f at the point q. Because the familiar two-dimensional line intersection formula cannot be reliably extended to three-dimensions, we treat the ray and all polygon edges e as skew lines and find a point q on an edge to be the intersection point if it is closest to the ray and lies between the start and end of the edge. In Figure 5.5 (main), the objective is to trace a path from the point r (on face f_1) to the point w (on face f_5). In this example, path tracing starts by projecting the direction vector $\mathbf{d}_r$ formed by the points r and w to the face f_1. The projection vector $\mathbf{p}_r$ is then fired from the point r and intersects the edge e_1 at the point s. Path tracing continues on the other polygon face f_2 that shares the edge e_1, and a direction vector $\mathbf{d}_s$ is formed by the points s and w. As before, a projection vector $\mathbf{p}_s$ is formed by projecting the direction vector to the current face and firing a ray from the point s in the direction $\mathbf{p}_s$. This ray intersects f_2 again at the point t along its edge e_2.

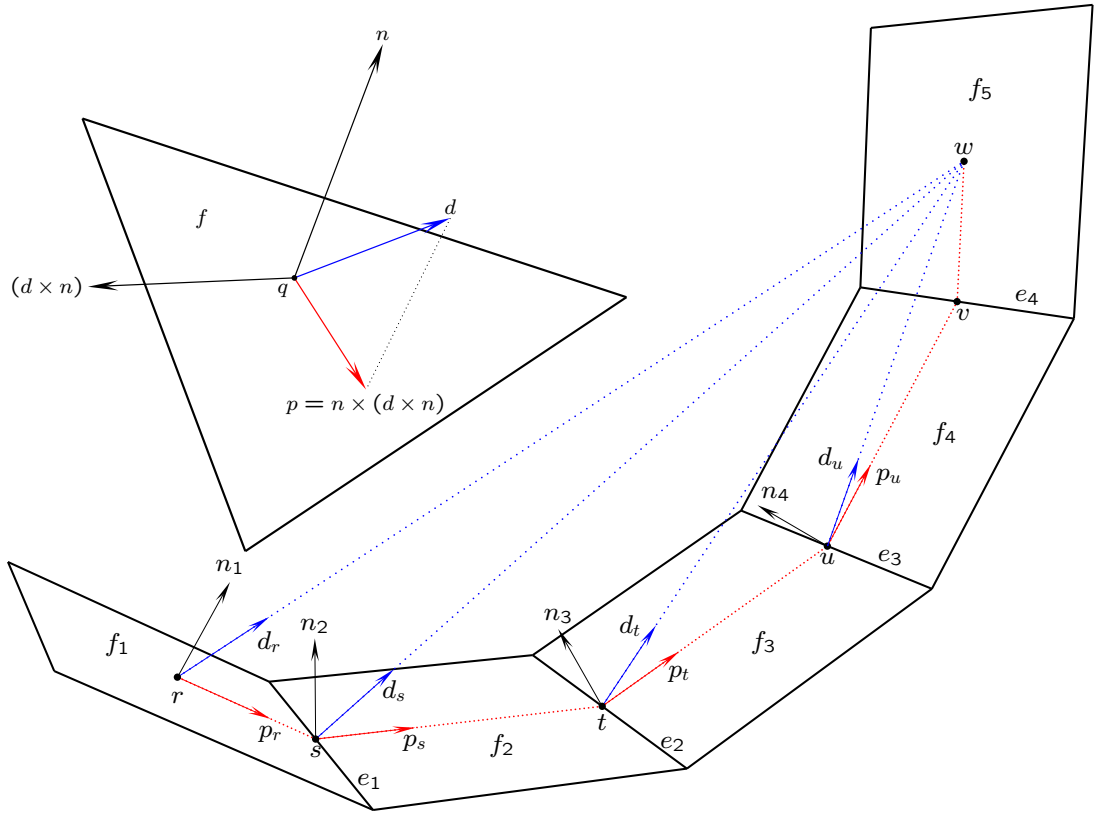


Figure 5.5 (a) Inset: projecting a direction vector $\mathbf{d}$ to a polygon face having normal $\mathbf{n}$. (b) main: path tracing by the repeated projection of the Euclidian distance vector to the surface of a polyhedral mesh.

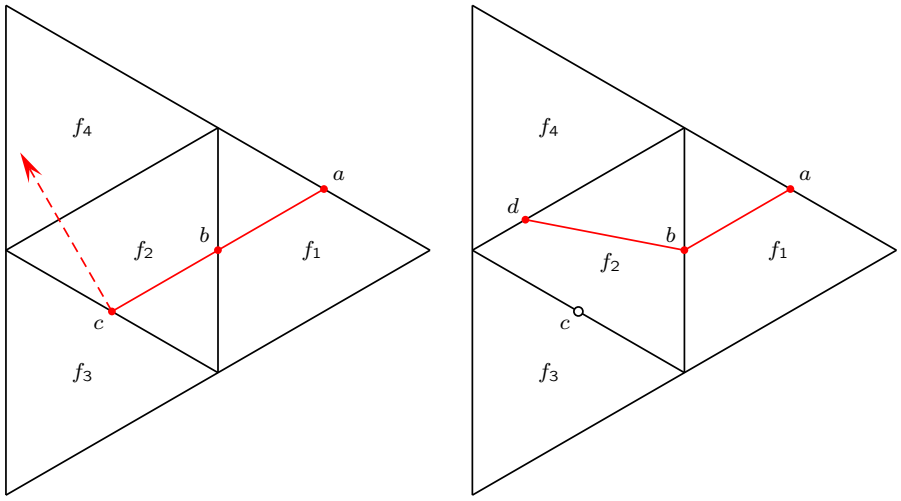
This sequence is repeated until the target face is encountered. This process is summarized in Algorithm 1. The output of this algorithm is a sequence of polygonal faces. In order to ensure that the path is simple, i.e. no face is repeated, each face encountered at each step is checked to see if it is already in the path. If it is, the path array is rolled back to the first instance of the repeated face, by deleting the section of the path between the instances of the repeated face. In the simplest case, backtracking is required if the projection vector points outside the current face in one step and back to it, instead of another face, in the next step, as shown in Figure 5.6(a). This diagram shows the discrete curve abc and triangle path $f_1f_2f_3$ are generated at an intermediate step of the vector projection algorithm. However at the next step the projected vector does not intersect the edges of the triangle f_3 but instead points back at triangle f_2. In order to ensure that the triangle f_2 is not encountered twice, f_3 is removed from that path, so that the discrete curve and triangle path traced are $f_1f_2f_4$ and abd respectively, as shown in Figure 5.6(b).

Because path tracing favours the most direct route to the target it is in fact a heuristic search for the straightest path. However, the sequence of polygonal faces traced by the algorithm often does not embed the straightest geodesic connecting the source and target

Algorithm 1: TRACEPATH

Input: source point (sp), source face (sf), target point (tp), target face (tf)
Output: point array (pa), face array (fa)
Data: current point (cp), current face (cf), current edge (ce) , direction vector (dv), projection vector (pv), projection ray (pr)

```
1  initialization: cp = sp, ce = 0, cf = sf
2  add cp to pa
3  while cf not equal to tf do
4      dv ←CREATEVECTOR(cp,tp)
5      pv ←PROJECT(dv,cf)
6      pr ←BUILDRAY(cp,pv)
7      foreach edge e in cp do
8          if e is not equal to cf then
9              q ←INTERSECT(e,pr)
10             if q is not equal to 0 then
11                 if cf is in fa then
12                     rollback fa n steps, to element before previous instance of cf
13                     rollback pa n steps
14                 end
15                 add q to pa
16                 add cf to fa
17                 ce ←e
18                 cf ←GETNEXTFACE(e,cf)
19                 exit foreach loop
20             end
21         end
22     end
23 end
24 add tp to pa
25 add tf to fa
```



(a) (b)

Figure 5.6 Backtracking.

points. Therefore the straightest discrete curve that can be embedded in the sequence of polygonal faces generated is extracted and straightened. This process, described in the next section, generates a new path or sequence of polygonal faces from which a straighter discrete curve is extracted, and is repeated until the straightest geodesic is found. Unfortunately, the path projection algorithm can fail on helical, open and severely undulating surfaces, or when the projection vector is normal to the surface.

5.3 PATH STRAIGHTENING BY BRIDGING

The path straightening algorithm proposed involves iteratively bridging the consecutive sections of a path traced by Algorithm 1. To demonstrate this idea, consider in Figure 5.7 (a) the polygonal curve $abcde$ consisting of four straight segments ab, bc, cd and de embedded in a sequence or path of polygonal faces (not shown). As a first step toward straightening this curve, the straight lines or bridges af, fg and ge are constructed to the mid points (f and g) of the intermediate segments (bc and cd). Together, the three segments form a straighter discrete curve $afge$ with a smaller total signed curvature (TSC) than the initial path. (Note that the first and last segments of the new curve are constructed from the start and end points a and e of the input path.) In the next iteration the discrete curve $afge$ is further straightened by bridging the start and end points of the curve and the mid point h of the intermediate segment fg, in order to form a new curve ahe, consisting of the two segments ah and he. But because ahe has no intermediate segments, a different bridging method is required in order to further straighten this (two-segment) curve. Such a curve is straightened using a two-step process that involves

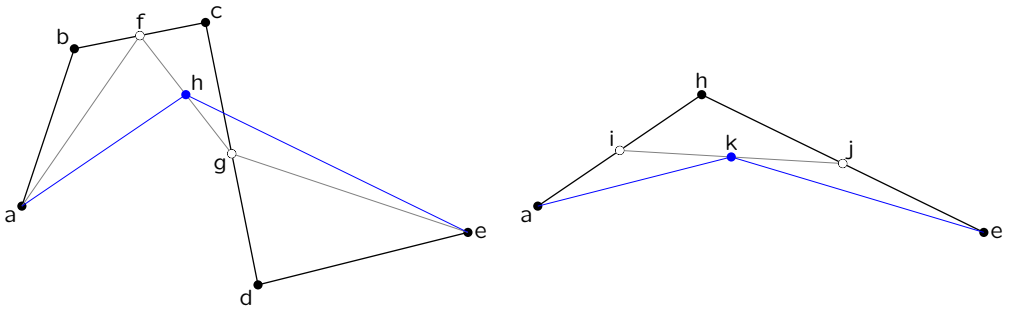


Figure 5.7 Progressive path straightening by bridging consecutive segments of two polygonal curves. (The sequences of flattened polygonal faces embedding the curves are not shown.) (a) left: bridging an N-segment curve, where $N > 2$. (b) right: bridging a 2-segment curve.

first bridging the mid points of both segments of the curve in order to form a new, single segment ij (see Figure 5.7(b)) and subsequently bridging the path ends (a and e) to the mid point k of the segment ij, in order to form a new, straighter path ake. The bridging method is summarized in Algorithm 2.

At this point, it is necessary to highlight that:

i. the above description outlines a single step of the iterative process of path straightening, by the method of bridging, albeit illustrated in two dimensions. In reality, the operation is performed in three dimensions on the surface of a discrete manifold of arbitrary connectivity. The algorithm is terminated when the sequence of faces that embed the (output) "bridging" discrete curve also embeds a straightest geodesic, or after a predetermined number of iterations, as described in the following section.

ii. as earlier highlighted, because the path tracing algorithm (see Section 5.2) that is used in bridging does not guarantee a straight path, the total number of segments embedded in the resulting polygonal path may not reduce by one after each iteration, as described in the simple case presented above and shown in Figure 5.7 (a). Nevertheless, the heuristic suggests that the polygonal path produced by bridging will embed a straighter discrete curve than the (section of the) original path that it bridges.

Now that the basic idea of bridging has been illustrated, what remains is to develop a method of extracting the required input sequence of line segments from a polygonal path.

5.3.1 *Number of straight lengths in a polygon path*

Assume a collection of the sets of discrete curves that can be embedded in a polygonal path, where each set in the collection groups curves with the same segment count, among

Algorithm 2: BRIDGING

Input: sourcePoint, sourceFace, targetPoint, targetFace, polygonPath

segments ← StraightLengths(polygonPath)
if *number of* segments *is equal to* 2 **then**
    $[p_1, f_1]$ ← MidPointAndFace(firstSegment)
    $[p_2, f_2]$ ← MidPointAndFace(secondSegment)
    subPath ← TracePath($[p_1, f_1]$, $[p_2, f_2]$)
    intermediateSegs ← StraightLengths(subPath)
else
    intermediateSegs ← IntermediateSegs(segments)
end
polygonPath ← [ ]
$[p', f']$ ← [sourcePoint, sourceFace]
foreach seg *in* intermediateSegs **do**
    $[p, f]$ ← MidPointAndFace(seg)
    subPath ← TracePath($[p, f]$, $[p', f']$)
    add subPath to polygonPath
    $[p', f']$ ← $[p, f]$
end
$[p', f']$ ← [targetPoint, targetFace]
subPath ← TracePath($[p, f]$, $[p', f']$)
add subPath to polygonPath

the collection the objective is to find the set containing curves with the smallest segment count, and within this set the discrete curve with the shortest length. Alternatively, such a polygon path, shown for example in Figure 5.9, can be considered as a tunnel in which there is no line of sight between the source and target points, s and t, at its two entrances. These points can be treated as the source and destination of a light beam. The goal is to install the smallest number of light beam relays required to forward photons as quickly as possible through the tunnel from the source to the destination. The locations of the relays are equivalent to the intermediate vertices of the shortest discrete curve with the minimum possible segment count embeddable in a polygon path.

In order for the system of relays to function as intended, there needs to be a line of sight between the source and the first relay, the target and the last relay, as well each pair of neighboring relays. To this end, it is essential to test whether there is a line of sight between each pair of inner corners along the tunnel and extend the line as far as possible in either direction until it touches the walls of the tunnel. A point on the wall of the tunnel is an inner corner vertex (ICV) if the angle, measured on the interior side of the tunnel, between the two outer edges that share it, is greater than π (see Figure 5.8). Therefore, ICVs protrude into the tunnel. In contrast, the remaining vertices protrude outward such that any line of sight passing through them cannot be extended further along the tunnel. The extended line of sight (ELOS), see Figure 5.8 and 5.9(a), between a pair of ICVs is locally-optimal because it is guaranteed to travel as far as possible within the tunnel without being obscured by either ICV. Lines of sight are also constructed from the source and target points but are only extended beyond the ICVs. Subsequently, all ELOSs are intersected against each other in order to create a visibility graph whose nodes are the source and target points, ICVs, and the intersections of the ELOSs, and whose edges are the ELOSs themselves, see Figure 5.9(a). Each point on an ELOS is independently connected by an edge to all other points on the ELOS.

The edges of the visibility graph are temporarily assigned unit weights and Dijkstra's algorithm Dijkstra (1959) is used in order to find all shortest paths from the source to the target nodes of the graph. Because the edges have unit weights the search returns the set of paths that have the minimum number of nodes between the source and target. The shortest by length of all these paths is the desired discrete curve, i.e. the shortest path having the smallest possible segment count. As shown in Figure 5.9(b), this curve can be obtained as a min-# simplification Chen *et al.* (2005) of the shortest path. This method however was not used because of the high cost of computing the shortest path. Crucially, the simplified curve has the same TSC as the shortest path, which is the shortest possible and therefore the straightest possible path connecting the source and target points. Therefore, the curve with the smallest-possible segment count is also a straightest possible curve.

However, because Dijkstra's algorithm has a memory and time complexity of $O(n^2)$ O'Rourke (1998) (page 297), a faster approximate search for the bridging path is developed for large graphs.

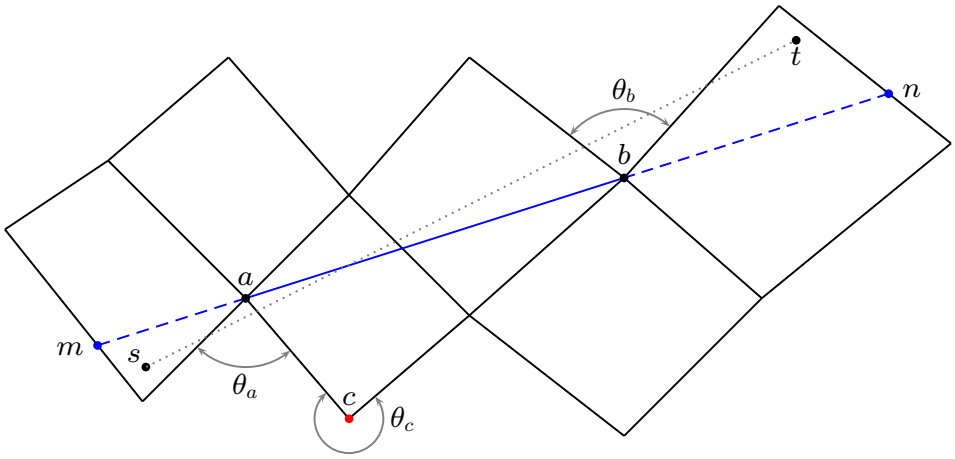


Figure 5.8 An unfolded path of faces representing a hypothetical discrete tunnel where there is no line of sight between the source and target points s and t its opposite ends. The points a and b are (two of the six) inner corner vertices (ICVs) on the walls of the tunnel, because $\theta_a, \theta_b < \pi$. However point c is not an ICV because $\theta_c > \pi$. The line segment ab is the line of sight between the ICVs a and b, and mn is the corresponding extended line of sight (ELOS).

The initial phase of the approximate algorithm starts by computing the longest possible line segment (or run) from the source and target points (also referred to as terminal points) through an ICV ahead of it, to the farthest possible shared segment along the polygon path without exiting the path, and formally begins from whichever terminal point that offers the longest run. This terminal point is referred to as the start, while the other terminal point is referred to as the end point. For example in Figure 5.10(a) the target point t offers a longer run than the source point s, therefore the algorithm begins from the point t.

If other terminal (or end) point is not visible from the end of a previous run, another run is similarly constructed as the longest possible line segment from the current point though any of the ICVs ahead of it to the farthest possible shared segment, until there is a line of sight between the current point and the end point. The discrete curve $tabcs$ in Figure 5.10(a) shows the typical outcome of this process.

The next phase of the algorithm tries to reduce the length and segment count of the initial discrete curve by intersecting the longest possible run from the end point and the farthest possible segment of the initial discrete curve. If such an intersection exists, i.e. the longest possible run from from the end point intersects any of the segments of the discrete curve, the line segment from the end point to the intersection replaces the portion of the discrete curve that it bridges. If the start point is not visible from the intersection (or current) point this step is repeated. In other words, the intersection of the farthest run and the longest possible run from the previous intersection point is computed, and a new

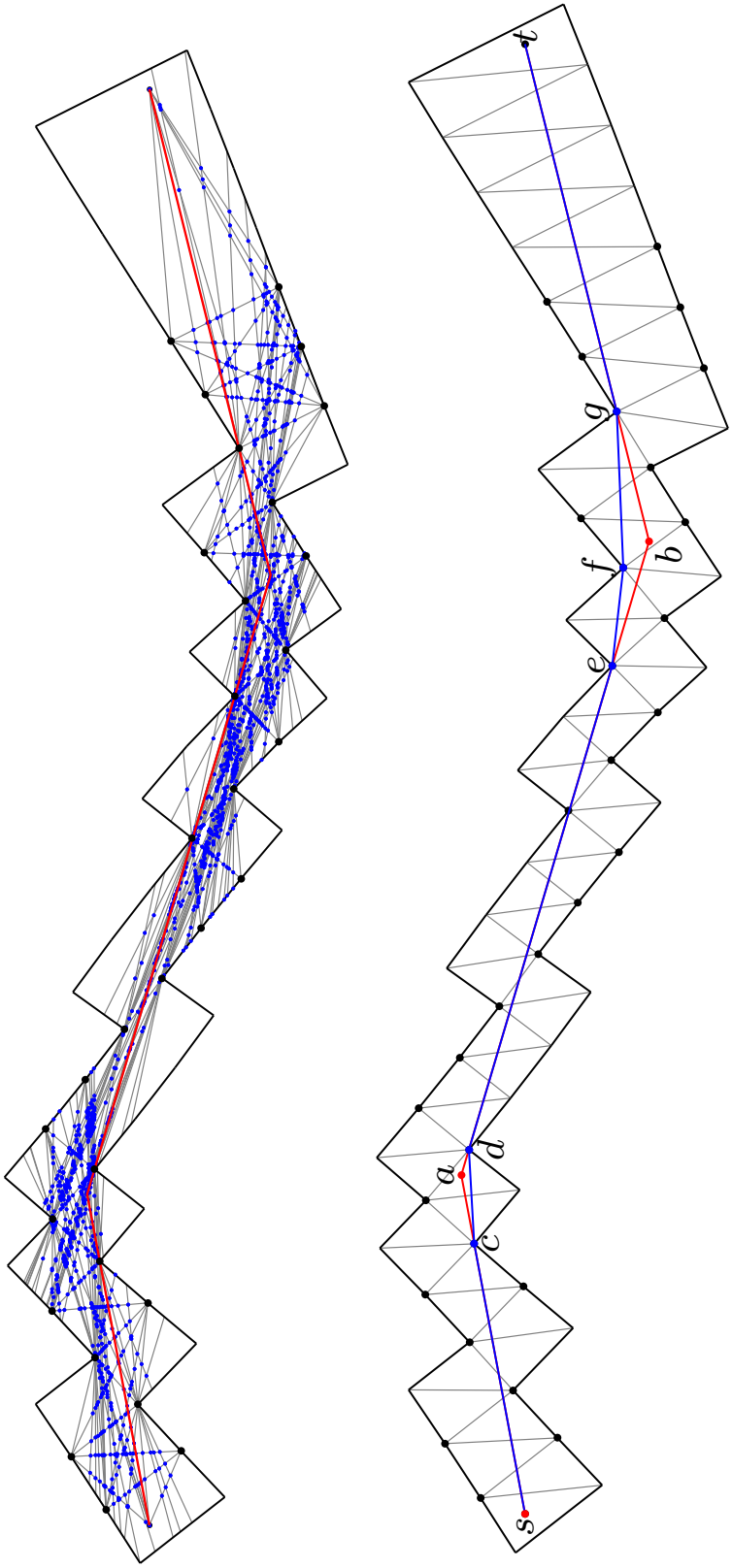


Figure 5.9 (a) Left: visibility graph search for the shortest path having the smallest possible segment count. Gray lines are the extended lines of sight (ELOSs) and the black dots are the inner corner vertices (ICVs). (b) right: the curve with the minimum segment count (red) can also be obtained by a simplification Chen *et al.* (2005) of the shortest possible path (blue).

line segment constructed between the old and new intersection points. Again this new line segment replaces the segments of the discrete curve that it bridges. This is repeated until there is a line of sight between the current intersection point and the start point, as shown in Figure 5.10(b). Both phases of this algorithm are outlined in Algorithm 3.

5.4 PATH CORRECTION AROUND VERTICES

Although bridging iteratively produces straighter discrete curves the algorithm does not always converge to a straightest geodesic, although one may exist. This is because bridging makes several relatively-large scale corrections to the input polygon path. Therefore it may be necessary to make smaller-scale stepwise corrections to the output of the bridging algorithm. Each correction is made to the current polygon path around an ICV v that is maximally displaced from the straight line l connecting the source and target points. As shown in Figure 5.11, v is chosen from the set of ICVs for which the perpendicular line segment from each ICV to l does not penetrate the polygon path. The selected vertex v has the longest such perpendicular line segment.

Although v is identified on the unfolded path, the correction is performed as shown in Figure 5.13(a) and (b) on the original, unflattened input polygon path and proceeds as follows. First, it is essential to point out that the input path to be corrected includes a subset of the ring of polygonal faces that share v, enters the ring at an entry face f_1, traverses it in a clockwise or counterclockwise direction and exits the ring at f_2, see Figure 5.13(a) – inset. Therefore, the input path can be divided into three sections: the first subpath begins from the first face of the input polygon path up to and including f_1, the second or middle subpath consists of the faces that belong to the ring excluding f_1 and f_2, and the third subpath begins from f_2 to the last face. Correction is simple and involves replacing the middle section with the sequence of faces, excluding f_1 and f_2, that are encountered when the ring is traversed from f_2 to f_1 – in the same direction as the input path traverses the ring (refer to Figure 5.13 (a)).

Tests indicate that this stepwise iterative path correction around maximally displaced vertices rapidly finds the sequence of polygonal faces that embeds a straightest geodesic if one exists, in the direction suggested by the straight line that connects two points. However when no such path exists, the output of the algorithm oscillates around 2-4 polygon paths. Typically the algorithm is terminated after 5–10 steps if a path that embeds a straightest geodesic is not found.

The combined set of heuristics is validated as follows. First a new surface S is constructed by widening or broadening the last output path produced during correction to include the sequences of faces that are offset from it on either side, as shown in Figure 5.13(c). S represents the domain or set of faces that a path that embeds the straightest geodesic is expected to include, if such a path exists. Thereafter, following Migliore *et al.* (1990), the sequence of all possible paths of faces that belong to S and connect the source and target faces are generated, and checked if they embed a straightest

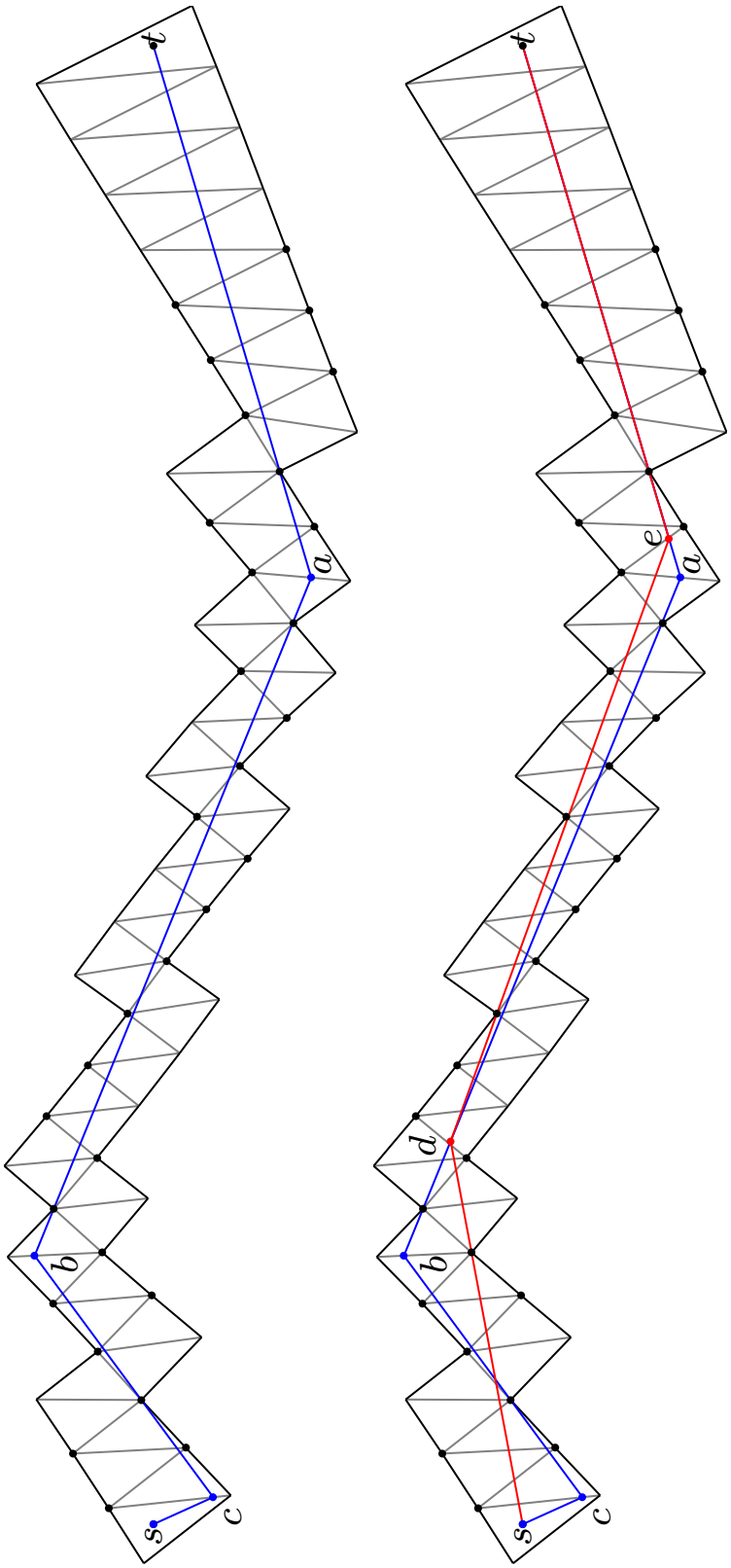


Figure 5.10 Approximate minimum segment-count path finder. (a) left: forward trace, generates the curve tabcs. (b) right: reverse trace, intersects initial (forward) trace, generates straighter curve, with smaller segment count, teds.

Algorithm 3: MINIMUM SEGMENT-COUNT PATH FINDER (APPROXIMATE)

Input: inner corner vertex (icv), source point (srcPt), target point (targPt)
Output: path/segment list (segList)
Data: start point (sP), end point (eP), current point (cp), current segment (s)
Data: previous segment (prevSeg)

```
1  pA ←EndOfLongestSegmentFromPointInDirOfTarget(targPt)
2  pB ←EndOfLongestSegmentFromPointInDirOfSource(srcPt)
3  if Distance(pA,srcPt) > Distance(pB,targPt) then
4      sP ←srcPt
5      eP ←targPt
6      cp ←pA
7  else
8      sP ←targPt
9      eP ←srcPt
10     cp ←pB
11 end
   // Forward trace (start to end)
12 seg ←CreateSegment(sP,cp)
13 add seg to segList
14 while no line of sight exists between cp and eP do
15     seg ←LongestSegmentFromPointInDirOfEnd(cp)
16     cp ←OtherEndPointOfSegment(seg,cp)
17     add seg to segList
18 end
19 seg ←CreateSegment(cp,eP)
20 add seg to segList
   // reverse trace (end to start)
21 cp ←eP
22 prevSeg ←last segment in segList
23 repeat
24     seg ←LongestSegmentFromPointInDirOfStart(sP)
25     cp ←OtherEndPointOfSegment(seg,eP)
26     foreach s in segList (iterating from start to end) do
27         cp ←Intersect(s,seg)
28         if cp is not equal to 0 then
29             set end point of s to cp
30             replace all segments between prevSeg and s (including prevSeg) with seg
31             prevSeg ←seg
32             exit foreach loop
33         end
34     end
35 until line of sight exists between cp and sP
```

geodesic. However, in order to curb the explosion of non-viable paths, after each face is added to the current path being generated, the path is tested for an interval of optimality Mitchell *et al.* (1987); Surazhsky *et al.* (2005) (see Figure 5.11) from the source vertex to the farthest internal edge of the polygon path, and the search along that path is terminated if no interval of optimality is found. In this figure, in which the sequence of faces is grown from the left to the right, no interval of optimality would exist after the addition of the gray-shaded triangle. An important feature of this recursive process, summarized in Algorithm 4, is that all paths generated contain at most one instance of a polygonal face. Such paths are termed simple.

During the process of muscle construction on the three head models shown in Figure 6.2 (see Section 7.2.4) an extended corridor was constructed and searched whenever all three heuristics failed to find a straightest geodesic (see Tables 7.1 and 7.2). This search failed to find a straightest geodesic in all cases.

As a natural extension of this process, $\mathcal{S}$ can be widened further and taken to be all the faces of the polyhedral mesh, and Algorithm 4 run in order to find all simple polygon paths on the mesh that embed a straightest geodesic connecting two points, as shown in Figure 5.12. This reevidences the non-uniqueness of straightest geodesics as well as the assertion that not all straightest geodesics are shortest geodesics. Only one of the paths on the object is found by the vector projection, path straightening and correction algorithms.

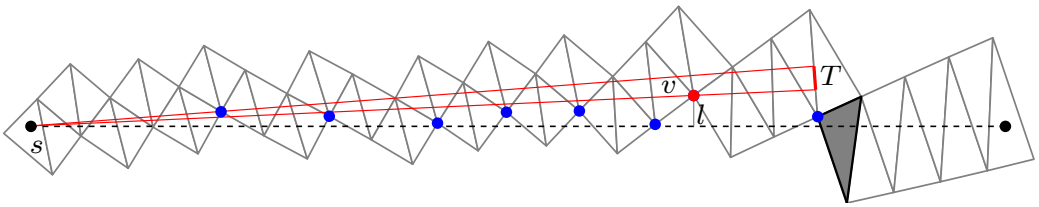


Figure 5.11 Path correction around maximally displaced ICVs. The unfolded input path shows the ICVs (blue dots) considered for correction. The line segment l from each ICV to the (dashed) line segment, connecting the source and target points, does not penetrate the polygon path. The selected ICV (red dot) is maximally displaced from the straight (dashed) line connecting the source and target points. The red triangle is interval of optimality from point s to face T.

5.4.1 *Straightest possible geodesics*

As discussed in the previous section, when no straightest geodesic can be found in the direction of the Euclidian vector connecting both points, the output of the path correction algorithm oscillates around several polygon paths. If the straightest geodesic is

Algorithm 4: Exhaustive Path Search : EXHAUSTIVE

```
Global data: source point (sp), source face (sf), target point (tp)
Global data: target face (tf), list of paths (paths)
Input: current face (cf), current edge (ce), current path (currPath)
Initialization (one-time): cf = sf, ce = 0, paths ← [ ]
Initialization (one-time): currPath ← [ ] , add cf to currPath
1  foreach edge in cf do
2      face ← GETNEXTFACE( edge, cf)
3      if face is not equal to 0 then
4          if currPath does not contain face then
5              clonedPath ← CLONE(currPath)
6              add face to clonedPath
7              if face equals tf then
8                  if clonedPath embeds straightest geodesic then
9                      add clonedPath to paths
10                 end
11             else
12                 IOpt ← INTOFOPTIMALITY(sp, edge)
13                 if IOpt is not equal to 0 then
14                     EXHAUSTIVE(face, edge, clonedPath)
15                 end
16             end
17         end
18     end
19 end
```

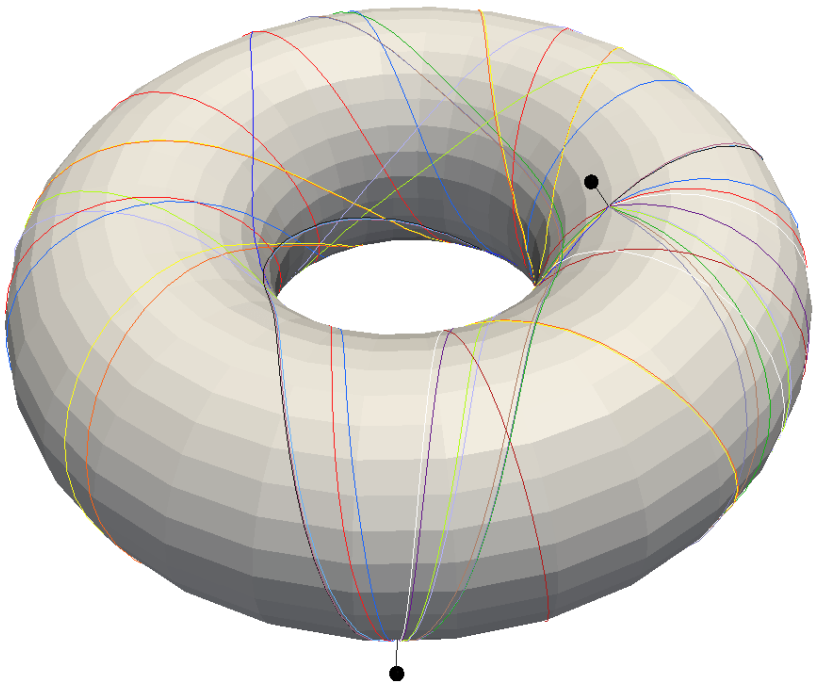


Figure 5.12 The 18 straightest geodesics connecting two points on a polyhedral surface, found by the exhaustive search technique.

nonetheless desired in this direction, the alternative is to compute the *straightest possible geodesic* that connects both points. Note that, although the adjectives "straightest" and "straightest possible" suggest the same meaning, the former term already has a meaning as a curve with zero TSC, while the latter implies the curve with the minimum TSC embeddable in a polygon path. The straightest possible geodesic is taken to be the straightest, as measured by the TSC, of all minimum-segment curves (see Section 5.3.1) that can be embedded in each of the polygonal paths about which the output of the path correction algorithm oscillates.

5.5 SUMMARY

Straightest geodesics are discrete curves with zero geodesic curvature, for which the initial value problem of geodesics is uniquely solvable. However, the initial value problem is unsuitable for the problem of generating facial muscle fibres, as are existing methods of generating shortest geodesics. In this chapter, a three-step, heuristically-based method for finding the path or sequence of polygonal faces that embed a straightest geodesic in the direction of the line connecting two given points has been developed.

Tests show that this method is several orders of magnitude faster than conventional methods of computing shortest paths on surfaces. For example computing the shortest geodesic between two given points on a high-resolution (23,576 triangle) SMAS model

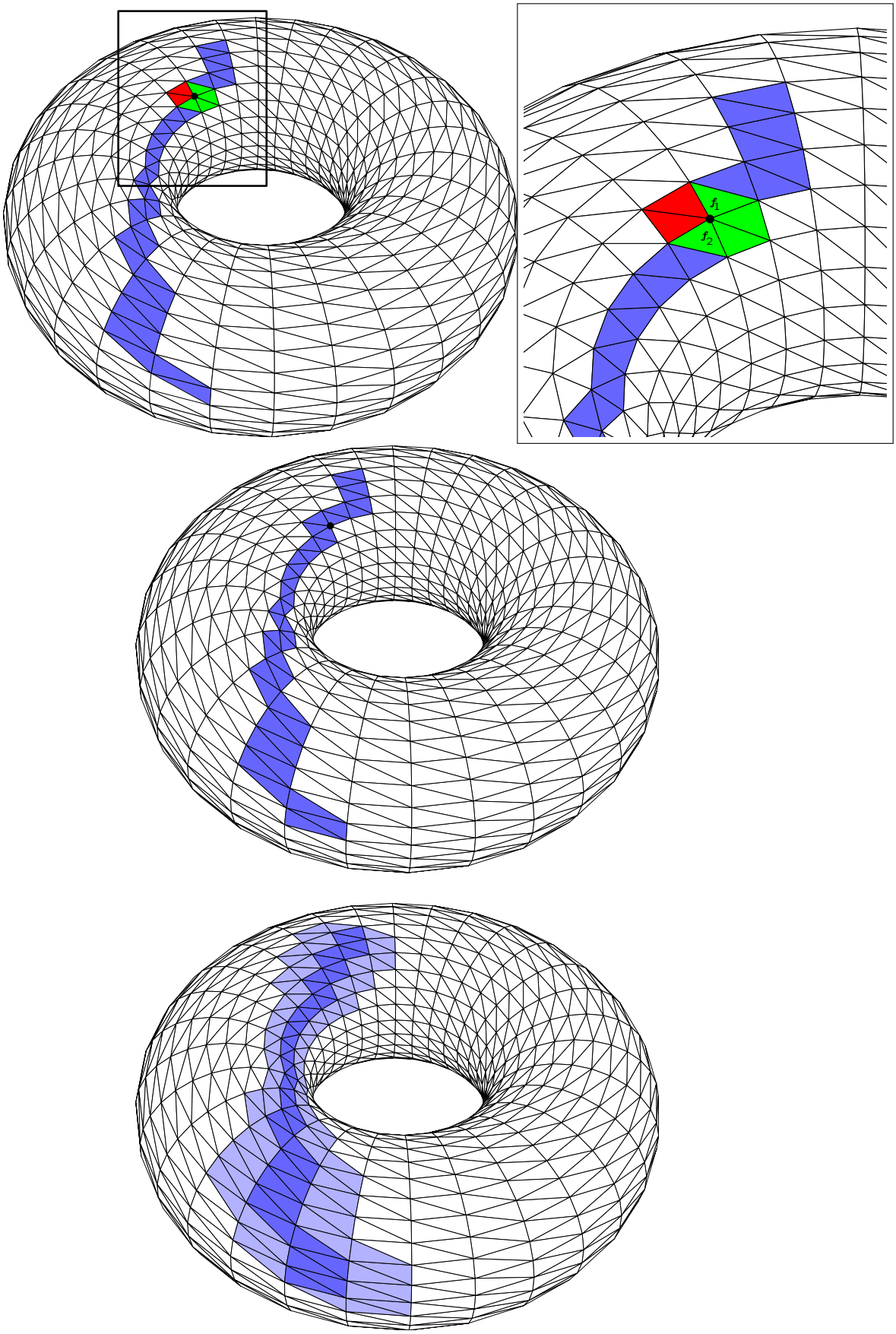


Figure 5.13 Path correction around maximally displaced ICVs (a) top: path (green and deep blue) before correction around a vertex (black dot). Faces to be added to the path are shown in red. (b) middle: path after correction. (c) bottom: a path broadened to include faces (lighter blue) offset from it in preparation for an exhaustive search.

(see Section 7.1) took over 100 seconds using the freely-available implementation of the MMP algorithm (available from http://code.google.com/p/geodesic/), while the method developed in this section required as little as 40 milliseconds in order to trace a straightest geodesic between the same pair of points. Even on a lower resolution (900 triangle torus) model, the heuristic algorithm was often at least 3 times faster than the MMP method.

Tables 7.1 and 7.2 show that an average of 1000 geodesic computations are typically required for generating a total of 176 muscle fibres for the nine muscles shown in Figures 7.13 and 7.14. The 824 additional geodesics are supplementary to the process of generating the 176 fibres. These results also show that the three-tiered algorithm is relatively quick and computes an average of 1000 (multiple source, multiple destination) geodesics in about 8 minutes, on commodity hardware, running unoptimized code.

The major findings of this chapter are as follows:

i. while a straightest geodesic may not always exist in the direction suggested by the straight line joining two points, several other less obvious straightest geodesics connect both points and can be traced, provided that such straightest geodesics are allowed to traverse no face more than once. The number of traceable paths grows exponentially if this restriction is removed. A simple, efficient and parallelizable algorithm for finding all such paths connecting any two points on a surface has been defined.

ii. where possible, the obvious or preferred straightest geodesic connecting any two points on a surface may be found in the direction suggested by a straight line connecting both points. Accordingly, three heuristically motivated algorithms for finding the sequence of polygons that embed a straightest geodesic connecting both points, in the preferred direction, was developed.

iii. where the straightest geodesic cannot be found, for example in the neighborhood of hyperbolic vertices, if the problem permits, the straightest possible geodesic can be computed instead as a fallback solution. Therefore, the concept of a straightest possible geodesics has been defined and a method for approximating such paths from the output of the previous algorithms presented.

5.5.1 *Future work*

Straightest geodesics are a relatively new concept and require further study with a view toward addressing a number of open questions. For example, the only known proof of the existence or otherwise of a straightest geodesic connecting two points, $\mathbf{p}_1$ and $\mathbf{p}_2$, requires that the faces, f_1 and f_2, on which both points lie share a vertex v. Unfortunately, this proof is too specific as it only asserts that a subset of the ring of faces that share v embeds a straightest geodesic or otherwise, and does not consider alternative sequences

of faces that connect f_1 and f_2. More generally, there is, as yet, no method of asserting the existence or otherwise of a straightest geodesic connecting $\mathbf{p}_1$ and $\mathbf{p}_2$ when f_1 and f_2 do not share a vertex. Furthermore, it is still unknown whether there exists a pair of points that cannot be connected by a straightest geodesic.

Therefore, in most cases, the only way to determine if there exists a straightest geodesic connecting $\mathbf{p}_1$ and $\mathbf{p}_2$ is to compute it. This chapter presented one such algorithm for computing straightest geodesics based on a Euclidian-distance, path straightening and correction heuristics. This algorithm does not always converge; and although in all cases in which it does not tests indicate that no straightest geodesic exists in the direction suggested by the Euclidian distance heuristic, this does not amount to a proof. One conceivable method of constructing the required proof might be to investigate the adequacy or otherwise of the combination of heuristics employed, and perhaps also the class of problems for which they are effective. This approach promises to yield valuable insight into the sensitivity of this algorithm to the properties of the mesh and the convergence properties of the algorithm.

In general, computing straightest geodesics appears to be an NP-problem because the brute-force, exhaustive search algorithm, described in Section 5.4, is currently only known method of finding any or all straightest geodesics that connects any two given points.

Also, the suitability of the cutting plane method of (Lee *et al.*, 2006) should be studied as an alternative to path projection on severely undulating surfaces, where the latter algorithm fails. In such cases, the discrete curve and sequence of faces generated by the cutting plane method are considered rough approximations that must be iteratively straightened by bridging and path correction.

And he said unto me, Son of man, can these bones live? — Ezekiel 37:3

6 A LANDMARK BASED METHOD FOR SKULL FITTING

In theory, a skull can be constructed for a CG head model either procedurally, from a number of parameters extracted from the model, or by deforming a generic skull to fit the given model. However, because the human skull has highly irregular and complex geometry, an equally elaborate algorithm would be needed to procedurally model it. Therefore, no such model exists; and even if one did exist it would require a large number of parameters that would have to be uniquely determined for every CG head model. In contrast morphing a generic skull to fit a face model is considerably simpler, and can be accomplished using three-dimensional thin-plate splines, computed from landmark correspondences, as briefly shown in Section 4. Kähler *et al.* (2002) used such a deformer and a reference skull in "generating animated head models with anatomical structure". However the approach employed in this work has several important differences, such as the use of sliding semilandmarks, experimentally-obtained facial tissue depth data in calculating skin thickness at established cephalometric (facial) landmarks, and an interactive technique for real-world to 3D unit conversion.

Accordingly this chapter describes the development of the resulting landmark-based method for fitting a generic skull to a given head model, with particular emphasis on the number (count), type, distribution (or placement) and other attributes of the landmarks.

6.1 SKULL FITTING PROCESS

6.1.1 *Model preparation*

The generic skull used in this study was based on the laser scan model of the adult male human head contained in the companion CD of the text Fleming and Dobbs (1999). Although the major features of the human skull were clearly identifiable on laser scan, the model consisted of too many polygons and had poor topology. In addition both rows of teeth had poor detail and were fused to the skull. Therefore a new model of the skull was constructed by an artist to match the proportions and gross features of the laser scan model. Care was also taken to minimize the polygon count of the new (generic) skull, while featuring the necessary detail and optimizing the topology of the mesh. Although the generic skull can be of any type (polygonal, subdivision surface or NURBS), the convenience offered by triangular models, and the availability of a corresponding C library[1] informed the choice of a triangular mesh model for the generic skull.

1 GNU Triangulated Surface Library (GTS) http://gts.sourceforge.net/

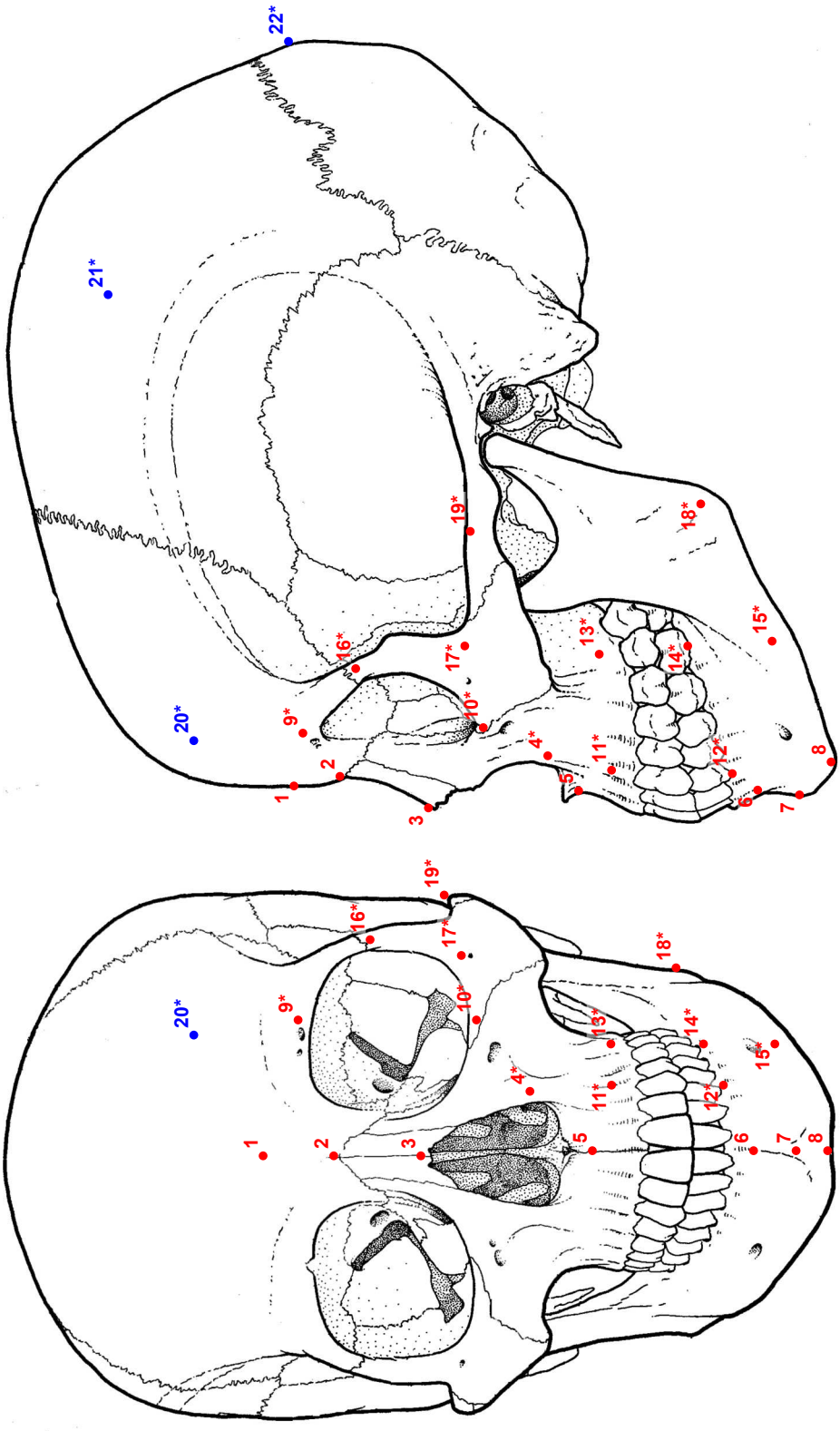


Figure 6.1 Craniometric landmarks. Landmarks prescribed by Manhein *et al.* (2000) are drawn in red, supplementary landmarks are drawn in blue, and paired landmarks are drawn with asterisks.

Thirty-one (31) landmarks (7 on the midline, and 12 on either side), as shown in Figure 6.1, were manually placed on the skull model, all in the region of the face, as prescribed by Manhein *et al.* (2000)[2]. However, in order to increase coverage, two additional landmarks were placed on the forehead, at both frontal eminences, and three at the rear of the skull; two on either parietal eminence and one on the external occipital protuberance. Therefore, 36 landmarks were defined on the generic skull. Each landmark was given an integer identifier. A corresponding set of 36 landmarks was also placed on three face models: one African, one European and the third from the Makehuman[3] project, as shown in Figure 6.2. The resolution of each head model is such that there is sufficient level of detail to adequately: represent the facial features and profile, locate the landmarks of the face and accurately compute the differential properties of each mesh at any given landmark, as described in Section 4.2.2.

Henceforth, landmarks on the skull model are termed craniometric, i.e. skull measurement, while landmarks on either head model are termed cephalometric, i.e. head measurement. Both sets of landmarks were readily placed, on the generic skull and the three head models, by a non-medical expert following the guidelines described by Manhein *et al.* (2000).

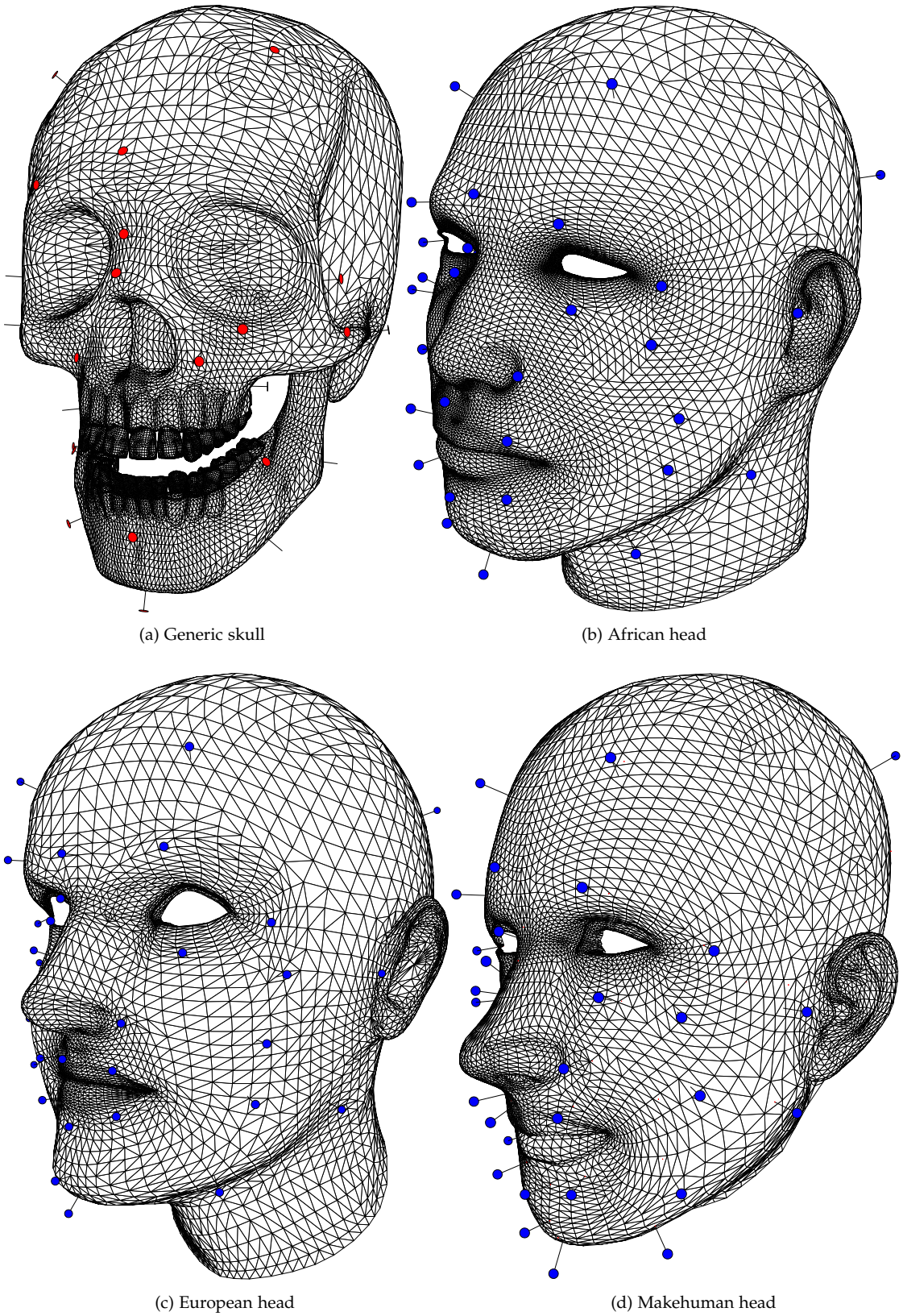
6.1.2 *Basic skull fitting process, using soft tissue depth data*

At any cephalometric landmark to which a generic skull has been correctly fitted, the distance between such a landmark and its corresponding craniometric landmark on the fitted skull will equal thickness of soft tissue at the given location. In order to guarantee this offset, cephalometric landmarks are assigned facial tissue depth values (from Manhein *et al.* (2000)) according to the age, sex, body type, and ethic origin assigned to the face model in consideration. (The five supplementary cephalometric landmarks are assigned the same depth value as the cephalometric landmark at the glabella, between the eyes in the forehead region, assuming the thickness of the soft tissue of the forehead and scalp are equal and constant.) Therefore, if a cephalometric landmark with a tissue depth d_t, is at the position $\mathbf{p}$ on the head model, and has a normal orientation $\mathbf{n}$ with respect to the surface of the model, as shown in Figure 6.3(a), the equivalent landmark on the skull should be at the position $\mathbf{q} = \mathbf{p} - d_t\,\mathbf{n}$, post fit.

However, because this computation is done for an arbitrary, unitless 3D space, the real world (millimeter) tissue depth d_t must be converted to 3D space units in order to be of valid or meaningful use. Such a conversion will answer the question: how

2 The study by Manhein *et al.* (2000) publishes facial tissue depth (FTD) data for 31 landmarks, collected from a generous sample of 551 children and 256 adults of various age ranges, ethnic origins, body types and genders. Measurements were conducted *in vivo* (on living subjects, thereby avoiding errors resulting from volumetric tissue changes *post mortem*), and used ultrasound technology which is safe and accurate. Furthermore, the study included a procedural guide for locating the flesh-covered bony points at which FTD measurements were made (in millimeters), and was selected for these reasons.

3 www.makehuman.org



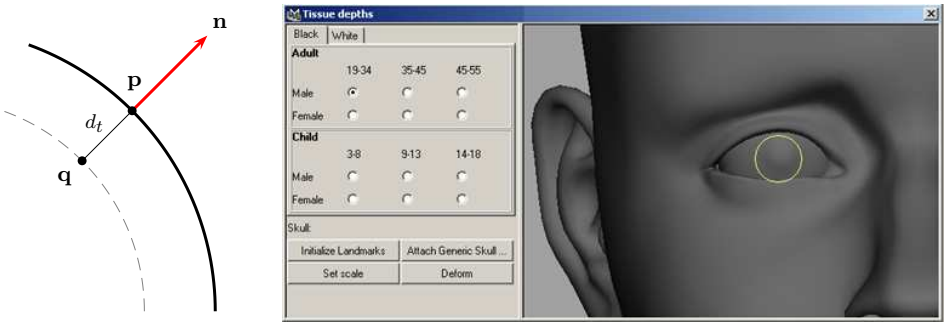
(a) Generic skull

(b) African head

(c) European head

(d) Makehuman head

Figure 6.2 3D models and landmarks. Craniometric landmarks are drawn in red, and cephalometric landmarks in blue.

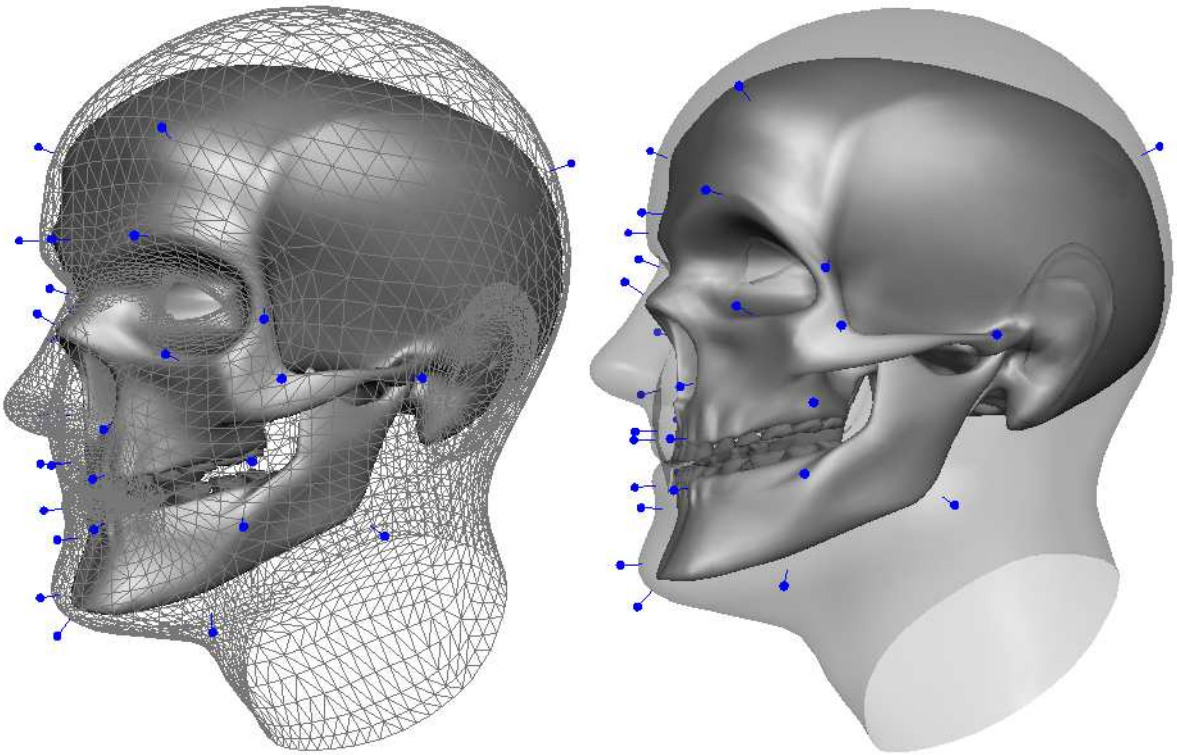


(a) (b)

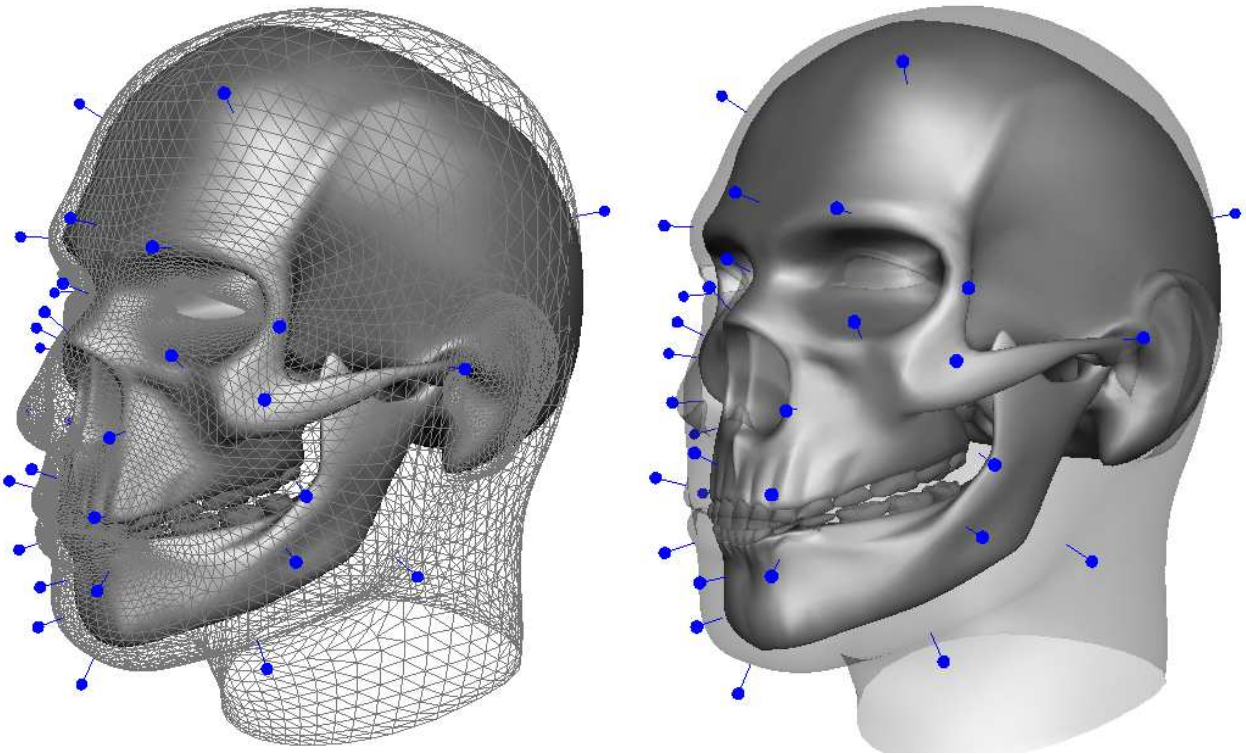
Figure 6.3 (a) Computing the position of an equivalent craniometric landmark **q**. The surface of the head model is drawn in black, surface normal in red, and the skull surface in dashed, light gray (**p** and **n** are computed using the patch-fitting, projection and transformation procedures described in section 4.2.2). (b) Interactively sketching (or sizing) a virtual cornea, shown as a faint yellow disk.

many millimeters (or real-world units of choice) equals one 3D space unit? One way to perform such a conversion would be to consider the dimensions of a facial feature that is known to be of fairly constant size in all real persons. An example of such a feature is the cornea – the transparent anterior part of the external coat of the eye covering the iris and the pupil. According to Adler and Scheie (1969) *"the diameter of the cornea in the adult eye averages 12mm"* (see Taylor (2000)). Therefore by interactively sketching, and thereby measuring the diameter of a virtual cornea d_{vc} on a given model as shown in Figure 6.3(b), the desired (3D space to real-world millimeter) conversion factor f is given simply as: $d_{vc}/12$, so that the position of the equivalent craniometric landmark (post fitting) becomes $\mathbf{p} - f\, d_t\, \mathbf{n}$, where **p** and **n** are computed using the patch-fitting, projection and transformation steps described in section 4.2.2.

The set of craniometric landmark points, paired with the corresponding tissue-depth adjusted cephalometric points can now be used to construct three 3D thin-plate splines that deform the generic skull to any of the three head models, as shown in Figure 6.4 for the European and African head models. In both cases there is a good fit of the skull in the region of the face, but an unsatisfactory fit in the region of the nose, and the top, as well as sides and rear of the head. This is due to the abundance of landmarks on the face, and the few number of landmarks on the forehead, as well as the top, sides and rear of the head. There is also an unnatural squashing of both rows of teeth, due in part to the slightly opened jaw of the generic skull. (This is the position that the human jaw assumes at rest.) Improving the fit therefore requires that new landmarks are added to the top of the head and the skull, while some of the ill-positioned landmarks along the rows of teeth are removed and the mandible of the generic skull is rotated around the condyle of its ramus in order to bring the upper and lower rows of teeth closer to each other.



(a) European head



(b) African head

Figure 6.4 Fitting generic skull to head models, using the full set of landmarks.

The pointy chin of the skull fitted to the European head model, see Figure 6.7 (and Figure 6.9) is perhaps caused by any, or both, of the following:

i. Incorrect soft tissue depths assigned to some or all of the landmarks 6, 7 and 8 (see Figure 6.1), on the lower face, which nevertheless appear to be correctly placed. It must be borne in mind that the (true) facial tissue depth of an individual will vary about the population average. Therefore soft tissue depths in and around certain "problem areas", like the chin in this case, can be varied within acceptable limits in order to obtain anatomically feasible skulls.

ii. Significant variation between the (concave) facial profile of the European head and that of the generic skull. In comparison, the facial profiles of the other two heads models are much closer to that of the generic skull. The African head has only a slightly convex profile while the MakeHuman head has a relatively straight profile.

   There are two possible solutions to this problem. First, a generic skull can be constructed for each possible facial profile, and each head model paired, i.e. fitted, with a generic skull having the same profile. Unfortunately, the number of possible facial profiles, caused by the relative growth of the maxilla and mandible beyond the normal limits of the face[4], makes this solution unattractive. Nevertheless, because of the considerable differences between adult humans and primates or even infants for example, this approach is the only option when generating skulls for some subjects. Alternatively, a protocol for distributing landmarks on the face-skull pair could be developed for various facial profiles. For example, it possible that a set additional semilandmarks suitably placed in the region of the chin of the European head model will produce a more rounded chin.

6.1.3 *Incorporating semilandmarks and derivative information*

Unfortunately implementing the first of the preceding recommendations is not as straightforward as it may seem. This is because arbitrarily distributing the new landmarks (henceforth referred to as semilandmarks), makes little sense considering that a landmark by definition ought to identify a biologically salient feature, and no obvious features exist on the top of the head (or skull). Fortunately, the semilandmark method described in Section 4.2.5 can be used to reliably place landmarks on such a featureless structure as the top of the human head (or skull). This technique requires that semilandmarks (also given

4 For example, maxillary prognathism (the maxilla protrudes beyond the normal limits of the face, or outgrows the mandible), mandibular prognathism also known as the "Habsburg jaw" (the maxilla protrudes beyond the normal limits of the face, or outgrows the mandible), bimaxillary prognathism (the maxilla and mandible protrude forward of the normal limit of the face), maxillary retrognathism (the maxilla is posterior to the normal limits of the face), mandibular Retrognathism (the mandible is posterior to the normal limits of the face) and bimaxillary retrognathism (the maxilla and mandible is posterior to the normal limits of the face). In extreme cases, some of these conditions require correction by maxillofacial surgery.

integer identifiers) be initially distributed on both models[5], but those semilandmarks on the skull be allowed to slide on its surface, so that the total bending required to morph the skull to the head is minimized. The semilandmarks of the skull are made to slide, while the semilandmarks of the head are kept stationary. Care must however be taken so that craniometric semilandmarks do not slide too far (e.g. deep into the temporal region). Sliding is performed using the procedure described by Gunz *et al.* (2005) for semilandmark "relaxation on surfaces". Given a set of k landmarks $\mathbf{x}_1, \cdots, \mathbf{x}_k$ and corresponding set of landmarks $\mathbf{x}'_1, \cdots, \mathbf{x}'_k$ on a source and target objects respectively, the last m of which are semilandmarks, the method proceeds as follows:

i. arrange the coordinates of both sets of landmarks, sliding and non-sliding, into a vector, e.g. $\mathbf{x} = (x_1[0], \cdots, x_k[0], x_1[1], \cdots, x_k[1], x_1[2], \cdots, x_k[2])$

ii. construct a $3k \times 2m$ matrix $\mathbf{U}$ of the two unit length vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ that span the tangent planes of the slidable landmarks. This matrix has two columns per semilandmark and

$$\begin{aligned} U_{i,j} &= v_i[0] & U_{i,j+m} &= w_i[0] \\ U_{k+i,j} &= v_i[1] & U_{k+i,j+m} &= w_i[1] \\ U_{2k+i,j} &= v_i[2] & U_{2k+i,j+m} &= w_i[2] \end{aligned}$$

where $i = 1, \cdots, k$, $j = 1, \cdots, m$ and all the other elements are zero. The vectors $\boldsymbol{v}$ and $\boldsymbol{w}$ are the principal directions obtained by fitting a patch at the position of the semilandmark, as described in Section 4.2.2.

iii. compute the $2m$ vector of parameters

$$T = (\mathbf{U}^T \mathbf{L}_m^{-1} \mathbf{U})^{-1} \mathbf{U}^T \mathbf{L}_m^{-1} \mathbf{x}$$

that determines the amount of sliding in the unit directions, so that the position of each semilandmark on the tangent plane, post-sliding, can be computed as

$$\tilde{\mathbf{x}} = \mathbf{x} + T[j]\,\boldsymbol{v}_i + T[j+m]\,\boldsymbol{w}_i$$

where

$$\mathbf{L}_m^{-1} = \begin{pmatrix} L_m^{-1} & 0 & 0 \\ 0 & L_m^{-1} & 0 \\ 0 & 0 & L_m^{-1} \end{pmatrix}$$

and L_m^{-1} is defined in Section 4.1.6 (page 62).

Because this procedure allows semilandmarks to travel on the tangent plane and escape the surface of the source object, $\tilde{\mathbf{x}}$ must be projected back to the source object after sliding.

5 Like the supplementary landmarks (located at the eminences), all new semilandmarks placed on the head are assigned the depth value of the cephalometric landmark at the glabella.

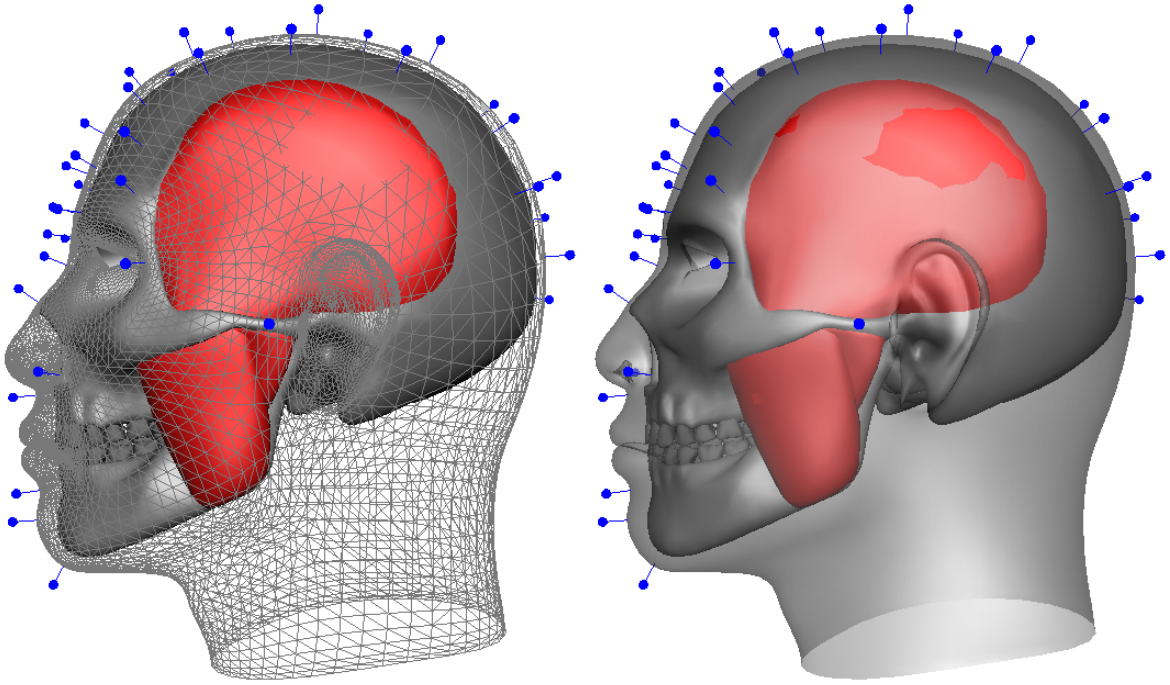
iv. the last two steps (sliding and projection) are repeated until the bending energy $\mathbf{x}^T \mathbf{L}_m^{-1} \mathbf{x}$ converges.

The paired landmarks 11–14 (eight in total) in the region of the mouth and teeth, see Figure 6.1, were removed on both sides of all three face models as well as the skull. In addition, two semilandmarks were placed in the nasal region and 31 semilandmarks placed in the forehead and cranial regions of all three head models as shown in Figures 6.5, 6.6 and 6.7. The above sliding procedure was then used to relax the positions of the semilandmarks on three copies of the generic skull against each head model. Subsequently three 3D thin-plate spline deformation functions (see Section 4.2.4) were computed based on the correspondence between the set of landmarks and semilandmarks placed on each head model and the copy of the generic skull associated to it. The deformation functions were then used to morph each generic skull to the corresponding head model, as shown in Figures 6.5, 6.6 and 6.7. In all three cases, the introduction of semilandmarks and the reduction in the set of landmarks produced a considerably better fit of the skull to the top, sides and rear of the head models, but not at the medial aspect of the infraorbital rim and the region of the zygoma for the European and MakeHuman heads respectively. Furthermore, in each case, the deformation function does not adequately fit an additional model of the temporalis muscle and fascia to the head model. (The generic skull package also includes models of the both masseter muscles, temporalis muscles and fascia and a collection of discrete curves. The relevance of this additional geometry is discussed in the next chapter.) Also, a careful examination shows that, occasionally, the curvature of the skull fails to match the curvature of the head in all three cases.

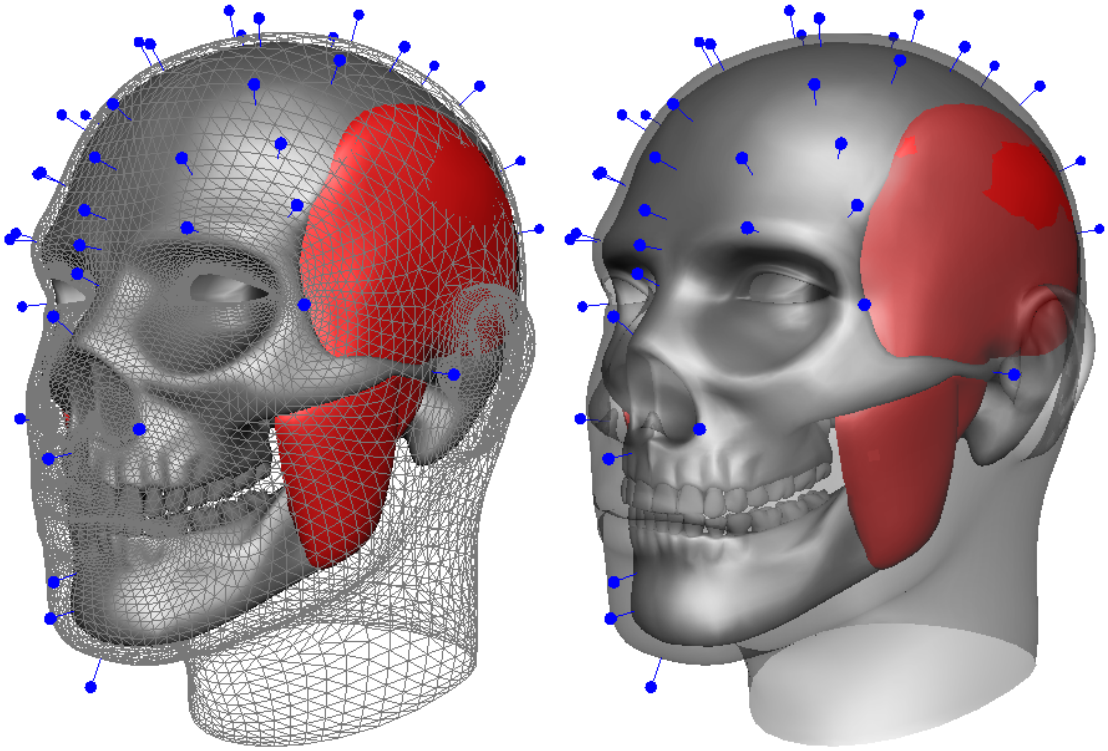
The pointy chin of the skull fitted to the European head model, see Figure 6.7 (and Figure 6.9), is probably due to

In order to correct the latter problem, another set of three 3D thin plate splines are computed for each head model and the copy of the generic skull fitted to it. These thin-plate splines are based on the correspondence between the normal and principal tangents at each landmark, using the technique outlined in Section 4.2.4 for Hermite data, and are used to deform the generic skull such that its normal $\mathbf{n}_s$ and principal tangents $\mathbf{t}_{s1}$ and $\mathbf{t}_{s2}$ at a subset of its landmark normals are in alignment with the corresponding normal $\mathbf{n}_h$ and principal tangents $\mathbf{t}_{h1}$ and $\mathbf{t}_{h2}$ of the landmarks of the head model to which it is fitted. (The normals and principal tangents are computed using the patch-fitting technique described in Section 4.2.2.)

However, $\mathbf{t}_{s1}$ and $\mathbf{t}_{s2}$ cannot be used as is in computing the Hermite thin-plate splines, because an arbitrary transformation, or deformation, $\mathbf{T}$ that brings $\mathbf{n}_s$ in alignment with $\mathbf{n}_h$ does not in general also align $\mathbf{t}_{s1}$ and $\mathbf{t}_{s2}$ with $\mathbf{t}_{h1}$ and $\mathbf{t}_{h2}$ respectively. Therefore, if $\mathbf{t}_{s1}$ and $\mathbf{t}_{s2}$ are used as is, the mismatch between the tangent vectors will generate local twists in the deformed object, as earlier illustrated in Figure 4.12(d), at each landmark. In other words, because $\mathbf{T}$ does not bring $\mathbf{t}_{s1}$ and $\mathbf{t}_{s2}$ into alignment with $\mathbf{t}_{h1}$ and $\mathbf{t}_{h2}$ an additional rotation about each landmark normal must be additionally built in to the deformation. Getting rid of the twists therefore requires that the principal directions

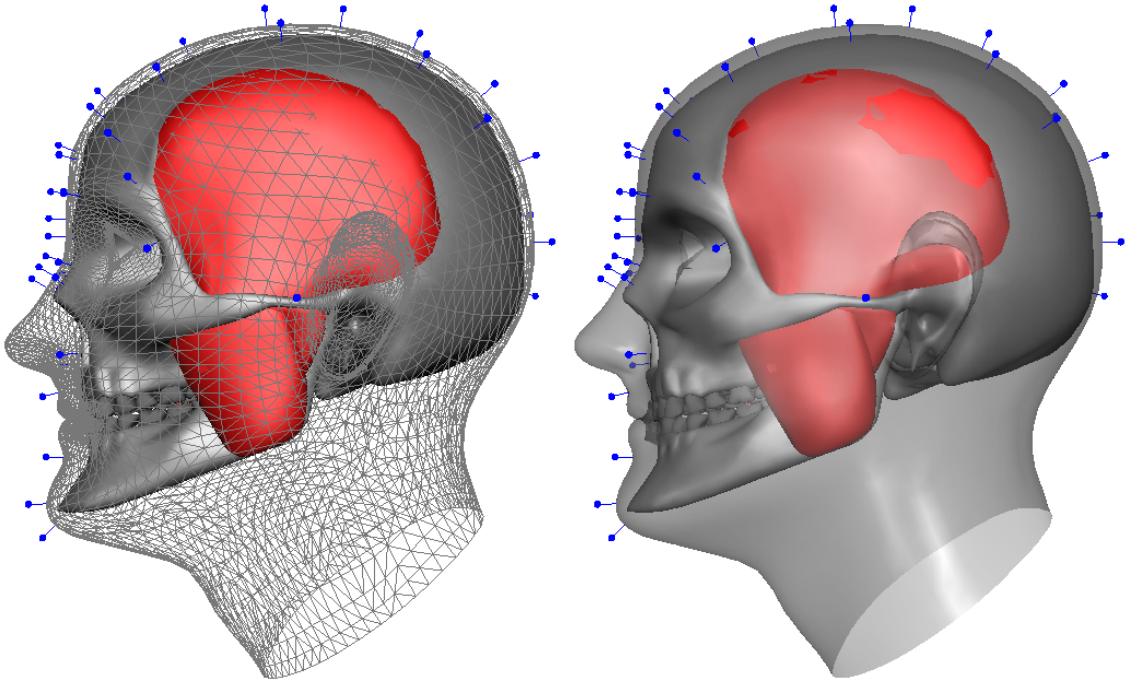


(a) Side view

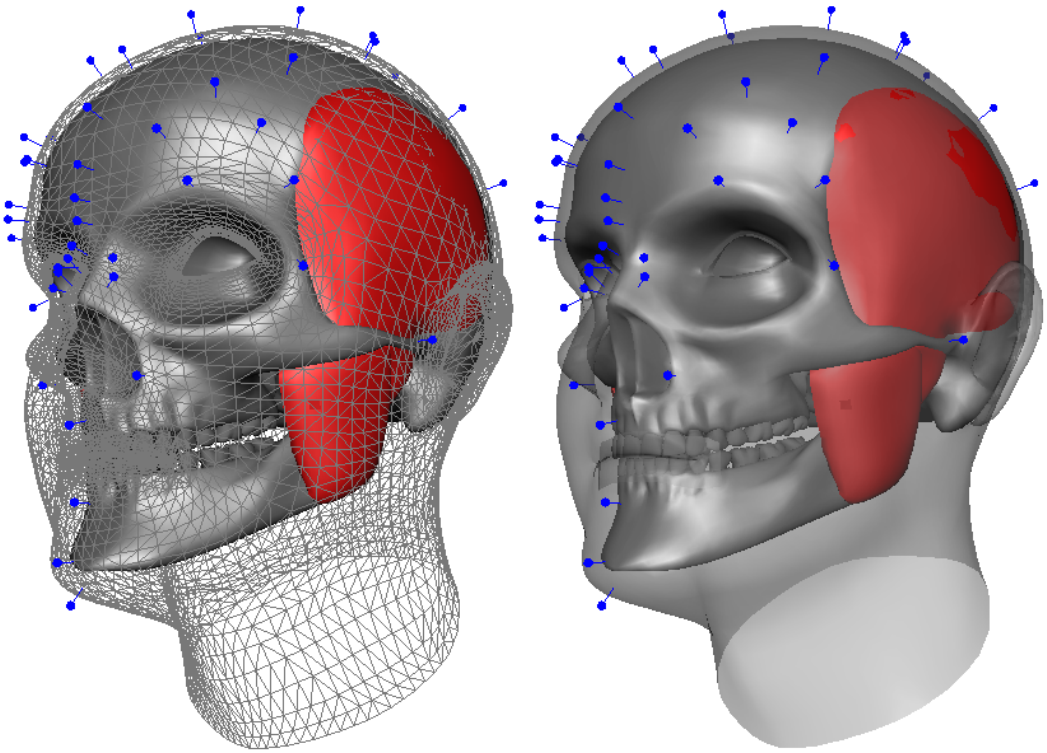


(b) Oblique view

Figure 6.5 Fitting generic skull to the African head model, based on subset of landmarks and 33 semilandmarks.

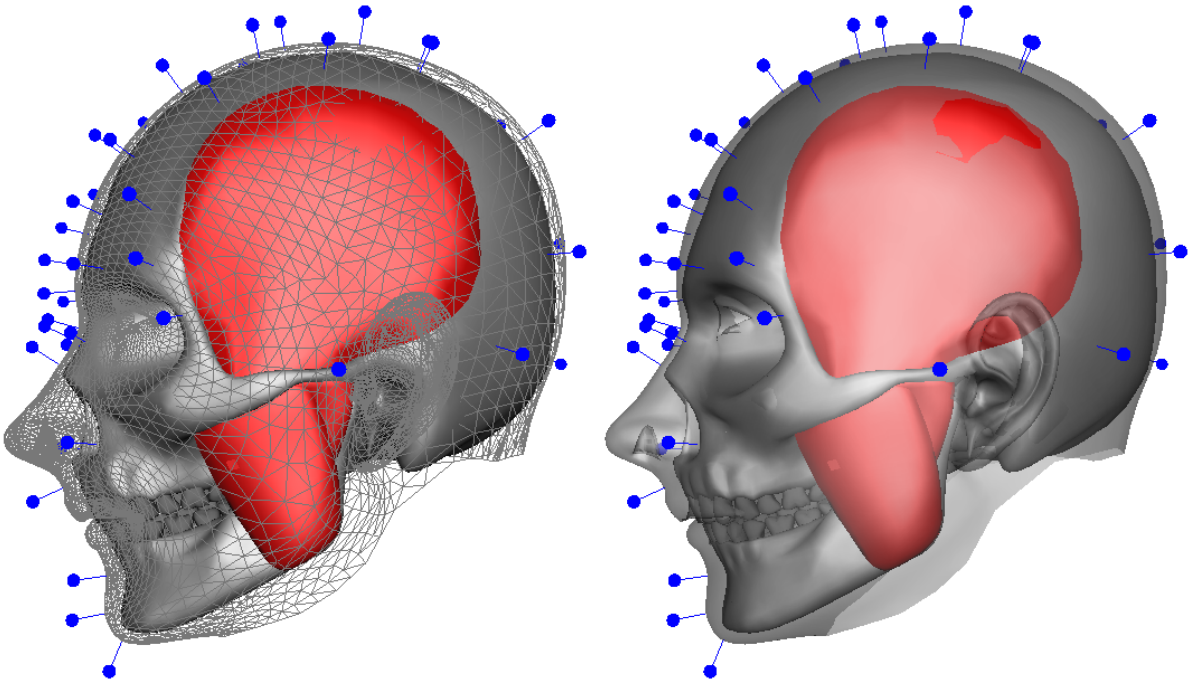


(a) Side view

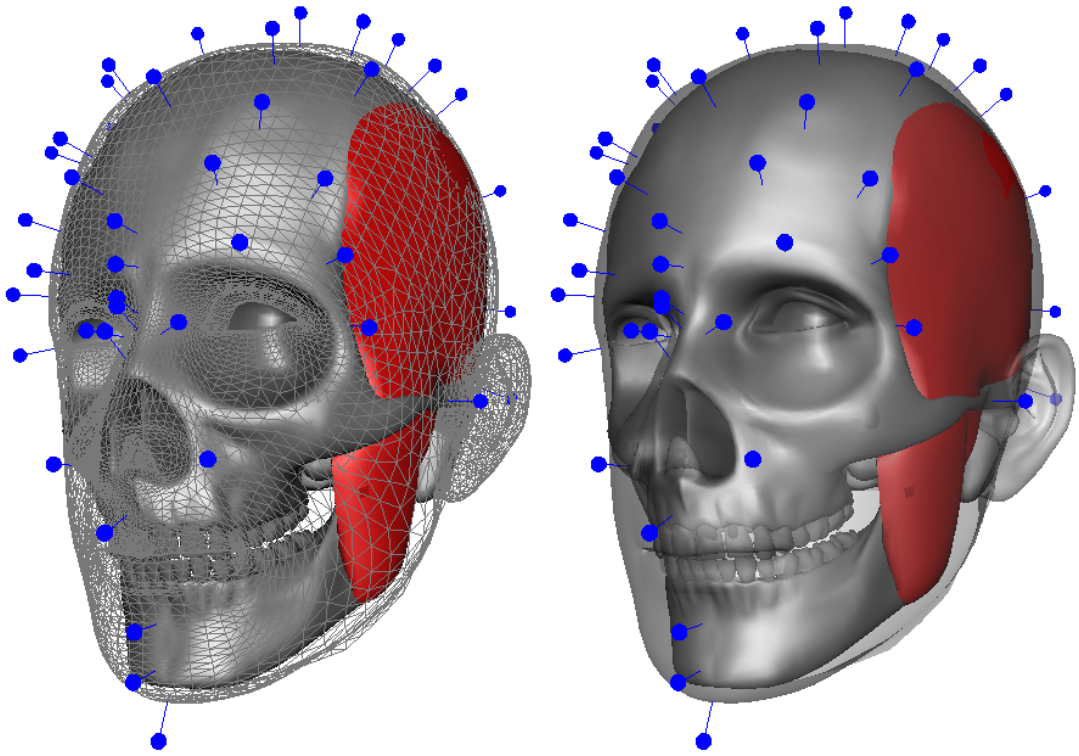


(b) Oblique view

Figure 6.6 Fitting generic skull to the European head model, based on subset of landmarks and 33 semilandmarks.



(a) Side view



(b) Oblique view

Figure 6.7 Fitting generic skull to the MakeHuman head model, based on subset of landmarks and 33 semilandmarks.

used be in alignment after applying $\mathbf{T}$. Because the inverse transformation $\mathbf{T}^{-1}$ aligns $\mathbf{n}_h$ with $\mathbf{n}_s$, and transforms $\mathbf{t}_{h1}$ and $\mathbf{t}_{h2}$ to the plane of (but not to) $\mathbf{t}_{s1}$ and $\mathbf{t}_{s2}$, the direction vectors in the plane of these vectors that will produce no twist can only be $\mathbf{T}^{-1}\,\mathbf{t}_{h1}$ and $\mathbf{T}^{-1}\,\mathbf{t}_{h2}$ respectively.

As shown Figures 6.8, 6.9 and 6.10, this secondary refitting of the initial generic skulls to all three head models corrects all of the aforementioned problems especially the ill-fitted temporalis muscle and fascia and the conspicuous deviations between the curvature of the generic skull and the head models.

Overall, skull fitting is a two-step process that involves first aligning landmarks and then their tangents. It is possible to merge both steps and simultaneously align landmarks positions and normals. However the best results are obtained by the two step process.

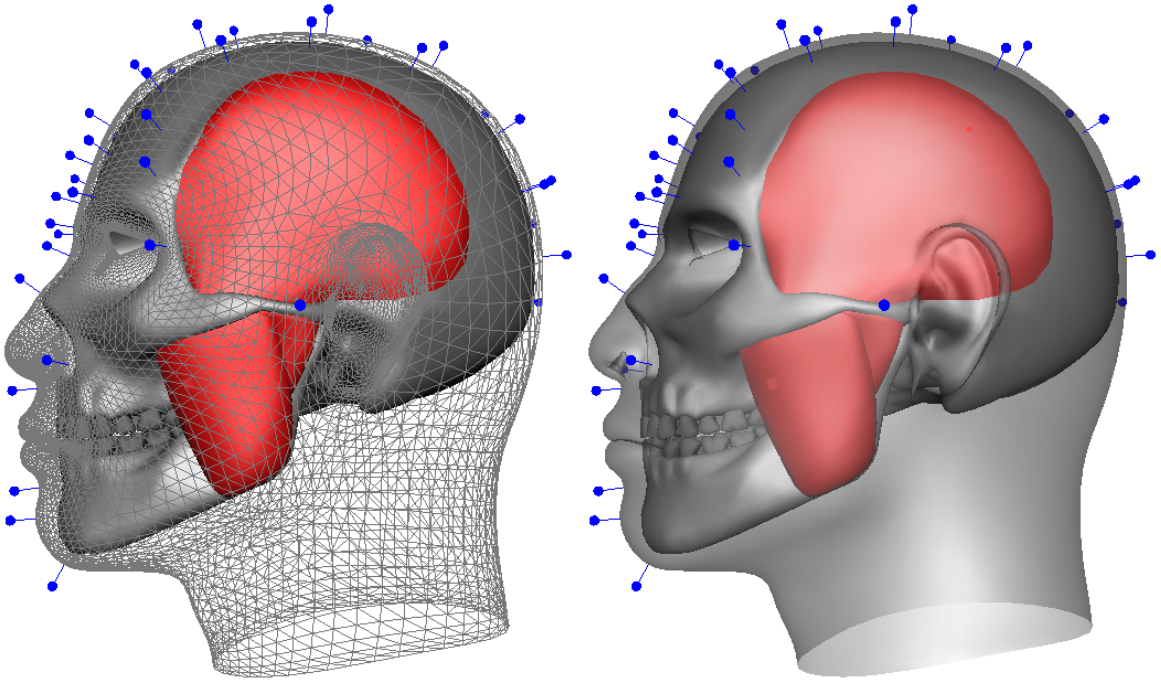
6.2 SUMMARY

In this chapter a two-step landmark-based, technique for fitting a generic skull to a 3D model of a human head was presented. This process incorporates real-world tissue depth data as well as the normal and tangent vectors at a subset of landmarks, with visually-satisfactory results. The most significant heuristics garnered from the process are as follows:

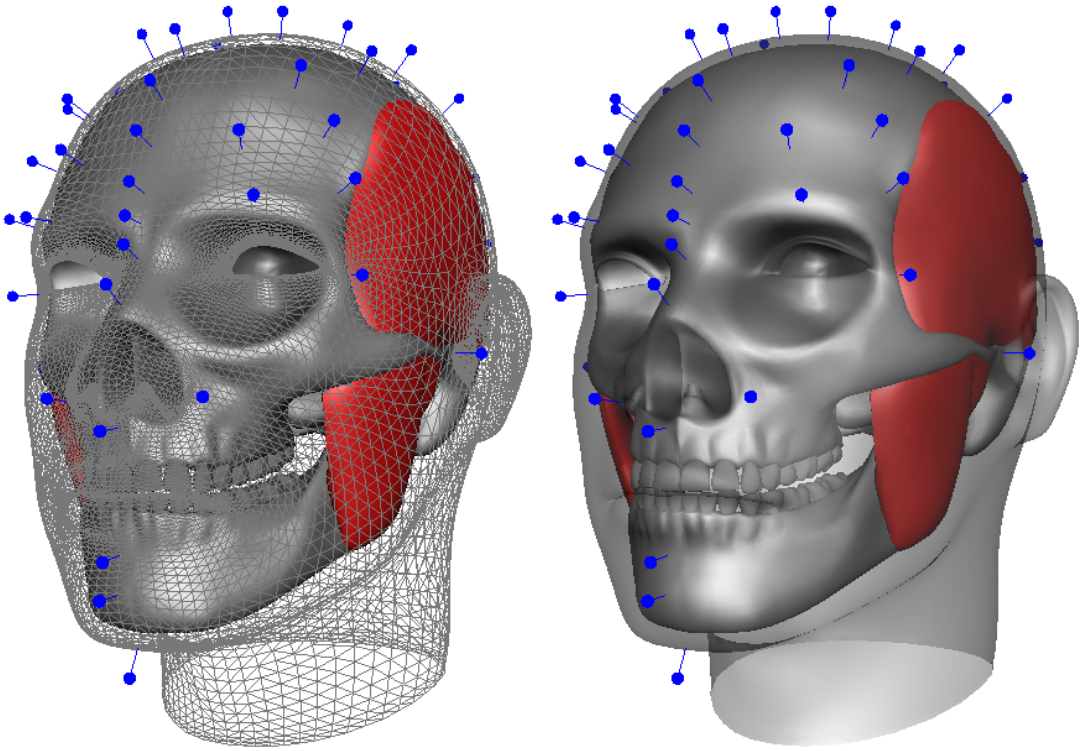
1. Because inaccurately placed landmarks adversely affect the quality of the fitting, only those cephalometric landmarks that can be reliably located on the face must be used in constructing the thin-plate spline deformers. At the present, the fitting is validated by visual inspection and comparison with references and descriptions sourced from anatomy texts.

2. In order to improve landmark coverage in the cranial region by the set of the original cephalometric landmarks, and ultimately the quality of the fitting, any number of sliding semilandmarks can be introduced wherever the skull and head surface are at a constant offset from each other. However, craniometric semilandmarks that slide too far (e.g. deep into the temporal region) and therefore fail to maintain the constant offset from the head should not contribute to the fitting process.

3. If a further improvement in the fit is desired, a secondary refitting should be performed by aligning on the gradient and tangent vectors at a subset of landmarks where the tangent planes of the skull and head are expected to match each other, e.g. on the cranial surface, excluding the deep temporal area, and at the nasal region.

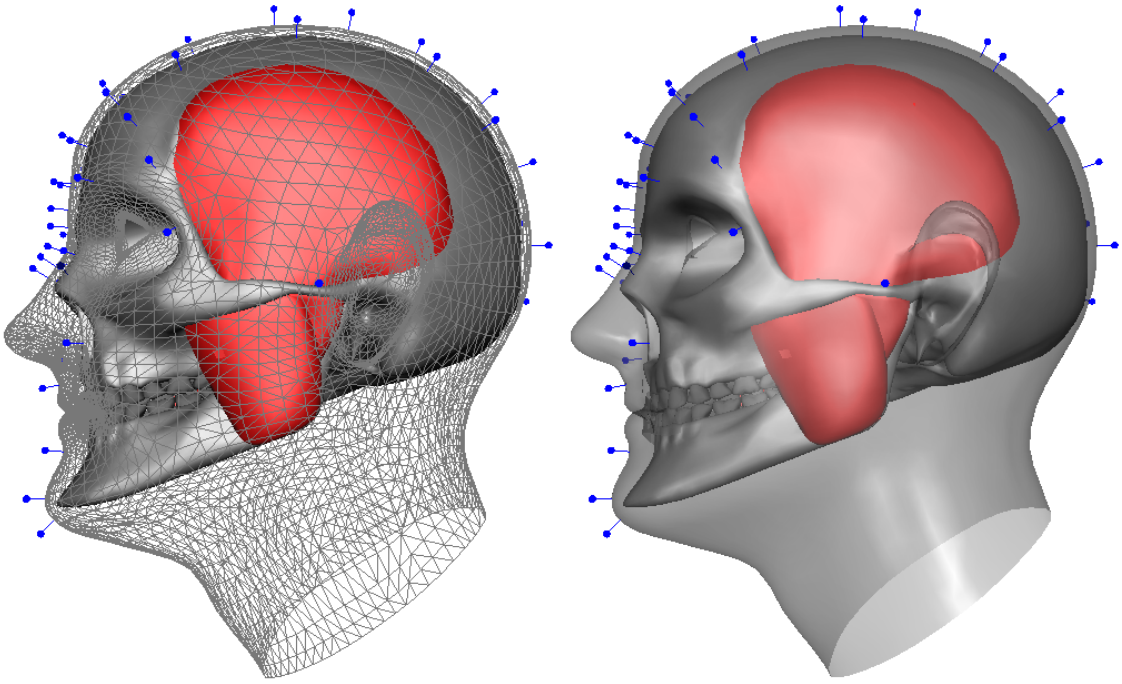


(a) Side view

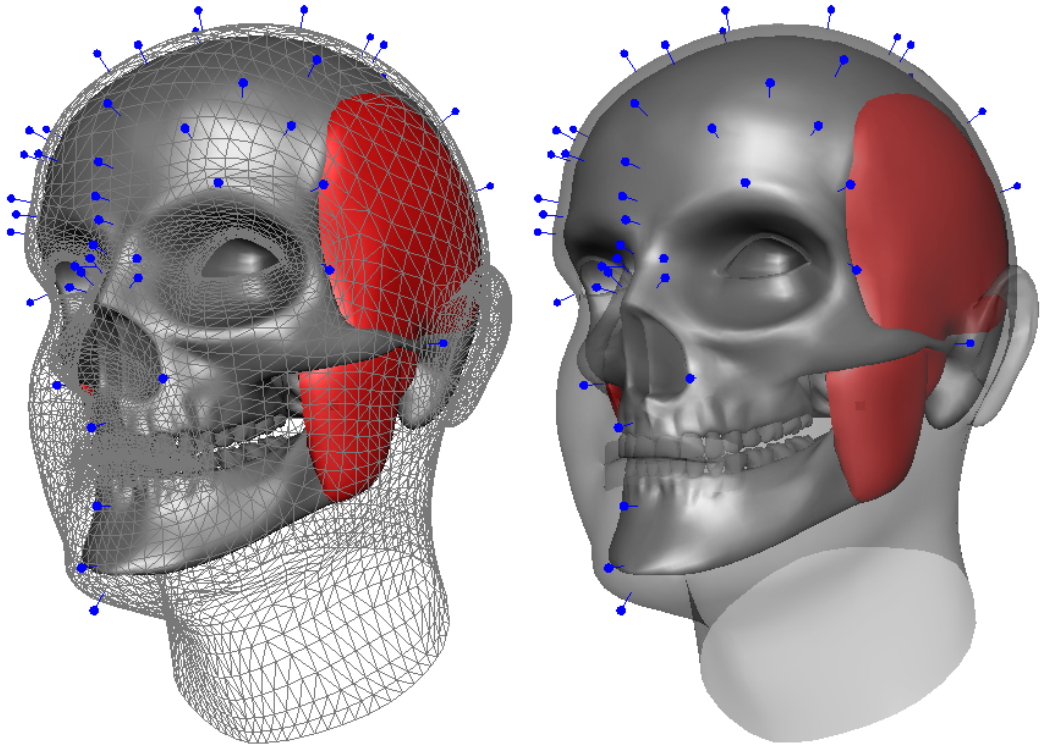


(b) Oblique view

Figure 6.8 Secondary fitting of generic skull to African head model, using landmark normals.

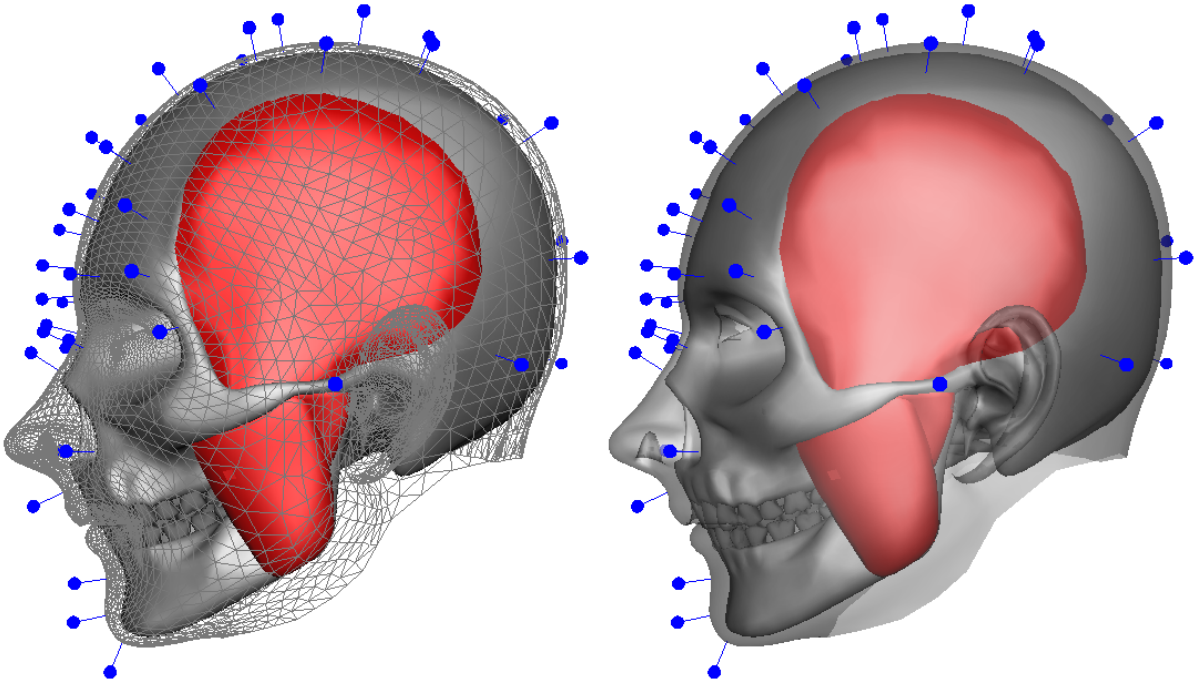


(a) Side view

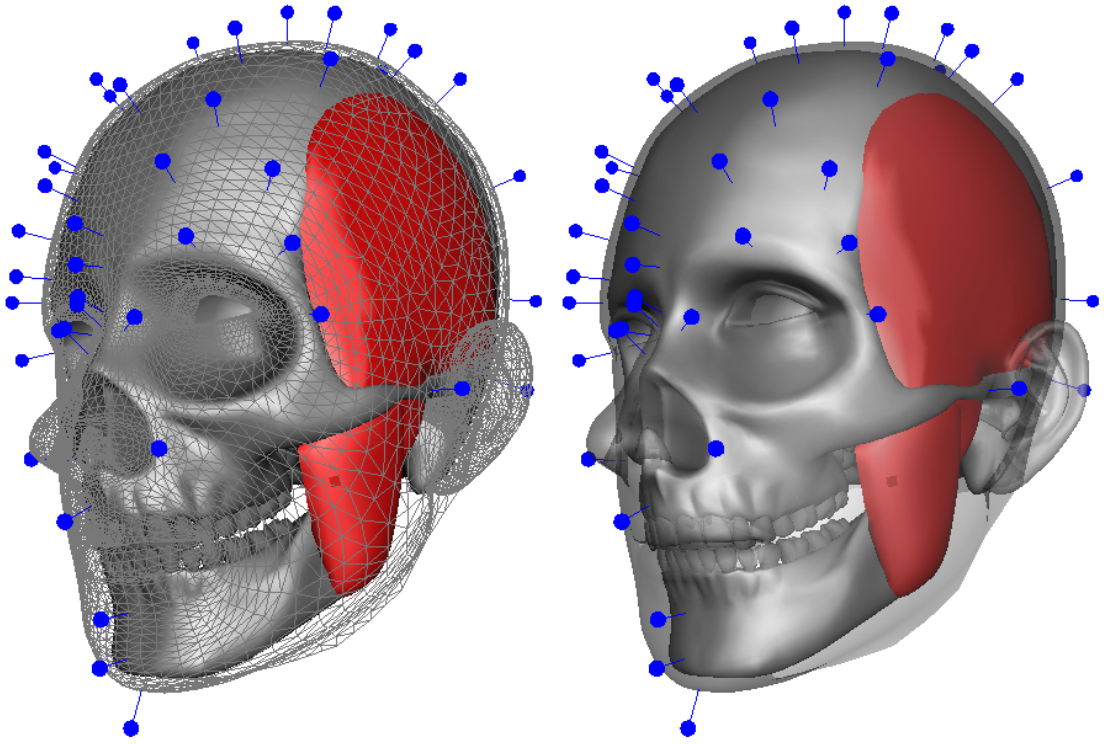


(b) Oblique view

Figure 6.9 Secondary fitting of generic skull to European head model, using landmark normals.



(a) Side view



(b) Oblique view

Figure 6.10 Secondary fitting of generic skull to MakeHuman head model, using landmark normals.

6.2.1 *Future work*

Possible areas of improvement to the overall process include: the automation of the landmark placement process (currently done manually), an investigation into the optimal number and distribution of landmarks for any given pair of models, and the development of a metric for evaluating the "goodness of fit" of any given deformation. In addition uncertainties in the placement of landmarks can be brought to bear on the fitting process by taking to account user-specified error ellipses as described by Rohr (2001). This technique also allows landmark localization errors to be calculated.

Furthermore, as a means of objectively assessing the quality of the above fitting procedure, a copy of the generic skull fitted to a head model obtained from medical imaging data, e.g. the Visible Human Project Ackerman (1998); Spitzer *et al.* (1996), can be compared with a skull model obtained from the same medical images as the head. Alternatively or in addition, the fitted skull can be validated by a forensic scientist or plastic surgeon.

Work should also be done on fitting generic skulls to head models with somewhat extreme head forms, face types and facial profiles such as those catalogued in 18th century works on physiognomy[6] (see Chapters 10 and 11 of Liggett (1974)). In addition, the use of alternative skull models – discussed on 111, should be investigated for subjects such as infants and primates, whose skulls differ from that of the adult human, to various extents.

Lastly, in addition to the temporalis and masseter, a pre-rigged model of the tongue can also be added to the generic skull package and fitted along with the skull to any given head model. This fitting process must preserve the basic shape of the tongue and the quality of the rig, as determined by the range of the various motions of the tongue as well as the extent to which it contacts and interacts with the lips, cheeks and oral cavity of any head model to which it is fitted. Alternatively, an unrigged model of the tongue can be added to the generic skull package and procedurally rigged after fitting. Another alternative option would be to procedurally model and rig the tongue after the skull is fitted to a head model. These three options should be investigated with a view toward determining which is the most effective.

Fitting a generic skull to a head model supplied with teeth

Although none of the head models used in this work were supplied with teeth, in some cases, (the shape, size and number of) teeth are an essential to defining a 3D character, and therefore must be supplied with the head model. In such cases, the generic skull should be made to fit not just the head, but the custom-modeled teeth supplied with it. To this end, cephalometric landmarks having zero depth values should be placed on any subset of the custom-modeled teeth (on the head model), preferably at the points

6 A discredited science founded on the assumption that face shapes are indicative of personality and temperament.

where each tooth meets the fleshy gum. Accordingly, a corresponding set of craniometric landmarks must be placed on the teeth supplied with the generic skull, in order to ensure that (post-deformation) the generic skull fits around the row of custom-modeled teeth supplied with the head model. (Each cephalometric landmark used in this subprocess is given a zero depth value because there is no soft tissue offset from its equivalent craniometric landmark.) In this scenario, the teeth supplied with the generic skull must not form part of the output 3D mesh. Furthermore, it is expected that the quality of such a specialized fit should improve with the number of landmarks used.

And I will lay sinews upon you, and will bring up flesh upon you, and cover you with skin — Ezekiel 37:6

7

CONSTRUCTING FACIAL MUSCLES AND THE SUPERFICIAL MUSCULOAPONEUROTIC SYSTEM (SMAS)

This chapter outlines a series of methods for constructing the muscles of facial expression and other soft tissues in the void between a given head model and a generic skull fitted to it (see chapter 6). First, the superficial musculoaponeurotic system (SMAS), a much-neglected but vital anatomical structure described in Section 3.4.3, is modeled as a variational implicit or radial basis function (RBF) surface, using constraints sampled from the head model and the generic skull fitted to it. Second, the fibres of the facial mimic muscle system, described in Section 3.2, are modeled as boundary-value straightest geodesics (refer to Section 5) on the SMAS, in contrast to the simple linear muscle vectors that appear in previous publications, for example Waters (1987). Furthermore, unlike other advanced physically-based FA techniques, for example Sifakis *et al.* (2005) and Barbarino *et al.* (2008), the technique developed in this chapter neither requires nor is constrained by medical imaging data. Instead, the approach taken is to automatically generate anatomically-plausible information for any given head model, based on its facial features and expert knowledge derived from recent advances in the field of facial plastic and reconstructive surgery. This approach promises to take much of the guesswork out of facial rigging by accurately computing the unique pattern of facial expressions, e.g. smiles or frowns, that can be produced by a virtual character given the constraints of its anatomy. Toward this end, as shown in Figure 7.2, the following modifications are made to the morphable generic skull model, hereafter referred to as the generic skull package:

i. Triangulated models of the left and right masseter and temporalis muscles and the temporalis fascia are added to the skull model.

ii. The anterior aspect of the skull is partitioned, with discrete curves, into various regions loosely based on the concept of facial aesthetic regions or units (see Section 3.7). As shown in Figure 7.1, these partitions include the frontal portion of the forehead unit and the dorsal subunit of the nose (coalesced) and the temporal region bounded inferiorly by the zygoma. The upper and lower extents of the buccal cavity are also marked out (for later use), as well as the lower border of the mandible and the anterior section of the masseter muscles.

iii. The skull model is UV-unwrapped and textured with a multilayer image indicating, wherever possible, the origins and insertions of naturally-occurring and/or imaginary muscles. Muscles such as the zygomaticus major and minor, whose insertions cannot be defined in relation to the surface of the skull, are defined interactively as described in Section 7.2. This approach facilitates the incorporation

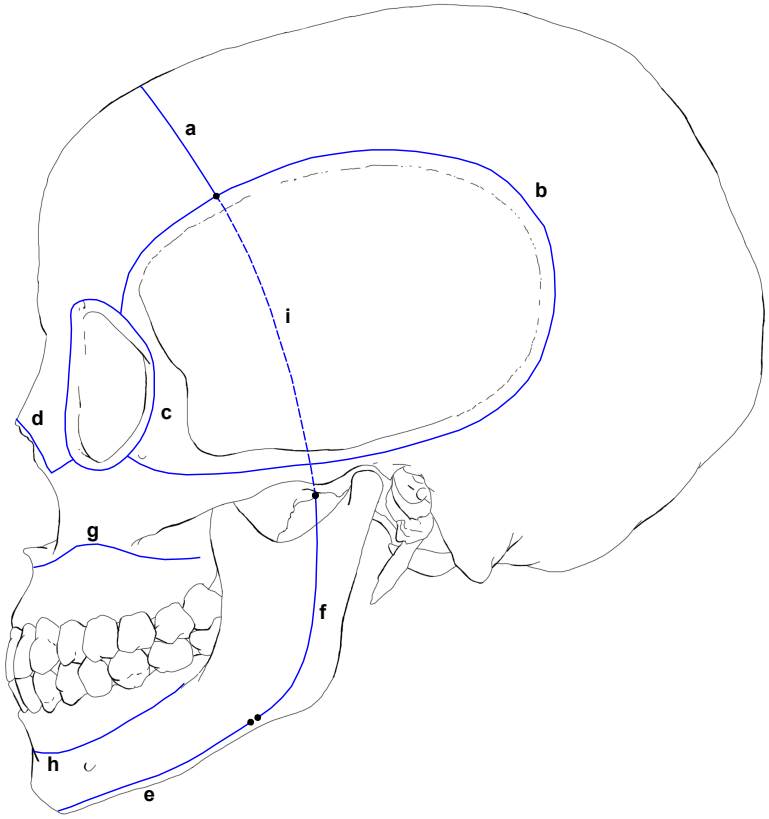


Figure 7.1 Discrete curves a – f specified on the generic skull package. The curves a, c and d partition the union of the frontal portion of the forehead unit and the dorsal subunit of the nose, while b and c partition the temporal unit. The curves g and h identify the upper and lower extents of the buccal cavity (for use in later work), e marks out the lower border of the mandible and f is specified on the masseter muscle (not shown). (The curve i is not specified as on the generic skull.)

of naturally-occurring or artistically motivated variations into the mimic muscle systems generated.

All of the added structures described above are "morphable" and are transferred along with the skull to the head models by the skull fitting process, as indicated in Figures 6.8, 6.9 and 6.10, for the masseter and temporalis muscles.

7.1 SMAS CONSTRUCTION

The SMAS is modeled as a variational implicit or RBF surface. This implicit surface is constructed using the Hermite formulation (Section 4.2.3) based on point and normal constraints sampled at the following locations, as shown in Figure 7.2(c):

i. the vertices contained in the coalesced forehead aesthetic unit and dorsal subunit of the nasal region, on the skull. These vertices are selected automatically.

ii. the vertices of the temporalis fascia – also selected automatically.

iii. a set of landmarks manually placed on each masseter.

iv. all of the points that define the discrete curves (show in Figure 7.1) on the skull, with the exception of those that identify the upper and lower extents of the buccal cavity. These vertices are selected automatically.

v. a set of landmarks manually placed just above the nostrils as well as the upper and lower lip and chin of the given head model.

vi. a set of landmarks manually placed along the zygomatic arch, lateral orbital rim and at the inner corner of each eye, on the skull.

All of the point constraints are offset by a small distance representing the distance of the SMAS from the respective surfaces that they lie on, and the normal constraints are taken as the surface normals of the respective surfaces that the point constraints lie on. Point constraints derived from the head are offset into the head model, while those derived from the skull package are away from it.

The resulting dense system of linear equations (see Equations 4.51,4.52 and 4.53), is easily constructed and solved in a few minutes with the LAPACK[1] and GotoBLAS[2] on an off the shelf PC. The implicit surface is triangulated using the marching triangles method Hartmann (1998), instead of the better known and faster marching cubes algorithm Lorensen and Cline (1987), because the former technique explicitly supports the creation of holes or boundaries on the resulting surface and produces a more uniform or coherent triangulation. Prior to triangulation, four closed discrete curves, representing both eyes, the nose and the mouth, were created in front of the face model and projected onto the implicit surface in order to form four internal boundaries of the triangulation as shown in Figure 7.2(d). The outermost ring of discrete curves on the generic skull are also projected to the implicit surface and linked to form an external boundary for the triangulation as follows. First, the curves a, f and e (see Figure 7.1) are projected to the surface of the SMAS. Thereafter the projections of a and f on the SMAS are connected by a straightest geodesic, i. The small gap between e and f on the SMAS is also bridged by a straightest geodesic, in order to create a ring of connected curves on the SMAS.

7.1.1 *Fast discretization of the SMAS by accelerating the marching triangles algorithm*

Similar to Cermak and Skala (2002), in order to avoid the excessive distance checks performed at each step of the marching triangles algorithm, all boundary points, initial or

1 Linear Algebra Package – http://www.netlib.org/lapack/
2 http://www.tacc.utexas.edu/tacc-projects/gotoblas2/

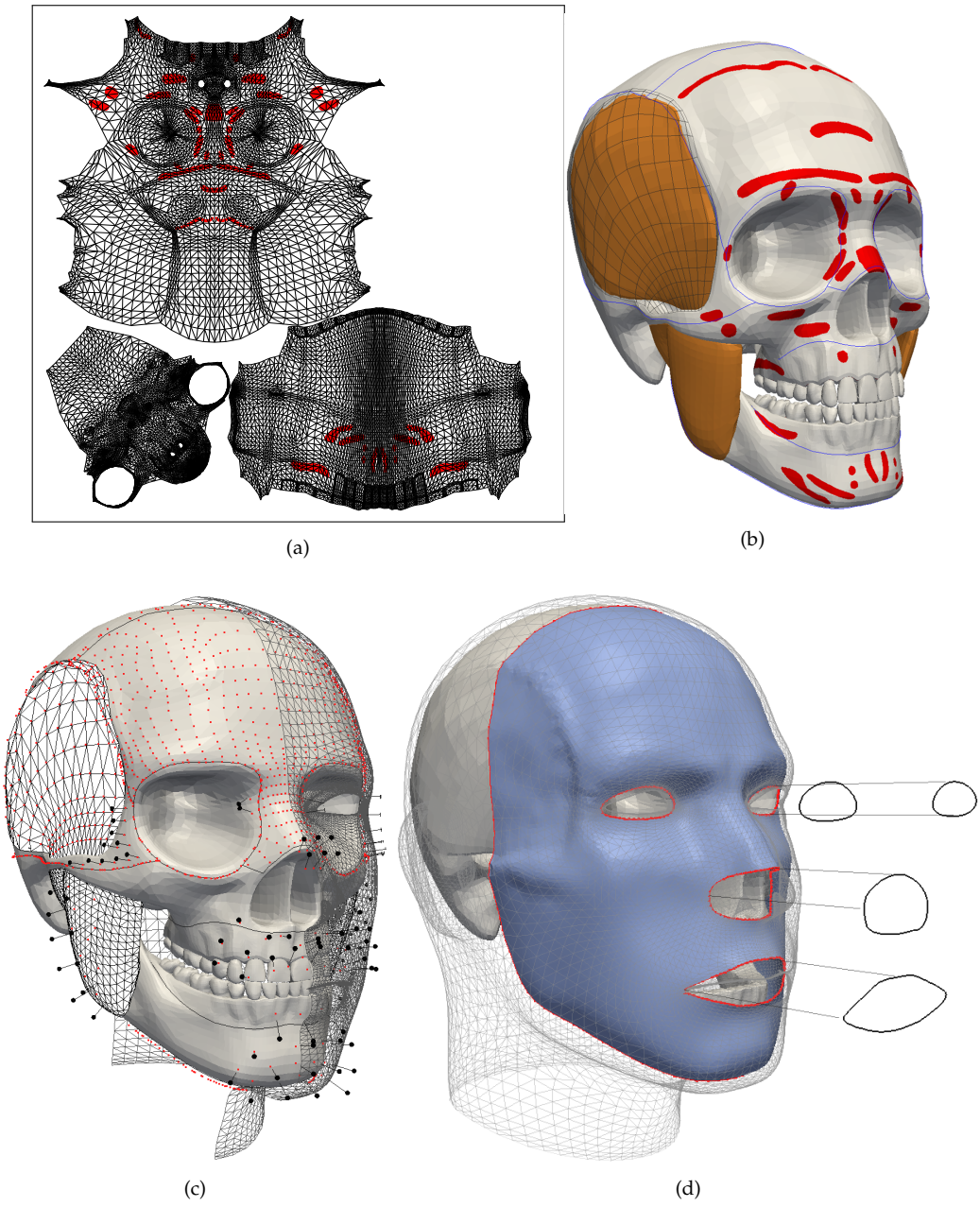


Figure 7.2 (a): UV-unwrapped mesh of generic skull overlaying an image showing the muscle origins and insertions (red). (b): generic skull textured with image indicating muscle attachment regions (red), temporalis and masseter muscles (brown), and aesthetic region boundaries (blue). (c): point constraints (red dots) used to construct variational implicit surface representation of SMAS. Black dots are manually located constraints. (d): Projecting four closed curves representing the eyes, nose and mouth to the surface of the SMAS (blue).

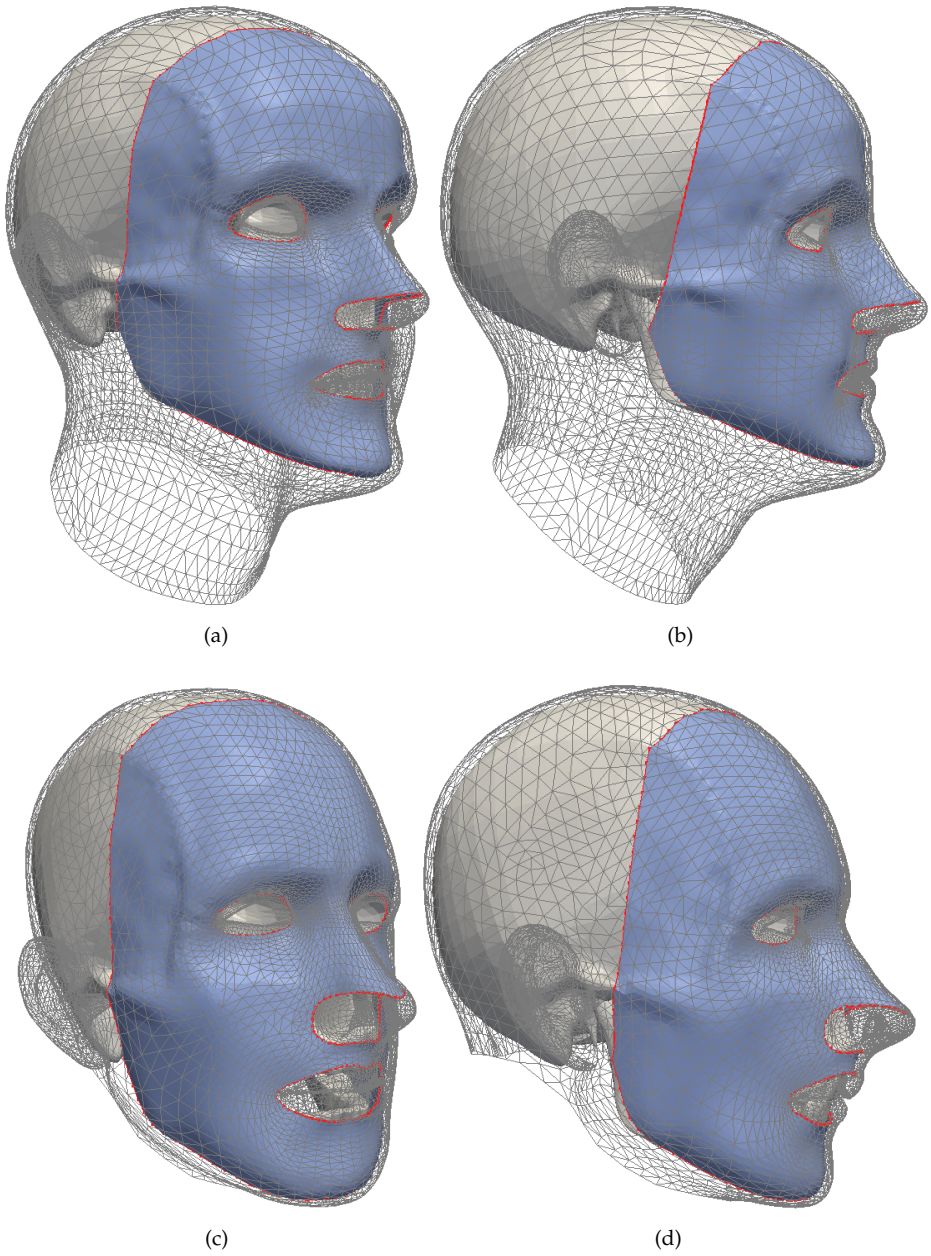
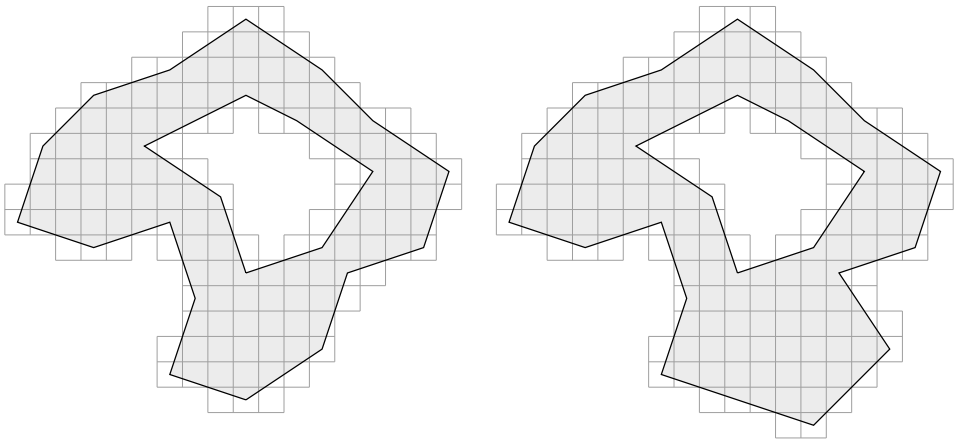


Figure 7.3 SMAS constructed for the European and MakeHuman head models.

otherwise, are embedded in dynamically created cube-like cells whose dimensions are a factor of (for example, two times) the triangle length (see Figure 7.4). However, in Cermak and Skala (2002) the entire space or volume thought to be occupied by implicit surface is subdivided into cells. Unfortunately, this space or volume is difficult to estimate for all but simple surfaces whose sizes or dimensions are known. Furthermore, by subdividing an entire volume, this approach creates cells unoccupied by face vertices of the object, for example in its interior. In the current approach however a cell is only created for a



(a) Earlier in the discretization process (b) Later in the discretization process

Figure 7.4 Snapshots of the discretization of a hypothetical implicit surface by the marching triangles method. Black lines show the current extent of the triangulation process, and the gray shaded region shows the output discretized surface. Gray boxes are the dynamically created cells that are used to perform distance checks during the marching triangles algorithm. Note that no cells are created beyond the extents of the discretized surface, e.g. where the surface contains holes.

(newly created) face vertex or front point that is not enclosed by an existing cell. The factor (based on the triangle length) is chosen such that the nearest front point to the active front point can be quickly found in the cell to which it belongs or any of its 26 neighboring cells[3]. However, the factor must not be too large or too small as to result in too many or too few front points in the cells.

The output of marching triangles algorithm is unchanged by the use of the dynamically created grid of cells. The sole purpose of this structure is to speedup the algorithm by avoiding needless and time consuming distance checks with non-viable front points that are too far from the current front point $\mathbf{p}$, and therefore do not belong to the cell to which $\mathbf{p}$ belongs or any of its 26 neighboring cells. This is why the cell size must not be too large as to contain too many front points that are too far from each other or have an unfeasibly large neighborhood also containing too many front points that are too far from each other. Conversely, the cell size must not be too small as to cull viable front points, by not assigning them to the cell to which $\mathbf{p}$ belongs or any of its 26 neighboring cells.

The surface shown in Figure 7.2(d) is obtained from an implicit surface constructed from 1494 surface and normal constraints, sampled at the prescribed locations, and solved using a Windows XP laptop computer with a 1.6GHz processor and 512Mb memory in

3 A cell in the centre of a $3 \times 3 \times 3$ block of cells has 26 neighbors.

3min 35secs. This implicit surface was discretized, using the marching triangles algorithm with the dynamic cell acceleration structure, into a 23,576 triangular mesh model of the SMAS, shown in Figure 7.2(d) in 5min 30secs.

7.2 MUSCLE CONSTRUCTION

The process of generating the fibres that constitute a muscle belly requires that the attachment regions (origin and insertion) of the muscle be specified as shown for example in Figure 7.5(a) so that the (joint) convex hull of both attachment regions can be computed, as shown in Figure 7.5(b). Thereafter, as shown in Figure 7.5(c), the lateral extents of the muscle are constructed as the mutual tangents of the convex hulls computed in the previous step. Finally, the muscle fibres are constructed by interpolating the mutual tangents, as shown in Figure 7.5(d)-(f).

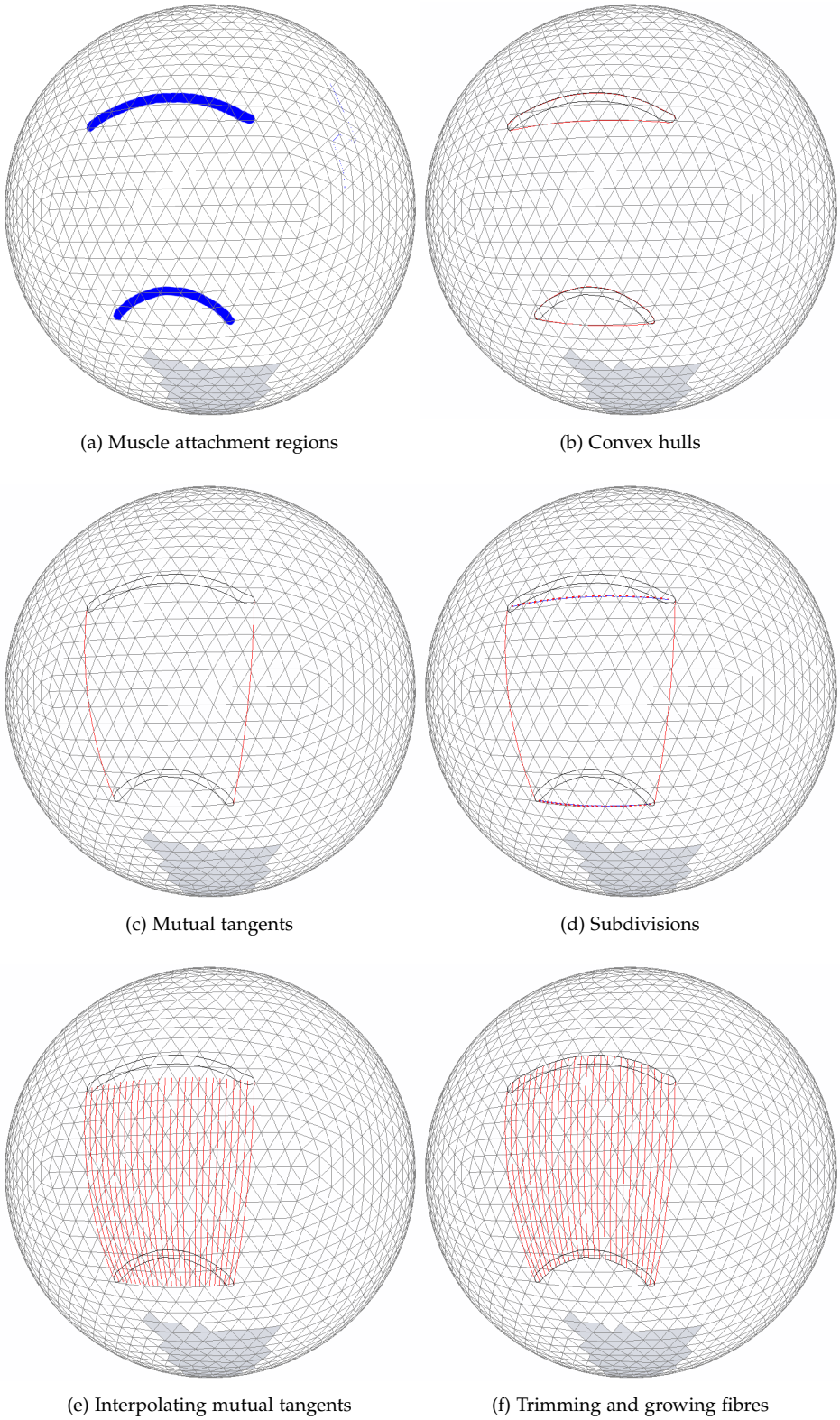
The remainder of this chapter describes each of the above steps in detail; nevertheless two points are worth emphasizing. First, the method requires that the concept of the convex hull be extended to non-planar discrete manifolds, of arbitrary connectivity and topology. Grima and Márquez (2001) previously explored this concept albeit for simple, analytic surfaces such as spheres, toruses, cubes and cylinders. Second, muscle fibres are constructed on the SMAS; however the toy example in Figure 7.5 makes no mention of this structure. Accordingly, the process begins by defining the muscle attachment regions on the SMAS.

7.2.1 *Defining muscle attachment (origin and insertion) regions on the SMAS*

As highlighted in the introduction, where possible the origin and insertion of each muscle is painted as a continuous region of a single colour on a separate bitmap image. This is done in order to simplify the task of identifying the various regions. However, in order to avoid complications associated with the management of multiple files, the attachment regions are painted on separate layers of a multilayer TIFF image. In either case, as shown in Figure 7.2(a), care is taken not to paint outside the various patches of the UV-unwrapped mesh representation of the generic skull.

Step 1: convert muscle attachment region image into a discrete polygon, in texture or UV space

As shown in Figure 7.6(a), the inner boundary of each painted region of an image, or image layer, is traced counterclockwise in order to obtain the chain code or sequence of steps to be taken in eight possible directions around the image boundary Sonka *et al.* (2007) (pp. 129–131). The chain code remains fairly constant over short intervals, and changes indicate variations in the gradient of the border pixels. The centre of each pixel where the chain code changes is taken as a turning point $\mathbf{p}$ on the boundary of the



(a) Muscle attachment regions

(b) Convex hulls

(c) Mutual tangents

(d) Subdivisions

(e) Interpolating mutual tangents

(f) Trimming and growing fibres

Figure 7.5 Toy example, illustrating stepwise process of muscle construction.

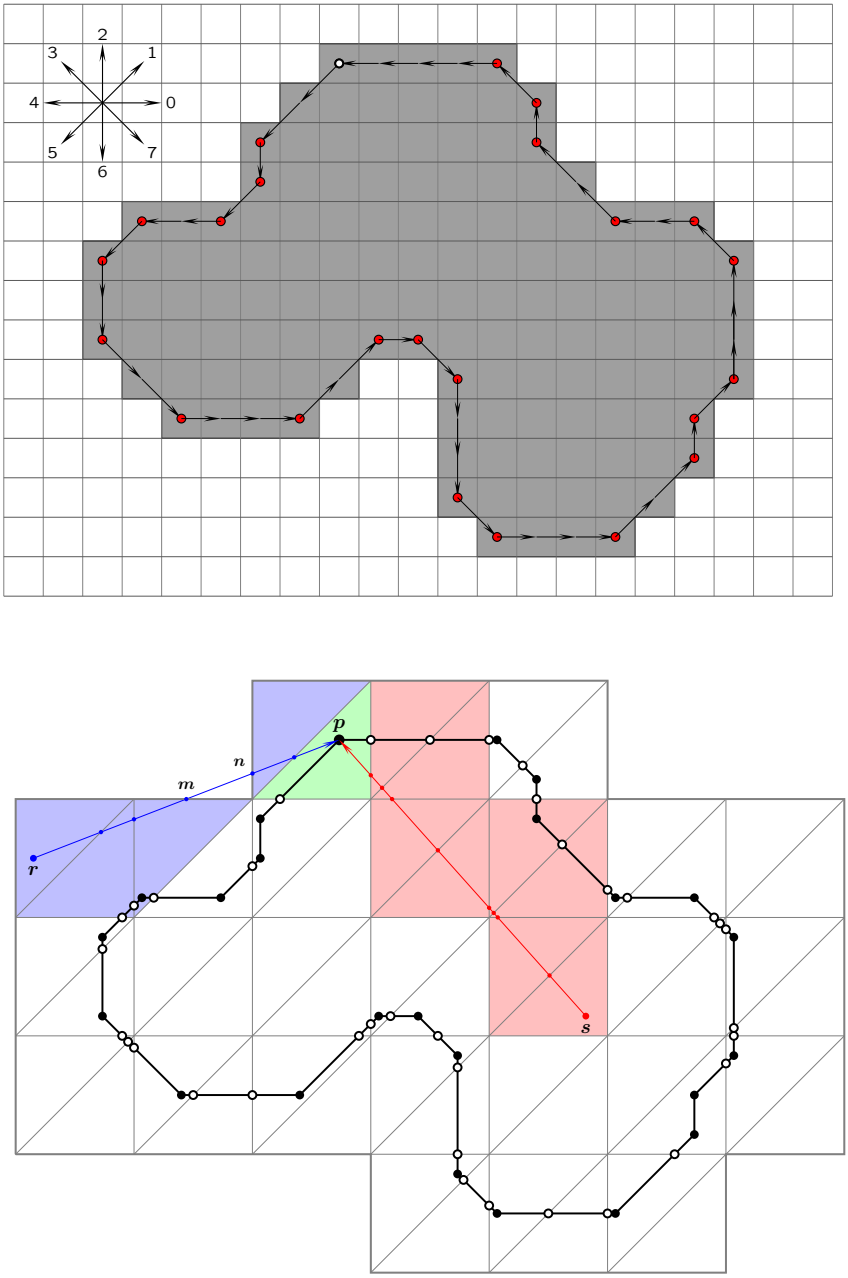


Figure 7.6 (a) top: chain code `55,6,5,44,5,66,77,000,11,0,7,666,7,000,11,2,1-,222,3,44,33,2,3,4444` (commas emphasize changes in direction) indicating the sequence of directions of the steps taken in eight possible directions `0-7` around the inner boundary of the above image, starting from an initial point (hollow dot). (b) bottom: finding the triangle (green) in which an arbitrarily chosen initial point **p** lies. This is done by following the line segment from the centroid, **r** or **s**, of an arbitrarily chosen triangle to **p**. Note that whereas a section of the blue path **pr** falls outside the UV patch, the red path **ps** lies completely inside it. The hollow dots, also considered as boundary points, are the intersections of the discrete boundary curve extracted from the chain code and the edges of the UV patch.

closed discrete curve representing the border of the painted region in texture space. The coordinates of $\mathbf{p}$ are expressed as UV coordinates and are calculated as follows:

$$u = \frac{x + 0.5}{W} \quad \text{and} \quad v = \frac{y + 0.5}{H}$$

where x and y are the pixel coordinates of $\mathbf{p}$ and W and H are the pixel width and height of the image. It bears repeating that at the end of this step, the muscle attachment region is a planar, discrete polygon in texture (UV) space.

Step 2: find the triangle corresponding to the start of the discrete polygon

As shown in Figure 7.2(a), the discrete polygon, hereafter referred to simply as the polygon, computed in the previous step is overlain with the UV-unwrapped mesh of the generic skull, with a view to finding the triangle on which each point on the polygon lies (in texture space and in three-dimensions). Therefore, the polygon can be thought of, and stored as, a sequence of discrete curves, consisting of lists of vertices, that lie on the same triangle face, as shown in Figure 7.8. This implicit association between (sections of) the polygon and the triangles of UV mesh is initiated by identifying the triangle face T_0 to which an arbitrarily chosen initial border vertex $\mathbf{p}_0$ belongs, and proceeding, by a simple method of line following, as described in the next step.

The triangle T_0 is found by randomly selecting a triangle T that is thought to lie in the same patch as $\mathbf{p}_0$ and following the line segment from the centroid of T to $\mathbf{p}_0$ as summarized in Algorithm 5, and illustrated by the red and blue line segments in Figure 7.6(b). However because the UV-unwrapped meshes typically consist of several patches the bounding rectangle of each patch is precomputed and T is randomly selected from the patch of the first bounding rectangle within which $\mathbf{p}_0$ lies. However, as shown in Figure 7.7, the wrong patch can be prematurely selected if $\mathbf{p}_0$ lies in the overlapping portion of two or more bounding rectangles. The other bounding rectangles within which $\mathbf{p}_0$ lies and their corresponding patches must therefore be considered if, during the course of the algorithm, it is found that the wrong bounding rectangle and patch were selected.

Step 3: intersect muscle attachment region polygon with UV mesh

Starting from an initial point $\mathbf{p}_0$, whose "container" triangle is known (from the previous step), and traveling counterclockwise, for reasons that will later become apparent, consider the next point $\mathbf{p}_1$, to the left of $\mathbf{p}_0$, on the boundary of the polygon, and note that $\mathbf{p}_1$ may either be in the same triangle T_0 as $\mathbf{p}_0$ or otherwise. If the latter is the case, a line segment l whose end points are $\mathbf{p}_0$ and $\mathbf{p}_1$ is created, and starting from T_0, the sequence of triangles and edge intersection points encountered while following l from the point $\mathbf{p}_0$ to the point $\mathbf{p}_1$ is obtained. Edge intersections are the points where l intersects the edges of the triangles that it traverses, and are shown as hollow dots in Figure 7.6(b). Although seemingly unnecessary, edge intersections are also stored or treated as points on the polygon boundary. The rationale for their inclusion will be supplied in the following section.

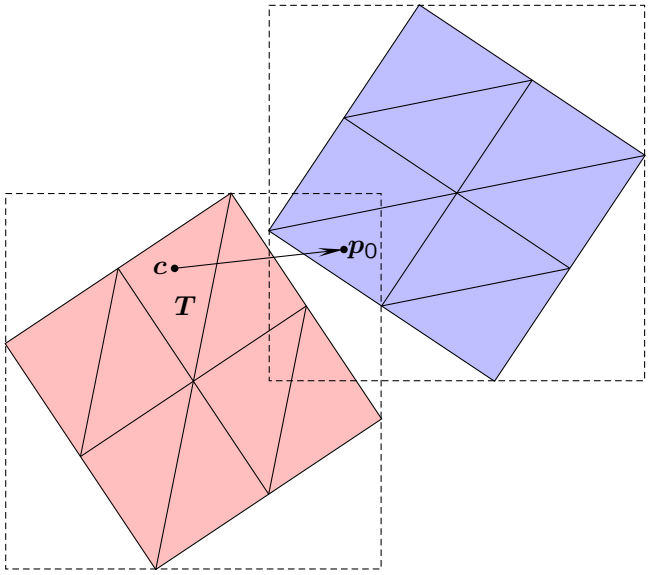


Figure 7.7 Example scenario in which the wrong patch can be wrongly selected because the initial point $\mathbf{p}_0$ lies within two (or more) bounding boxes. In this situation, if the (bounding box of the) red patch is tested first, any of its triangles, e.g. $\mathbf{T}$ will be selected, instead of the blue patch, and its centroid $\mathbf{c}$ chosen for the start of line following process.

Algorithm 5: FIND UV-SPACE TRIANGLE CORRESPONDING TO A GIVEN POINT P

Input: list of UV patches (uvPatchArray)
Data: current triangle (T) , current patch BBox (patchBBox) , line segment (L)
Data: previous edge (pe) , current point (cp) , boundary edges (boundaryEdges)

```
1  foreach patch in uvPatchArray do
2      patchBBox ← getPatchBBox( patch)
3      if p is in patchBBox then
4          boundaryEdges ← getPatchBoundaryEdges( patch)
5          T ← randomTriangleInPatch( patch)
6          c ← centroid( T)
7          construct L from p (start) to c (end)
8          pe ← 0
9          while p is not in T do
10             foreach edge e in T do
11                 if e ≠ pe then
12                     if e ∈ boundaryEdges then
13                         remove e from boundaryEdges              // do not test e
                           // intersected edges e*, intersection points p*
14                         [ e*, p* ] ← intersect( boundaryEdges, L)
15                         if e* is of zero length then
16                             exit while loop
17                         else
18                             e ← EdgeWithNearestIntersection( e*, p*)
19                             T ← BoundaryTriangle( e)
20                             pe ← e
21                         end
22                     else
23                         cp ← intersect( e, L)
24                         if cp is not equal to 0 then
25                             set L end to cp
26                             T ← GetNextTriangle( e, T)
27                             pe ← e
28                             exit (innermost) foreach loop
29                         end
30                     end
31                 end
32             end
33         end
34     end
35 end
```

T_0						T_1		T_2			
q_0	p_0	p_1	p_2	p_3	q_1	q_1	q_2	q_2	p_4	p_5	q_3

...

Figure 7.8 Discrete Curve on Mesh (DCoM) data structure: a list, associating groups of points that form a discrete curve on a mesh with the triangles that they traverse. Note that each group on the list begins and ends with an edge intersection point $\mathbf{q}_i$, and between these two points are any number of intermediate points. Furthermore, neighboring groups on the list share the same edge intersection point.

All pairs of (original points) $\mathbf{p}_i$ and $\mathbf{p}_{i+1}$ on the polygon are similarly "followed", in order to complete the process of overlaying the polygon and UV mesh, and each section of the polygon is associated with the triangle face that it traverses. Each section of the polygon starts and ends with an edge intersection point $\mathbf{q}$, while neighboring sections share an edge intersection point. As previously mentioned this information is stored in a DCoM data structure.

Step 4: map muscle attachment region from UV space to the generic skull

This is done by reversing the texture mapping process, noting that whereas ordinary texture mapping involves computing the UV coordinates corresponding to a 3D point, in the reverse process the 3D coordinates $\mathbf{P}$ that correspond to a given UV coordinate $\mathbf{p}'$, are determined using the formula:

$$\mathbf{P} = u.\mathbf{P}_1 + v.\mathbf{P}_2 + w.\mathbf{P}_3 \tag{7.1}$$

where u, v, w are barycentric coordinates of the point $\mathbf{p}'$ and $\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3$ are the vertices of the 3D triangle that corresponds to the UV triangle within which $\mathbf{p}'$ lies. This reverse texture mapping process maps the UV-space polygon to the surface of the 3D (generic skull) object. Because the term polygon often suggests a planar curve, the mapping of the UV-space to a non-planar surface is not referred to as a polygon but as the boundary curve (of a muscle attachment region). Nevertheless, such boundary curves are also treated and stored as a list of discrete curves, consisting of groups of points that lie on the same triangle.

As a result of this process, any two points on neighboring triangle faces, in texture space, almost never remain collinear with the edge (intersection) point that they create in texture space (step 3), when all three points are transformed to 3D. For example, in Figure 7.9, the (UV space) edge point e' is formed by the intersection of the line segment $c'd'$ with the triangle edge $S'Q'$, and the points c', e' and d' are collinear. However upon transformation to 3D the corresponding points c, e and d are generally not collinear, because the point e forms a kink, representing the distortion of the texture in 3D as

a result of the reverse mapping process. This is why edge intersection points were introduced in step 3.

However, it is essential to ensure that the triangle faces of the UV unwrapped and 3D (generic skull) meshes are both oriented counterclockwise. For example in Figure 7.9 the UV-space triangle faces $S'Q'T'$ and $S'Q'R'$ do not have the same orientations as the 3D-space triangle faces SQT and SQR to which they correspond. In this figure, because the bitmap image from which the polygon $a'f'b'c'e'd'$ was extracted was traced counterclockwise (correctly, as described in step 1), the sequence of UV-space boundary points $e'd'a'f'$ and $f'b'c'e'$ are ordered correctly (counterclockwise). However the corresponding sequence of 3D points $edaf$ and $fbce$ are ordered incorrectly (clockwise), because the 3D triangle face on which they lie has the opposite orientation to its equivalent in UV space.[4] But if in such cases it is assumed that the triangles of the 3D mesh are oriented correctly (counterclockwise), and all have outward facing normals, the only option is to conclude that the UV triangles are not oriented correctly.

Therefore, the orientations of all the triangles of the UV-unwrapped mesh and their 3d equivalents (on the generic skull) are compared. This can be done on a patch-by-patch basis if all triangles in a UV-mesh patch have the same orientation, noting also that muscle attachment regions do not cross patch boundaries. If there is a mismatch in orientations of the triangles the order of appearance of the points of the boundary curve is reversed. This requires a deep reversal of the DCoM, and is characterized by a reversal of the order of appearance of the triangles as well as the sequence of points in each triangle. For example in Figure 7.9 the orderings of the sequence of border points $edaf$ and $fbce$ are reversed to $fade$ and $ecbf$ respectively. Also, the order in which both sequences are listed in the DCoM is reversed, so that $ecbf$ appears before $fade$.

Step 5: project the muscle attachment region to the SMAS

This is based on the assumption that all muscles are lie on the SMAS, although, as noted in Section 3.4.3, this is only true for a subset of muscles. Therefore, after correcting the ordering of points that constitute the boundary curve (if necessary) these points are projected to the SMAS, in the order in which the points appear in the DCoM. The sequence of points projected on the SMAS is then traversed (in the order in which they were projected) and connected by straight paths (see Chapter 5) in order to create a boundary curve representation of the muscle attachment region on the SMAS. As in the previous two steps, the boundary curve on the SMAS is stored, in a DCoM data structure, as a list of pairs of triangles and the sequence of points that traverse it.

Three possible cases arise when connecting any two consecutive points on the SMAS. In the first case, both points lie on the same triangle face and the desired path is the straight line connecting both points. In the second case, both points lie on neighboring

4 The terms clockwise and counterclockwise used to describe the ordering of points is only meaningful in this very simple, convex, sequence of points. Generally, the sequence of points on a triangle face is not convex and the adjectives clockwise and counterclockwise are inappropriate for describing the ordering of points.

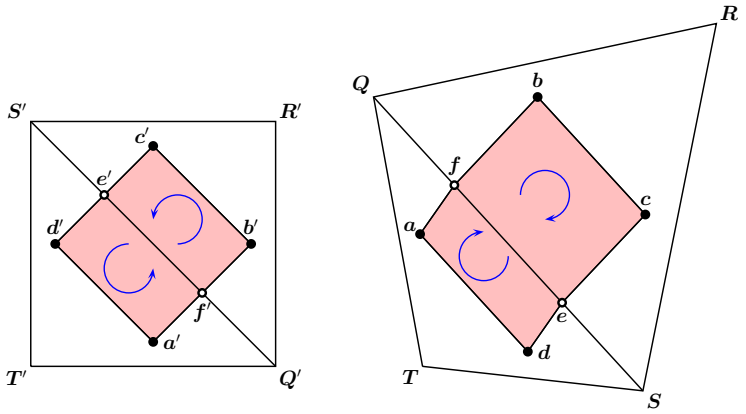


Figure 7.9 The 3D surface points a, b, c, d, e and f that correspond to the border and intersection points a′, b′, c′, d′, e′ and f′ situated on the boundary of the painted region (pink) of a bitmap image. Note that, the triangle faces Q′T′S′ and R′Q′S′ are in UV space, while triangle faces QST and RSQ are in 3D space. (Arrows indicate the sequence of the border points, and not the orientations of the triangles.)

triangle faces and a straight path can be constructed by rotating the second triangle to the plane of the first. A point is then created which lies on the shared edge of both triangles and is collinear with both points. In the third case, both points lie on separate, non-adjacent triangles, and a geodesic is constructed between both points. In the second and third cases the paths created cross triangle edges and thus create edge intersections.

For muscles such as the zygomaticus major and minor, whose insertions cannot be defined relative to the skull, i.e. painted on the UV-unwrapped mesh, discrete polygons are created, in space, lateral to the face and projected to the SMAS, similar to what is done in Figure 7.2(d). As before, the projected points are traversed counterclockwise and intersected with the edges of triangles of the SMAS mesh in order to create the boundary curve of the projected polygon.

7.2.2 *Computing the convex hulls of muscle attachment regions, on the SMAS*

As shown in Figure 7.5, the penultimate step in the process of muscle construction is to model the lateral extents of each muscle as the mutual tangents of the convex hulls of a pair of muscle attachment (origin and insertion) regions. These mutual tangents are a by-product of the divide-and-conquer technique for constructing the joint convex hull of two convex hulls. O'Rourke (1998) (pp. 91 - 95) describes this algorithm for problems in two dimensional planes and in three dimensions. In the following section (7.2.3) this technique is extended to discrete manifolds. However the said extension still requires two convex hulls as its inputs. Accordingly, the convex hulls of all boundary curves (on the SMAS) must be computed. Because each convex hull is derived from a boundary

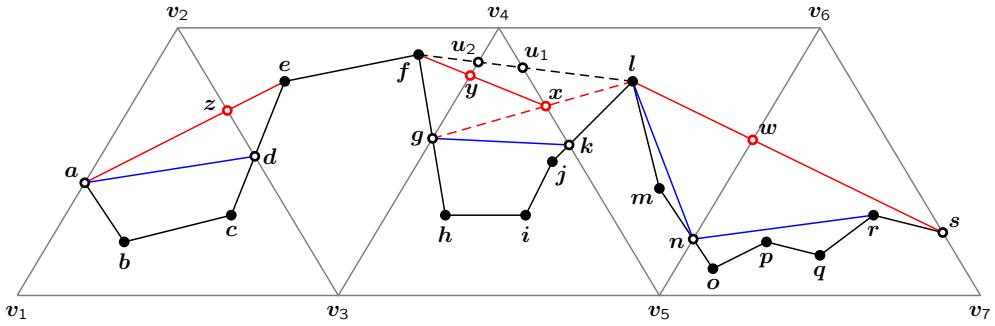


Figure 7.10 Hierarchical removal of boundary points from a candidate convex set. The black curve represents the input boundary or starting hull, while the blue and red lines highlight changes to the hull after steps 1 and 2 respectively.

curve, a copy of each boundary curve is made and the points that constitute the duplicate curve are hierarchically deleted (or pruned) if they do not belong to the convex set. The original boundary curve is not altered as it is used in the final step of muscle construction (see Section 7.2.4), in order to determine the lengths of muscle fibres. Hereafter, in this section and the next, the term boundary curve will refer to the duplicated curve.

Computing the convex hull of a boundary curve is a three-step process, at the start of which it is assumed that all points on the boundary curve are candidates for the convex set. At each step, any boundary point that forms an angle less than π with their neighbors is removed from the "candidate set". Such points are referred to as concave boundary points. Pruning is performed repeatedly until the candidate set contains no concave boundary points. This set of remaining candidate points is the convex set. The (points in the) convex set form the edges of the convex hull. In a planar, two dimensional domain, these edges are straight lines, however on a non-planar discrete manifold the edges of the hull are geodesics, many of which connect points that do not lie in the same triangle. As such, the current view of a boundary curve, as lists of points ordered according to triangle faces that they traverse, is no longer sufficient, especially in the second and third steps (see below) where the edges of the candidate set commonly connect points that do not lie in the same triangle. Nevertheless, the prevailing concept of a boundary curve is employed in the first step described below.

Essentially, in the course of this algorithm the representation of the boundary curve changes from a DCoM to a sequence of geodesics connecting the candidate set, and ultimately the convex set.

Step 1: per triangle pruning

The objective of this step is to remove concave intermediate boundary points from the candidate set on a per triangle basis – bearing in mind that in the prevailing representation of a boundary curve (see Figure 7.8), each list of points in a triangle has an edge intersection point at its start and end, and between these points are any number of

intermediate points. In practice, an intermediate point $\mathbf{p}_i$, having the adjacent points $\mathbf{p}_{i-1}$ and $\mathbf{p}_{i+1}$, is removed from the candidate set if the cross product of the vectors $\mathbf{p}_i\mathbf{p}_{i-1}$ and $\mathbf{p}_i\mathbf{p}_{i+1}$ points in the same direction as the normal of the face on which all three points lie. For example, in Figure 7.10, the intermediate vertex c, in triangle $v_1v_3v_2$, is removed from the candidate set because the cross product of the vectors $\mathbf{cd}$ and $\mathbf{cb}$ point in the same direction as the outward facing normal. In other words the point $\mathbf{c}$ forms an angle less than π with the neighboring points b and d.

If however the order of the boundary points is reversed, the vector $\mathbf{p}_i\mathbf{p}_{i-1} \times \mathbf{p}_i\mathbf{p}_{i+1}$ will point in the opposite direction as the face normal, and the pruning process produces a concave hull instead of a convex hull. This is why it is essential that corresponding UV space and 3D triangles are both oriented clockwise, so that the (sequence of the) boundary points that they embed is ordered correctly. This process is repeated until no intermediate concave points remain in the triangle. For example, in Figure 7.10, the candidate set of the (input or initial) sequence of points in the triangle $v_5v_7v_6$, after per-triangle pruning, are the points $\mathbf{s}$, $\mathbf{r}$ and $\mathbf{n}$, while the points $\mathbf{q}$, $\mathbf{p}$ and $\mathbf{o}$ have been removed from the candidate set. Overall, the number of points in the candidate set of the discrete curve, spanning the five triangles, is reduced from 19 to 10, as a result of this step.

Step 2: pruning per pair of neighboring triangles

This step is essentially the same as the previous, but with one apparent difference; namely that pruning of the candidate set is performed over pairs of consecutive triangles that sections of the boundary curve traverses, and not per-triangle. In other words, for each pair of neighboring triangles T_i and T_{i+1} concave intermediate boundary points are iteratively removed from the candidate set, until no such points remain in either triangle. However, prior to checking for concave intermediate points, the triangle T_{i+1} and the candidate points contained in it are first rotated to the plane of T_i. In addition, the edge intersection point shared by T_i and T_{i+1} is treated as an intermediate point, and thus can be removed from the candidate set. This point does not need to be rotated to the plane of T_i as it is on the edge of both triangles.

This process considerably reduces the size of the candidate set, for example from 10 to 5 points in Figure 7.10, and often produces new edge intersection points, for example $\boldsymbol{w}$, $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$ in Figure 7.10. These new edge points belong to the geodesics that connect the points in the candidate set, but are not added to it, because they are not on part of the initial or input boundary curve. Note that repeatedly applying this step on the pairs of triangles $v_4v_5v_6$ and $v_3v_5v_4$, and $v_2v_3v_4$ and $v_3v_5v_4$ cannot remove concavities at the edges v_3v_4 and v_4v_5, e.g. the point $\mathbf{x}$. This situation is remedied in the next step.

Step 3: pruning concave "meeting" points in a sequence of geodesic edges

Whereas in steps one and two, pruning of concave points was performed within individual triangles and pairs of consecutive triangles respectively, in this third step pruning takes place regardless of the number of triangles traversed by two or more geodesics that constitute the edges of the candidate set. This is required because not all pair of consecutive geodesics form a convex point where they meet. In fact, each geodesic in a set of N geodesic edges can form a concave point with its neighboring edge. In this case the concave (meeting) points are removed from the candidate set and a new geodesic is constructed between the first point of the first edge and the last point of the last edge. For example in Figure 7.10 the meeting point $\mathbf{x}$ of the edges $\mathbf{lx}$ and $\mathbf{xf}$ is concave. Therefore the geodesic $\mathbf{lu}_1\mathbf{u}_2\mathbf{f}$ is used to connect the start point $\mathbf{l}$ of the first edge with the end point $\mathbf{f}$ of the second edge. As before, this process is repeated until the meeting points of all geodesics connecting the candidate set are convex. The geodesics left over from this process form a convex hull, while the start (or end) points of the geodesics form the convex set.

7.2.3 *Computing mutual tangents*

As highlighted at the start of the previous section, the mutual tangents of a pair of convex hulls are computed by a generalization of the divide-and-conquer method (see O'Rourke (1998)) to discrete manifolds. This is done in two major steps, as follows.

Step 1: connect both convex sets by a geodesic edge

While the objective of this first step is straightforward, its execution is complicated by the restriction that the geodesic edge must not penetrate either hull, as shown for example in Figure 7.11(a) where the path $\mathbf{gn}$ penetrates both hulls and $\mathbf{iv}$ penetrates the lower hull. In the two-dimensional version of the algorithm described by O'Rourke (1998), the end points of the corresponding edge are the rightmost and leftmost points of either hull. However on a non-planar manifold, such notions are unreliable. As such this technique will not be used.

Instead, a geodesic $\mathcal{G}$ is traced between two randomly-selected points on either hull, and checked for intersections with both hulls. In the ideal case, $\mathcal{G}$ intersects neither hull, for example the path $\mathbf{cr}$ in Figure 7.11(a). If however $\mathcal{G}$ intersects an edge $\mathcal{E}$ of either hull at the point $\mathbf{p}$, $\mathcal{G}$ is recomputed from (or to) the end point of $\mathcal{E}$ that is closest to point $\mathbf{p}$. For example in Figure 7.11(a) the path $\mathbf{gn}$ intersects the edges $\mathbf{st}$ and $\mathbf{ab}$ of the upper and lower hulls, at the points $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively. And because the end points $\mathbf{s}$ and $\mathbf{a}$ (of the edges $\mathbf{st}$ and $\mathbf{ab}$) are closest to $\mathbf{p}_1$ and $\mathbf{p}_2$ respectively, the path connecting both hulls is the line segment $\mathbf{sa}$. Also in Figure 7.11(a) the path $\mathbf{iv}$ intersects the edge $\mathbf{kl}$ of the lower hull only at the point $\mathbf{p}_3$. As before, because the end point $\mathbf{k}$ is closer to $\mathbf{p}_3$, the desired path connecting both hulls is the line segment $\mathbf{kv}$.

Step 2: compute the tangents to the convex hulls

The objective of this second main step is to find a pair of geodesics tangent to both hulls. As shown in Figure 7.11(c), there are in fact four possible geodesics, only two of which (the non crossing ones) are valid for the present application. The following simple test is used to prevent the algorithm from converging to either of the non-valid tangents that connect the opposite sides of both hulls. This is done by comparing the left, right and direction vectors, $\boldsymbol{v}_l$, $\boldsymbol{v}_r$ and $\boldsymbol{v}_d$, at either end of each tangent to the hull at a point $\mathbf{p}$ of the tangent $\mathcal{T}$. As shown in Figures 7.11(c) and 7.12, the left and right vectors $\boldsymbol{v}_l$ and $\boldsymbol{v}_r$ are formed by the (last two points on the) edges of the hull to the left and right of $\mathbf{p}$ respectively, while $\boldsymbol{v}_d$ is formed by the last two or first two points of $\mathcal{T}$. The two possible conditions which can exist at either end point are as follows:

i. the angle between $\boldsymbol{v}_d$ and $\boldsymbol{v}_r$ is less than the angle between $\boldsymbol{v}_d$ and $\boldsymbol{v}_l$ i.e. $\boldsymbol{v}_d.\boldsymbol{v}_r < \boldsymbol{v}_d.\boldsymbol{v}_l$.

ii. the angle between $\boldsymbol{v}_d$ and $\boldsymbol{v}_l$ is less than the angle between $\boldsymbol{v}_d$ and $\boldsymbol{v}_r$ i.e. $\boldsymbol{v}_d.\boldsymbol{v}_l < \boldsymbol{v}_d.\boldsymbol{v}_r$.

If the tangent $\mathcal{T}$ is valid, opposite conditions, e.g. (i) and (ii), hold at its start and end points $\mathbf{p}_s$ and $\mathbf{p}_e$. However, if $\mathcal{T}$ is not valid, the same conditions, e.g. (i) or (ii), hold at $\mathbf{p}_s$ and $\mathbf{p}_e$. For example, in Figure 7.11(c) the first condition holds at the point $\boldsymbol{w}$, while the second condition holds at the point $\mathbf{j}$ of the valid geodesic $\mathbf{jw}$. Similarly, the second condition holds at the point $\mathbf{q}$, while the first condition holds at the point $\mathbf{d}$ of the valid geodesic $\mathbf{dq}$. However, the first condition holds at both ends of the non-valid geodesic $\boldsymbol{cv}$, while the second condition holds at both ends of the non-valid geodesic $\mathbf{kr}$. This test is made only when a tangent is found.

A geodesic $\mathcal{G}$ is tangent to a hull, at one of its end points $\mathbf{p}$, if the vector $\boldsymbol{v}_d$ at this point satisfies the condition

$$(\boldsymbol{v}_l \times \boldsymbol{v}_d) \cdot (\boldsymbol{v}_r \times \boldsymbol{v}_d) > 0$$

This expression simply checks if the angle between $\boldsymbol{v}_d$ and $\boldsymbol{v}_l$ or $\boldsymbol{v}_r$ (measuring clockwise, by the right hand rule) is less than π. If both angles are less than π, the cross products of both pairs of vectors point in the same direction. For example, in Figure 7.11(d) the vector $\boldsymbol{v}'_d$ is tangent to the hull and thus lies on the same side of the half-plane created by $\boldsymbol{v}_l$. However the vector $\boldsymbol{v}''_d$ is not tangent to the hull as it lies on the opposite side of the half-plane, and the angle between it and $\boldsymbol{v}_l$ is greater than π. $\mathcal{G}$ is a tangent only if it is tangent to both hulls, at its start and end points. These tests are at the core of the algorithm, which proceeds as follows.

Following O'Rourke (1998), the geodesic $\mathcal{G}$ connecting both convex sets is duplicated. By convention, the original geodesic, now referred to as $\mathcal{G}_1$ is the candidate for the first geodesic, while the copy $\mathcal{G}_2$ is the candidate for the second geodesic. As shown in Figure 7.11(b), the end points of either geodesic are advanced in opposite directions

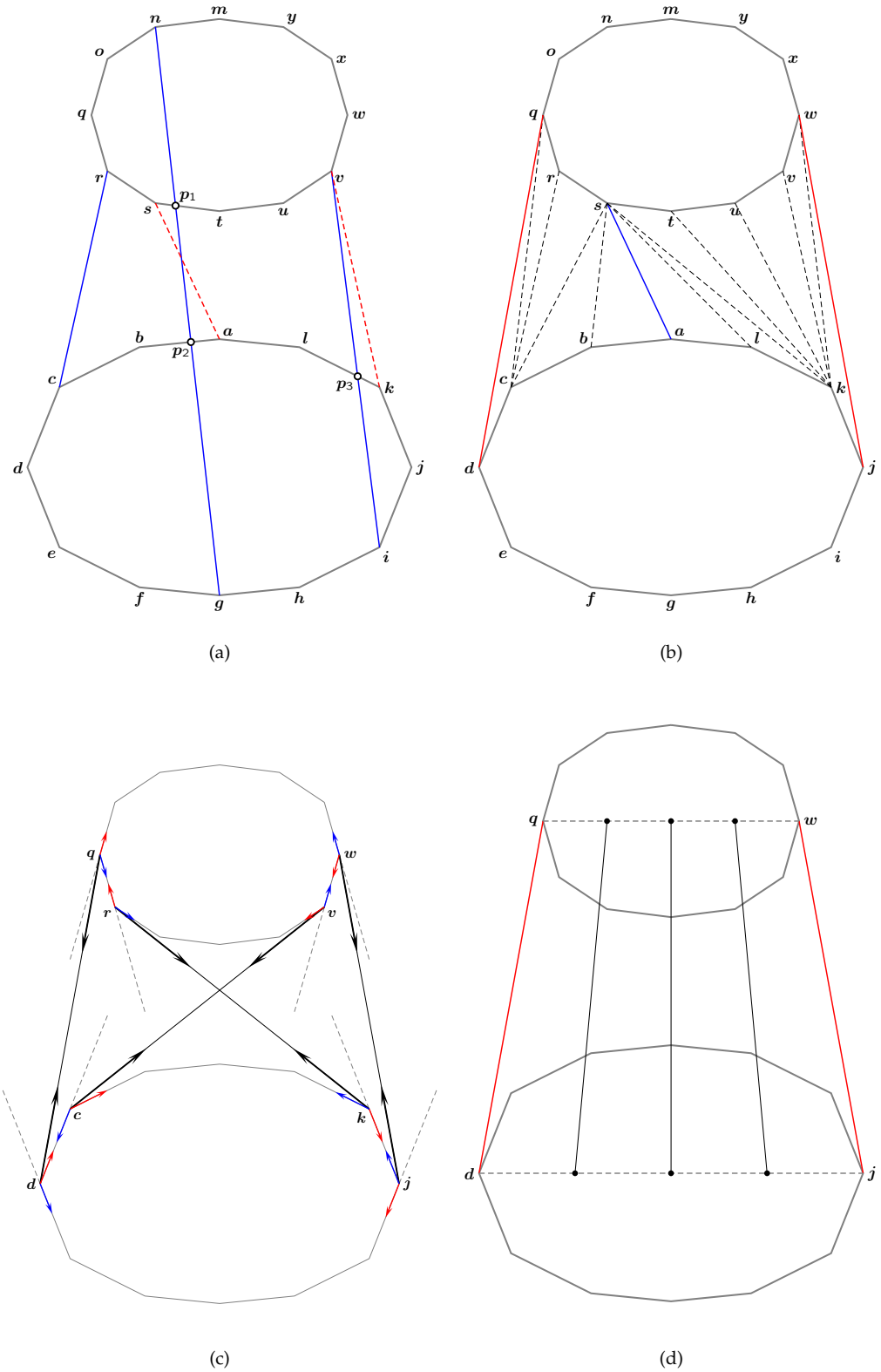


Figure 7.11 (a) three possible types of paths connecting two convex hulls. (b) stepwise bifurcation of a connecting geodesic and its convergence to the mutual tangents of a convex hull. Right and left vectors are drawn as red and blue arrows respectively, while direction vectors are drawn as black arrows. (c) the four possible mutual tangents connecting two convex hulls. (d) interpolating the mutual tangents $\mathbf{dq}$ and $\mathbf{jw}$. Black dots are the subdivisions of the "connecting" geodesics $\mathbf{dj}$ and $\mathbf{qw}$.

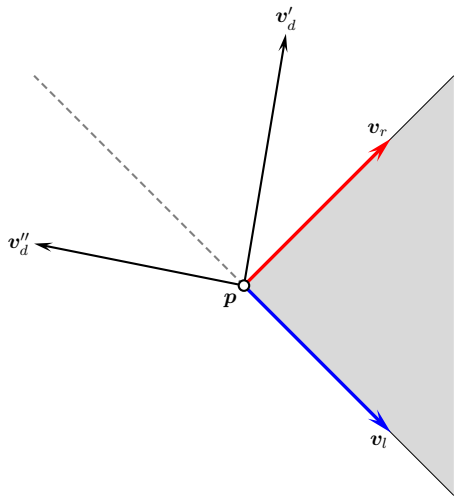


Figure 7.12 Checking for the tangency of a vector v_d.

around the hulls they connect, until the geodesic is tangent to both hulls and valid. For example the end point of $\mathcal{G}_1$ on the first hull can be advanced in the clockwise direction while the end point of $\mathcal{G}_2$ on the same hull is advanced in the, opposite, counterclockwise direction. In this case, the end points of $\mathcal{G}_1$ and $\mathcal{G}_2$ on the second hull must be advanced in the reverse directions (i.e. the end point of $\mathcal{G}_1$ on the second hull is advanced in the counterclockwise direction, while the end point of $\mathcal{G}_2$ on the same hull is advanced in the clockwise direction). This is done until $\mathcal{G}_1$ and $\mathcal{G}_2$ are tangent to both hulls and valid. The result of this convention is that both geodesics bifurcate and converge to the mutual tangents, as shown in Figure 7.11(b). In this figure the lower end of one copy the connecting geodesic **as** is advanced clockwise about the lower hull while the upper end of the geodesic is advanced counterclockwise about the upper hull, as follows:

- Advance lower end CW: **as** $\rightarrow$ **ls** $\rightarrow$ **ks**
  ks is tangent to lower hull, but not upper hull
- Advance upper end CCW: **ks** $\rightarrow$ **kt** $\rightarrow$ **ku** $\rightarrow$ **kv** $\rightarrow$ **kw**
  kw is tangent to upper hull, but not lower hull
- Advance lower end CW: **kw** $\rightarrow$ **jw**
  jw is tangent to lower and upper hull, and is therefore a mutual tangent

On the other hand, the lower end of the other copy of the connecting geodesic is advanced counterclockwise about the lower hull while the upper end of the geodesic is advanced clockwise about the upper hull, as follows:

- Advance lower end CCW: **as** $\rightarrow$ **bs** $\rightarrow$ **cs**
  cs is tangent to lower hull, but not upper hull
- Advance upper end CW: **cs** $\rightarrow$ **cr** $\rightarrow$ **cq**
  cq is tangent to upper hull, but not lower hull

- Advance lower end CCW: $\mathbf{cq} \rightarrow \mathbf{dq}$
  $\mathbf{dq}$ is tangent to lower and upper hull, and is therefore a mutual tangent

7.2.4 *Generating muscle fibres*

The final phase of muscle construction involves the interpolation of the mutual tangents, generated in the second step of the previous phase, in order to form the basic muscle fibres. This final phase consists of the two following steps:

Step 1: interpolate mutual tangents

The end points of $\mathcal{G}_1$ and $\mathcal{G}_2$ on each hull are connected by the geodesics $\mathcal{G}'$ and $\mathcal{G}''$. $\mathcal{G}'$ connects the end points of $\mathcal{G}_1$ and $\mathcal{G}_2$ on the first hull, where $\mathcal{G}''$ connects the end points of $\mathcal{G}_1$ and $\mathcal{G}_2$ on the second hull. $\mathcal{G}'$ and $\mathcal{G}''$ are then subdivided into a number of segments equal to the number of muscle fibres to be generated, and new geodesics $\mathcal{G}_i$ constructed between each corresponding subdivision point along $\mathcal{G}'$ and $\mathcal{G}''$, as shown in Figures 7.11(d) and 7.5(d).

Step 2: grow and trim ends of muscle fibres

As shown in Figure 7.5(e), the geodesics (muscle fibres) $\mathcal{G}_i$ do not precisely terminate at the outermost boundaries of the boundary curves representing the muscle attachment regions, but instead either exceed or fail to reach the extremes of the boundary curve at both ends. Correcting these anomalies is straightforward. A geodesic $\mathcal{G}_i$ is extended at either end if it fails to reach the extremes of the boundary curve, or trimmed if it exceeds the extents of the boundary curve. It is for this reason that a copy of the boundary curves is used to compute the convex hull while the original is not altered, see Section 7.2.2.

In order to determine whether a geodesic $\mathcal{G}_i$ should be grown or trimmed at either of its ends, the geodesic is extended beyond both of its end points, using the initial value algorithm of Polthier and Schmies (1998), until it intersects either the boundary curve or the convex hull at that end, whichever is met first. The extended geodesic is distinguished by an asterisk, e.g. $\mathcal{G}_i^*$. If $\mathcal{G}_i^*$ intersects the boundary curve first, i.e. before it intersects the convex hull, it is obvious that the original geodesic $\mathcal{G}_i$ was short of the boundary curve, and required extension. Fortunately, in this case, this exploratory extension doubles as the amount of growth required at this end of $\mathcal{G}_i$. If however $\mathcal{G}_i^*$ fails to intersect the boundary curve and instead intersects the convex hull it is obvious that $\mathcal{G}_i$ originally exceeded the boundary curve and therefore requires trimming at this end. This is done by reducing the length of $\mathcal{G}_i$ or $\mathcal{G}_i^*$ to the point where it first intersects the boundary curve. This reduction starts from the end point to be trimmed and is also adequate for trimming geodesics to convoluted boundary curves that are typically intersected more than twice – in contrast to simpler boundary curves that are intersected just twice.

It is worth emphasizing that, in both scenarios (growing and trimming), especially the second, the convex hull provides a reliable basis for terminating the search for the boundary curve. An example of the result of this trimming and growing process is shown in Figure 7.5(f).

Figures 7.13 and 7.14 show the result of the above processes on the three head models. In all, 176 muscle fibres were generated for the nine muscles for which origins and insertions are defined on the SMAS. These muscles are, the left and right frontalis (42 fibres each), procerus (20 fibres) left and right corrugator supercilii (12 fibres each), left and right zygomaticus major (12 fibres each), and the left and right zygomaticus minor (12 fibres each). As shown in Tables 7.1 and 7.2, the process of generating the muscle fibres requires a considerable number of supplementary geodesic computations, far more than the final number of geodesic fibres themselves.

In view of the variability of mimic muscles, discussed in Section 3.3, it is unclear how to definitively or conclusively validate the shapes of the muscles generated in this chapter, without putting the muscles to use in the synthesis of facial expressions. Nevertheless the shapes of these muscles, shown in Figures 7.13 and 7.14, match the shapes of the muscles in Figures 3.2, 3.3 and 3.4.

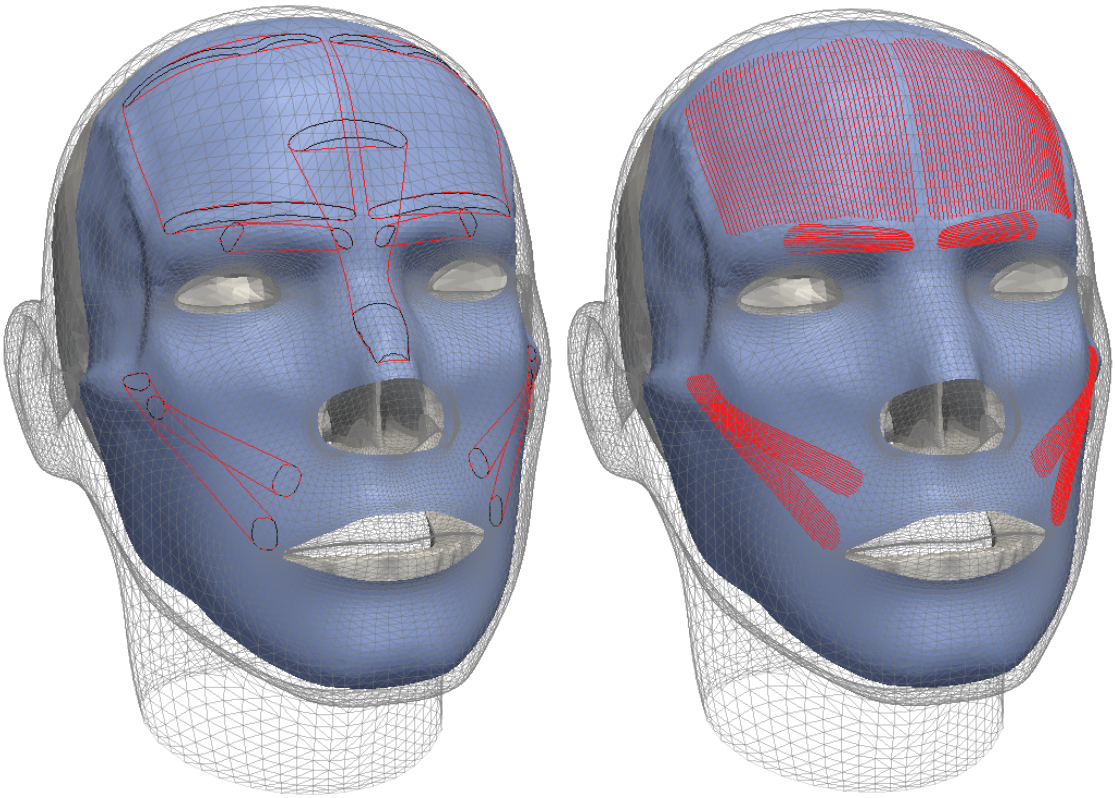
7.3 SUMMARY AND RECOMMENDATION FOR FURTHER WORK

This chapter outlines a series of algorithms for generating any number of, naturally-occurring or phantom, facial muscle fibres whose origins and insertions are defined by painting on the texture map of a generic skull model. This method of specifying muscles is particularly attractive because editing muscle attachment regions is easy to perform by repainting the textures. The muscle fibres are created on the surface of the superficial musculoaponeurotic system (SMAS) – a much neglected anatomical substructure, which is modeled as a variational implicit or RBF surface. The process of muscle generation involves first computing the geodesic convex hulls of the muscle attachment regions on the SMAS, after which the lateral extents of each muscle are constructed as the geodesic mutual tangents of the convex hulls.

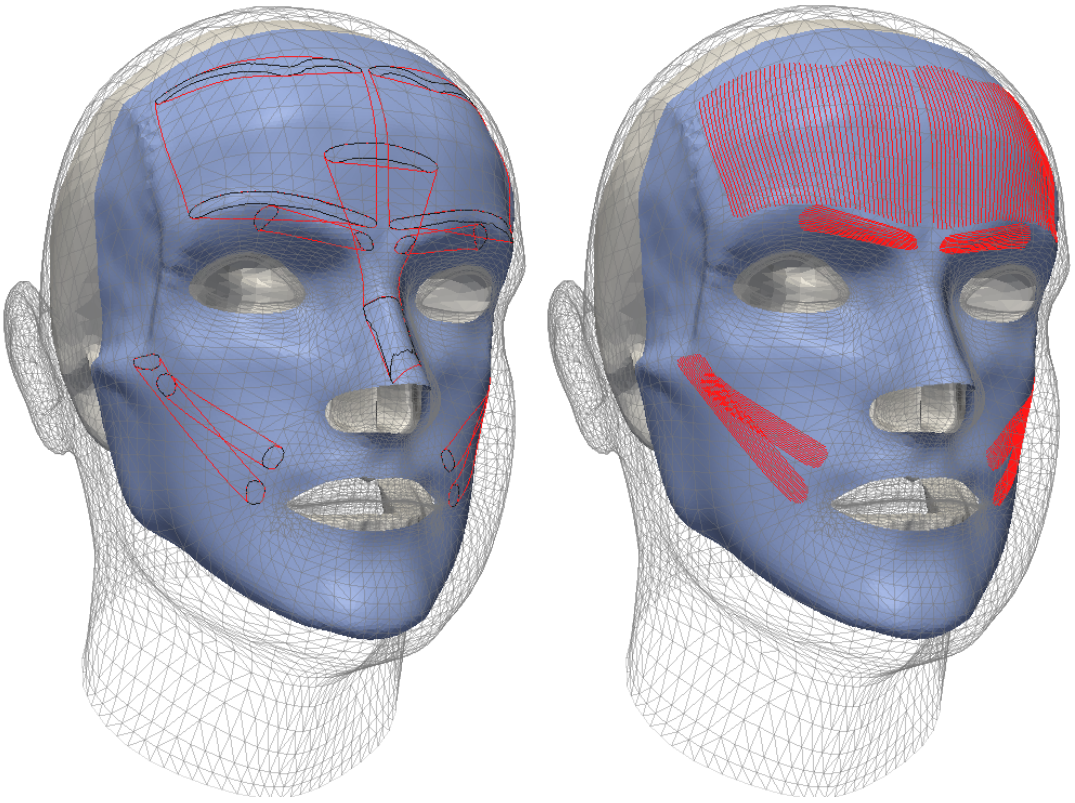
7.3.1 *Future work*

The current work has a several limitations. Naturally, these shortcomings will be the focus of the following future work:

i. Implement new techniques for user control of muscle shapes. Currently the shape of the muscle belly is automatically determined by the mutual extents of its origin and insertion regions. As such, this method is inappropriate for modeling bifid muscles, severely curved muscles (e.g. the risorius, levator labii superioris alaeque nasi) and sphincter muscles (e.g. the orbicularis oris and orbicularis oculi).

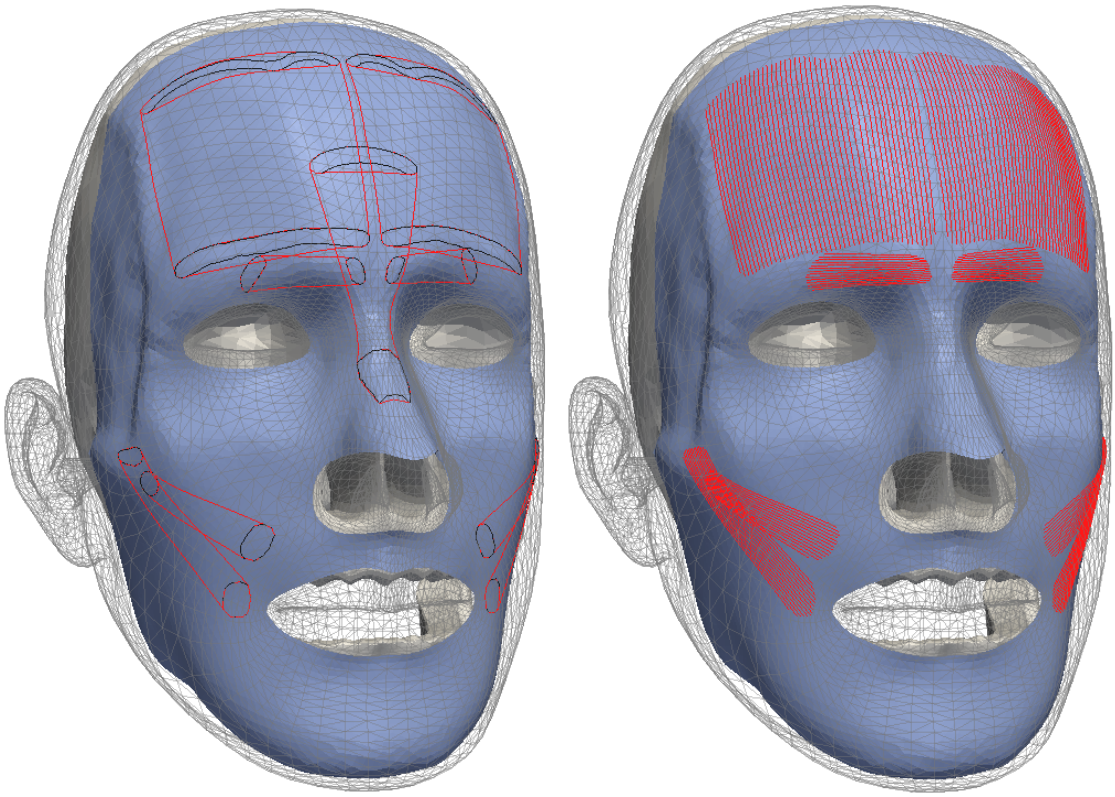


(a) African Head



(b) European Head

Figure 7.13 Left (top and bottom): mutual tangents and convex hulls (both red) of muscle attachment regions (black) constructed on SMAS surface (blue). right (top and bottom): muscle fibres (red) as boundary-value straightest geodesics. Fibres of the procerus muscle not shown.



(a) MakeHuman Head

Figure 7.14 Left: mutual tangents and convex hulls (both red) of muscle attachment regions (black) constructed on SMAS surface (blue). right: muscle fibres (red) as boundary-value straightest geodesics. Fibres of the procerus muscle not shown.

ii. Take into consideration those mimic muscles that are not in continuity with the SMAS. In this work it is assumed that facial muscles have their origins and insertions on the SMAS. This is only true for a subset of mimic muscles.

iii. Generate a volumetric model of the SMAS, as well as other fine substructures for greater physical realism. Currently, the SMAS is modeled as a surface having no thickness. Although the SMAS is relatively thin and is sometimes referred to as a plane, i.e. superficial musculoaponeurotic plane (SMAP), it is more accurately described as an anatomical structure with varying thickness.

iv. Exploit the abundant opportunities for parallelism inherent in the various steps of the muscle generation process. Currently convex hulls, mutual tangents and muscle fibres are computed sequentially. However, all convex hulls can be generated in parallel as they are independent tasks, and do not exchange data. The same is true for mutual tangents and muscle fibres. The exhaustive search for straightest geodesics discussed in Chapter 5 is equally trivial to parallelize.

Head Model	SMAS triangle count	# SG	# Computed directly	# SG Corrected	# SG Approximated	Total runtime	Geodesic calc runtime
African	23,931	1102	994	102	56	643 sec. (10.72 min.)	499 sec. (8.32 min.)
European	13,593	1313	1081	193	93	1089 sec. (18.15 min.)	887 sec. (14.78 min.)
MakeHuman	37,659	943	838	64	41	590 sec. (9.83 min.)	386 sec. (6.43 min.)

Table 7.1 Runtime statistics for muscle generation on a laptop computer with a 1.6GHz (single core) processor and 512Mb main memory. (SG: straightest geodesic)

Head Model	SMAS triangle count	# SG	# Computed directly	# SG Corrected	# SG Approximated	Total runtime	Geodesic calc runtime
African	23,931	1102	944	102	56	328 sec. (5.47 min.)	215 sec. (3.58 min.)
European	13,593	1332	1100	137	95	744 sec. (12.40 min.)	414 sec. (6.89 min.)
MakeHuman	37,659	943	838	64	41	203 sec. (3.38 min.)	122 sec. (2.03 min.)

Table 7.2 Runtime statistics for muscle generation on a PC with a 3.2GHz processor (single core) and 2Gb main memory. (SG: straightest geodesic)

Let us hear the conclusion of the whole matter — Ecclesiastes 12:13

8

CONCLUSION: SUMMARY, FUTURE WORK, POSSIBILITIES AND PERSPECTIVES

The task of creating convincing facial animation can be cast as two related subtasks: one, of producing life-like facial deformations or expressions, and the other of correctly timing the transition between the sequence of facial expressions. Whereas the former problem is governed by physical laws and processes that can be simulated, the latter problem is invariably artist-driven, and must remain so. In fact, although seemingly random, the pattern of expressive wrinkles produced by the human face is in reality a (deterministic) function of the dimensions of the mimic muscles, the thickness of facial skin and the shape of the skull. Evidence of this assertion is the observation that each human face naturally, and repeatedly, produces one distinct pattern of wrinkles. The same ought to be true for 3D models imbued with anatomical information, so that the facial expressions unique to a given character can be computed given the constraints of its anatomy. This task can be thought of as the forward problem. The inverse problem, i.e. computing the properties of the anatomical substructures required in order to produce a set of user-supplied facial expressions, is much harder, and should also be investigated, as it will enable greater artistic input into the rigging process.

Toward this end, this thesis developed a series of techniques for constructing a skull, muscular and superficial musculoaponeurotic system (SMAS) for any given head model. Skull construction was accomplished by fitting a generic skull to a head model, using landmarks and 3D thin-plate splines. Although neither technique is new, nor is their combined use, the introduction of an interactive scaling technique for incorporating experimentally obtained soft-tissue depth data into the morphing process does not appear in literature. Furthermore, the skull fitting technique used is a two-stage process based on sliding semilandmarks and Hermite (tangent) information in order to improve the accuracy of the fit.

The SMAS is modeled as a variational implicit or radial basis function (RBF) surface, overlying the masseter, temporalis muscles and the temporalis fascia, all of which are morphed together with the generic skull to fit a head model. The fibres of the mimic muscle system are modeled as boundary-value straightest geodesics interpolating the mutual tangents of muscle attachment regions, projected to the SMAS, directly from a closed discrete curve in 3D space or indirectly by painting them on the morphable skull model. The latter approach is however favored as painted bitmaps are easier to define and edit, and are automatically transferred to a target head model during skull fitting, and need only be done once. On the other hand, directly projecting muscle attachment regions to the SMAS must be done for each head model, after the SMAS is constructed.

In other words, unlike closed discrete curves, painted bitmaps are part of the generic skull package and are reusable.

The idea of boundary-value straightest geodesics is also new and introduced for the first time in this thesis. This method of constructing straightest geodesics is heuristics-based and iteratively finds a path or sequence of polygonal faces that embed a straightest geodesic in the direction of the line connecting two given points. In addition, the concept of a straightest possible geodesic is introduced as the straightest path that can be traced between two points when no straightest geodesic connecting both points can be found using the heuristics.

This workflow allows the creation of muscle fibres that link to the SMAS and follow the curvature of the human head and skull. This is perhaps the most accurate representation of the human facial mimic muscle system to date. In addition, the muscle generation algorithm developed is independent of the shape, size and location of the muscle attachment regions, and as such is not limited to creating fibres for the standard set of muscles. Furthermore, the workflows developed are fast, customizable, emphasize reuse and are easy to deploy for head models, irrespective of topology.

However, as previously stated, this method is ill-suited for constructing bifid, severely curved and sphincter muscles, because the shapes of these muscles cannot be obtained my merely specifying their origins and insertions. This shortcoming should be addressed in subsequent work, by developing techniques for user specification of muscle shapes. Equally important is the development of an efficient facial expression engine that uses the anatomical substructures generated in order to produce realistic facial expressions. Although this expression engine will require significant computation Clutterbuck and Jacobs (2010) have recently shown how to interpolate a set of expensive physical simulations over a parameter and pose space, using the pose-space deformation technique Lewis *et al.* (2000). (The pose space deformation technique is a scattered data interpolation technique based on Gaussian radial basis functions.) This method was used to generate the body/muscule simulations of the *Navi* characters in the movie "Avatar". Ninety percent (90%) of the simulations generated, using this technique, did not require manual adjustment. Therefore, contrary to popular belief, physically-based facial animation techniques are viable for CG production, especially when there exists a fast and customizable rig-builder.

With further research and development on the generation of SMAS and other detailed anatomical substructures, this workflow should also result in physically-based facial animation systems that more readily lend themselves to applications such as the planning of modern facial plastic surgery procedures. These procedures, pioneered by Skoog (1974), involve the plication (folding), resection and advancement of the SMAS Mendelson (2009), Hamra (1990), Barton Jr (1992), Mendelson (1995b), LaTrenta (2004), and are an essential part of rhytidectomy (surgical removal of facial wrinkles), face lifts and contouring, and the effacement of nasolabial folds and other age-related features.

8.1 RELATIONSHIP WITH COMPUTERIZED FORENSIC FACIAL RECONSTRUCTION

The skull fitting process outlined in this thesis amounts to a reversal of the registration-based computerized forensic facial reconstruction methods described in publications such as Jones (2001) and Salas and Maddock (2010). However, these publications limit their scope to the determination of the external or superficial facial features and therefore do not generate anatomical substructures. Kähler *et al.* (2003), on the other hand, fit a set of virtual muscles to a volume scan of a human skull in addition to registering an (animation-ready) reference head model pre-rigged with a mass-spring system. Unfortunately, because the registration process employed makes use of non-affine deformation functions, e.g. thin-plate splines, the straightness property of the muscles fibres – if the virtual muscles used possess such a quality – will not be preserved. It is for this reason that muscles are not precomputed (as part of the generic skull) in this work. The other reason why muscles are not precomputed (on the generic skull) is that the insertions of many muscles, e.g. the zygomaticus major and minor, must be defined in relation to the face. Recall also that mimic muscles are invested and interlinked with the SMAS, which can only be reconstructed from a set of point constraints derived from the head and skull models.

A

APPENDIX

A.1 FOURIER ANALYSIS

Any three-dimensional Euclidian vector $\mathbf{v}$ can be expressed in terms of its its three orthogonal unit (or basis) vectors $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ as follows

$$v = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$$

The components or coordinates of $\mathbf{v}$ can be calculated via the dot products

$$x = \mathbf{v}.\mathbf{i} \qquad y = \mathbf{v}.\mathbf{j} \qquad z = \mathbf{v}.\mathbf{k}$$

Similarly, the coordinates of any function $f(t)$ in an infinite dimensional space spanned by the basis vectors $e^{-2\pi i\omega t}$ can be computed by taking the inner product or Fourier transform

$$\tilde{f}(\omega) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\omega t}dt \tag{A.1}$$

In other words, the Fourier transform decomposes $f(t)$ into its basis vector components. Accordingly, the function $f(t)$ can be represented as a combination of the basis functions weighted by their fourier transforms as follows

$$f(t) = \int_{-\infty}^{\infty} \tilde{f}(\omega)e^{2\pi i\omega t}d\omega \tag{A.2}$$

This expression is referred to as an inverse Fourier transform.

In n-dimensions t and ω become the n-tuples

$$\mathbf{t} = (t_1, t_2, \ldots, t_n) \qquad \boldsymbol{\omega} = (\omega_1, \omega_2, \ldots, \omega_n)$$

and

$$\boldsymbol{\omega}.\mathbf{t} = \omega_1 t_1 + \omega_2 t_2 + \ldots + \omega_n t_n$$

Then the Fourier and inverse Fourier transforms can be written as

$$\mathcal{F}[\,f(\mathbf{t})\,] = \tilde{f}(\boldsymbol{\omega}) = \int_{\mathbb{R}^n} f(\mathbf{t})e^{-2\pi i\boldsymbol{\omega}.\mathbf{t}}d\mathbf{t} \tag{A.3}$$

and

$$\mathcal{F}^{-1}[\,f(\boldsymbol{\omega})\,] = f(\mathbf{t}) = \int_{\mathbb{R}^n} \tilde{f}(\boldsymbol{\omega})e^{2\pi i\boldsymbol{\omega}.\mathbf{t}}d\boldsymbol{\omega} \tag{A.4}$$

It is important to note that at $\mathbf{t} = 0$, Equation A.1 becomes

$$f(0) = \int_{\mathbb{R}^n} \tilde{f}(\boldsymbol{\omega})d\boldsymbol{\omega} \tag{A.5}$$

Furthermore, upon negating $\mathbf{t}$,

$$f(-\mathbf{t}) = \int_{\mathbb{R}^n} \tilde{f}(\boldsymbol{\omega}) e^{-2\pi i \boldsymbol{\omega}.\mathbf{t}} d\boldsymbol{\omega}$$

The complex conjugate of this expression is

$$\overline{f(-\mathbf{t})} = \int_{\mathbb{R}^n} \overline{\tilde{f}(\boldsymbol{\omega})} e^{2\pi i \boldsymbol{\omega}.\mathbf{t}} d\boldsymbol{\omega}$$

Therefore, if $g(\boldsymbol{\omega}) = \overline{\tilde{f}(\boldsymbol{\omega})}$,

$$\overline{f(-\mathbf{t})} = \int_{\mathbb{R}^n} g(\boldsymbol{\omega}) e^{2\pi i \boldsymbol{\omega}.\mathbf{t}} d\boldsymbol{\omega} = g(\mathbf{t}) \tag{A.6}$$

A.1.1 *Shift theorem*

If the function $f(\mathbf{t})$ is shifted by a value $\mathbf{s}$ to say $f(\mathbf{t}-\mathbf{s})$, the Fourier transform of the shifted function would be

$$\int_{\mathbb{R}^n} f(\mathbf{t}-\mathbf{s}) e^{-2\pi i \omega \mathbf{t}} d\mathbf{t}$$

Substituting $\mathbf{t}' = \mathbf{t}-\mathbf{s}$, so that $d\mathbf{t}' = d\mathbf{t}$ ($\mathbf{s}$ is a constant), the above relationship becomes

$$\int_{\mathbb{R}^n} f(\mathbf{t}') e^{-2\pi i \boldsymbol{\omega}(\mathbf{t}-\mathbf{s})} d\mathbf{t}' \;=\; e^{-2\pi i \boldsymbol{\omega}.\mathbf{s}} \int_{\mathbb{R}^n} f(\mathbf{t}') e^{-2\pi i \boldsymbol{\omega}.\mathbf{t}} d\mathbf{t}'$$

And because the Fourier transform is unchanged by the choice of term ($\mathbf{t}$ or $\mathbf{t}'$)in which it is written,

$$\int_{\mathbb{R}^n} f(\mathbf{t}-\mathbf{s}) e^{-2\pi i \boldsymbol{\omega}.\mathbf{t}} d\mathbf{t} \;=\; e^{-2\pi i \boldsymbol{\omega}.\mathbf{s}}\; \tilde{f}(\boldsymbol{\omega}) \tag{A.7}$$

This relationship is known as the shift theorem.

A.1.2 *Convolution theorem*

In n-dimensions, the convolution operation is given as

$$h(\mathbf{t}) = (f * g)(\mathbf{t}) = \int_{\mathbb{R}^n} g(\mathbf{r})\; f(\mathbf{t}-\mathbf{r})\; d\mathbf{r}$$

The Fourier transform of $h(\mathbf{t})$ is given as

$$\begin{aligned}
& \int_{\mathbb{R}^n} \left[\int_{\mathbb{R}^n} g(\mathbf{r})\; f(\mathbf{t}-\mathbf{r})\; d\mathbf{y} \right] e^{-2\pi i \boldsymbol{\omega}.\mathbf{t}} d\mathbf{t} \\
= & \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} g(\mathbf{r})\; f(\mathbf{t}-\mathbf{r})\; e^{-2\pi i \boldsymbol{\omega}.\mathbf{t}}\; d\mathbf{r}\; d\mathbf{t} \\
= & \int_{\mathbb{R}^n} \int_{\mathbb{R}^n} g(\mathbf{r})\; f(\mathbf{t}-\mathbf{r})\; e^{-2\pi i \boldsymbol{\omega}.\mathbf{t}}\; e^{-2\pi i \boldsymbol{\omega}.\mathbf{r}}\; e^{2\pi i \boldsymbol{\omega}.\mathbf{r}}\; d\mathbf{r}\; d\mathbf{t} \\
= & \int_{\mathbb{R}^n} g(\mathbf{r})\; e^{-2\pi i \boldsymbol{\omega}.\mathbf{r}} \left[\int_{\mathbb{R}^n} f(\mathbf{t}-\mathbf{r})\; e^{-2\pi i \boldsymbol{\omega}.(\mathbf{t}-\mathbf{r})}\; d\mathbf{t} \right] d\mathbf{r}
\end{aligned}$$

(Let $\mathbf{s} = \mathbf{t} - \mathbf{r}$, so that $\mathbf{ds} = \mathbf{dt}$ inside the integral)

$$= \int_{\mathbb{R}^n} g(\mathbf{r})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{r}} \left[\int_{\mathbb{R}^n} f(\mathbf{t}-\mathbf{r})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{s}}\, \mathbf{ds} \right] \mathbf{dr}$$

(Separating out the two integrals)

$$= \left[\int_{\mathbb{R}^n} g(\mathbf{r})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{r}}\, \mathbf{dr} \right] \left[\int_{\mathbb{R}^n} f(\mathbf{s})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{s}}\, \mathbf{ds} \right] = \tilde{g}(\boldsymbol{\omega})\, \tilde{f}(\boldsymbol{\omega}) \qquad \text{(A.8)}$$

A.1.3 *Fourier transform of differential operators*

The Fourier transform of the first derivative $\frac{\partial f(\mathbf{x})}{\partial x_i}$ (or f_{x_i}) is

$$\mathcal{F}[f_{x_i}] = \int_{\mathbb{R}^n} f_{x_i}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}}\, \mathbf{d\omega} = f(\mathbf{x})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \Big|_{-\infty}^{+\infty} - (-2\pi i \omega_i) \int_{\mathbb{R}^n} f(\mathbf{x})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \mathbf{dx}$$

where the last expression is a result of integration by parts. Also as the first term of this expression is zero

$$\mathcal{F}[f_{x_i}] = 2\pi i \omega_i \int_{\mathbb{R}^n} f(\mathbf{x})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \mathbf{dx} = 2\pi i \omega_i \tilde{f}(\boldsymbol{\omega}) \qquad \text{(A.9)}$$

By the same logic, the Fourier transform of the second derivative $\frac{\partial^2 f(\mathbf{x})}{\partial x_i^2}$ (or $f_{x_i^2}$) is

$$\mathcal{F}[f_{x_i^2}] = \int_{\mathbb{R}^n} f_{x_i^2}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}}\, \mathbf{d\omega} = f_{x_i}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \Big|_{-\infty}^{+\infty} - (-2\pi i \omega_i) \int_{\mathbb{R}^n} f_{x_i}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \mathbf{dx}$$

Substituting A.9 into the last term

$$\mathcal{F}[f_{x_i^2}] = (2\pi i \omega_i)^2 \int_{\mathbb{R}^n} f(\mathbf{x})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \mathbf{dx} = (2\pi i \omega_i)^2\, \tilde{f}(\boldsymbol{\omega}) \qquad \text{(A.10)}$$

Furthermore, the Fourier transform of the mixed derivative $\frac{\partial^2 f(\mathbf{x})}{\partial x_i x_j}$ (or $f_{x_i x_j}$) is

$$\mathcal{F}[f_{x_i x_j}] = \int_{\mathbb{R}^n} f_{x_i x_j}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}}\, \mathbf{d\omega} = f_{x_j}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \Big|_{-\infty}^{+\infty} - (-2\pi i \omega_i) \int_{\mathbb{R}^n} f_{x_j}\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \mathbf{dx}$$

As before, this is obtained by integration by parts. Equating the first term to zero and substituting A.9 into the last term

$$\mathcal{F}[f_{x_i x_j}] = (2\pi i)^2 \omega_i \omega_j \int_{\mathbb{R}^n} f(\mathbf{x})\, e^{-2\pi i \boldsymbol{\omega}.\mathbf{x}} \mathbf{dx} = (2\pi i)^2\, \omega_i \omega_j \tilde{f}(\boldsymbol{\omega}) \qquad \text{(A.11)}$$

A.1.4 *Plancherel theorem*

From the convolution theorem (section A.1.2), if $h = f * f$ (h is the convolution of f with itself) and from A.6, $\tilde{f}(\boldsymbol{\omega}) = \overline{\tilde{f}(\boldsymbol{\omega})}$

$$\tilde{h}(\boldsymbol{\omega}) = \tilde{f}(\boldsymbol{\omega})\, \overline{\tilde{f}(\boldsymbol{\omega})} = |\, \tilde{f}(\boldsymbol{\omega})\, |$$

Consequently,

$$\int_{\mathbb{R}^n} |\,\tilde{f}(\omega)\,|^2\,d\omega \;=\; \int_{\mathbb{R}^n} \tilde{h}(\omega) \;=\; \tilde{h}(0) \qquad \text{(from A.5)}$$

Again from the convolution theorem,

$$\tilde{h}(0) \;=\; \int_{\mathbb{R}^n} f(y)\,f(0-y)\,dy \;=\; \int_{\mathbb{R}^n} f(y)\,f(-y)\,dy$$

And because $f(-y) \;=\; \overline{f(y)}$

$$\int_{\mathbb{R}^n} f(y)\,f(-y)\,dy \;=\; \int_{\mathbb{R}^n} f(y)\,\overline{f(y)}\,dy \;=\; \int_{\mathbb{R}^n} |\,f(y)\,|^2\;dy$$

Thus (replacing the variable y with x)

$$\int_{\mathbb{R}^n} |\,f(x)\,|^2\;dx \;=\; \int_{\mathbb{R}^n} |\,\tilde{f}(\omega)\,|^2\,d\omega \tag{A.12}$$

This relationship is known as Plancherel theorem. Crucially, this relationship still holds if f is the derivative (of any order) of some other function.

A.1.5 *Beppo-Levi semi-norm in three dimensions*

In three dimensions, the Beppo-Levi semi-norm is given as

$$\|\,f\,\|^2 = \tag{A.13}$$
$$\int_{\mathbb{R}^3} \left(\left|\frac{\partial^2 f}{\partial x_1^2}\right|^2 + \left|\frac{\partial^2 f}{\partial x_2^2}\right|^2 + \left|\frac{\partial^2 f}{\partial x_3^2}\right|^2 + 2\left|\frac{\partial^2 f}{\partial x_1 \partial x_2}\right|^2 + 2\left|\frac{\partial^2 f}{\partial x_2 \partial x_3}\right|^2 + 2\left|\frac{\partial^2 f}{\partial x_1 \partial x_3}\right|^2 \right) dx_1\,dx_2\,dx_3$$

By a termwise application of the Plancherel theorem

$$\|\,f\,\|^2 = (2\pi)^4 \int_{\mathbb{R}^3} (\omega_1^4 + \omega_2^4 + \omega_3^4 + 2\,\omega_1^2\,\omega_2^2 + 2\,\omega_2^2\,\omega_3^2 + 2\,\omega_1^2\,\omega_3^2\,)\;|\,\tilde{f}(\omega)\,|^2\,d\omega$$

Because

$$(\omega_1^4 + \omega_2^4 + \omega_3^4 + 2\,\omega_1^2\,\omega_2^2 + 2\,\omega_2^2\,\omega_3^2 + 2\,\omega_1^2\,\omega_3^2\,) \;=\; (\,\omega_1^2 + \omega_2^2 + \omega_3^2\,)^2 \;=\|\,\omega\,\|^4$$

The semi-norm can be expressed as,

$$\|\,f\,\|^2 = (2\pi)^4 \int_{\mathbb{R}^3} \|\,\omega\,\|^4\;\;\tilde{f}(\omega)\,|^2\,d\omega$$

This is the same form as Equation 4.6.

A.1.6 *Derivative reproducing property*

$$\left\langle\, p(x)\,,\; t^T \nabla k(\,x\,,\,x'\,)\,\right\rangle \;=\; t^T \nabla p(x')$$

Proof: because

$$\mathbf{t}^{\mathsf{T}} \nabla k(\mathbf{x}, \mathbf{x}') = \lim_{\epsilon \to 0} \sum_{l=1}^{d} \mathbf{t}[l] \frac{k(\mathbf{x}, \mathbf{x}' + \epsilon \mathbf{e}[l]) - k(\mathbf{x}, \mathbf{x}')}{\epsilon}$$

Therefore,

$$\begin{aligned}
\left\langle p(\mathbf{x}), \mathbf{t}^{\mathsf{T}} \nabla k(\mathbf{x}, \mathbf{x}') \right\rangle &= \lim_{\epsilon \to 0} \sum_{l=1}^{d} \mathbf{t}[l] \left\langle p(\mathbf{x}), \frac{k(\mathbf{x}, \mathbf{x}' + \epsilon \mathbf{e}[l]) - k(\mathbf{x}, \mathbf{x}')}{\epsilon} \right\rangle \\
&= \lim_{\epsilon \to 0} \sum_{l=1}^{d} \mathbf{t}[l] \frac{\langle p(\mathbf{x}), k(\mathbf{x}, \mathbf{x}' + \epsilon \mathbf{e}[l]) \rangle - \langle p(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') \rangle}{\epsilon} \\
&= \lim_{\epsilon \to 0} \sum_{l=1}^{d} \mathbf{t}[l] \frac{p(\mathbf{x}' + \epsilon \mathbf{e}[l]) - p(\mathbf{x}')}{\epsilon} \\
&= \mathbf{t}^{\mathsf{T}} \nabla p(\mathbf{x}')
\end{aligned}$$

A.1.7 *Sundry relationships involving terms of the Beppo-Levi semi-norm of order 2*

For an infinite thin-plate spline in $\mathbb{R}^m$,

$$\int_{\mathbb{R}^m} \left| \frac{\partial^2 f(x)}{\partial x_i \partial x_j} \right|^2 d\mathbf{x} = \int_{\mathbb{R}^m} \frac{\partial^2 f}{\partial x_i} \frac{\partial^2 f}{\partial x_j} d\mathbf{x} = \int_{\mathbb{R}^m} \frac{\partial^4 f}{\partial x_i^2 \partial x_j^2} d\mathbf{x}$$

for $i, j = 1 \cdots m$, where $i \neq j$

Proof: If $m = 2$,

$$\int_{\mathbb{R}^2} \left| \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|^2 dx_1\, dx_2 = \int_{\mathbb{R}^2} \frac{\partial^2 f}{\partial x_1^2} \frac{\partial^2 f}{\partial x_2^2} dx_1\, dx_2$$

rewriting the LHS as

$$\int_{\mathbb{R}^2} \left| \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|^2 dx_1\, dx_2 = \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \frac{\partial^2 f}{\partial x_1 \partial x_2} \frac{\partial^2 f}{\partial x_1 \partial x_2} dx_1 \right) dx_2$$

and integrating (the inner integral) by parts

$$\int_{\mathbb{R}} \frac{\partial^2 f}{\partial x_1 \partial x_2} \frac{\partial^2 f}{\partial x_1 \partial x_2} dx_1 = \cancel{\left[\frac{\partial^2 f}{\partial x_1 \partial x_2} \frac{\partial f}{\partial x_2} \right]_{-\infty}^{+\infty}} \to 0 - \int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial f}{\partial x_2} dx_1 = - \int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial f}{\partial x_2} dx_1$$

because a thin-plate spline is flat, i.e. has zero gradient, at infinity. Hence,

$$\int_{\mathbb{R}^2} \left| \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|^2 dx_1\, dx_2 = - \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial f}{\partial x_2} dx_2 \right) dx_2$$

noting that the order of integration has been swapped. Again, integrating (the inner integral) by parts,

$$\int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial f}{\partial x_2} dx_2 = \cancel{\left[\frac{\partial^2 f}{\partial x_1^2} \frac{\partial f}{\partial x_2} \right]_{-\infty}^{+\infty}} \to 0 - \int_{\mathbb{R}} \frac{\partial^2 f}{\partial x_1^2} \frac{\partial^2 f}{\partial x_2^2} dx_1 dx_2 = - \int_{\mathbb{R}} \frac{\partial^2 f}{\partial x_1^2} \frac{\partial^2 f}{\partial x_2^2} dx_2$$

so that

$$\int_{\mathbb{R}^2} \left| \frac{\partial^2 f}{\partial x_1 \partial x_2} \right|^2 dx_1\, dx_2 = \int_{\mathbb{R}^2} \frac{\partial^2 f}{\partial x_1^2} \frac{\partial^2 f}{\partial x_2^2} dx_1\, dx_2 \tag{A.14}$$

as required. The more general expression follows noting that because the mixed partial derivative $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ is with respect to two (unique) directions only, only these two directions are worth taking into consideration in the integral $\mathbb{R}^m$, when $m > 2$. As such the bulk of the proof reduces to the case of $m = 2$.

The second part of the proof is very similar, and starts by considering again the case $m = 2$ of the more general integral

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right) \left(\frac{\partial^2 g}{\partial x_1 \partial x_2} \right) dx_1\, dx_2 = \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right) \left(\frac{\partial^2 g}{\partial x_1 \partial x_2} \right) dx_1 \right) dx_2$$

as before, integrating the inner integral by parts,

$$\begin{aligned} \int_{\mathbb{R}} \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right) \left(\frac{\partial^2 g}{\partial x_1 \partial x_2} \right) dx_1 &= \cancel{\left[\frac{\partial^2 f}{\partial x_1 \partial x_2} \frac{\partial g}{\partial x_2} \right]_{-\infty}^{+\infty}} \cancel{0} - \int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial g}{\partial x_2} dx_1 \\ &= - \int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial g}{\partial x_2} dx_1 \end{aligned}$$

because a thin-plate spline has zero gradient at infinity. Hence

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right) \left(\frac{\partial^2 g}{\partial x_1 \partial x_2} \right) dx_1\, dx_2 = - \int_{\mathbb{R}} \left(\int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial g}{\partial x_2} dx_2 \right) dx_1$$

noting again that the order of integration has been reversed, so that

$$\int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2} \frac{\partial g}{\partial x_2} dx_2 = \cancel{\left[\frac{\partial^3 f}{\partial x_1^2 \partial x_2} g \right]_{-\infty}^{+\infty}} \cancel{0} - \int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2^2} g\, dx_2 = - \int_{\mathbb{R}} \frac{\partial^3 f}{\partial x_1^2 \partial x_2^2} g\, dx_2$$

because a thin-plate spline is flat at infinity, so that

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right) \left(\frac{\partial^2 g}{\partial x_1 \partial x_2} \right) dx_1\, dx_2 = \int_{\mathbb{R}^2} \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} g\, dx_1\, dx_2 \tag{A.15}$$

As before, the more general expression follows noting that because the mixed partial derivative $\frac{\partial^2 f(x)}{\partial x_i \partial x_j}$ is with respect to two (unique) directions only, such that the bulk of the proof reduces to the case $m = 2$.

It is of interest to note that, when $f = g$,

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x_1 \partial x_2} \right)^2 dx_1\, dx_2 = \int_{\mathbb{R}^2} \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} f\, dx_1\, dx_2 \tag{A.16}$$

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x_1^2} \right)^2 dx_1\, dx_2 = \int_{\mathbb{R}^2} \frac{\partial^4 f}{\partial x_1^4} f\, dx_1\, dx_2 \tag{A.17}$$

$$\int_{\mathbb{R}^2} \left(\frac{\partial^2 f}{\partial x_2^2} \right)^2 dx_1\, dx_2 = \int_{\mathbb{R}^2} \frac{\partial^4 f}{\partial x_2^4} f\, dx_1\, dx_2 \tag{A.18}$$

And from these three relations,

$$\begin{aligned}\langle f , g \rangle &= \int_{\mathbb{R}^2} \left(\frac{\partial^4 f}{\partial x_1^4} + 2 \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} + \frac{\partial^4 f}{\partial x_2^4} \right) g \, dx_1 \, dx_2 \\ &= \int_{\mathbb{R}^2} g \nabla^4 f \, dx_1 \, dx_2\end{aligned}$$

for $\mathbb{R}^2$. Similarly,

$$\begin{aligned}\langle f , g \rangle &= \int_{\mathbb{R}^3} \left(\frac{\partial^4 f}{\partial x_1^4} + \frac{\partial^4 f}{\partial x_2^4} + \frac{\partial^4 f}{\partial x_3^4} + 2 \frac{\partial^4 f}{\partial x_1^2 \partial x_2^2} + 2 \frac{\partial^4 f}{\partial x_2^2 \partial x_3^2} + 2 \frac{\partial^4 f}{\partial x_1^2 \partial x_3^2} \right) g \, dx_1 \, dx_2 \, dx_3 \\ &= \int_{\mathbb{R}^3} g \nabla^4 f \, dx_1 \, dx_2 \, dx_3\end{aligned}$$

for $\mathbb{R}^3$, and in general

$$\langle f , g \rangle = \int_{\mathbb{R}^m} g \nabla^4 f \, d\mathbf{x} \tag{A.19}$$

for $\mathbb{R}^m$, recalling that $\langle f , g \rangle$ is defined by Equation 4.3.

BIBLIOGRAPHY

Ackerman M., mar 1998. The visible human project. *Proceedings of the IEEE*, **86**(3), 504 –511. (Cited on page 118.)

Adler F. H. and Scheie H. G., 1969. *Adler's Textbook of Ophthalmology*. Saunders, W B Co. (Cited on page 109.)

Alexander O., Rogers M., Lambeth W., Chiang M. and Debevec P., 2009. The digital emily project: photoreal facial modeling and animation. In *SIGGRAPH '09: ACM SIGGRAPH 2009 Courses*, New York, NY, USA. ACM, 1–15. (Cited on pages 10 and 11.)

Barbarino G., Jabareen M., Trzewik J. and Mazza E., 2008. Physically based finite element model of the face. In *ISBMS '08: Proceedings of the 4th international symposium on Biomedical Simulation*, Berlin, Heidelberg. Springer-Verlag, 1–10. (Cited on pages 14, 16, 25, and 123.)

Barbarino G., Jabareen M., Trzewik J., Nkengne A., Stamatas G. and Mazza E., 2009. Development and validation of a three-dimensional finite element model of the face. *Journal of Biomechanical Engineering*, **131**(4). (Cited on pages 14 and 25.)

Barton Jr F., 1992. Rhytidectomy and the nasolabial fold. *Plastic and Reconstructive Surgery*, **90**(4), 601–7. (Cited on page 148.)

Beeson C., 2004. Animation in the "dawn" demo. In Fernando R., editor, *GPU Gems: Programming Techniques, Tips and Tricks for Real-Time Graphics*. Addison-Wesley. (Cited on page 8.)

Benedetto A. V. *Cosmetic uses of Botulinum Toxin A in the midface.*, 4, 121–162. Informa Healthcare, 2005. (Cited on page 41.)

Bentsianov B. and Blitzer A., 2004. Facial anatomy. *Clinics in Dermatology*, **22**(1), 3–13. (Cited on page 31.)

Bibliowicz J. An automated rigging system for facial animation. Msc thesis, Cornel University, 2005. (Cited on page 18.)

Blanz V. and Vetter T., 1999. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co., 187–194. (Cited on page 7.)

Bookstein F. L., 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.*, **11**(6), 567–585. (Cited on pages 77 and 81.)

Bookstein F. L., 1997. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, **1**(3), 225–243. (Cited on pages 10, 81, and 83.)

Bookstein F., 1991. *Morphometric Tools for Landmark Data*. Cambridge University Press. (Cited on page 30.)

Borshukov G., Montgomery J. and Werner W., 2006. Playable universal capture: compression and real-time sequencing of image-based facial animation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, New York, NY, USA. ACM Press, 8. (Cited on page 10.)

Borshukov G., Piponi D., Larsen O., Lewis J. P. and Tempelaar-Lietz C., 2003. Universal capture: image-based facial animation for "the matrix reloaded". In *SIGGRAPH '03: ACM SIGGRAPH 2003 Sketches & Applications*, New York, NY, USA. ACM Press, 1–1. (Cited on page 10.)

Bui T. D., Heylen D., Nijholt A. and Poel M., 2003. Exporting vector muscles for facial animation. In Butz A., Krüger A. and Olivier P., editors, *Smart Graphics 2003, Third International Symposium*, Lecture Notes in Computer Science 2733, Berlin. Springer, 251–259. ISBN=3-540-40557-7. (Cited on page 17.)

Cazals F. and Pouget M., 2005. Estimating differential quantities using polynomial fitting of osculating jets. *Computer Aided Geometric Design*, **22**, 121–146. Conference version: Symp. on Geometry Processing 2003. (Cited on page 73.)

Cermak M. and Skala V., October 10 – 11 2002. Space subdivision for fast polygonization of implicit surfaces. In *5th International scientific conference on Electronic Computers and Informatics (ECI)*, Solovakia. (Cited on page 125.)

Chen D. Z., Daescu O., Hershberger J., Kogge P. M., Mi N. and Snoeyink J., November 2005. Polygonal path simplification with angle constraints. *Computational Geometry. Theory and Applications.*, **32**, 173–187. (Cited on pages 11, 96, and 97.)

Choe B., Lee H. and Ko H.-S., 2001. Performance-driven muscle-based facial animation. *Journal of Visualization and Computer Animation*, **12**(2), 67–79. (Cited on pages 14 and 16.)

Cloward B., 2010. Automated emotion: Facial animation in star wars: The old republic. (Cited on pages 9 and 23.)

Clutterbuck S. and Jacobs J., 2010. A physically based approach to virtual character deformations. SIGGRAPH 2010 Talk : Avatar in Depth. (Cited on page 148.)

Coleman S. R. and Grover R., 2006. The anatomy of the aging face: volume loss and changes in 3-dimensional topography. *Aesthetic surgery journal*, **26**(1S), S4–9. (Cited on pages 47 and 48.)

Deng X. *A Finite Element Analysis of Surgery of the Human Facial Tissue*. PhD thesis, Columbia University, 1988. (Cited on page 13.)

Deng Z. and Neumann U., 2008. *Data-Driven 3D Facial Animation*. Springer. (Cited on page 10.)

Dettman J. W., 1974. *Introduction to linear algebra and differential equations*. McGraw-Hill, New-York. (Cited on page 53.)

Deuflhard P., Weiser M. and Zachow S., 2006. Mathematics in facial surgery. *AMS Notices*, **53(9)**, 1012–1016. (Cited on page 24.)

Dijkstra E. W., December 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, **1**. (Cited on page 96.)

Donofrio L. M., 2000. Fat distribution: a morphologic study of the aging face. *Dermatol Surg*, **26**(12), 1107–12. (Cited on pages 47 and 48.)

Duncan J., 2008. All the way. *Cinefex*, (112), 44–59. (Cited on page 11.)

Duncan J., 2010. The seduction of reality. *Cinefex*, (120), 68–139. (Cited on page 11.)

Eisert P., 2003. Mpeg-4 facial animation in video analysis and synthesis. *International Journal of Imaging Systems and Technology*, **13**, 245–256. (Cited on page 25.)

Ekman P. and Friesen W., 1978. The facial action coding system: A technique for the measurement of facial movement. In *Consulting Psychologists*. (Cited on page 8.)

Ersotelos N. and Dong F., 2007. Building highly realistic facial modeling and animation: a survey. *Visual Computing*, **24**(1), 13–30. (Cited on page 7.)

Ezzat T., Geiger G. and Poggio T., 2002. Trainable videorealistic speech animation. In *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, New York, NY, USA. ACM, 388–398. (Cited on page 7.)

Farkas L. G., 1994. *Anthropometry of the Head and Face*. Raven Press, 2 edition. (Cited on page 30.)

Fattahi T., 2003. An overview of facial aesthetic units. *Journal of Oral and Maxillofac Surgery*, **61**(10), 1207–11. (Cited on pages 9, 48, and 49.)

Ferreira L. M., Hochman B., Locali R. F. and Rosa-Oliveira L. M. Q., 2006. A stratigraphic approach to the superficial musculoaponeurotic system and its anatomic correlation with the superficial fascia. *Aesthetic Plastic Surgery*, **30**(5), 549–52. (Cited on page 49.)

Fleming B. and Dobbs D., 1999. *Animating facial features and expressions*. Number v. 1 in Charles River Media graphics. Charles River Media. (Cited on page 107.)

Fratarcangeli M. *A Computational Musco-Skeletal Model for Animating Virtual Faces*. PhD thesis, Universitá Degli Studi Di Roma "Sapienza", 2008. (Cited on page 19.)

Furnas D. W., 1989. The retaining ligaments of the cheek. *Plastic and Reconstructive Surgery*, **83**(1), 11–6. (Cited on page 46.)

Furnas D., 1993. Festoons, mounds, and bags of the eyelids and cheek. *Clinics in Plastic Surgery*, **20**(2), 367–85. (Cited on page 48.)

Gelder A. V., 1998. Approximate simulation of elastic membranes by triangulated spring meshes. *journal of graphics, gpu, and game tools*, **3**(2), 21–41. (Cited on page 13.)

Geller T., 2008. Overcoming the uncanny valley. *IEEE Computer Graphics and Applications*, **28**(4), 11–17. (Cited on page 20.)

Ghassemi A., Prescher A., Riediger D. and Axer H., 2003. Anatomy of the SMAS revisited. *Aesthetic Plastic Surgery*, **27**(4), 258–264. (Cited on page 44.)

Gladilin E., Zachow S., Deuflhard P. and Hege H.-C., 2001. Towards a realistic simulation of individual facial mimics. In *VMV '01: Proceedings of the Vision Modeling and Visualization Conference 2001*. Aka GmbH, 129–134. (Cited on pages 14 and 16.)

Goldfeather J. and Interrante V., 2004. A novel cubic-order algorithm for approximating principal direction vectors. *ACM Trans. Graph.*, **23**(1), 45–63. (Cited on page 73.)

Goldfinger E., 1991. *Human Anatomy for Artists: The Elements of Form*. Oxford Univ. Press, New York. (Cited on pages 31 and 40.)

Gonzáles-Ulloa M., 1956. Restoration of the face covering by means of selected skin in regional aesthetic units. *British Journal of Plastic Surgery*, **9**(3), 212–221. (Cited on page 48.)

Gordon L., 1989. *Drawing the Human Head*. B T Batsford. (Cited on pages 30 and 31.)

Gosain A. K., Yousif N. J., Madiedo G., Larson D. L., Matloub H. S. and Sanger J. R., 1993. Surgical anatomy of the SMAS: a reinvestigation. *Plastic and Reconstructive Surgery*, **92** (7), 1254–63; discussion 1264–5. (Cited on pages 43 and 44.)

Green R. D., MacDorman K. F., Ho C.-C. and Vasudevan S., September 2008. Sensitivity to the proportions of faces that vary in human likeness. *Computers in Human Behavior*, **24**, 2456–2474. (Cited on page 20.)

Grima C. I. and Márquez A., editors, 2001. *Computational Geometry on Surfaces*. Kluwer Academic Publishers, Dordrecht, Boston, London. (Cited on page 128.)

Gunz P., Mitteroecker P. and Bookstein F. L. 2005. Semilandmarks in three dimensions. In Slice D. E., editor, *Modern Morphometrics in Physical Anthropology*, Springer. (Cited on pages 81 and 113.)

Ha R. Y., Nojima K., Adams W. P. and Brown S. A., 2005. Analysis of facial skin thickness: defining the relative thickness index. *Plastic and Reconstructive Surgery*, **115**(6), 1769–73. (Cited on page 43.)

Haber J. and Terzopoulos D., 2004. Facial modeling and animation. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Course Notes*, New York, NY, USA. ACM, 6. (Cited on page 7.)

Hamra S., 1990. The deep-plane rhytidectomy. *Plast Reconstr Surg*, **86**(1), 53–61; discussion 62–3. (Cited on page 148.)

Hannam A. G. and McMillan A. S., 1994. Internal organization in the human jaw muscles. *Critical Reviews Oral Biology and Medicine*, **5**(1), 55–89. (Cited on page 31.)

Har-Shai Y., Bodner S., Egozy-Golan D., Lindenbaum E., Ben-Izhak O., Mitz V. and Hirshowitz B., 1997. Viscoelastic properties of the superficial musculoaponeurotic system (SMAS): a microscopic and mechanical study. *Aesthetic Plastic Surgery*, **21**(4), 219–24. (Cited on page 43.)

Har-Shai Y., Sela E., Rubinstien I., Lindenbaum E. S., Mitz V. and Hirshowitz B., 1998. Computerized morphometric quantitation of elastin and collagen in SMAS and facial skin and the possible role of fat cells in smas viscoelastic properties. *Plastic and Reconstructive Surgery*, **102**(7), 2466–70. (Cited on page 43.)

Hartmann E., 1998. A marching method for the triangulation of surfaces. *The Visual Computer*, **14**(3), 95–108. (Cited on page 125.)

Havaldar P., 2006. Performance driven facial animation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, New York, NY, USA. ACM Press, 23–42. (Cited on page 10.)

Hellard P., 2010. Driving the avatar characters. `http://features.cgsociety.org/story_custom.php?story_id=5501`. Date of access: 22/05/2010. (Cited on page 15.)

Hollinshead H. W., 1968. *Anatomy for Surgeons: Volume 1. The Head and Neck*. Harper and Row, Philadelphia. (Cited on page 31.)

Hu K.-S., Jin G.-C., Youn K.-H., Kwak H.-H., Koh K.-S., Fontaine C. and Kim H.-J., 2008. An anatomic study of the bifid zygomaticus major muscle. *J Craniofac Surg*, **19**(2), 534–6. (Cited on page 42.)

Isaaks E. H. and Srivastava R. M., 1989. *An Introduction to Applied Geostatistics*. Oxford University Press. (Cited on page 51.)

Jones M. W., 2001. Facial reconstruction using volumetric data. In *Vision, Modeling, and Visualization*, 135–142. (Cited on page 148.)

Kähler K. *A Head Model with Anatomical Structure for Facial Modeling and Animation*. PhD thesis, Universität des Saarlandes, May 2003. (Cited on pages 12 and 18.)

Kähler K., Haber J. and Seidel H.-P., 2001. Geometry-based muscle modeling for facial animation. In *Graphics Interface 2001*, Toronto, Ont., Canada, Canada. Canadian Information Processing Society, 37–46. (Cited on pages 12, 16, and 19.)

Kähler K., Haber J. and Seidel H.-P., 2003. Reanimating the dead: reconstruction of expressive faces from skull data. In *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, New York, NY, USA. ACM, 554–561. (Cited on page 148.)

Kähler K., Haber J., Yamauchi H. and Seidel H.-P., July 2002. Head shop: Generating animated head models with anatomical structure. In Spencer S. N., editor, *Proceedings of the 2002 ACM SIGGRAPH Symposium on Computer Animation*, San Antonio, USA. Association of Computing Machinery (ACM), ACM SIGGRAPH, 55–64. (Cited on pages 18 and 107.)

Keeve E., Girod S., Pfeifle P. and Girod B., 1996. Anatomy-based facial tissue modeling using the finite element method. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, Los Alamitos, CA, USA. IEEE Computer Society Press, 21–ff. (Cited on pages 13 and 24.)

Keller G., 1997. *Endoscopic facial plastic surgery*. Mosby. (Cited on pages 9 and 45.)

Kent J. and Mardia K. 1994. 24, 325–339. The link between kriging and thin-plate splines. In Kelly F. P., editor, *Statistics and Optimization*, John Wiley & Sons. (Cited on pages 51 and 59.)

Kimmel R. and Sethian J. A., July 1998. Computing geodesic paths on manifolds. *Proceedings of the National Academy of Sciences of the United States of America*, **95**(15), 8431–8435. (Cited on page 87.)

Koch R. M., Roth S. H. M., Gross M. H., Zimmermann A. P. and Sailer H. F., 2002. A framework for facial surgery simulation. In *SCCG '02: Proceedings of the 18th spring conference on Computer graphics*, New York, NY, USA. ACM, 33–42. (Cited on page 24.)

Koch R. M., and Bosshard A. A., 1998. Emotion editing using finite elements. *Computer Graphics Forum*, **17**(3). (Cited on pages 13 and 16.)

Koch R. M., Gross M. H., Carls F. R., von Büren D. F., Fankhauser G. and Parish Y. I. H., 1996. Simulating facial surgery using finite element models. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, New York, NY, USA. ACM, 421–428. (Cited on page 24.)

Krinidis S., Buciu I. and Pitas I., 2003. Facial expression analysis and synthesis: A survey. In *10th International Conference on Human-Computer Interaction (HCI) 2003*, 22–27. (Cited on page 7.)

Krogman W. M., 1962. *The human skeleton in forensic medicine*. Springfield, Ill. C.C. Thomas. (Cited on page 29.)

Larrabee W. F. and Galt J. A., 1986. A finite element model of skin deformation. iii. the finite element model. *Laryngoscope*, **96**(4), 413–419. (Cited on page 13.)

Larrabee W. F., Makielski K. H. and Henderson J. L. *Surgical Anatomy for Endoscopic Facial Surgery.*, 1, 3–33. Mosby, 1997. (Cited on page 44.)

Larrabee W. F., Makielski K. H. and Henderson J. L. *Superficial Musculoaponeurotic System*, 6, 49–57. Lippincott Williams and Wilkins, 2008. (Cited on page 44.)

LaTrenta G. S., 2004. *Two-Layer Approches to the Midface and Neck*. Saunders. (Cited on page 148.)

Lee H., Kim L., Meyer M. and Desbrun M., 2001. Meshes on fire. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, New York, NY, USA. Springer-Verlag New York, Inc., 75–84. (Cited on page 89.)

Lee H., Tong Y. and Desbrun M., January 2005. Geodesics-based one-to-one parameterization of 3d triangle meshes. *IEEE MultiMedia*, **12**, 27–33. (Cited on page 89.)

Lee S., Han J. and Lee H., 2006. Straightest paths on meshes by cutting planes. In *Geometric Modeling and Processing (GMP)*, Berlin, Heidelberg. Springer-Verlag, 609–615. (Cited on pages 89, 90, and 105.)

Lee S. and Lee H., 2007. Parameterization of 3d surface patches by straightest distances. In *Proceedings of the 7th international conference on Computational Science, Part II*, ICCS '07, Berlin, Heidelberg. Springer-Verlag, 73–80. (Cited on page 89.)

Lee Y., Terzopoulos D. and Waters K., May 1993. Constructing physics-based facial models of individuals. In *Graphics Interface '93*, Toronto, ON. 1–8. (Cited on page 18.)

Lee Y., Terzopoulos D. and Waters K., August 1995. Realistic modeling for facial animation. In *Computer Graphics* Proceedings, Annual Conference Series, Proc. SIGGRAPH '95 (Los Angeles, CA). ACM SIGGRAPH, 55–62. (Cited on page 18.)

Lewis J. P., 2005. Audience perception of clone realism. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, New York, NY, USA. ACM, 7. (Cited on page 22.)

Lewis J. P., Cordner M. and Fong N., 2000. Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '00, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co., 165–172. (Cited on page 148.)

Liggett J., 1974. *The human face*. Constable, London. (Cited on page 122.)

Little J. A. *Generalized Kriging with Medical Imaging Applications*. PhD thesis, The University of Leeds, 1995. (Cited on page 78.)

Lorach T., 2007. GPU blend shapes. Technical report, nVIDIA. (Cited on page 8.)

Lorensen W. E. and Cline H. E., July 1987. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, **21**(4), 163–169. (Cited on page 125.)

MacDorman K. F., Green R. D., Ho C.-C. and Koch C. T., 2009. Too real for comfort? uncanny responses to computer generated faces. *Comput. Hum. Behav.*, **25**(3), 695–710. (Cited on page 22.)

Macêdo I., Gois J. a. P. and Velho L., 2009. Hermite interpolation of implicit surfaces with radial basis functions. 1–8. (Cited on pages 74 and 75.)

Maddock S., Edge J. and Sanchez M., 2005. Movement realism in computer facial animation. In *Proceedings of the Workshop on Human-animated Characters Interaction at The 19th British HCI Group Annual Conference HCI 2005*. (Cited on page 22.)

Manhein M. H., Listi G. A., Barsley R. E., Robert Musselman E. N. B. and Ubelaker D. H., 2000. In vivo facial tissue depth measurements for children and adults. *Journal of forensic sciences*, **45**(1), 48–60. (Cited on pages 11, 107, 108, and 109.)

Matheron G. 1981. 77–95. Splines and Kriging; their formal equivalence. In Merriam D. F., editor, *Down-to-Earth Statistics: Solutions looking for Geological Problems*, Syracuse University Geology Contributions. (Cited on page 51.)

Maurel W., Wu Y., Magenat-Thalmann N. and Thalmann D., 1998. *Biomechanical Models for Soft Tissue Simulation*. Springer, Berlin. (Cited on page 43.)

Mendelson B. C., 1995a. Extended sub-SMAS dissection and cheek elevation. *Clinics in Plastic Surgery*, **22**(2), 325–39. (Cited on page 47.)

Mendelson B. C., 1995b. Extended sub-SMAS dissection and cheek elevation. *Clinics in plastic surgery*, **22**(2), 325–39. (Cited on page 148.)

Mendelson B. C., 2001. Surgery of the superficial musculoaponeurotic system: principles of release, vectors, and fixation. *Plastic and Reconstructive Surgery*, **107**(6), 1545–52; discussion 1553–5, 1556–7, 1558–61. (Cited on pages 46, 47, and 48.)

Mendelson B. *Advances in Understanding the Surgical Anatomy of the Face*, 18, 145–149. Springer, 2008. (Cited on pages 45 and 48.)

Mendelson B. *Facelift anatomy, SMAS, retaining ligaments and facial spaces*, 6, 53–72. Saunders, 2009. (Cited on pages 9, 42, 43, 44, 45, 46, 48, 49, and 148.)

Mendelson B. C., Freeman M. E., Wu W. and Huggins R. J., 2008. Surgical anatomy of the lower face: the premasseter space, the jowl, and the labiomandibular fold. *Aesthetic Plast Surg*, **32**(2), 185–95. (Cited on page 48.)

Mendelson B. C. and Jacobson S. R., 2008. Surgical anatomy of the midcheek: facial layers, spaces, and the midcheek segments. *Clin Plast Surg*, **35**(3), 395–404; discussion 393. (Cited on pages 46 and 48.)

Mendelson B. C., Muzaffar A. R. and Adams W. P., 2002. Surgical anatomy of the midcheek and malar mounds. *Plastic and Reconstructive Surgery*, **110**(3), 885–96; discussion 897–911. (Cited on page 48.)

Menick F. J., 1987. Artistry in aesthetic surgery. aesthetic perception and the subunit principle. *Clinics in Plastic Surgery*, **14**(4), 723–735. (Cited on page 49.)

Meyer M., Desbrun M., Schröder P. and Barr A. H., 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In Hege H.-C. and Polthier K., editors, *Visualization and Mathematics III*, Heidelberg. Springer-Verlag, 35–57. (Cited on page 73.)

Migliore M., Martorana V. and Sciortino F., March 1990. An algorithm to find all paths between two nodes in a graph. *Journal of Compututational Physics*, **87**, 231–236. (Cited on page 101.)

Miller A. J., 1991. *Craniomandibular Muscles: Their Role in Function and Form*. CRC Press, Boca Raton. (Cited on page 31.)

Mitchell J. S. B., Mount D. M. and Papadimitriou C. H., August 1987. The discrete geodesic problem. *SIAM Journal on Computung*, **16**(4), 647–668. (Cited on pages 10, 88, and 101.)

Mitz V. and Peyronie M., 1976. The Superficial Musculo-aponeurotic system (SMAS) in the parotid and cheek area. *Plastic and Reconstructive Surgery*, **58**(1), 80–88. (Cited on pages 43 and 44.)

Moss C. J., Mendelson B. C. and Taylor G. I., 2000. Surgical anatomy of the ligamentous attachments in the temple and periorbital regions. *Plastic and Reconstructive Surgery*, **105**(4), 1475–90; discussion 1491–8. (Cited on page 46.)

Müller M., Heidelberger B., Hennix M. and Ratcliff J., 2007. Position based dynamics. *Journal of Visual Communication and Image Representation*, **18**(2), 109–118. (Cited on page 20.)

Noh J.-Y. and Neumann U., 2001. Expression cloning. In Fiume E., editor, *SIGGRAPH 2001, Computer Graphics Proceedings*. ACM Press / ACM SIGGRAPH, 277–288. (Cited on page 15.)

Noh J. and Neumann U., 1998. A survey of facial modeling and animation techniques. Technical Report 99-705, USC. (Cited on page 7.)

Olea R. A., 1999. *Geostatistics for Engineers and Earth Scientists*. Springer. (Cited on page 51.)

O'Rourke J., 1998. *Computational Geometry in C*. Cambridge University Press, New York, NY, USA, 2nd edition. (Cited on pages 96, 136, 138, and 139.)

Orvalho V. C. T. *Reusable Facial Rigging and Animation: create once, use many*. PhD thesis, Universitat Politècnica de Catalunya, June 2007. (Cited on page 17.)

Pan R., Meng X. and Whangbo T., 2009. Hermite variational implicit surface reconstruction. *Science in China Series F: Information Sciences*, **52**(2), 308–315. (Cited on page 74.)

Pandzic I. S. and Forchheimer R., editors, 2003. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. John Wiley & Sons, Inc., New York, NY, USA. (Cited on page 25.)

Parke F. I. and Waters K., 1996. *Computer facial animation*. A. K. Peters, Ltd., Natick, MA, USA, second edition. (Cited on pages 1 and 7.)

Parke F. I., 1972. Computer generated animation of faces. In *ACM'72: Proceedings of the ACM annual conference*, New York, NY, USA. ACM Press, 451–457. (Cited on page 7.)

Patrinely J. R. and Anderson R. L., 1988. Anatomy of the orbicularis oculi and other facial muscles. *Advances in Neurology*, **49**, 15–23. (Cited on page 31.)

Pensler J. M., Ward J. W. and Parry S. W., 1985. The superficial musculoaponeurotic system in the upper lip: an anatomic study in cadavers. *Plastic and Reconstructive Surgery*, **75**(4), 488–94. (Cited on page 44.)

Perlman S., October 25 2006. Volumetric cinematography: The world no longer flat. White paper, MOVA. (Cited on page 10.)

Pessa J. E., Zadoo V. P., Adrian E. K., Yuan C. H., Aydelotte J. and Garza J. R., 1998a. Variability of the midfacial muscles: analysis of 50 hemifacial cadaver dissections. *Plastic and Reconstructive Surgery*, **102**(6), 1888–93. (Cited on page 47.)

Pessa J. E., Zadoo V. P., Garza P. A., Adrian E. K., Dewitt A. I. and Garza J. R., 1998b. Double or bifid zygomaticus major muscle: anatomy, incidence, and clinical correlation. *Clinical Anatomy*, **11**(5), 310–3. (Cited on page 42.)

Pighin F. and Lewis J. P., editors. *Performance-driven facial animation*, New York, NY, USA, 2006a. ACM Press. (Cited on page 9.)

Pighin F. and Lewis J. P., 2006b. Introduction. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, New York, NY, USA. ACM, 1. (Cited on page 10.)

Platt S. M. and Badler N. I., 1981. Animating facial expressions. In *SIGGRAPH '81: Proceedings of the 8th annual conference on Computer graphics and interactive techniques*, New York, NY, USA. ACM Press, 245–252. (Cited on page 11.)

Podlubny I., 1999. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*. Mathematics in science and engineering. Academic Press. (Cited on page 59.)

Pogrel M. A., Shariati S., Schmidt B., Faal Z. H. and Regezi J., 1998. The surgical anatomy of the nasolabial fold. *Oral Surgery, Oral Medicine, Oral Pathology, Oral Radiology and Eendodontics.*, **86**(4), 410–415. (Cited on page 46.)

Polthier K. and Schmies M. *Straightest Geodesics on Polyhedral Surfaces*, 135–150. Springer-Verlag, Berlin, Heidelberg, 1998. (Cited on pages 10, 87, 88, 89, and 141.)

Polthier K. and Schmies M., 1999. Geodesic flow on polyhedral surfaces. In *Proceedings of Eurographics-IEEE Symposium on Scientific Visualization '99*, 179–188. (Cited on page 89.)

Polthier K. and Schmies M., 2006. Straightest geodesics on polyhedral surfaces. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA. ACM, 30–38. (Cited on page 87.)

Radovan M. and Pretorius L., 2006. Facial animation in a nutshell: past, present and future. In *SAICSIT '06: Proceedings of the 2006 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, Republic of South Africa. South African Institute for Computer Scientists and Information Technologists, 71–79. (Cited on page 7.)

Raskin E. and Latrenta G. S., 2007. Why do we age in our cheeks? *Aesthet Surgery Journal*, **27**(1), 19–28. (Cited on pages 46 and 48.)

Ravyse I. *Facial Analysis and Synthesis*. PhD thesis, Vrije Universiteit Brussel, May 2006. (Cited on page 7.)

Rohr K., 2001. *Landmark-Based Image Analysis: Using Geometric and Intensity Models*. Kluwer Academic Publishers, Norwell, MA, USA. (Cited on pages 57 and 118.)

Rohrich R. J. and Pessa J. E., 2007. The fat compartments of the face: anatomy and clinical implications for cosmetic surgery. *Plastic and Reconstructive Surgery*, **119**(7), 2219–27; discussion 2228–31. (Cited on pages 9, 43, and 44.)

Rohrich R. J. and Pessa J. E., 2008. The retaining system of the face: histologic evaluation of the septal boundaries of the subcutaneous fat compartments. *Plastic and Reconstructive Surgery*, **121**(5), 1804–9. (Cited on page 43.)

Rohrich R. J., Pessa J. E. and Ristow B., 2008. The youthful cheek and the deep medial fat compartment. *Plastic and Reconstructive Surgery*, **121**(6). (Cited on page 47.)

Rubin L., 1974. The anatomy of a smile: its importance in the treatment of facial paralysis. *Plastic and Reconstructive Surgery*, **53**(4), 384–7. (Cited on pages 9, 41, and 42.)

Salas M. and Maddock S. C., 2010. Craniofacial reconstruction based on skull-face models extracted from mri datasets. In Collomosse J. P. and Grimstead I. J., editors, *TPCG*. Eurographics Association, 143–150. (Cited on page 148.)

Schechter M., 2001. *Principles of Functional Analysis. 2nd ed.* American Mathematical Society (AMS). (Cited on page 56.)

Shimada K. and Gasser R. F., 1989. Variations in the facial muscles at the angle of the mouth. *Clinical Anatomy*, **2**(3), 129–134. (Cited on pages 40 and 41.)

Sifakis E., Neverov I. and Fedkiw R., 2005. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph.*, **24**(3), 417–425. (Cited on pages 14, 16, and 123.)

Skoog T., 1974. *Plastic surgery: new methods and refinements*. Saunders. (Cited on page 148.)

Smith J. and Séquin C., October 2003. Differential geometry of surfaces. http://www.cs.berkeley.edu/~sequin/CS284/TEXT/diffgeom.pdf. Date of access: 28/09/2011. (Cited on pages 72 and 73.)

Sonka M., Hlavac V. and Boyle R., 2007. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering. (Cited on page 130.)

Spitzer V., Ackerman M. J., Scherzinger A. L. and Whitlock D., 1996. The visible human male: a technical report. *Journal of the American Medical Informatics Association*, **3**(2), 118–30. (Cited on page 118.)

Srinark T., July 2008. Lecture note on differential geometry. (Cited on page 73.)

Steele D. G. and Bramblett C. A., 1988. *The Anatomy and Biology of the Human Skeleton*. Number 4. Texas A&M University Press. (Cited on page 29.)

Stuzin J. M., Baker T. J. and Gordon H. L., 1992. The relationship of the superficial and deep facial fascias: relevance to rhytidectomy and aging. *Plastic and Reconstructive Surgery*, **89**(3), 441–9; discussion 450–1. (Cited on pages 44, 46, 48, and 49.)

Stuzin J. M., 2007. Restoring facial shape in face lifting: the role of skeletal support in facial analysis and midface soft-tissue repositioning. *Plastic and Reconstructive Surgery*, **119**(1), 362–76; discussion 377–8. (Cited on page 47.)

Surazhsky V., Surazhsky T., Kirsanov D., Gortler S. J. and Hoppe H., 2005. Fast exact and approximate geodesics on meshes. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, New York, NY, USA. ACM, 553–560. (Cited on pages 87 and 101.)

Tao T. Function spaces. Date of access: 21/08/2010, 2008. (Cited on pages 53 and 54.)

Taylor K. T., 2000. *Forensic Art and Illustration*. TF-CRC, second edition. (Cited on page 109.)

Terzopoulos D. and Waters K., 1993. Analysis and synthesis of facial image sequences using physical and anatomical models. *IEEE Trans. Pattern Anal. Mach. Intell.*, **15**(6), 569–579. (Cited on page 18.)

Terzopoulos D. and Waters K., 1990. Physically-based facial modeling, analysis, and animation. *Journal of Visualization and Computer Animation*, **1**(2), 73–80. (Cited on pages 12 and 16.)

Thompson S. and Menick F., 1994. Aesthetic facial reconstruction: blending human perception and the facial subunit theory. *Plastic Surgical Nursing*, **14**(4), 211–6, 224. (Cited on page 49.)

Turk G. and O'Brien J. F., October 2002. Modelling with implicit surfaces that interpolate. *ACM Transactions on Graphics*, **21**, 855–873. (Cited on page 74.)

Vandewalle P., Schutyser F., Van Cleynenbreugel J. and Suetens P., 2003. Modelling of facial soft tissue growth for maxillofacial surgery planning environments. In *IS4TM'03: Proceedings of the 2003 international conference on Surgery simulation and soft tissue modeling*, Berlin, Heidelberg. Springer-Verlag, 27–37. (Cited on page 24.)

Wahba G., 1990. *Spline Models for Observational Data*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics. (Cited on page 51.)

Walker J. H., Sproull L. and Subramani R., 1994. Using a human face in an interface. In *CHI '94: Proceedings of the SIGCHI conference on Human factors in computing systems*, New York, NY, USA. ACM, 85–91. (Cited on page 24.)

Wallraven C., Breidt M., Cunningham D. W. and Bülthoff H. H., 2005. Psychophysical evaluation of animated facial expressions. In *APGV '05: Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, New York, NY, USA. ACM, 17–24. (Cited on page 22.)

Watanabe Y., 2010. Metal gear solid 4 (guns of patriots). `http://features.cgsociety.org/story_custom.php?story_id=4735`. Date of access: 22/05/2010. (Cited on page 23.)

Waters K. and Terzopoulos D., 1991. Modeling and animating faces using scanned data. *The Journal of Visualization and Computer Animation*, **2**(4), 123–128. (Cited on page 18.)

Waters K., 1987. A muscle model for animation three-dimensional facial expression. In *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, New York, NY, USA. ACM Press, 17–24. (Cited on pages 12, 13, 16, 17, and 123.)

Wilkinson C., 2004. *Forensic Facial Reconstruction*. Cambridge University Press, 1 edition. (Cited on page 30.)

Williams L., 1990. Performance-driven facial animation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, New York, NY, USA. ACM Press, 235–242. (Cited on page 9.)

Williams P. L., Warwick R., Dyson M. and Bannister L. H., 1989. *Gray's Anatomy*. Churchhill Livingstone, thirty-seventh edition. (Cited on pages 30, 31, and 34.)

Zelditch M. L., Swiderski D. L., Sheets D. H. and Fink W. L., 2004. *Geometric morphometrics for biologists*. Elsevier Academic Press. (Cited on page 78.)

Zhang Y., Prakash E. C. and Sung E., 2002. Anatomy-based 3d facial modeling for expression animation. *Machine Graphics and Vision*, **11**(1), 53–76. (Cited on pages 12 and 16.)

Zhang Y., Prakash E. C. and Sung E., 2004. A new physical model with multilayer architecture for facial expression animation using dynamic adaptive mesh. *IEEE Transactions on Visualization and Computer Graphics*, **10**(3), 339–352. (Cited on page 12.)

Zhang Y., Sim T. and Tan C. L., 2005. Generating personalized anatomy-based 3d facial models from scanned data. *Machine Graphics and Vision*, **14**(1), 3–28. (Cited on page 19.)

Zhang Y., Sim T., Tan C. L. and Sung E., 2006. Anatomy-based face reconstruction for animation using multi-layer deformation. *Journal of Visual Launguages and Computing*, **17**(2), 126–160. (Cited on page 19.)

Zufferey J., 1992. Anatomic variations of the nasolabial fold. *Plastic and Reconstructive Surgery*, **89**(2), 225–31; discussion 232–3. (Cited on pages 40, 46, and 47.)

Zufferey J., 2003. Modiolus: dynamic angular stone of the nasolabial fold. *European Journal of Plastic Surgery*, **25**(7-8), 352–356. (Cited on page 46.)

Zufferey J. A., 1999. The nasolabial fold: an attempt at synthesis. *Plast Reconstr Surg*, **104**(7), 2318–20; discussion 2321–2. (Cited on page 46.)