CAPS: Computer-Aided Plastic Surgery

by
Steven Donald Pieper
S.M., Massachusetts Institute of Technology (1989)
B.A., University of California at Berkeley (1984)

Submitted to the Media Arts and Sciences Section, School of Architecture and Planning,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
at the
Massachusetts Institute of Technology
February 1992

Author____

Steven Pieper
Media Arts and Sciences Section
September 22, 1991

Certified by____

David Zeltzer
Associate Professor of Computer Graphics
Media Arts and Sciences Section

Accepted by____

Stephen A. Benton
Chairperson, Departmental Committee on Graduate Students

CAPS: Computer-Aided Plastic Surgery
by
Steven Donald Pieper
Submitted to the Media Arts and Sciences Section,
School of Architecture and Planning
on September 22, 1991,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract

This thesis examines interactive computer graphics and biomechanical engineering analysis as components in a computer simulation system for planning plastic surgeries. A new synthesis of plastic surgery planning techniques is proposed based on a task-level analysis of the problem. The medical goals, procedures, and current planning techniques for plastic surgery are reviewed, and from this a list of requirements for the simulator is derived. The requirements can be broken down into the areas of modeling the patient, specifying the surgical plan, and visualizing the results of mechanical analysis. Ideas from computer-aided design and the finite element method are used to address the problem of patient modeling. Two- and three-dimensional computer graphics interface techniques can be used to interactively specify the surgical plan and the analysis to perform. Three-dimensional computer graphics can also be used to display the analysis results either in an interactive viewing system or through the generation of animated sequences on video tape. The prototype system combines modeling, interaction, and visualization techniques to allow iterative design of a surgical procedure. Several clinicians were shown the prototype system and were interviewed about its applicability. Text of these interviews is included in this thesis. The clinicians' comments indicate that the CAPS system represents a new approach to plastic surgery planning and is viable for clinical application.

Thesis Supervisor: David L. Zeltzer
Title: Associate Professor of Computer Graphics
This work was supported by NHK and by equipment grants from Hewlett-Packard.

Thesis Committee

Chairman____

David Zeltzer
Associate Professor of Computer Graphics
MIT Media Arts and Sciences Section

Member____

Joe Rosen
Associate Professor in Plastic and Reconstructive Surgery
Dartmouth Medical School

Member____

Robert Mann
Whitaker Professor of Biomedical Engineering
MIT Department of Mechanical Engineering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Contributions of this Work

I have implemented a prototype Computer-Aided Plastic Surgery (CAPS) system in which an operation can be planned, analyzed, and visualized.[1] In the CAPS system, the *planning* of the operation is accomplished by selecting incision paths, regions of tissue to remove, and suture connections using an interactive computer graphics system. The operation is performed directly on a 3D graphical model reconstructed from scans of the patient. *Analysis* of the operation uses the Finite Element Method (FEM) and a generic model of the physical properties of the soft tissue in order to estimate the biomechanical consequences

[1] The term CAPS is an extension of the term Computer-Aided Surgery (CAS), coined by Mann to refer to systems which use engineering analysis and computer modeling to assist in surgical planning and evaluation[43].

of the proposed surgery. *Visualization* includes 3D color rendering, animation, and hardcopy images of the surgical plan and the analysis results.

Computerized planning represents an important development for surgeons because their current techniques do not allow iterative problem solving. Today, a surgeon must observe and perform many operations to build up the experience about the effect of changes in the surgical plan. Since each of these operations is unique, it is difficult to isolate the effects of different surgical options since the result is also influenced by many patient specific variables. The CAPS system allows exploration of the various surgical alternatives with the ability to modify the existing plan, or to create a new plan from scratch. This process may be repeated as many times as needed until the surgeon is satisfied with the plan.

The thesis work contributes to the field of computer graphics in the following areas:

- I present an analysis of Computer-Aided Surgery (CAS) systems based on the *patient model*, the way the patient model and a description of the operation are *input* to the system, and how the data about the patient and data about the operation are *output* by the system. This analysis is used to justify my design of the CAPS prototype.
- I present a solid modeling representation of human soft tissue which supports mechanical simulation and which can be derived from scans of the patient.
- I have developed techniques for *in situ* planning of plastic operations using interactive graphics techniques on the reconstructed patient model. The automatic mesh generation algorithm used in the prototype system works directly from the patient scan and a description of the surgery so the user is not burdened with the details of the simulation technique. This provides a task-level interface for the application[79].

- I have evaluated the practicality of this CAPS system by simulating a small set of standard plastic procedures and by having practicing surgeons use and critique the system.
- I discuss the hardware limitations of a computer graphics and finite element simulation approach to surgical planning.

In this thesis work, I have focused on the problem of planning plastic operations with the understanding that a simulation system that can be used successfully for planning would also be extremely valuable for training surgeons. I also note that the techniques described here, with extensions, would be applicable to planning and training in other areas of surgery as well.

1.2 This Work in Relation to Previous Approaches

The CAPS system represents a unique view of about the way computer systems should be used in surgical planning. This is because it follows a *task level* analysis of the goals of plastic surgery. Previous work has concentrated on either building mechanical models of the soft tissue which can analyze specific test cases, or on imaging systems which present renderings of the patient. This work brings those two components together in a graphical simulation environment where the simulation procedures are attached to the graphical model --- this combination allows the surgeon to operate on the graphical model in a manner directly analogous to operating on the real patient.

This approach is crucial for the successful clinical application of mechanical analysis of soft tissue because the surgeon has neither the time nor the training to formulate a specific surgical case at the level of detail required for analysis without the assistance of a computer graphics tool. Previous research in biomechanical analysis of plastic surgery has not included methods for automatically converting a surgical plan into a form appropriate for the analysis programs. For example, In her work on analysis of plastic surgeries, Deng describes a system in which the user is required to type an input file which describes the incision geometries, regions of tissue to simulate, and constraint conditions on the tissue in terms of their world space coordinates[16]. Kawabata *et. al.* describe an analysis of surgical procedures but describe no method for automatically analyzing a particular plan. Larrabee discusses the problem of modeling arbitrary incision geometries using graphical input devices, but the solution he proposes requires the user to define each of the dozens of analysis nodes and elements. While Larrabee's approach is useful for small two-dimensional analyses (which is the way Larrabee used it), the approach becomes unmanageable for three-dimensional structures with a greater numbers of nodes.

In the context of computer animation, the difficulties associated with describing in detail all the parameters of a simulation are referred to as the *degrees of freedom* problem. *Task level* analysis of the problem refers to the process of chunking these parameters into manageable subpieces which are at the correct level of abstraction for the given problem, and then providing appropriate computer interaction techniques for control[80]. This approach is applied to the problem of plastic surgery planning by identifying the specific variables which determine the plan, namely the lines of incision. The incision lines are specified directly on the reconstructed graphical object representing the patient, and all subsequent analysis is performed automatically.

This approach differs from past research in computer graphics applied to surgical planning because the graphics model is attached to the simulation model. Previous computer graphics work has emphasized special purpose rendering algorithms for visualization of data obtained from volumetric scans of the patient[17,41,53,75,76], or geometric methods for extracting and repositioning pieces of the volume data[1,14,25,77].

The CAPS approach offers significant advantages over current methods of planning because it includes both a database of information about the anatomy and physical properties of the patient's tissues and an analysis technique which can predict the behavior of those tissues in response to certain types of surgical intervention. The CAPS approach also allows iterative exploration of the surgical design options and documentation of the selected plan. In contrast, current planning techniques rely solely on the surgeon's experience and do not allow systematic examination of the surgical alternatives. The first section of this thesis discusses the planning problem and develops the requirements for the CAPS system.

1.3 Overview

The approach described here relies on an integration of several computer simulation and interaction techniques which have only over the past decade been made practical by the increase in computational performance and decrease in memory costs. Interactive computer graphics, including both 2D graphical user interface (GUI) techniques (made popular by, for example, the Apple Macintosh interface) and direct manipulation of three-dimensional graphics (as used, for example, in solid modeling and animation programs), is becoming a practical method for specifying, controlling, and visualizing complex simulations. The finite element method (FEM) is rapidly becoming the method of choice for simulations of the mechanical behavior of objects and structures in mechanical and civil engineering, where it was developed to numerically solve the differential equations of elasticity and related problems. Both graphics and FEM demand significant computational resources as the simulations become increasingly complex and realistic. In the coming years, we can expect to see further improvements in computer technology which will make it possible to simulate more more accurate models using graphics and FEM. Improved computer hardware will also shorten simulation time and allow these techniques to be more easily incorporated into a clinical setting. The second section of this thesis describes these techniques and how they have been applied in the CAPS prototype.

A system which proposes to guide the surgeon's knife bears a special responsibility to prove its validity before it is applied clinically. The computational requirements alluded to above place limits on the level of detail which can be included in the CAPS system on current hardware. Great care must be taken to assure that the analysis provides meaningful predictions of the surgical outcome and that the assumptions which went into the analysis are clear to the user of the system. Only when these two requirements have been satisfied can the system be added to the set of tools used by the surgeon. The third section of this thesis shows a number of examples which demonstrate the behavior of the simulation prototype under a range of inputs.

1.4 Limitations of the Physical Model

Physical modeling of human soft tissue presents many challenges which can only be addressed by making simplifying assumptions about the behavior of the tissue. The complexity of the tissue includes the fact that it is alive, that it has a complex structure of component materials, and that its mechanical behavior is nonlinear. The design of the CAPS system is an attempt to model those features of the tissue which have direct bearing on the outcome of plastic surgery, but in doing so it ignores the following effects: the physiological processes of healing, growth, and aging are not included in the model; the multiple layers of material which make up the skin are idealized as a single elastic continuum; and the system uses only a linear model of the mechanical behavior of the tissue, and does not include a model of the pre-stress in the tissue (i.e., the skin does not open up when cut). Under these assumptions, the model gives an estimate of the instantaneous state of the tissue after the procedure has been performed.

Within the computational framework of the CAPS system, these assumptions could be relaxed to build a more complete model of tissue behavior. The complex structure of the tissue could be addressed by creating a more detailed finite element mesh with multiple layers of differing material properties. The nonlinear mechanical response of the tissue could be better approximated using a nonlinear finite element solution technique. Both of these improvements will make the solution process more computationally complex, but will become more feasible as computers become faster. A series of clinical trial should be performed to identify the parameters which have the most influence in the surgical result.

The incorporation of physiological processes presents a more fundamental problem, since the processes themselves are not well understood. In this realm, the physical modeling approach offers a possible method for determining the action of these processes. For example, if the physical model is calibrated such that it gives a nearly exact prediction of the immediate post-operative state of the tissue, then subsequent changes in the patient's skin due to healing could be determined by changing the material property assumptions of the model until it again matches the skin. It is possible that this analysis would lead to a method of predicting the effect of healing which could then be included in the planning system.

1.5 Summary of Clinical Evaluation

An important aspect of the CAPS system is that it is designed to be used directly by the surgeon in planning the operation. To meet this goal, a number of surgeons have been consulted during the implementation. After the prototype was completed, the system was presented to surgeons who were asked to consider a set of questions about the use of the system in a clinical setting. The questions and the text of the surgeons' responses are given in section 5.6. Overall, the results of this evaluation process have been very positive.

The surgeons noted that current planning tools do not account for the elasticity of the soft tissue and thus rely on the surgeon's experience to predict the result of the procedure. The use of the finite element method to simulate the tissue was accepted as a valuable technique for quantifying the prediction and providing an objective basis for comparing surgical options. The surgeons indicated that the preliminary results obtained by the CAPS system showed good agreement with their prediction of the surgical results. However it is clear that direct comparison of the simulation results with the actual case histories will be needed in order to verify the model.

The interface to the system appears to be adequate for the task of simulating the set of procedures described in this thesis, but for more complex procedures, the interface must be extended to allow more general incision and suturing options. The surgeons indicated that the system will be much more valuable if it can be used to plan more complex and patient specific operations for which the standard procedures must be extensively modified. It is in these more complex procedures that the planning times become a significant portion of total time of the procedure. The surgeons indicated that the mouse-based graphics user interface used in the CAPS system would be well accepted by surgeons for the planning task.

1.6 Organization

The remainder of the thesis is organized as follows. The next chapter is devoted to a discussion of plastic surgery and a review of other research on CAS systems; the chapter concludes with the beginning of a scenario dramatizing the practical need for a CAS system for plastic surgery. Chapter three reviews the technologies available to implement a CAPS system. It is broken down into three sections which describe in detail the three major components of a CAPS system in terms of requirements, the available hardware and software technologies, related work in the area, and the implementation choices for the thesis project. Section 3.1 discusses techniques for modeling the patient --- in terms of both shape description and mechanical analysis. Section 3.2 discusses input to the model for defining the patient and for defining the operation. Section 3.3 deals with display of the patient model, the surgical plan, and the analysis results on various output media. Chapter four describes how the model, input, and output techniques have been integrated into the CAPS system. Chapter five presents the results of a number of experiments run using the system. Chapter five also looks at the problem of evaluating the CAPS project in terms of its ability to accurately model the patient and the applicability of the system in a clinical setting. The results of interviews with practicing surgeons are presented in chapter five. Chapter six summarizes the discussion of the CAPS prototype and discusses how future systems can benefit from and extend the work presented here. Problem areas which require further research are also identified and discussed in chapter six.

1.7 A Note on the Figures

Figures in this thesis showing the output of the CAPS system show basically what the screen looks like, but the quality of the figures is significantly reduced. They were prepared by reading the frame buffer of the Hewlett-Packard TurboSRX graphics accelerator which runs the application software for the interactive system. The actual resolution shown on the monitor is 1280 pixels wide by 1024 pixels high, with eight bits of color resolution each for red, green, and blue. These images were reduced to gray scale using the equation $gray = 0.3\ red + 0.59\ green + 0.11\ blue$ which is the standard NTSC luminance encoding[21][2]. The gray scale images were then reduced to one bit halftone images of 50 lines per inch at a 45 degree angle. Halftoning operations were performed using the

[2] NTSC stands for National Television Standards Committee, which defined the standard for color television broadcasts. The luminance encoding defines the "black and white" portion of the color signal.

Adobe Photoshop image processing program on the Apple Macintosh computer. The images shown here were printed on a laser printer with 300 dots per inch resolution.

Figures whose captions contain bibliography references are reproduced from the indicated works. These figures were scanned at 300 dots per inch using a Hewlett-Packard ScanJet. Unreferenced line drawings were prepared using the *idraw* drawing program written by John Vlissides and John Interrante. This document was formatted using the LaTeX document preparation system[38].

Chapter 2

Problem Domain and Requirements

> The proper application of biomechanics by surgeons has saved the lives of many patients but unfortunately many have been hurt and some have died when biomechanics was ignored or misapplied.
>
> *Richard M. Peters* [60]

> The goal of plastic surgery is to create a proper contour by making the best distribution of available materials. Operations take place on relatively limited surface areas and, in local procedures, skin cover is not brought from distant areas.[1] Rather, skin should be borrowed and redistributed in the area where the operation is being carried out. In this way, surgeons should be able to perform typical plastic operations that will restore proper form to distorted surfaces. Different maneuvers are used in various combinations as either simple or complex figures. The location, form, and dimensions of the incisions necessary for plastic redistribution of tissues determine the plan of the operation.
>
> *A. A. Limberg, M.D.* [42]

[1] In contrast to skin grafting operations.

2.1 Plastic Operations

Planning a plastic operation on the soft tissue of the face is difficult due to the complex geometries, the biomechanics of the tissue, and the conflicting medical goals. This chapter discusses these problems and the traditional approaches to used in solving them.

Medical Goals In performing plastic or reconstructive surgery, the surgeon is faced with a task of lessening the state of abnormality in the patient by reshaping the patient's tissue.[2] In so doing, the surgeon must attempt to satisfy a range of medical *goals* including:

- correct surface geometry;
- positioning scars along anatomical boundaries;
- proper range of motion and muscle attachment;
- physiologically normal stress levels for tissue viability;
- strength of underlying hard tissue;
- matching tissue properties of donor and recipient sites;
- minimal distortion of donor site;
- sufficient blood supply and proper innervation; and
- additional cosmetic considerations such as skin color and hair type.

[2] Material in this section was compiled from standard texts in plastic and reconstructive surgery and discussions with practicing surgeons[2,6,27,42,49,64].

The technique of reshaping tissue is applied in order to correct physical abnormalities created by trauma, tumor resection,[3] or congenital defect. The surgical procedure must be robust in the face of ongoing or potential physiological processes in the patient such as growth, disease, and aging. In addition to the above considerations, the surgeon must be able to carry out the surgical techniques quickly and efficiently in a manner that does not itself damage the tissue.

Surgical Plan The primary technique to meet the medical goals listed above is the generation of the surgical plan. The *surgical plan* is important for a number of reasons:

- to specify the steps to be taken during the operation;
- to fully inform the patient of the procedure;
- to coordinate the surgical team;
- to document treatment (for future treatment and/or litigation); and
- to evaluate treatments (i.e., compare pre- and post-op conditions).

Procedures Used The most common plastic operation is the double-convex lenticular, or "surgeon's ellipse," to close an opening caused by tumor resection (see figure 2.1)[27]. This shape is used in order to minimize the distortion of the tissue at the corners of the closure, known as "dog's ears." Planning this procedure requires selection of the lengths of the major and minor axes of the lenticular, the orientation of the major axis, and the curvature of the lenticulars. These parameters are selected with respect to the mechanical

[3] Cutting away of abnormal tissue.

Elliptical excision. If the ellipse is too short (A), dog ears (arrows) will form at the ends of the closed wound. The correct method is shown in B.

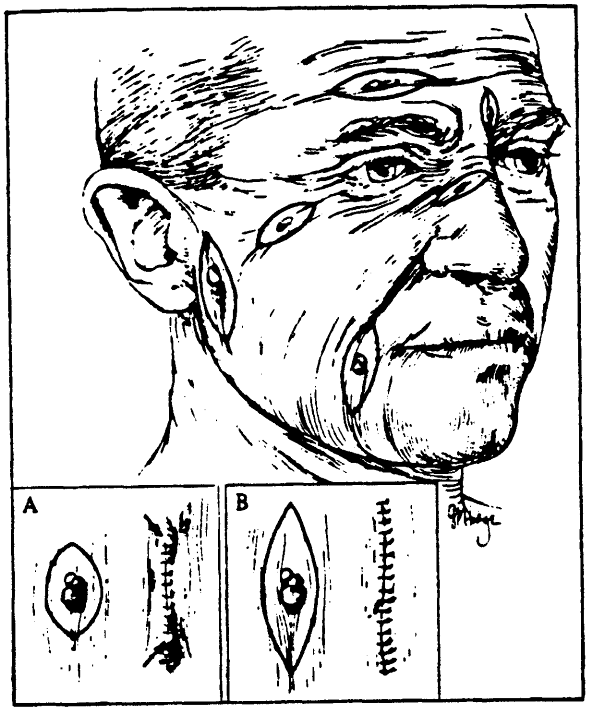


Figure 2.1: Example elliptical excisions[27]. Note that the orientation of the ellipse is chosen so that the scar lies along the natural lines of the face.

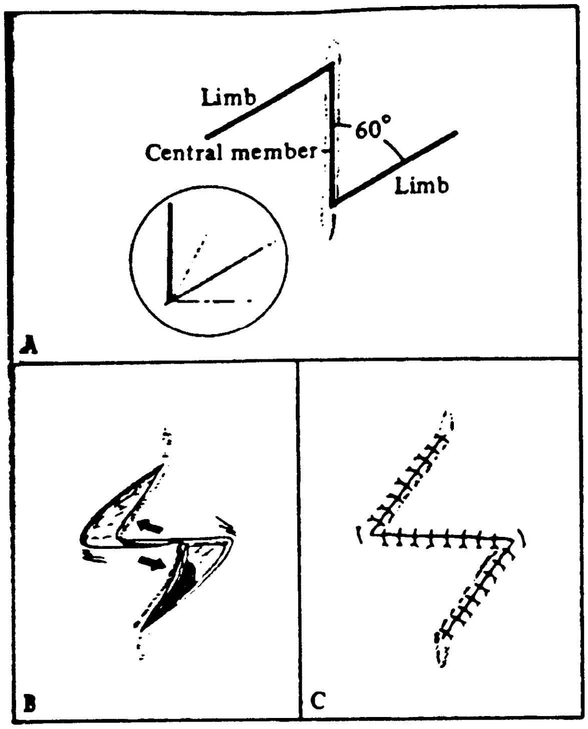
properties of the patient's tissue (depending on age and state of health) and the natural lines of the skin. The natural lines of the face include lines of pre-stress in the skin (Langer's lines), wrinkle lines, and contour lines (at the juncture of skin planes such as where the nose joins with the cheek). Although this technique is widely used, no standard procedure exists for selecting the parameters of the operation. In fact, the guidelines for the choice of these parameters have been developed through clinical experience and there appears to be no theoretical justification for the convex lenticular shape itself[6].

Another common operation is the Z-plasty (see figure 2.2). This procedure involves a

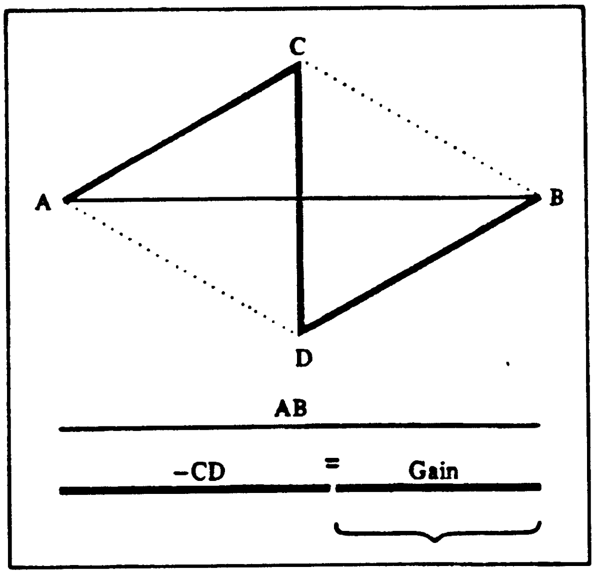
Z-shaped incision with undermining and transposition of the two resulting triangular flaps. This procedure has the effect of exchanging length of the central member of the Z with the distance between the tips of the limbs. The procedure can be used to relieve excess stress along an axis by bringing in tissue from a less stressed region. The parameters of the procedure include the length of the members of the Z, the angle between the central member and each of the limbs, and the orientation of the Z. These are selected with respect to the existing state of stress in the tissue, the mechanical properties of the tissue and the lines mentioned above.

The Z-plasty is one example of the more general technique of *flap transposition* in which a variety of incision geometries can be used to rearrange tissue. For a given patient, a combination of excision and flap transposition may be required to obtain optimal results[42].

Current Planning Techniques The most widely used methods for planning operations currently rely on the training and skill of the surgeon to select procedures appropriate for a given patient. Even in futuristic scenarios of telerobotic surgery and "smart surgical tools," the planning stage remains a task performed by humans[24]. Development of new simulation systems should therefore be guided by an understanding of the planning techniques that surgeons currently find useful. The history of tools for surgical planning is long, but straightforward. The primary techniques of making clay models and drawings have been in use for thousands of years and are still practiced today[45]. In spite of their utility, these techniques represent gross approximations to the state of the actual patient, and their utility is correspondingly limited. The following list outlines the major practices in use today for planning soft tissue surgery. Once a plan is determined, it is transferred to the patient by drawing the incision lines on the skin surface using a sterile marking pen



The classic 60-degree-angle Z-plasty. Inset shows method of finding the 60-degree angle by first drawing a 90-degree angle, then dividing it in thirds by sighting. The limbs of the Z must be equal in length to the central member.



Calculating the theoretical *gain in length of the Z-plasty. The theoretical gain in length is the difference in length between the long diagonal and the short diagonal. In actual practice the gain in length has varied from 45 percent less to 25 percent more than calculated geometrically, due to the biomechanical properties of the skin [283, 284].*

Z-plasty, Angles, and the Theoretical Gain in Length

Angles of Z-plasty	Theoretical Gain in Length
30°–30°	25 percent
45°–45°	50 percent
60°–60°	75 percent
75°–75°	100 percent
90°–90°	120 percent

Figure 2.2: Example Z-plasty plans[27].

(see figure 2.3).

- The surgeon may make a storyboard of drawings of the patient at various stages of the planned operation in order to organize the procedure and study the results. For simple procedures, the plan is drawn directly on the patient's skin.
- Clay models are used to estimate the amount of tissue required to perform reconstructive operations and to experiment with geometries for the reconstructed features. Clay can be plastically shaped to design the new feature and then flattened to make a pattern for cutting the tissue that will be used for the feature. This technique is applied, for example, in the reconstruction of a nose using a flap of skin from the forehead.
- String, wire, or other measuring tools are used to match the incision lengths as required, for example, in the Z-plasty operation.
- Paper models can be used to design plastic operations on the body surface by performing the procedure and observing the resulting formation of cones and creases in the surface[42].
- The procedure may be practiced on animals or cadavers. However, animal models do not represent accurate human anatomy, and cadavers only approximate the conditions in the living patient and are generally in short supply. In addition, these techniques cannot be used to represent a particular patient, who may have rare congenital defects or a unique wound. This approach does not allow the surgeon to go back to the beginning to explore different planning options.

The primary advantage of these techniques is that they provide a familiar environment for the surgeon to work through the surgical plan and develop an intuition about the procedure. A computer-based simulation system should also allow the surgeon to test various alternatives and study their effect on the patient.

Basic Requirements of a Simulator The CAPS system is a testbed in which the surgeon can work through various alternatives in order to create a surgical plan which best satisfies the medical goals listed above. The advantages of the CAPS system can be broken down into the following four categories:

- **Geometric model:** The CAPS system provides a computer graphic model of the patient that the surgeon can use as a tool for thinking about the operation. By analogy to the storyboard planning technique discussed above, the surgeon can use his or her expert knowledge to predict the outcome while using the computer graphics system as a 3D drawing pad. An example of a geometric planning tool for bone surgery in current use involves the creation of a plastic model of the bones to be operated on. This plastic model can then be cut to simulate the procedure or it may be used as a template for a bone graft or for an artificial implant.
- **Analysis:** The CAPS system can analyze the patient model and the plan to provide the surgeon with feedback about the requirements and outcome of proposed surgery. The geometric model can also be used to measure quantities such as lengths and angles. This type of measurement is used, for example, in reconstructive surgery to match the size and shape of a reconstructed feature to its normal counterpart or to anthropometric norms. More complex analyses can make use of non-geometric information about the patient to predict the behavior of the tissue after surgery.

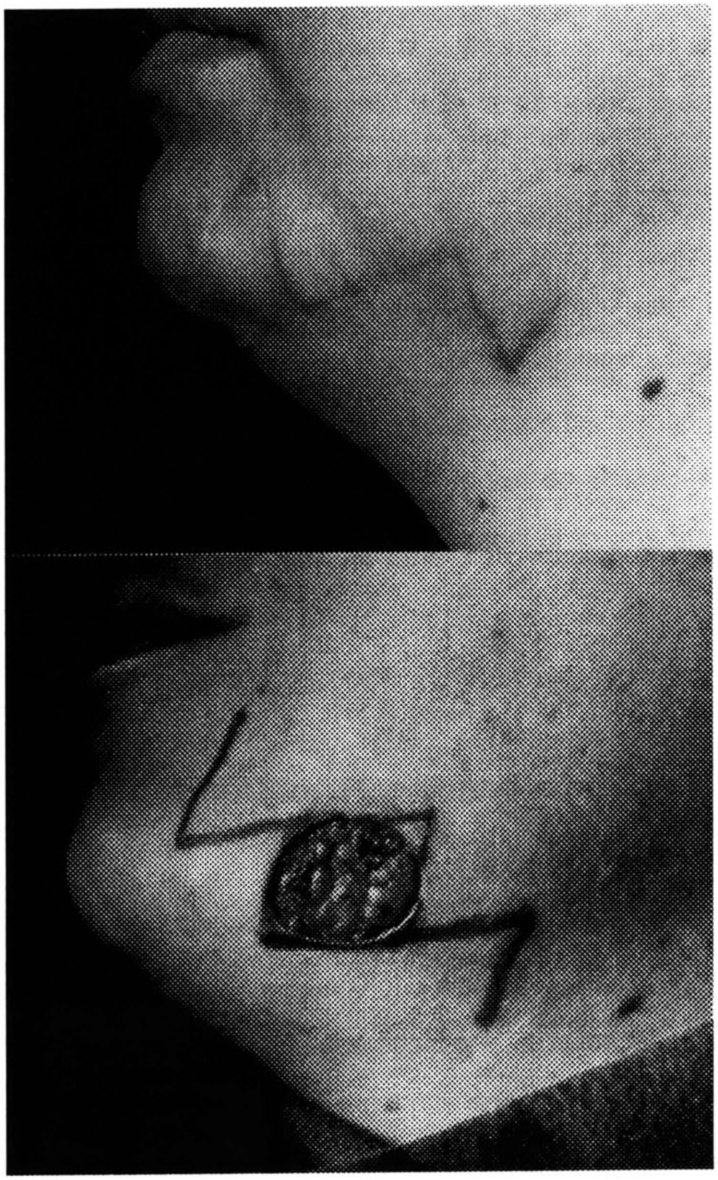


Figure 2.3: Multiple Z-plasty with excision intra-operative (lower) and post-operative (upper). Incision lines drawn with marker on patient skin[6].

Depending on the type of surgery, there may be many different methods of analysis. For plastic surgery, I discuss finite element analysis of soft tissues in section 3.1.2

- **Rehearsal:** The CAPS system provides a platform in which the members of the surgical team can familiarize themselves with the planned procedure before the operation. This allows them to study the plan in the context of the patient scan and organize the sequence of steps to be used in executing the procedure.

- **Documentation:** Finally, the CAPS system provides a mechanism for documenting the surgical plan. The plan itself may include "blueprints" for each stage of the operation, graphical simulations of the procedure to be performed, and observations or measurements that ensure the operation is not deviating from the plan. The plan may include multiple scenarios to handle most problems that arise during the procedure.

2.2 Related Surgical Simulation Research

Although the field is new, a number of investigations have looked at adapting computer techniques to the problem of surgical planning. I classify the techniques as follows: 2D geometric only, expert system-based, 3D geometric only, and mechanical analysis-based. For each technique, I describe the model, the input, and the output. While each of these approaches has unique features, an ideal system would combine the best features of all the approaches. For example, a system based on mechanical analysis should also be able to give geometric information about the model. The CAPS system described in chapter four combines three-dimensional geometric and mechanical simulation approaches.

2.2.1 Two-dimensional, geometric only

This type of planning is essentially a computerized version of the traditional hand-drawn storyboard method of surgical planning. Image processing and computer painting systems are currently used to design rhinoplasties, midface advancements, and mandibular osteotomies[2]. The operation is planned using digitized photographs of the patient, generally side view, and graphical painting tools. The resulting image of the patient is then used to guide the execution of the operation.

Model: changes in the 2D image directly represent proposed changes in the tissue.

Input: drawing operations, primarily "painting" and "scaling" using 2D input devices.

Output: image of proposed surgical result, display of selected length and angles measurements.

2.2.2 Expert system-based

In this approach, the computer system serves as an interactive textbook for training by presenting images and descriptions from case histories. The user of the system is presented with a patient and a choice of standard procedures to apply. The system describes and displays a prediction of the result based on the procedures chosen. All possible results are pre-stored in the simulation system[12].

Model: state transition table specifying the result of each given procedure.

Input: answers to multiple choice questions.

Output: images and text descriptions of patient state.

2.2.3 Three-dimensional, geometric only

The term *volume visualization* refers to the computer graphics techniques used to render the 3D datasets resulting typically from Computed Tomography (CT) and Magnetic Resonance (MR) scans. The primary goal of volume visualization is to help the radiologist communicate diagnostic findings to the physician responsible for treatment[53]. For this purpose, algorithms have been designed which capture the detail of the scan data and make it easier to visually differentiate various tissue types[17,41,75,76]. A number of surgical simulation systems have been based on the segmentation and rearrangement of the volume data. One application of this technique is planning bone surgery such as craniofacial surgery[1,14,77]. The plan is generated either by interactively repositioning portions of the bone or by calculating bone movements or bone graft dimensions that make the injured side of the face match the unaffected side[14,77]. Computer graphics hardware has also been developed to directly render volumentric data[22,25]. Three-dimensional geometric models are also used in planning radiation therapy. For this application, regions of tumor growth are identified in the volume scan of the patient and paths for radiation therapy are planned that provide lethal radiation to the tumor while minimizing the radiation exposure of the normal tissue[65].

Model: 3D image data directly represents the tissue.

Input: scan data, 3D positioning, 3D "erasing" of tissue.

Output: surface or volume renderings of new bone locations, measurements of proposed bone movements, and treatment paths.

2.2.4 Mechanical analysis

In this technique, a physical description of the patient is added to the geometric data and together these serve as input to a program that analyzes relevant mechanical consequences of a proposed surgical procedure. Although I know of no systems based on this technique in clinical use at this time, this procedure is the subject of much research. Delp et al. have built a biomechanical model of the lower extremity musculo-tendon system which can calculate the total moment acting at joint degrees of freedom from the geometry and physiological characteristics of the muscles; this model has been used to analyze tendon transfer operations[15]. Mann and his coworkers have used geometric and physiological models to analyze the impact of surgical intervention on various aspects of human movement including locomotion and manipulation[9,13,43]. Their work has also looked at the mechanics of joint lubrication to design surgical treatments for arthritis[73]. Thompsen et al. use a geometric and mechanical model of the human hand to design tendon transfer surgeries[74]. Balec used a linear finite element model of the femur to analyze hip replacement surgeries[4]. Her system included a model of bone remodeling based on the stress introduced by the implant. Larrabee used a 2D linear finite element model of the skin to analyze wound closures and flap advancements; he found good qualitative agreement between his simulation results and the results of experiments on pig skin[39]. Kawabata et al. used a 2D linear finite element model of skin to analyze the mechanical effect of various Z-plasty parameters[32]. Deng used a 3D non-linear finite element model of skin and the underlying tissues to analyze wound closures[16]. Among the work presented in the literature, Deng's provided the most thorough treatment of the soft tissue modeling problem. Her work is similar to the work presented in this thesis, in that she built a continuum finite element model of the soft tissue and adapted it to sample surface scans of patients. Her work was different, however, in that she did not attempt to build an interface for the surgeon through which a variety of plans could easily be implemented. Also, her work did not address the simulation of muscle action.

Model: biomechanical description of the patient.

Input: 3D specification of the proposed procedure in terms of changes to the patient model.

Output: quantitative data about the physical response of the patient and 3D renderings of the altered patient model.

2.3 A Scenario for the CAPS System: I

It's been a hard year for Molly. Eight months ago, she was doing homework at a desk by the window in her Boston home when a stray assault rifle bullet shattered the left side of her mandible and tore away skin from her cheek. Molly underwent surgery right after the incident in which the left side of her mandible was reconstructed using bone grafts to closely match the undamaged right side, and dental prostheses have restored most of her chewing function.

But 14 year old Molly feels anything but normal. The tissue she lost from her cheek was not restored in the first operation and her face is now highly asymmetric. A ragged scar runs from the left front of her chin nearly to her left ear, and the tightness of the left side of her face pulls down the corner of her left eye (an ectropion) giving her a slight "droopy" look on that side. Molly's parents are convinced that Molly's withdrawn and moody behavior lately is due to her looks.

Dr. Emily Flanders is the plastic surgeon working on Molly's case. In addition to many clinical exams of Molly, Dr. Flanders has ordered an MR scan of Molly's head. From this, Dr. Flanders must generate a plan of the operation. This thesis addresses the tools we might provide Dr. Flanders to help Molly lead a happy life. After discussing the relevant technology, we will return to Dr. Flanders and Molly's case.

Chapter 3

Literature Review

This chapter describes the range of technologies which can be drawn upon to implement a CAPS system. This review is broken down into three main sections covering the patient model, interactive input, and visualization. For each of these topics, a description of the CAPS requirements is derived, followed by a discussion of the options available in the literature for satisfying those requirements. These sections conclude with a statement of which combination of techniques is best suited for the prototype implementation. This chapter provides a context for the detailed description of the implementation given in chapter 4. This chapter discusses a range of options for implementing each of the components, though not all of the simulation techniques, sources of data, interface devices, and output media described here are included in the CAPS prototype.

3.1 Modeling the Patient

The description of the body state has two components: the geometry of the body and the biomechanics of the tissue. Each of these is discussed in more detail below.

3.1.1 Geometric Modeling

Modeling Requirements The primary requirement of the patient model is that it *accurately represent* the geometry of the patient's tissue and the tissue type at each location in the body. *Solid Modeling* refers to the computer representation of shapes and regions for computer simulation. The features of interest for representing the patient are the locations, shapes, and material properties of the body parts, their boundaries (or more accurately their regions of transition), and their connectivity and degrees of freedom. It must be possible to identify these features as the body undergoes transformations due to normal motion of the body and under the influence of surgical procedures. Thus, we must define a *body coordinate system* by which specific locations may be identified and tracked. In addition to the geometric and material descriptions of the patient, the model should also be able to represent information about the physiological dependence of one region of tissue on another. Physiological dependencies include information about the network of blood supply, innervation of tissue, and muscle and tendon action. This information can be incorporated into the patient model by specifying the pathways of each of these networks as a collection of body coordinate control points that move with the body as it is deformed or modified.

An allied requirement to that of accuracy is the requirement of *flexibility*, which states that the modeling technique must be able to represent the range of geometries and material variations to the level of detail required for the given purpose. In the case of patient modeling, the tasks are to visualize the patient's anatomy and to perform mechanical simulations. For visualization it must be possible to render a very detailed model for documentation or for generation of an animation offline, or to generate a less detailed model for use during interactive manipulation. For mechanical analysis we need to select a level of detail in the model that is within the computation and memory limits of the host computer.

The final requirement for the patient model is that it be *modifiable*. For the patient model in a CAPS system, the modifications are cutting (incision and undermining), removal of material (excision), and reconnection of free edges (suturing). In general, it should also be possible to modify the model during the procedure to reflect information unavailable pre-operatively.

Solid Modeling Representations There are three major techniques in traditional solid modeling as it has been applied to computer aided design for manufacturing and analysis. The first two, which are commonly used in design systems, are *boundary representation* and *constructive solid geometry* (CSG)[21,51]. The boundary representation technique uses piecewise planar or curved surfaces to represent the surface of an object that implicitly defines the solid interior. The CSG technique represents solid objects through boolean combinations of solid half-space primitives such as spheres, cubes, cylinders, etc. The boundary representation is useful for describing shapes with complex, detailed surfaces such as organic objects, while the CSG representation is most appropriate for objects with regular geometry interrupted by holes or protrusions (such as a machined part with drilled holes). Both techniques are successful for representing objects to be manufactured using computer controlled machine tools or injection molding of homogeneous materials, but are difficult to extend to inhomogeneous or deformable objects.

The third technique for representing solid objects, used in structural analysis, is the *finite element formulation*[3]. This formulation combines the technique of piecewise definition of the object in a local region (as in boundary representation) with the use of solid primitives (as in CSG). In this formulation, an object is subdivided into regions in its undeformed configuration, or *elements*, through which a smooth interpolation function is used to describe the internal geometry. *Nodes* are reference points in the object at which geometric, deformation, and material properties of the object are stored. The internal surfaces at which two elements come together are defined by identical interpolation functions for the two elements and thus *compatibility* is maintained through the object. The collection of elements (the *mesh*) can be refined to represent the required detail of the internal variation of the object. Since the subdivision into elements is with respect to the undeformed object shape, this is a Lagrangian formulation. Under arbitrary displacement of the nodal points, a point in the object can still be uniquely identified by specifying the element in which it is located and its local coordinates in that element; thus the finite element formulation can be used to define the body coordinate system for patient modeling.

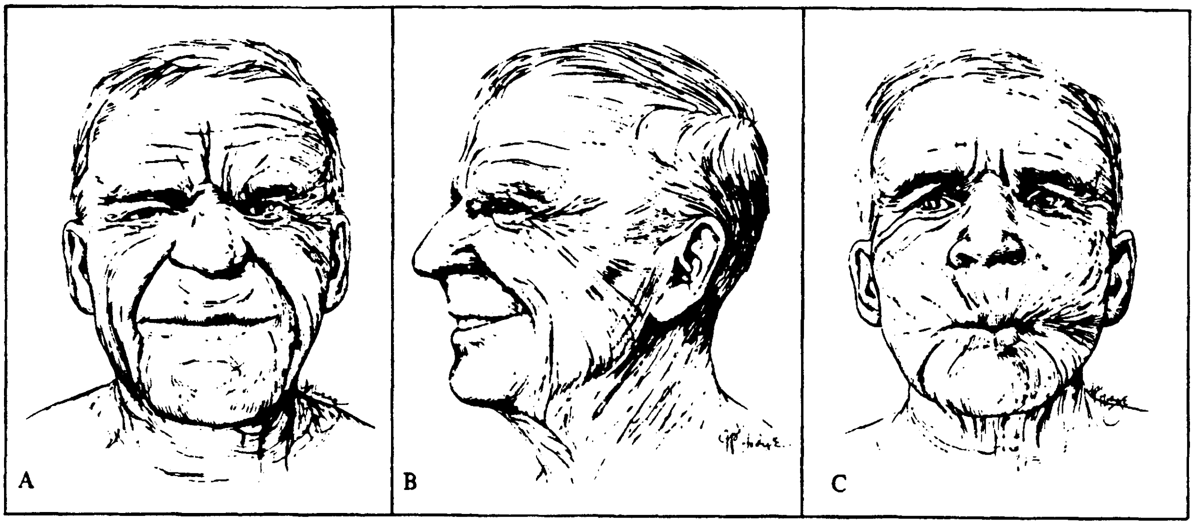
A fourth method of representing shape that is common in medical imaging applications (as opposed to the computer-aided design representations discussed above) is the *volumetric* data representation. Where the other techniques subdivide the object to represent shape, the volumetric technique subdivides world space and describes the material within each division. The volume data can be derived from various scanning technologies including

X-ray, MR, Single Photon Emission Computed Tomography (SPECT), Positron Emission Tomography (PET), and ultrasound. A number of specialized rendering routines have been developed to visualize the data in volume data sets[11,17,41,53,75,76]. Existing volumetric representations are not well suited for simulation of deformable solids because they are based on world space coordinates (i.e., the model is represented using an Eulerian formulation). For the purpose of creating a body state description, volumetric data can be used to guide the construction of a finite element mesh. By redefining the volumetric data in terms of the element local (body) coordinates, the volumetric data can be deformed according to the results of the mechanical simulation.

3.1.2 Mechanical Simulation

Desired Analysis A description of the geometry and material properties of the patient is only part of the total representation. Coupled with that must be a simulation technique for predicting the response of the tissue under the application of forces and displacements. These forces come from external effects, such as gravity, and internal effects, such as material stiffness, inertia, and muscle action.

The response of human soft tissue to these effects can be quite complex because it is organic matter of complex composition leading to nonlinearities in its response. The tissue can exhibit viscoelastic properties such as hysteresis, stress relaxation, creep, and pre-conditioning. In addition, the tissue is inhomogeneous (with different material moduli for different tissue types and at different locations) and anisotropic (with different responses to loading conditions in different directions and orientation dependent lines of pre-stress, see figure 3.1).



Skin lines – the lines of facial expression, contour lines, and lines of dependency.

Figure 3.1: Skin lines on the face[27].

The importance of these nonlinearities is debated in the literature. Kenedi et. al. stress that a complete understanding of tissue mechanics for clinical applications must take into account anisotropy and time dependence[33], while Larrabee points out that tissue behaves in an essentially linear manner in the range of extensions found clinically and that the immediate results of wound closure closely approximate the final result[39]. The work described in this thesis is restricted to linear static isotropic analysis. This follows the procedure proposed by Bathe for mechanical analysis of non-linear materials, namely that the initial analysis should be a simple linear analysis, with non-linearities added only when the linear results indicate that they would be required[3]. The techniques developed here for mesh generation and visualization are directly applicable to more detailed FEM analyses.

Available Techniques Two methods for mechanical simulation of tissue have been discussed in the literature: the finite element method (FEM), and the discrete simulation method (DSM). Both methods approach the problem of complex geometries by dividing the material into regions (elements) that, taken together, approximate the behavior of the complex geometry; so, in fact, both are examples of finite element analyses. The difference between the two methods lies in the way information is passed from element to element as the material deforms. For FEM, the material properties of the elements are combined into a global stiffness matrix that relates loads on the material at the nodal points to the nodal point displacements. This matrix can then be used directly to find the static solution. In the discrete simulation method, the nodal points are iteratively displaced until the load contributions of all adjacent elements are in equilibrium.

The discrete simulation method is commonly used as a simulation method for computer animation[30,78]and has been studied for biomechanical analyses[26,61]. Published uses of this technique have been limited to spring/dashpot elements connected in lattices to approximate solid materials. This method has the advantages that the model is easier to code than FEM, can incorporate nonlinear and time dependent effects, and is easily adapted to parallel processing architectures[54]. Disadvantages of the discrete method are that it is inefficient to implement volumetric elements since the strain in the elements must be re-integrated at each iteration (two-node spring elements are used almost exclusively because the strain across the element is constant and thus trivial to integrate). Also, it is inefficient to mix elements of different sizes in the same simulation because the iteration step size is limited by the highest frequency (i.e., smallest and/or stiffest) element in the model. In addition, the discrete simulation approach requires that the system be repeatedly integrated until it "settles down" beneath some threshold selected by the analyst.

The finite element method has a long history of application in structural analysis of manufacturing and construction materials[3]. It has also been used successfully in a variety of biomechanical applications such as orthopedics, cardiology, and injury studies[23,66]. Researchers have begun to apply the finite element method to the analysis of wound closures and the design of plastic operations[16,32,39]. Since the FEM uses a global stiffness matrix for solving the nodal point equilibrium equations, the matrix can be decomposed once and the structure can be analyzed under various loading conditions. Elements of various sizes and shapes can easily be mixed which makes the model building process less complex.

A number of types of elements are available for use in modeling. The simplest element, the linear spring (or truss) element, is defined using two nodes at the endpoints. The displacement of the spring is linearly interpolated between the nodes, and thus, the strain (derivative of the displacement along the spring) in the element is constant. Higher order elements can be created by using more nodes in each element. A parabolic spring element can be defined, for example, by using three nodes in the spring with the appropriate parabolic interpolation function. This results in a linear strain element. Higher order elements have the advantage that they can better approximate the geometry of curving objects (parabolic or higher order interpolations rather than linear) and higher order strain variations can better approximate the ideal strain variations in the material (for example, beam theory predicts a 3rd order deformation of a tip-loaded cantilever beam which can be achieved exactly using a cubic element). Note that even the use of high order elements leads to discretization error since the real material has an essentially infinite number of degrees of freedom.

Dynamic analysis of structures refers to the prediction of material behavior under rapidly changing loading conditions (i.e., the acceleration-dependent inertia forces and velocity-dependent damping forces are included in the analysis). For DSM, dynamic analysis is easily incorporated into the solution process by making a correspondence between each step of the iterative solution and the time varying loads on the structure. For FEM, dynamic analysis also requires an iterative scheme[3]. In this case, a set of matrices describing the effective nodal mass and nodal damping is added to the stiffness equation. By rearranging these matrices, a new effective stiffness matrix is formed which predicts the displacements at the next time step based on the previous history of deformation. This new effective stiffness matrix remains constant with respect to time, and thus, can be decomposed once and used repeatedly during the analysis. Dynamic analysis using FEM has been extensively studied and mathematical proofs are available to guide the selection of the appropriate timestep to achieve accuracy and stability.

Nonlinear analysis of structures refers to the prediction of the material response when the displacements in the structure are not infinitesimally small or when the material itself is not well approximated with a linear stress-strain relation. Large displacements arise, for example, when the structure undergoes translation or rotation from its original position which is large with respect to the size of the structure. Non-linear material behavior occurs when the material is strained beyond the nearly-linear portion of its stress-strain curve. For the DSM approach, these non-linearities are handled by re-calculating the stress and strain in each element at each time step with respect to the current nodal positions. For FEM, a similar approach is required, with re-integration of the global stiffness matrix to reflect the current configuration of the structure[3]. Since the problem is no longer linear, an iterative solution approach is required to find the displacements under a given loading condition. A modified Newton-Raphson iteration is commonly used in FEM to find the displacement values at which the internal stresses corresponding to the current solution estimate are in equilibrium with the external loads on the structure. The iteration is continued until the difference between these two types of loads (the *out-of-balance* load) is sufficiently close to zero.

The FEM technique can become quite computationally expensive for simulation of complex structures due to the number of degrees-of-freedom in the stiffness matrix. So to accelerate the process, it is useful to look at what precomputation can be performed. One technique is to calculate the free vibration modes of the structure and thereafter transform all loading patterns into their corresponding loads on these modes[10,59]. This *modal* technique is particularly effective for structures undergoing global vibrations, but is inefficient for local variations and rapidly changing loading situations, and is not applicable to materials with non-linear material properties[3]. Another technique involves precomputation of displacement patterns associated with stereotyped loading patterns. These displacement patterns can then be proportionally superimposed according to the relative strength in each of the loading patterns. This technique may be thought of as an extension of the common technique of decomposition of forces and superposition of response as used, for example, to project components of a force onto orthogonal coordinate axes. The extended technique is promising for patient modeling because the anatomy of muscles is predefined, and thus, a deformation pattern can be calculated for each of the muscles. A weighted average of the deformation patterns can then be used to find the total deformation of the face for any given combination of muscle activation levels. This technique has been used by computer graphics researchers for facial animation[57,78], although in these systems the displacement patterns were specified by hand rather than by physical simulation.

3.1.3 Implementation Choice

Body Coordinate System To satisfy the varied needs of the different aspects of the complete system, a mixed representation of the patient model is required. For interaction, a boundary representation of the patient's skin surface is created from the scan data. This can then be drawn quickly by the graphics hardware and is used to map user input events onto the skin surface. A finite element representation is created for the analysis portion of the system. The surface of the finite element mesh is converted back into a boundary representation for visualization of the analysis results. To implement these features, the CAPS system maintains the highest possible resolution description and discretizes as needed for each stage. The high resolution description for the patient is the original scan data. The high resolution description of the operation is the boundary of the incision path and the description of the suture locations.

In addition to its role in mechanical simulation, the finite element formulation has important advantages for visualization for the following reasons. One advantage is that the faces of the finite elements may be discretized into polygons at arbitrary resolution. Another advantage is that by sampling the original scan data at the vertices of these polygons, a displacement vector can be calculated which specifies the distance from the element face to the scan data. By saving this displacement vector in a data structure associated with the vertex, the post-operative skin surface can be displayed by combining the finite element displacement function with the displacement to the scan data at the original position. This results in a high-resolution image of the patient moving under the influence of the mechanical simulation. The same technique could be applied to the volume inside the finite element in order to displace values from a volumetric scan (MR or CT, for example). The important quality of the finite element formulation in visualization applications is that it maintains a coordinate system that remains valid as the body deforms.

Mechanical Simulation The mechanical simulation model in the CAPS system is based on the linear static finite element method. The main reasons for selecting this representation are that it will work well with the body coordinate system discussed above, and that it most easily supports a range of volumetric elements. In my previous work on physical modeling of tissue using the DSM approach, I found that single spring elements can lead to problems in representing continuum effects such as volume preserving deformations[61]. Volumetric elements also allow calculations such as would be used, for example, to estimate the volume of tissue removed in an excision. I also prefer the finite element method because it more efficiently implements high order elements. The CAPS system uses a variable number of node parabolic elements with a maximum size of 20 nodes. This 20 node element forms a 6 sided shape (a *cuboid*, or cube with curved edges), but 4 sided shapes (tetrahedra with curved edges) and 5 sided shapes (curved wedges formed by extruding each vertex of a curved triangle along a curved path) can be generated by reducing the number of nodes (see figure 3.2). These elements are used abundantly in structural engineering practice because they can represent curved geometries and because of their robust numerical properties[3]. In general, the 6 sided shape is preferred for modeling because it represents a more uniform sampling of the material.

As discussed above, there is reason to believe that a non-linear finite element formulation could provide more accurate predictions of the behavior of the tissue. For example, Deng used this formulation in her soft tissue model. Thus, an additional advantage of adopting the finite element formulation is that the patient models generated by the CAPS system could be written out in a format for use in a general purpose FEM program such

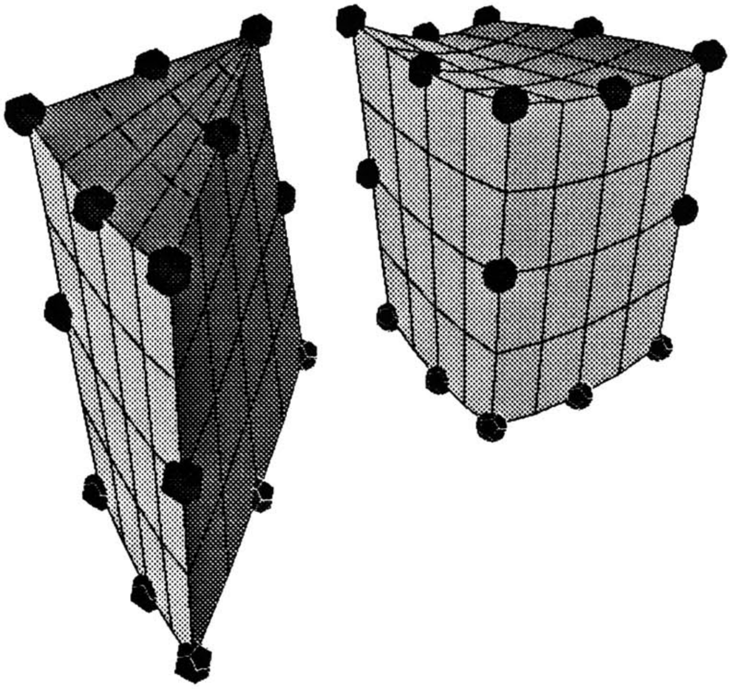


Figure 3.2: The two types of elements used in the CAPS FEM simulation. Spheres are located at the nodal points. Each element face is subdivided into 16 quadrilaterals.

as ADINA[1], a batch-oriented program which supports nonlinear analysis on high order elements. While the interface to ADINA did not allow it to be used as a module of the interactive system described here, it could be used to verify the results obtained from the interactive system. In addition, the ADINA system could be used to explore the impact of nonlinearities to see if they need to be incorporated into the interactive system. ADINA runs on a variety of computer platforms such as the Cray line of supercomputers, and thus, would make more complex analyses feasible.

Although it would have been possible to find a standard finite element package which could have been adapted for use in the prototype system, I decided it would be more appropriate from a research point of view to have all the variables and algorithms from the analysis available for the other components of the system. This has proven valuable for the visualization module, which uses the interpolation functions to subdivide element faces, animate displacements, and calculate the stress and strain values at arbitrary points in the structure.

3.2 Input to the System

The two important classes of input to the CAPS system are 1) the description of the patient model, and 2) the description of the operation to be performed.

[1] The ADINA system, a well known and widely used finite element analysis program, was developed by Bathe et al. and is available at minimal cost to educational institutions[3]. ADINA stands for Automatic Dynamic Incremental Non-linear Analysis.

3.2.1 Defining the Model

This class of input is concerned with the generation of the patient model for simulation. The modeling problem is one of the most difficult aspects of computer simulation, and in surgical simulation, it is particularly difficult due to the complexity of the model to be created. One valuable source of input to the model building process comes from some form of 3D medical imaging scanning technology such as magnetic resonance (MR) or X-ray computed tomography (CT) scanning. The scan data provides internal and external surface information and partial information about the material types, but does not directly detect connectivity of tissue regions or local material properties. Thus, additional processing is required to extract the body state description from scan data. Scan data must be extensively augmented to build the entire patient model.

Collaboration with the MR Lab at the Brigham and Women's Hospital has provided both CT and MR data for use in building the patient models. Their data is processed into a format consisting of lists of 3D points and corresponding point normals using the dividing cubes algorithm[11]. The lists correspond to identifiable tissue types, i.e., there is a file of skin points and a file of bone points. No connectivity data is included in these files. The files represent a significant data reduction over the raw scan data (which may contain many megabytes of data), but a typical file of skin data still consists of approximately 500,000 3D points. One of the major goals of this work involves extracting a finite element mesh from the scan data which captures the essential geometry of the patient in a resolution appropriate for finite element modeling with current workstation hardware.

Surface models of a patient can also be obtained from Cyberware[2] surface scans of a

[2] Cyberware Laboratory Inc. 2062 Sunset Drive, Pacific Grove, CA 93950

subject's head. The original data from the scan provides 512 samples around the head and 256 samples from top to bottom (a sample range map is shown in figure 3.3). The manufacturer reports that the device has a resolution of 0.7mm and is accurate to 1% over a 350mm x 350mm space. The raw data from the device includes some sampling errors (which are visible as light and dark vertical lines in the hair region of figure 3.3). Two-dimensional image processing algorithms can be applied to limit the effect of these errors. Figure 3.4 shows the range data after interactive smoothing was performed using a standard image painting program.[3]

In addition to the 3D data, a 24 bit color sample is obtained at each sample point. A monochrome version of the color data is shown in figure 3.5. To make the rendering object, this data is converted to a polyhedral model with vertices at the sample points and a regular quadrilateral mesh between the sample points. The color data is used as the vertex colors for the object, so that the quadrilateral mesh has a smooth color interpolation (see figure 5.14). The original data can be filtered or subsampled to obtain facial models at various resolutions.

Anatomical references such as Gray's Anatomy are another promising source of data for defining the patient model particularly for those parameters which cannot be extracted from the scan data[28]. To input data from this source, the user must convert 2D drawings and text descriptions into 3D structures. Generally, the requirements of a patient model editing system are similar to those of a computer-aided design or object modeling system[51]. Specifically for this task, the input tool must allow the user to specify the element representation to be used in the model, the articulations of the model (e.g., joints

[3] Image painting operations described here were performed using Adobe Photoshop on an Apple Macintosh Computer. Adobe Photoshop ©Adobe Systems Incorporated. Apple Macintosh ©Apple Computer, Incorporated.

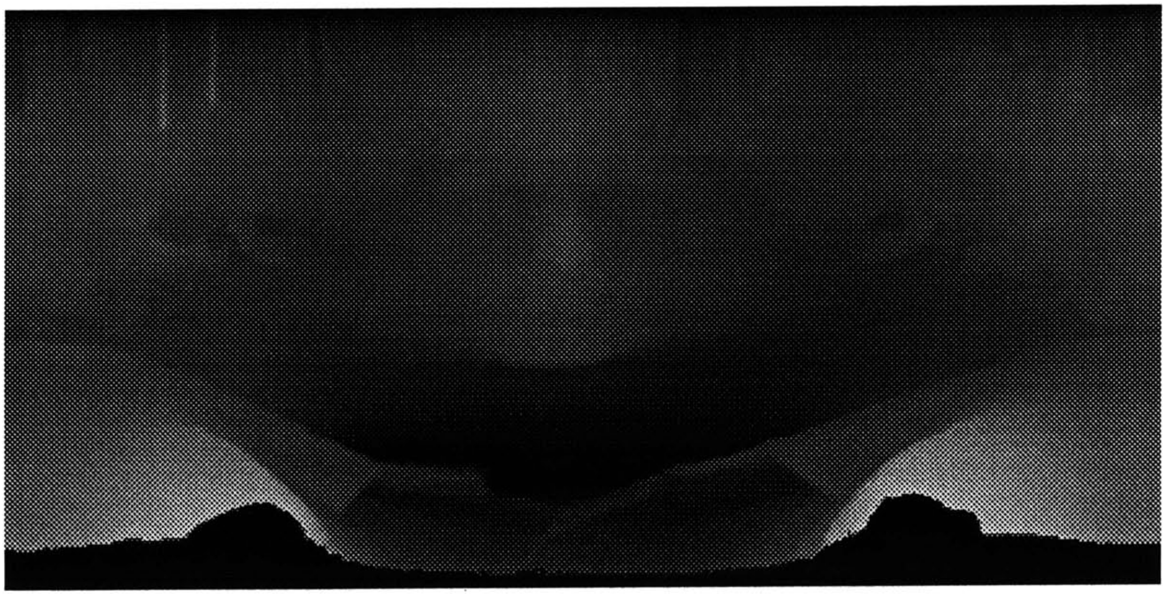


Figure 3.3: Raw range data from the Cyberware scanner.

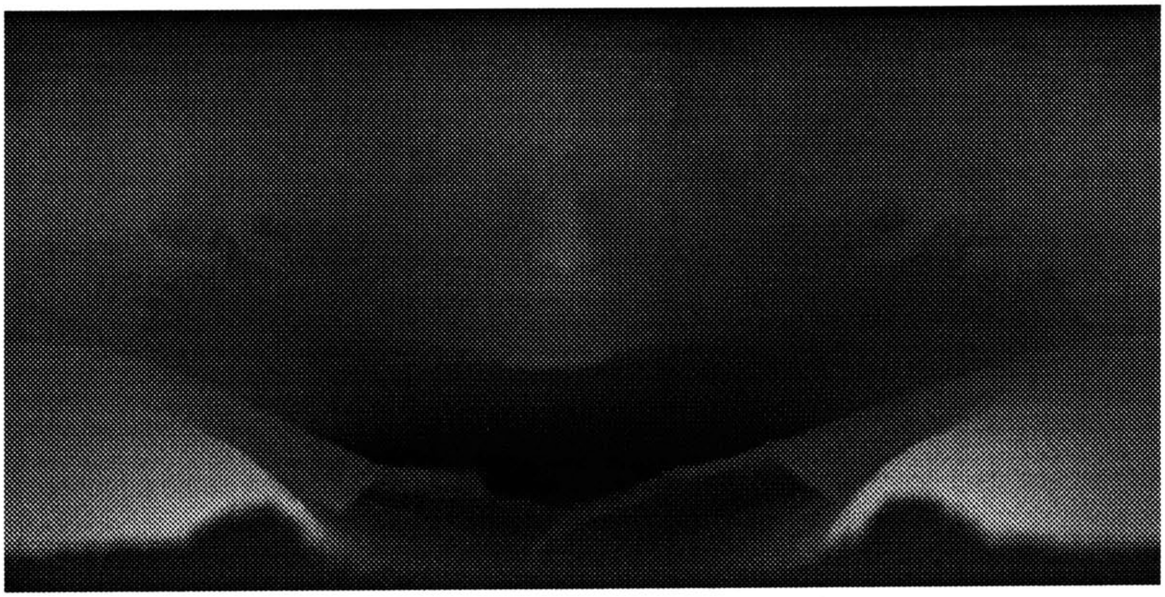


Figure 3.4: Range data from the Cyberware scanner after interactive smoothing.

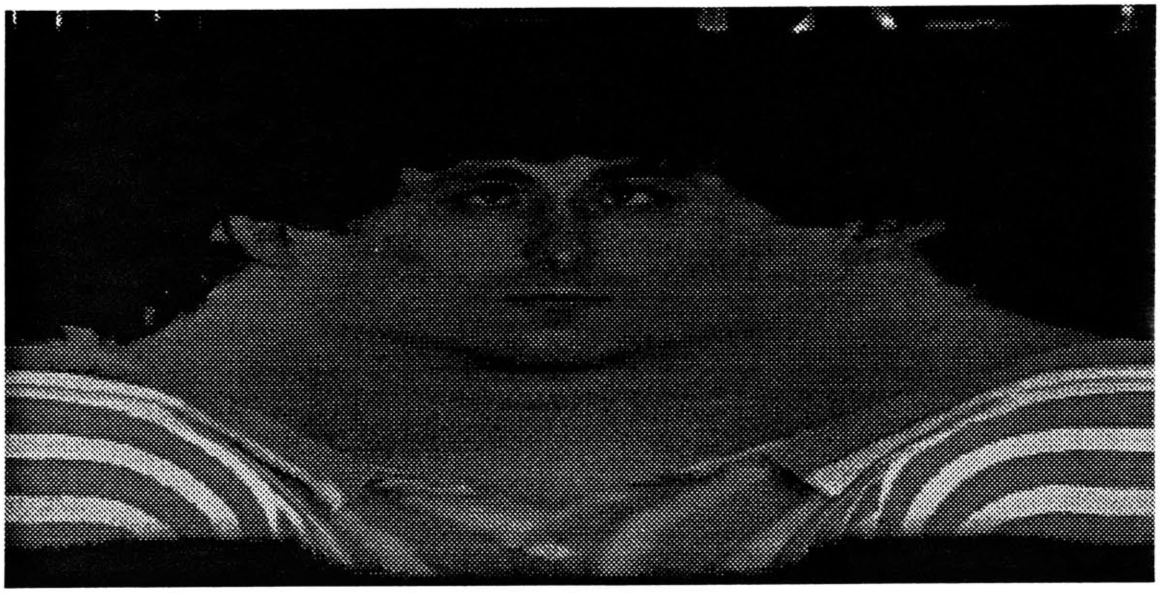


Figure 3.5: Color data from the Cyberware scanner.

and sliding tissue layers), and the origins and insertions for muscles. Specifying these features requires accurate selection of points and regions within the 3D model. While this modeling task seems dauntingly complex, it should be noted that for most situations a generalized patient model may be used and slightly modified to match a particular patient.

Previous Work in Patient Modeling Levesque presented an approach for automatically creating finite element meshes from scan data for skeletal structures based on the work of Chae[7,8,40]. This approach takes as input a set of successive contours (outlines of the inner and outer surfaces of a bone, for example) extracted from CT slices. Levesque's algorithm creates near-equilateral triangles between the scan contours, then passes the triangulated surfaces to Chae's volume tetrahedralization algorithm. Chae's algorithm proceeds by "paring" successive near-equilateral tetrahedra from the object until a single tetrahedron remains. The resulting set of tetrahedra can then be used for finite element analysis. Levesque's technique has the advantage that it is nearly automatic (with the exception of the identification of the initial contours) and that it creates near-equilateral elements that result in a well conditioned finite element mesh. Chae's original work also includes procedures for generating parabolic tetrahedral elements to match curved surface geometries.

Deng used a 3D laser moire digitizer to record the skin surface of a human face to use as the outer surface of a finite element mesh[16]. The mesh was built back from the skin surface at a fixed thickness and connected to a bone support. Deng used the ADINA-IN finite element mesh generator associated with the ADINA finite element package to create meshes from the scan data. ADINA-IN is a script-based program with no interactive graphics interface. Waters used a similar surface-based technique on a model built using photographs of the subject's face to generate a lattice structure for his discrete simulation model[78]. Since both these efforts were based on surface geometry, varying skin thickness could not be automatically generated.

In their computer graphics model of lower extremity biomechanics, Delp et al. were able to find origin and insertion points of the skeletal muscles in published anatomical texts[15]. This data had been collected from cadaver studies and was expressed with respect to the bone segments so it could be typed into a data file read by the simulator. The bones for the model were digitized by drawing a polygon grid on the surface of dried human bones and recording the points using a Polhemus device[63].

In her finite element model of hip replacement surgery, Belec used a set of eleven measurements derived from CT scans to warp a generic femur mesh into a patient specific mesh[4]. She then used a combination of *sectioning* operations (removal of all tissue on one side of a cutting plane) and *drilling* operations (removal of all tissue within a specified cylinder) to define the operation.

Hardt combined a geometric model of the lower extremity with kinematic data acquisition to estimate the forces generated by individual muscles during the walking cycle[29]. Working backwards from observed limb motion, estimated mass and inertia of limb segments, and force plate data of end effector force and torque, Hardt and his coworkers were able to estimate the net torque at each joint. From the geometric model, they were able to estimate the time-varying moment arms for each of the muscles. To determine the combination of individual muscle forces which result in a given net torque, Hardt proposed a linear programming-based approach. This approach requires the selection of a minimization criterion --- Hardt used minimum total force and minimum energy formulations. Patriarcho et al. found that the results of this type of analysis are more dependent on accurate determination of joint angles and joint torques than on the particular optimization scheme employed[58].

3.2.2 Interactive Input Requirements

The final class of input to the simulation is a description of the surgical procedure itself. The surgical procedures that I analyze in the prototype system are plastic operations to the skin such as tumor resection and Z-plasty. Specification of the procedures can be broken down into primitive components of 1) cutting along the skin surface to a specified depth, 2) undermining the skin to create flaps, 3) removing regions of tissue, and 4) specifying wound edges which are to be brought together with sutures.

3.2.3 Interactive Input Options

A number of multiple degree-of-freedom input devices were available as options for implementing the interactive portion of the prototype system to address the input problems discussed above. The devices included a mouse-based graphical user interface (GUI), the VPL DataGlove, the Exos Dexterous HandMaster (DHM), the Spatial Systems Spaceball, and a 3D force input/output joystick. The next paragraphs look more closely at each of the devices and the interaction styles they support.

Mouse-based GUI The Starbase graphics library of the HP workstations can be used for hardware rendering in an X window.[4] X window routines can be used to get mouse events from the window in order to support screen space picking of 3D primitives. In this mode of interaction, the X window cursor is drawn in the hardware overlay planes and does not require rerendering the 3D image as the mouse is moved. The scene only needs to be rerendered when mouse events cause a change in the scene description. Although the mouse is a 2D input device, its 2D space can be mapped to a surface in 3D space through the inverse viewing transformation (see section 4.4.1).

The Motif[5] toolkit of user interface widgets can also be rendered in X windows. The Motif toolkit supports widgets for buttons, pulldown menus, sliders, file selection boxes, and text input. These widgets can be assembled into a control panel which is used to select modes, set parameters, and issue commands to the simulation processes. Since both the 3D scene and the Motif widgets are rendered in X windows, they can be moved, resized,

[4] The X Window System is a portable, network-transparent window system developed by Project Athena at MIT which has been adopted as a standard by a variety of workstation vendors. X Window System is a trademark of the Massachusetts Institute of Technology.

[5] Motif is a product of the Open Software Foundation, Inc., Cambridge, Massachusetts.

raised, and lowered using the same window manager protocol.

VPL DataGlove The DataGlove provides a method of direct interaction with 3D models[71]. It monitors 6 degrees of position and orientation information measured from the user's palm using a Polhemus tracker to control a 3D cursor[63]. Ten additional degrees of freedom are measured from the user's finger flexion and extension. By detecting hand postures from the 10 finger flexion values, the system can map the position and orientation degrees of freedom onto parameters of the simulation. The DataGlove can be sampled at up to 60 Hertz. Because the DataGlove controls a 3D cursor representation of the hand, the entire 3D scene must be redrawn every time the glove moves. To provide visual feedback of cursor position for manipulation of the models, the redraw loop must run at a minimum of about 5 Hertz. For the HP workstation configuration used for the prototype, this limits the size of the polygonal model to approximately 2,000 polygons for direct manipulation with the DataGlove.

Exos Dexterous HandMaster The Exos DHM is an alternative hand position sensing device which offers greater accuracy and more degrees of freedom than the DataGlove[44]. This is achieved through an instrumented exoskeleton strapped to the user's hand. The DHM monitors 20 joint angles at a rate of up to 60 Hertz. Drawbacks of the DHM are that it is awkward to put on and take off, and that it is more fragile than the DataGlove.

Spatial Systems Spaceball The Spaceball is a 6 degree of freedom force/torque sensor with 9 buttons. The Spaceball can be used to control the position and orientation of a cursor in the 3D scene by mapping force to translational velocity and torque to rotational velocity. The Spaceball device can be sampled at up to 60 Hertz. The Spaceball is similar to the DataGlove in that visual feedback is needed in order to effectively control the cursor. Thus, direct manipulation with the Spaceball is also limited to scenes with less than 2,000 polygons on the HP workstation.

3D Force Input/Output Joystick The MIT Media Laboratory, in collaboration with the MIT Mechanical Engineering Department, has built a 3 degree of freedom force input/output joystick[50]. The joystick has a gimbal which allows normal joystick motion, and a sliding shaft which provides a third axis of movement in and out from the center of the gimbal. Each of these degrees of freedom is connected to an optical encoder, a motor, and a brake. There is also a 3 degree of freedom force sensor on the tip of the joystick. These devices are connected to a control loop running on an Apple Macintosh computer. The Macintosh can run a simple joystick control loop at up to 200 Hertz. As the mapping from input joystick position and load onto output motor and brake activation becomes more complex, the speed at which the Macintosh can execute the control loop goes down. The system can go unstable if the user changes the position or loading of the joystick at a higher frequency than the control loop can handle.

In the context of the surgical simulator, there are a number of interaction techniques that could be explored using force input/output. One example is using the joystick as a cursor in the 3D scene with force generated to keep the cursor from passing through the skin surface. This could be used to select the incision path on the skin surface. The same effect can be achieved, however, by using the mouse to select points on a 2D image of the skin and mapping the picked points back onto the 3D model. Another application that takes direct advantage of the fact that the joystick operates in the force domain would be to feel the force/displacement relationship at any point within the patient model. An approximation to this relationship is directly available from the finite element stiffness matrix for each of the nodal points in the model. A 3x3 stiffness matrix can be formed from the global stiffness matrix elements which lie on rows and columns corresponding to the nodal point degrees of freedom. This 3x3 matrix could be downloaded into the Macintosh and incorporated into the joystick control loop.

Since the primary feedback of the joystick involves forces generated by the control loop, the complexity of the 3D graphics scene is irrelevant. What is critical is the amount of complexity which can be incorporated into the Macintosh control loop and how often that information needs to be modified. In previous research with the joystick, we found that even a few force generating primitives in the joystick loop (such as springs or collision surfaces) can significantly degrade performance[62]. The computational complexity of algorithms needed to support the joystick as a useful interface device in interactive simulations remains an open question.

3.2.4 Interactive Input Choices

Two aspects of the planning problem for plastic surgery lead to the choice of the mouse-based GUI as the appropriate interface for the prototype system. The first observation, as described in more detail below (see section 4.2), is that although it is important that the patient model and the simulation be three-dimensional, the operation itself takes place on the skin surface which is well approximated by a 2D space embedded in a 3D space. Because this simplification can be made, it is possible to implement 3D picking operations without resorting to the inherently 3D input devices such as the DataGlove or Spaceball.

The second observation about the plastic surgery planning problem is that the planning tool should provide a familiar environment for the surgeon. Since most surgeons are not expert computer users, it is important that a computer-based tool exploit other computer knowledge they have developed. The mouse-based GUI has come into such wide-spread use over the last decade that there is ample opportunity for the surgeon to gain experience with similar user interfaces. Currently, the other interface device options are much less widely available and it is unclear when they will come into widespread use, if at all.

The interactive input requirements are implemented using the mouse-based GUI by combining the following user actions. Referring back to the primitive components required for specification of the surgical procedure given in section 3.2.2, the following interface was selected. The user is presented with a window system screen containing a graphics window showing a rendered image of the patient model, and various GUI control panels. 1) Cutting on the skin surface can be implemented by detecting the location of mouse button presses in the graphics window and finding the corresponding point on the surface of the patient model. Sequences of these points define an incision path. The depth of the incision is specified by typing into a text-entry area on the GUI control panel or in a configuration file for the program. 2) Undermining skin to create flaps can be handled by the automatic mesh generator by assuming that every FEM node associated with a sutured edge is to be undermined. If this assumption does not yield the desired condition, the 3D object picking operation can be used to select the nodes to be undermined and a menu item can be selected which executes the appropriate command on the selected nodes. 3) Removing regions of tissue can be done implicitly by reshaping the wound boundary. This requires selecting the appropriate vertex along the wound description and a point on the skin surface where this vertex is to be moved. The automatic meshing algorithm described below is then able to create a mesh in which the tissue within the excision is removed. 4) Specifying edges to be sutured can be done by selecting pairs of wound edges which are to be brought together. Selection could be implemented using the 3D picking operation. However, the CAPS system takes advantage of fact that there are standard suture patterns for each common surgery. This allows specification of the sutures through a single control panel menu selection rather than through a sequence of edge selections.

Although I think the force input and output would provide for more natural interaction, in view of the uncertainty about its value and its status as a prototype device, I did not make the joystick an integral part of my thesis research.

3.2.5 Requirements for Combining the Plan and the Patient Model

To complete the input section, we must have a method to convert patient scans and the interactively specified plan into a well-posed problem for finite element analysis. This method must fill in the gap between the input data -- the high-level description of the plan (in terms of the modeling primitives discussed in section 3.2.2) and the raw patient scan -- and the output data -- the finite element mesh which captures the geometry of the patient and models the incision. The method should also be parameterized, so that the user can control the number of elements in the mesh and their distribution through the analysis area.

3.2.6 Options for Combining the Plan and the Patient Model

The options for this procedure are best thought of as the question of when to discretize the input data into the FEM mesh: 1) perform editing operations on the fully 3D mesh, 2) edit a 2D mesh, 3) define the operation before creating the mesh. The first option is to create a fully 3D mesh of the patient and then perform editing operations on that volume. These editing operations could be implemented as a set of CSG operators for subdividing the mesh and subtracting tissue. This approach was used by Belec in her femur simulator, although her implementation was limited to linear elements[4]. The implementation becomes significantly more difficult when applied to higher order elements because of the problem of calculating the intersections of non-linear volumes. As described earlier, high order elements are preferable for soft tissue modeling because they can more closely approximate curved geometries and they give non-linear displacement patterns.

An alternative to the fully 3D non-linear CSG method is to perform the CSG operation on a simplified version of the model and then convert it back to a fully 3D mesh. For example, a mesh of parabolic finite elements could be approximated locally by linear elements on which Belec's algorithm could be used. These linearized elements could then be converted back to parabolic elements for the analysis stage. Another approach would be to perform the CSG operations only on a surface mesh and then convert it to a non-linear 3D mesh after editing. As part of the development of the CAPS system, I implemented a version of the surface mesh CSG technique in which a mouse-based GUI could be used to define cutting planes and to delete sections of the polyhedron.

While this approach has intuitive appeal, because the CSG primitive operations are nicely analogous to cutting and removing tissue, this technique results in fragmented elements and unsymmetric edges. This fragmentation makes it difficult to apply suturing constraints to the incision. This problem led me to explore the option of creating the finite element meshes by hand using mouse clicks to define nodes and group collections of nodes into elements. This technique is useful for debugging the finite element code, but proved overly tedious for defining large analyses.

As mentioned earlier, the problem can be viewed as a question of when to discretize the scan data and incision parameters. I concluded that the drawbacks of the CSG approach are due to the fact that the scan data is discretized (either to a 3D mesh or to a polyhedron) before the incision is planned. This prediscretization is the cause of the fragmented elements in the area of the incision because the discretization grid for the original mesh is not sensitive to the geometry of the incision. It became clear that a better mesh can be generated if all the input data is assembled before discretizing. The automatic surface meshing scheme used in the prototype system is a modified version of the 2D mesh generation algorithm described by Chae for creating meshes of objects with holes, such as plates which have been drilled[8]. Since, in the analysis of incisions and closures, the the behavior of the tissue immediately surrounding the incision is important, my mesh generator uses the local shape information from the incision to define the geometry of the surrounding elements (this results in more regularly shaped elements than are obtained from the CSG approach). The concept of growing out from local geometry for mesh generation is similar to the approach used in mesh generation for computational fluid dynamics (CFD), where an object such as an airplane is embedded within a simulation grid[52].

3.2.7 Choices for Combining the Plan and the Patient Model

The automatic mesh generation technique is most appropriate for the task of combining the plan and the patient model data into a finite element mesh because:

- It maintains the highest resolution description of the patient.
- It does not commit to a mesh discretization until the surgery is specified, i.e., it uses information about the operation to select a mesh.
- It can generate arbitrary resolution meshes for more accurate analyses.

The technique I have adopted consists of two steps: surface mesh generation followed by continuum mesh extrusion. These techniques are briefly discussed here and are presented in more detail in section 4.5.

The surface mesh generator works from a description of the incision geometry entered using the interactive graphics editor. This incision, together with the Cyberware range data or processed CT data, is used to generate a mesh of the appropriate resolution that follows the contours of the patient's skin surface. The algorithm is an extension of the work described by Chae for automatic mesh generation[7,8]. Chae's algorithm was not directly applicable to this problem because it generates a mesh of triangular elements; as discussed above, is is best to use an algorithm which generates quadrilateral elements wherever possible. Chae was interested in mesh generation for closed regions containing holes, and his algorithm works by adding elements at the boundaries which eventually fill the region. Thus, his algorithm needs to ensure that meshes grown from different boundary edges meet without passing through one another or creating undesired artifacts.

A simplifying assumption for plastic surgery application allows the surface mesh generator to avoid many of the complex conditions imposed by Chae. For this work, the incision boundary was considered the only boundary of interest, so the elements are always being added to the outer boundary of the current mesh. The procedure used in the CAPS system works by first adding triangles to the surface mesh to create a nearly convex border, then by adding quadrilaterals until the desired size is reached.

The continuum mesh extrusion process works by turning triangles in the surface mesh into wedge elements and quadrilaterals into cuboid elements. For each vertex in the surface mesh, a finite element node is created in the same location. Additional nodes are created inward from the skin surface by transforming the point into cylindrical space, reducing the r value by a chosen thickness, and transforming back into world space. The nodes are connected by elements if their corresponding vertices were connected by polygons in the surface mesh. Parabolic elements are created by adding nodes at the midpoint of each edge in the surface mesh and extruding them using the same technique. To better fit the patient, middle nodes which lie on the surface mesh are offset to the range data. By taking advantage of the regular surface characteristics of the head and neck, I can avoid the general triangulation technique proposed by Levesque.

One technique I have explored as an alternative to the Cyberware scan uses the data files generated by the Brigham and Women's Hospital segmentation algorithms. This is done by converting the scan data to the same coordinate system given by the Cyberware scanner by sorting the data points into buckets corresponding to ranges along the z-axis passing from the neck to the top of the head and ranges in angle θ measured around the head. Within each of these buckets, a vertex for the polyhedron is defined using the center z and θ values and the maximum r value of the points in that bucket, where r is the distance from the z axis. The resulting r values are then written out in the Cyberware range format for use with the rest of the system. The parameters of this conversion are the sizes of the buckets in z and θ, and the range of data to be converted (expressed as minima and maxima in z and θ).

The data files from a CT scan include data from both the soft tissue and the bone. The conversion procedure described above gives a map of the outer surfaces of these tissue types. The distance between these surfaces is the thickness of the soft tissue. This information could be used with the automatic mesh generation procedure to create continuum meshes that better approximate the three-dimensional anatomy.

It may be necessary to use a CT scan from one patient to develop a generic patient model and then apply this soft tissue thickness to a Cyberware scan of another patient. This may be required since CT scans impose some risk to the patient from radiation exposure and are not routinely done for plastic surgery on the skin surface. To align CT and Cyberware data from different subjects, landmarks on the skin surface and an alignment procedure would need to be defined. The non-linear interpolation functions used in FEM are likely to be useful in creating such an alignment procedure. If the scans are from the same patient, the calibration procedure described in section 4.3 may be sufficient.

3.3 Display of Results

There are two basic applications of the output from the CAPS system. The first is the interactive output that helps the user of the system control the input devices to define the model and to define the operation. The second application is the creation of a record of the planned procedure for use in discussing the procedure with the rest of the surgical team, for guiding the execution of the procedure in the operating room, and for documenting the procedure. The visualization requirements are basically the same for both applications --- the patient geometry and the planned procedure should be unambiguously rendered and the analysis results clearly displayed --- but the preferred level of detail and output media for the two applications will be different. In this section, I discuss the data in the CAPS system which must be rendered, the techniques through which this data can be converted to images, and the preferred implementations for the interactive and documentation applications.

3.3.1 Data to be Viewed

The general goal of the output section of the simulator is to map all the relevant parameters of the system onto graphic primitives for display in the context of the patient geometry and in a manner consistent with existing illustration conventions. This section describes mappings for different types of data.

Patient Model The geometry of the patient is perhaps the easiest of the mapping tasks since the model is already subdivided into elements for the mechanical simulation. The typical mapping used to visualize finite element assemblages is to draw the edges of the elements as lines and the faces of elements as polygons. This works well for simple linear elements (triangles and tetrahedra) but must be extended for use with higher order elements. For the parabolic elements I use in my analysis, each face is a bi-quadratic surface which can be polygonalized by imposing a grid in parameter space which is mapped to world space using the current nodal displacements. Important quantities to be mapped onto the subdivided element faces are:

- displacements to the original range data,
- vertex colors from the source color scan,
- interpolation of displacement conditions to make animation, and
- other analysis results blended with the source color.

For each of the three degrees of freedom at each node of the finite element mesh there can be either displacement or force boundary conditions. Displacement boundary conditions fix the position of that degree of freedom in world space. In engineering drawings, displacement boundary conditions are typically drawn as small triangles at the nodes if all three degrees of freedom are constrained, or as circles at the nodes if only one or two degrees of freedom are fixed. Loading boundary conditions are typically drawn as lines from the node in the direction of the force and with a length or thickness proportional to the magnitude to the force.

Surgical Plan Medical illustrations of surgeries are typically schematic in nature with the following conventions. Cutting paths are shown as solid or dashed lines on the skin surface with cross sectional views through the skin when incision depth or undermining needs to be shown. Post-operative diagrams show solid lines along wound closure path with cross lines showing suture placement.

Analysis Results The analysis results include the stresses and strains in the material as well as the deformed tissue geometry. The deformed geometry can easily be handled by simply regenerating the polygonal mesh of the skin surface using the new nodal point positions. Visualization of the stresses and strains presents a more challenging graphics problem since each is a 6-dimensional vector defined over the volume. A common technique for rendering these results on the structure is to map the magnitude of the vector to a color, for example, making regions of high stress appear redder. Another technique is to generate contour bands showing where the magnitude of the vector passes a threshold.

Analysis results are also commonly shown in standard x-y graphs, for example, function plots, bar charts, or scatter plots. The y-axis of the graph could show any of the analysis results (stress, strain, or displacement) with respect to any of a number of simulation parameters. Desirable x-axis variables include distance along a path on the skin surface, variation in a parameter of the operation (such as Z-plasty angle), or time (in a time dependent analysis).

3.3.2 Available Display Techniques

From the discussion above, we have the following set of graphical objects to display: 3D lines, 3D polygons, 3D polygon meshes with colors defined at the vertices, and 2D plots.

3D Color Rendering The 3D primitives can be rendered using standard computer graphics algorithms[21]. The basic technique requires specifying the parameters of a virtual camera with which to view the data. Parameters of the camera include the position of the camera in space, the direction of view, the orientation of the camera, and (for perspective projections) the field of view. These parameters define a transformation which maps the 3D primitives into 2D primitives for display on a 2D device (such as a CRT screen). Scan conversion of the primitives is the calculation of which display picture elements (pixels) are covered by a given primitive. Hidden surface removal is the process of determining which of the primitives is closest to the camera at each pixel. The Z buffer algorithm, the most common form of hidden surface removal, maintains a separate depth value for each pixel of the display. When a primitive is being scan converted, its depth at each pixel is compared with the value currently stored in that pixel. If it is closer, the pixel color and depth are set to the values from this primitive. If not, the existing pixel values are left unchanged.

The colors of the primitives are determined by both their own color (either the natural color, as for the skin surface, or by the mapping from analysis results) and by the virtual light sources in the scene. Different shading models are obtained depending on how often the lighting for the primitive is calculated. Faceted shading is obtained when lighting is calculated once for each polygon and the same color is used for each pixel in the polygon. Gouraud shading refers to the effect achieved when the shading calculation is performed once for each vertex of a polygon and the pixel colors are interpolated from vertex values. Phong shading refers to the effect achieved when the lighting calculation is performed once for each pixel in the polygon.

Special Purpose Rendering Hardware Computer graphics rendering techniques are inherently computationally expensive since the final rendered scene on a current high-resolution display has over one million pixels. However, the repetitive nature of the algorithms for graphics leads to efficient hardware implementation. Graphics accelerators are available which perform the common rendering operations using specialized architectures. The CAPS system was implemented on a Hewlett-Packard 9000 series 835 workstation with a TurboSRX graphics accelerator. This machine can render approximately 10,000 Gouraud-shaded polygons per second using a hardware Z-buffer. Already, machines of much higher performance are available from a variety of manufacturers with performances up to 1 million polygons per second. In the future, we can expect even greater graphic performance through improved computer chip technology and parallel processing[21].

Stereo Displays The 3D primitives can be rendered twice with the virtual camera offset by the distance between the viewer's eyes to generate images which can be fused by the viewer into a true 3D image. The 3D image has inherent depth cues which are important for appreciation of complex shapes and detailed manipulation. Since the images must be presented individually to each of the viewer's eyes, a special display device is required. Techniques for delivering the images from a standard CRT involve alternating the images on the screen and shuttering the viewer's eyes so that each eye sees only its image (see, for example [31]).

The importance of stereo displays is currently unclear. Research by McWhorter et al. indicates that stereo graphics is preferred by viewers of solid models in CAD applications and Takemura et al. found that stereo displays improved user performance in a 3D pointing task[47,72]. However, Sollenberger and Milgram found that for viewing the branching blood vessel patterns stereo display was actually a weaker cue than motion parallax from a rotating 3D model[68]. Similarly, in an evaluation of a three-axis tracking task, Kim et al.

found that, although stereo cues were generally superior, the use of depth cues such as background grids and vertical reference lines along with optimal perspective parameters in monoscopic displays led to essentially equivalent user performance[34]. Merritt et al. found that stereo cues are particularly important in remote vehicle driving applications in wooded terrain[48]. They attribute this to the fact that the user must distinguish the size and location of irregularly shaped objects such as tree limbs and other road hazards.

Another technique for displaying stereo images involves the construction of special head coupled hardware with one display for each eye. This type of display, called a head mounted display (HMD), is used in combination with a head tracker that allows the virtual camera parameters to be updated to reflect the viewer's head movements. When the rendering is fast enough, the combination of stereo display and motion parallax can give the viewer an enhanced appreciation of the 3D structure of the scene[20,46].

New stereo display technologies currently under development will project *autostereoscopic* views of 3D scenes, i.e., they will not require special hardware worn by the viewer. This type of display technology, which is based on the computer calculation of the holographic interference patterns in a synthetic scene, will allow group viewing of a 3D environment[69].

Hardcopy Color renderings can be saved as bitmaps (storing the color value at each pixel in a disk file) or they can be transferred to various other media for storage or publication. A common hardcopy medium for images is standard color photography. Photographs are advantageous because they retain the spatial and color resolution of the original image. Laser printer output is another mode of hardcopy production which is commonly available and useful for image output. Current laser printers have high spatial resolution (300 or 400 dots per inch) but limited color resolution (each dot is either black or white). The technique of halftoning can be used to convert a low spatial resolution/high color resolution image (such as a color rendering) into a high spatial resolution/low color resolution monochrome image for output on a laser printer. New hardcopy devices include color image printers that accept video or bitmaps directly and, further in the future, holographic stereogram printers which will produce white light viewable stereo images[5].

Animation A sequence of images can be generated each with a different value of one or more parameters to generate an animated sequence. The parameters may be aspects of the virtual camera, the position or orientation of the objects in the 3D scene, or parameters of the simulation which generates the scene. When the generation of the images is performed in real time in response to user input, the technique is called interactive graphics. When the images are generated off line, the sequence is single frame animation. Interactive graphics can be used to provide feedback to the user to aid in the selection of parameter values being controlled by an input device. Single frame animations may be stored either as bitmaps in disk files for replay on the computer or recorded on video tape.

3.3.3 Implementation Choices for Visualization

Interaction To effectively implement the interactive geometric modeling operations described above, I have implemented a system which generates interactive graphics animations of the output primitives in response to user input. The system takes advantage of the TurboSRX hardware color rendering device connected to the simulation computer. A major challenge in designing such a system is dealing with the tradeoff between image complexity and fast screen update. Higher resolution, more complex scenes can provide a better appreciation of the model, while a less complex scene which renders more quickly can have the advantage that user input can be immediately reflected in the scene. The user can select the level of subdivison to be performed on the element faces in the patient, and thus, the number of polygons in the 3D scene. In this way, a low resolution image can be used interactively, and a higher resolution image can be generated for documentation purposes. Due to a lack of appropriate hardware, I have been unable to fully evaluate the applicability of stereo displays to this problem, although stereo pair pictures can be displayed on the CRT screen for viewing with the "cross-eyed" technique.

Documentation The output medium most consistent with current planning techniques is a photographic print, slide, or the output of a color or black and white printer showing the shaded image of the patient and a schematic representation of the operation. This type of output can be added to the patient's file in order to document the procedure. This type of output is also appropriate for discussing the procedure with the other members of the surgical team, can be supplemented with animations showing the results of the analysis, and would be valuable for justifying the selected surgical approach.

The CAPS system uses a window system-based interface which allows the creation of "snapshot" windows. These windows contain static images of different stages of the operation or multiple views of the model. This capability allows the user to create a composite sequence of images which serve as a storyboard for the operation.

Chapter 4

Implementation

This chapter goes over the technical details of the implementation of the CAPS system. After a brief overview of the main components of the system, each of the required techniques is examined in roughly the order they are used when performing an analysis.

4.1 Conceptual Overview

The process used in the CAPS system includes an interactive planning stage, an analysis stage, and a visualization stage. These steps are performed in sequence for each surgical plan. The system allows the user to modify the existing plan and re-run the analysis and visualization steps. This process is diagrammed in figure 4.1.

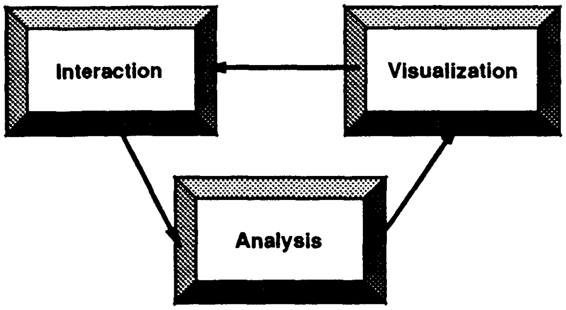


Figure 4.1: User interaction with the CAPS system.

The process begins with a Cyberware scan of the patient. This scan data contains both range and color data of the skin surface in a format discussed below. This data is first processed into a pre-operative three-dimensional model of the patient which is used during the interactive planning stage. Figure 5.1 shows the patient model in the interactive system.

The model is loaded into the interactive planning system where the user is able to interactively control the computer graphics view camera and rendering options using the mouse-based GUI. The plan is then entered into the system using the mouse to select locations on the skin surface which define the incision boundaries. As the incision is defined, a set of slice planes is superimposed on the patient model showing the location of the cuts. The user may then go back and modify the locations of the incision points and view the resulting cut planes.

When the user is satisfied with the locations of the cuts, an analysis of the plan is performed by selecting the appropriate menu item corresponding to the type of procedure. Currently, the prototype system supports two types of surgeries --- a single Z-plasty and a multiple Z-plasty combined with removal of tissue. Adding new types of surgeries is done by creating a text file which describes the suture connections needed for that procedure. Once those suture conditions are specified, the new procedure can be used with all the other tools described here.

The analysis stage of the system can be run with a variety of parameters which allow the user to control the resolution of the model (and thus the time required for analysis). These parameters control the mesh generation process, telling it how many finite elements to generate to represent the region of the face surrounding the operation, and how large a region of the face is to be included in the analysis. With these parameters specified, the rest of the analysis is automatic. The mesh is first converted from a surface description into a full three-dimensional mesh so that volumetric effects (such as the dog's ears phenomenon discussed in section 2.1) will be represented. The volumetric mesh also contains boundary condition information specified in the suture file. The finite element mesh is then integrated to form a global stiffness matrix for the structure. This stiffness matrix is then decomposed into a form which can be solved quickly under a range of loading conditions. Solving the matrix for each loading condition results in a displacement function describing the movement of each location in the mesh.

For the visualization stage, the mesh is discretized into polygons. This process involves calculating a set of sample points on each face of the mesh and saving both their world space position and their body coordinates (see section 4.2.4). These points serve as the vertices of the polyhedron showing the skin surface and the analysis results. The original scan data is also sampled at these vertices to color the polyhedron and to displace the sampled vertices to match the range data. By selecting a large number of sample points, the polyhedron can make use of all the data in the original scan. By specifying a smaller number of sample points, the resulting polyhedron can be drawn more quickly and is therefore easier to use in the interactive system.

This displacement function from the analysis stage can then be applied to each point in the displayed polyhedron in order to generate a post-operative view of the patient. The user may then choose to view the polyhedron in the pre-operative state, the post-operative state, or at positions in between. By selecting a menu item, the user can view an animated sequence which interpolates between the pre- and post-operative states. This animation may viewed on the screen in the interactive system or it can be recorded onto videotape at high resolution for real-time playback.

The original pre-operative patient model and the incision plan description remain loaded in the system during the visualization stage. If, based on the analysis results, the user decides to modify the plan, the analysis can be re-run as many times as needed until the desired result is achieved.

4.1.1 Modular Decomposition of the System

The CAPS system for this thesis is implemented as a set of processes in the bolio graphical simulation system[81]. This system has been in use and under development for the past five years at the Computer Graphics and Animation Group at the Media Laboratory, and has a number of features useful for this work. These features include an object-based interface to various graphics rendering devices, a constraint-based message passing system for associating events with changes in the object data[70], and a multi-process communication protocol for distributing computation among machines on a local network. In the context of the bolio system, each of the major modules of the CAPS system is implemented as a single process which communicates with the other modules by passing messages through bolio. This implementation is transparent to the user, to whom the system is an integrated graphical application.

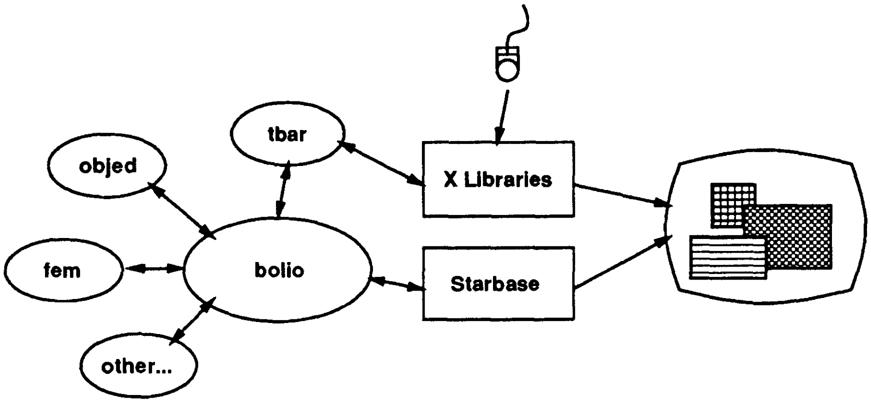


Figure 4.2: Modules.

Tools There are four main processes in the final program: *bolio*, *tbar*, *fem*, and *objed*. Bolio is responsible for rendering with the Hewlett-Packard Starbase graphics library, communication over Unix pipes and sockets, and management of graphical objects such as markers and cursors. Tbar implements a mouse-based GUI and a user-level interpreted programming interface to the other modules (see the discussion of tbar below). Fem is a program for performing finite element analysof structures and for visualizing the structure and the analysis results. Objed (the name comes from *Obj*ect *Ed*itor) contains the code for intersection and picking, for maintaining incision and suturing data structures, and for mesh generation. Other smaller processes are used during creation of specific test cases, for example, as motor control programs to animate muscles, or animation programs to interpolate solutions. These smaller programs send messages to bolio or through bolio to other simulation processes.

The separate process approach to modular decomposition is in one sense merely an implementation detail of the program --- the same basic functionality could be written as a single process. However, the main advantages of the approach are that it allows both distributed processing and run-time re-linking. These features have both been used extensively in the development of the prototype, especially with respect to the fem module. Distributed processing refers to the fact that the fem module may be run on a different machine than the user interface and rendering processes. This is an advantage since the available memory is the constraining variable on the level of detail included in the finite element analysis. The fact that fem is implemented as a separate process allows the user to dedicate the entire capacity of another machine on the network to the analysis task while using the interactive graphics interface locally. The fact that the computation is distributed in multiple processes also means that certain calculations can be overlapped; e.g., the fem module can be building the stiffness matrix while the interactive viewing system is used to examine the finite element mesh. The run-time re-linking feature was particularly important when writing and debugging the modules since the buggy module could be removed from the simulation, recompiled, and then restarted without restarting the entire system and losing the interactively defined data.

Messages A sequence of commands and arguments to a process is referred to as a *message.* Each process has a command set that it recognizes. The command set of a process is the set of keywords (commands) that are passed to it through its connection to bolio. Bolio sets up connections to the processes so that the process receives commands on its standard input and so that commands it sends to bolio are written to its standard output. This is useful during debugging because when the process is on a terminal (or terminal emulator in a window system), the standard input is the keyboard and the standard output is the display screen. Any Unix program which uses these standard input/output methods can in principle be used as a bolio process.

The commands do one or more of the following: modify internal data structures, send messages to other tools, print messages to the user. Thus, a single message may start a long sequence of events. E.g., a menu selection in tbar can cause fem to read in a script file which installs a new set of graphical objects in bolio or starts a new simulation process. Commands and arguments for the processes used in the CAPS system are parsed using either a parsing library I wrote (bolio, objed) or *tcl* language (tbar, see below) or a combination (fem).

User Interface

Starbase The Hewlett-Packard Starbase graphics library provides fairly sophisticated 3D rendering with hardware assist and multi-window output using X Windows. The Hewlett-Packard TurboSRX graphics processor is capable of drawing approximately 10,000 polygons per second.

X Windows The windows used by Starbase can also be opened directly through the X Windows Library (xlib) in order to sample the mouse location, inquire window sizes, and draw 2D graphics.

Motif The Motif library is a set of user interface widgets, built on top of the X Window Library, which support buttons, menubars, sliders, text entry boxes, and a variety of layout tools to create user interface control boxes.

Tool Command Language The Tool Command Language (*tcl*) is an extensible parsing language which supports simple programming constructs (e.g., **if**, **for**, **while**, etc.), procedures, and lists[55,56]. The *tcl* parser has hooks which allow arbitrary C functions to be added to the list of commands.

Tbar Tbar is a bolio tool program which uses *tcl* to provide an interpreted interface to X Windows and Motif libraries. The user interface to the program is specified by a startup script of *tcl* code. The interface can subsequently be modified by *tcl* commands sent as messages from any of the tool processes in bolio. For each of the user interface widgets there is an associated list of callback events (e.g., **activate** for a pushbutton, **arm** and **disarm** for a togglebutton). Each widget has a list of *tcl* procedures which implement that widget's callbacks. The callback procedures may modify the user interface (e.g., drawing submenus or setting a mode to change the effect of subsequent button presses) or they may cause bolio tools to execute specified commands by sending messages through bolio. Since tbar runs as a separate process, GUI events are handled asynchronously; this allows the user to select menu items while other operations are being performed (rather than having the program "freeze" during calculations).

4.2 Spaces

An important aspect of the patient modeling problem for plastic surgery is that the planning process takes place on the skin surface. This observation leads naturally to the use of a two dimensional organization of both the plan and the anatomical data. In the case of plastic surgery of the head and neck, this two-dimensional surface is well approximated by a generalized cylinder. The coordinate system of the cylinder consists of a z coordinate which represents the position along a line which goes from the center of the top of the head straight down through the body, and a θ coordinate which represents the angle around the body. This cylindrical coordinate system is extended to three dimensions by adding an r coordinate which represents the distance from the z axis to the skin surface at each point (θ, z). Figure 4.3 shows the source data and normalized cylindrical spaces described below.

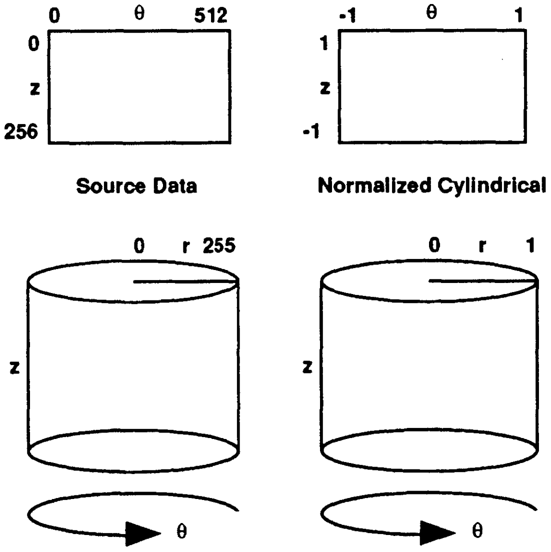


Figure 4.3: The relationship between the source data and normalized cylindrical coordinate spaces.

For the purposes of analysis and display, this coordinate system is mapped to a cartesian coordinate system. The following sections describe the coordinate spaces used in the CAPS system. For each space, there is a discussion of why it is useful in the simulation process.

4.2.1 Source Data Space

The Cyberware scanner works directly in the cylindrical coordinate space of the head and neck and generates a two-dimensional array of samples organized by θ and z. The scanner is mounted on an arm which travels 360° around the head, with the nose roughly centered in the image. The point samples represent equal angle increments from $-\pi$ to π. Each sample has four components: r, red, green, and blue; each channel is discretized to an eight bit number. The resolution of the scan is w_s samples in θ and h_s samples in z. The scan data produced by the current generation of Cyberware scanners is $(w_s, h_s) = (512, 256)$. In the CAPS system, the r channel of the scan data is stored in a block of memory $w_s h_s$ bytes long starting at location $range_{byte}$. A sample at integer locations (θ, z) is obtained using the following formula:

$$r_{byte}(\theta, z) \quad = \quad range_s[(zw_s) + \theta] \tag{4.1}$$

where z is clamped between 0 and $h_s - 1$, and θ is clamped between 0 and $w_s - 1$.

Since the range data is sampled at a resolution of 8 bits, the reconstructed surface (discussed in section 4.7 below) can show noticeable ridges at single bit changes in r. To minimize this effect, the range data is converted to a floating point array $range_{float}$ and then smoothed. The following simple box filter results in some smoothing of the single bit changes without significant loss of detail. For each sample location (θ, z):

$$range_{float}[\theta, z] = \frac{\sum_{i,j=-1}^{1} r_{byte}(\theta + i, z + j)}{9} \tag{4.2}$$

A function r_{int} can now be defined which samples the filtered floating point data at the integer sample locations:

$$r_{int}(\theta, z) \quad = \quad range_{float}[(zw_s) + \theta] \tag{4.3}$$

where z is clamped between 0 and $h_s - 1$, and θ is clamped between 0 and $w_s - 1$.

The scan samples at any (θ, z) are assumed to be point samples taken at the integer coordinates. To sample the data at non-integer locations, the following bi-linear interpolation is used. First, the sample point coordinates are split into integer and fractional components. Let:

$$\theta_{int} \quad = \quad floor(\theta) \tag{4.4}$$

$$z_{int} \quad = \quad floor(z) \tag{4.5}$$

$$\theta_{frac} \quad = \quad \theta - \theta_{int} \tag{4.6}$$

$$z_{frac} \quad = \quad z - z_{int} \tag{4.7}$$

The sample values at the four neighboring integer locations are then weighted by their distance from the sample point and summed to form the final sample value:

$$\begin{aligned} r(\theta, z) \quad = \quad & ((1 - \theta_{frac})(1 - z_{frac})r_{int}(\theta_{int}, z_{int})) + \\ & (\theta_{frac}(1 - z_{frac})r_{int}(\theta_{int} + 1, z_{int})) + \\ & ((1 - \theta_{frac})(z_{frac})r_{int}(\theta_{int}, z_{int} + 1)) + \\ & (\theta_{frac} z_{frac} r_{int}(\theta_{int} + 1, z_{int} + 1)) \end{aligned} \tag{4.8}$$

For computer implementation, this equation can be rewritten in the following form which requires fewer multiplications:

$$\begin{aligned} r(\theta, z) &= ((1-\theta_{frac})r_{int}(\theta_{int}, z_{int}) + \theta_{frac}r_{int}(\theta_{int}+1, z_{int}))(1-z_{frac}) + \\ &\quad ((1-\theta_{frac})r_{int}(\theta_{int}, z_{int}+1) + \theta_{frac}r_{int}(\theta_{int}+1, z_{int}+1))z_{frac} \end{aligned} \tag{4.9}$$

The red, green, and blue channels of the scan data are handled in an analogous manner with respect to non-integer sampling. Since there are 24 bits of color data in the original scan, there is no need to apply a smoothing filter to the color data.

Since the source data is laid out as a two-dimensional array of samples, it can be viewed and manipulated as an image using standard image processing programs. Using this approach, anomalous samples in both range and color data can be modified interactively before the analysis process begins. This approach can also be used to draw directly on the color data --- either to insert reference points and lines or to add synthetic wounds for testing and training purposes.

4.2.2 Normalized Cylindrical Space

For the surface mesh generation process described in section 4.5.1, it is important that the source data space be normalized so that equal increments in r and θ will result in lines of equal length when transformed back into world space. This is achieved by creating a normalized cylindrical space where z goes from -1 to 1 over the region scanned, and θ is reparameterized to go from -1 to 1 over the range $-\pi$ to π. Accessing the source data at a point defined in normalized cylindrical (ncyl) space is performed using the following transformation:

$$r = \frac{r(\frac{(\theta_{ncyl}+\frac{3}{2}\pi)w_s}{2\pi}, h_s - (\frac{h_s}{2}(z_{cyl}+1)))}{255} \tag{4.10}$$

4.2.3 Rectangular World Space

To graphically render the patient model, the scan data must be converted into three-dimensional world space. For the purposes of the CAPS system, a normalized coordinate system with x, y, and z going from -1 to 1 was chosen, with the z axis aligned with the cylindrical space z axis, the $-y$ axis coming out of the front of the face, and the x axis going out to the patient's left. Conversion from cylindrical space to rectangular space is achieved with the following transformation:

$$(x, y, z) = (r\cos(\theta), r\sin(\theta), z) \tag{4.11}$$

Conversion from rectangular to cylindrical space is achieved with the following transformation:

$$(r, \theta, z) = (\sqrt{x^2+y^2}, \text{arctan}_2(y, x), z) \tag{4.12}$$

where arctan_2 is the standard C language math library routine `atan2` which returns the arctangent of y/x in the range $-\pi$ to π using the signs of both arguments to determine the quadrant of the returned value. The relationship between normalized cylindrical space and rectangular world space is shown in figure 4.4.

In addition to rendering, natural boundary conditions of the model (i.e. gravity and muscle forces) are expressed in world space.

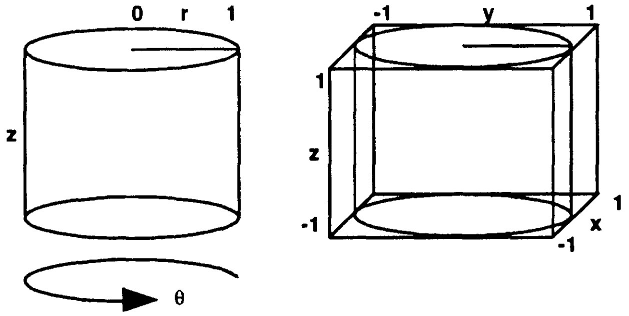


Figure 4.4: The relationship between the source data and normalized cylindrical coordinate spaces.

4.2.4 Body Coordinates

The body coordinate space is used to keep track of the locations in the model as the structure is deformed. A piecewise coordinate system is defined using the finite element discretization. Each point in the undeformed model can be given a unique coordinate by keeping track of which finite element contains the point and the local space coordinates of the point in that element. The isoparametric elements have well defined local coordinate systems used to interpolate geometry and displacements in the mechanical analysis described below (see section 4.6).

A body coordinate point is represented as a four component vector (e, r, s, t), where e is an integer identifying the element in the list of elements for the mesh, and r, s, and t are real numbers $-1 \leq r, s, t \leq 1$ which give the position within the element. The mapping from body coordinates to world coordinates is performed using the isoparametric element interpolation functions and the world space positions of the elements nodes. For the elements used in the CAPS system, the interpolation function is a second order polynomial in r, s, and t. The body coordinate to world space function is:

$$\begin{aligned} x_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)x_i^e \\ y_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)y_i^e \\ z_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)z_i^e \end{aligned} \tag{4.13}$$

where (x_w, y_w, z_w) is the world space point, h_i is the interpolation function for the i^{th} node of element e, and (x_i^e, y_i^e, z_i^e) is the world space position of the i^{th} node of element e.

It is also useful to know the displacement vector at a body coordinate point. This is accomplished using the interpolation functions to combine the nodal displacements given by the solution to the finite element matrix equations. The body coordinate to world space displacement function is:

$$\begin{aligned} u_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)u_i^e \\ v_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)v_i^e \\ w_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)w_i^e \end{aligned} \tag{4.14}$$

where (u_w, v_w, w_w) is the world space displacement of the point, h_i is the interpolation function for the i^{th} node of element e, and (u_i^e, v_i^e, w_i^e) is the world space displacement vector of the i^{th} node of element e.

Combining equations 4.13 and 4.14 leads to the following function to map body coordinates to displaced world coordinates (x'_w, y'_w, z'_w):

$$x'_w(r,s,t) = \sum_{i=0}^{19} h_i(r,s,t)(x_i^e + u_i^e)$$

node number	r, s, t	node number	r, s, t
0	(1, 1, 1)	10	(0, -1, 1)
1	(-1, 1, 1)	11	(1, 0, 1)
2	(-1, -1, 1)	12	(0, 1, -1)
3	(1, -1, 1)	13	(-1, 0, -1)
4	(1, 1, -1)	14	(0, -1, -1)
5	(-1, 1, -1)	15	(1, 0, -1)
6	(-1, -1, -1)	16	(1, 1, 0)
7	(1, -1, -1)	17	(-1, 1, 0)
8	(0, 1, 1)	18	(-1, -1, 0)
9	(-1, 0, 1)	19	(1, -1, 0)

Table 4.1: Undeformed local space nodal coordinates of the eight to twenty variable-number-of-nodes isoparametric element.

$$\begin{aligned} y'_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)(y_i^e + v_i^e) \\ z'_w(r,s,t) &= \sum_{i=0}^{19} h_i(r,s,t)(z_i^e + w_i^e) \end{aligned} \tag{4.15}$$

Bathe provides the following definition for the interpolation functions for the eight to twenty variable-number-of-nodes three-dimensional element[3, page 201][1]

$$\begin{aligned} h_0 &= g_0 - (g_8 + g_{11} + g_{16})/2 \\ h_1 &= g_1 - (g_8 + g_9 + g_{17})/2 \\ h_2 &= g_2 - (g_9 + g_{10} + g_{18})/2 \\ h_3 &= g_3 - (g_{10} + g_{11} + g_{19})/2 \\ h_4 &= g_4 - (g_{12} + g_{15} + g_{16})/2 \\ h_5 &= g_5 - (g_{12} + g_{13} + g_{17})/2 \end{aligned}$$

[1] Bathe uses the Fortran numbering convention with arrays starting at one. The equations shown here use the C numbering convention with arrays starting at zero.

$$\begin{aligned} h_6 &= g_6 - (g_{13} + g_{14} + g_{18})/2 \\ h_7 &= g_7 - (g_{14} + g_{15} + g_{19})/2 \\ h_i &= g_i \text{ for } j = 8, ..., 19 \end{aligned}$$

where

$$\begin{aligned} g_i &= 0 \text{ if node } i \text{ is not included; otherwise} \\ g_i &= G(r, r_i)G(s, s_i)G(t, t_i) \\ G(\beta, \beta_i) &= \frac{1}{2}(1 + \beta_i\beta) \text{ for } \beta_i = \pm 1 \\ G(\beta, \beta_i) &= (1 - \beta^2) \text{ for } \beta_i = 0 \end{aligned} \tag{4.16}$$

where r_i, s_i, and t_i are the coordinates of node i in undeformed local space. Table 4.1 shows the coordinates for the eight to twenty variable-number-of-nodes element.

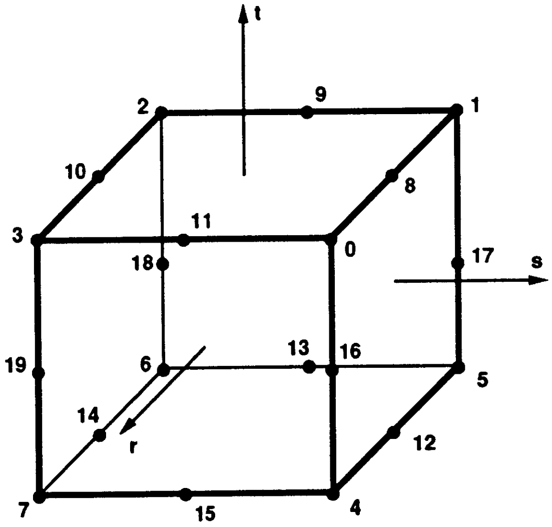


Figure 4.5: Node numbering for the eight to twenty variable-number-of-nodes isoparametric element.

These transformations are used in the visualization portion of the system to make the correspondence between the discretized surfaces of the elements and the color and range data in the patient scan. For this operation, the elements are subdivided in r, s, t space and the vertices are mapped to world space using equations 4.13. These locations are then mapped to source data space using equation 4.12. The data is then obtained using equation 4.9 or the analogous color sampling function. After solution of the finite element matrix equation, the new world space locations of the vertices are found using 4.15.

4.3 Calibration of Cyberware Data

In order to calculate a conversion factor from Cyberware range space to standard dimensions, a set of equivalent measurements were made on both the three-dimensional reconstruction and on the person whose head was scanned. A standard set of measurements in use clinically for planning reconstructive surgeries (*anthropometric* or *cephalometric* measurements) define a set of landmarks on the head which can be located fairly precisely on any patient[2,18,19,35]. The measurements are shown in tables 4.2 and 4.3. The horizontal metrics are the following: inter-ocular is the distance between the pupils; outer-canthal is the distance between the outside corners of the eyes; nasal width is the distance from *alare*[2] to *alare*; mouth width is the distance between the corners of the mouth (the *cheilions*); zygomatic width is the distance between skin points above the extreme lateral points on the zygomatic arch[3]. The vertical metrics are the following: upper facial height is the distance between the *nasion* (the point where the nose joins the brow) and the *subnasali* (at the juncture of the nose and the upper lip); the lower facial height is

[2] The *alare* is the most extreme point on the outer side of the nose.

[3] The zygomatic arch is commonly known as the cheekbone. The anthropometric landmark is known as the *zygion*.

metric	actual measurement (mm)	from model	scale factor
inter-ocular	56	0.2604	215
outer-canthal	96	0.3398	282
nasal width	37	0.1476	250
mouth width	53	0.1885	281
zygomatic width	128	0.4985	256

Table 4.2: Horizontal cephalometric measurements.

metric	actual measurement (mm)	from model	scale factor
upper facial height	57	0.2876	197
lower facial height	63	0.3207	196
upper lip height	19	0.1027	185

Table 4.3: Vertical cephalometric measurements.

the distance between the *subnasali* and the *gnathion* (the bottom of the chin); the upper lip height is the distance from the *gnathion* to the *stomion* (the center of the mouth where the lips join).

From these measurements, the average horizontal scale factor is 256.8 and the average vertical scale factor is 192.7. The ratio of vertical to horizontal scale is approximately 0.75. Figure 4.6 shows the reconstructed patient model with and without this correction factor. For this analysis, a single scale factor was used for horizontal measurements (corresponding to the x and y axes in world space) since this comes from the range data in the Cyberware scan. The rather wide variation in the horizontal scale factors indicates that separate scale factors in x and y might lead to a more accurate correction factor. However, this would require a new set of anthropometric measurements which isolate the x and y axes.

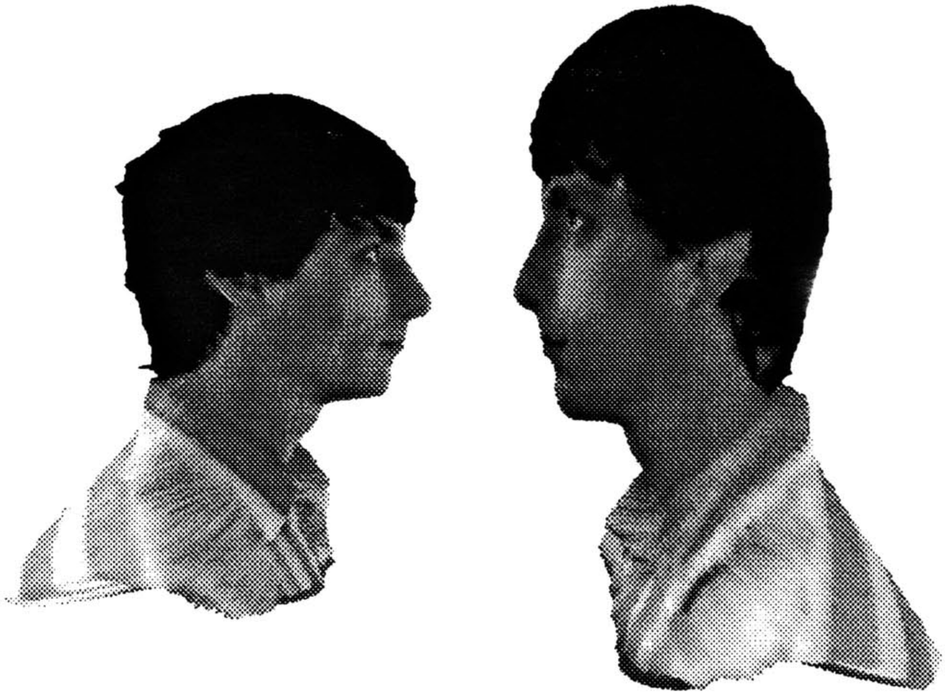


Figure 4.6: Reconstructed patient model with anthropometric scaling (left) and without (right).

This analysis indicates that a more elaborate calibration procedure should be explored before the Cyberware data is used in a clinical setting. It would be advantageous, for example, to apply markers (e.g., dots of ink) to the patient's skin before the Cyberware scan is performed rather than relying on cephalometric landmarks which can be difficult to reliably match in reconstruction and on the patient. These markers could be used to generate a local error estimate in each of the axes in the three-dimensional reconstruction. If this error was found to be significant, an error correction function would need to be defined over the space of the scan (cephalometric measurements taken by hand are usually recorded to the nearest millimeter, an equivalent or better accuracy would be expected from the computer). These landmarks would also help the user interpret the displacement results of the analysis.

4.4 Interactive Planning

The patient model exists in 3D world coordinates, but computer graphics rendering shows only a 2D projection of the model. For any view there is a screen space coordinate system which gets mapped back into world space for picking operations and then to normalized cylindrical space for operating on the skin surface. In order to operate on the entire model, methods are needed to control the view of the model and for performing the required transformation and picking operations. These are described in the following sections.

parameter	type	meaning
ViewPoint	real 3 vector	position of camera
ViewNormal	real 3 vector	direction out front of camera
ViewUp	real 3 vector	direction out top of camera
ViewDistance	real number	distance to projection plane
NearDistance	real number	distance to front clipping plane
FarDistance	real number	distance to back clipping plane
WindowCenter	real 2 vector	center of window on projection plane
WindowHalfsize	real 2 vector	1/2 size of window on projection plane
ProjectionType	boolean	orthographic or perspective projection

Table 4.4: View parameters for virtual camera model[67].

4.4.1 View Control

There are a number of choices for interactive view selection: typing 3D coordinates and vectors is accurate but tedious and is only appropriate for graphics programmers; selecting views from a menu is easy but provides limited options; continuous view control using 2D/3D input requires fast screen update to keep the user oriented; a mouse click and keyboard button interface is used in the CAPS system because it is independent of the complexity of the scene. For small scenes which are quickly rendered, repeated clicks lead to an interface much like continuous control.

The parameters used to render a computer graphics image of a 3D scene are given in table 4.4. These parameters describe the position and orientation of a virtual camera in rectangular world space. Interactive control of these parameters gives the user the ability to look at the objects in world space from different points of view.

parameter	type	meaning
Window	window identifier	X window to get input events from
WorldUp	real 3 vector	ViewUp to use when camera roll inhibited
use_WorldUp	boolean	inhibit/allow camera roll
FocalDepth	real number	distance to center of interest
ViewCamera_stepsize	real number	distance to move viewpoint

Table 4.5: Extra view parameters for interactive camera control.

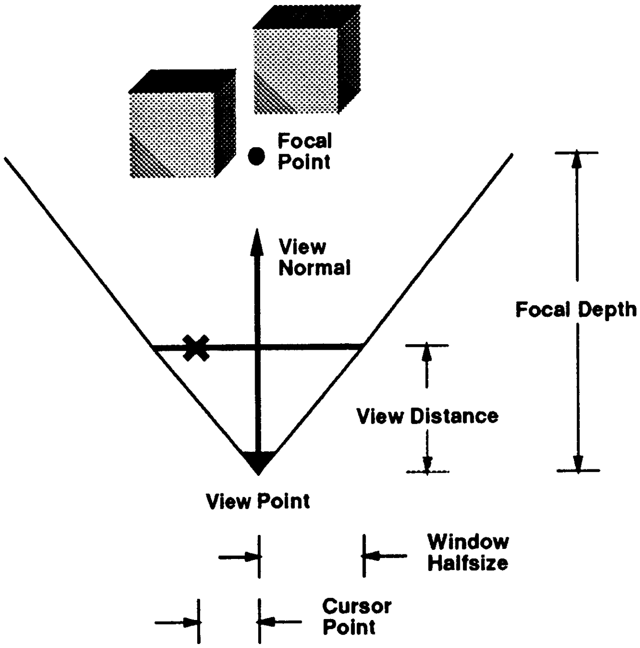


Figure 4.7: Schematic of view parameters as seen from above.

Preliminaries Rotation around a point in world space is performed using the click-based interface using the following technique. First, the location of the mouse click is converted from device coordinates (x, y) (in units of screen pixels) to normalized screen coordinates (s_x, s_y) which go from -1 to 1:

$$\begin{aligned} s_x &= 2x/w - 1 \\ s_y &= 2(1 - y/h) - 1 \end{aligned} \tag{4.17}$$

where w is the width of the window in pixels, and h is the height of the window in pixels. The origin of device coordinates is the upper left corner of the window, and the origin of normalized screen coordinates is the lower left corner.

Next, a camera-space coordinate system is defined using the view parameters. The origin v_p is at the ViewPoint, and the axes are the ViewNormal, v_n, the ViewUp, v_u, and a vector perpendicular to v_u and v_n called v_r calculated using the cross product operator:

$$v_r = v_n \times v_u \tag{4.18}$$

where the v_n and v_u are normalized, perpendicular vectors. The other quantity of interest is the center of interest in the scene about which to rotate. This point, p_{coi} is defined in terms of the camera-space coordinate system as follows:

$$p_{coi} = v_p + f_d v_n \tag{4.19}$$

where f_d is the FocalDepth.

Rotation Around Object With these values calculated, the rotation is calculated by first translating the ViewPoint to lie on the line defined by p_{coi} and the mouse click location (s_x, s_y) with:

$$v_p' = v_p - f_d s_x v_r + f_d s_y v_u \tag{4.20}$$

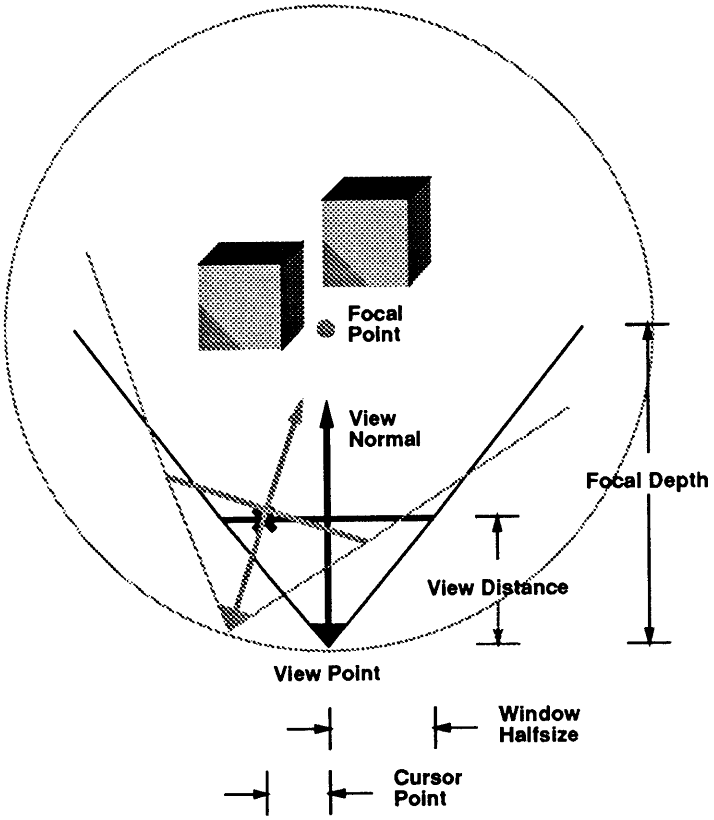


Figure 4.8: Rotation around FocalDepth by amount selected with screen mouse click.

then translating it so that it remains f_d away from the p_{coi}. This final translation is performed by first calculating the new ViewNormal v_n^{new} as a normalized vector from the new ViewPoint to the center of interest:

$$v_n^{new} = \frac{p_{coi} - v'_p}{||p_{coi} - v'_p||} \tag{4.21}$$

and then calculating the new position of the ViewPoint:

$$v_p^{new} = p_{coi} - f_d v_n^{new} \tag{4.22}$$

If the use_WorldUp boolean of the extra view parameters is $true$, then the WorldUp vector is copied into the ViewUp. The ViewUp is then modified to ensure that is perpendicular to the new ViewNormal by subtracting parallel component and normalizing:

$$v_u^{new} = \frac{v_u - (v_u \cdot v_n^{new})v_n^{new}}{||v_u - (v_u \cdot v_n^{new})v_n^{new}||} \tag{4.23}$$

where $||v||$ denotes the euclidean length of v.

Rotation Around Camera For rotation around the camera (panning), only the ViewNormal and the ViewUp need to be modified. The operation described here updates the view parameters so that the point directly under the mouse click moves to the center of the view. Note that this does not require any information about the objects other than direction to the object in camera space (which the user supplies through the location of the mouse click).

The rotation is performed by the new p_{coi} in the plane defined perpendicular to the ViewNormal at a distance FocalDepth from the ViewPoint. This point, called the lookat point (l_p) is calculated as:

$$l_p = p_{coi} + (s_x w_h^x / v_d)v_r + (s_y w_h^y / v_d)v_u \tag{4.24}$$

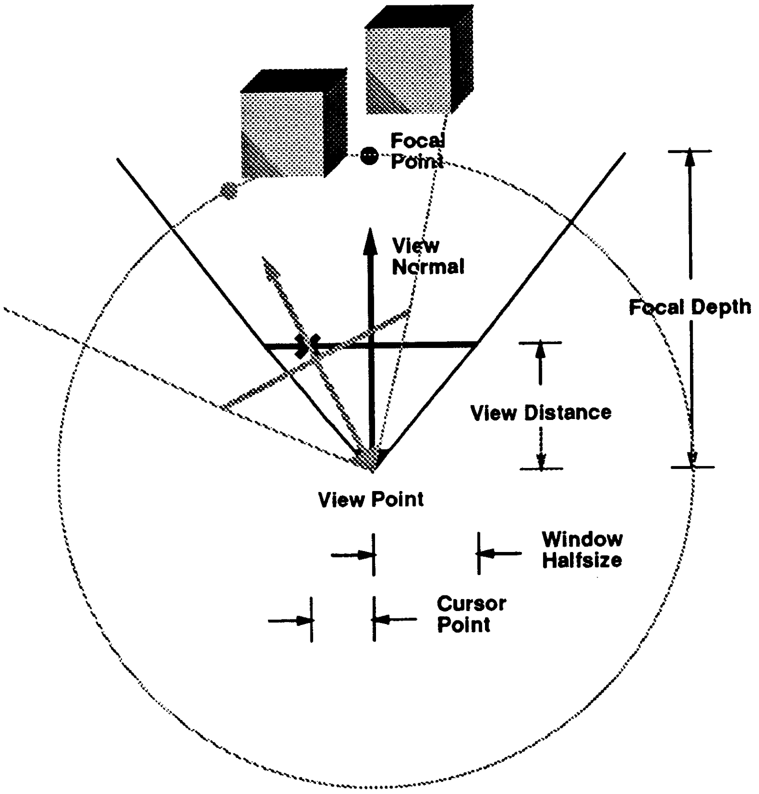


Figure 4.9: Rotation around camera by amount selected with screen mouse click.

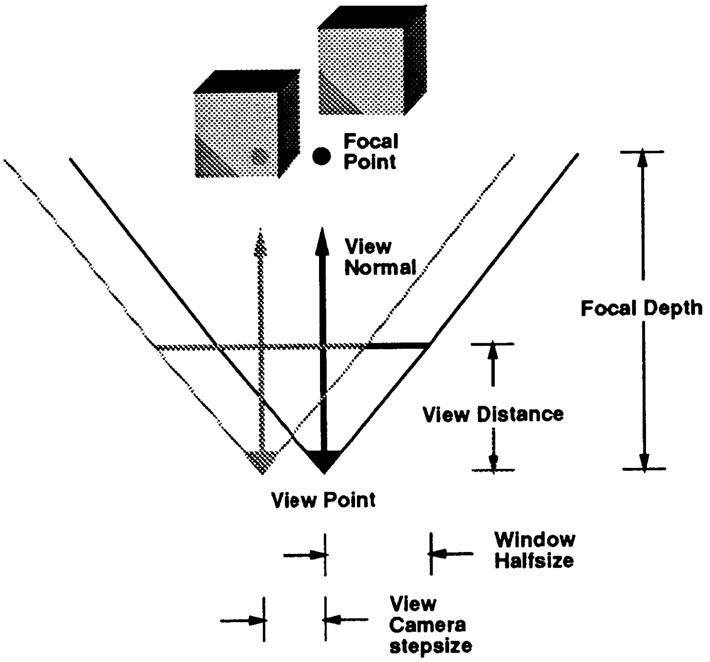


Figure 4.10: Translation of ViewPoint in camera space.

where w_h^x is the WindowHalfsize in x, w_h^y is the WindowHalfsize in y, and v_d is the ViewDistance. The new ViewNormal is the normalized vector from the ViewPoint to the lookat point:

$$v_n^{new} = \frac{l_p - v_p}{||l_p - v_p||} \tag{4.25}$$

The final operation is to normalize the ViewUp vector to the new ViewNormal as given in equation 4.23.

Movement in Camera Space The final type of interactive view control available in the CAPS system is translation of the ViewPoint in camera space. This simply corresponds to adding to the ViewPoint an increment of ViewCamera_stepsize in the direction of one of the camera space axes. For example, moving to the left (trucking left) is accomplished with the following calculation:

$$v_p^{new} = v_p - v_s v_r \tag{4.26}$$

where v_s is the ViewCamera_stepsize. This is illustrated in figure 4.10.

4.4.2 Operating on the Surface

Planning the operation on the skin surface requires a technique for mapping selections on the screen window back onto the surface of the object. I.e., after selecting a view and switching the mouse to pick mode, a mouse click on the window should pick a point on the patient model which appears directly beneath the mouse location. This is implemented in the CAPS system using the polyhedral reconstruction of the scan data. The polyhedron is created by making vertices at the scan data sample points (transformed from source data space to world coordinates) and connecting each set of four adjacent pixels with a polygon. To reduce the number of polygons, a subsampled version of the source data can be used.

The surface point is found by calculating the intersection between the polyhedron and a line, called the pick line, from the ViewPoint through the point selected on the window. The direction of the line is found in a manner analogous to the method used to find the new view normal in a rotation around the camera as in equation 4.24; for this operation the vector v_n^{new} is called the pick vector, v_{pick}. The pick line to be intersected is expressed in parametric form as:

$$v_p + t v_{pick} \tag{4.27}$$

The pick line must be checked in turn against each of the polygons in the polyhedral patient model. Checking the intersection of a line with a polygon is done in two steps: first calculating the intersection of the line with the plane of the polygon, and then checking to see if that point lies within the boundaries of the polygon. A parametric representation of the plane of a polygon can be found by finding the vectors along any two non-collinear edges of the polygon and making them perpendicular (using the method shown in equationrefeqn:normup); call these vectors b and c. The parametric equation of the polygon plane is then $a + ub + wc$, where a is the position of any point of the polygon. The following equations, adapted from Mortenson[51] give the equation for calculating the value of t at which intersection of pick line and a plane:

$$t = \frac{(b \times c) \cdot a - (b \times c) \cdot v_p}{(b \times c) \cdot v_{pick}} \tag{4.28}$$

The intersection point p_{pick} is then given by equation 4.27.

This intersection point must then be checked against the edges of the polygon to see if it is contained in the polygon. The CAPS system does the operation by checking the point against the set of planes defined by each of the edges of the polygon and the polygon normal. If the point lies on the same side of all these planes, then it is within the (convex) polygon. This algorithm was selected because it is robust in the case of non-planar polygons, such as those generated in the quadrilateral mesh defined by the range data. Faster algorithms for this task have been developed for use in ray-tracing renderers which could be adapted to this application (see, for example, the discussion in [21]).

Once the intersection point of the pick line with the polyhedron has been found, it is converted to source data space and offset in r to corresponding sample point in the range data and then transformed back to world space.

After a set of points on the surface is created, it is useful to be able to pick a point by clicking the mouse on that point. This is accomplished by finding the closest point to the pick line. This operation is implemented in the CAPS system by calculating the vector from the ViewPoint to each point, normalizing it to unit length and taking the dot product of that vector with a normalized version of the v_{pick} vector. The point which yields the dot product closest to 1 is the closest point to the position of the mouse click.

4.4.3 Defining a Hole — Incision

An incision through the skin is topologically a hole, but geometrically it is infinitesimally thin until it is deformed by the mechanical simulation. Rather than requiring the user to enter the points on both sides of the incision, the incision is entered by picking a sequence of points corresponding to the cutting path of the scalpel. This list of points is then converted into a loop of points describing the hole. Figure 4.11 illustrates this mapping. The points are entered by selecting locations on the skin surface using the screen space to skin surface transformation described in the previous section.

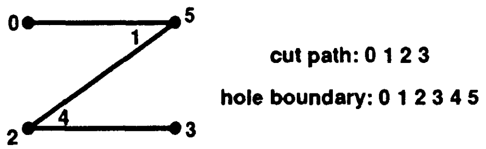


Figure 4.11: The relationship between the points along the cutting path (entered by the user) of the scalpel and the automatically generated border of the hole.

The incision line can be modified by picking one of the points and then selecting a new point on the skin surface to move that vertex. At this point, all data about edge lengths and border angles is available in world space. This data could be drawn on the screen at the appropriate locations on the patient model to help the surgeon meet geometric requirements of the incision type, although this feature is not yet fully implemented in the prototype.

4.4.4 Modifying the Hole — Excision

An excision of tissue is defined by picking one of the points in the hole border and offsetting it from its corresponding point on the other side of the hole, with the result that the hole is no longer infinitesimally thin. Moving one border point creates a quadrilateral, while moving more than one creates an arbitrary polygonal shape. The point picking algorithm described above cannot be used for this picking operation because the two points on either side of the hole are coincident. The algorithm could be modified to distinguish between coincident points by determining on which side of the incision line the pick line intersects the skin surface. For the CAPS system a simpler approach is used in which a menu item on the control panel is used to select which point is to be moved.

4.4.5 Closing the Hole — Suturing

Suturing refers to the sewing together of edges of the incision hole. In the finite element simulation, this is accomplished by suture constraint equations for the individual nodes in the continuum mesh. Even for a simple wound closure dozens of nodes must be sutured. Selecting each pair of nodes by hand would be tedious. Instead, the continuum mesh generator defines the nodes to be sutured from a description of which edges of the hole border are to be brought together. Figure 4.12 shows the pre- and post-operative topology desired for a simple excision. For this configuration, the edge sutures are specified as ((0,1), (0,5)), ((1,2), (5,4)), and ((2,3), (4,3)). When the same point is included in both of the edges to be sutured, the mesh generator recognizes this as a corner being closed and does not define any sutures for the nodes corresponding to that point. The suture edges for the Z plasty shown in figure 4.11 are ((5, 0), (5, 4)), ((2, 1), (2, 3)), and ((0, 1), (3, 4)).

In the CAPS system, the suture edges are specified by selecting a menu item corresponding to the type of surgical procedure being performed. This technique works because the suture relationships depend only on the pre-defined topology of the procedure and not the interactively specified geometry. The menu item approach has the advantage that the suture conditions do not need to be re-entered for each simulation of the same surgical procedure. The drawback of this approach is that in order to simulate a new procedure, the suture relationships described above must be worked out by hand and added to the menu by editing the tbar start-up script. While this is not a very difficult task, a more flexible solution would again require an algorithm for picking points on the hole border as discussed above. Picking these points in the proper sequence would then define the suture relationships for a new surgical procedure. These suture relationships could then be added to the menu for use in future analyses.

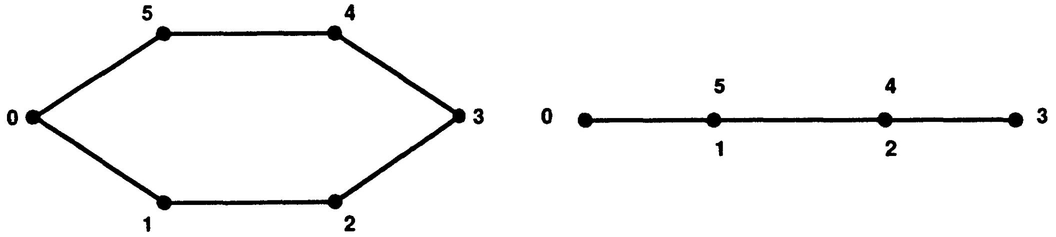


Figure 4.12: Pre- and post- operative topology of a simple excision.

4.5 Mesh Generation

The mesh generation algorithm consists of two major steps: surface meshing and continuum meshing. The surface meshing portion of the algorithm grows a mesh out from the incision hole border along the skin surface. Surface meshing is performed in normalized cylindrical space ignoring the r coordinate. After the surface mesh is generated, the mesh is snapped back to the skin surface by looking up the r coordinate with equation 4.9. The surface mesh consists of triangles and quadrilaterals.

The continuum meshing portion of the algorithm grows the surface mesh in from the skin surface to the bone surface along the r axis. Triangles are extruded into wedge elements and quadrilaterals are extruded into cuboid elements. Edges shared by polygons in the surface mesh are extruded into shared faces in the continuum mesh. Each vertex in the surface mesh defines a set of nodes in the continuum mesh. Figure 4.14 shows a cross section of the nodes and elements created by the continuum mesh algorithm. Heavy lines are edges from the surface mesh, and filled circles are nodes from the surface mesh.

A suture condition specified between two edges on the incision boundary is converted into suture constraints between each pair of nodes generated from those edges. Nodes on the bottom layer of the continuum mesh which do not have suture constraints are marked as fixed in all three degrees of freedom. All other nodes in the continuum mesh are unconstrained.

4.5.1 Surface Meshing

The surface meshing algorithm consists of two stages: 1) the traverse border of the incision looking for angles larger than a set threshold t_1 --- convert them to triangles in the surface mesh and update the border --- continue until no more angles need to be filled; 2) go around border adding a layer of quadrilaterals of thickness l_1 --- a quadrilateral is added for each edge in the border, and an extra quadrilateral is added at edges which join at an angle less than a specified threshold t_2. A surface mesh generated from a Z-plasty incision is shown in figure 4.13. The triangles numbered 1 and 2 were are generated by stage 1 operations, while the quadrilaterals numbered 3-8 were generated by one pass through stage 2.

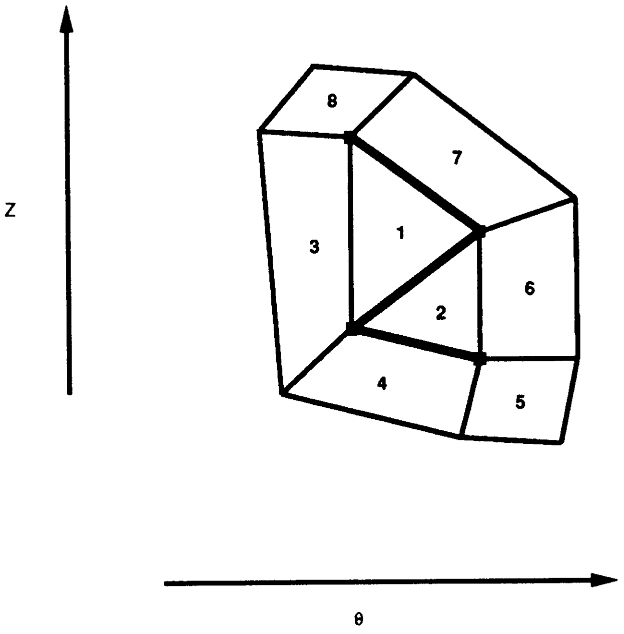


Figure 4.13: Example polygons generated by the surface meshing algorithm from Z-plasty incision. See text for explanation.

Stage 1 is implemented as follows. For each vertex v_i in the border list, examine the angle between the edges (v_i, v_{i+1}) and (v_{i+1}, v_{i+2}).[4] If this angle is greater than t_1, add triangle (v_{i+2}, v_{i+1}, v_i) to the surface mesh (30° is the default t_1 threshold angle in the CAPS system) and delete vertex v_{i+1} from the border list. Continue this process until no more triangles are added in a complete traversal of the border list. After stage 1, the region defined by the border list will be nearly convex (no concavities will be greater than t_1).

Stage 2 has two substages: creating the new border list and joining the new and old border lists with quadrilaterals. The first substage proceeds as follows. Create an empty list to store the new border. For each vertex v_i in the current border list, let n_1 be the outward normal from edge (v_{i-1}, v_i) and n_2 be the outward normal from edge (v_i, v_{i+1}).[5] Examine the angle between the edges (v_{i-1}, v_i) and (v_i, v_{i+1}). If the angle is greater than t_2 then add a vertex to the new border with vertex position of $v_i + l_1(n_1 + n_2)$. If the angle is less than t_2 then mark v_i as expanded, add three vertices to the new border with vertex positions of $v_i + l_1 n_1$, $v_i + l_1(n_1 + n_2)$, and $v_i + l_1 n_2$. (90° is the default t_2 threshold in the CAPS system.)

The second substage of stage 2 is to connect new and old border lists with quadrilaterals as follows. Let j index the new border list and i index the current border list; initialize i and j to zero. For each vertex v_i, if v_i is marked as expanded, add quadrilateral $(v_i, v_j, v_{j+1}, v_{j+2})$, increment j by two. Add quadrilateral $(v_i, v_j, v_{j+1}, v_{i+1})$. Increment i and j by one. Make the new border the current border. The entire stage 2 process is repeated once for each layer to be added to the surface mesh.

[4] Accesses to vertices in the node list wrap around if the $i + n$ is greater than the length of the list. Similarly, negative indices wrap back to the end of the list.

[5] In normalized cylindrical space the outward normal of an edge (θ, z) is $(-z, \theta)$.

4.5.2 Continuum Meshing

Generation of the continuum mesh from the surface mesh is accomplished by making a mapping from vertices and polygons in the surface mesh to nodes and elements in the continuum mesh. First we look at the numbering of nodes in the standard isoparametric element, then we look at the numbering of the vertices and edges in the surface mesh, and then at the correspondence between these numbering schemes. The continuum meshing algorithm converts the surface mesh into an arbitrary number of layers of elements, each layer being of an arbitrary thickness.

Table 4.1 and figure 4.5 show the ordering of nodes which must be created. Nodes 0-3 called the top_nodes, are the corners of face 0 ($t = 1$); nodes 4-7, called the bottom_nodes, are the corners of face 1 ($t = -1$); nodes 8-11, called the top_mid_nodes are the nodes in the middles of the edges on the top face; nodes 12-15, called the bottom_mid_nodes are the nodes in the middle of the edges on the bottom face; nodes 16-19, called the center_nodes are the nodes in the center of the edges joining face 0 to face 1.

In the surface mesh, there is a set of vertex points connected by a set of polygons. Each polygon has a list of the vertices which define its shape. An edge of the polygon is defined by each pair of vertices in the list and the last and first vertices in the list. Call the number of vertices in the surface mesh N_v, the number of edges N_e, the number of polygons N_p, and the number of element layers in the output mesh N_l. The number of nodes generated from vertices is $(2N_l + 1)N_v$. The number of nodes generated from edges is $(N_l + 1)N_e$. The total number of nodes created by the continuum meshing algorithm is $N_l(2N_v + N_e) + N_v + N_e$. The total number of elements created is N_lN_p. A data structure is maintained for each layer of elements which keeps track of the numbering of nodes in the layer. As each node is created, its position is calculated and its index in the list of nodes for the structure is recorded in the layer data structure.

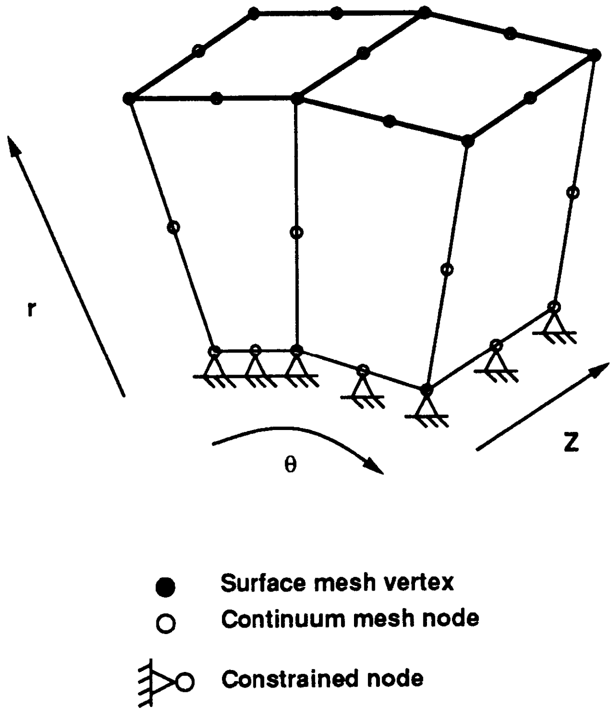


Figure 4.14: Two elements of the continuum mesh.

For the top layer of elements, the top_nodes are positioned at the points of the surface mesh vertices. The positions of the top_mid_nodes of the top face are calculated by taking the midpoints of each polygon edge and offsetting those points to lie on the skin surface. The positions of the bottom_nodes are calculated by offsetting the positions of the top_nodes in r by the thickness of the layer. The positions of the bottom_mid_nodes are calculated by offsetting the positions of the top_mid_nodes by the thickness of the layer. The positions of the center_nodes are calculated by offsetting the positions of the top_nodes by 1/2 the layer thickness. Subsequent layers of elements are generated in an analogous manner with the exception that rather than creating new nodes for the top_nodes and the top_mid_nodes, the indices of the previous layer's bottom_nodes and bottom_mid_nodes are copied instead.

Once all the nodes have been created, the elements must be created. One element per layer is created for each polygon in the surface mesh. These elements must contain a correctly ordered list of the node indices. This list of indices for the top_nodes is obtained by looping through the vertices of the polygon and looking up the node indices from the data structure of the layer corresponding to the top of the element. The indices for the bottom_nodes and the center_nodes are obtained in the same manner, except using the appropriate node indices from the layer data structure. The list of indices for the top_mid_nodes and the bottom_mid_nodes are found by looping over the edge list for each polygon and finding that edge's index in the list of edges for the surface mesh. That index is then used to find the appropriate node index by looking up the node in the appropriate layer data structure.

4.6 FEM Formulation

The *fem* module of the CAPS system implements a linear static displacement-based formulation of the finite element method to solve the elasticity equilibrium equations. The implementation closely follows the procedure described in Bathe[3]. The following sections describe the steps required for analysis and calculation of the solution variables used in the visualization module. The main differences between the algorithm presented here and a standard finite element program are in the calculation of the boundary conditions, so these topics are treated in greater detail. Readers are referred to Bathe's excellent text for derivations, further details, and motivation on FEM.

Figure 4.15 shows an example screen dump of the fem program. The largest window shows an arch structure deformed by a uniform load. The user can interactively select the loading conditions using the tbar sliders in the upper left of the image. The arch consists of nine isoparametric finite elements with fixed support on the floor. Lines indicate the original shape of the structure. The pressure stresses are mapped onto the surface as changes in brightness.

All the code described here was written by the author specifically for this project. In addition, *fem* makes use of several libraries developed by the author and others at the MIT Computer Graphics and Animation Group. Most notably, I have made extensive use of a library of matrix utility routines developed by David Chen. I also make use of a library of matrix routines optimized for sparse matrices to perform LU decomposition of the assembled K matrix[36,37].

4.6.1 Solution Algorithm

The program uses the following procedure to solve for the displacements of a structure. Steps 1, 2, and 3 must be performed for each change in the displacement boundary conditions. For each change in the loading boundary condition, only step 3 needs to be performed.

1. Allocate and clear K ($3nx3n$, where n is the number of nodes), R ($3nx1$), U ($3nx1$), C ($6x6$), $\overline{h}$ ($3x20$), γ ($3x20$), B ($6x20$), J ($3x3$), J^{-1} ($3x3$), and Km ($60x60$).

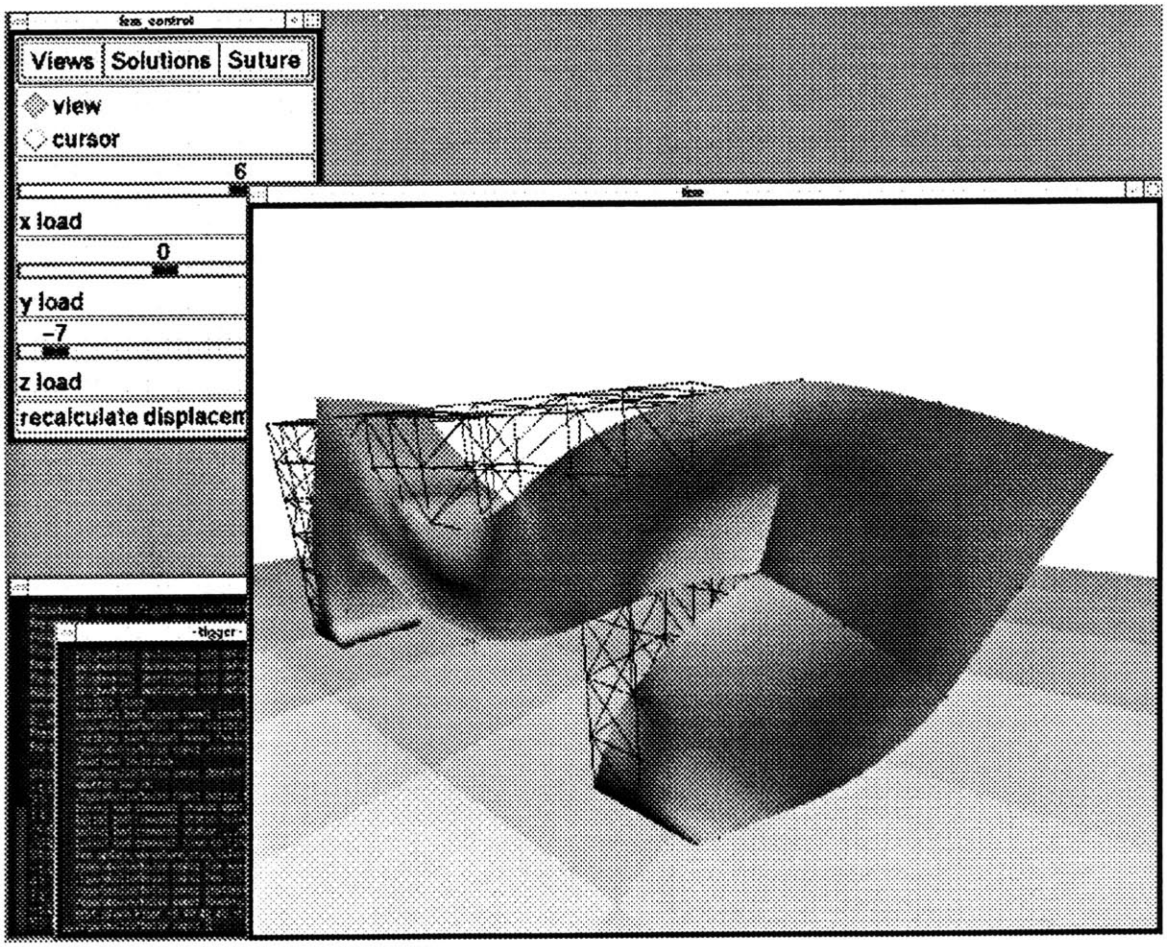


Figure 4.15: A screen image of an arch structure under uniform loading as analyzed by the fem program.

2. For each element:

   (a) Clear Km.

   (b) fill the material properties (stress-strain) matrix.

$$C = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & & & \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & & & \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & & & \\ & & & \frac{1-2\nu}{2(1-\nu)} & & \\ & & & & \frac{1-2\nu}{2(1-\nu)} & \\ & & & & & \frac{1-2\nu}{2(1-\nu)} \end{bmatrix} \tag{4.29}$$

   where E is the Young's modulus of the element, ν is the Poisson's ratio, and blank elements are zeros.

   (c) For each integration point (of the 27 combinations of values given in table 4.6):

      i. calculate the Jacobian

$$J = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix} \tag{4.30}$$

      where

$$\frac{\partial \kappa}{\partial \lambda} = \sum_{i=0}^{19} \kappa_i^e \frac{\partial h_i}{\partial \lambda}; \quad \kappa = x, y, z; \lambda = r, s, t \tag{4.31}$$

      ii. calculate J^{-1}

      iii. calculate $\overline{h}$ where $\overline{h}_{1,i} = \frac{\partial h_i}{\partial r}$, $\overline{h}_{2,i} = \frac{\partial h_i}{\partial s}$, and $\overline{h}_{3,i} = \frac{\partial h_i}{\partial t}$; with i running from 0 to 19.

      iv. calculate $\gamma = J^{-1}\overline{h}$

v. fill the displacement to strain matrix B:

$B_{0,i} = \gamma_{0,i}$,

$B_{1,20+i} = \gamma_{1,i}$,

$B_{2,40+i} = \gamma_{2,i}$,

$B_{3,i} = \gamma_{1,i}$,

$B_{3,20+i} = \gamma_{0,i}$,

$B_{4,20+i} = \gamma_{2,i}$,

$B_{4,40+i} = \gamma_{1,i}$,

$B_{5,i} = \gamma_{2,i}$,

$B_{5,40+i} = \gamma_{0,i}$;

with i running from 0 to 19.

vi. calculate $Km = Km + \alpha_{r,s,t} B^T C B \det J$, where $\det J$ denotes the determinant of the matrix J.

(d) Add element stiffness Km to global stiffness K:

$K_{LM_i,LM_j} = K_{LM_i,LM_j} + Km_{i,j}$,

$K_{LM_i,n+LM_j} = K_{LM_i,n+LM_j} + Km_{i,20+j}$,

$K_{LM_i,2n+LM_j} = K_{LM_i,2n+LM_j} + Km_{i,40+j}$,

$K_{n+LM_i,LM_j} = K_{n+LM_i,LM_j} + Km_{20+i,j}$,

$K_{n+LM_i,n+LM_j} = K_{n+LM_i,n+LM_j} + Km_{20+i,20+j}$,

$K_{n+LM_i,2n+LM_j} = K_{n+LM_i,2n+LM_j} + Km_{20+i,40+j}$,

$K_{2n+LM_i,LM_j} = K_{n+LM_i,LM_j} + Km_{40+i,j}$,

$K_{2n+LM_i,n+LM_j} = K_{n+LM_i,n+LM_j} + Km_{40+i,20+j}$,

$K_{2n+LM_i,2n+LM_j} = K_{n+LM_i,2n+LM_j} + Km_{40+i,40+j}$;

with i and j going from 0 to 19, where LM is the 60x1 vector relating the element degrees of freedom to the structure degrees of freedom.

(e) Apply gravity and muscle boundary conditions (see section 4.6.2).

(f) Apply displacement and suture boundary conditions (see section 4.6.2).

(g) Decompose the stiffness matrix into product of lower triangular and upper triangular matrices:

$$\text{Decompose}(K) = K_{LU}. \tag{4.32}$$

3. For each load condition:

(a) Calculate new load vector R.

(b) Solve $K_{LU}U = R$.

(c) Copy displacements from U back into the node data structure.

(d) Repeat the last two steps under changing load and displacement conditions while K is still valid for the structure.

The degrees of freedom are stored in the displacement vector as:

$$U^T = (u_0, u_1, \ldots, u_{n-1}, v_0, v_1, \ldots, v_{n-1}, w_0, w_1, \ldots, w_{n-1}) \tag{4.33}$$

4.6.2 Imposition of Boundary Conditions

The basic governing equation for a linear static finite element formulation is

$$KU = R \tag{4.34}$$

This equation relates the generalized nodal point loads R to the nodal point displacements U through the structure stiffness matrix K. K is found using the algorithm described above. To obtain a solution to the equation, there must be a boundary condition specified for each of the degrees of freedom in the problem (i.e., there must be one known quantity on the right hand side for each row of the matrix equation). The generalized nodal point load vector is filled with either nodal point loads (natural boundary conditions) or with known values of displacement (essential boundary conditions). For a three-dimensional structure, essential boundary conditions must be imposed at no fewer than six degrees of freedom to constrain the six rigid body motions of the structure. The next sections describe the boundary conditions applied in the fem program.

Gravity Forces

Loading boundary conditions used in this program are obtained by integrating world space body forces over the volume of the structure to find the nodal point equivalent forces. An example of a body force is gravity, which is assumed to be a load dependent on the mass density of the material ρ. The nodal point gravity load vector is given by:

$$R_G = \int_V \rho H^T f_g dV \tag{4.35}$$

where f_g is a three-vector indicating the magnitude and direction of the gravity load per unit volume, H^T is the displacement interpolation matrix, and V is the volume of the structure. (In the fem program, we assume that ρ is constant over the volume and incorporate its size into the magnitude of f_g.) This calculation is effectively carried out by integrating the load on the individual elements and summing the result into a load for the entire structure using the equation:

$$R_G = \sum_{m=1}^{M} P^{(m)} R_G^{(m)} \tag{4.36}$$

where P is a $3Nx60$ matrix which maps the element degrees of freedom to the structure degrees of freedom, N is the number of nodes in the structure, and M is the number of elements.[6] The element gravity load vector is given by the following equation:

$$R_G^{(m)} = \int_{V^{(m)}} H^{(m)T} f_g dV^{(m)} \tag{4.37}$$

where $V^{(m)}$ is the volume of element m, and $H^{(m)}$ is given by:

$$H^{(m)} = \begin{bmatrix} h_0 & h_1 & \ldots & h_{n-1} & & & & & & & & \\ & & & & h_0 & h_1 & \ldots & h_{n-1} & & & & \\ & & & & & & & & h_0 & h_1 & \ldots & h_{n-1} \end{bmatrix} \tag{4.38}$$

where blank elements indicate zeros. The integration in equation 4.37 is carried out using Gauss-Legendre numerical integration where the integral is replaced by a weighted sum of the function evaluated at specific integration points. This summation is given as:

$$R_G^{(m)} = \sum_{i,j,k=1}^{3} \alpha_{i,j,k} H^{(m)T} R_G^{(m)} \det J \tag{4.39}$$

where the weights $\alpha_{i,j,k} = \alpha_i \alpha_j \alpha_k$, and $H^{(m)T}$ and J are evaluated at the integration points (r_i, s_j, t_k). Numerical values of the integration points and weighting values are given in table 4.6. By summing over the 27 combinations of i, j, and k, the value of the integral over the parabolic space defined by an eight-to-twenty node isoparametric element is calculated exactly.

Muscle Forces

A particular load condition of importance for simulation of soft tissue is loading due to muscle action in the tissue. Facial muscles are closely meshed with the other soft tissue

[6] The matrix P is introduced here as a notational convenience; in the code the appropriate load values are added to the global vector as they are calculated and P is not explicitly constructed.

i,j,k	integration point	α
1	−0.774596	0.555555
2	0.000000	0.888888
3	0.774596	0.555555

Table 4.6: Sample points and weights for Gauss-Legendra numerical integration over the interval −1 to 1[3].

parameter	type	meaning
p_o	real 3 vector	origin point
r_o	real number	origin radius
p_i	real 3 vector	insertion point
r_i	real number	insertion radius
l	real number	muscle load

Table 4.7: Parameters of the facial muscle model.

and do not act with single points of attachment (unlike skeletal muscles which attach through tendons at fairly localized origins and insertions in bone). To model this effect, I use a version of the body force calculation given above and assume a spherical volume of muscle attachment over which the force is applied. The muscle is described by five parameters given in table 4.7. Muscle parameters are given in world space so that the same parameters can be used for different finite element meshes.

At a given point q, the muscle force vector f_m^i corresponding to the action of muscle i is given by:

$$f_m^i(q) = w(q, p_i^i, r_i^i) l^i (p_o^i - p_i^i) + w(q, p_o^i, r_o^i) l^i (p_i^i - p_o^i) \tag{4.40}$$

where w is a weighting function which scales the force at a point based on its distance from the point of origin or insertion. For the spherical muscles used in the fem program, this function is given by:

$$w(q,p,r) = \begin{cases} 1 & \text{if } \|p-q\| < r \\ 0 & \text{otherwise} \end{cases} \tag{4.41}$$

Other functions w could be used, for example, to make a different origin or insertion shape, or to make the muscle force fall off over a region based on the distance from the attachment shape. For muscles which have their origin in bone, and thus generate no force on the soft tissue, the parameter r_o is set to zero.

These muscle forces can be applied to the structure using the same technique described above for gravity forces, with the exception that the force is now dependent on the world space position of the integration point. This is implemented using the body coordinate to world space function given in equation 4.13 at each of the integration points; let the vector function $x(r,s,t) = (x_w(r,s,t), y_w(r,s,t), z_w(r,s,t))$. The muscle force vector for an element m is given by:

$$R_m = \sum_{i,j,k=1}^{3} \left(\alpha_{i,j,k} H^{(m)T} \sum_{a=1}^{N_m} f_m^a(x(r_i, s_j, t_k)) \right) \tag{4.42}$$

where N_m is the number of muscles acting on the structure. A number of muscle load functions may be combined to approximate the effect of complete muscles. In the face, muscles have complex forms such as sheets (a number of parallel fibers) or sphincters (fibers surrounding an opening such as the mouth), these forms can be approximated using a collection of muscle force functions arranged in the shape of the muscle.

Prescribed Displacements

The second type of boundary conditions of interest are the displacement or essential boundary conditions. These correspond to points on the structure at which a known displacement is required. For example, we require that the inner layer of nodes in the soft tissue model have zero displacement at the points where it is fixed to the bone. These boundary conditions are implemented by rearranging the equation $KU = R$ so that the desired condition is enforced through a constraint equation. For example, if we know that a given node i is fixed in the world space x direction, we use the constraint equation $u_i = 0$. This condition is imposed by replacing row i of K with row i of the identity matrix and setting the ith element in R to zero. During the solution of $KU = R$, this has the effect of forcing the value of u_i to zero. To specify a non-zero known displacement, i.e., $u_i = \delta_i$, the same procedure is used, except with δ_i placed in the ith element of R.

Suture Constraints

In the simulation of plastic surgery, we encounter the unique problem of suturing together edges of an open wound. In essence, what we wish to do is take elements which were unconnected in the original mesh and bring them together so that they share a common face, while at the same time accounting for the stress generated in the structure in response to this rearrangement. This can be accomplished using a set of constraint equations similar to those used to impose prescribed displacement boundary conditions. To bring two faces together, we need to constrain the corresponding nodes of the two elements to have the same position after the solution of $KU = R$. The calculation of which nodes to bring together is described above (see section 4.5.2). All three degrees of freedom for each node must be constrained for the nodes to be coincident. Here we look at the technique for constraining one degree of freedom u_i to u_s where u_i is called the *constrained* degree of freedom and u_s is the *unconstrained* degree of freedom, or the degree of freedom being *sutured to*. The desired constraint equation is

$$x_i + u_i = x_s + u_s \tag{4.43}$$

i.e., the initial position plus the displacement for the two degrees of freedom must be equal. The first step is to calculate the initial distance between the two nodes in this degree of freedom. Let

$$\beta = x_s - x_i \tag{4.44}$$

Then, in terms of the stiffness equation, we wish to enforce the equation:

$$u_i - u_s = \beta \tag{4.45}$$

Since we know this equation must hold, we can modify the equation $KU = R$ so as to combine the effect of u_i into the terms for u_s. This is done first by adding to the R vector the load corresponding to displacing degree of freedom i by a distance β:

$$R = R - \beta K_i^T \tag{4.46}$$

i.e. β times the ith column of K is subtracted from R. Next, we combine the terms involving degree of freedom i to those of s (both the stiffness and load components). This is done by adding row K_i to row K_s, adding column K_i^T to column K_s^T, and adding element R_i to element R_s. The final step is to express the constrained degree of freedom u_i in terms of the unconstrained degree of freedom u_s. This is implemented by adding equation 4.45 directly into $KU = R$ in the row formerly used for the stiffness components of degree of freedom u_i. for this, the elements of row K_i and column K_i^T are set to zero, with the exceptions that $K_{i,s} = -1$ and $K_{i,i} = 1$. R_i is set to β. With this set up, the solution to $KU = R$ directly gives the desired displacements in the vector U.

4.7 Visualizing Results

There are two reasons to subdivide element faces: first to show the parabolic shapes of the deformed elements; and second to adequately sample the source scan data for display. Each element face is the surface generated by holding one of the (r, s, t) local space variables constant and letting the others range between -1 and 1. Table 4.8 shows the numbering scheme for the elements used in the CAPS system. A set of sample points is generated on an element surface by looping over the unconstrained local space variables at equal steps. A vertex is created at each sample point the world space location of which is calculated by applying the equation 4.13 to the sample point's (r, s, t) values. These vertices are then connected into quadrilaterals for rendering. The scan direction column of table 4.8 shows whether to go from -1 to 1 or 1 to -1 for each of the unconstrained variables in order to create a quadrilateral mesh where all the polygons face out from the element. The user is able to type into a menu item to change the resolution of the element face subdivision. The number selected by the user specifies the number of steps to take in the unconstrained variables when the face is being subdivided. The user can change the resolution of the subdivision in order to trade off between image quality and rendering speed.

The element face quadrilateral meshes are downloaded into bolio for interactive viewing or animating. Both the positions and the colors of the vertices of a quadrilateral mesh can be modified via messages passed from the fem module to bolio. These values are changed when the user selects a new mapping mode from the menu. For each sample point, the fem module precalculates the data shown in table 4.9 so that it can more quickly update the element face quadrilateral meshes when the mapping mode is changed. There are four possible mapping modes for the positions of the vertices based on combinations

face number	constraint	scan direction
0	$t = 1$	$r:\ -1 \rightarrow 1,\ s:\ -1 \rightarrow 1$
1	$t = -1$	$r:\ -1 \leftarrow 1,\ s:\ -1 \rightarrow 1$
2	$s = 1$	$r:\ -1 \leftarrow 1,\ t:\ -1 \rightarrow 1$
3	$r = -1$	$s:\ -1 \leftarrow 1,\ t:\ -1 \rightarrow 1$
4	$s = -1$	$r:\ -1 \rightarrow 1,\ t:\ -1 \rightarrow 1$
5	$r = 1$	$s:\ -1 \rightarrow 1,\ t:\ -1 \rightarrow 1$

Table 4.8: Face numbering scheme for the eight to twenty variable-number-of-nodes isoparametric element.

of the options undeformed/deformed and range/no range. The undeformed/deformed option selects whether or not to add the displacement vector to the position vector. The undeformed option corresponds to the pre-operative case, and the deformed option corresponds to post-operative. When the deformed option is selected, an additional scale parameter can be specified which scales the displacement vector before it is added to the position vector. This parameter can then be used for animating the transition from pre- to post-operative state. A button on the user interface can be used to generate an interactive version of this animation.

The range/no range option selects whether or not to add the range displacement vector to the position of the vertex. The range displacement vector is calculated by mapping the undeformed world space position of the sample point into cylindrical space and looking up the actual value of r in the range data using equation 4.9. This mapping option makes it possible to show the full resolution of the scan data with the element face quadrilateral meshes.

Options for mapping the color of the element face quadrilateral meshes are source color/no source color, stress/no stress, and strain/no strain. If the source color option is selected,

parameter	type	meaning
el_no	integer	which element
rst	real 3 vector	local (r, s, t) coordinates
position	real 3 vector	world space undeformed coordinates
range_displacement	real 3 vector	world space offset to scan data
source_color	real 3 vector	(red, green, blue) from scan color
displacement	real 3 vector	solution under current loading
strain	real 6 vector	solution under current loading
stress	real 6 vector	solution under current loading

Table 4.9: Information precalculated for each sample point on the element face for faster rendering update.

the undeformed world position of the sample point is used to access the source data color array to set the color of the vertex. For a high resolution discretization of the element faces, this results in an image which shows the full resolution of the original scan data. With no source color, the quadrilateral mesh color defaults to gray.

If the stress or strain option is selected, the color for the vertex is determined by a mapping of the pressure stresses or pressure strains in the material at that sample point. The engineering strain ϵ at a point (r, s, t) is given by the following relation:

$$\epsilon(r, s, t) = B^{(m)T} U^{(m)} \tag{4.47}$$

where $U^{(m)}$ is the vector of nodal point displacements for the current element and $B^{(m)}$ is the strain-displacement matrix at the point (r, s, t) (the calculation of $B^{(m)}$ is shown in step 2.c of the algorithm given in section 4.6.1. The engineering stress τ at point (r, s, t) is given by the relation:

$$\tau = C\epsilon \tag{4.48}$$

where C is the stress-strain matrix given in equation 4.29.

Chapter 5

Results

5.1 Surgery Examples

The figures in this section show the CAPS system in operation. Figures 5.1 and 5.2 show the user interface and the simulation of a multiple Z-plasty with excision. Figures 5.3 through 5.8 illustrate the stages in the analysis process.

5.2 Reconstruction from CT Data

The figures in this section illustrate meshes reconstructed from CT data. The CT data was first processed at Brigham and Women's Hospital into a set of 3D points identified as skin

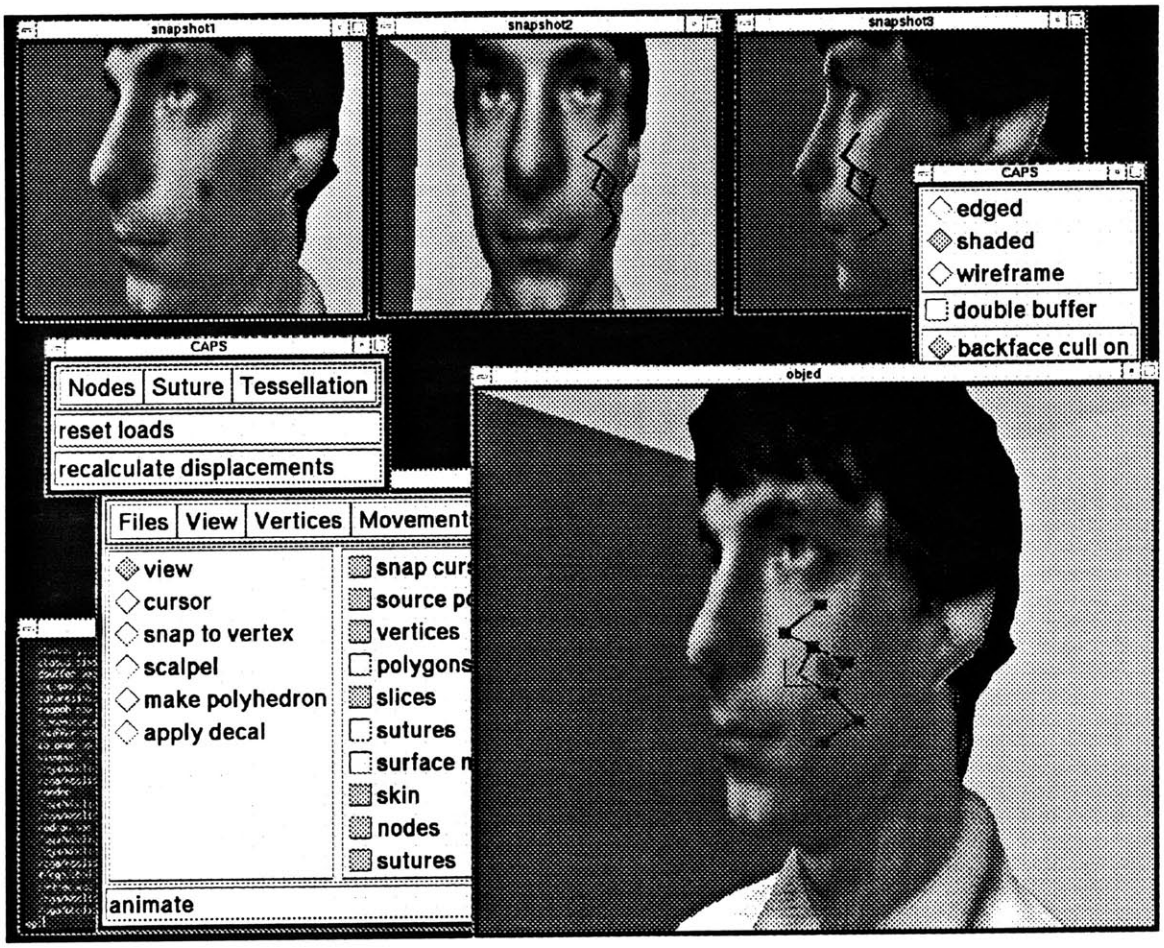


Figure 5.1: A screen image of the CAPS system in operation showing the patient model and the interactively defined surgical plan.

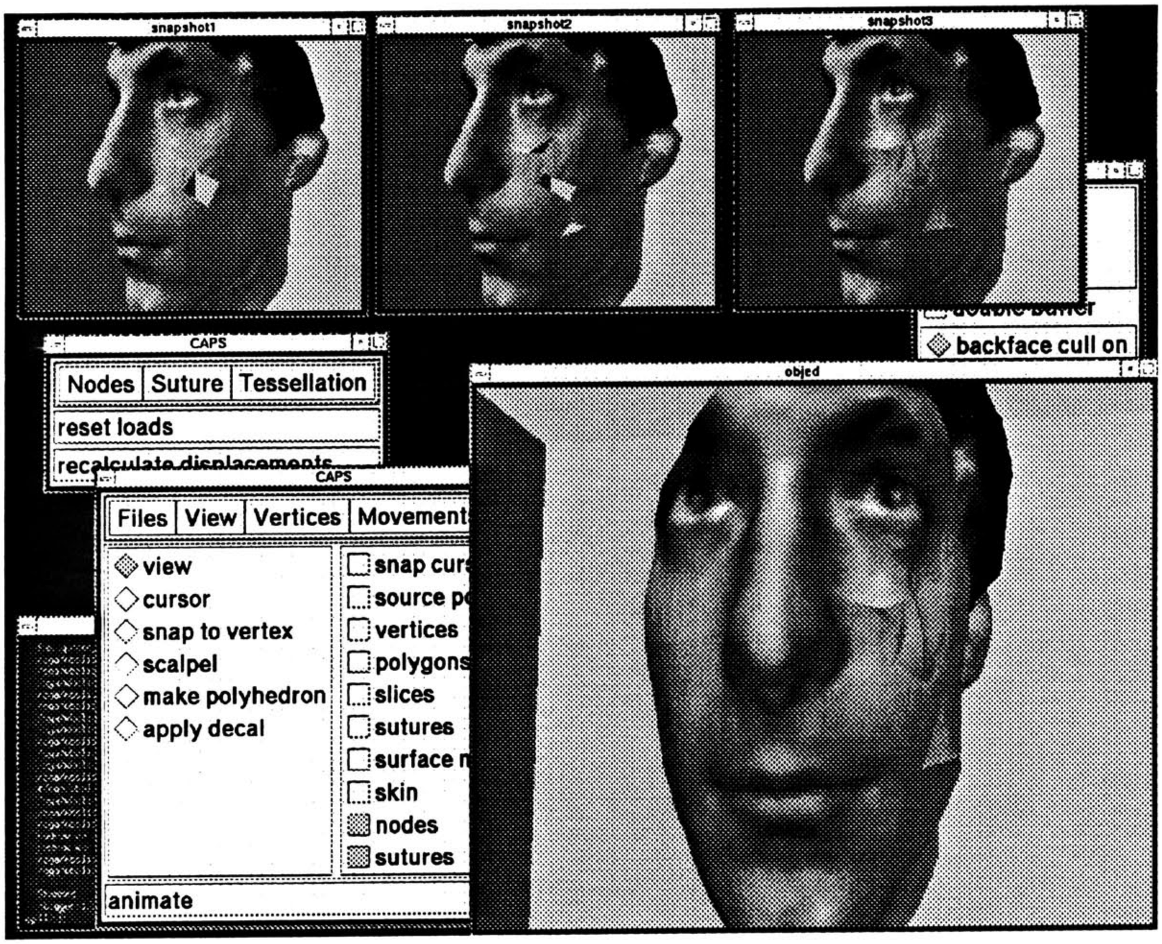


Figure 5.2: A screen image of the CAPS system in operation showing the simulated results of the operation.

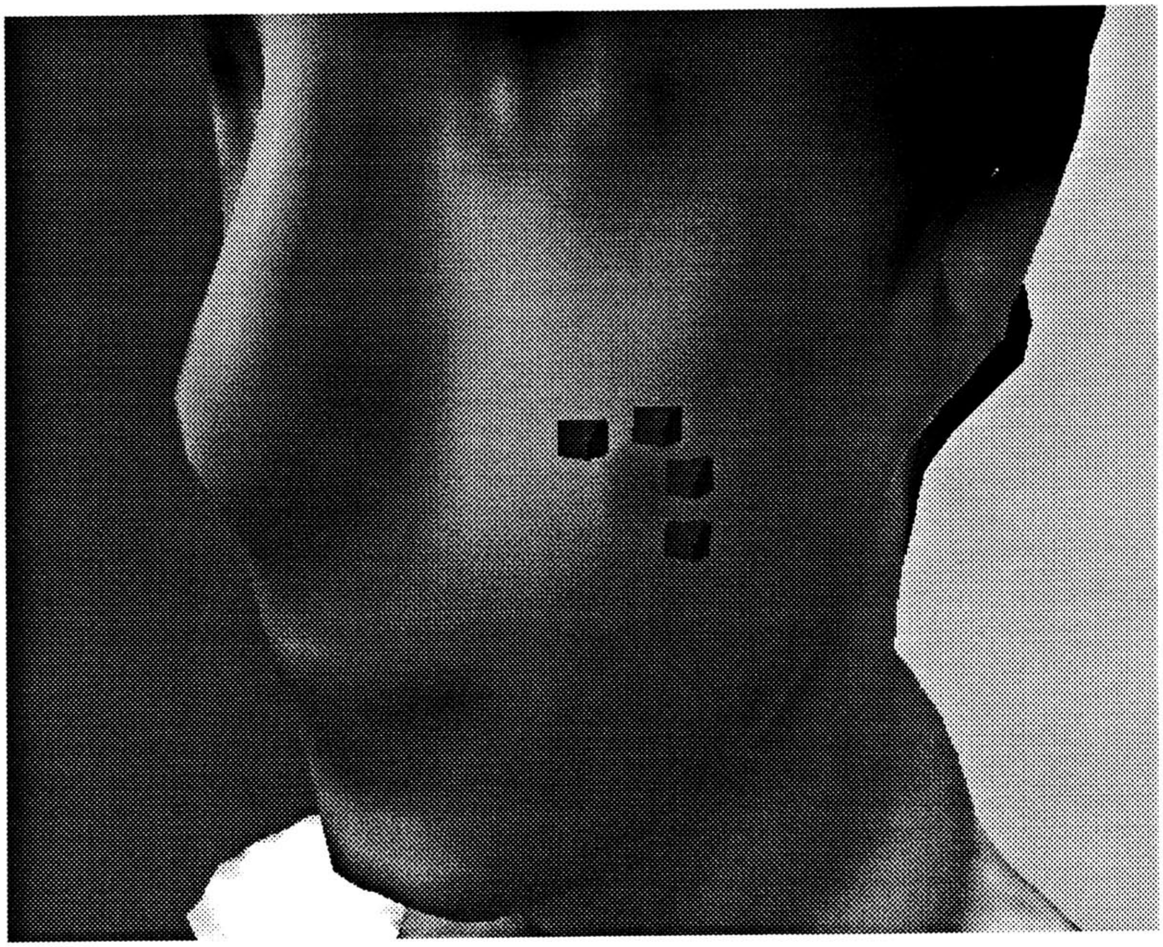


Figure 5.3: Patient model with user selected elliptical incision points.

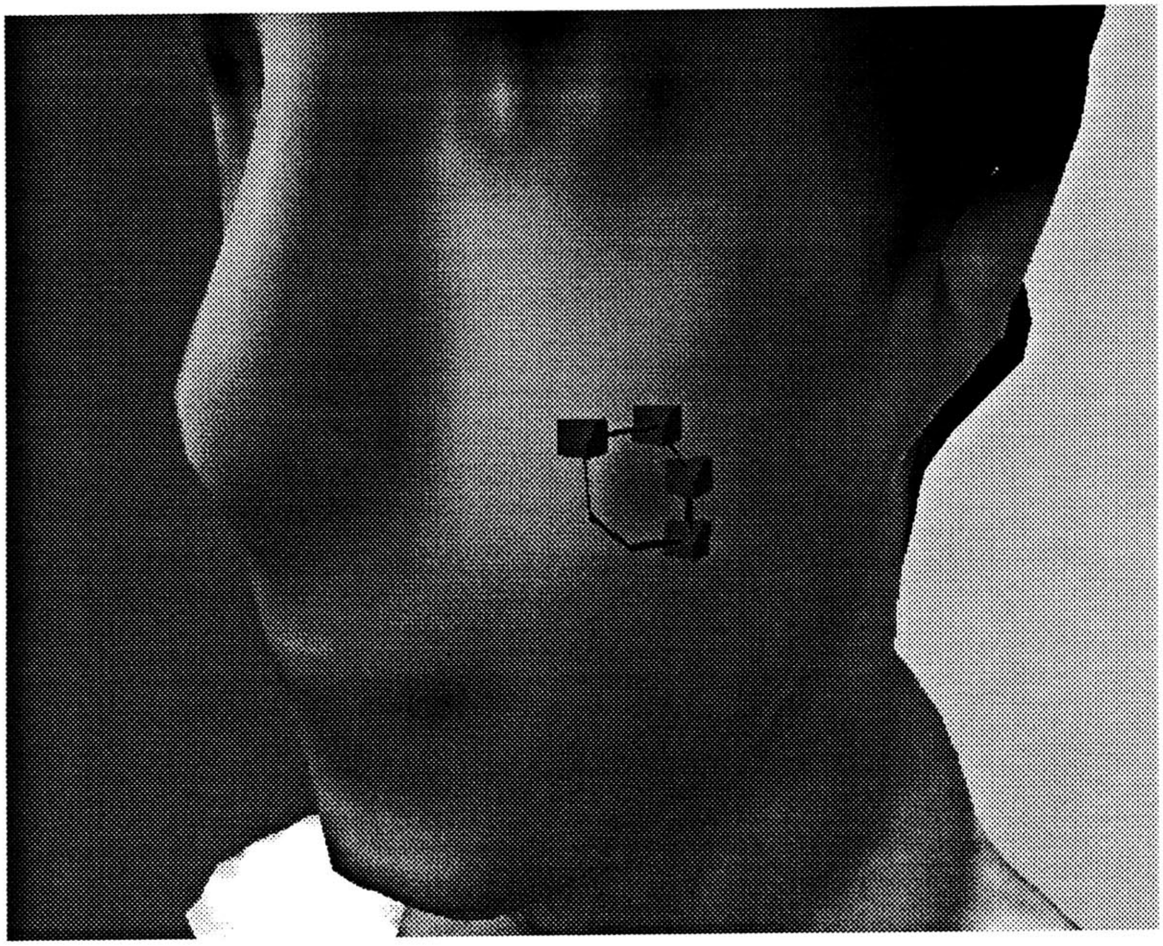


Figure 5.4: Incision boundary modified to create excision.

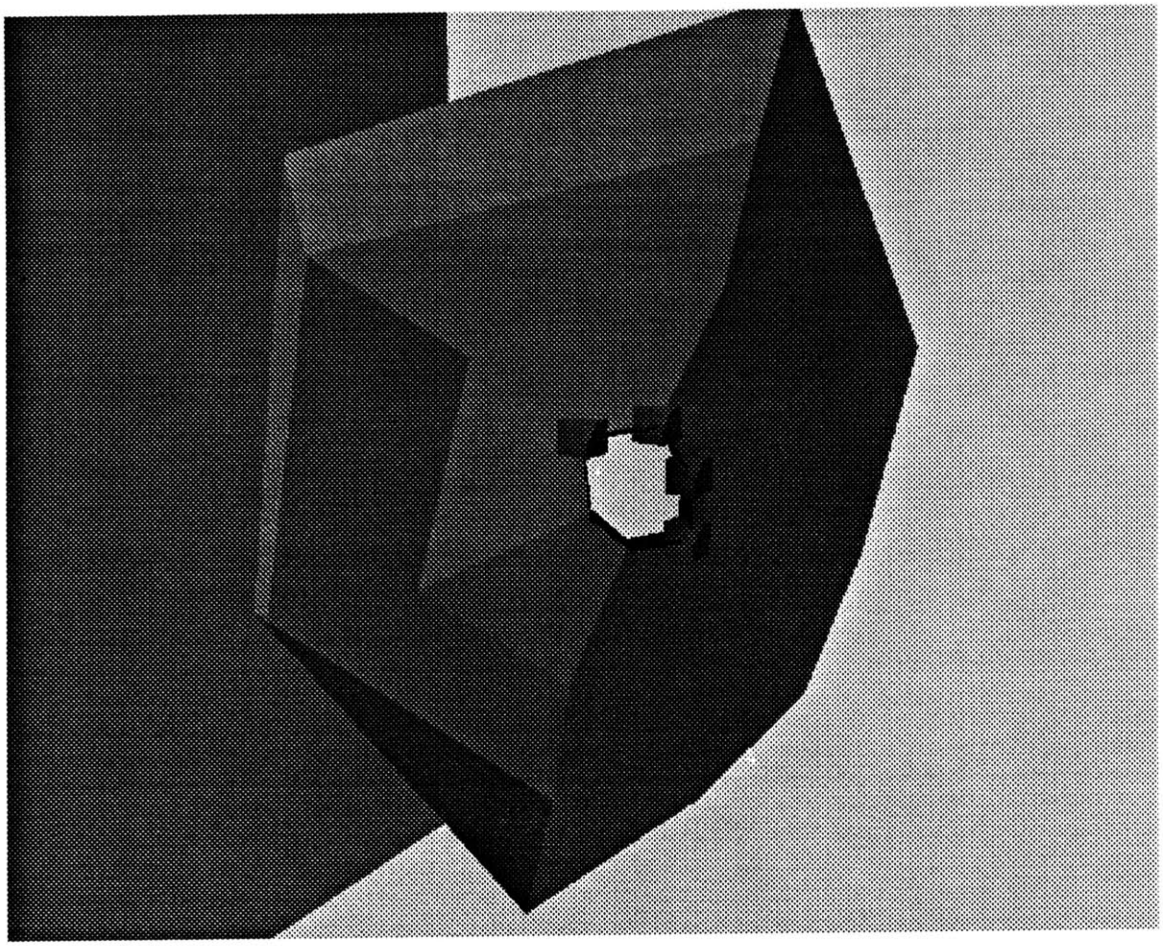


Figure 5.5: Automatically generated surface mesh for elliptical excision.

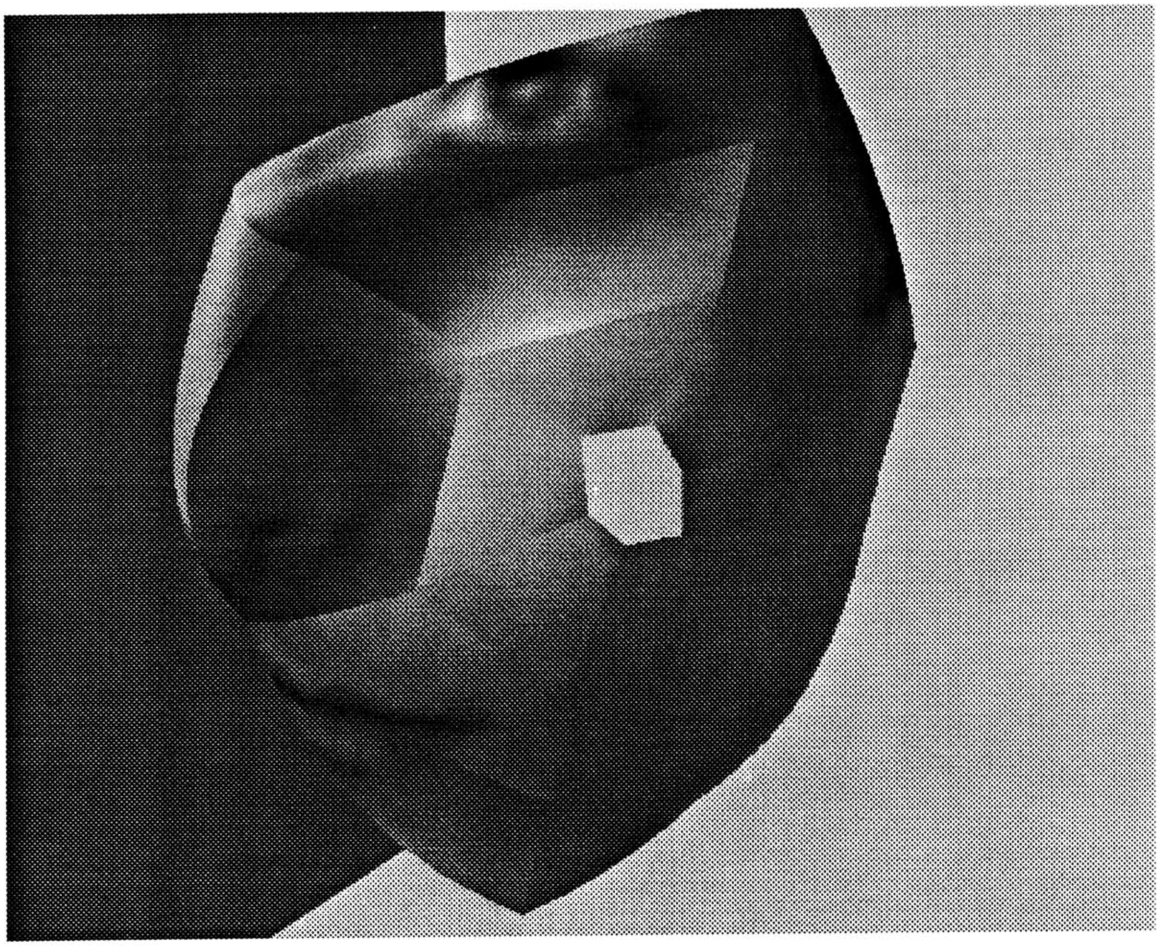


Figure 5.6: Curved surfaces of the continuum mesh elements with color data texture mapped on the surface.

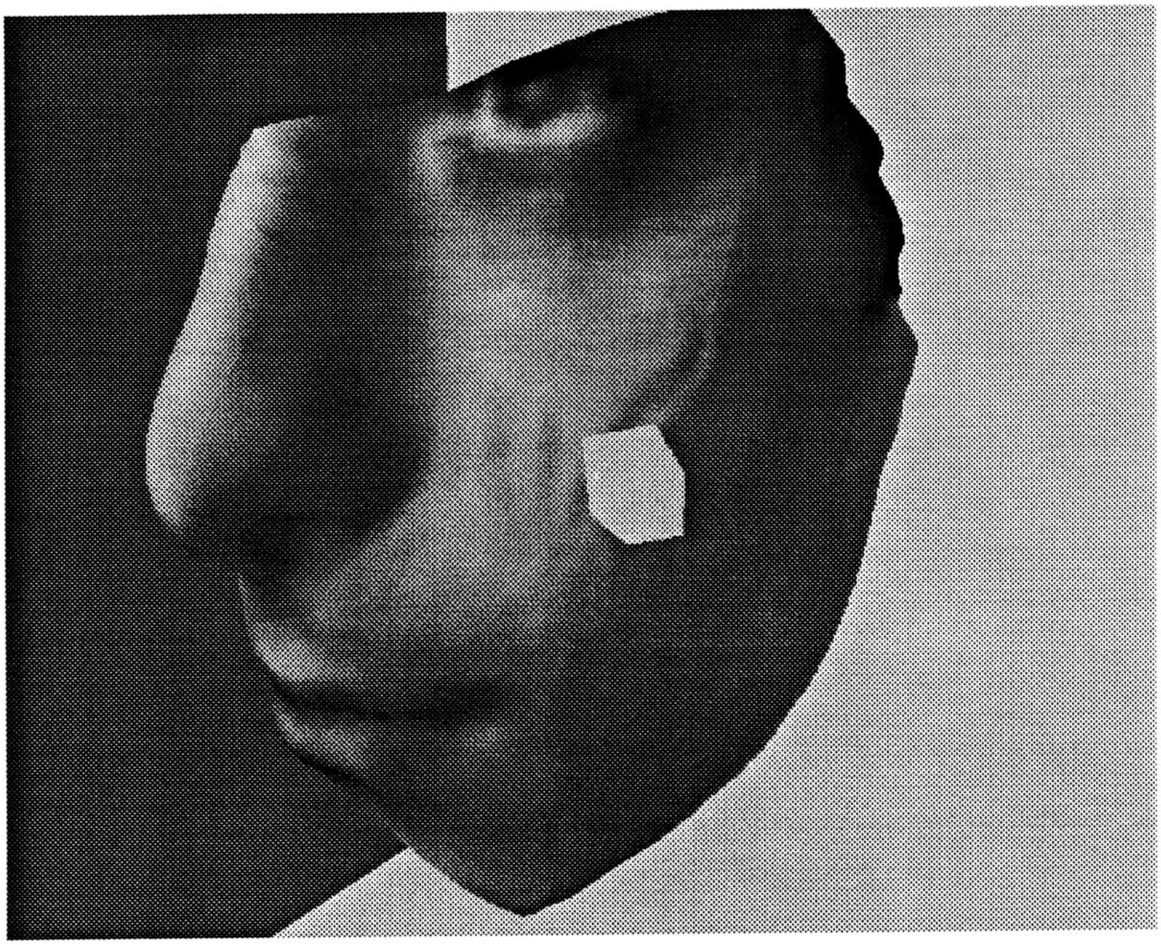


Figure 5.7: Continuum mesh elements with range displacements mapped on the surface.

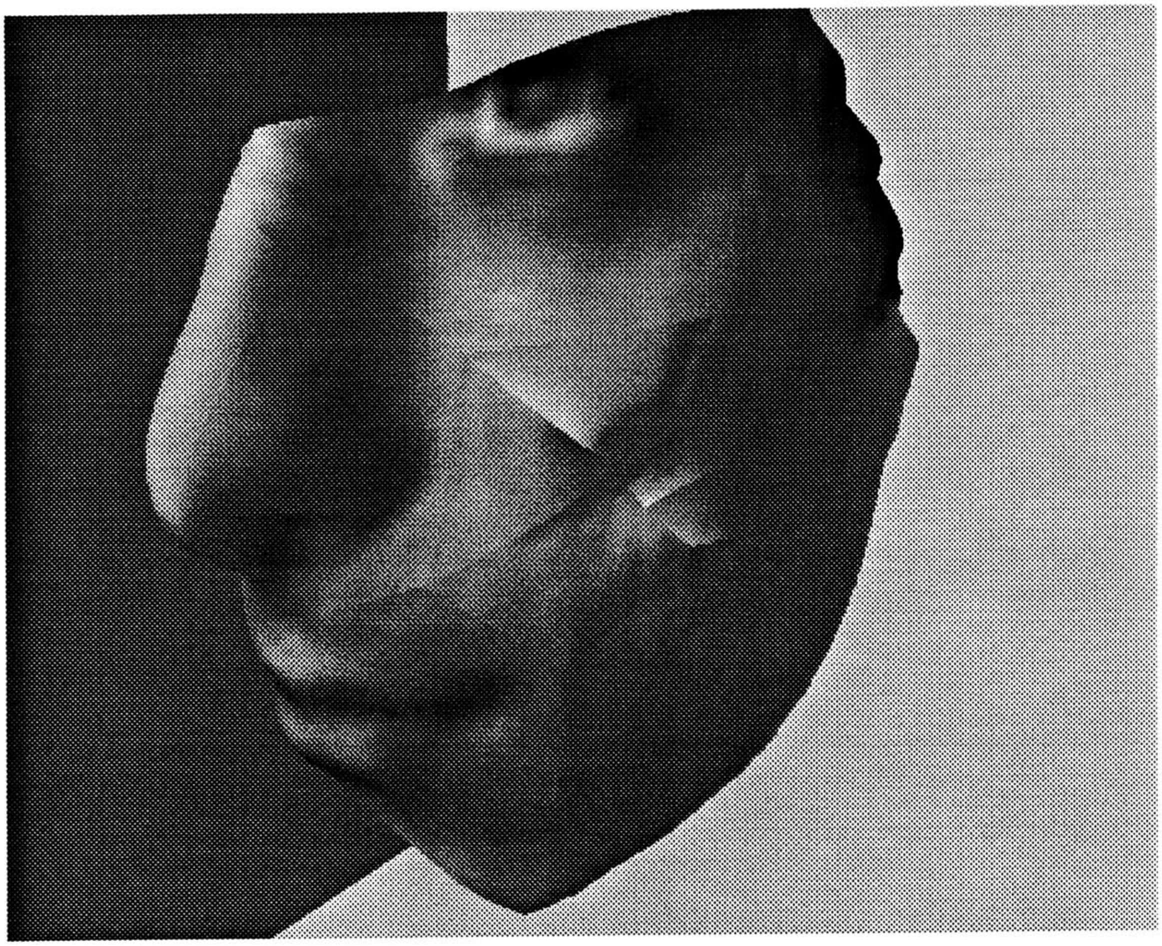


Figure 5.8: Result of finite element simulation of elliptical wound closure.

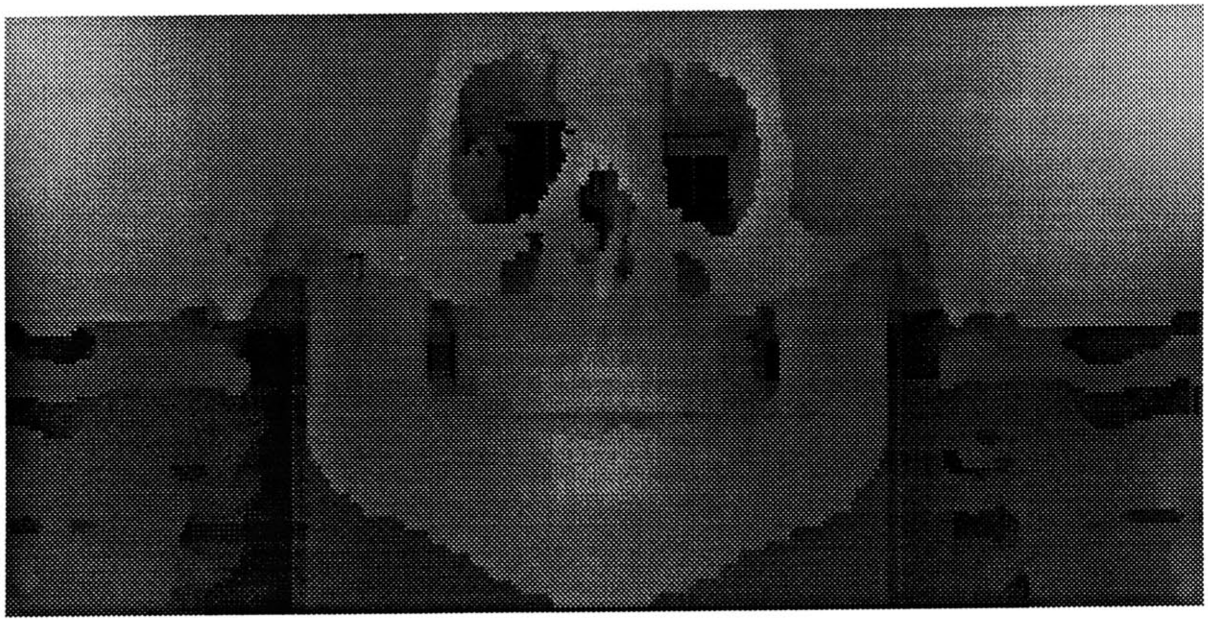


Figure 5.9: Bone surface range map derived from CT scan.

and bone. To use this data in simulated plastic surgery, the CAPS system converts them into the cylindrical range format. This is accomplished by converting each point in the list into cylindrical space and then sorting the points according to their θ and z coordinates. For each θ, z location in the range map, the maximum r value of all points which map into that location is used as the outer surface at that location. Figure 5.9 shows the range map created from a list of bone points, and figure 5.10 shows the corresponding skin surface range map. Figures 5.11 and 5.12 show polyhedra reconstructed from these range maps, and figure 5.13 shows the two polyhedra together with the skin drawn in wireframe.

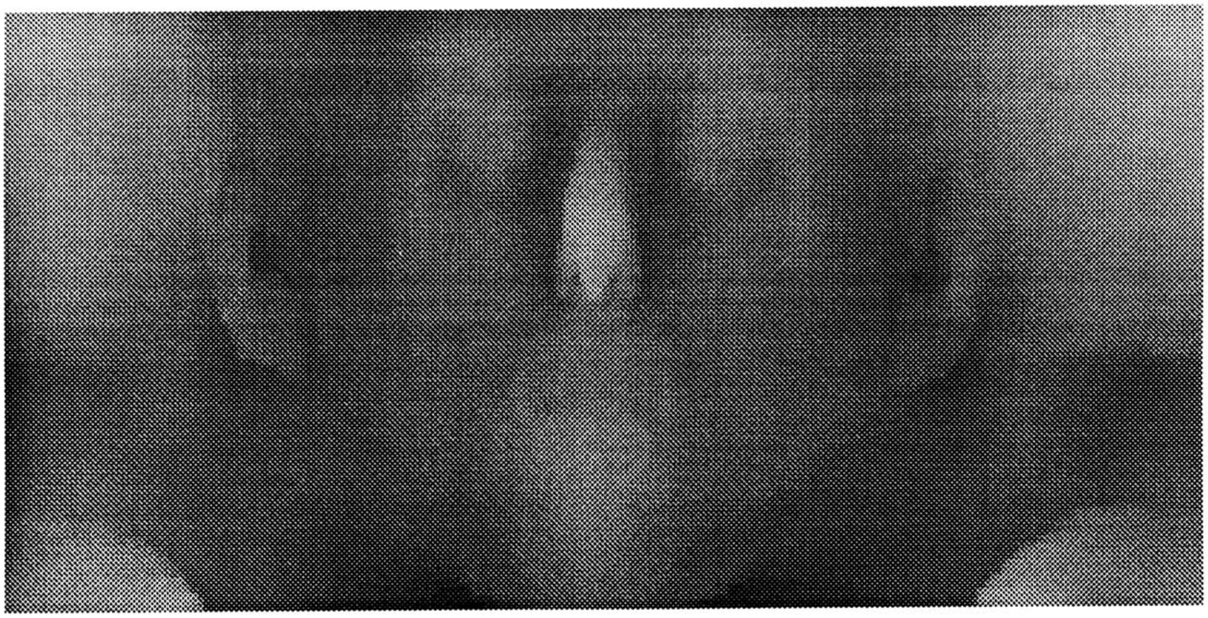


Figure 5.10: Skin surface range map derived from CT scan.

5.3 Facial Animation Using Muscle Force Functions

The techniques developed here for modeling facial skin and muscles are applicable in situations where the model is not modified via plastic surgery, i.e., they can be used to generate animations of facial expressions for other applications. The process of generating these animations is simply to generate the mesh for the patient without specifying any suture conditions and then simulate the effect of various muscle loads.

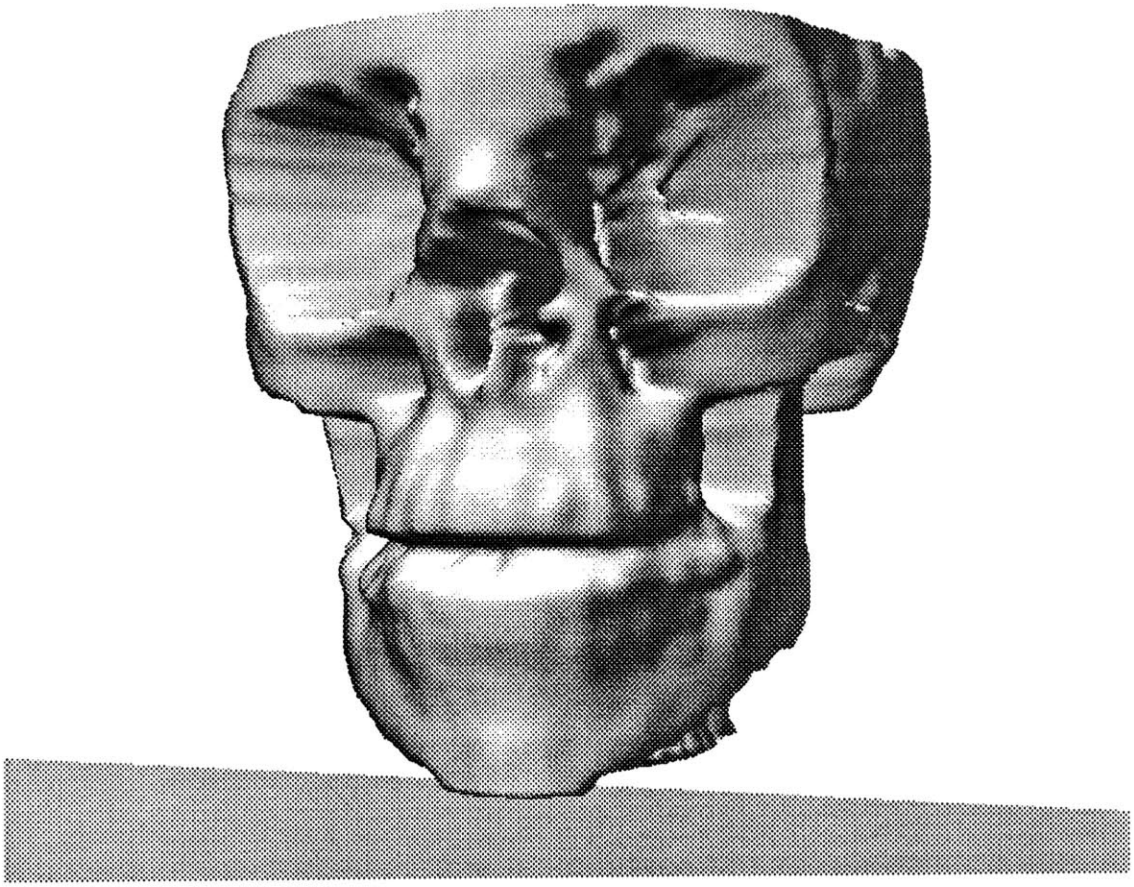


Figure 5.11: Bone surface reconstructed from CT scan.

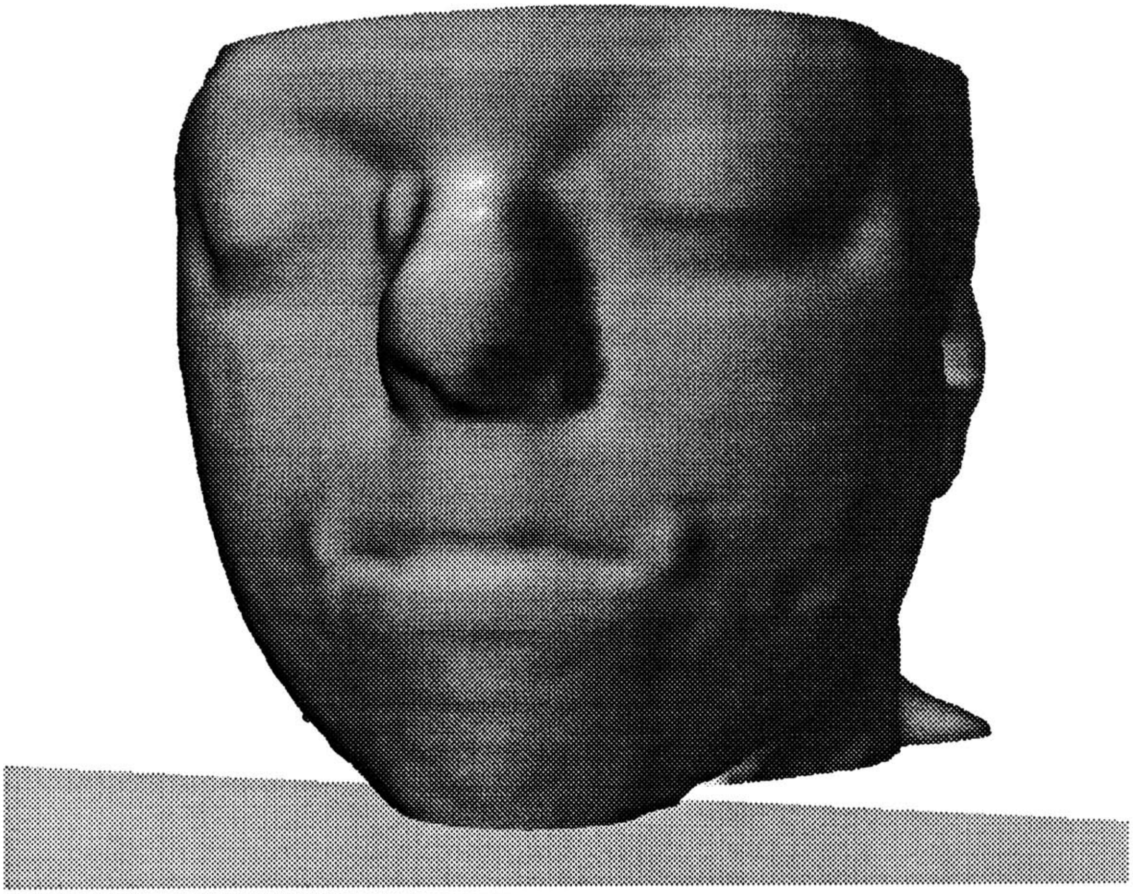


Figure 5.12: Skin surface reconstructed from CT scan.

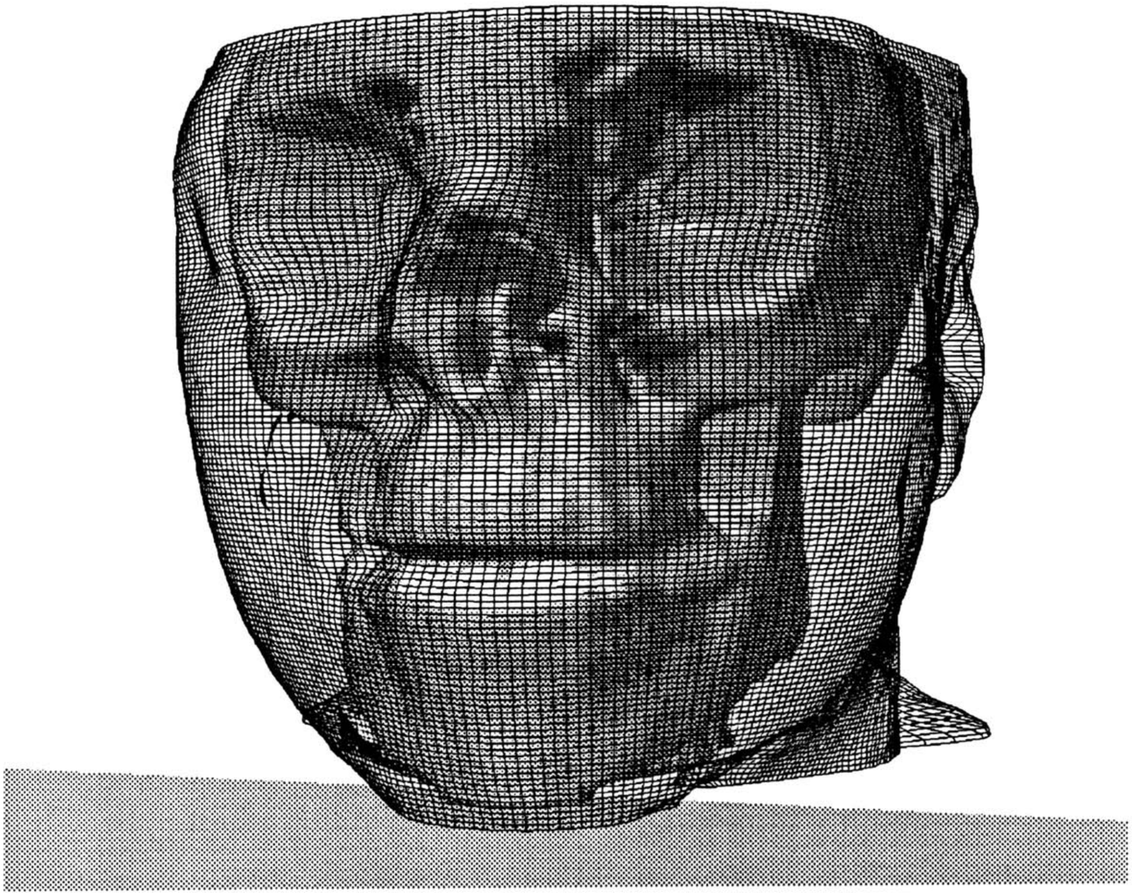


Figure 5.13: CT scan reconstruction with skin drawn as lines.

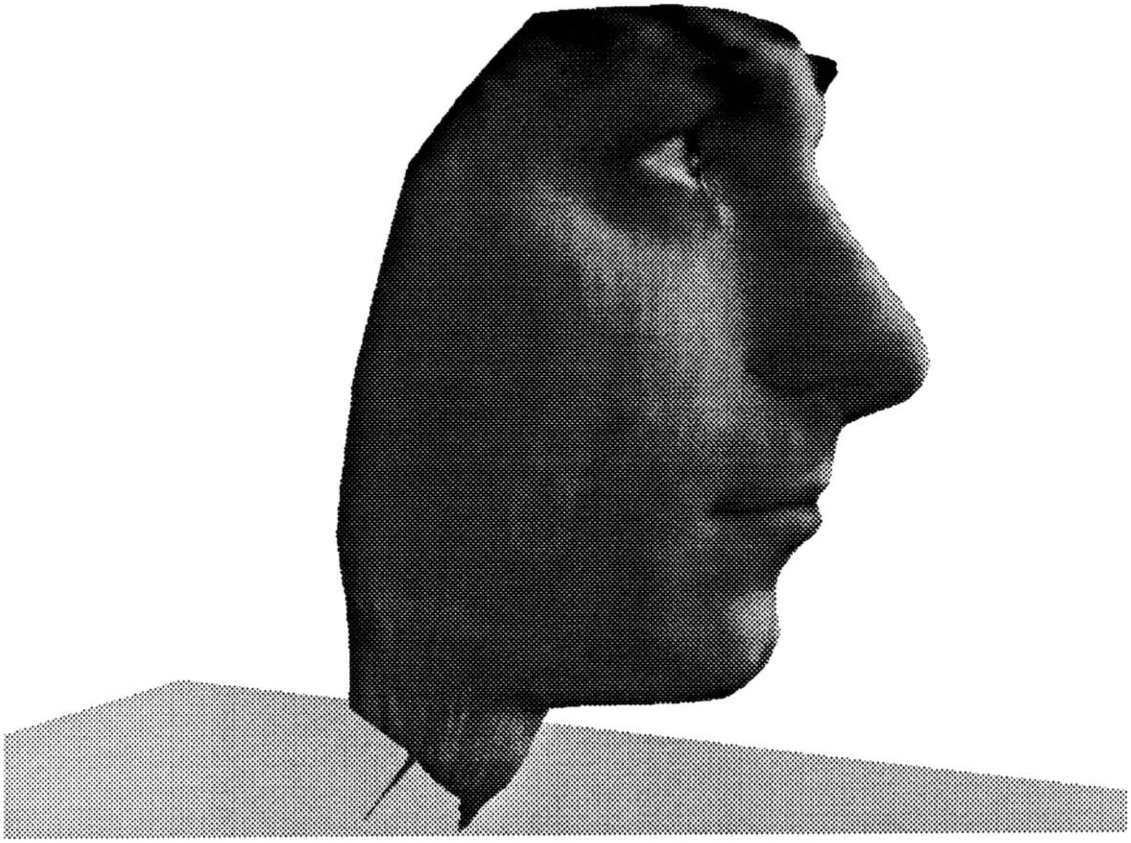


Figure 5.14: Polyhedron reconstructed from Cyberware scan with no muscles acting.

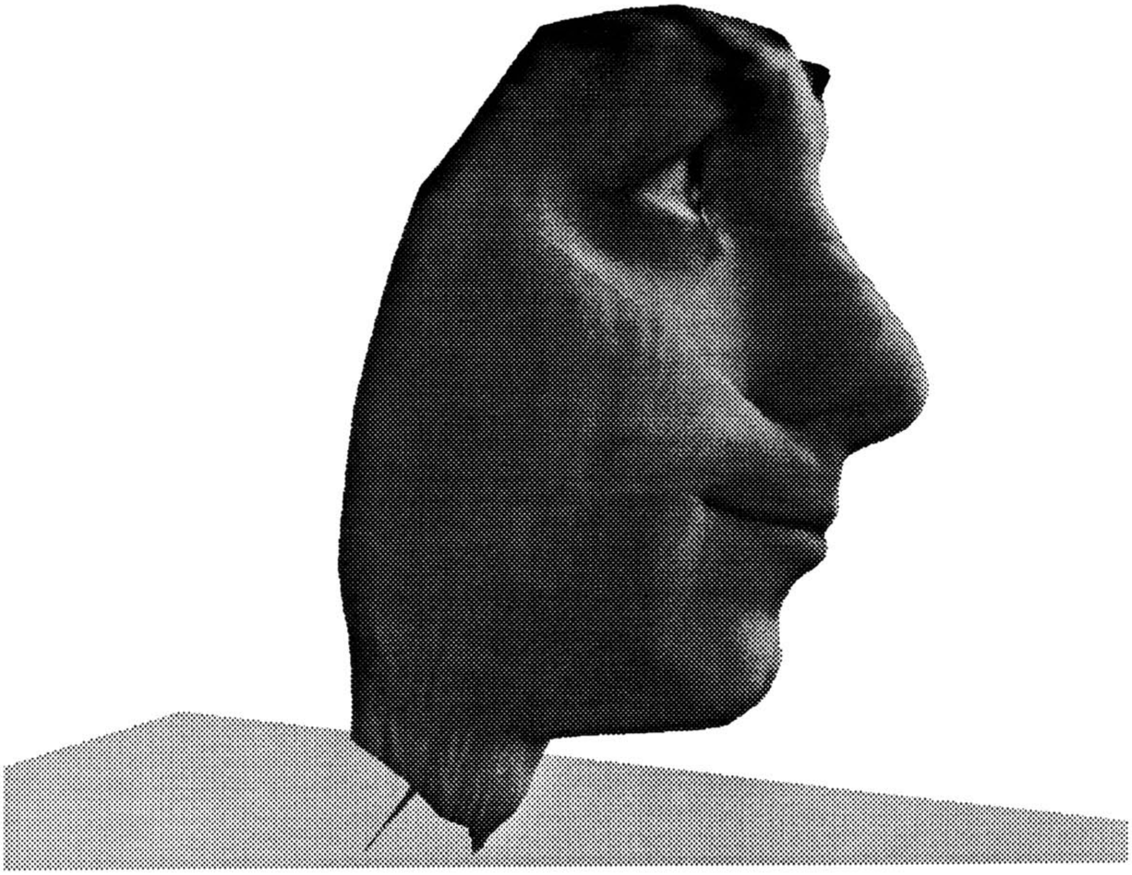


Figure 5.15: Expression generated by activating left and right levator labii muscles and the left and right temporalis muscles.

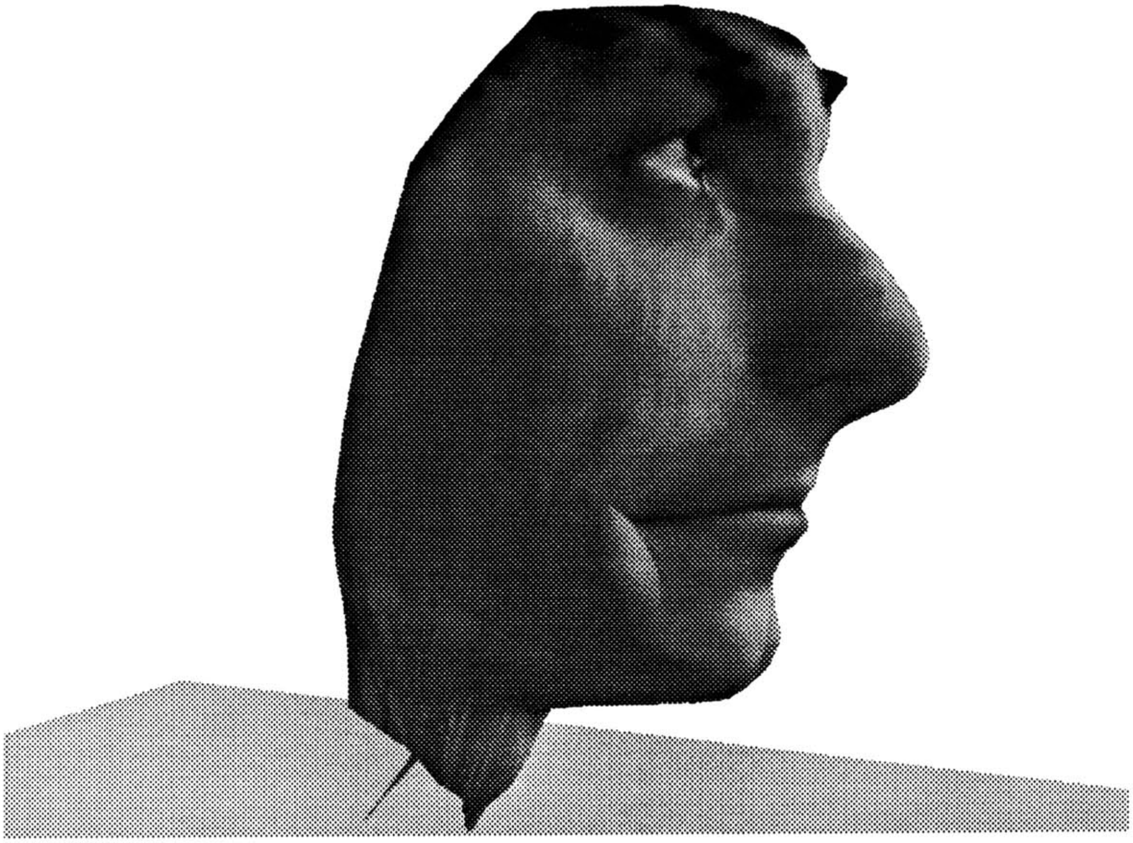


Figure 5.16: Expression generated by activation of left and right risorius muscles, the left and right alaeque nasi muscles, and the corrugator muscle.

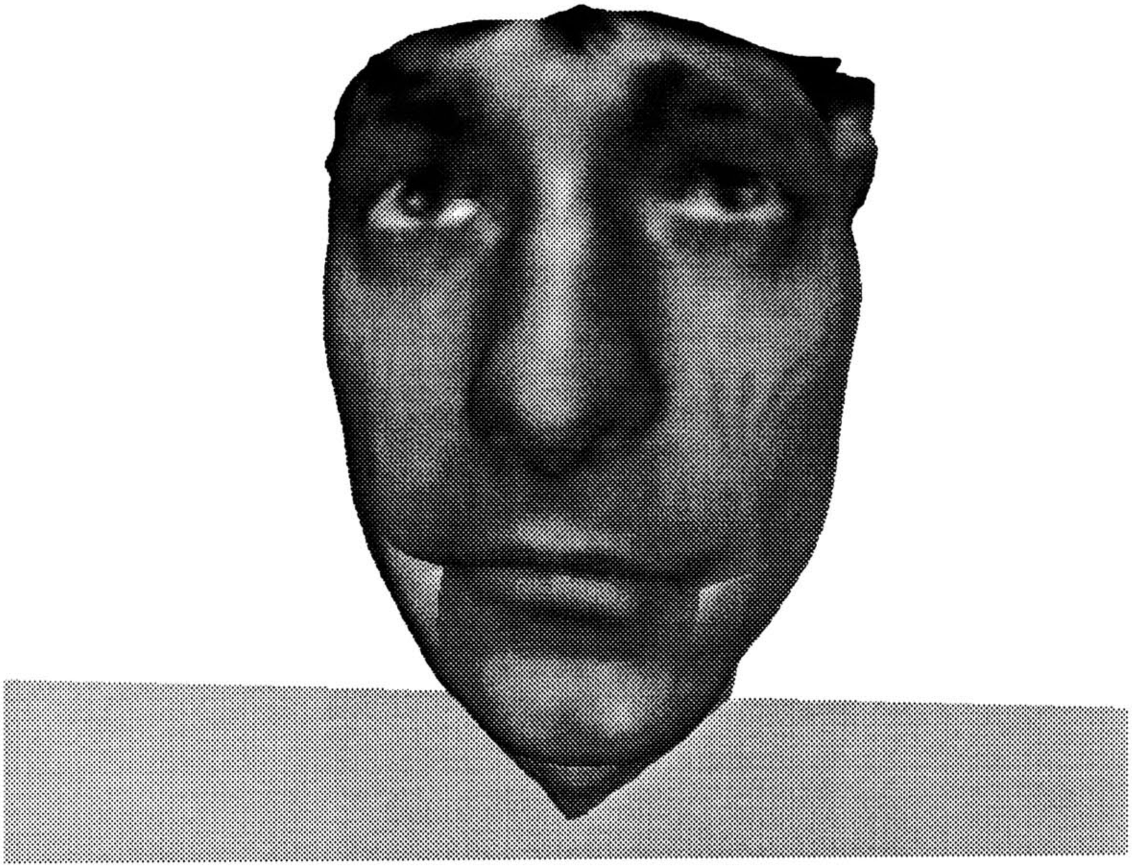


Figure 5.17: Expression generated by activating left and right depressor labii muscles.

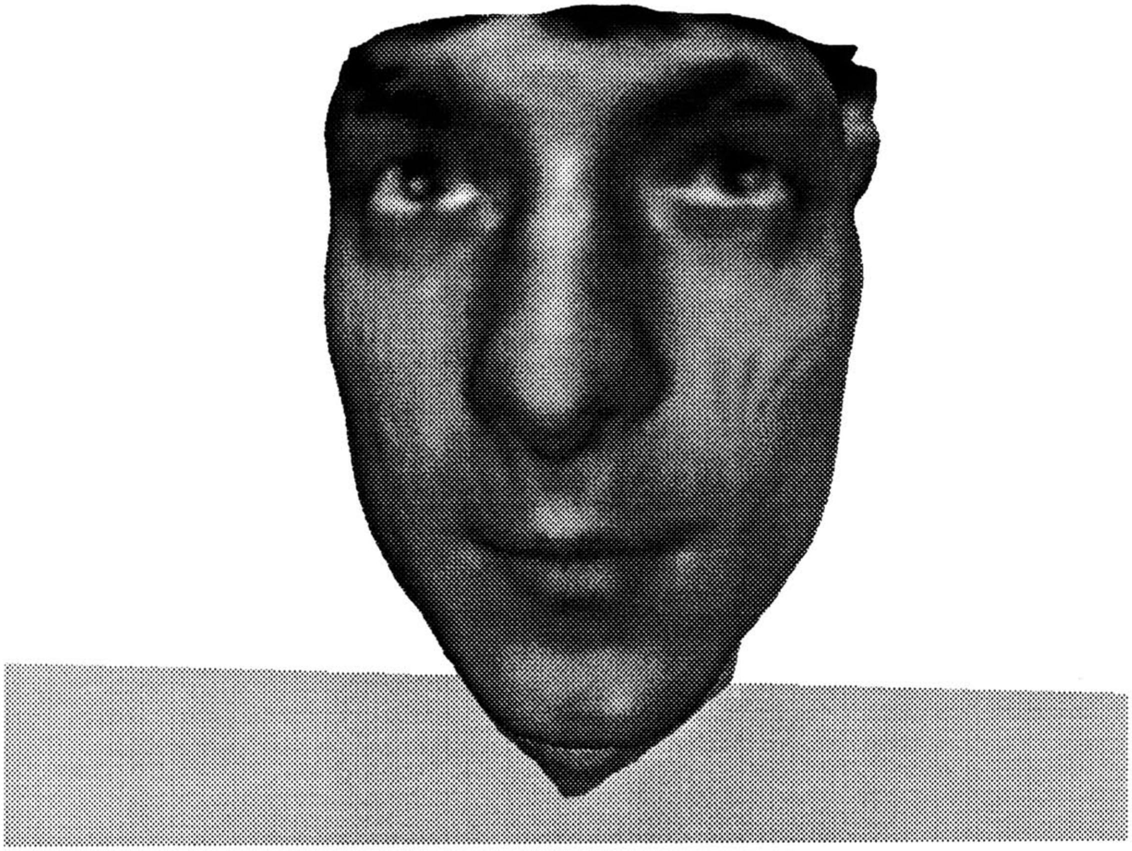


Figure 5.18: Expression generated by activation of upper and lower orbicularis oris muscles.

5.4 A Scenario for the CAPS System: II

Dr. Flanders sits down at the console of her new CAPS system to work on Molly's case. She selects Molly's MR scan from the menus of available files and watches a 3D reconstruction of Molly appear on the screen. The reconstruction captured the shape of Molly's face, but had no way of detecting that the skin on the left side of the face was in tension. Dr. Flanders invokes the object editor and uses a mouse to select the region of scar tissue and indicates that it is under pre-stress.

Having prepared the skin mesh, she goes to work on the actual procedure. She knows from past experience that a combination of Z-plasties can be used to rotate the scar line so that it lies along the nasolabial fold and reduces the excess tension on the cheek. She also knows of a V-shaped flap which can reduce the ectropion. Her problem lies in determining the angles, member lengths, and orientations for those procedures. She sketches out an initial design on the mesh surface and moments later, the results of the simulation appear on the screen. The analysis of the procedure indicates that Dr. Flanders' initial plan eliminates the ectropion, but her planned double Z-plasty was insufficient to rotate the scar. So, Dr. Flanders goes back and tries an additional Z-plasty with extra tissue to allow the scar to rotate fully.

Having satisfied her initial goals, Dr. Flanders looks more closely at the predicted contour of Molly's face. The Z-plasties have resulted in the formation of some unnatural bumps at the tips of the Z's. To see how noticeable these bumps will be, Dr. Flanders sets up an animation sequence in which a small region of nodes on the surface of Molly's skin is displaced along a path. The displacement is selected so that it is as if Dr. Flanders had placed her fingertip at the corner of Molly's mouth and pushed her cheek up about a half inch. The resulting animation shows that the bumps actually get worse when the cheek is compressed, indicating that these bumps might be unflattering when Molly smiles. To attack the bump problem, Dr. Flanders modifies the plan to include extra small excisions at the tips of the Z's so that there is less tissue bunched up in those locations. Recalculating the animation with the modified surgical plan convinces Dr. Flanders that this will solve the problem. Over the next few hours Dr. Flanders goes back and tries different combinations of values for Z-plasty parameters, looks at the influence of changing the suture locations, and looks at the response of the tissue to a variety of displacement conditions until she finds a combination that gives a pleasing surface shape and doesn't excessively stress the skin.

Armed with diagrams of the operation and a video tape of her animation sequences, Dr. Flanders meets with the rest of the surgical team. When all members are satisfied that the operation is well planned, Dr. Flanders meets with Molly and her parents. Molly understands that after the surgery she'll still have some noticeable scars on her cheek but she's excited by the prospect of a new face and a new opportunity for an improved self-image.

5.5 Evaluation of the CAPS System

An important aspect of this thesis work is the evaluation of the various techniques with respect to the problem of planning plastic surgeries. During the course of the research work, I have been in contact with practicing surgeons for advice about how best to implement various features. As the prototype system became functional, I discussed the system with surgeons in order to get feedback about how they would be able to use such a system in their work. The evaluation process really takes place at two levels: at a technical level, the issues are whether individual modeling/input/output techniques meet the goals outlined in this proposal; at the "meta" level the question is to see what future techniques might meet those goals. The ways in which the techniques described in this document are insufficient to meet the clinical needs of the surgeon are described here and my ideas for addressing those needs are discussed in the next chapter (section 6.1).

In general, the results of the clinical evaluation process have been very positive. The surgeons agreed that current planning techniques lack an adequate model of the elasticity of the skin, and that this makes it impossible for surgeons to use the planning techniques for an accurate prediction of the result of the surgery. Instead, current techniques demand that the surgeon rely on past experience to guide the choice of procedures. The surgeons noted that the CAPS system would be particularly valuable in training young surgeons, and that their early experiences with the system would be an important avenue for subsequent clinical application of the system.

Among the areas for further research, the surgeons emphasized that validating the results of the physical model will be an important step in brining the system into general use. Several of the surgeons proposed protocols by which this process could be integrated clinical process. Another important consideration for clinical application of the system is the development of an optimality measure to give surgeons direct feedback to help guide the selection of one surgical approach over another. The post-operative stress within the tissue is one such measure, although there could be a number of other important measures such as the blood supply to the tissue or the range of motion in the tissue. Further contact with surgeons will help focus the research on the important criteria for successful operations. The following evaluation sections provide an indication of where how this reseach should proceed.

5.6 Evaluation by Clinicians

Practicing surgeons were shown the prototype system and were asked to consider the following questions.

1. What is your medical training and experience?
2. Have you ever used a computer tool for surgical planning? for training?
3. How much time do you think an average surgeon spends planning a plastic operation? For example, in the Shoopman case we looked at (scar across the naso-labial fold), or in the case of a more complex operation such as a cleft lip repair? Is planning the surgery one of the more time-consuming or difficult aspects of the surgery?
4. What are your reactions to the computer interface? I.e. do you feel that the computer graphics technique of selecting points on the skin surface is close to the way you think about planning an operation, or would it be better to simulate dragging a scalpel across the skin? or drawing on the skin with a marking pen?
5. What level of accuracy in the model would be required for clinical applicability of the system? What experiments would convince you that the system could be relied on for prediction?
6. How does the system compare with your current planning techniques in terms of:

- accuracy: do you have tools now which give you accurate predictions, or do you rely on your experience?
- ease of use: do you draw detailed plans of a surgery, or use the techniques proposed by Limberg?
- clinical applicability: are there important aspects of the planning problem which are not addressed by the computer graphics model which are addressed, for example, by a plaster cast or clay model of the patient?

Responses from Plastic Surgeon Dr. Joe Rosen:

1. Practicing plastic and reconstructive surgeon since 1985.
2. Yes, I've used a Macintosh 2D paint program to show the patient what a rhinoplasty would look like. I've shown those to residents for training. I've also used 3D models made by Cemax[1] to design bone grafts.
3. Average planning time is less than an hour. The Z-plasty and the cleft lip repair are standard operations, so planning time is short. But *deciding* which standard operation to do is difficult.
4. The menu interface is good, but there are too many options that you don't seem to use. Picking points on the surface is good. A scalpel isn't needed, but a marking pen would be good.
5. Present system okay for accuracy. About 1mm needed clinically. Best validation for the model would be matching clinical cases.

[1] Cemax is a company which creates styrofoam models of bone from CT scans.

6.
- Both accurate tools and experience are used. Wires and strings give good accuracy. Also orthognathic impressions.
- No, I only drew detailed plans for my board exams. I generally just sketch things for most procedures. I don't use Limberg analysis, but I wish I could.
- The plaster cast gives more detail, but the clay model is more flexible. I use clay when I need to move a toe up to replace a thumb, because I can rotate, bend, and shape it.

Responses from Plastic Surgeons Dr. Francisco Canales and Dr. Heather Furnas:

1. Both have been practicing plastic and reconstructive surgeons since 1990.
2. Although we have used computers for the preparation of papers for publication, we have never used a computer for surgical planning.
3. Different plastic surgeons will spend different amounts of time planning operations. With the more difficult operations, especially cosmetic cases, we tend to spend several hours preparing for the operation. This may include some time spent reading and a certain amount of time drawing and planning out our incisions. In some cases, we will make life size photographs and use tracing paper to have an image with which to "play with". This is particularly helpful in cases such as rhinoplasty where it is sometimes difficult to predict the final shape of the nose. In cases such as the Shoopman case, we will probably go over that in detail only at the time of operation. This may mean that we would spend five to ten minutes drawing things out in the operating room and not necessarily prior to arriving in the operating room. In the case of a cleft lip operation, it would be extremely helpful to be able to work out a

simulated computer plan prior to the operation, especially if one could include the underlying lip musculature.

4. The graphics techniques of selecting points on the skin surface is adequate for minor operations such as Z-plasty, but for more complicated operations, it would be better to simulate dragging a scalpel across the skin.

5. The system that you have, as it stands, can easily be used for clinical work. It would be particularly helpful for first year and second year plastic surgery residents to work with as it would give them the ability of seeing the results of a particular intervention in advance. Many senior plastic surgeons would probably not be interested in a graphics computer planning tool as they have done their operations for many years without one and would be loathe to change.

   However, if young plastic surgeons began their training by using such a system, they would in all likelihood continue to use it throughout their career at least for their more complicated cases. Certainly, we as plastic surgeons spend a lot of time drawing with marking pens on operating room drapes while the patient is asleep. If one had a definite plan prior to arriving in the operating room, he would no doubt save both money and time.

6. Compared to our current planning techniques, it seems that the graphics system is at least as accurate as our system of drawing on tracing paper. It is rare that we actually use a clay model of a patient and we tend to rely on our three dimensional interpretation of a two dimensional tracing paper result. In this particular area, your system is clearly better.

Synopsis of responses from Surgical Resident Dr. Rafael Ortiz-Colberg:

1. Third year resident in general surgery with interest in plastic surgery. In addition, B.S. and M.S. in structural engineering with 6 years work experience as a structural engineer.

2. Has never used computer for surgical planning, but has used finite element programs to analyze construction plans and written programs to automate engineering design. Has used other computer simulations for medical training, for example, a text-based system for evaluating blood acid levels. Is also currently working on a breast surgery planning computer system based on geometry calculated from MR scans.

3. Planning is certainly not the most time consuming part of the surgery since the surgeon bases the plan on past experience and on examination of the patient. Maybe this is because there has been no easy to use planning technique which accounts for the elasticity of the skin.

4. Drawing on the surface with a light pen would be easier than using a mouse because it more closely approximates what the surgeon actually does when drawing a plan on the patient's skin. A virtual reality system (i.e., a computer graphics with stereo head mounted display and 3D input device), would be the best approach because it would allow drawing on the actual surface, rather than on a projected image and thus the lengths of the incisions would be accurately represented.

5. Was impressed that the CAPS system is based on a model of skin elasticity rather than merely moving rigid pieces, and felt that this would be important to plastic surgeons also. Also felt that prestress[2] in the tissue is an important effect that surgeons will expect to see modeled in the system. Also, it is important to represent different mechanical properties in different areas of the face according to how tightly

[2] Resulting in the wound opening up when cut.

connected the dermis is to the underlying tissue. Felt that to test the model, it would be good to test the output against case histories, but also to test the program against standard FEM packages. Agrees with Dr. Canalez that the system is at least as accurate as paper drawings, and that surgeons in training are most likely to use the system in training at first, and then continue using it in practice.

6. Surgeons now always base their plan on their experience. Sometimes they have to modify the plan in the middle of the operation to make it work.

   There is no current planning technique that is accurate and easy to use. Clay and plaster models are better than drawing on photographs because lengths and angles are accurate since they are measured on the 3D surface rather than on a projection. CAPS can also give these surface measurements, and is easier to work with than clay or plaster.

   To make a surgeon feel comfortable using the system the capabilities and assumptions of the model must be explicitly laid out. The surgeon could then use experience to interpret the analysis results.

   The system could be made easier by presenting the user with fewer menu choices.

Responses from oral and maxillofacial surgeon Dr. David E. Altobelli:

1. Oral and Maxillofacial Surgeon, practicing since 1988.

2. I use the computer frequently for planning maxillofacial, craniofacial, and facial cosmetic surgery. Anthropometric, cephalometric two dimensional techniques have been automated, and we are developing three dimensional approaches based on CT and MRI tomographic data, in the near future surface scan data.

3. Planning times:

   (a)
   - Small excisions; 5-10 minutes at beginning of case.
   - Large excisions; 20-40 minutes to plan rotational or free flap design.
   - Orthognathic surgery; 1-2 hours.
   - Craniofacial surgery; 1-5 hours.
   - Cleft lip; 15-30 minutes at beginning of case.

   (b) Planning is never more difficult than the procedure, if you have chosen the correct procedure. Clearly, having an adequate data base and a detailed plan should optimize the results and minimize time and/or complications.

4. Computer Interface: To individuals unfamiliar to GUI, a mouse and pull-down menus may initially seem awkward. This approach though is usually rapidly grasped and becomes comfortable. The optimal interface would be 3-D mice and displays with virtual instruments instead of mouse cursors.

   The mouse controlled cursor for outlining the incision lines is satisfactory, I don't recall though whether one could change the vantage point for positioning different segments of the incision line. It sometimes is difficult to perceive all details of the area from just one view.[3]

5.
   - For bone and soft tissue surgery, one millimeter is the needed resolution.
   - Studies will be necessary to validate and optimize the computer model. Structured light surface scanning techniques to measure tissue surface movements after manipulation will be important. Cadaver studies could be used as a first approximation.

[3] Author's note: The CAPS system does allow changing the viewpoint during the design of the incision.

- Surgeons would want to see the skin surfaces separate as a function of skin tension lines as expected with any incision.

6. - Regardless of how detailed the treatment plan, the limiting factors continue to be:

     (a) registration of the image data directly with the patient,

     (b) a method of intra-operative navigation.

   - I am not familiar with the Limberg method.
   - Clinical applicability: The fabrication of stents, templates, implants or prosthetics are easily accomplished from a physical model. The data from the surface scans or the computer models can be converted to models by CAM techniques, although the details to make this easy are not yet in place.

7. Other Suggestions:

   (a) Need to provide adequate feedback to the clinician to know that an optimized solution has been reached. This may include:

   - relationship of incision lines to skin tension lines
   - tissue stress/strain: try to minimize these parameters--this could be represented with some form of color coding scheme.
   - surface contour, distortion of adjacent anatomic structures.

   (b) Eventually build in some hypercard like interface to provide guidance to the clinician; e.g. overlay of skin tension lines, examples of favorable incisions specific to anatomic location, basic z-plasty, flap templates based on anatomy and prior experience/favorable results.

Responses from Dr. Dennis P. Orgill, Chief Resident in Plastic Surgery, Brigham and Women's Hospital. Dr. Orgill's comments were contained in a letter which is included here verbatim.

Dear Steve:

I would like to thank you for the excellent presentation which you gave at the Surgical Grand Rounds last week. This was very good for us as plastic surgeons to see what has happened in the field of media animation as well as the finite element model that you have developed.

With regard to your survey, I think we have covered many of the questions that you raised. I think that what you are developing will provide an excellent teaching tool for surgeons in training. I believe that using a the surface scanner will also allow people to follow what happens after surgical procedures in time, in addition as a research tool. This will be very important in planning surgery.

As I indicated after your presentation, I feel it will be very useful to obtain some hard data on the mechanical property of the skin, particularly the face, as these are known to vary from region to region and also with age. Also a nonlinear approach to the problem may give a more realistic solution to the various simulated procedures which you are doing.

Again, I would like to thank you very much for your participation in the Grand Round Series. We would certainly be interested in hearing updates as you continue in this very interesting field and also I would be willing to help you with regards to clinical evaluation.

Sincerely,

Dennis P. Orgill, M.D.

Chapter 6

Future Work and Conclusions

6.1 Future Work

Major improvements in all stages of this type of system can be made merely by running it on faster computers. In addition, many of the techniques could easily be expanded beyond the implementation used in the prototype system. And, of course, there are issues which remain unsolved and should be the focus of further research.

6.1.1 Interaction

In the future, the availability of faster computer graphics hardware will allow the much more complex models to be manipulated in real-time. These computers will make it possible to use interactive techniques such as 3D input devices and head-coupled stereo displays on models complex enough to be interesting and useful.

6.1.2 Analysis

A faster computer with more memory will also allow greatly improved analysis in two ways. Right now the limiting factors on the complexity of the analysis one can perform are the memory available on the machine and the time the user wishes to wait for the result. The ideal situation would be analysis that ran so fast that the user could immediately see the analysis results for each change in parameter rather than having to specify all the parameters and wait for the result. In addition, on a faster computer, analysis could be improved both by using finite element meshes with more elements and by employing a nonlinear finite element model.

The analysis used in the CAPS system results in a prediction of the post-operative state of the patient. However, it currently makes no judgment on how well a given procedure will satisfy the medical goals of the surgery. That is, it relies on the surgeon's interpretation of the tissue movements in order to determine which surgical plan will best address the patient's needs. An interesting area for further research will be an attempt to quantify the factors which determine the success of a given approach. For example, it may be possible to develop a function which gives a measure of the optimality of the procedure by taking into account the stresses developed in the tissue, the resulting mobility of the tissue, and the location of the resulting scar. Such a measure could then be presented to the surgeon in the interactive system, or could be used as part of an automated optimization procedure which selects the best combination of surgical parameters with respect to that measure.

6.1.3 Visualization

The polyhedral meshes created from the patient model can already display the full resolution of the patient scan, but they require several seconds per frame to draw. Real-time manipulation of the model would improve the ability to see the shape changes that occur in response to the planned surgery. Computer graphics hardware which displays polyhedral meshes of this size in near real-time are already becoming available.

A noticeable artifact of the element subdivision process described in section 4.7 is that the element boundaries are visible as lines of shading discontinuity. This can be seen, for example, along the cheek in figure 5.15 and around the corner of the mouth in figure 5.16. This shading discontinuity occurs due to the finite element approximation to the elastic behavior of the skin. That is, within each element, the displacement interpolation function assumes a second order variation. Since the nodes of adjacent elements are shared, the displacement function is compatible and the displacements at shared edges are the same. However, since only the nodal positions are shared at the edges, the curvature at these edges changes abruptly. While it would be possible to apply computer graphics techniques to smooth the change in curvature at the element boundaries and thus make these shading discontinuities less noticeable, a more direct approach to solving this problem would be to formulate the finite element approximation in terms of elements which ensure normal continuity at the edges. This type of element can be created by extending the definition of the nodal points to include the surface normal at that node. The interpolation functions for an element would then be extended to include the effect of changes in these normals and thus the normals would be included in the stiffness analysis.

6.2 Conclusions

Interactive computer graphics has transformed mathematical models from abstractions into realistic objects which respond naturally to manipulation. In this thesis I have tried to apply that revolution to the problem of planning plastic surgeries. I hope that the techniques described here may someday improve the lives of people faced with the need for this type of surgery. In addition, I hope that the combination of technologies and the new techniques described here will help other researchers in implementing useful applications.

Chapter 7

Acknowledgments

I dedicate this thesis to my dear wife, Carol, without whose love and support nothing would be possible for me. My parents, brothers and everyone in my extended family deserve credit for anything I've accomplished, and I am deeply grateful.

Special thanks go to my advisor, David Zeltzer, not only for what I have learned from him, but for the creation of the Computer Graphics and Animation Group and for filling the Snakepit with a group of students with unmatched skill and creativity.

I also thank the members of my thesis committee. Joe Rosen provided the vision for this thesis and has helped in an uncountable number of ways. Thank you to Robert Mann for sharing the benefit of his experience with me.

I am grateful for the time spent by clinicians in helping me evaluate the CAPS system and for their feedback about the design of the system. Discussions with David Altobelli about his direct experience with computer planning tools for surgery has been enormously helpful.

I read somewhere that the friends you make during your twenties are the ones most likely to be your friends for the rest of your life. I hope this is true, because it would be

painful to lose the close friends I've made at the Media Lab. From now on when people ask why the Media Lab has such a wide variety of research groups I will say that it is because this collection of topics attracts the finest students, faculty, and staff in the world. I couldn't possibly list everyone from the lab who has helped me out, but I thank them wholeheartedly.

Finally, I must acknowledge the special role played by my labmates in the Snakepit. David Chen and Mike McKenna have been a constant source of inspiration and brilliant advice. My thesis has also been enriched through the assistance of David Sturman, Tinsley Galyean, Margaret Minsky, Steven Drucker, Irfan Essa, Mike Halle, Scott Delp, and Peter Schroeder.

[1] Altobelli, David E. & Ron Kikinis, "Best Face Forward: New Ways to Navigate Craniofacial Surgery," ***Harvard Medical Alumni Bulletin*** 63 (Spring 1990).

[2] Altobelli, D.M.D, M.D., David, ***Oral and Maxillofacial Surgeon***, Brigham and Women's Hospital, Personal Communication.

[3] Bathe, Klaus-Jurgen, ***Finite Element Procedures in Engineering Analysis***, Prentice-Hall, Englewod Cliffs, New Jersey, 1982.

[4] Belec, Lisa, "Computer Modeling of Total Hip Replacement: Application to Joint Geometry, Stress Analysis and Bone Remodeling," Thayer School of Engineering, Dartmouth College, PhD Thesis, 1990.

[5] Benton, Stephen A., "Photographic Holography," in ***Optics in Entertainment***, SPIE, 1983.

[6] Cedars, M.D., Mike, ***Plastic Surgeon***, Berkeley, California, Personal Communication.

[7] Chae, Soo-Won, "On the Automatic Generation of Near-Optimal Meshes for Three-Dimensional Linear-Elastic Finite Element Analysis," MIT, PhD Thesis, 1988.

[8] Chae, Soo-Won & Klaus-Jurgen Bathe, "On Automatic Mesh Construction and Mesh Refinement in Finite Element Analysis," ***Computers & Structures*** 32 (1989), 911--936.

[9] Chang, Grace Hwa-Pei, "Computer-Aided-Surgery -- An Interactive System for Intertrochanteric Osteotomy," MIT, SM Thesis, 1987.

[10] Chen, David, "Pump It Up: Simulating Muscle Shape From Skeleton Kinematics A Dynamic Model of Muscle for Computer Animation," MIT, PhD Thesis Draft, 1991.

[11] Cline, Harvey E., William E. Lorensen, Sigwalt Ludke, Carl R. Crawford & Bruce C. Teeter, "Two Algorithms for Reconstruction of Surfaces from Tomographs," ***Medical Physics*** (June, 1988).

[12] Constantian, M.D., Mark B., Charles Ehrenpreis, Ph.D. & Jack H. Sheen, M.D., "The Expert Teaching System: A New Method for Learning Rhinoplasty Using Interactive Computer Graphics," ***Plastic and Reconstructive Surgery*** (February, 1987).

[13] Cromie, William J., "Computer-Aided Surgery," ***MIT Report*** XVI (November, 1988).

[14] Cutting, M.D., Court, Fred L. Bookstein, Ph.D., Barry Grayson, D.D.S., Linda Fellingham, Ph.D. & Joseph G. McCarthy, M.D., "Three-Dimensional Computer-Assisted Design of Craniofacial Surgical Procedures: Optimization and Interaction with Cephalometric and CT-Based Models," ***Plastic and Reconstructive Surgery*** (June, 1986).

[15] Delp, S., P. Loan, M. Hoy, F. Zajac, S. Fisher & J. Rosen, "An Interactive Graphics-Based Model of the Lower Extremity to Study Orthopaedic Surgical Procedures," ***IEEE Transactions on Biomedical Engineering*** 37 (August, 1990), Special issue on interaction with and visualization of biomedical data.

[16] Deng, Xiao Qi, "A Finite Element Analysis of Surgery of the Human Facial Tissues," Columbia University, PhD Thesis, 1988.

[17] Drebin, Robert A., Loren Carpenter & Pat Hanrahan, "Volume Rendering," ***Computer Graphics*** 22 (August, 1988).

[18] Farkas, L. G., ***Anthropometry of the Head and Face in Medicine***, Elsevier North Holland, New York, New York, 1981.

[19] Farkas, L. G. & I. R. Munro, ***Anthropometric Facial Proportions in Medicine***, Charles C. Thomas, Springfield, Illinois, 1987.

[20] Fisher, S. S., M. McGreevy, J. Humphries & W. Robinett, "Virtual Environment Display System," ***Interactive 3D Graphics 1986 Workshop***, Chapel Hill, North Carolina (October 23-24, 1986).

[21] Foley, James D., Andries van Dam, Steven K. Feiner & John F. Hughes, ***Computer Graphics: Principles and Practice***, Addison-Wesley, Reading, Massachusetts, 1990.

[22] Fuchs, H., J. Poulton, J. Eyles, T. Greer, J. Goldfeather, D. Ellsworth, S. Molnar, G. Turk, B. Tebbs & L. Israel, "A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories," ***Computer Graphics*** 23 (July, 1989), 79--88.

[23] Gallagher, Richard H., ed., ***Finite Elements in Biomechanics***, Wiley Interscience, 1982.

[24] Gibbons, Ann, "Surgery in Space," ***Technology Review*** (April, 1989).

[25] Goldwasser, Samuel M., R. Anthony Reynolds, Ted Bapty, David Baraff, John Summers, David A. Talton & Ed Walsh, "Physician's Workstation with Real-Time Performance," ***IEEE Computer Graphics and Applications*** 5 (December 1985), 44--57.

[26] Gordon, M. J. V., B. Strauch & S. A. Blau, "A Computer Model of Wound Approximation," Abstract of paper in progress and personal communication.

[27] Grabb M.D., William C. & James W. Smith M.D., ***Plastic Surgery***, Little, Brown and Company, Boston, 1986.

[28] Gray F.R.S., Henry, ***Anatomy, Descriptive and Surgical***, Stein and Day, New York, 1977.

[29] Hardt, D. E., "Determining Muscle Forces in the Leg During Normal Human Walking --- An Application and Evaluation of Optimization Methods," ***Transactions of the ASME*** 100 (May 1978), 72--78.

[30] Haumann, David, "Modeling the Physical Behavior of Flexible Objects," in ***SIGGRAPH '87 Course Notes on Topics in Physically Based Modeling***, August 1987.

[31] Hodges, Larry F. & David F. McAllister, "Stereo and Alternating-Pair Techniques for Display of Computer-Generated Images," ***IEEE Computer Graphics and Applications*** (September 1985).

[32] Kawabata, M.D., Hidehiko, Hideo Kawai, M.D., Kazuhiro Masada, M.D. & Keiro Ono, M.D., "Computer-Aided Analysis of Z-Plasties," ***Plastic and Reconstructive Surgery*** 83 (February 1989), 319--325.

[33] Kenedi, R. M., T. Gibson, J. H. Evans & J. C. Barbenel, "Tissue Mechanics," ***Physics in Medicine and Biology*** 20 (1795), 699--717.

[34] Kim, Won S., Stephen R. Ellis, Mitchell E. Tyler, Blake Hannaford & Lawrence W. Stark, "Quantitative Evaluation of Perspective and Stereoscopic Displays in Three-Axis Manual Tracking Tasks," ***IEEE Transactions on Systems, Man, and Cybernetics*** SMC-17 (January/February 1987).

[35] Kolar, Ph.D., John C., ***Anthropometry for Planning Craniofacial Surgery***, ASPRS/-PSEF/ASMS Annual Scientific Meeting 1990, Notes for Instructional Course #2-456.

[36] Kundert, Kenneth S., "Sparse Matrix Techniques," in ***Circuit Analysis, Simulation and Design***, Ruehli, Albert, ed., North-Holland, 1986.

[37] Kundert, Kenneth S. & Alberto Sangiovanni-Vencentelli, ***Sparse User's Guide: A Sparse Linear Equation Solver***, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, April, 1988, Version 1.3a.

[38] Lamport, Leslie, in ***A Document Preparation System*** LaTeX***Users's Guide & Reference Manual***, Addison-Wesley Publishing Company, 1986.

[39] Larrabee, Jr., M.D., Wayne F., "A Finite Element Model of Skin Deformation," ***Laryngoscope*** (July, 1987).

[40] Levesque, Sylvain, "Automatic Three-Dimensional Mesh Generation of Skeletal Structures," MIT, SM Thesis, 1989.

[41] Levoy, Mark, "A Hybrid Ray Tracer for Rendering Polygon and Volume Data," ***IEEE Computer Graphics and Applications*** 10 (March 1990).

[42] Limberg, M.D., A. A., ***The Planning of Local Plastic Operations on the Body Surface: Theory and Practice***, Government Publishing House for Medical Literature, Leningrad, U.S.S.R., 1963, D. C. Heath and Company, 1984.

[43] Mann, R. W., ***Computer-Aided Surgery***, RESNA 8th Annual Conference, Memphis, Tennessee, 1985.

[44] Marcus, Beth A. & Philip J. Curchill, "Sensing Human Hand Motions for Controlling Dexterous Robots," ***The Second Annual Space Operations Automation and Robotics Workshop, held at Wright State University*** (July 20-23, 1988), Sponsored by NASA and the USAF.

[45] McDowell, Frank, ***The Origin of Rhinoplasty***, Silvergirl Inc., Austin, Texas, 1987.

[46] McDowell, I. E., M. Bolas, S. Pieper, S. S. Fisher & J. Humphries, "Implementation and Integration of a Counterbalanced CRT-Based Stereoscopic Display for Interactive Viewpoint Control in Virtual Environment Applications," in ***Stereoscopic Displays and Applications***, SPIE, 1990.

[47] McWhorter, Shane Wm., Larry F. Hodges & Walter Rodriguez, "Evaluation of 3-D Display Techniques for Engineering Design Visualization," ***Conference Proceedings of ASEE Engineering Design Graphics Annual Mid-Year Meeting***, Tempe, Arizona (November 18-20, 1990).

[48] Merritt, J. O. & V. G. CuQlock-Knopp, "Perceptual Training with Stereoscopic Cues for Hazard Detection in Off-Road Driving," in ***Stereoscopic Displays and Applications II***, SPIE, 1991.

[49] Minkin, D.D.S., Dale, ***Orthognathic Surgeon***, Stanford University, Personal Communication.

[50] Minsky, Margaret, Ming Ouh-young, Oliver Steele, Frederick P. Brooks, Jr. & Max Behensky, "Feeling and Seeing: Issues in Force Display," ***Computer Graphics*** 24 (March, 1990).

[51] Mortenson, Michael E., ***Geometric Modeling***, John Wiley and Sons, 1985.

[52] NATO, ***Applications of Mesh Generation to Complex 3-D Configurations***, May, 1989, AGARD-CP-464.

[53] Ney, Derek R., Elliot K. Fishman, Donna Magid & Robert A. Drebin, "Volumetric Rendering of Computed Tomography Data: Principles and Techniques," ***IEEE Computer Graphics and Applications*** 10 (March 1990).

[54] Oh, Jin S. & Karl T. Ulrich, "Using Fine-Grained Parallelism Simulation for Design," May 20, 1989, Paper in preparation.

[55] Ousterhout, John K., "Tcl: An Embeddable Command Language," in ***1990 Winter USENIX Conference Proceedings***.

[56] Ousterhout, John K., ***Tcl Command Language Library***, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, January, 1990, Sprite version 1.0.

[57] Parke, Fredrick, "Parameterized Models for Facial Animation," ***Computer Graphics and Animation*** (November, 1982).

[58] Patriarcho, A. G., R. W. Mann, S. R. Simon & J. M. Mansour, "An Evaluation of the Approaches of Optimization Models in the Prediction of Muscle Forces During Human Gait," ***Journal of Biomechanics*** 14 (1981), 513--525.

[59] Pentland, Alex & John Williams, "Good Vibrations: Modal Dynamics for Graphics and Animation," ***Computer Graphics*** 23.

[60] Peters, Richard M., "Biomechanics and Surgery," in ***Biomechanics: Its Foundations and Objectives***, Fung, Y. C., N. Perrone & M. Anliker, eds., Prentice-Hall, Englewood Cliffs, New Jersey, 1972, 15.

[61] Pieper, Steven, "More Than Skin Deep: Physical Modeling of Facial Tissue," MIT, SM Thesis, 1989.

[62] Pieper, Steven, Steven Drucker & Rory O'Connor, "Three-dimensional Force Input/Output Joystick Interfacing Issues," 1990, Unpublished research.

[63] Raab, Frederick H., Ernest B. Blood, Terry O. Steiner & Herbert R. Jones, "Magnetic Position and Orientation Tracking System," ***IEEE Transactions on Aerospace and Electronic Systems*** AES-15 (September, 1979), 709--718.

[64] Rosen, M.D., Joe, ***Plastic and Reconstructive Surgeon***, Stanford University, Personal Communication.

[65] Shoor, Rita, "Targeting Treatment: New treatment planning systems may foster greater acceptance of graphics in medicine," ***Computer Graphics World*** (February 1990).

[66] Simon, B. R., ed., "International Conference on Finite Elements in Biomechanics," Tuscon, Arizona, 1980.

[67] Smith, Alvy Ray, ***The Viewing Transformation***, Computer Graphics Project, Computer Division, Lucasfilm Ltd., Marin, California, May, 1984.

[68] Sollenberger, Randy L. & Paul Milgram, "Stereoscopic Computer Graphics for Neurosurgery," in ***Designing and Using Human-Computer Interfaces and Knowledge Based Systems***, Salvendy, G. & M. J. Smith, eds., Elsevier Science Publishers, Amsterdam, 1989.

[69] St. Hilaire, Pierre, Stephen A. Benton, Mark Lucente, Mary Lou Jepsen, Joel Kollin, Hiroshi Yoshikawa & John Underkoffler, "Electronic Display System for Computational Holography," in ***Practical Holography IV***, SPIE, 1991.

[70] Sturman, David, David Zeltzer & Steve Pieper, "The Use of Constraints in the bolio System," in ***Course Notes #29, Implementing and Interacting with Realtime Microworlds***, ACM SIGGRAPH '89.

[71] Sturman, David, David Zeltzer & Steven Pieper, "Hands-On Interaction with Virtual Environments," ***Proc. UIST '89: ACM SIGGRAPH/SIGCHI Symposium on User Interface Software and Technology***, Williamsburg, VA (Nov 13-15, 1989).

[72] Takemura, Haruo, Akira Tomono & Yukio Kobayashi, "An Evaluation of 3-D Object Pointing Using a Field Sequential Stereoscopic Display," ***Graphics Interface*** (1988).

[73] Tepic, Slobodan, Thomas Macirowski & Robert W. Mann, "Simulation of Mechanical Factors in Human Hip Articular Cartilage During Walking," ***Proceedings of the 1984 Summer Computer Simulation Conference*** (1984).

[74] Thompson, David E., William L. Buford, Jr., Loyd M. Myers, David J. Giurintano & John A. Brewer, "A Hand Biomechanics Workstation," ***Computer Graphics*** 22 (August 1988).

[75] Tiede, Ulf, Karl Heinz Hoehne, Michael Bomans, Andreas Pommert, Martin Riemer & Gunnar Wiebecke, "Investigation of Medical 3D-Rendering Algorithms," ***IEEE Computer Graphics and Applications*** 10 (March 1990).

[76] Upson, Craig & Michael Keeler, "V-BUFFER: Visible Volume Rendering," ***Computer Graphics*** 22 (August, 1988).

[77] Vannier, M.D., Michael W., Jeffrey L. Marsh, M.D. & James O. Warren, M.D., "Three Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation," ***Computer Graphics*** 17 (July, 1983).

[78] Waters, Keith, "A Muscle Model for Animating Three-Dimensional Facial Expression," ***Computer Graphics*** 21.

[79] Zeltzer, D., "Towards an Integrated View of 3-D Computer Animation," ***The Visual Computer*** 1 (December 1985), 249--259.

[80] Zeltzer, David, "Task-level Graphical Simulation: Abstraction, Representation, and Control," in ***Making Them Move: Mechanics, Control, and Animation of Articulated Figures***, Badler, Norman I., Brian A. Barsky & David Zeltzer, eds., Morgan Kaufmann Publishers, Inc., San Mateo, California, 1991.

[81] Zeltzer, David, Steve Pieper & David Sturman, "An Integrated Graphical Simulation Platform," in ***Proceedings of Graphics Interface '89***, London, Ontario, June 19-23, 1989, 266--274.