Interactive Graphics for Plastic Surgery:
A Task-Level Analysis and Implementation

Steven Pieper
Thayer School of Engineering
Dartmouth College
Hanover, NH 05037

Joseph Rosen
Medical Center
Dartmouth College
Hanover, NH 05037

David Zeltzer
Computer Graphics and Animation Group
MIT Media Laboratory
Cambridge, MA 02139

ABSTRACT

We have implemented a system for Computer-Aided Plastic Surgery. Planning plastic surgery procedures is complex because the surgeon needs to stretch and reshape the patient's skin to replace missing tissue while minimizing distortion of the surrounding tissue. Traditional planning techniques rely on the surgeon's experience to select among a myriad of possible procedure designs. While mathematical techniques for predicting the outcome of surgery have been proposed in the past, these are not in widespread use by surgeons because they require the surgeon to perform manual constructions and geometric calculations. Our system makes the analysis process easier by allowing the surgeon to draw the surgical plan directly on a 3D model of the patient. An automatic mesh generator is used to convert that drawing into a well-formulated problem for finite element analysis.

Key Words

Interactivity, 3D Graphics, Computer-Aided Surgery, Plastic Surgery, Surgical Simulation.

1. INTRODUCTION

This paper describes our experience designing a Computer-Aided Plastic Surgery (CAPS) system. The system provides surgeons with a computer graphics environment in which they can explore the biomechanical implications of surgical alternatives. The CAPS system uses a combination of interactive 3D computer graphics, automatic mesh generation algorithms, physically-based modeling using the Finite Element Method, and animated visualization of the surgical result. We have implemented the system and have had it evaluated by a number of practicing plastic surgeons with very positive results.

Computerized planning represents an important development for plastic 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. Each of these operations is unique, and 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.

In our view, it is crucial that the user interface to the system not burden the physician with the implementation details of the computational model Specifically, the physician should not be required to manipulate points and polygons, or nodal points and elements of the finite element model. Our work follows a task-level analysis[33] of the goals of plastic surgery: in this system the surgeon only deals directly with the problems associated with the task --- identifying the clinical problem, selecting the surgical procedure to apply, and specifying the execution of the procedure. All other aspects of the analysis are carried out automatically.The interface to the CAPS system is designed to simulate the process of drawing on the patient's skin with a marker, as is done when the surgery is transferred to the patient in the operating room.

The remainder of this paper describes the techniques used in the implementation of the CAPS system. This is motivated by a review of related work and a brief discussion of the goals of plastic surgery and the problems faced by the clinician. The following sections describe the simulation model, and the clinician's interface to the system. We then look in detail at the mesh generation algorithms that convert the surgical plan into a well-formed problem for finite element analysis.

2. BACKGROUND

Previous work has concentrated on either building mathematical models of the soft tissue mechanics in order to analyze specific test cases, or on imaging systems that present renderings of volumetric scans of the patient. Our work is an attempt to bring these two components together with a powerful user interface. This results in a system where the simulation procedures are attached to the graphical model --- a combination which 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 without the assistance of a computer graphics tool the surgeon has neither the time nor the training to formulate a specific surgical case at the level of detail required for analysis.

Mechanical Analysis of Plastic Surgery

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[11]. Kawabata and his coworkers describe their techniques for analysis of surgical procedures but report no method for automatically generating a mesh for a particular plan[16]. 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[18]. 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 number of nodes. The user interface and mesh generation techniques described in this paper begin to address these three-dimensional problems.

Computer Graphics Models of Skin

Waters describes a system based on the for simulating the expressive action of facial muscles through a combination of pre-defined action units[30]. Waters and Terzopoulos subsequently extended this technique to include physically-based dynamics of the skin in response to the muscle action[31]. However, their system could not be used directly for plastic surgery simulation because it does not support cutting and suturing. In addition, their physical model is based on the mass-and-spring lattice approach, which we feel is more difficult to control and less accurate than the finite element method.

Volumetric Approaches

Previous computer graphics work has emphasized special purpose rendering algorithms for visualization of data obtained from volumetric scans of the patient[22;19;9], or geometric methods for extracting and repositioning pieces of the volume data[7;28]. Our approach differs since the CAPS system integrates a biomechanical simulation with a graphic presentation.

Interactive Computer Graphics for Surgical Simulation

The terms *surgical simulation*[24] and *Computer-Aided Surgery*[21;5] have both been used to refer to the combination of physically based modeling of the human body and interactive computer graphics applied to planning and analysis of surgical procedures. In an example of this approach, Delp *et al.* have created a system for simulating tendon transfer operations on the lower extremity[10]. This system includes a geometric model of the major bones of the hip and leg, a kinematic model of six joints, and a mechanical model of 43 muscle-tendon actuator units. A 3D graphics interface can be used to select and move tendon attachment points. Thompson *et al.* have developed a similar system for hand surgery[27]. Our work on the CAPS system is most similar in spirit to, and was inspired by the work of these groups.

3. GOALS OF PLASTIC SURGERY

> *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.* 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.[20]

Applications of plastic surgery include repairing lesions caused by disease, replacing skin lost to burns or amputations, rebuilding features misshapen by injury or birth defects, and removing excess tissue to reduce the visual effects of aging[13]. This is accomplished through the precise application of surgical techniques including excision (removal) of tissue, direct closure of a wound site, and a variety of flap transposition and rearrangement surgeries. Each of these results in a redistribution of the available tissue and requires the application of plastic surgery principles to produce the optimum contour.

An example plastic surgery (simulated on the CAPS system) is shown in figures 6 and 7. This procedure combines excision of a tumor with two flap transpositions. The flap transpositions have the effect of using tissue from the area surrounding the excision to relieve the stress caused by covering the wound. The resultant effects on the surrounding tissue contour can be seen. This includes distortions, redistribution, and standing cones (dog ears) at the point of rotation of the flaps. The CAPS system can be used to compare various flap transposition and excision options, and provides an environment that allows the surgeon to iteratively approach the planning problem.

4. THE PATIENT MODEL

The model of the patient used in the CAPS system is a combination of patient specific geometric data and a generic mechanical model of the soft tissue.

Sources of Patient Geometric Data

The patient specific geometry we have used to date is derived from either a Cyberware surface scan of the patient[8] or from a CT scan. The Cencit scanner system is also a promising technology for use in this application[29]. The mesh generation algorithms make use of a cylindrically-mapped range image of the type produced by the Cyberware and Cencit scanners. In order to create a solid model of the skin, our current system assumes a constant soft tissue thickness when working with this type of data. Full volumetric scans (CT or MR scans) of the patient provide enough information to create a solid model with the appropriate variation in soft tissue thickness. We have experimented with some techniques for building models directly from volumetric scans[25], however, we feel that the surface scanners will be more appropriate for use in plastic surgery because of the time, expense, and radiation hazards associated with volumetric scanners. In the future we will be working on techniques for creating a generic map of facial soft tissue thickness in order to generate more accurate solid models from surface scan data.

Model of Soft Tissue Biomechanics

The finite element method is a well established technique for biomechanical analyses[12] and provides a basis for detailed modeling of skin nonlinearities[11]. Finite element methods can also be used to model the shape changes and force generating properties of other parts of the body, such as the muscles[6]. Although we use a relatively simple linear solution technique in the CAPS system, the user interface and mesh generation techniques described below can be used directly with a nonlinear finite element back end. The finite element module of the CAPS system uses the displacement-based formulation to solve the elasticity equilibrium equations. The implementation closely follows the procedure described in Bathe[1]. Readers are referred to Bathe's excellent text for further details on the implementation of finite element codes.

Visualization of the Finite Element Model

The two components of the patient model, the scan of the patient and the finite element mesh, exist different resolutions. A typical Cyberware patient scan contains 512x256 range and color samples, while the finite element meshes we can easily simulate contain only 50 elements, with each element covering approximately a square inch of skin. In order to display the full resolution of the original scan data both before and after the finite element solution (corresponding to pre- and post-operative conditions), we use the following texture and displacement mapping technique. First, we subdivide the outer face of each element into micropolygons (the outer face being the one which lies on the skin surface). The position of each micropolygon vertex is transformed back into cylin-

---

* In contrast to skin grafting operations.

drical coordinate space, and the θ and z coordinates are used to sample the Cyberware range and color data (a bilinear interpolation is used to sample between pixels). The color value is stored as the vertex color of the micropolygon vertex. The sampled range point is transformed back into cartesian space and used as the position of the micropolygon vertex. The user can select the number of micropolygons created for each element and thus can visualize the full resolution of the Cyberware data. We maintain a data structure for each micropolygon vertex in which we store the vector from the point on the surface of the element to the corresponding position on the range data.

This vector is then used to display the full resolution post-operative model. The output of the finite element solution is a set of displacements for each nodal point in the finite element mesh. These nodal displacements are interpolated through the element to define a displacement vector at each point in the element. Thus, for each micropolygon vertex, there is a displacement vector. By adding the finite element displacement vector to the range data displacement vector, we can generate post-operative images using the full resolution of the original scan. Images generated using this method are shown in figure 7.

5. SPECIFYING THE PLAN

The heart of the interactive system is the user interface which allows the surgeon to input the parameters of the surgical procedure. For this task, we selected an interface based on a combination of 2D and 3D computer graphics techniques using the X Window System with the Motif toolkit, and on a set of 3D interaction tools built on top of the Starbase graphics library from Hewlett Packard. The CAPS system is built on top of the bolio simulation system[32]. The clinician is presented with an X Window System screen containing a menu bar and buttons, and a 3D graphics window showing a rendered image of the geometric model of the patient. The user controls the 3D view of the patient model and modifies other rendering parameters using the mouse. The user interface also allows the surgeon to switch between the pre- and post-operative patient geometry, or to animate the transition between them.

Mouse actions are used to select points on the rendered image of the patient. These points are used to define the incision lines on the skin surface and the tissue to be excised. The system converts this into a data structure for subsequent use by the mesh generator.

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., a mouse click on the window should pick a point on the patient model which appears directly beneath the mouse location. For use in their 3D object painting system, Hanrahan and Haeberli describe a technique for hardware-assisted calculation of this location that makes use of an object ID buffer[15].

Since our graphics hardware did not support this feature, we implemented this operation with ray tracing as follows. A ray is cast from the view point to the selected point on the view plane and is intersected with a 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 vertices with a polygon. This operation requires checking the ray against each of the polygons in the polyhedral reconstruction. To reduce the number of polygons, a filtered version of the source data is used. The operation could be made more efficient with octree sorting of the polygons or other ray tracing optimizations. It turned out that we did not need to explore this since the point is picked on a 2D image, and feedback can be given instantly when the button is pressed; the system can then be calculating the 3D intersection in the background while the user is selecting the next point.

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. Again, we chose a ray tracing approach to select the nearest point to the ray from the view point through the picked point on the view plane.

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 draw a hole by entering 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 2 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. The incision line can be modified by picking one of the points and moving it.

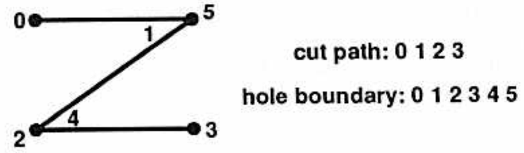
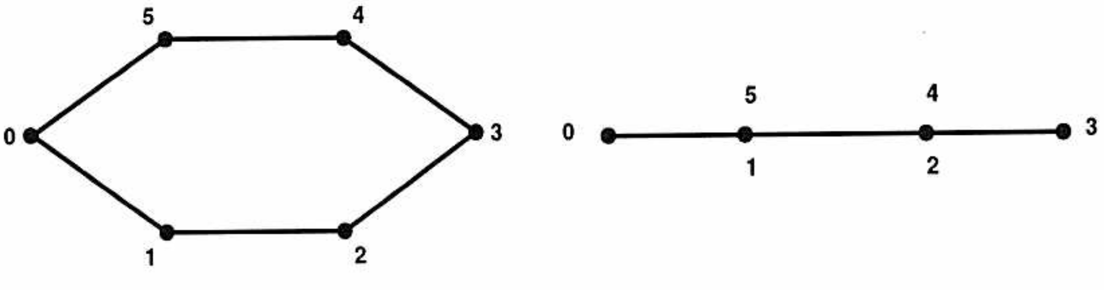


Figure 2. This figure shows the relationship between the cutting path entered by the surgeon and the boundary of the incision hole. The surgeon selects the points 0, 1, 2, and 3 to define a simple Z-plasty incision. The system adds points 4 and 5 coincident with points 2 and 1. The boundary of the hole is then stored as the ordered list 0, 1, 2, 3, 4, 5. Note that no tissue was removed in the incision shown. Tissue could be removed by interactively picking and moving the points 1, 2, 4, or 5 in order to enclose the tissue to remove within the hole boundary.

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. A simple point picking algorithm cannot be used for this picking operation because the two points on either side of the hole are coincident. A modified algorithm could be devised to distinguish between coincident points by determining on which side of

Figure 1. This figure shows the node numbering and pattern for an elliptical excision, both before and after wound closure. The surgeon originally enters the points 0, 1, 2, and 3. The system then adds points 4 and 5, initially coincident with 2 and 3. The surgeon then moves points 4 and 5 to enclose the excision region.



the incision line the user picks. In our current prototype a menu selection is used to indicate the point to be moved.

Closing the Hole: Suturing

Suturing refers to the sewing together of edges of the incision. 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 pairs of nodes must constrained together in order to suture the entire wound. Selecting each pair of nodes by hand would be unnecessarily tedious. Instead, the continuum mesh generator automatically creates a list of nodes to be sutured from a description of which edges of the hole border are to be brought together. Figure 1 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 2 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 (e.g. elliptical excision or Z-plasty). 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 menu-based 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 user interface configuration file. While this is not a very difficult task, a more flexible solution would be to allow the user to define the suture relationships by selecting pairs of wound edges. The system could differentiate between coincident edges by determining which side of the incision line the user picked. Picking edges in the proper sequence would then define the suture relationships for the surgical procedure. These suture relationships could then be added to the menu for use in future analyses.

6. MESH GENERATION

The surgical plan is entered in the CAPS system using a graphical interface which corresponds to the way the surgeon draws on the patient's skin in the operating room. An important part of this interface is the mesh generator, which creates a well-formed finite element mesh directly from the surgical plan and the original scan of the patient geometry.

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 a normalized cylindrical space ignoring the r (radial) coordinate. After the surface mesh is generated, the mesh is snapped back to the skin surface by looking up the r coordinate in the Cyberware range data.

The continuum meshing portion of the algorithm refers to the process of creating a continuum finite element mesh representing the skin thickness. This is accomplished by growing the surface mesh radially 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 which lie along the line from that vertex to the central axis of the cylindrical space of the patient scan data. Note that this extrusion process assumes that the incision cuts into the skin along the r axis.

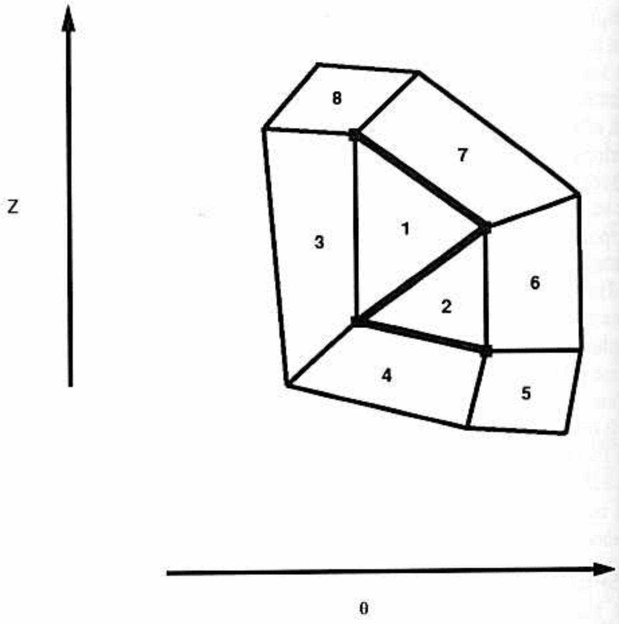


Figure 3. A surface mesh generated from a Z-plasty incision. The original incision lines are indicated in bold. The first stage of the surface meshing algorithm traverses the border of the incision hole and identifies the two concave regions which become surface mesh polygons 1 and 2. The second stage of the algorithm adds polygons 3, 4, 5, 6, 7, and 8. Polygons 5 and 8 result from vertices that were "expanded" because they meet at too sharp an angle.

Figure 4 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.

Surface Meshing Algorithm

The surface meshing approach used in the CAPS system is based on the automatic mesh generation work of Chae and Bathe[3;4]. Their algorithm, which addresses the problem of automatic meshing of CAD parts such as a plate with holes drilled in it, works by creating layers of elements along the borders of the object and working inward until the rows meet. We have modified this approach to work outward from the incision boundary hole and have made the algorithm create quadrilateral elements wherever possible.

Our algorithm consists of two stages: 1) Traverse the border of incision looking for angles larger than a set threshold t_1, convert them to triangles in the surface mesh and update the border. This process continues until no more angles need to be filled. 2) Go around the 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.

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}).* 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 prototype) 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}). Examine the angle between the edges (v_{i-1}, v_i) and (vi, 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, and 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 $v_i + l_1 n_2$.

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 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. Figure 3 shows the surface mesh generated for a Z-plasty incision.

Continuum Meshing Algorithm

Generation of the continuum mesh from the surface mesh is accomplished by extruding the surface mesh inward along the r axis to form solid elements and then 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.

Figure 5 shows the standard finite element used in the CAPS system. The algorithm must generate elements with the proper node ordering. Nodes 0-3 called the top_nodes, are the corners of face 0; nodes 4-7, called the bottom_nodes, are the corners of face 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 we have a set of vertex points connected by a set of polygons. Each polygon has a list of the vertices which defines its shape. An edge of the polygon is defined by each pair of vertices in the list and by the last and first vertices in the list. 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.

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 one half the layer thickness. For continuum meshes with more than one layer of elements in the r direction, 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 sur-

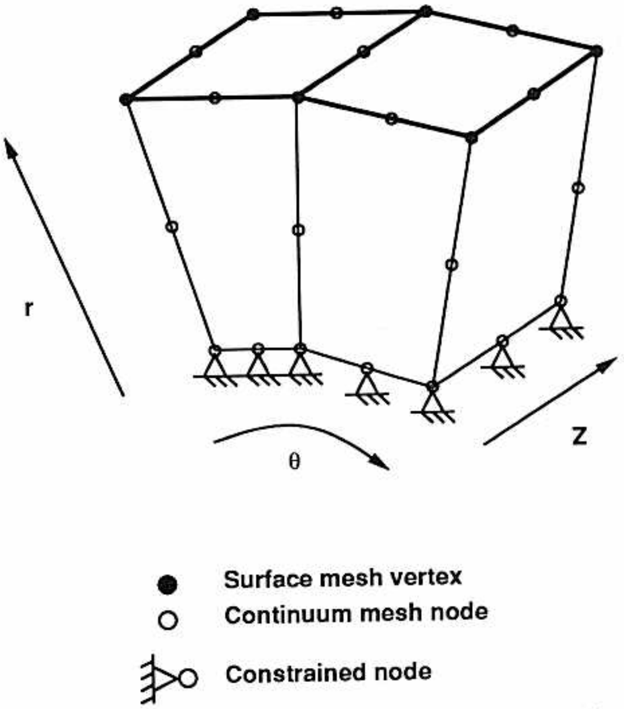


Figure 4. Two elements from a continuum mesh. this shows the relationship between the surface mesh polygons (corresponding to the top faces shown in bold) and the continuum elements. The continuum mesh algorithm generates elements extruded along the r (into the skin) following the topology defined by the surface mesh. The bottom layer of nodes are constrained to remain fixed to represent the bony support. The figure shows a single layer of 20 node elements.

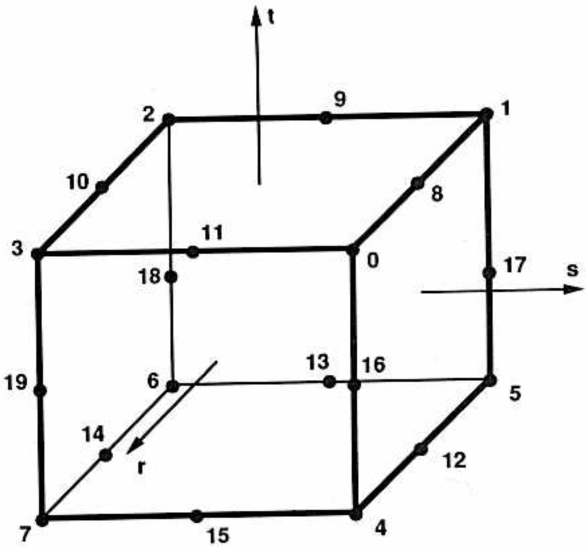


Figure 5. Node numbering for the standard 20 node isoparametric element used in the CAPS system.

---

* 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.

face 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 the element. The indices for the bottom_nodes and the center_nodes are obtained in the same manner, but 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.

Triangles in the surface mesh are handled as a special case by creating wedge shaped elements. This can be accomplished by collapsing one of the side faces of the isoparametric element. In this case, only 15 nodes are created for the element, and a shared node index is used for nodes 2, 10, and 3, for nodes 18 and 19, and for nodes 6, 14, and 7.

7. RESULTS

To date the system has been used in two ways. We have been able to use the system to simulate a number of plastic surgeries of the face and have obtained good visual match between the simulation results and post-operative photos of actual patients. In addition, we have shown the system to over a dozen practicing plastic surgeons and have obtained very positive feedback. Surgeons have noted, for example, that this system is completely different than any current form of surgery planning because it contains an actual model of the elasticity of the skin. This critical feature is missing from most current planning techniques such as drawings or paper models. The other planning techniques which do have some model of skin elasticity (namely cadavers or animal models) do not allow easy iterative design of the procedure.

8. Future Work

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[17;23]. The design of the CAPS system, we have attempted 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;

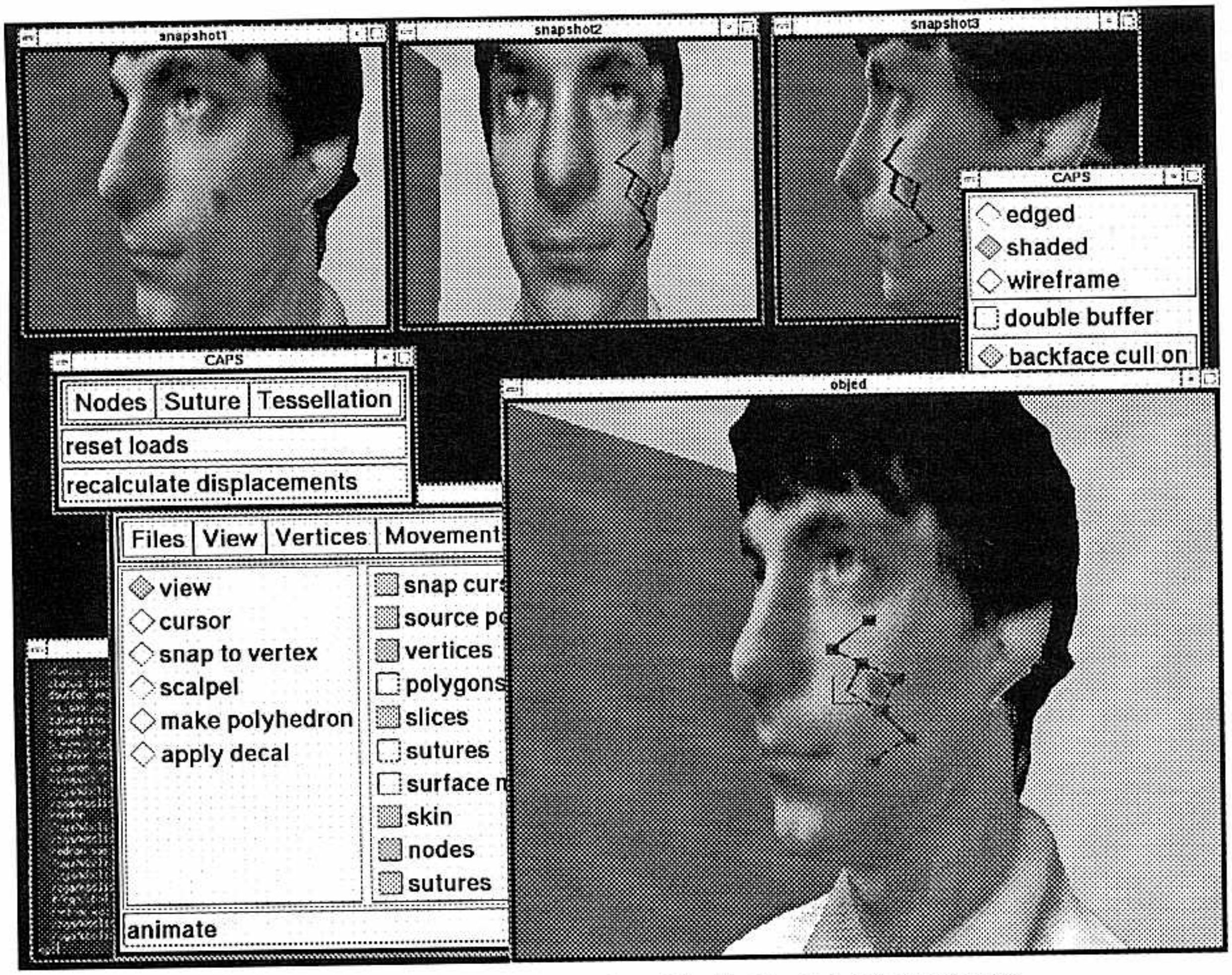


Figure 6. A screen image of the CAPS system in operation showing the patient model and the interactively defined surgical plan.

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.

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. We plan to perform a series of clinical trials to identify the parameters which have the most influence in the surgical result and to obtain accurate estimates of the elastic and viscous moduli of the soft tissue.

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.

The field of plastic surgery simulation is still very new and there are many promising directions for future work. For example, more work is needed to improve modeling of the soft tissue to more accurately model its nonlinear mechanical response and its long term physiological changes. In the future, we would also like to see improved user interface techniques to give the surgeon more control over the direction and depth of the incisions. The current incision technique is adequate for planning surface incisions, but cannot be used for internal surgery.

9. CONCLUSIONS

Simulation of plastic surgery presents many challenging problems which can be addressed by interactive 3D graphics techniques. Each patient presents the surgeon with a unique set of problems for which there are many possible courses of action. The surgeon's goal is to optimize the rearrangement of tissue, to correct the tissue deficiency, and to minimize distortion of the surrounding tissue.

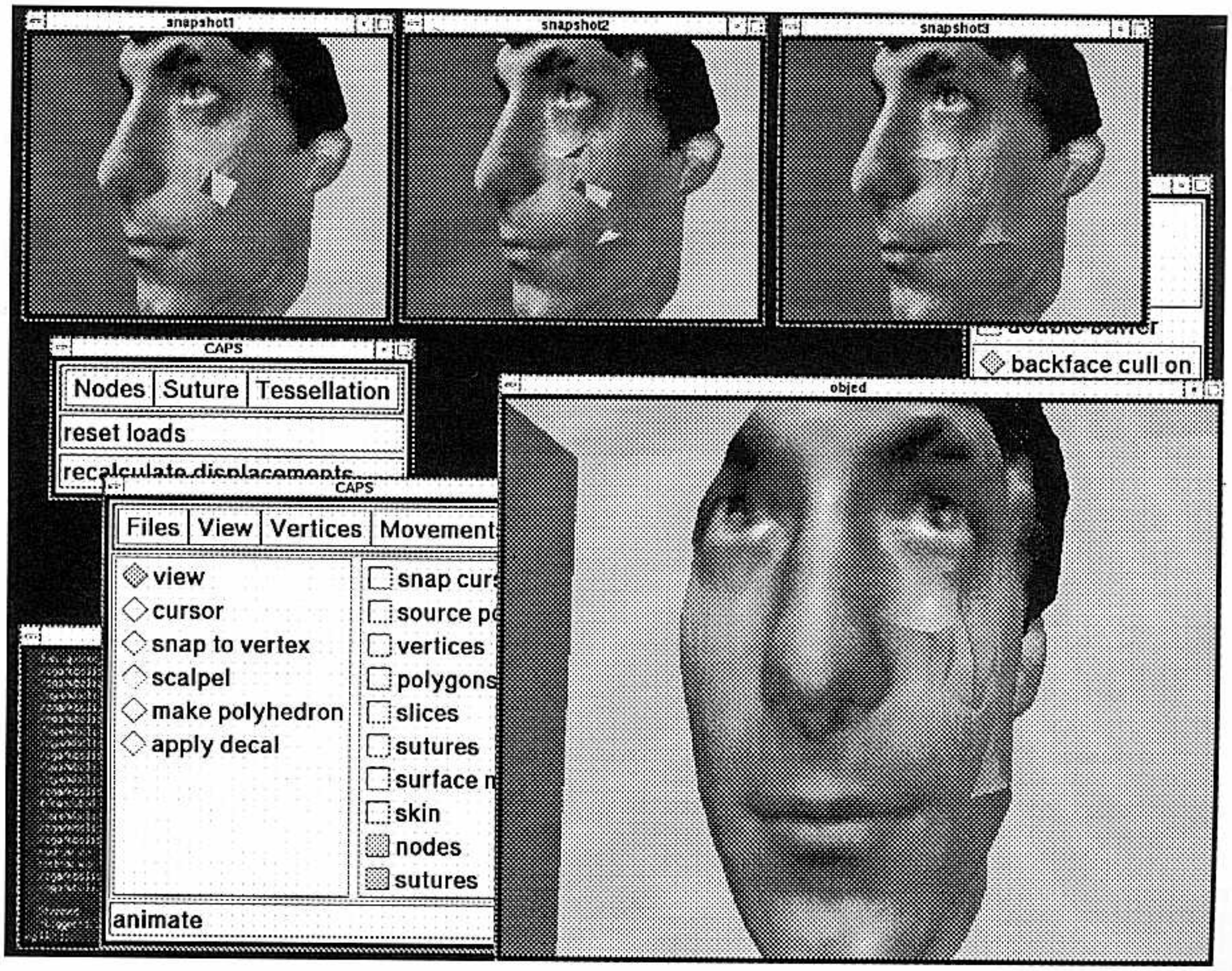


Figure 7. A screen image of the CAPS system in operation showing the simulated results of the operation.

The surgical plan must take into account the complex geometry and mechanical behavior of the soft tissue.

In this paper we have shown how a task level analysis of the plastic surgery planning problem has guided our development and implementation of a computer-aided plastic surgery system. The user interface techniques and mesh generation algorithms we have presented directly address the requirements of the task without burdening the surgeon with the implementation details of the finite element model. Our approach has been well received by clinicians, who report that they would be comfortable using this system to plan operations. However, before we take that step, we will be putting the software through a series of clinical trials to validate the simulation results through retrospective analysis of case histories.

10. ACKNOWLEDGMENTS

This work was supported in part by NHK (Japan Broadcasting Co.) and by equipment grants from Apple and Hewlett Packard. The authors would like to thank David Sturman for being a willing virtual patient.

11. BIBLIOGRAPHY

1. Klaus-Jurgen Bathe, Finite Element Procedures in Engineering Analysis, Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
2. Mike Cedars M.D., Plastic Surgeon, Berkeley, California, Personal Communication.
3. Soo-Won Chae, On the Automatic Generation of Near-Optimal Meshes for Three-Dimensional Linear-Elastic Finite Element Analysis, Ph.D. Thesis, MIT (1988).
4. Soo-Won Chae and Klaus-Jurgen Bathe, On Automatic Mesh Construction and Mesh Refinement in Finite Element Analysis, Computers & Structures, Maxwell Pergamon Macmillan plc. (1989) 32:3/4 pp. 911-936.
5. Grace Hwa-Pei Chang, Computer-Aided Surgery -- An Interactive System for Intertrochanteric Osteotomy, SM Thesis, MIT (1987).
6. David Chen, Pump It Up: Simulating Muscle Shape From Skeleton Kinematics A Dynamic Model of Muscle for Computer Animation, Ph.D. Thesis, MIT (1991).
7. Court Cutting, M.D., Fred L. Bookstein, Ph.D., Barry Grayson, D.D.S., Linda Fellingham, Ph.D., and 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, 77:6, pp. 877-885, (June, 1986).
8. Cyberware Laboratory, Inc. Monterey, California.
9. Robert A. Drebin, Loren Carpenter, Pat Hanrahan, Volume Rendering, Computer Graphics, 22:4, pp. 65-74 (August, 1988).
10. S. Delp, P. Loan, M. Hoy, F. Zajac, S. Fisher, and J. Rosen, An Interactive Graphics-Based Model of the Lower Extremity to Study Orthopaedic Surgical Procedures, IEEE Transactions on Biomedical Engineering, Special issue on interaction with and visualization of biomedical data, 37:8, (August, 1990).
11. Xiao Qi Deng, A Finite Element Analysis of Surgery of the Human Facial Tissues, Ph.D. Thesis, Columbia University (1988).
12. Richard H. Gallagher (ed.), Finite Elements in Biomechanics, Wiley Interscience (1982).
13. William C. Grabb M.D. and James W. Smith M.D., Plastic Surgery, Little, Brown and Company, Boston (1986).
14. Mark Gorney, M.D., Plastic Surgeon, Personal Communication.
15. Pat Hanrahan, Paul Haeberli, Direct WYSIWYG Painting and Texturing on 3D Shapes, Computer Graphics 24:4, pp. 215-223 (1990).
16. Hidehiko Kawabata, M.D., Hideo Kawai, M.D., Kazuhiro Masada, M.D., Keiro Ono, M.D., Computer-Aided Analysis of Z-Plasties, Plastic and Reconstructive Surgery 83:2, pp. 319-325 (February 1989).
17. R. M. Kenedi, T. Gibson, J. H. Evans, J. C. Barbenel, Tissue Mechanics, Phys. Med. Biol. 20:5, pp. 699-717 (1975).
18. Wayne F. Larrabee Jr., M.D., A Finite Element Model of Skin Deformation, Laryngoscope, pp. 399-419 (July, 1987).
19. Mark Levoy, A Hybrid Ray Tracer for Rendering Polygon and Volume Data, IEEE Computer Graphics and Applications 10:2, pp. 33-40 (March, 1990).
20. A. A. Limberg, M.D., 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), english translation S. Anthony Wolfe, M.D., D. C. Heath and Company (1984).
21. R. W. Mann, Computer-Aided Surgery, RESNA 8th Annual Conference, Memphis, Tennessee (1985).
22. Derek R. Ney, Elliot K. Fishman, Donna Magid, and Robert A. Drebin, Volumetric Rendering of Computed Tomographic Data: Principles and Techniques, IEEE Computer Graphics and Applications 10:2, pp. 24-32 (March 1990).
23. Richard M. Peters, Biomechanics and Surgery, in Biomechanics: Its Foundations and Objectives, Prentice-Hall, Englewood Cliffs, New Jersey (1972).
24. Steve Pieper, David Chen, Scott Delp, Michael McKenna, David Zeltzer, and Joseph Rosen, Surgical Simulation: From Computer-Aided Design to Computer-Aided Surgery, Proceedings of IMAGINA '92, Monte Carlo (January 1992).
25. Steven Pieper, CAPS: Computer-Aided Plastic Surgery, Ph.D. Thesis, MIT (1991).
26. Steven Pieper, More Than Skin Deep: Physical Modeling of Facial Tissue, SM Thesis, MIT (1989).
27. David E. Thompson, William L. Buford, Jr., Loyd M. Myers, David J. Giurintano, and John A. Brewer, A Hand Biomechanics Workstation, Computer Graphics 22:4, pp. 335-343 (August 1988).
28. Michael W. Vannier, M.D., Jeffrey L. Marsh, M.D., James O. Warren, M.D., Three Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation, Computer Graphics 17:3, pp. 263-273 (July 1983).
29. Michael W. Vannier, Tom Pilgram, Gulab Bhatia, Barry Brunsden, and Paul Commean, Facial Surface Scanner, Computer Graphics and Applications 11:6, pp. 72-79 (November 1991).
30. Keith Waters, A Muscle Model for Animating Three-Dimensional Facial Expression, Computer Graphics 21:4, pp. 17-24 (1987).
31. Keith Waters and Demetri Terzopoulos, A Physical Model of Facial Tissue and Muscle Articulation, Proceedings of the First Conference on Visualization in Biomedical Computing, Atlanta, Georgia, pp. 77-82 (May 22-25, 1990).
32. David Zeltzer, Steve Pieper, and David Sturman, An Integrated Graphical Simulation Platform, Proceedings of Graphics Interface '89, London, Ontario, pp. 266-274 (June 19-23, 1989).
33. David Zeltzer, Task-level Graphical Simulation: Abstraction, Representation, and Control, in Making Them Move: Mechanics, Control, and Animation of Articulated Figures, Norman Badler, Brian Barsky & David Zeltzer (eds.), Morgan Kaufmann Publishers, Inc., San Mateo, California, pp. 3-33 (1991).