

Volumetric object modeling for surgical simulation

Sarah Gibson^{1*}, Christina Fyock¹, Eric Grimson², Takeo Kanade³, Ron Kikinis⁴, Hugh Lauer¹, Neil McKenzie¹, Andrew Mor^{1,3}, Shin Nakajima⁴, Hide Ohkami¹, Randy Osborne¹, Joseph Samosky^{2,4} and Akira Sawada^{1,5}

¹MERL, 201 Broadway, Cambridge, MA 02139, USA

²Massachusetts Institute of Technology, Cambridge, MA, USA

³Carnegie Mellon University, Pittsburgh, PA, USA

⁴Brigham and Women's Hospital, Boston, MA, USA

⁵Mitsubishi Electric Corporation, Hyogo, Japan

Abstract

Surgical simulation has many applications in medical education, surgical training, surgical planning and intra-operative assistance. However, extending current surface-based computer graphics methods to model phenomena such as the deformation, cutting, tearing or repairing of soft tissues poses significant challenges for real-time interactions. This paper discusses the use of volumetric methods for modeling complex anatomy and tissue interactions. New techniques are introduced that use volumetric methods for modeling soft-tissue deformation and tissue cutting at interactive rates. An initial prototype for simulating arthroscopic knee surgery is described which uses volumetric models of the knee derived from 3-D magnetic resonance imaging, visual feedback via real-time volume and polygon rendering, and haptic feedback provided by a force-feedback device.

Keywords: surgical simulation, volume graphics, volumetric modeling

Received March 3, 1997; revised July 7, 1997; accepted September 11, 1997

1. INTRODUCTION

Computer-based surgical simulation has many applications in medical education, surgical training, surgical planning and intra-operative assistance. In education and training, surgical simulation can reduce costs associated with cadaver specimens, provide experience with a greater variety of pathologies and complications, and provide the ability to repeat or replay training procedures. In surgical planning, simulators can enable rehearsal of difficult procedures or planning on patient-specific anatomy and can enhance communication among medical professionals or between doctors and their patients. Intra-operatively, computer modeling can aid in navigation by augmenting the limited surgical field with a more global view of a patient's anatomy. For example, computer modeling could provide guidance by preventing the surgical instrument from entering sensitive regions.

In order to provide useful feedback to the user, surgical simulators must provide adequate realism. Tissue models should respond in a realistic way when they are manipulated. Rendered images of the surgical field must be realistic enough to be compelling. Haptic or force feedback must mimic forces experienced in real-life because the sense of touch provides important cues in surgery. These requirements impose significant demands on the surgical simulator. These challenges include a need for physically realistic modeling (such as soft-tissue deformation and tissue cutting or tearing) and tradeoffs among physical and visual realism, the need for real-time interaction and cost.

In computer graphics, objects are commonly represented by surface-based polygonal models. Because graphics workstations have special-purpose hardware for fast polygon rendering and because algorithms and systems have been developed for modeling physical interactions between (rigid) polygonal objects, currently there are advantages to using polygonal models in a surgical simulator. However, because

*Corresponding author
(e-mail: gibson@merl.com)

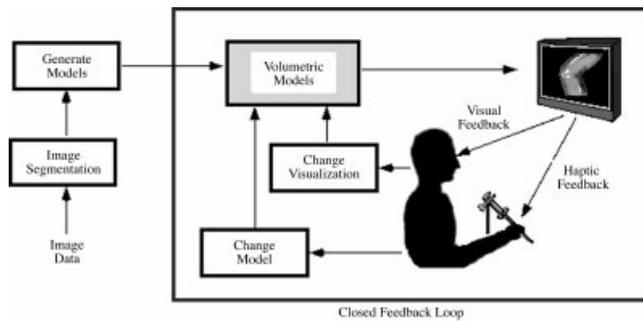


Figure 1. Surgical simulation system components. This system provides a closed feedback loop between the user and the simulator. The user can manipulate object models and observe the results both visually and haptically.

they cannot represent interior structure, surface-based models are inadequate for modeling objects and tissues with complex interiors, for modeling the deformation of arbitrary or heterogeneous volumes and for simulating cutting through tissues. In addition, surface-based methods cannot model fluid flow, tissue tearing or bone fracturing. In contrast, volumetric models can incorporate a great deal of information about interior structure and the mechanical properties of heterogeneous tissues. By adjusting forces that hold volumetric elements together, it may be possible to model fluids, tearing and fracturing using volumetric models. In this paper, we discuss techniques for deforming volumetric object models, for detecting collisions between volumetric objects and for modeling tissue cutting. In addition, we describe an initial prototype system for simulating arthroscopic knee surgery.

The long-term goal of this project is a computer-based surgical simulation system that uses models generated from patient specific data. As illustrated in Figure 1, the system will provide both visual and haptic feedback to the user via real-time rendering and a force-feedback device. Physically realistic interactions between anatomical models and user-controlled surgical instruments will provide a closed feedback loop between the user and the simulator. Important technologies required by the simulator include: image acquisition and segmentation of patient-specific 3-D data to generate object models; visual and haptic feedback through real-time rendering and an electromechanical force-feedback device; manipulation of visual parameters and physically realistic interactions with the model through a haptic interface.

2. PRIOR WORK

There are several surgical simulation systems under development both commercially and in research laboratories.

For example, a number of endoscopy simulators enable navigation through stationary object models (Vining *et al.*, 1993, 1994; Geiger and Kikinis, 1995; Hong *et al.*, 1995; Lorensen *et al.*, 1995; Ziegler *et al.*, 1995). In some navigation or virtual fly-throughs, path planning and collision avoidance between the virtual endoscope tip and static object surfaces are incorporated into the models (Geiger and Kikinis, 1995; Hong *et al.*, 1995; Lorensen *et al.*, 1995). Deformation of surface models is used to model soft-tissues by some surgical simulation systems (e.g. Cover *et al.*, 1993; Kuhnappel *et al.*, 1997). These systems use either mass-spring models or control points on spline-based surfaces for interactive deformation of surface-based object models.

A number of groups have used volumetric methods for modeling deformation and cutting of tissue volumes. Finite-element analysis has been applied to facial and muscle modeling (Waters, 1987; Terzopoulos and Waters, 1990; Chen, 1991), and in surgical simulation (e.g. Pieper, 1992; Hunter *et al.*, 1993; Cotin *et al.*, 1996). Manipulation of voxel-based objects (Kaufman, 1996) has been used in object modeling (Gibson, 1995) and combined with force feedback for haptic exploration of voxel-based objects (Avila and Sobierajski, 1996). Sculpting of volumetric objects is described in Wang and Kaufman (1995) and Galyean and Hughes (1991).

3. VOLUMETRIC METHODS FOR SURGICAL SIMULATION

Modern medical image scanners can provide a great deal of information about internal anatomical structure and function in individual patients. Figure 2 shows a single slice of a 3-D magnetic resonance imaging (MRI) image with resolution $0.25 \times 0.25 \times 1.4 \text{ mm}^3$. Internal structure, such as the heterogeneous muscle and fatty interstitial tissues indicated by arrows in the image, can be important for modeling tissue biomechanics. In surface-based methods, this internal detail is discarded and an object is modeled by a hollow shell. Volumetric object representations provide a way of incorporating all of the information available from the MRI data into a functioning anatomical model.

3.1. Volumetric representation

The volumetric object is represented by a 3-D array of sampled data points. The data can be either regularly or irregularly spaced. Examples of regularly spaced data include image volumes generated by 3-D medical scanners such as MRI or computed tomography (CT) devices. Examples of irregular volumes include data at the nodes of a finite-element simulation or data samples from geological surveys.

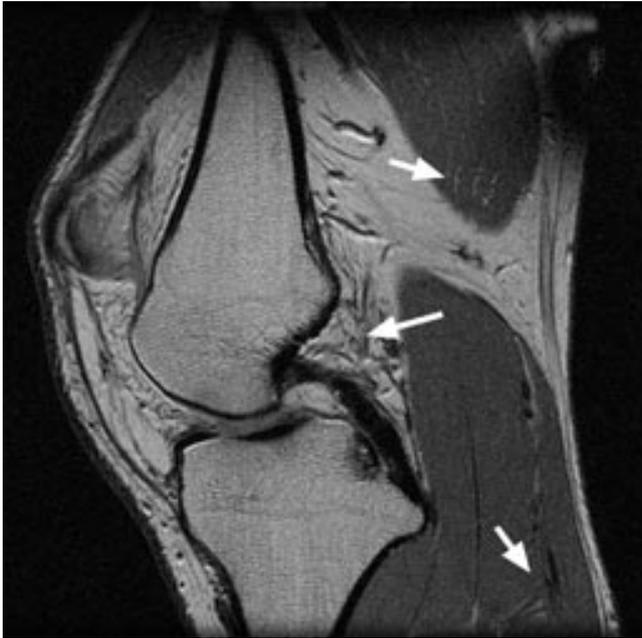


Figure 2. Single plane from a high-resolution MRI image volume with voxel resolution: $0.25 \times 0.25 \times 1.4 \text{ mm}^3$. Arrows indicate structure in muscle and fatty tissue that are important in modeling tissue behavior but that are hard to incorporate into surface-based models.

Elements in the volume encode information about visual and/or physical properties of the object. Each element can be represented by a single value, such as the sampled image intensity, or a more complex data structure that encodes information such as color, transparency, edge strength, elasticity, tissue type and connections to neighboring elements. The ability to encode many different attributes for each element enables the modeling of complex materials, structures and behaviors. However, since volumetric objects can consist of thousands to millions of elements (an MRI image of size $256 \times 256 \times 256$ contains 16 million voxels) this representation poses significant challenges for rendering, data storage and retrieval and tissue modeling. Some approaches for dealing with these large volumes in rendering and tissue modeling are presented here. Future research into hierarchical data representations and data compression will make volumetric methods even more practical for real-time applications.

3.2. Soft-tissue deformation with volumetric models

3.2.1. Mass-spring systems and linear finite-element methods

The two most common methods used for modeling volumetric deformation are mass-spring systems and linear finite-element methods (FEM). Though mass-spring systems and

linear FEM methods are different techniques, in a static simulation both result in a large system of linear equations of the form

$$Ku = F,$$

where K is the object stiffness matrix (a function of material elastic properties which depend on the method), u is the vector representing element displacements and F is a vector representing internal and external forces in the system. In a dynamic simulation, where changes in the shape are observed as the object moves towards equilibrium, the effect of inertial and damping forces are modeled using a system of second-order differential equations:

$$M\ddot{u} + C\dot{u} + Ku = F$$

where M and C are mass and damping matrices respectively, and \dot{u} and \ddot{u} are the first- and second-order derivatives of u with respect to time. In a dynamic computer simulation, this system is evolved through time using numerical techniques.

In both static and dynamic simulations, solving for the displacements, u , is computationally demanding for a large system. The interaction speed can be increased by a significant reduction in the number of elements in the system (Hunter *et al.*, 1993) and/or pre-processing of the data. Bro-Nielsen simplifies the system by solving for displacements of only surface elements and by assuming a small number of externally applied forces (Bro-Nielsen, 1997). Pentland and Williams (1989) pre-calculate the deformation modes of a given object and calculate deformations for an arbitrary force as a superposition of these deformation modes. Bro-Nielsen and Cotin (Bro-Nielsen, 1995; Bro-Nielsen and Cotin, 1996; Cotin *et al.*, 1996) use a similar method, pre-calculating responses to infinitesimal forces and deformations for each node in the element and then approximating the global deformation as a linear superposition of these pre-calculated responses.

These methods have a number of limitations. First, even with significant preprocessing, the size of the deformation is limited and the number of elements must remain relatively small for real-time modeling. Cotin *et al.* use the above method in a system that can deform a liver model with 8000 elements at 15 frames per second^a. This is still a relatively small number of elements, especially if the complex structure of internal vessels and ducts are modeled. In addition, the system requires hours of preprocessing which prohibits interactive changes in topology due to tissue cutting, tearing or suturing.

Second, the assumptions of linearity and superposition are not valid for human tissue. Figure 3 shows a typical stretch versus applied force curve for human tissue [see Fung (1993)

^aPersonal communication from Stephan Cotin, April, 1997.

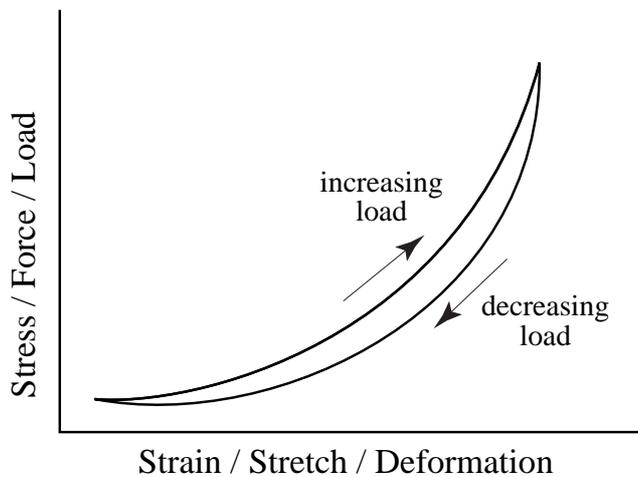


Figure 3. Typical stress–strain or force–stretch curve for human tissues [see Fung (1993) for examples]. Note that the relationship is nonlinear and that the system undergoes hysteresis (the force–stretch curve is different for increasing and decreasing forces).

for several results from experimental data]. Notice that the curve is non-linear, that it exhibits hysteresis (different curves result for increasing and decreasing forces) and that, just before tissue failure or tearing, the tissue stretches very little as the force increases. While non-linear FEM methods could be used for modeling this behavior, these non-linear methods suffer from greater computational complexity.

Finally, standard FEM methods were developed to model metals and other materials with limited elasticity and plasticity. They are based on assumptions of small deformations with a maximum stretch on the order of 1–10% of the object length. In human tissue, deformations of 100–200% of the original tissue length are not uncommon and standard FEM techniques are therefore not appropriate^b.

3.2.2. 3-D ChainMail and elastic relaxation

While FEM methods are promising, current FEM formulations and software implementations are not practical for real-time modeling of complex geometry and detailed interior structure. For this reason, we have developed an algorithm for deforming volumetric objects based on the observation that complex system behavior can result when a large number of elements each follow simple behavior patterns (Gibson, 1997). The tissue deformation algorithm consists of two processes that are applied relatively independently. Both

^bPersonal communication from H. Murakami and T. Impelluso, Department of Applied Mechanics and Engineering Sciences, University of California, San Diego, CA, May, 1997.

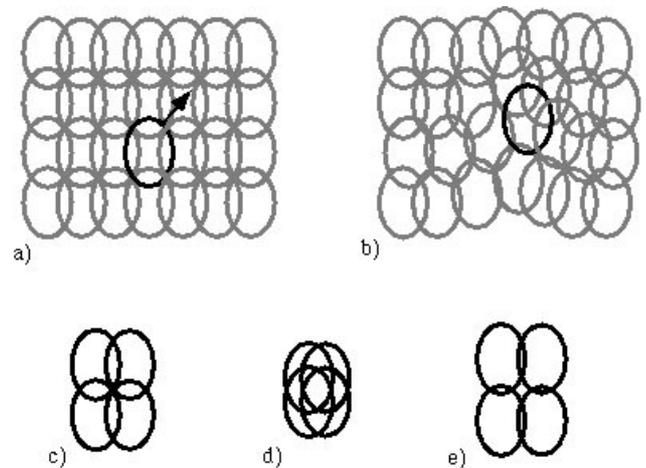


Figure 4. 2-D ChainMail. (a), (b) When one link of the 2-D object is moved along the path of the arrow, its neighboring links move to satisfy maximum and minimum distance requirements between elements. Sets of four links at the bottom of the figure show relative link positions when the four links are in (c) a relaxed state, (d) a maximally compressed state and (e) a maximally stretched state.

processes maintain simple relationships between elements and their neighbors. As in real tissue, disturbances affect only local elements and are propagated to other parts of the volume by local interactions.

The first process of the algorithm, 3-D ChainMail, responds to applied displacements or forces, approximating the new shape of the deformed object quickly, so that collision detection and other physical interactions can be computed. Tissue elements at the site of the disturbance are moved and, much like the links in a chain, neighbors respond to a given node's movement if the constraints on distances between local nodes are violated. This concept is illustrated in the 2-D system of Figure 4. If a link between two nodes is stretched or compressed to its limit, displacements are transferred to neighboring links. In this way, small displacements in a relatively slack system result in only local deformations of the system, while displacements in a system that is already stretched or compressed to its limit causes the whole system to move. Changing the constraints on link lengths allows the modeling of both rigid and deformable objects. 3-D ChainMail is particularly fast for tissues with homogeneous (though possibly anisotropic) material properties because disturbances are propagated through the volume by considering each volume element only once and by comparing each element to, at most, one neighbor.

The second process, elastic relaxation, adjusts element positions to reduce the local system energy—a function of

the distances between neighboring elements. When these distances are within some specified range, the system is in a low-energy state. Larger or smaller distances result in a higher system energy. If the forces applied to reduce the system energy are a linear function of displacements from some optimal spacing, then the system is a linear elastic system. By modifying these relaxation forces or allowing the system to have a range of optimal configurations, non-linear elastic and plastic materials can also be modeled. Elastic relaxation is applied iteratively between applications of 3-D ChainMail and whenever the system is idle. Application of this process tends to relax the shape of the object, smoothing out the approximate shape produced by 3-D ChainMail. The result is that when an object is manipulated directly, it quickly deforms to an approximate shape and then relaxes to a more natural shape as the simulation proceeds. The time constant of this relaxation can be controlled by scaling forces in the elastic relaxation process.

The combined behavior of the two processes exhibits a quasi-static deformation. While inertial and damping forces are not modeled directly, the tendency of elements to not move unless they violate an inequality constraint gives the system inertial behavior, and the fact that the elastic relaxation is a closed feedback loop that moves the system a fraction of the way towards equilibrium with each time interval gives the system a damped behavior.

Both 3-D ChainMail and elastic relaxation have been implemented in 2-D and 3-D. It has been shown that deformation times increase at worst linearly with the number of elements in the object models. In tests on a single-processor (R10K) SGI Onyx workstation, volumetric objects with as many as 125 000 elements have been deformed at interactive rates. These tests have been performed without effort to optimize or parallelize code. This system has not yet been used to simulate human tissue deformation although there are plans to adapt it for soft-tissue in the knee model in the near future. Determining and validating parameters in the deformable model will be an important part of this next step. These issues are discussed further in Subsubsection 4.2.3 in the context of knee modeling applications.

3.3. Collision detection and response

In addition to modeling the deformation of individual objects, a surgical simulation system must model interactions between objects. This requires detecting and reacting to collisions among deforming objects. Collision detection for voxel-based objects is straightforward. We use a method described by Gibson (1995), in which the interaction space is represented by a regular 3-D grid of occupancy map cells. In this approach, each element of each object is mapped into a cell in the occupancy map. As an object is moved, cell contents in

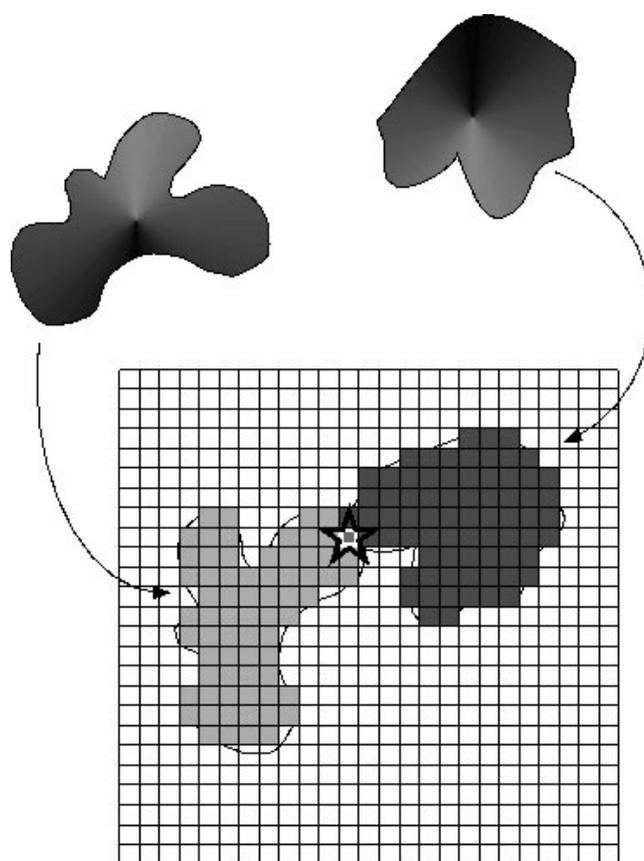


Figure 5. Simple collision detection for volumetric objects. Object elements are mapped into cells of the regular grid of the 3-D occupancy map. As objects are moved, cells of the occupancy map are updated. If an element is written into a cell that is already occupied, a collision is detected.

the occupancy map are updated. A collision between objects is detected if an element is mapped into an occupied cell in the occupancy map. Figure 5 demonstrates this algorithm in 2-D. Using this approach, it is possible to detect collisions between both rigid and deformable objects and to detect and prevent self-intersections. This approach has been used interactively to detect collisions between several objects of size 64^3 in real-time on an SGI workstation with an R10K processor. We are investigating a number of methods to improve the speed of the collision-detection algorithm. These include mapping only surface elements into the occupancy map, storing the object in a hierarchical data structure such as octrees to reduce the search space for collision detection (He, 1997), and the use of a dynamic occupancy map whose size corresponds only to the overlap regions of subvolumes in a hierarchical data structure rather than the entire interaction space.

In the current system, collisions are detected and object interpenetration is prevented but the system does not yet model collision response. In future systems, we intend to model system dynamics so that physically realistic reactions to collisions can be simulated. Recently, there has been significant progress in computer graphics for real-time simulation of collision response for rigid surface models (e.g. Baraff, 1989; Mirtich and Canny, 1995). While these methods cannot be applied directly to volumetric models because of the large number of contact points between volumetric objects, we believe that by averaging collision contacts over the contact regions rather than considering all contact points individually, collision responses could be calculated at reasonable rates. We are currently implementing a system for computing collision responses between rigid volumetric models of the bones in the knee.

3.4. Tissue cutting, tearing and suturing

Modeling surgical interventions requires the simulation of tissue cutting, tearing and repairing or suturing. Using a volumetric model with links between neighboring elements, cutting and tearing are performed by breaking connections between neighbors. For cutting, these connections are broken along the path of the knife instrument as it is passed through the virtual object. Intersections between the knife path and the object are detected by moving the knife volume through the occupancy map and checking for collisions. If the knife path encounters a cell occupied by an element or a line linking surface elements, the appropriate neighbor connections are broken. Tearing occurs when the distance between two elements is stretched beyond an allowable limit, for example, when two parts of an object are pulled in opposite directions. When a limit violation between two elements cannot be resolved by moving neighboring elements, the connection between the elements is broken.

For joining or suturing objects, elements along the path of the joining instrument that have missing neighbor connections are paired and joined. These edge elements are detected by searching occupancy map cells within the vicinity of the joining instrument. Figure 6 shows the user interface of a 2-D system that allows object translation, deformation, arbitrary cutting, joining (or gluing), element erasing and tacking of elements into place. Application of this system to simulating cutting through knee ligaments is described below.

4. SIMULATION OF ARTHROSCOPIC KNEE SURGERY

As part of an ongoing collaboration between MERL, Carnegie Mellon University, Massachusetts Institute of Technology, and Brigham and Women's Hospital, we have



Figure 6. User interface for an interactive 2-D application for manipulated 2-D volumetric objects. Object elements can be moved (resulting in object translation), cut, grasped and moved (resulting in object deformation), tacked into place, glued together, and erased interactively using the computer mouse. Pointing and clicking on the buttons on the right-hand side of the user interface switches between these modes.

built a prototype simulation system to implement and test some of the ideas that have been presented in this paper (Figure 7). The current system consists of volumetric models of knee anatomy created from MRI image scans, a haptic interface device for interacting with the knee model and real-time rendering for visualization of the interaction. The current status and future plans for this system are described in more detail below.

4.1. Motivation

For our initial prototype system, we have focused on arthroscopic knee surgery, a relatively new and minimally invasive procedure that is often used to diagnose and treat knee injuries. In arthroscopy, the joint is visualized and accessed through small portals. An optical endoscope equipped with a video camera allows visualization of the procedure through one of the portals, while surgical probes and other instruments are inserted into additional portals (see Figure 8).

We have focused on arthroscopic knee surgery for a number of reasons. First, there are a large number of arthroscopic knee surgeries each year. In the US, it is estimated that knee procedures represent 88–90% of the approximately 1.8 million arthroscopic procedures performed in 1996^c. While

^cPersonal communication from Alan Praemer, Department of Research and Scientific Affairs, American Academy of Orthopedic Surgeons, August, 1996.



Figure 7. Prototype surgical simulation system. A 3-D volumetric knee model has been generated from MRI images. A force-feedback device allows the user to haptically explore the bony surfaces of the knee model while real-time polygon and/or volume rendering provide visual feedback to the user.

arthroscopic procedures have been shown to reduce costs and increase patient recovery rates, they suffer from specific technical limitations, namely, limited visibility through the arthroscope; difficulty in orienting the camera to the surgeon's viewpoint and restricted motion of surgical tools. Computer simulation will enhance the education and training of surgeons, and help them deal with these technical challenges.

Second, the knee offers an ideal initial platform for this project. Important structures lie in a limited volume so that the size of a volumetric knee model is reasonable. Many of the important structures in the knee are rigid so initial haptic and rendering implementations can be performed on static data sets. In addition, the soft-tissues that are of primary importance—cartilage layers, the menisci and cruciate ligaments—are small enough for reasonable testing of tissue deformation and cutting algorithms and for performing real-time rendering. As faster hardware, more efficient algorithms and better data representations are developed, techniques developed for the arthroscopic knee simulator will be extended to other systems.

Finally, the knee has been modeled in a number of studies focusing on biomechanics and bio-dynamics modeling (e.g. Kaufman, 1988; Fijan, 1990; Blankevoort *et al.*, 1991) experimental measurements (e.g. Blankevoort *et al.*, 1984; Fung, 1993) and surgical simulations (e.g. Logan *et al.*, 1996; Bajaj *et al.*, 1997). This provides a body of literature for testing

dynamic predictions of our model, obtaining measured tissue parameters and comparing our results with other surgical simulation systems.

4.2. Prototype system: current status

In the prototype system, object models consist of the bony structures of the joint, the articular cartilage, the menisci and the cruciate ligaments, all of which were hand-segmented from MRI data. The integrated system currently allows probing of bony structures with real-time visual and haptic feedback. Visual feedback is provided by volume rendering, polygon rendering and sectional imaging (Samosky, 1993).

The research platform for technology development and the current simulator prototype consists of an eight-processor SGI Challenge with MIPS R10K processors, Infinite Reality graphics, four RM6 raster manager boards with 3-D texture mapping (for interactive volume rendering) and 512 Mbytes of RAM. Haptic feedback is provided by a SensAble Technologies' PHANToM with three degrees-of-freedom force reflection and six degrees-of-freedom sensing of position and orientation. The PHANToM is currently controlled via an SGI Indigo2 connected to the SGI Challenge with an ethernet connection.

4.2.1. Image acquisition

For the initial prototype system, a T-1 weighted proton density MRI knee image sequence was acquired. The image size was $256 \times 256 \times 124$ with a voxel size of $0.63 \times 0.63 \times 0.9 \text{ mm}^3$. This image was hand-segmented^d into bony structures (femur, tibia, fibula, patella), cartilage (femoral, tibial and patellar), lateral and medial menisci, and anterior and posterior cruciate ligaments. These structures are illustrated in the surface rendered image of Figure 9.

We have recently acquired a high-resolution knee image of size $512 \times 512 \times 90$, with voxel size $0.25 \times 0.25 \times 1.4 \text{ mm}^3$. The image acquisition time was 50 min. Both T1-weighted proton density images and fat-suppressed images were acquired. This data set has been hand segmented into bony structures, articular cartilage, menisci, cruciate ligaments and the quadriceps and gastrocnemius muscles and tendons. These higher resolution models will be used to replace existing models in the near future.

^dUnlike CT images, where thresholding techniques can be used effectively for segmentation, there are no general automatic segmentation tools for MRI images. Although semi-automatic segmentation techniques can be fine tuned within a limited application, we do not currently have such a system customized for the knee. While hand-segmentation is tedious and time consuming, it has provided the models required by other components of the surgical simulation system. Future plans include the development of image segmentation tools specialized for knee anatomy that would reduce the segmentation time and facilitate the use of patient-specific models.



Figure 8. Illustration of arthroscopic knee surgery. The knee is accessed through small portals. A video endoscope inserted through one portal captures a live image of the surgical field and projects it onto a video monitor. The surgeon manipulates surgical instruments through additional portals.

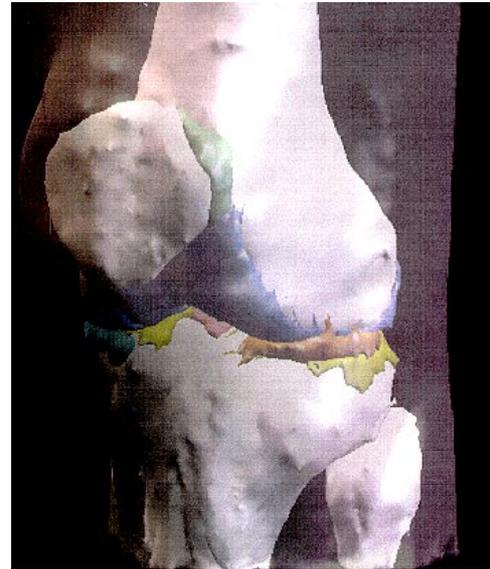


Figure 9. Surface rendered image of a segmented knee. The knee was segmented into bony structures (femur, tibia, fibula, patella), cartilage (femoral, tibial and patellar), lateral and medial menisci, and anterior and posterior cruciate ligaments.

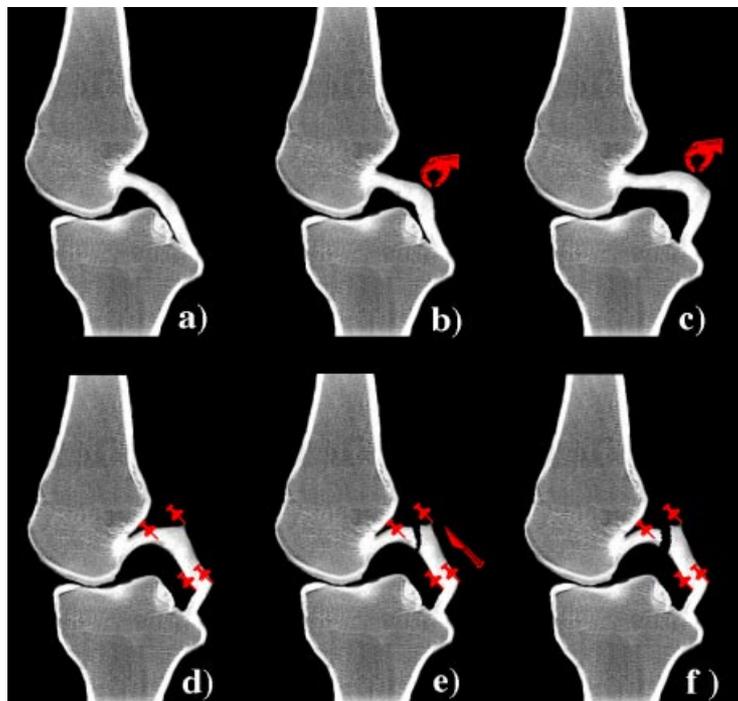


Figure 10. Two-dimensional system showing soft-tissue deformation and cutting. In (b) and (c), the posterior cruciate ligament is grasped and pulled. In (d), the ligament is stretched and tacked into place. In (e) the ligament is cut. (f) Shows how the elastic ligament tissue pulls back from the cut.

4.2.2. Deformation, cutting and suturing

Tissue deformation, cutting and suturing have not yet been integrated into the prototype surgical simulation system. As described above, we have implemented a test system that performs interactive deformation of a 3-D object containing 125 000 elements, significantly more elements than the cartilage and ligaments in our knee model. Hence we believe that interactive deformation of these structures will not be prohibitively expensive. We have also implemented the 2-D system shown in Figure 6 that allows simulation of cutting, suturing and tissue deformation on a 2-D model. Figure 10 is a sequence of images captured during interaction with this system. The 2-D objects consist of rigid models for the tibia and femur and deformable soft-tissue models for the tibial and femoral cartilage and the posterior cruciate ligament. In Figure 10b and c, the grasping tool is used to grab and pull on the posterior cruciate ligament. Figure 10d shows how the posterior cruciate ligament is stretched and tacked in place. Figure 10e illustrates how the ligament is cut with the knife tool. In Figure 10f, note that the elastic nature of the ligament causes its shape to relax away from the cut edge after cutting.

Validation of the tissue models will be a crucial focus of the next phase of this project. While 3-D ChainMail has shown promising speed and has been shown to capture complex tissue behaviors, it has not yet been validated against measurements from real tissues. In prototypes of this deformation technique, rather than modeling particular tissues, the focus was on testing for specific behaviors such as rigidity versus deformability and elasticity versus plasticity. Parameters for material properties must be set and tailored to patient-specific data. These parameters can be derived from results reported in the literature or from laboratory measurements, or they can be hand-tuned by surgeons using the interactive system. Methods for validating the resultant model include comparing the model predictions to experimental data, comparing the resultant simulation to predictions from FEM calculations, and user testing with surgeons and other experienced users.

4.2.3. Haptic interaction

The surgical simulation prototype uses SensAble Technologies' PHANToM to provide force reflectance and feedback for haptic interaction (Figure 7). The PHANToM provides six degrees of sensing (3-D position and orientation) and three-degrees of force feedback (position only). Currently the force-feedback device allows the user to explore rigid object models using their sense of touch. We plan to provide force feedback during simulation of tissue deformation and cutting in the next prototype system. The PHANToM device provides gross motion and force feedback similar to what one would feel by grasping a pen and running its tip over a 3-D surface. Adding

tactile feedback, using sensors such as those of Howe *et al.* (1995), could greatly enhance the system.

The object models in the surgical simulation system are derived from segmented MRI images. Because the haptic resolution of the PHANToM is better than the resolution of the data, the binary-segmented object models must be smoothed before surface normals are calculated. If the models are not smoothed, the binary data causes the direction of the surface normals to vary widely with small changes in position, resulting in unstable force feedback.

In our prototype implementation, we use a haptic model similar to the density field method of Avila and Sobierajski (1996). A discrete 3-D density field image was generated by smoothing the binary-segmented data of bony structures of the knee with a $5 \times 5 \times 5$ Gaussian filter. This provided a smooth gray-scale image whose intensities vary from zero outside objects to 255 within an object. The magnitude of the feedback force is taken to be the product of a stiffness constant k , and the value of the density field at the PHANToM tip. For relatively small penetrations, the density values are a function of the distance from the surface of the binary volume. The stiffness constant was interactively hand-tuned by a surgeon until the interaction with the bone models felt realistic. The direction of the feedback force is taken to be the normalized gradient of the density field image, which points outward in the direction of the surface normal. The gradient is calculated using a central-difference method from trilinearly interpolated densities of six positions located one unit distance above, below, behind, in front, to the right and to the left of the tip position. The resultant force is

$$\underline{f}(x, y, z) = k \cdot d(x, y, z) \cdot \underline{g}(x, y, z) [\|\underline{g}(x, y, z)\|]^{-1},$$

where $\underline{g}(x, y, z) = (d(x-1, y, z) - d(x+1, y, z), d(x, y-1, z) - d(x, y+1, z), d(x, y, z-1) - d(x, y, z+1))$, and $d(x, y, z)$ is the linearly interpolated density at (x, y, z) , the PHANToM tip position. This method of calculating the reflected forces requires static data and is only suitable for rigid-body interactions. It does not model the forces that would be reflected by elastic tissues, an issue that will be addressed in future systems.

In the current prototype, the PHANToM is controlled by an SGI Indigo2 while graphics and modeling are performed on an SGI Challenge. In order to enable interactions at rates greater than 1 kHz we store two models of the knee; a haptic model on the Indigo2 and a visual model on the Challenge. An ethernet connection is used to exchange information about the PHANToM tip and orientation. This use of separate haptic and visual object representations is not practical for deformable model manipulation since changes to the models must be communicated over the (relatively slow) ethernet connection. We are currently investigating ways to control

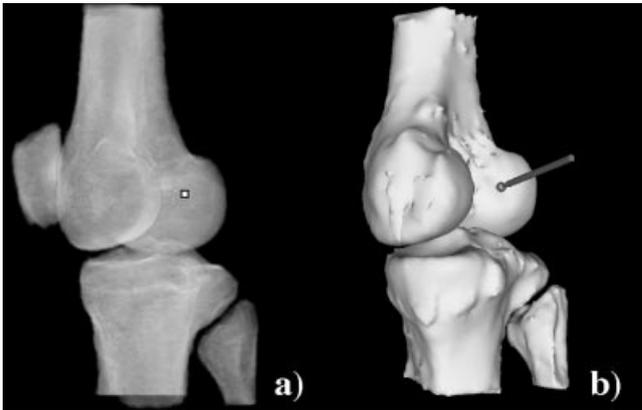


Figure 11. Rendered images of knee bones in the surgical simulation prototype system: (a) volume rendered using SGI texture map hardware and (b) polygon rendered image of surface models.

the PHANToM directly from the SGI Challenge to reduce this problem.

4.2.4. Real-time rendering

Interactive visualization of the simulation is essential for the prototype system. The current system uses two approaches for visualizing object models: volume rendering using the 3-D texture map memory available on our research platform (Cabral *et al.*, 1995) and hardware-accelerated polygon rendering of surface models that were generated from the segmented images using marching cubes and triangle decimation (Lorenson and Cline, 1987; Schroeder *et al.*, 1992). A cursor indicating the position of the tip of the force-feedback device is rendered in both approaches. Figure 11 shows two sample renderings.

Unlike the texture map volume rendering implementation, polygon rendering allows us to improve the visual appearance of the rendered image by adding shading and texture. However, because polygonal models are not easily updated for objects that are deforming and changing topologically, and because we will require a consistent representation for physical modeling, haptic feedback and visual rendering, we intend to use higher-quality volume rendering in future systems. We are currently investigating both software and hardware approaches for real-time volume rendering.

As part of a parallel project, MERL and Mitsubishi Electric are developing special purpose volume rendering hardware for interactive PC-based applications (Osborne *et al.*, 1997). The immediate goal of this project is hardware acceleration for volume rendering. A longer term goal is the integration of volumetric and surface-based rendering in 3-D graphics

hardware. The availability of fast rendering hardware for real-time, high-quality rendering of volumetric models will complement our activities in surgical simulation.

While deformable models have not yet been integrated into the prototype system, we have investigated a number of methods for rendering irregular volumes. We have implemented several of these, including raycasting (Ramamoorthy and Wilhelms, 1992), cell projection (Shirley and Tuchman, 1990; Williams, 1992) and splatting (Westover, 1990). While it may be possible to integrate some of these techniques into hardware, they are currently too slow for real-time object deformation. Deformed volumetric models are currently visualized using OpenGL to render surface elements as points. The resultant images provide enough visual feedback to guide the interaction, but they lack visual quality. We are currently investigating splatting and polygon-based rendering methods that exploit the known connections between neighboring surface elements to provide higher-quality rendering of deformable models.

5. DISCUSSION AND CONCLUSIONS

Volumetric methods provide powerful tools for modeling objects with complex interior structure and tissue properties. In this paper, we have discussed some of the advantages and technical challenges that are faced in using volumetric methods in surgical simulation. We have also introduced techniques for modeling and interacting with volumetric objects and we have described an initial prototype simulator that uses volumetric methods for modeling arthroscopic knee surgery.

While we advocate the use of volume methods, we readily acknowledge that there are many times when surface models are more suitable. For example, if a CAD model of a surgical instrument exists, there may be no advantage to converting the model to a (possibly) less accurate volumetric model. Hybrid methods that combine both polygonal and volumetric models for rendering, haptics and physically realistic modeling would be ideal. However, there are many challenges in modeling and rendering that must be overcome before these hybrid methods become feasible. In the meantime, volumetric methods provide a powerful complement to surface graphics for modeling, visualizing and interacting with computer models.

ACKNOWLEDGEMENTS

We gratefully acknowledge the Wellness Business Development Center, Mitsubishi Electric Corporation, Japan, for financial support in this collaboration. We thank Edward Gibson and three anonymous reviewers of this paper for many helpful and insightful comments and suggestions.

REFERENCES

- Avila, R. and Sobierajski, L. (1996) A haptic interaction method for volume visualization. In Yagel, R. and Nielson, G. (ed.), *Proc. Visualization '96*, pp. 197–204.
- Bajaj, C., Bernardini, F., Cutchin, S., Lin, K., Sacks, E. and Schikore, D. (1997) *Comprehensive Analysis of Joints from Patient Clinical Data*. Technical Report CSD-97-019, Department Computer Science, Purdue University.
- Baraff, D. (1989) Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Proc. SIGGRAPH, Computer Graphics*, Vol. 24, pp. 19–28.
- Blankevoort, L., Huijskes, R. and de Lange, A. (1984) An *in-vitro* study of the passive kinematic behavior of the human knee joint. In Spilker, R. L. (ed.), *1984 Advances in Bioengineering*, pp. 57–58. ASME, New Orleans, LA.
- Blankevoort, L., Kuiper, J. H., Huijskes, R. and Grootenboer, H. J. (1991) Articular contact in a three-dimensional model of the knee. *J. Biomech.*, 24, 1019–1031.
- Bro-Nielsen, M. (1995) Modeling elasticity in solids using active cubes—application to simulated operations. In *Computer Vision, Virtual Reality and Robotics in Medicine, Lecture Notes in Computer Science*, Vol. 905, pp. 535–541. Springer-Verlag, Berlin.
- Bro-Nielsen, M. and Cotin, S. (1996) Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In *Proc. Eurographics, '96*, Vol. 15, pp. 57–66.
- Bro-Nielsen, M. (1997) Fast finite elements for surgery simulation. In *Proc. Medicine Meets Virtual Reality V*.
- Cabral, B., Cam, N. and Foran, J. (1995) Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proc. Symp. on Volume Visualization*, Washington, DC, October, pp. 91–98.
- Chen, D. (1991) *Pump it up: Computer Animation of a Biomechanically Based Model of Muscle Using the Finite Element Method*. Ph.D. Thesis, Media Lab, MIT.
- Cotin, S., Delingette, H., Clement, J. M., Soler, L., Ayache, N. and Marescaux, J. (1996) Geometrical and physical representations for a simulator of hepatic surgery. In *Proc. Medicine Meets Virtual Reality IV*.
- Cover, S., Ezquerro, N., O'Brian, J., Rowe, R., Gadacz, T. and Palm, E. (1993) Interactively deformable models for surgery simulation. *IEEE Comp. Graphics Appl.*, 13, 68–75.
- Fijan, R. S. (1990) *A Three-dimensional Mathematical Model of the Human Knee Joint*. Ph.D. Thesis, Massachusetts Institute of Technology.
- Fung, Y. C. (1993) *Biomechanics: Mechanical Properties of Living Tissues*. Springer-Verlag, New York.
- Galyean, T. and Hughes, J. (1991) Sculpting: an interactive volumetric modeling technique. In *Proc. SIGGRAPH, Computer Graphics*, Vol. 25, pp. 267–274.
- Geiger, B. and Kikinis, R. (1995) Simulation of endoscopy. In Ayache, N. (ed.), *Proc. CVRMed '95*, pp. 277–281. Springer-Verlag, Berlin.
- Gibson, S. F. (1995) Beyond volume rendering: visualization, haptic exploration, and physical modeling of voxel-based objects. In Scateni, R., van Wijk, J. and Zanarini, P. (eds), *Visualization in Scientific Computing*, pp. 10–24. Springer-Verlag, Berlin.
- Gibson, S. F. (1997) 3D ChainMail: a fast algorithm for deforming volumetric objects. In *Proc. Symp. on Interactive 3D Graphics, ACM SIGGRAPH*, pp. 149–154.
- Hong, L., Kaufman, A., Wei, Y., Viswambharen, A., Wax, M. and Liang, Z. (1995) 3D virtual colonoscopy. In Loew, M. and Gershon, N. (eds.), *Proc. of Biomedical Visualization '95*, pp. 26–32. IEEE Computer Society Press, Los Alamitos.
- Howe, R., Peine, W., Kontarinis, D. and Son, J. (1995) Remote palpation technology for surgical applications. *IEEE Engng Med. Biol. Mag.*, 14, 318–323.
- Hunter, I., Doukoglou, T., Lafontaine, S., Charette, P., Jones, L., Sagar, M., Mallinson, G. and Hunter, P. (1993) A teleoperated microsurgical robot and associated virtual environment for eye surgery. *Presence*, 2, 265–280.
- Kaufmann, A. (1996) *Volume Visualization*. CRC Handbook of Computer Science and Engineering.
- Kaufman, K. R. (1988) *A Mathematical Model of the Muscle and Joint Forces in the Knee During Isokinetic Exercise*. Ph.D. Thesis, North Dakota State University.
- Kuhnappel, U. G., Kuhn, C., Hubner, M., Krumm, H. G., Mass, H. and Neisius, B. (1997) The Karlsruhe Endoscopic Surgery Trainer as an example for virtual reality in medicine. *Minimally Invasive Therapy Allied Technol.*, 6, 122–125.
- Logan, I. P., Wills, D., Avis, N., Mohsen, A. and Sherman, K. (1996) Virtual environment knee arthroscopy training system. *Society for Computing Simulation, Simulation Series*, Vol. 28, pp. 17–22.
- Lorensen, W. E. and Cline, H. E. (1989) Marching cubes: a high resolution 3D surface construction algorithm. *Comp. Graphics*, 21, 163–169.
- Lorensen, W. E., Jolesz, F. A. and Kikinis, R. (1995) The exploration of cross-sectional data with a virtual endoscope. In Satava, R. and Morgan, K. (eds), *Interactive Technology and the New Medical Paradigm for Health Care*, pp. 221–230.
- Mirtich, B. and Canny, J. (1995) Impulse-based simulation of rigid bodies. In *Proc. 1995 Workshop on Interactive 3D Graphics*, pp. 181–188.
- Osborne, R., Pfister, H., Lauer, H., McKenzie, N., Gibson, S., Hiatt, W. and Ohkami, H. (1997) EM-Cube: an architecture for low-cost real-time volume rendering. *1997 SIGGRAPH/Eurographics Hardware Workshop*, Los Angeles, CA.
- Pentland, A. and Williams, J. (1989) Good vibrations: modal dynamics for graphics and animation. In *Proc. SIGGRAPH, Computer Graphics*, Vol. 23, pp. 215–222.
- Pieper, S. D. (1992) *CAPS: Computer-aided Plastic Surgery*. Ph.D. Thesis, MIT.
- Ramamoorthy, S. and Wilhelms, J. (1992) *An Analysis of Approaches to Ray-tracing Curvi-linear Grids*. Technical Report UCSC-CRL-92-07, Computer Research Laboratory, UC Santa Cruz.

- Samosky, J. (1993) *SectionView—a System for Interactively Specifying and Visualizing Sections Through Three-dimensional Medical Image Data*. MS Thesis, MIT.
- Shirley, P. and Tuchman, A. (1990) A polygonal approximation to direct scalar volume rendering. In *Proc. San Diego Workshop on Volume Visualization*, pp. 63–70.
- Schroeder, W. J., Zarge, J. A. and Lorensen, W. E. (1992) Decimation of triangle meshes. *Comp. Graphics*, 26, 65–70.
- Terzopoulos, D. and Waters, K. (1990) Physically-based facial modeling, analysis, and animation. *J. Visualization Comp. Animation*, 1, 73–80.
- Vining, D., Liu, K., Choplin, R. and Hapnoik, E. (1996) *Virtual Bronchoscopy: Relationships of Virtual Reality Endobronchial Simulations to Actual Bronchoscopic Findings*. *Chest*, 109, 549–553.
- Vining, D., Stelts, D., Ahn, D., Hemler, P., Ge, Y., Hunt, G., Siege, C., McCorquodale, D., Sarojak, M. and Ferretti, G. (1997) FreeFlight: A Virtual Endoscope System. *Proc. CVRMed-MRCAS'97* (Joint conference on Computer Vision and Virtual Reality and Medical Robotics and Computer Assisted Surgery), Grenoble, pp. 413–417.
- Wang, S. and Kaufman, A. (1995) Volume sculpting. In *ACM Symp. on Interactive 3D Graphics*, Monterey, CA, April, pp. 151–156.
- Waters, K. (1987) A muscle model for animating three-dimensional facial expression. *Comp. Graphics*, 21, 17–24.
- Westover, L. (1990) Footprint evaluation for volume rendering. In *Proc. SIGGRAPH, Computer Graphics*, Vol. 24, pp. 144–153.
- Williams, P. L. (1992) Interactive splatting of non-rectilinear volumes. In *Proc. IEEE Visualization '92*, pp. 37–44.
- Ziegler, R., Fischer, G., Muller, W. and Gobel, M. (1995) A virtual reality medical training system. In Ayache, N. (ed.), *Proc. CVRMed '95*, pp. 282–286. Springer-Verlag, Berlin.