Virtual Cutting of Anatomical Structures

Gerrit Voß[1] , **James K. Hahn (PhD)**[2] ,
Wolfgang Müller[1] and **Rob Lindeman**[2]

[1] Fraunhofer Institute for Computer Graphics
Dep. Visualization & Virtual Reality
Rundeturmstr. 6
64283 Darmstadt
Germany
e-mail {vossg,muellerw}@igd.fhg.de
URL http://www.igd.fhg.de/igd-a4

[2] Labratory for Advanced Computer Applications in Medicine
Institute for Computer Graphics
The George Washington University
Washington, DC 20052
email {hahn,gogo}@seas.gwu.edu

Abstract

In the context of medicine cutting is one of the most important operation. Many surgical tasks start with an incision, allowing the surgeon to access the region of interest, using either conventional or minimal invasive surgery techniques. The work presented herein should be seen in the context of surgical training and preoperative planning using computer assisted medical simulation systems. The main result of this work is the proposed cutting algorithm based on predefined templates. To integrate this approach in the existing surgical training simulator and to handle additional tasks like soft tissue deformation, needed by the medical environment, a generic simulation framework is introduced. This framework is able to utilize given multiprocessor- and network-environments to achieve the desired realtime interactivity.

1 Introduction

The use of virtual reality has been demonstrated in different application domains, in medicine the use of virtual environments opens up new possibilities in the areas of training and treatment planning, see [12], [3]. Previous work focused on providing training environments for minimal invasive surgery, covering instrument handling and simple diagnostic tasks, see [9] for details. Using virtual reality interaction techniques the trainee is able to acquire basic skills of instrument handling before facing a real patient. In the current scenario the students are facing plastic replicas of the instruments and the cover of the anatomical region. Observing the computer generated camera image the trainees learn to coordinate their three dimensional instrument locomotions with the two dimensional camera image. First approaches in simulation of tissue deformation were realized. The deformation of anatomical structures by using a probe is the main method used during a diagnostic procedure. The next step towards a realistic medical simulator is the integration of cutting procedures. This work focuses on cutting of anatomical structures. In order to provide a realistic simulation of tissue cutting by surgical instruments the detection of collisions and the simulation of tissue deformations have to taken into account.

2 Methods

Since cutting has impact on both the rendering and the simulation part of the current system we will start with the description of the generic simulation architecture underlying this work, followed by the different simulation parts.

2.1 Simulation Architecture

One major problem in realtime graphics is the generation of at least ten frames per second at a nearly constant framerate. There are two ways to achieve this constraint, either to finish all calculations in time or to delay the transfer of the simulation results to the rendering database for a few frames until the calculations deliver displayable results. To finish the calculation in time one can either reduce the complexity of the simulation or distribute the calculations among different processors or machines. The second solution is realized by duplicating the databases using one copy for simulation and one for rendering. The synchronisation between the two databases is done after the simulation calculations are finished and before the next frame is drawn. During the simulation the drawing is done using the not updated rendering database to provide the necessary framerate. The needed database update must fit in between to drawing calls. One crucial point in this solution is the need to deliver a good approximation in some cases, e.g. in the simulation of deformable objects, if a tool and an object are colliding an immediate update has to be done otherwise the user sees the instrument penetrating the deformable object. Figure 1 shows the block diagram of the basic system components. The system is divided into four parts rendering, generic simulation, specialized simulation engines and communication.

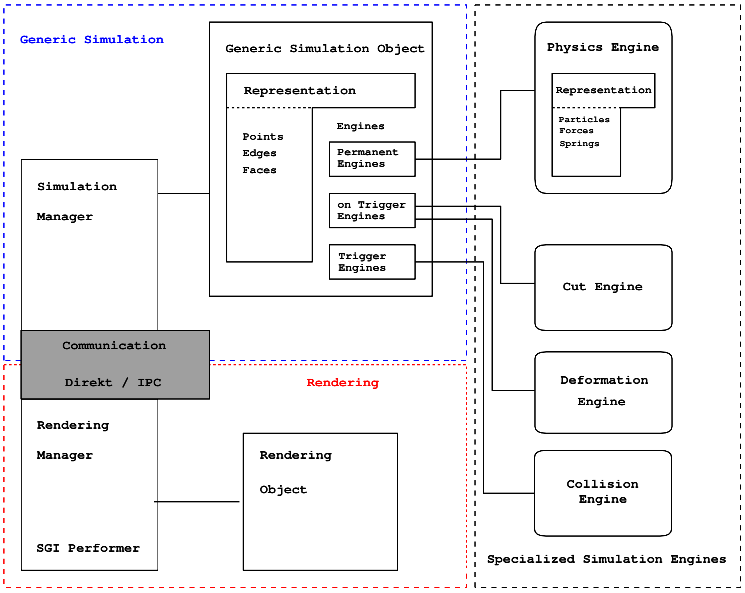

Figure 1: System Architecture

The simulation and the rendering parts are separated, both parts contain a manager and the objects in different representations, where each representation is specialized for the different requirements and each manager is storing the objects he is controlling. The simulation and rendering parts are connected via the communication channels. This approach offers the advantages that either part could be changed without modifying the other one and that they could work in a distributed environment. The last block, the spezialized simulation engines, is realizing the simulation functionality and the extendability. Each engine implements a different aspect of the simulation, e.g. collision or deformation. Like shown in figure 1 these engines work either on the given topology representation, points, edges and faces, or on own data structures, which extend the default ones, e.g. particles and springs.

2.2 Deformation

The deformation operation modifies the shape of an object. During this shape modification the object topology is not changed in a permanent way. The approach described in this section introduces new elements to the topology, but those elements are only used as long as the object undergoes the deformation. After the object has its normal shape the temporal elements are deleted restoring the original topology.

A spring mass sytem is used to simulate the dynamic object behaviour. These systems are widely used in physical simulation systems because their calculation complexity is small enough to allow realtime simulations and there exist well known numerical methods to solve the involved ordinary differential equations, see [1], [2] or [10] for details. The spring mass system is realized as an permanent running engine, called “Physical Engine”. To approximate the tool form in a better way no original particle is moved after a collision was detected. Instead new particles are created on the fly using the following templates.

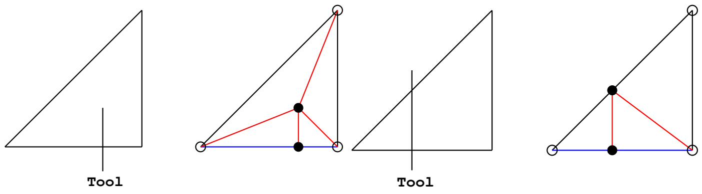

Figure 2: Deformation Templates

If one edge is active two new particles (filled circles) are created, four new springs connect the central new particle with the remaining vertices, including the second new particle. The active edge is splitted and two new springs connecting two old with the new particle. In the case of two active edges two particles are generated, one on each active edge. As before four new springs are generated preserving the triangulation. There are two occurring special cases. They occur if the two new particles coincide. In this case some triangle degenerate to lines. To solve this problem only one new particle is generated and connected. The inactive elements are marked and not deleted, in the case getting back to the original face shape they are reactivated and the new elements are deleted as before.

If only one active face is present the following templates are used, depending on the number of instrument vertices inside the object. For each vertex inside the object a new particle is generated. If one instrument vertex is inside the object, gray circle figure 3 left, the new particle is connected to the old particles as shown in figure 3 left.

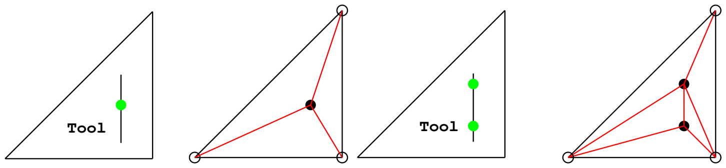

Figure 3: Deformation Templates, One Active Face

In the case of two vertices inside the object two new particles are generated. To preserve the surface triangulation six new springs are generated. The new springs are attached to the old and new particles as shown in 3 right. Also in this case a better approximation of the tool is achieved by the generation of new particles.

The deformation using a tool is realized in the follwing way: After the collision detection determined the active faces the direct affected particles are displaced by a calculated amount and marked fixed. Then a simulation step is executed to adjust the remaining particles and before returning to the main loop the marked ones are released. If, in one of the following simulation steps, a collision between the new faces and the tool occurs , no new elements are generated, instead the current ones are displaced as if the collision occurred with the old face. Also the used template is changed if needed.

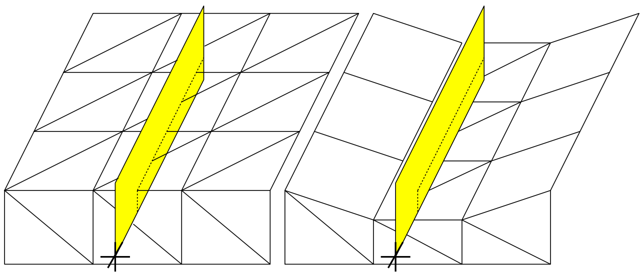

Figure 4: Deformation Point Movement, Old

Figure 4 is showing the deformation results realized in [2], all points belonging to active faces were moved in the way, that an interscection between tool and object no longer occurs. The result using the additional masses and springs is shown in figure 5.

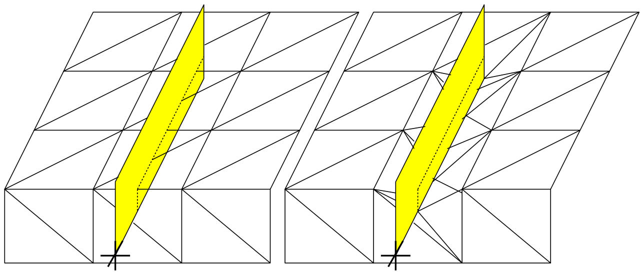

Figure 5: Deformation Point Movement, New

2.3 Cutting

In contrast to deformation the cutting operation is changing the object topology in a permanent way. It affects the rendering as well as the simulation databases, as it deletes and adds elements. Since the results are highly concave objects the simulation complexity is increasing, adding such problems as self-intersection. A second result is the splitting of one object into two parts, where one maybe become freely movable instead of being fixed. This free part is changing its response to tool interactions from deformation to rigid motion.

There are two major types of cutting instruments, knives and scissors. The main difference between those two types is the number of blades they have. Knives usually have one blade divided into different areas, a sharp and a blunt one. Scissors are built out of two blades, in the following sections scissors with U-form blades are called medical scissors.

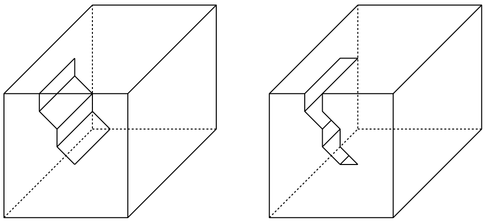

Figure 6: Knife Cutting Sequences

A knife cut is the result of a sequence of smaller cuts over time. Figure 6 is showing such a sequence of discrete cuts, each of the four planes is the result of a single cutting operation. All four together form the cut shown in the right half of figure 6. This cut motion could have been stopped and continued, for example in a first motion the knife could create the first two planes, resting for a period of time and than create the last two. The result of a cutting sequence could be a cut in, see figure 6, or a complete cut dividing the object into two parts, see figure 7.

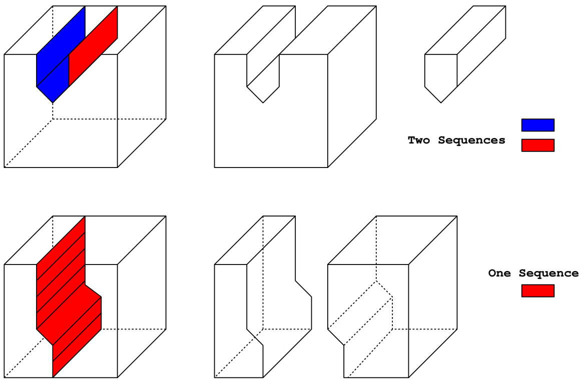

Figure 7: Different Complete Cutting Sequences

The situations in figure 7 differ from each other in the number of sequences needed to create the complete cut. In situation one two sequences with different starting points are used to create the splitting situation. These sequences are not related to each other in any way, their execution could be right after another or with other actions in between. In the second case only one sequence is used, but as seen above also this one sequence could be interrupted. As a result of the different occurring points in time of the sequences the system has to provide a mechanism to recognize this situations.

The cutting algorithm is based on the usage of different templates to create the structures generated by the cutting operation of a knife like tool. After being triggered by the collision detection the cutting algorithm takes the following steps to determine the needed templates. In a first step the algorithm decides if the cutting threshold is exceeded. This threshold represents the border between deformation and cutting. The value could either be an object specific penetration depth or a force exerted by the tool. After finding the appropriate templates the old object surface is changed in two steps. First the old active surface faces are splitted to simulate the incision, after that the structures inside the old object are created to ensure a closed surface. Figure 8 is showing a general cross section of the cutting operation. The parts marked “new side faces” and “new bottom face” are the structures generated inside the old object, they will be refered as “inside or subsurface structures” in the following. As shown in this figure the bottom faces are not placed beneath the tools tip, instead these bottom faces undergo a deformation in the following simulation step to adapt the tools shape .

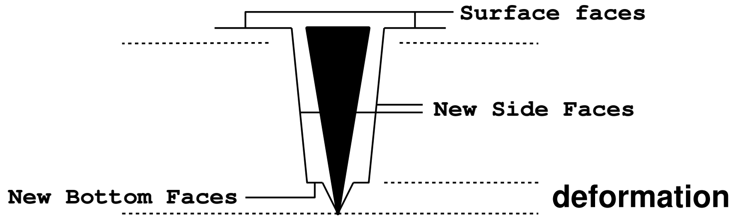

Figure 8: Cutting Cross Section

The decision which template is used for the various active faces is depending on the angle between the tool and face normal and the number of active edges. After the templates are choosen the algorithm calculates the needed vertices like illustrated in figure 9 in a general case. The input data, intersection point between the face edge and tool, used by this calculation is generated by the collision engine. Using the two system constants $open_{top}$ for the top and $open_{bottom}$ for the bottom width of the opening, the opening vectors are calculated.

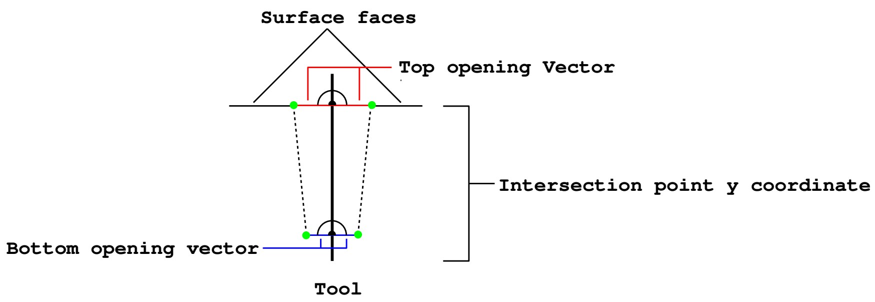

Figure 9: Cutting Opening Outline

In the following the different templates used to create the cuts are presented. The first three groups of templates are the ones used if a new sequence is started. Each of these groups has specific templates depending on the number of active eges. The last group of templates is used if an old cutting sequence is continued. Each of the templates choosen if a new cut is started contains two parts. One part describes how to split the active face to preserve the overall triangulation, the other defines the new subsurface structures.

If one edge is active, figure 10 left, the active face is replaced by four new faces in the following way. The active edge is splitted at the intersection creating two new vertices and a third vertex is created at the tool tip. The triangle generated by the new vertices forms the incision of the instrument and is removed. The remaining area of the old face is triangulated using the new and old vertices.

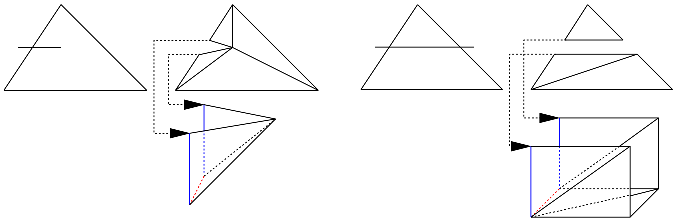

Figure 10: Cutting Templates, Normal

In the case of two active edges, figure 10 right, three faces replace the active one. This time the old face is separated at both intersection points on the active edges and four new vertices are created. This separation forms two not connected areas, one above and one below the tool. The subsurface structure has a U-form shape where each of the three sides is built out of two triangles. The two bottom triangles are called "bottom faces". Figure 10 is showing the active faces with the tool as a line and the resulting faces after splitting operation. Underneath these faces the subsurface structures are shown and the two arrows indicate how to place them.

In the special case of 90 degrees (front face figure 11) two modifications are made. The first is that no subsurface structures are generated in both active edge situations. The second change is that in this case the subsurface structures of the neighbouring active faces are changed to achieve a closed surface. In Figure 11 the effect on the neighbouring subsurface structure is illustrated. In the one active edge case the U-form of the adjoining structure is changed on the left side to adjust this side to the triangluar shape of the active face incision. Figure 11 right is showing the combination of a one and a two active edge case. As before the right side of the neigbouring U-form structure is modified.

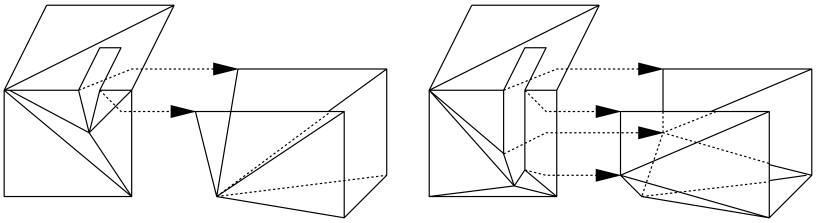

Figure 11: Cutting Templates, Special Cases, 90 degrees

Figure 12 is showing all possible situations before the cutting is done, the line on the surface illustrates the upcomming cut.

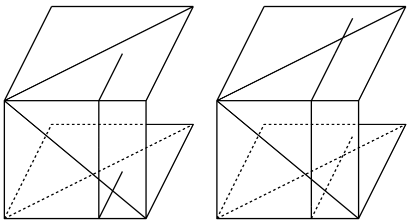

Figure 12: Cutting, Special Cases, greater than 90 degrees

If the normal templates would be applied to the top and bottom active faces the following problems would occur. First the new side faces would be generated twice, once for the top, once for the bottom active faces. The second error would be the position of the bottom vertices. They would be below respectively above the active faces. To avoid this no subsurface structures for the active bottom faces are created. The position of the bottom vertices for the new side faces, create by the active top face template, are displaced to coincide with vertices on the bottom side of the object. These object vertices could be old ones or new ones, generated by the face splitting.

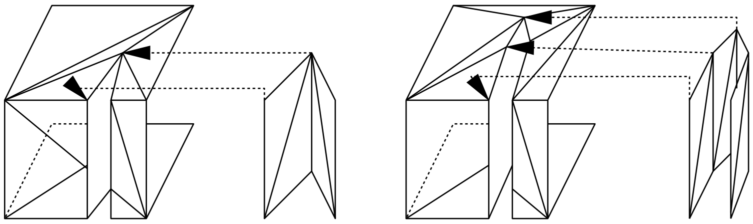

Figure 13: Cutting Templates, Special Cases, greater than 90 degrees

If the instrument is able to cut a single face, no active edges occur. In this situation the templates applied is depend on the number of instrument vertices inside the object. The surface subdivision is the same for both templates.

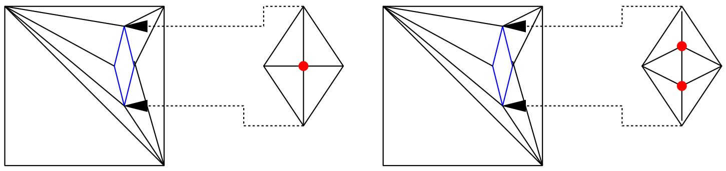

Figure 14: Cutting Templates, Normal

Besides the particles needed by the face subdivision, a new particle is created for each vertex inside the object. As these new particles are lying inside the object they form the vertices of the subsurface structures. Figure 14 left illustrates the structures for the one vertex case, figure 14 right the situation if two vertices are inside the object. To reduce the amount of newly generated triangles, cuts of the bottom faces, representing a continuation of a previous cut are treated differently. The old bottom faces are discarded and the sidefaces are adjusted to match the current cut.

3 Results

The main task was the development of a cutting algorithm for virtual objects in medical applications. Besides the cutting operation object specific tissue properties should be simulated. To achieve this a generic simulation architecture was developed. This resulting architecture has the following advantages. The simulation could be extended, changed or improved in an easy way. To change or extend the simulation new engines could be integrated using the old or a new representation without modifying the remaining parts. The algorithms of the various engines could be changed, either to incorporate latest research results or to utilize new hardware available, and improve the power of the simulation in this way. Besides the reusability and extendability this architecture is able to utilize the advantages of multiprocessor machines or network environments to improve the simulation possibilities. If the simulations of the various objects are independent from each other the simulations could be distributed to different processors or machines.

The drawbacks of this architecture are that the objects are stored multiple times, each process needs it's own copy and that some computation time is lost while synchronizing the databases.

The introduced cutting algorithm is based on templates, each time the needed templates are determined and applied. The advantages of this approach are that only local information is needed to execute the cut and that only a small number of choices has to be made to find the approbiate templates. While executing the cut only the current active face list has to be processed to find the needed template, which reduces the amount of data touched and only a few criterions, in most cases the angle and the number of active edges, are used to determine the ocurring case.

Overall the simulation system developed during this work is capable of doing realtime cuts in combination with a physical based object model. The architecture could be extended providing a basis for future developments either the in area of object simulation or different cutting instruments. The following figures are showing some simulation results. Figure 15 is showing the new elements inserted during the deformation process and the result of cutting operation.

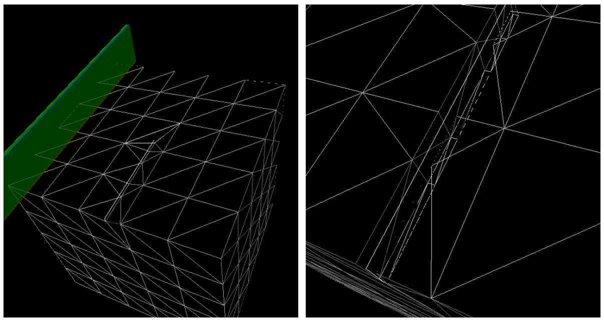

Figure 15: Deformation and Cutting Examples

References

[1] David Barraf and Andrew Witkin. Physically Based Modeling: Principles and Practice. SIGGRAPH'97 Course Notes, 1997.

[2] Ulrich Bockholt. Deformierung von anatomischen Strukturen in virtuellen Umgebungen. Master's thesis, Fachbereich Mathematik/Informatik, Johannes Gutenberg-Universität Mainz, 1997.

[3] Grigore C. Budera. Virtual reality and robotics in medicine. 1997.

[4] Oliver Deussen, Leif Kobbelt, and Peter Töke. Using Simulated Annealing to Obtain Go od Nodal Approximations of Deformable Bodies. Eurographics Workshop on Simulation and Animation'95, 1995.

[5] Oliver Deussen and Christian Kuhn. Echtzeitsimulation deformierbarer Objecte über nodale Modelle. ASIM Mitteilungen, 1995.

[6] Christian Kuhn, Uwe Kühnapfel, and Oliver Deussen. Echtzeitsimulation deformierbarer Objekte zur Ausbildungsunterstützung in der Minimal-Invasive Chirurgie. MVD'95, 1995.

[7] A. Mazura and S. Seifert. Virtual Cutting in Medical Data. Medicine Meets Virtual Reality, 1997.

[8] Dimitri Metaxas and Demetri Terzopoulos. Dynamic Deformation of Solid Primitives with Constraints. SIGGRAPH'92, 1992.

[9] Wolfgang Müller, Rolf Ziegler, André Bauer, and Edgar H. Soldner. Virtual Reality in Surgical Arthroscopic Training. Journal of Image Guided Surgery, 1995.

[10] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. Numerical Recipes in C. Cambridge University Press, 1992.

[11] Karl D. Reinig, Charles G. Rush, Helen L. Pelster, Victor M. Spitzer, and James A. Heath. Real-Time Visually and Haptically Accurate Surgical Simulation. Healthcare in the Information Age, 1996.

[12] R.M. Satava. Telesurgery, virtual reality and the new world order of medicine. Proc. VR World, 1995.

[13] Gyeong-Jae Song and Narender P. Reddy. Tissue Cutting in Virtual Environments. Interactive Technologie and the New Paradigm for Healthcare, 1995.

[14] Naoki Suzuki, Asaki Hattori, Shinya Kai, Takeshi Ezumi, and Akihiro Takatsu. Surgical Planning System for Soft Tissues Using Virtual Reality. Medicine Meets Virtual Reality, 1997.