Scott L. Delp
MusculoGraphics, Inc.
Evanston, IL
and
The Rehabilitation Institute of Chicago
Chicago, IL 60611
delp@nwu.edu

Peter Loan
MusculoGraphics, Inc.
Evanston, IL

Cagatay Basdogan
Research Laboratory of Electronics
M.I.T.
Cambridge, MA

Joseph M. Rosen
Dartmouth-Hitchcock Medical Center
Lebanon, NH

Surgical Simulation: An Emerging Technology for Training in Emergency Medicine

Abstract

The current methods of training medical personnel to provide emergency medical care have several important shortcomings. For example, in the training of wound debridement techniques, animal models are used to gain experience treating traumatic injuries. We propose an alternative approach by creating a three-dimensional, interactive computer model of the human body that can be used within a virtual environment to learn and practice wound debridement techniques and Advanced Trauma Life Support (ATLS) procedures. As a first step, we have developed a computer model that represents the anatomy and physiology of a normal and injured lower limb. When visualized and manipulated in a virtual environment, this computer model will reduce the need for animals in the training of trauma management and potentially provide a superior training experience. This article describes the development choices that were made in implementing the preliminary system and the challenges that must be met to create an effective medical training environment.

1 Introduction

Highly trained personnel are an essential element of effective emergency medical care. The routine use of animal models in the training of medical personnel to treat traumatic injuries, however, has several important limitations. The use of animals is expensive and does not represent human anatomy. In the teaching of wound management techniques, usually only one injury can be studied per animal, and the training experience is very limited in duration. We believe that a simulation system that accurately represents human anatomy, models the physiology of many different traumatic injuries, and allows for extensive practice of emergency medical techniques will provide a more effective training experience.

A number of surgical simulators have been developed to help users learn anatomy and practice medical procedures (Rosen et al., 1996; Ota et al., 1995; Hunter et al., 1993). For example, Satava (1993) developed one of the first surgical simulators, which included a computer model of abdominal anatomy that was viewed in three dimensions with a head-mounted display. Pieper et al. (1995) created a system to simulate facial reconstructions. The key element of this system was a graphics-based finite-element model that estimated the biomechanical consequences of various surgical reconstructions. Stredney et al. (1995) developed a virtual environment for learning epidural anesthesia. This system incorporated a simple force-feedback device to facilitate the develop-

Presence, *Vol. 6, No. 2, April 1997, 147–159*

ment of manual skills. Langrana et al. (1994) recently developed a model of the knee that included force feedback. Working in collaboration with the NASA Ames Research Center, our group developed the first musculoskeletal simulation for a virtual environment (Pieper et al., 1991). This system not only provided a stereo view of the anatomy, but also allowed the user to practice surgical manipulations and to estimate the functional consequences of the simulated surgeries.

Several factors have changed in the last few years that now make the use of medical simulations applicable on a much broader scale. First, the price of computer graphics systems has decreased dramatically while performance has increased by several orders of magnitude. Second, expertise and software now exist to create highly accurate anatomical models from medical image data (Chan et al., 1991; Robb and Hanson, 1991). Third, a number of important clinical problems have been solved with surgical simulations, and these solutions have been published in the medical literature (Kawabata et al., 1989; Altobelli et al., 1993; Delp et al., 1990). As a result, there is broader acceptance of surgical simulation technology and greater confidence in the fidelity of computer models of the body (see Satava, 1995, and Dumay, 1995, for review).

We are developing a surgical simulator to facilitate the teaching of trauma and life support techniques. The long-term goal of our work is to demonstrate that computer simulation of traumatic injuries can be used to effectively train medical personnel in injury assessment and management for both civilian and military applications. As a first step, we have focused on the development of computer graphics representations of traumatic injuries of the lower-limb. We have also produced virtual surgical instruments that allow trainees to interact with simulated injuries to practice wound assessment and management. The focus of this proof-of-concept system is the assessment and debridement of a gunshot wound to the thigh.

This paper describes the development of the lower-limb model and the virtual surgical instruments for our proof-of-concept system. The first step in this process was to create a high-resolution three-dimensional (3D) model of a healthy thigh. An injury was introduced to this anatomy so that the trainee can interact with the affected structures to assess and manage the trauma. A means to evaluate the effect of the injury and the subsequent repair procedures upon the musculoskeletal and circulatory systems has been implemented. Although a single training scenario is described here, the algorithms and software created for this simulation can be adapted to simulate a variety of musculoskeletal injuries and to simulate other procedures, such as the Advanced Trauma Life Support (ATLS) techniques.

2 Lower-Limb Model

We have used the data collected in the Visible Human Project (Ackerman, 1994), which is being conducted by the National Library of Medicine, as a basis for our lower-limb model. We believe that use of this readily accessible data set will facilitate the sharing of models with other simulation developers and the integration of our technology into other simulation environments.

2.1 Healthy Thigh Model

The Visible Human Project data were used to build a 3D model of the thigh. These data were created from a healthy male cadaver using CT and MR imaging, as well as digital photographs. To create the photographs, the body was frozen, fixed in a gelatin, and sliced in the transverse plane at 1.0 mm intervals. Each slice was photographed in color and then digitized at an image size of 2048 × 1216 pixels, thus providing a resolution of 0.33 mm × 0.33 mm. Each pixel contained three bytes of data, one each for the red, green, and blue components. We downloaded the digitized photograph data for the thigh region (22 centimeters long), and processed the images to reduce the size of the data file. We cropped the images so they contained only images of the right thigh, compressed the data for each pixel into one byte of grayscale information, and then saved every other slice to an output file. The resulting data file contained 110 images (spaced every 2 mm along the long axis of the thigh), and each image was 656 × 700 pixels.

This file was loaded into our PAtient Modeling Soft-

ware (PAMS) for segmentation of the two-dimensional (2D) images and construction of the 3D tissue surfaces. PAMS is a general-purpose package that performs several important functions needed to create 3D anatomical models, including locating boundaries of anatomical structures using an edge filter and constructing 3D representations of structures using Bézier surface patches.

For each anatomical structure (skin, fat, bone, muscle, tendon, nerve, artery, vein), we determined the range of slices in which the object appeared. For each of these slices, the object's shape was outlined using a semiautomated process. This process employed a Canny edge filter (Canny, 1986) and a contour snaking algorithm (Kass et al., 1988) to aid the user in finding the edges of the object. Once this had been done for all of the slices, the contour data were combined into a 3D mesh of data points, from which a set of Bézier surface patches was generated. The surface patches were then tessellated to form a polygonal surface representing the anatomical structure.

To make the images of the 3D thigh model appear more realistic, we created texture maps for each of the tissue types in the model (cutaneous, subcutaneous, muscle, nerve, cortical bone, cancellous bone, artery, vein, devitalized tissues). These texture maps are 2D pictures that are wrapped over the polygonal surfaces that define each tissue. Figure 1 illustrates the healthy thigh model. The cortical and cancellous bone of the femur can be seen near the center of the cross-section. The surfaces of the muscles have been modeled and are displayed using texture maps.

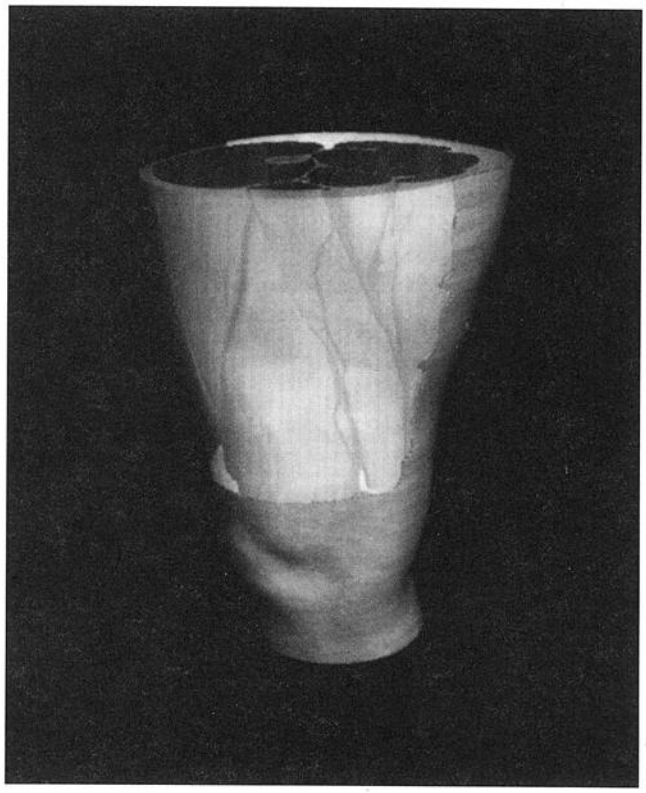

Figure 1. *Thigh model created from the Visible Human Project data. The 3D surfaces of the skin, fat, muscles, bones, nerves, arteries, and veins were reconstructed from the transverse cryosection data. Photographs of the various tissue types were scanned in to create texture maps that are mapped onto the polygonal surfaces. The skin and fat have been rendered partially transparent so that the other structures are more readily visible.*

2.2 Injured Thigh Model

To accurately model the geometry of a penetrating injury, the shapes of the permanent and temporary cavities must be considered. The shapes of these cavities were approximated based on bullet parameters (e.g., mass, speed, trajectory) and tissue viscoelastic properties. In the prototype system, we modeled the injury that results from a high-velocity gunshot wound to the distal thigh. The 4.0 g, 5.5 mm diameter bullet was assumed to enter the anterior aspect of the thigh at a speed of approximately 500 m/s. The thigh was assumed to be in the upright standing position. After hitting the femur, the bullet exited the posterior/lateral section of the thigh. Because several bone fragments broke from the femur and were propelled posteriorly, there was extensive damage to the back of the thigh, especially to the biceps femoris muscle and its associated nerve and vascular supplies.

As a bullet passes through the thigh, it lacerates, crushes, and burns the tissue in its path. The injuries caused in this way define the permanent wound cavity. To determine the geometry of the permanent cavity, we examined the relevant literature (Fackler, 1988; Hollerman et al., 1990; Zajtchuk, 1990). We then used 3D modeling software to build a polygonal surface with ap-

propriate size and shape for the thigh model and the chosen bullet parameters. The beginning of the wound tract is cylindrical (approximately 1 cm in diameter). When the bullet hits the lateral edge of the femur, it is assumed to break the bone, and deflect approximately 15 degrees laterally. The wound tract behind the femur is roughly cone-shaped, with several large spikes representing the tissue destroyed by bone fragment projectiles.

Further damage is caused by the transfer of kinetic energy from the bullet to the surrounding tissue. This energy transfer stretches and devitalizes some tissue outside of the permanent cavity. This region of devitalized tissue is known as the temporary cavity. To approximate the geometry of the temporary cavity, we performed nonlinear scaling operations on the permanent cavity and increased the average diameter of the cavity. We also smoothed the surface of the temporary cavity to match the images of wound cavities that are described in reference texts (U.S. Dept. of Health and Human Services CCC Guidelines, 1991; Fackler, 1988; Hollerman et al., 1990).

The process of creating a three-dimensional model of the injured thigh begins with a healthy thigh model, consisting of closed 3D polygonal surfaces of each structure in the thigh, as described above. The injury model consists of two closed 3D surfaces, representing the permanent and temporary cavities of the wound. Assuming that the tissue in the permanent cavity is expelled from the thigh, we determined which portions of the thigh structures were within the permanent cavity, and performed a boolean subtraction operation to remove them from the model. We then calculated which portions of the remaining structures were within the temporary cavity (i.e., the region of devitalized tissue) using a boolean intersection operation. If the temporary cavity intersected the thigh, the intersecting portion of the thigh structure was labeled as nonviable by marking the vertices and polygons within that region. The portion of the structure outside of the temporary cavity was labeled as viable. The nonviable portion of the structure was then rendered using a darker color and altered texture map to indicate its devitalized state.

In our initial simulation, four bone fragments were created by determining the intersection of the permanent cavity and the femur. The resulting piece of bone was divided into four smaller pieces of roughly equal size. One fragment was assumed to have been expelled from the thigh; the other three were embedded in the thigh model, at the ends of the spikes in the permanent cavity. For future surgical simulations, a library of traumatic injuries will be created using software being developed by Mission Research Corporation (Costa Mesa, CA). This software calculates the sizes and positions of bone fragments, as well as the shapes of the permanent and temporary cavities, given the parameters of the projectile and the properties of the anatomical structures (Eisler et al., 1996). The accuracy with which this software determines injury geometry will be tested by comparing the simulated injuries to cadaver studies and wound geometry documented in autopsies. We will then integrate the results of the injury-mechanism simulations with the software for assessing and treating the wounds (Eisler and Delp, 1994).

2.3 Functional Consequences of Injury

We have evaluated the functional consequences of the injury by computing how it affects the musculoskeletal and circulatory systems. To determine the consequences of the injury to the musculoskeletal system, we calculated muscle strengths before and after the injury. Strength is quantified as the maximum moment-generating capacity of each muscle group, based on the muscles' physiologic cross-sectional areas, fiber lengths, pennation angles, and moment arms. To make these calculations, we used a model of the lower limb that represents the force-generating properties of 43 muscles (Delp et al., 1990; Delp and Loan, 1995). To evaluate the functional consequences of the injury, we compared muscle strengths after the injury to the strengths required for walking. This analysis provides an estimate of the short-term consequences of the injury and the long-term disability of the simulated patient. The postoperative musculoskeletal function can also be estimated as a measure of the success of the simulated surgery.

We have also analyzed the effects of the injury on the circulatory system. To simulate the treatment process, a model of the circulatory effects of the injury is needed because these effects alter the simulated patient's vital signs and the bleeding of the tissues. Our circulatory model calculates the time-dependent changes of several key hemodynamic properties, such as blood loss, heart rate, blood pressure, and cardiac output. The model is based on theoretical fluid flow models and on experimental results published in the literature (Shen et al., 1990; Zamir et al., 1992; Raines et al., 1974). As the user interacts with the wound using the virtual surgical instruments (e.g., scalpel, forceps), the patient's tissues must bleed or stop bleeding as appropriate. The corresponding changes in blood volume affect the vital signs, which are updated and displayed.

The circulatory model has two components. The first component is a mathematical model governing the flow of fluid through a network of closed conduits (Zamir et al., 1992; Raines et al., 1974). These equations calculate the steady-state flow of blood through the arteries, veins, and capillaries. The flow must satisfy the basic principles of the conservation of mass and the conservation of energy. The second component is a set of equations describing the transient response of several hemodynamic properties as blood is lost from the system. The key properties that must be computed are cardiac output, heart rate, and arterial pressure. Other properties that affect the calculations are peripheral resistance and stroke volume. We used the results of experimental studies on animals (Shen et al., 1990) to generate the relations between these properties and the total amount of blood loss in our human physiology model.

3 Injury Assessment and Interaction

We have developed a 3D graphical interface to the injured thigh model. The graphical interface allows the user to assess and treat the injury with a set of virtual surgical instruments, such as scalpel and forceps. We have implemented the virtual surgical instruments with the use of industry-standard development tools to achieve some level of computer-platform independence. We have chosen to use the C++ programming language and the Open Inventor 3D software toolkit. C++ is an object-oriented language whose constructs are useful for manipulating hierarchical databases such as the thigh model. Open Inventor is an object-oriented 3D graphics toolkit developed by Silicon Graphics, Inc. (Mountain View, CA). It allows predefined object classes, interactive manipulation of objects, and high-performance picking and rendering. It also uses a standard file format for storage of 3D databases, which will facilitate data exchange with other software systems. The system is implemented on a Silicon Graphics Indigo2 computer workstation.

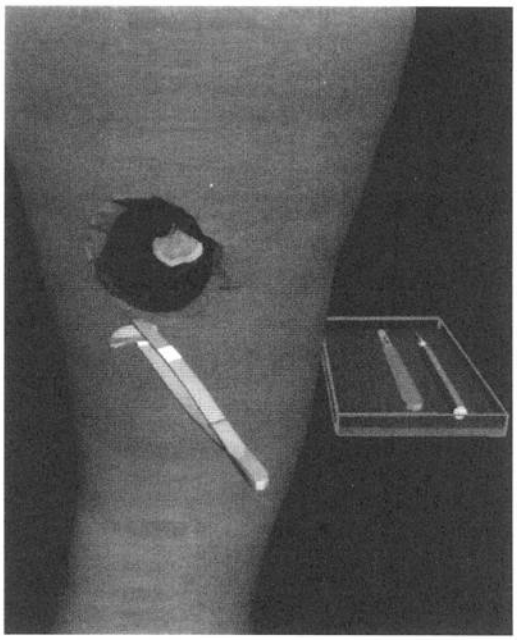

Figure 2. *Virtual surgical instruments interacting with the thigh model. The scalpel and wound probe are shown on the instrument tray on the left. The forceps are being controlled by a six degree-of-freedom input device (not shown) to grab a bone fragment and remove it from the exit wound, as part of the wound debridement process. The dark tissue surrounding the wound is the nonviable tissue.*

The user interacts with the model by selecting instruments from a virtual surgical tray (Figure 2) and using them to perform activities such as realigning fractured bones, viewing the color and assessing the contractility

of muscle, debriding the wound, and repairing blood vessels.

3.1 Physically Based Algorithms

To simulate the surgical activities mentioned above, the deformation of the anatomical models must be represented with algorithms governing the mechanical and physiological behaviors of the tissues. These physically based algorithms define how the simulated tissues respond to manipulation by the virtual surgical instruments. This interaction consists of three basic types of behavior: tissue deformation, cutting, and bleeding.

3.1.1 Tissue Deformation. The soft tissues in the anatomical model can be deformed either by pushing or pulling. In both cases, a region of the tissue surface near the surgical instrument is deformed to represent the effect of the instrument. The size of the region that is deformed depends on the material properties of the tissue and the amount of deformation depends on the location of the instrument.

The deformation algorithm works in three steps. The first step calculates the contact point between the surgical instrument and the tissue model. This is done by finding the intersection between a simple geometric model of the instrument and the surface model of the tissue. For example, the instrument may be represented by a single line segment along the axis of the device. The intersection of the line segment and the surface model of the tissue defines a single contact point. The second step determines which vertices in the tissue's surface model should be deformed. All vertices within a certain distance of the contact point are assumed to be deformed. This distance is known as the *radius of deformation,* and its value depends on the material properties of the tissue. The third step calculates the deformation vector for each of the vertices. This vector depends on the location of the surgical instrument and the distance between the vertex and the contact point. The location and the distance are used to scale and interpolate a second-order polynomial that smoothly represents the deformation region.

To prevent the user from pushing or pulling the tissues in the model farther than is realistically possible, we have imposed limits on the amount of deformation that can be achieved. For each type of tissue in the model, we set a maximum distance for tissue pushing and for tissue pulling. If the user attempts to deform the tissue beyond these limits, the software decouples the input device from the virtual surgical instrument so that it cannot be moved outside this range.

The deformation algorithm assumes that the tip of the instrument contacts the tissue. This assumption is reasonable for most uses of the instruments, but may not hold in certain cases, depending on the surgical procedure being simulated. To handle these cases, we have devised enhancements to the deformation algorithm that we will implement as needed for specific procedure simulations. For example, if some region of the instrument other than the tip contacts the tissue, a point within this region should be used as the effective instrument tip. This point is determined by calculating the point on the surface of the instrument that is the furthest inside the tissue structure. This method works well for all of the instrument shapes that we have modeled thus far.

Object grabbing is a special case of deformation. Grabbing involves moving an entire object when it is gripped with a surgical instrument, such as a hemostat or forceps. When the user grips an object with the forceps, the object is moved rather than deformed if one or more of the following conditions is true: (1) the object is labeled as a hard substance (e.g., bone fragment, shrapnel), (2) the object is not connected to any other object (e.g., an excised piece of skin), or (3) all of the object's vertices are within the radius of deformation (e.g., a small piece of devitalized muscle tissue). When the surgical instrument contacts an object in the model, each of these conditions is checked; if one or more are true, the object-grabbing subroutine is called to handle the interaction. If none is true, the tissue-deformation subroutine is called.

3.1.2 Tissue Cutting. To perform tissue cutting, we have implemented boolean subtraction and boolean intersection algorithms. These algorithms operate on

two closed polyhedra of arbitrary shape. The polyhedra can contain any number of vertices and planar polygons, and can be convex or concave. Both algorithms begin by calculating the intersections of each polygon in the first polyhedron (P_1) with the polygons in the second polyhedron (P_2). In the general case, a polygon either lies completely inside the other polyhedron, completely outside the other polyhedron, or it intersects it and is divided into two or more pieces.

The next step in the process is to build the output polyhedron by collecting the polygons and polygon pieces computed in the first step. The specific set of polygons collected depends on which boolean operation is being performed. The subtraction algorithm, which computes $P_1 - P_2$, collects all of the polygons and polygon pieces in P_1 that are outside P_2, and all of the polygons in P_2 that are inside P_1 (with normals reversed). This second set of polygons defines the new cut surface of P_1. For boolean intersection, the algorithm collects all of the polygons and pieces from P_1 that are inside P_2, and vice versa.

To implement tissue cutting with boolean subtraction, a *cutting object* must be created (P_2) and then subtracted from the tissue being cut (P_1). The specific method of creating this cutting object depends on the surgical instrument being used, but the basic process is to use the path of the cutting blade[s] to build the polyhedron. As the cutting instrument is moved through the tissue, the coordinates of the blades for each iteration are used as vertices of the cutting object. The blades are modeled as thin objects (approximately 0.5 mm thick) so that cutting does not remove much material from the model.

After the boolean subtraction algorithm is used to perform tissue cutting, the intersection algorithm is used to label the newly created polygons as either healthy or devitalized. This labeling step is needed when the user excises a section of devitalized tissue from the thigh model. As the user cuts away tissue, the new tissue surface must be colored and texture mapped correctly so that the user can determine if the tissue is healthy or not. To do this, the newly created polygons are compared for intersection with the temporary cavity of the penetrating wound. All of the polygons that are inside the cavity are labeled as devitalized.

3.1.3 Tissue Bleeding. We have produced a set of algorithms to simulate the flow of blood over tissue surfaces (surface-flow) and the flow of blood particles discharged from an open vessel (particle-flow). The surface-flow simulation is initialized when the user cuts into soft tissue, such as skin, fascia, or muscle. The flow model simulates the oozing of blood over the tissue surface and the pooling of blood in depressions. The particle-flow algorithms treat the blood as a collection of tiny particles, and calculates their movement based on the physics of projectile motion. The particle-flow model is used to represent pulsatile flow, which occurs when a blood vessel is severed.

To represent blood that flows over a surface and pools at the bottom of a depression, we have created a surface-flow algorithm. Fluid flow is governed by a set of partial differential equations known as the Navier-Stokes equations. Integration of these equations over time is computationally expensive, and thus real-time simulations are not possible. A stable and computationally less expensive technique was developed by Kass and Miller (1990). This method is based on the solution of a simplified form of the Navier-Stokes equations, known as the wave equations. Using this technique, we represent the fluid surface as a 3D grid and solve the wave equations using a finite-differencing scheme to update the fluid height at each node of the grid. The process starts with the following wave equation:

$$\frac{\partial^2 h}{\partial t^2} = gd\left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial y^2}\right) \tag{1}$$

where $h(x, y)$ is the height of the fluid surface at location (x, y) in a global reference frame, $d(x, y)$ is the depth of the fluid at location (x, y) relative to the base (tissue surface), g is the gravitational constant, and t is time. For computational simplification, this equation is solved for fluid height in the x and y directions separately. The resulting fluid height values are then averaged and utilized in the graphical simulations and subsequent numerical iterations. The solution describes the flow of blood over the polyhedral surfaces that represent the anatomical structures and is displayed as a continuous polyhedral surface drawn over the anatomy. Although artifacts occasionally appear as a result of the simplifications, the nu-

merical solutions are stable, and the computations are fast enough to calculate the results in real time.

We have developed a particle-flow algorithm to model pulsatile flow, which occurs when a blood vessel is cut, either fully or partially. The flow calculations use the diameters and connectivity of the vessels, determined from the Visible Human Project data, to determine the flow rate of blood through each vessel. These flow calculations are updated each time step and stored in a blood flow data structure. The display of the blood involves groups of particles that are placed at the vessel outlet; the particles are given an initial velocity in a gravity field, so they shoot out of the vessel with a projectile motion. The particle-flow simulation calculates the speed of the blood particles, their direction of movement, and the number of particles in the flow. The speed of the blood particles is estimated based on the diameter value from a vessel database we developed. This database estimates the flow at any cross section, given the vessel segment and current vital signs. When a cutting instrument contacts a vessel, the contact point is used to find the nearest point in the database that contains diameter and flow information for that vessel.

The direction of flow is also estimated using the flow model and the coordinates of the contact point. If the vessel is severed completely, the particles flow perpendicular to the vessel's cross section at the location of the cut. Otherwise, if the vessel is only partially cut, the particles flow in the direction of the surface normal at the location of the cut. Once the velocity (direction and speed) is known, the position of each particle can be calculated at any time frame using the following formulation, based on the physics of projectile motion:

$$
\begin{aligned}
P^X_{t+\Delta t} &= P^X_t + V^X_{initial} \cdot \Delta t \\
P^Y_{t+\Delta t} &= P^Y_t + V^Y_{initial} \cdot \Delta t - \tfrac{1}{2} \cdot g \cdot \Delta t^2 \\
P^Z_{t+\Delta t} &= P^Z_t + V^Z_{initial} \cdot \Delta t
\end{aligned}
\tag{2}
$$

where P^X, P^Y, P^Z represent the positions in x, y, and z, $V^X_{initial}$, $V^Y_{initial}$, $V^Z_{initial}$ represent the initial velocities of the particles in the global reference frame, g is the gravitational constant, and Δt is the time increment.

The number of particles that are needed to accurately represent the blood flow is estimated from the vessel's cross-sectional area at the location of the cut. We computed the number of particles by dividing the cross-sectional area by the size of the blood particles. The size of the particles in the 3D model is based on their screen size (2 pixels) and was adjusted by trial and error to obtain a realistic appearance of the simulation.

The surface and particle bleeding models are connected by a set of subroutines that are called whenever a tissue is cut. These subroutines determine which bleeding model is appropriate in each instance and call either the surface-flow or particle-flow routines.

3.2 Surgical Instruments

We have developed models of four surgical instruments (scalpel, forceps, hemostat, and wound probe) to interact with the lower-limb model. To model each instrument, we first created 3D geometric descriptions of the instruments using a standard CAD program. The geometric models are loaded into the simulation software and are manipulated by means of 3D input devices. While an instrument is being moved by the user (Figure 3), the software checks for contact between the instrument and one of the objects in the thigh model. When a contact is detected, software implementing the specific behavior of the instrument is called to handle the interaction with the object. This instrument-specific behavior uses portions of the physically based tissue algorithms (Figure 4). For instance, the following describes the four instruments we have modeled and the algorithms associated with each one.

The scalpel is used to cut tissue. Cutting is modeled using the boolean subtraction software. As the scalpel moves through the tissue, its path is used to construct a thin cutting object, which is then subtracted from the model of the tissue. This creates either an object with a thin slice through it, or, if the scalpel passes completely through the tissue, two or more separate objects.

The forceps and hemostat are used to remove small

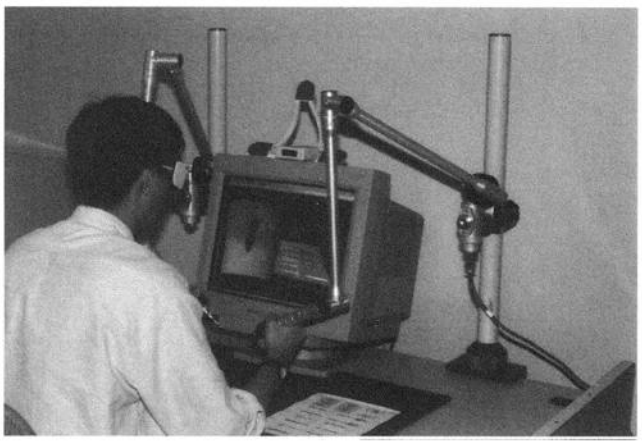

Figure 3. *Photograph of surgical simulation system. The user views the virtual thigh model through stereo glasses synchronized to the computer screen. He manipulates the virtual surgical instruments using six degree-of-freedom input devices. Switches on the devices are used to select a particular instrument, and to perform instrument-specific functions, such as gripping and releasing objects with the forceps.*

objects from the wound (e.g., bone fragments, shrapnel) and to grab and move regions of soft tissue (e.g., lifting a section of muscle to look underneath it). The forceps can be closed by pressing the buttons on the 3D input devices. When the forceps are closed around an object, the three object-grabbing conditions are checked (see above). If any grabbing condition is true, then the entire object can be moved. If not, the deformation algorithm is used to deform the tissue, and the motion-constraint algorithm is used to prevent the forceps from deforming the tissues too much.

Wound probes are straight, rigid implements used to probe the wound tract to determine the trajectory of the projectile and the locations of shrapnel and other debris. They use the tissue deformation algorithm described above, without checking for the special conditions of grabbing. The motion-constraint algorithm is used to prevent the probe from passing through the tissues.

We also plan to implement additional surgical instruments, including scissors, needles, and sutures, in order to simulate other procedures, such as ATLS techniques. The mechanical function of these instruments will be based on the generic behaviors of pushing, pulling, and cutting, as developed for the scalpel, forceps, and wound probe.

4 Discussion

We have developed a working prototype of a surgical simulator to demonstrate the feasibility of interactive simulation for medical training. This prototype incorporates the essential elements that allow us to evaluate the system with medical training personnel. We believe that effective surgical simulators will be developed only through close collaboration between technical and medical personnel. Experienced trauma physicians are using our system to assess and debride wounds and are providing feedback on its effectiveness. Johnston et al. (1996) have developed a protocol for evaluating the effectiveness of a surgical simulator, which will be adapted to provide a quantitative measure of the simulator's effectiveness.

There are several important shortcomings of our system and existing technology that represent challenges for future development. For example, although our system has essential elements of stereo visualization and 3D interaction, it does not include force feedback. We believe this is an essential element of the system and are collaborating with Boston Dynamics, Inc. to develop the next generation system (Figure 5).

However, current limitations in force-feedback technology present a technical barrier to this proposed enhancement. State-of-the-art haptic devices with a sufficiently large work volume have only three or four degrees of freedom of force feedback, so the user would not be able to feel all of the reaction torques as he interacts with the model. We believe that this would be a minor limitation, however, because the surgical procedures that we simulate primarily involve translations of the instruments as they manipulate the tissues (e.g., cutting with a scalpel, pushing with a hemostat). Nevertheless, torque feedback would make the simulations more realistic, and six degree-of-freedom force-torque feedback devices should be incorporated into the training system in the future.

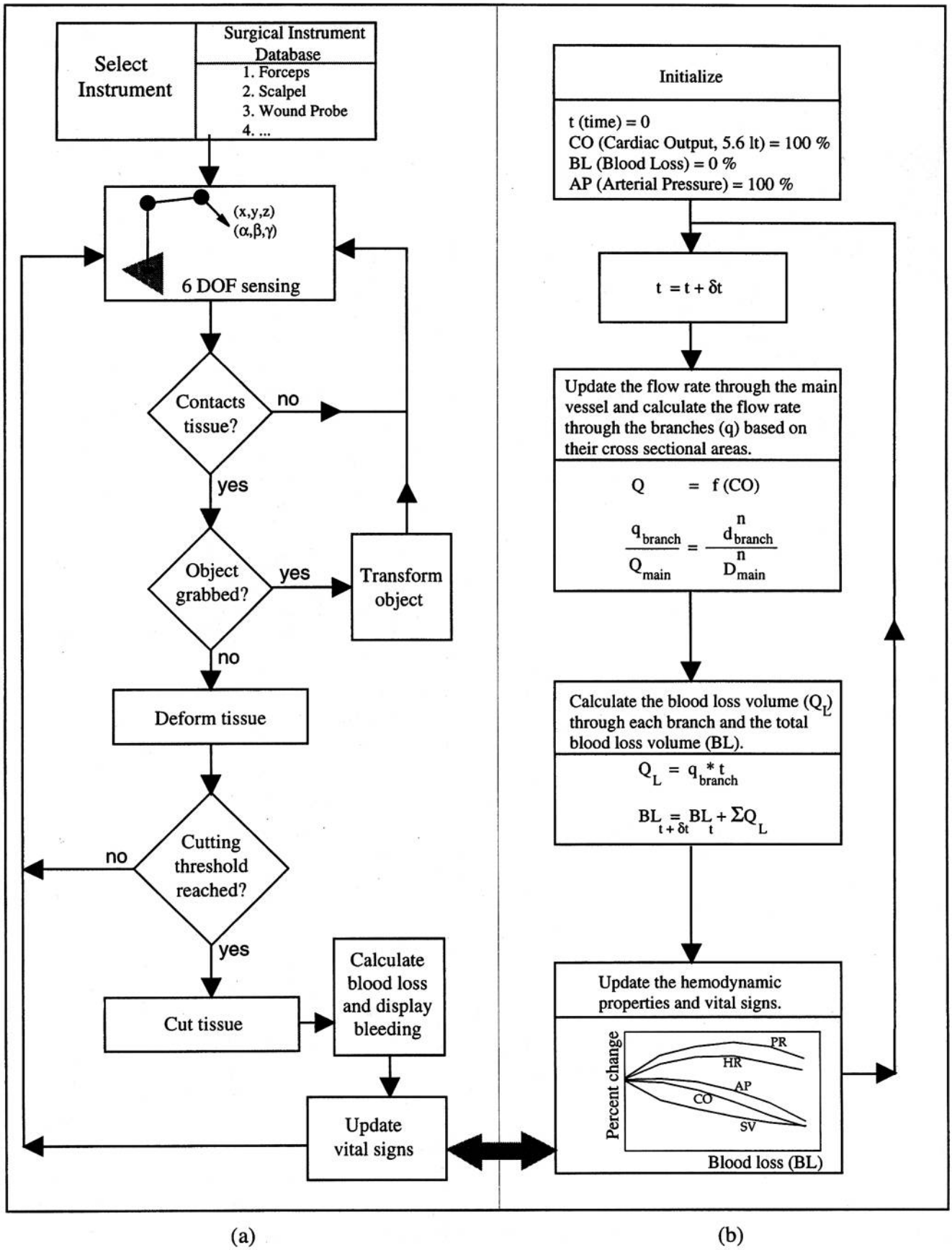

Figure 4. *Procedural flowchart for simulating surgical instrument interaction (a) and calculation of the patient's vital signs (b). In (a), the instrument interaction flowchart, the user selects instruments and manipulates objects using six degree-of-freedom input devices. While the probe moves, contacts with tissues are detected, and the appropriate actions of the instrument are applied. If the tissue bleeds as a result of cutting, the flow of blood is simulated graphically. In (b), the circulatory system flowchart, principles of conservation of mass and energy are utilized to describe the steady-state behavior of blood flowing through the arteries and veins. The transient behavior (due to blood loss) is based on the results of experimental studies. Vital signs and hemodynamic properties, such as heart rate (HR), stroke volume (SV), and peripheral resistance (PR), are updated at each time interval, taking into account the changes due to manipulation of the instruments.*

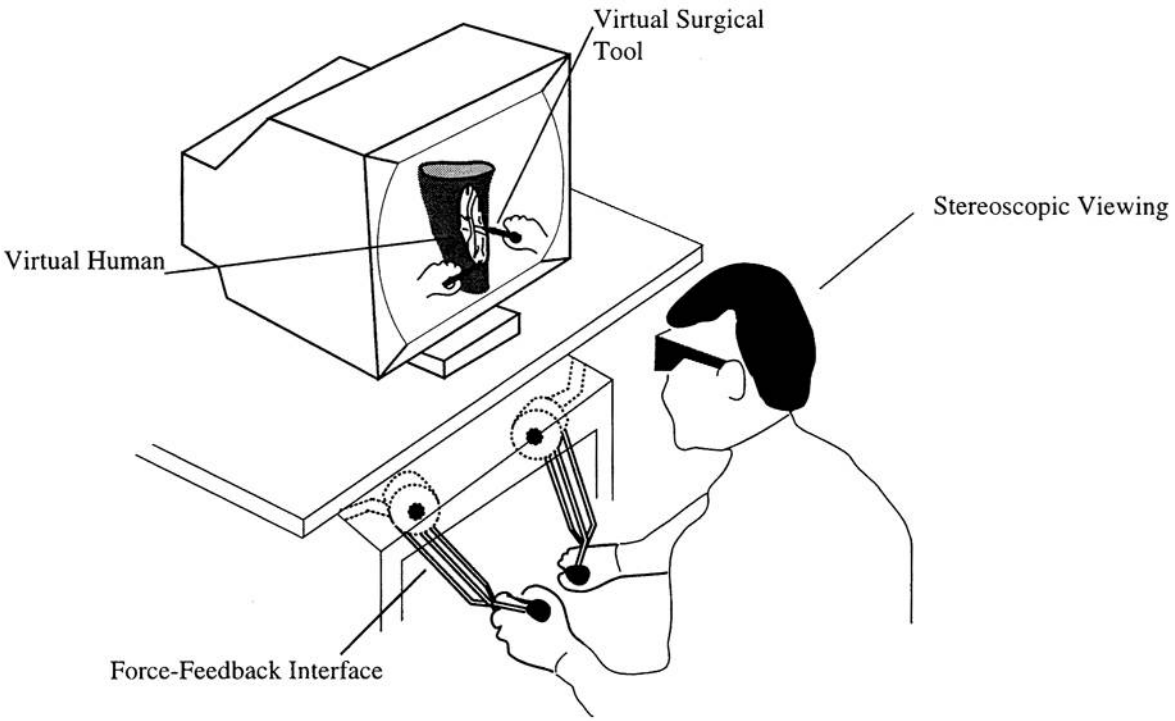

Figure 5. *Schematic of future trauma-management training system that includes force feedback. As in the current system, the user views the virtual body models through stereo glasses synchronized to the computer screen. In the next-generation version, however, the user will interact with the models by inserting surgical instruments into the force-feedback devices and manipulating them. Corresponding virtual instruments will manipulate virtual body models, and appropriate response forces will be calculated and reflected to the user's hands.*

Since one of the major goals of the simulation system is to reduce the need to use animals in the training of medical personnel, the system must be capable of simulating all of the training procedures that involve animals. In addition to debridement of gunshot wounds, a procedure which can no longer be performed on animals, we plan to simulate standard lifesaving techniques, such as those taught in ATLS courses. These procedures (e.g., inserting a chest tube and performing a venous cutdown) are commonly performed on animals, and we plan to add them to the system to create a complete training environment.

However, there are many technical barriers to the development of a complete surgical simulation environment. Some of the ATLS procedures are difficult to simulate (e.g., pericardiocentesis) because they involve deformable medical instruments, dynamic anatomical structures, and a fine sense of touch. Because of their complexity, we have chosen not to simulate these procedures in the first generation system; however, for a comprehensive training system, these procedures must be included in future versions.

Another technical barrier is the graphics performance of computer workstations. Current high-end workstations are capable of displaying complex, texture-mapped scenes at high frame rates. However, workstations with computational and graphics support for real-time calculation and display of blood flowing through a deformable wound during a simulated surgery are just beginning to emerge.

Although there are a number of difficult problems that must be solved to develop a practical training system, the further development of the surgical simulator described here will have several important payoffs. First, it will result in a reduction or elimination of the use of animals in the training of traumatic injury treatment.

Second, the surgical simulator will accurately represent human anatomy, so the user can practice surgical skills on anatomical structures that he will encounter in the field. Also, the user can use the simulator repeatedly, and on different types of injuries and ATLS procedures, in order to gain more experience with the surgical procedures. Third, because the framework we have developed is general, it will facilitate the development of simulators for other procedures. Our physically based algorithms and modeling tools can be used with models of any part of the body, and they can also be incorporated into other simulation environments, such as the telepresence surgery system being developed by SRI (Bowersox et al., 1996; Green et al., 1991) and the SIMNET virtual battlefield (Satava, 1995). Fourth, the technology described here has commercial applications as well. Virtual body models could be used as visualization aids in minimally invasive surgical procedures and as anatomy teaching tools in medical schools. The surgical simulator could also be used to design and analyze new surgical procedures. This simulation would enable a new procedure to be safely and thoroughly evaluated on a virtual patient before it is attempted on a live one.

Acknowledgments

This work is supported by a cooperative agreement (NCC2-9001) with the Technology Reinvestment Project, under the supervision of the ARPA Biomedical Technologies Program and NASA Ames Research Center. We are grateful to Abraham Komattu for assistance with computer programming, and Thomas Buchanan for help with the texture maps.

References

Ackerman, M. J. (1994). The Visible Human Project. In *Medicine meets virtual reality II: Interaction technology and healthcare* (pp. 5–7). Washington, D.C.: IOS Press.

Altobelli, D. E., Kikinis, R., Mulliken, J. B., Cline, H., Lorensen, W., & Jolesz, F. (1993). Computer-assisted three-dimensional planning in craniofacial surgery. *Plastic and Reconstructive Surgery, 92,* 576.

Bowersox, J. C., LaPorta, A. J., Cordts, P. R., Bhoyrul, S., & Shah, A. (1996). Complex task performance in cyberspace. In *Medicine meets virtual reality IV: Health care in the information age* (pp. 320–326). Washington, D.C.: IOS Press.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions of Pattern Analysis and Machine Intelligence, 8,* 679–698.

Chan, W. P., Lang P., Chieng, P. U., Davison P. A., Huang S. C., & Genant H. K. (1991). Three-dimensional imaging of the musculoskeletal system: An overview. *Journal of the Forensic Medical Association, 90,* 713–722.

Delp, S. L., & Loan J. P. (1995). A graphics-based software system to develop and analyze models of musculoskeletal structures. *Computers in Biology and Medicine, 25,* 21–34.

Delp, S. L., Loan, J. P., Hoy, M. G., Zajac, F. E., Topp, E. L., & Rosen, J. M. (1990). An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering, 37,* 757–767.

Dumay A. C. M. (1995). Triage simulation in a virtual environment. In *Medicine meets virtual reality III: Interactive technology and the new paradigm for healthcare* (pp. 101–111). Washington, D.C.: IOS Press.

Eisler, R. D., & Delp, S. L. (1994). Simulation and assessment of musculoskeletal trauma due to missile penetration. Submitted in response to ARPA Advanced Biomedical Technology Solicitation, BAA94-14.

Eisler, R. D., Chatterjee, A. K., & Burghart, G. H. (1996). Simulation and modeling of penetrating wounds from small arms. In *Medicine meets virtual reality IV: Health care in the information age* (pp. 511–522). Washington, D.C.: IOS Press.

Fackler, M. L. (1988). Wound ballistics: A review of common misconceptions. *Journal of the American Medical Association, 259* (18), 2730–2736.

Green, P. S., Hill, J. H., & Satava, R. M. (1991). Telepresence: Dextrous procedures in a virtual operating field. *Surgical Endoscopy, 57,* 192.

Hollerman, J. J., Fackler M. L., Coldwell, D. M., & Ben-Menachem, Y. (1990). Gunshot wounds: 1. Bullets, Ballistics, and mechanism of injury. *American Journal of Radiology, 155,* 685–690.

Hunter, I. W., Doukoglou, T. D., Lafontaine, S. R., Charatte, P. G., Jones L. A., Sagar, M. A., Mallison, G. D., & Hunter, P. J. (1993). A teleoperated microsurgical robot and associated virtual environment for eye surgery. *Presence: Teleoperators and Virtual Environments, 2* (2), 265–280.

Johnston, R., Bhoyrul, S., Way, L., Satava, R., McGovern, K.,

Fletcher, J. D., Rangel, S., & Loftin, R. B. (1996). Assessing a virtual skills simulator In *Medicine meets virtual reality IV: Health care in the information age* (pp. 608–617). Washington, D.C.: IOS Press.

Kass, M., & Miller, G. (1990). Rapid stable fluid dynamics for computer graphics. *Computer Graphics, 24* (4), 49–57.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision,* 321–331.

Kawabata, H., Kawai, H., Masada, K., & Ono, K. (1989). Computer-aided analysis of Z-plasties. *Plastic and Reconstructive Surgery, 83,* 319.

Langrana, N. A., Burdea, G., Lange K., Gomez, D., & Deshpande, S. (1994). Dynamic force-feedback in virtual knee palpation. *Artificial Intelligence in Medicine, 6,* 321–333.

Ota, D., Loftin, B., Saito, T., Lea, R., & Keller, J. (1995). Virtual reality in surgical education. *Computers in Biology and Medicine, 25* (2), 127–137.

Pieper, S. D., Laub, D. R., & Rosen, J. M. (1995). A finite-element facial model for simulating plastic surgery. *Plastic and Reconstructive Surgery, 96* (5), 1100–1105.

Pieper, S. D., Delp, S. L., Rosen, J. M., & Fisher, S. S. (1991). A virtual environment system for simulation of leg surgery. In *Stereoscopic Displays and Applications II. SPIE 1457,* 188–197.

Raines, J. K., Jaffrin, M. Y., & Shapiro, A. H. (1974). A computer simulation of arterial dynamics in the human leg. *Journal of Biomechanics, 7,* 77–91.

Robb, R. A., & Hanson, D. P. (1991). A software system for interactive and quantitative visualization of multidimensional biomedical images. *Australasian Physical & Engineering Sciences in Medicine, 14* (1), 9–30.

Rosen, J. M., Lasko-Harvill, A., & Satava, R. (1996). Virtual reality and surgery. In R. Taylor, S. Lavallee, & G. Burdea (Eds.), *Computer-Integrated Surgery: Technology and Clinical Applications,* (pp. 231–243). Cambridge, MA: MIT Press.

Satava, R. M. (1993). Virtual reality surgical simulator: The first steps. *Journal of Surgical Endoscopy, 7,* 203–205.

Satava, R. M. (1995). Virtual reality and telepresence for military medicine. *Computers in Biology and Medicine, 25* (2), 229–236.

Shen, Y., Knight, D. R., Thomas, J. X., & Vatner, S. F. (1990). Relative roles of cardiac receptors and arterial baroreceptors during hemorrhage in conscious dogs. *Circulation Research, 66* (2), 397–405.

Stredney, D., Sessanna, D., McDonald, J. S., Hiemenz, L., & Rosenberg, L. B. (1996). A virtual simulation environment for learning epidural anesthesia. In *Medicine meets virtual reality IV: Health care in the information age* (pp. 164–175). Washington, D.C.: IOS Press.

U.S. Dept. of Health and Human Services. (1991). *Combat Casualty Care Guidelines—Operation Desert Storm* Washington, D.C.: Office of the Surgeon General.

Zajtchuk, R. (1990). *Textbook of military medicine, conventional warfare,* Part I, Vol. 5, Washington, D.C.: Office of the Surgeon General.

Zamir M., Sinclair, P., & Wonnacott, T. H. (1992). Relation between diameter and flow in major branches of the arch of the aorta. *Journal of Biomechanics, 25* (11), 1303–1310.