Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration

Cagatay Basdogan, Chih-Hao Ho, and Mandayam A. Srinivasan

***Abstract*—We have developed a computer-based training system to simulate laparoscopic procedures in virtual environments (VEs) for medical training. The major hardware components of our system include a computer monitor to display visual interactions between three–dimensional (3-D) virtual models of organs and instruments together with a pair of force feedback devices interfaced with laparoscopic instruments to simulate haptic interactions. In order to demonstrate the practical utility of the training system, we have chosen to simulate a surgical procedure that involves inserting a catheter into the cystic duct using a pair of laparoscopic forceps. This procedure is performed during laparoscopic cholecystectomy (gallbladder removal) to search for gallstones in the common bile duct. Using the proposed system, the user can be trained to grasp and insert a flexible and freely moving catheter into the deformable cystic duct in virtual environments. As the catheter and the duct are manipulated via simulated laparoscopic forceps, the associated deformations are displayed on the computer screen and the reaction forces are fed back to the user through the force feedback devices. A hybrid modeling approach was developed to simulate the real-time visual and haptic interactions that take place between the forceps and the catheter, as well as the duct; and between the catheter and the duct. This approach combines a finite element model and a particle model to simulate the flexible dynamics of the duct and the catheter, respectively. To simulate the deformable dynamics of the duct in real-time using finite element procedures, a modal analysis approach was implemented such that only the most significant vibration modes of the duct were selected to compute the deformations and the interaction forces. The catheter was modeled using a set of virtual particles that were uniformly distributed along the centerline of catheter and connected to each other via linear and torsional springs and damping elements. In order to convey to the user a sense of touching and manipulating deformable objects through force feedback devices, two haptic interaction techniques that we have developed before were employed. The interactions between the particles of the catheter and the duct were simulated using a *point-based* haptic interaction technique. The interactions between the forceps and the duct as well as the catheter were simulated using the *ray-based* haptic interaction technique where the laparoscopic forceps were modeled as connected line segments.**

Manuscript received April 15, 2000; revised February 28, 2001. Recommended by Guest Editor N. Xi. This work was supported in part by the Center for Minimally Invasive Surgery of the Harvard Medical School.

C. Basdogan is with the Laboratory for Human and Machine Haptics, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA. He is also with the Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA 91109 USA (e-mail: Cagatay.Basdogan@jpl.nasa.gov).

C.-H. Ho and M. A. Srinivasan are with the Laboratory for Human and Machine Haptics, Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: chihhao@mit.edu; srini@mit.edu).

Publisher Item Identifier S 1083-4435(01)08145-5.

***Index Terms*—Collision detection, computer-based training, deformable objects, finite element modeling, haptic feedback, laparoscopic surgery, real-time interaction, surgical simulation, virtual reality.**

I. Laparoscopic Surgery

LAPAROSCOPY HAS BEEN used in a range of procedures since the early 1960s. This technology uses a small video camera and a few customized instruments to perform surgery. The camera and instruments are inserted into the abdomen through small skin incisions that enable the surgeon to explore the internal cavity without the need of making large openings. Major advantages of this type of surgery to the patient are short hospital stay, timely return to work, and less pain and scarring after the surgery. Some of the most common laparoscopic procedures include cholecystectomies, appendectomies, and hernia repair. In general, there is a trend toward performing other open surgeries laparoscopically as well. It can also be predicted that new laparoscopic techniques will be developed in the near future as the instruments get smaller and more dexterous.

Cholecystectomy (gallbladder removal: the gallbladder is drained of bile, collapsed, and then pulled out through one of the smaller cuts made in the abdominal region) is one of the most commonly performed laparoscopic procedure in the U.S. The gallbladder is a pear-shaped sac found on the liver (Fig. 1). Its purpose is to store bile (released to help with digestion when we eat foods with fat content) that is made by the liver. Gallstones that are made out of cholesterol, bile salts, calcium, and bilirubin stay in our gallbladder, without causing problems. However, if they block the outlet of the gallbladder, you may have attacks of pain in the abdomen. Occasionally, these stones may come out of the gallbladder, causing jaundice or inflammation of the pancreas. Sometimes very small stones grow in the bile ducts, which are the tubes that transfer bile to the intestines. Gallstones and gallbladder disease affect approximately one out of 10 people in the United States. In the past, patients were hospitalized for about seven days after having undergone open gallbladder surgery and spent another six weeks recovering at home. The laparoscopic technique makes it possible for most patients to go home the day after surgery and resume normal activities within a very short time. During a laparoscopic cholecystectomy, the surgeon inspects the bile ducts in search of gallstones. If stones are found, they may be removed depending on the case at hand.

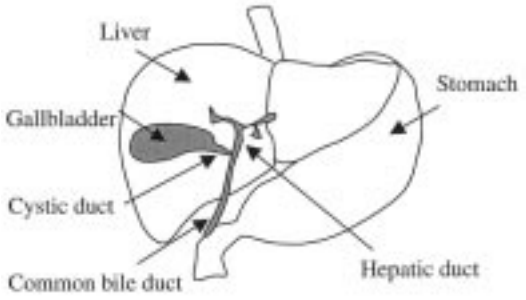


Fig. 1. Anatomy of the gallbladder.

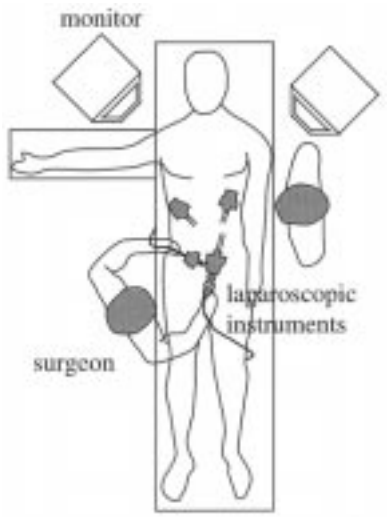


Fig. 2. Operating room setting for laparoscopic surgery (top view).

In general, a laparoscopic cholecystectomy is performed as follows: After the patient is placed under general anesthesia, the surgeon accesses the abdominal cavity with a narrow tube-like guiding instrument called trocar. A miniature camera is then inserted through the trocar. The camera projects a clear magnified image of the patient's internal organs on a video monitor, which provides the surgeon with a view of the inside of the abdomen. Three additional trocars are inserted at precise locations (one in the upper abdomen and two on the right side under the ribs) to accommodate other miniature surgical instruments such as the electrocautery, which is used to detach the gallbladder from the liver, the clipping tool that seals surrounding structures (i.e., blocks the duct leading from the gallbladder to the bile ducts and the arteries going to the gallbladder), and graspers, which are used to manipulate tissues and remove the gallbladder. The abdomen is inflated with carbon dioxide gas to expand the workspace and help the surgeon see better (Fig. 2).

II. The Need for Laparoscopic Trainers

Although the laparoscopic surgery has several advantages over the traditional surgery, the laparoscopic surgeons are still handicapped by the limitations of the current technology. The laparoscopic surgeons face four main types of problems in the operating room. First, the visualization of the internal organs is achieved by means of a wide-angle camera, but the vision is monoscopic and is limited by the field of view of camera. Second, hand and eye coordination is difficult since the TV monitor reflects mirror images of the actual hand movements and the anatomical landmarks [1], [2]. Third, the haptic (tactile sensing and force feedback) cues to the surgeon are substantially reduced since the surgeon has to interact with internal organs by means of surgical instruments attached to long thin tubes. Fourth, the laparoscopic instruments rotate about a fixed point (i.e., the entrance point: the point where the trocar is inserted into abdomen) and it is not possible to make direct translational movements while interacting with organs. Hence, it is necessary to find new training approaches or develop new devices to reduce the risks and constraints associated with laparoscopic procedures. Many others, including the authors, believe that surgical simulators that provide the surgeon with visual and haptic cues promise to be a powerful aid for training medical personnel and monitoring their performance [3]–[10].

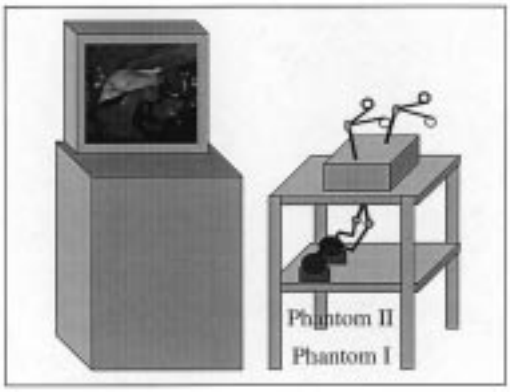


Fig. 3. Our training system is designed to simulate laparoscopic procedures in virtual environments. The schematic shows the interface between the force feedback devices and the generic laparoscopic instruments.

III. The Selected Procedure for Simulation: Insertion of a Catheter Into the Cystic Duct for Common Bile Duct Exploration

Routine or selective cholangiography is performed during a laparoscopic cholecystectomy to explore the common bile duct for gallstones. Multiple steps are followed during an intraoperative cholangiography to properly insert a catheter into the cystic duct: first, the cystic duct is dissected and a clip is placed at the junction of the infundibulum of the gallbladder with the cystic duct. A needle is inserted directly through the abdominal wall such that the catheter will naturally point toward the operative site. A catheter attached to a syringe of saline is flushed to remove bubbles, and inserted through the needle. With an instrument behind the cystic duct to stabilize it, a microscissors is used to incise the cystic duct. The catheter is then carefully inserted into the duct. Under direct fluoroscopic guidance, a radioactive material is injected through the catheter and the cholangiogram is obtained. There are several risks associated with this procedure: 1) improperly inserted catheter may cause the stones and the bile to come out; 2) injury to the bile duct may require major reconstructive procedures; and 3) the common bile duct can be easily misidentified as the cystic duct.

An important component of a cholangiography is the proper insertion of the flexible catheter (about 2–3 mm in diameter) into the cystic duct (about 8–10 mm in diameter). Due to the

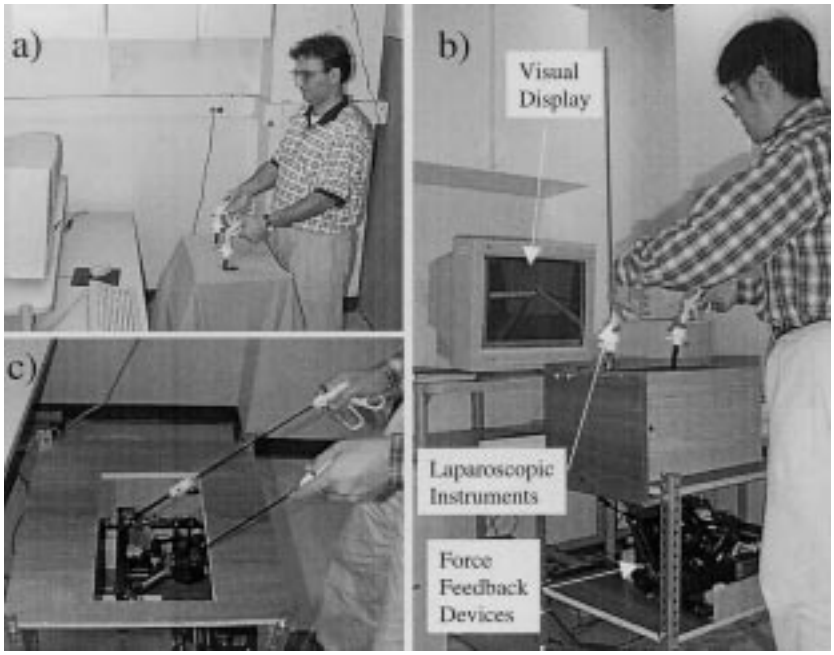


Fig. 4. (a) The complete setup. (b) The components of our surgical simulation and training system include a visual display, a pair of force feedback devices, a frame, and the other peripheral components. (c) A pair of generic (i.e., no specialized tip) laparoscopic instruments are attached to the force feedback devices to convey to the user a tactual sense of virtual objects (the cover and the enclosing box were removed in this picture to display the interface).

limited visual and haptic cues and the size of the flexible tubes that are manipulated, it is not a trivial task to perform for a junior surgical resident. To learn this procedure, the surgical residents are usually trained with a laparoscopic training box that encloses a pair of laparoscopic tools and an endoscopic camera. The laparoscopic training box, widely available in the hospitals, is utilized to imitate several part-task surgical procedures such as the catheter insertion [11], [12]. Under the guidance of an endoscopic camera, the surgical residents, for example, learn to insert a group of beads on a wire to simulate the catheter insertion procedure. However, the laparoscopic training that relies on standard laparoscopic training box and the associated procedures has limitations: 1) the simulated surgical procedures are usually the poor imitations of the actual ones; 2) the training system cannot be easily customized to the needs of the trainee; and 3) the trainee's performance cannot be measured easily.

Our simulator was designed to enhance the existing laparoscopic training system by providing a more physically based and realistic environment to the trainee. In order to demonstrate the developed concepts and the practical use of the simulator, we have selected the simulation of catheter insertion into a flexible vessel. This procedure requires a good hand-eye coordination, and relies on haptic cues as much as visual ones. From the technical point of view, the development has a challenging component of simulating physically based visual and haptic interactions: a) between two deformable objects and b) between rigid and deformable objects in real-time. Although we have focused on simulating catheter insertion in this study, the proposed techniques and the training setup can be used to simulate other endoscopic procedures such as bronchoscopy and colonoscopy, which require the insertion of flexible tubes into bronchial tubes and intestines, respectively.

IV. Components of Our VR-Based Training Setup

We have developed a multimodal training system for simulating laparoscopic procedures (Figs. 3 and 4). The hardware components of our training system include a personal computer (Dual Pentium-II 300 MHz processors) with a high-end graphics card, two force-feedback devices (Phantom, available from SensAble Technologies Inc.) to simulate haptic feedback and a box with instrumented laparoscopic tools that are interfaced with force feedback devices [force feedback devices are hidden inside the box and are attached to the distal end of the laparoscopic tools (see Figs. 4 and 5)].

As the user manipulates the actual instruments and interacts with virtual organs using the simulated instruments in virtual environments, the associated deformations of the objects are displayed on the computer screen and the reaction forces are fed back to the user through a pair of haptic devices (Fig. 4). The haptic devices enable us to reflect the interaction forces along X, Y, and Z-axes for simulating tissue pulling and pushing.

In addition, a separate encoder and actuator couple was attached to the distal end of each generic laparoscopic instrument to acquire the rotations of the tool handle and to reflect forces for simulating tissue grasping (Fig. 5). The force transmission between the actuators and the tool handle is provided through a cable connection. The details of actuator design are given in Ottensmeyer *et al.* [13]. As the user squeezes the tool handle, the angular rotations of the handle are estimated using the encoder attached to the tool tip. This information is then sent to the collision module to detect if the simulated jaws of the instrument have collided with the virtual objects in the scene during the simulation of grasping. Following the detection of collisions, the grasping forces are computed by the physics-based model

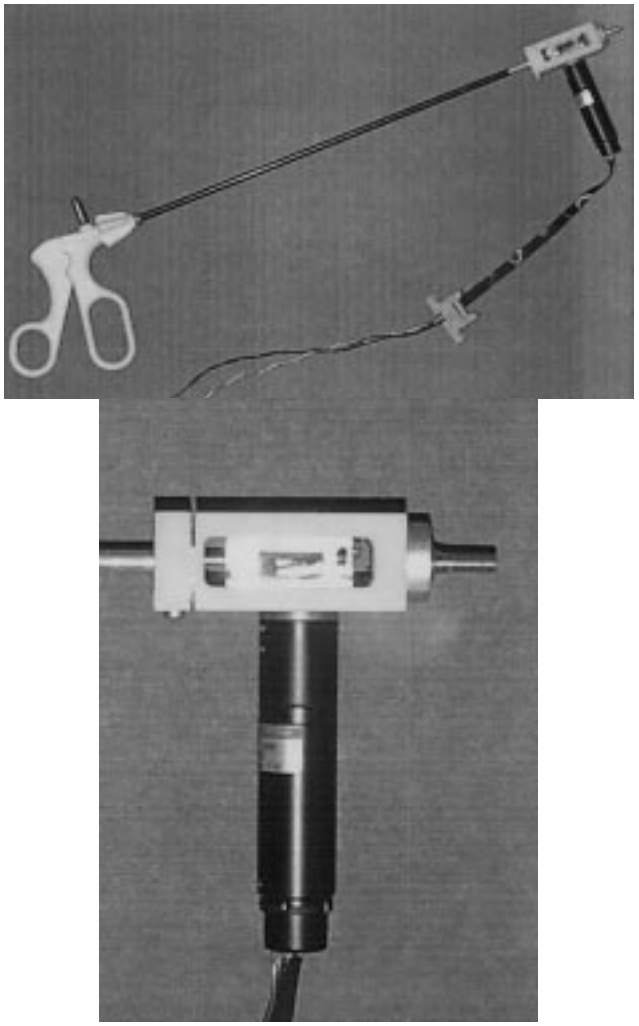


Fig. 5. Actuated laparoscopic tools for simulating tissue grasping (Ottensmeyer *et al.* [13]): An encoder-actuator couple was attached to the distal end of each laparoscopic instrument to measure the amount of squeezing with the help of the encoder and to supply haptic feedback through the actuator.

and reflected back to the user by means of the actuator attached to the distal end of the laparoscopic instrument (Fig. 5).

The software code for simulations was written in C/C++, using Open Inventor Graphics Tool Kit. Multithreading techniques were used to synchronize visual and haptic control loops, thereby increasing the haptics servo rate to about a kHZ while simultaneously satisfying the requirements of graphics update rate of at least 30 Hz.

V. Development of Algorithms for Interactions between Articulated Surgical Tool, Flexible Catheter, and Soft Tissues

This includes the computer generated models of surgical instruments and organs, collision detection, development of algorithms for simulating the physically based behavior of soft-tissues, and development of algorithms for simulating haptic interactions.

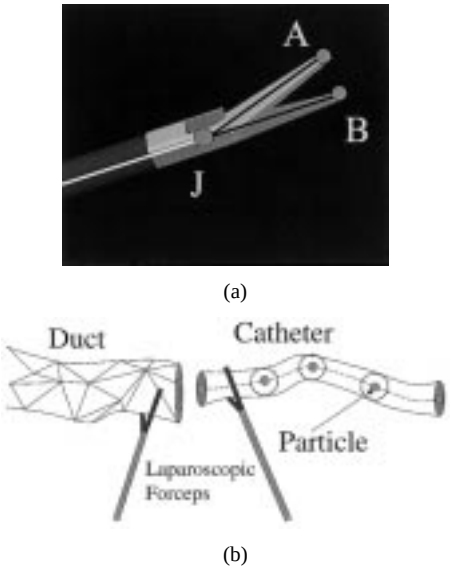


Fig. 6. (a) The graphical and computational models of the laparoscopic forceps used in the simulations: during real-time simulations, the graphical model of the forceps was displayed to the user while the collisions were detected with its computational model. The 3-D graphical model of the forceps was developed using a CAD package. The computational model of the forceps was based on three connected line segments (one for the long tube and two for the jaws of the forceps). (b) A set of particles, distributed along the center line of the catheter, were used to detect the collisions between the catheter and the inner surface of the cystic duct.

A. Geometric Models of Surgical Instruments and Organs

Three-dimensional (3-D) computer models of the laparoscopic forceps were constructed from their actual counterparts. The graphically rendered 3-D model of forceps were made of several triangles and modeled in exact dimensions using a computer-aided design package. Although they were displayed as 3-D texture-mapped objects to the user, they were modeled as connected line segments to reduce the number of collision computations during real-time interactions. Our simulations also required the geometric models of a catheter and the cystic duct, which were constructed from indexed triangle sets using a modeling package. Texture maps, scanned from a medical book, were wrapped around the geometric model of the duct to make it look more realistic.

B. Collision Detection

An important part of surgical instrument-soft tissue modeling is the fast collision detection algorithms. Although collision detection has been extensively studied in computer graphics ([14]–[16]), its implementation with haptic devices is not straightforward, as discussed in Ho *et al.*, [17]. During real-time simulations, we check the collisions: 1) between the forceps and the catheter as well as the duct and 2) between the catheter and the duct. For the purposes of detecting collisions between the laparoscopic forceps and the 3-D flexible objects during our simulations, the forceps were modeled as three connected line segments [18]. The long rigid part of the laparoscopic forceps was modeled as a line segment whose position and orientation were provided by the encoder signals of the haptic device. Similarly, the jaws of the laparoscopic forceps were modeled as line segments as well [see JA and JB in

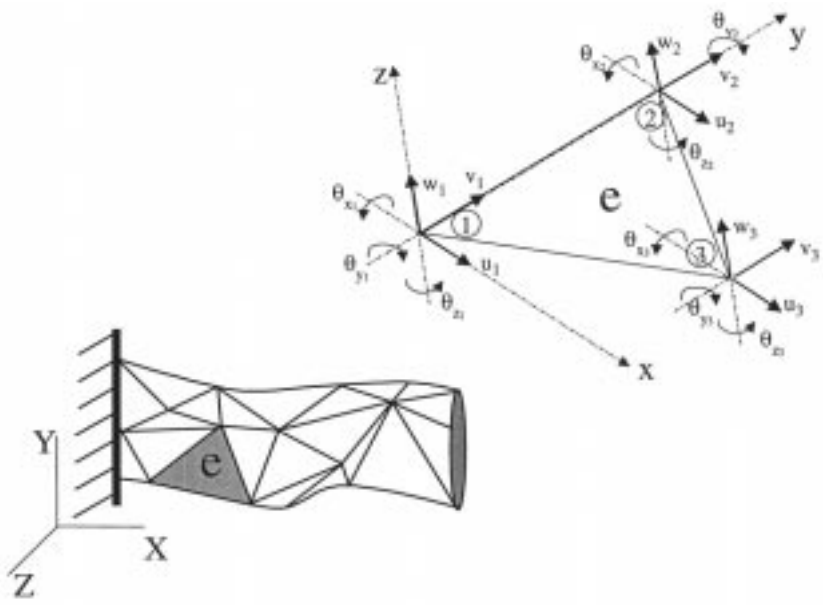


Fig. 7. The cystic duct was discretized into triangular elements. A local coordinate system, with an origin defined as the coordinates of node 1 and local y axis along the edge "12," was constructed for each element "e" of the duct. Inplane and bending loads were considered in formulating FEM equations. This led to six dof for each node of the triangle.

Fig. 6(a)]. During the simulations, the jaws that are connected to the rigid shaft at the junction point [Point J in Fig. 6(a)] enable the gripping action. The rotation of each jaw relative to this junction point was acquired by means of the encoder attached to the distal end of the actual laparoscopic instrument (Fig. 5). During the simulations, the collisions between: 1) the line segment model of the laparoscope and the duct as well as the catheter and 2) between the line segment models of the jaws ($|JA|$ and $|JB|$) and the catheter and duct were detected in real-time to simulate the interactions between instruments and the flexible bodies in the scene.

For the purposes of detecting collisions between the catheter and the cystic duct, we modeled the catheter using a group of particles connected to each other with virtual springs and dampers along its centerline [see Fig. 6(b)]. We defined a radius of influence for each particle to detect the collisions between the outer surface of the catheter and the inner surface of the duct in real-time.

C. Physically Based Modeling of Deformable Objects for Medical Simulation

Modeling and simulation of soft tissue behavior is a challenging problem. Once the soft tissue has been contacted and the collision point is identified, the problem centers around tool-tissue interaction. This involves a realistic haptic force-feedback to the user and a realistic graphical display of tissue behavior depending on what task the user chooses to perform on the tissue. This is a nontrivial problem that calls for prudence in the application of mechanistic and computer graphics techniques in an endeavor to create a make-believe world that is realistic enough to mimic reality, but efficient enough to be executable in real time. Soft-tissue mechanics is complicated not only due to nonlinearities, rate, and time dependence in material behavior, but also because the tissues are layered and nonhomogeneous. Major characteristics of living tissues include: 1) *nonlinear response*: displacement versus force profile is nonlinear; 2) *hysteresis*: displacement versus force profiles are different for continuously increasing and decreasing forces; 3) *nonhomogeneity*: displacement versus force curve changes across the tissue; 4) *anisotropy*: displacement versus force profile depends on the direction of the applied force; and 5) *large deformations*: typical forces result in large and complex deformation fields. Although sophisticated tissue models have been developed in the past by mechanics community, their integration with medical simulators has been difficult due to real-time requirements. With the addition of haptic displays, this task became even more challenging since a haptic loop requires much higher update rate than visual loop (e.g., displaying rigid objects that can be touched and felt via haptic devices in VE's requires an update rate of around 1 kHz).

We developed a hybrid modeling approach to simulate in real-time the visual and haptic interactions that take place: 1) between the forceps and the catheter, as well as the duct and 2) between the catheter and the duct. We modeled the cystic duct using a finite element method (FEM) technique to be able to integrate its tissue characteristics into the model. On the other hand, the catheter, made of an inorganic material, was modeled using a particle-based technique to reduce the computations and achieve higher rendering rates. To implement the FEM, the cystic duct was divided into triangular surface elements, properties of each element was formulated, and the elements were assembled together to study the deformation states of the duct for the forces applied by the catheter and the instruments. To implement the particle model, a set of virtual particles were defined along the centerline of the catheter. They were connected to each other via springs and dampers. Each particle was represented by its own position, velocity, and acceleration and moved

under the influence of forces applied by the cystic duct and the instruments. The implementation of both techniques in graphical simulation of deformable objects (see for example, Ng *et al.* [19] for particle-based techniques and Terzopoulus *et al.* [20] for the FEM) and surgical simulation (see for example, Cover *et al.* [3] for particle-based techniques and Bro-Nielson and Cotin [21] for FEM) can be found in the literature.

FEM techniques, though they demand more CPU time and memory, seem more promising than particle-based methods in modeling tissue characteristics. However, the finite element models developed for surgical simulation applications so far can only simulate static deformations with limited similarities to the actual tissue behavior [21]. More recently, Cotin *et al.* [7] presented a model for simulating the nonlinear response of soft tissues, but only static deformations were considered. We made modeling simplifications to simulate deformable dynamics of the cystic duct using finite elements. We reduced the number of real-time computations significantly by implementing a modal transformation. Although our FEM model can simulate the inertial effects, the high computational cost and the requirement of maintaining high haptic update rate forced us to use *linear* and *isotropic* elements. We expect that sophisticated tissue characteristics can be integrated into our finite element model over time as computers get faster and our knowledge about the behavior of soft tissues improves.

We used a particle-based model to simulate the behavior of catheter. Since particle models are less sophisticated than FEM and soft tissues show complex characteristics, a particle system may not be a good choice for modeling tissue behavior. However, they can be used to model the dynamics of flexible objects that show relatively less complex behavior such as a piece of cloth [19] or a catheter in our simulations. In addition, particle models are computationally less expensive and easier to implement than finite element models. For example, it was easier to simulate freely moving behavior of the catheter using a particle system in our simulations, since we were able to define the behavior of each particle separately. Particle-based techniques can also be advantageous in simulating interactions between objects. For example, the particles of the catheter were used to detect collisions with the polygons of the cystic duct to simulate the interactions of the catheter with the inner surface of the cystic duct in real-time.

1) FEM for Flexible Duct:

a) The finite element formulation: The cystic duct was approximated as an assemblage of discrete triangular surface elements interconnected to each other through nodal points. The displacements of these nodal points for applied external forces were the basic unknowns of our FEM analysis. The coordinates of vertices (nodal points), the polygon indexing, and the connectivity of vertices were derived from the geometric model of the duct. In order to analyze the deformations of the cystic duct under various loading conditions, we considered a combination of membrane and bending elements in our model. This facilitated the continuity in the formulation and enabled us to compute the displacements of nodal points in X, Y, and Z directions for both inplane and bending loads. For each node of the triangular element subjected to inplane loads, the displacements (u, v) in the local x and y directions were taken as the degrees of freedom. The resulting system equations were expressed in local coordinate system as

$$F_m = [k_m^e]\, U_m \tag{1}$$

where

m membrane action;

F_m $\equiv [F_{x_1}, F_{y_1}, F_{x_2}, F_{y_2}, F_{x_3}, F_{y_3},]^{\mathrm{T}}$ active membrane forces at nodes 1, 2, and 3 of the triangular element;

U_m $\equiv [u_1, v_1, u_2, v_2, u_3, v_3,]^{\mathrm{T}}$ nodal displacements in local x and y directions due to inplane loading.

In addition, the inplane stiffness matrix $[k_m^e]$ can be written as

$$[k_m^e]_{6\times 6} = \begin{bmatrix} k_{2\times 2}^{11} & k_{2\times 2}^{12} & k_{2\times 2}^{13} \\ k_{2\times 2}^{21} & k_{2\times 2}^{22} & k_{2\times 2}^{23} \\ k_{2\times 2}^{31} & k_{2\times 2}^{32} & k_{2\times 2}^{33} \end{bmatrix}. \tag{2}$$

The entries of the stiffness matrix for membrane element are given in Appendix A of this paper.

For each node of the triangular element subjected to bending loads, the displacement in local direction and slopes (rotations) about the local x and y axes ($\partial w/\partial y$ and $-\partial w/\partial x$) were considered (Fig. 7). The relation between the vertex displacements and forces were written as

$$F_b = [k_b^e]\, U_b \tag{3}$$

where

b bending action;

F_b $\equiv [F_{z_1}, M_{x_1}, M_{y_1}, F_{z_2}, M_{x_2}, M_{y_2}, F_{z_3}, M_{x_3}, M_{y_3}]^{\mathrm{T}}$ active membrane forces at nodes 1, 2, and 3;

U_b $\equiv [w_1, \Theta_{x_1}\Theta_{y_1} w_2, \Theta_{x_2}\Theta_{y_2} w_3, \Theta_{x_3}\Theta_{y_3}]^{\mathrm{T}}$ nodal displacements in local z direction and the slopes of the considered triangular element in x and y directions due to bending loading.

For example, w_1 indicates the deflection in the local z direction, Θ_{x_1} indicates slope of the triangle in y direction, and Θ_{y_1} indicates the slope of the triangle in the x direction. In addition, the bending stiffness matrix, $[k_b^e]$, can be written as

$$[k_b^e]_{9\times 9} = \begin{bmatrix} k_{3\times 3}^{11} & k_{3\times 3}^{12} & k_{3\times 3}^{13} \\ k_{3\times 3}^{21} & k_{3\times 3}^{22} & k_{3\times 3}^{23} \\ k_{3\times 3}^{31} & k_{3\times 3}^{32} & k_{3\times 3}^{33} \end{bmatrix}. \tag{4}$$

The entries of the stiffness matrix for bending element are given in Appendix A of this paper. In order to obtain the local stiffness matrix for each triangular element, the inplane and bending stiffness matrices were combined. Since six degrees of freedom (dof) were assumed for each of the vertices, the resulting combined local stiffness matrix ($[k^e]$) became 18×18 for each of the triangular elements of the object. See (5) at the bottom of the next page.

b) Transformations: Global to local coordinates: The stiffness matrix derived in the previous section utilizes a system of local coordinates. However, the geometric model of the cystic duct was generated based on the global coordinate system. In order to apply the computations described in the

previous section, a transformation from the global coordinates to local coordinate system was required [22]. The forces and displacements of a node were transformed from the global to the local system by a matrix L as:

$$a_i' = La_i \tag{6}$$

where

$$L_{6\times 6} = \begin{bmatrix} \lambda_{3\times 3} & 0 \\ 0 & \lambda_{3\times 3} \end{bmatrix} \tag{7}$$

with λ being a direction cosine matrix formed between the two sets of axes as

$$\lambda_{3\times 3} = \begin{bmatrix} \lambda_{xX} & \lambda_{xY} & \lambda_{xZ} \\ \lambda_{yX} & \lambda_{yY} & \lambda_{yZ} \\ \lambda_{zX} & \lambda_{zY} & \lambda_{zZ} \end{bmatrix} \tag{8}$$

where λ_{xX} represented the cosine of angle between x (local) and X axes (global). The local axes and their direction cosines were defined arbitrary and the decision on this was made in advance. During our precomputations, we specified the y-axis to be directed along the side 1–2 of the each triangle and surface normal of each triangle as the local z-axis (refer to Fig. 7). As can be seen from the figure, the vertices of the duct were ordered in clockwise direction for the proper implementation of the finite element model. However, they were ordered in counterclockwise direction to make sure that the surface normals pointed outward for proper visual display of the cystic duct. Once the side 1–2 of the each triangle was defined as local y-axis, the components of the direction cosine vector $v_y = [\lambda_{yX}, \lambda_{yY}, \lambda_{yZ}]^{\mathrm{T}}$ along this line were computed as

$$\lambda_{yX} = \frac{(X_2 - X_1)}{d_{12}}, \quad \lambda_{yY} = \frac{(Y_2 - Y_1)}{d_{12}}, \quad \lambda_{yZ} = \frac{(Z_2 - Z_1)}{d_{12}} \tag{9}$$

where

$$d_{12} = \left((X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2\right)^{1/2} \tag{10}$$

and X_1, X_2, Y_1, Y_2, Z_1, Z_2, are the coordinates of corner 1 and 2 in the global coordinate system. The direction cosines of the local z-axis, $v_z = [\lambda_{zX}, \lambda_{zY}, \lambda_{zZ}]^{\mathrm{T}}$ were computed from a cross product of the direction cosines of two sides of each triangle. Finally, the cross product of direction cosine vectors along the local y and z axes provided us the direction cosine vector along the local x axis:

$$\nu_x = \nu_y \times \nu_z. \tag{11}$$

Local to global coordinates: Following the finite element analysis, it was necessary to transform the results back to the global reference frame. The new positions of each nodal point for a given force were computed using system equations in the local coordinate system. Then, these new coordinates were transformed to the global coordinate system in order to update the graphics

$$[K^e] = [L']^{\mathrm{T}}[k^e][L'] \tag{12}$$

where $[K^e]$ represents the global stiffness matrix for the triangular element. The matrix that transforms the local coordinates back to global coordinates $(L'_{18\times 18})$ can be written as

$$L'_{18\times 18} = \begin{bmatrix} L_{6\times 6} & 0 & 0 \\ 0 & L_{6\times 6} & 0 \\ 0 & 0 & L_{6\times 6} \end{bmatrix} \tag{13}$$

c) Assembly of element stiffness matrices: The element stiffness matrices (K^e) were put together to construct the

$$[k^e]_{18\times 18} = \left[\begin{array}{ccc|ccc|ccc|ccc|ccc|ccc} \multicolumn{3}{c|}{\left(k^{11}_{2\times 2}\right)_m} & \multicolumn{3}{c|}{0\ 0\ 0 \atop 0\ 0\ 0} & \multicolumn{3}{c|}{\left(k^{12}_{2\times 2}\right)_m} & \multicolumn{3}{c|}{0\ 0\ 0 \atop 0\ 0\ 0} & \multicolumn{3}{c|}{\left(k^{13}_{2\times 2}\right)_m} & \multicolumn{3}{c}{0\ 0\ 0 \atop 0\ 0\ 0} \\ \hline \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{13}_{3\times 3}\right)_b} & \multicolumn{1}{c|}{0 \atop 0 \atop 0} & \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{12}_{3\times 3}\right)_b} & \multicolumn{1}{c|}{0 \atop 0 \atop 0} & \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{13}_{3\times 3}\right)_b} & \multicolumn{1}{c}{0 \atop 0 \atop 0} \\ \hline \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & 0 \\ \hline \multicolumn{3}{c|}{\left(k^{21}_{2\times 2}\right)_m} & \multicolumn{3}{c|}{0\ 0\ 0 \atop 0\ 0\ 0} & \multicolumn{3}{c|}{\left(k^{22}_{2\times 2}\right)_m} & \multicolumn{3}{c|}{0\ 0\ 0 \atop 0\ 0\ 0} & \multicolumn{3}{c|}{\left(k^{23}_{2\times 2}\right)_m} & \multicolumn{3}{c}{0\ 0\ 0 \atop 0\ 0\ 0} \\ \hline \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{23}_{3\times 3}\right)_b} & \multicolumn{1}{c|}{0 \atop 0 \atop 0} & \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{22}_{3\times 3}\right)_b} & \multicolumn{1}{c|}{0 \atop 0 \atop 0} & \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{23}_{3\times 3}\right)_b} & \multicolumn{1}{c}{0 \atop 0 \atop 0} \\ \hline \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & 0 \\ \hline \multicolumn{3}{c|}{\left(k^{31}_{2\times 2}\right)_m} & \multicolumn{3}{c|}{0\ 0\ 0 \atop 0\ 0\ 0} & \multicolumn{3}{c|}{\left(k^{32}_{2\times 2}\right)_m} & \multicolumn{3}{c|}{0\ 0\ 0 \atop 0\ 0\ 0} & \multicolumn{3}{c|}{\left(k^{33}_{2\times 2}\right)_m} & \multicolumn{3}{c}{0\ 0\ 0 \atop 0\ 0\ 0} \\ \hline \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{33}_{3\times 3}\right)_b} & \multicolumn{1}{c|}{0 \atop 0 \atop 0} & \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{32}_{3\times 3}\right)_b} & \multicolumn{1}{c|}{0 \atop 0 \atop 0} & \multicolumn{2}{c}{0\ 0 \atop 0\ 0 \atop 0\ 0} & \multicolumn{2}{c|}{\left(k^{33}_{3\times 3}\right)_b} & \multicolumn{1}{c}{0 \atop 0 \atop 0} \\ \hline \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & \multicolumn{2}{c}{0\ 0} & \multicolumn{3}{c|}{0\ 0\ 0} & 0 \end{array}\right]. \tag{5}$$

overall stiffness matrix (K). This process can be symbolically written as

$$K = \sum_{e=1}^{p} K^e \tag{14}$$

where p represents the number of triangles. The details for assembling element stiffness matrices to construct a stiffness matrix are given in the Appendix A of this paper.

d) Implementation of boundary conditions: In order to obtain a unique solution for finite element equations, at least one boundary condition must be supplied. The implemented boundary conditions modify the stiffness matrix K and make it nonsingular. There are multiple ways of implementing boundary conditions as discussed in the literature [23]. The easiest way to implement the boundary conditions is to modify the diagonal elements of the K matrix and the rows of the force vector F at which the boundary conditions will be applied. In our model, one end of the cystic was fixed, which implied zero displacements for the associated fixed nodes. To implement this boundary condition, diagonal elements of the K matrix and the rows of the F vector associated with those fixed nodes were multiplied by a large number. This procedure makes the unmodified terms of K very small compared to the modified ones.

e) Eliminating the rotational degrees of freedom (condensation): The global stiffness matrix was assembled as a symmetric square matrix and its length was six times the number of nodes of the object (recall that we defined six dof, three translations, and three rotations, for each node of the cystic duct). Although the rotational dof were necessary for the continuity of the solution, their computation was not required for our simulations. Since our main interest was to obtain the translational displacements of each node, the overall stiffness matrix K was condensed such that the rotational dof were eliminated from the formulation. In addition, the condensation of K matrix automatically reduced the number of computations to half, which was helpful for achieving real-time rendering rates. To condense the K matrix, we first partitioned the displacement and load vectors of the static problem as

$$\begin{bmatrix} K_{\mathrm{tt}} & K_{\mathrm{tr}} \\ K_{\mathrm{rt}} & K_{\mathrm{rr}} \end{bmatrix} \begin{bmatrix} U_{\mathrm{t}} \\ U_r \end{bmatrix} = \begin{bmatrix} F_{\mathrm{t}} \\ F_r \end{bmatrix} \tag{15}$$

where subscripts t and r represent the translational and rotational dof, respectively. Then, we set the forces acting on the rotational degrees of freedom to zero, and condensed the stiffness matrix as

$$K_{\mathrm{condensed}} = K_{\mathrm{tt}} - K_{\mathrm{tr}}(K_{\mathrm{rr}})^{-1}K_{\mathrm{rt}}. \tag{16}$$

The condensed stiffness matrix was a full square matrix and its length was three times the number of nodes of the object.

f) Dynamic equilibrium equations and modal analysis: The dynamic equilibrium equations for a deformable body using FEM can be written as

$$M\ddot{U} + B\dot{U} + KU = F \tag{17}$$

where M and B represent the mass and the damping matrices, respectively. Once the equations of motion for deformable body are derived, the solution is typically obtained using numerical techniques. However, as it will be discussed in detail later, the real-time display of FEM becomes increasingly more difficult as the number of elements is increased. A particular choice of the mass and damping matrices significantly reduces the number of computations. If the mass matrix is assumed to be diagonal (mass is concentrated at the nodes) and the damping matrix is assumed to be linearly proportional with the mass matrix $(B = \alpha M)$, the equations are simplified. A further modeling simplification can be implemented if we assume that high frequency vibration modes contribute very little to the computation of deformations and forces. If dynamic equilibrium equations are transformed into a more effective form, known as modal analysis, real-time solutions can be obtained with very reasonable accuracy. Pentland and Williams [24] demonstrated the implementation of this technique in graphical animation of 3-D deformable objects. In modal analysis, global coordinates are transferred to modal coordinates to decouple the differential equations. Then, one can either obtain the explicit solution for each decoupled equation as a function of time or integrate the set of decoupled equations in time to obtain the displacements and forces. Moreover, we can also reduce the dimension of the system, as well as the number of computations, by picking the most significant vibration modes and rearranging the mass, damping, and stiffness matrices. This procedure is also known as modal reduction.

We implemented the modal reduction approach to achieve a real-time performance. The errors introduced by the modal reduction were insignificant compare to the computational advantage gained through the approximation (see Appendix B: the number of computations for numerical integration of the reduced modal system were compared with the direct integration of original system in terms of *flops*).

g) Modal transformation: We defined the following transformation to transform our differential system into a modal system:

$$U(t)_{n\times 1} = \Phi_{n\times n} X(t)_{n\times 1} \tag{18}$$

where Φ is the modal matrix and U and X represent the original and modal coordinates, respectively. The modal matrix was obtained by solving the eigen problem for free undamped equilibrium equations:

$$K\phi = \omega^2 M\phi \tag{19}$$

where ω and ϕ represent the eigenvalues (i.e., vibration frequencies) and eigenvectors (i.e., mode shapes) of the matrix $(M^{-1}K)$, respectively. The modal matrix was constructed by first sorting the frequencies in ascending order and then placing the corresponding eigenvectors into the modal matrix in column-wise format $(0 \le \omega_1 \le \omega_2 \le \omega_3 \cdots \le \omega_n$, $\Phi = [\phi_1, \phi_2, \phi_3, \ldots, \phi_n])$

Finally, a set of decoupled differential equations (i.e., modal system) was obtained using the modal matrix and the transformation defined by (18)

$$\ddot{X}_i + \alpha_i \dot{X}_i + \omega_i^2 X = f_i \ , \qquad i = 1, \ldots, n \tag{20}$$

where n is the degrees of freedom (dof) of the system, $\alpha_i = 2\omega_i\zeta_i$, and $f_i = \phi_i^{\mathrm{T}} F$ are the modal damping and force, respectively. Note that ζ is known as the damping ratio or modal damping factor.

h) Modal reduction: To implement the modal reduction, we first reduced the modal matrix by picking only a few significant modes (i.e., the first "r" columns of the modal matrix). Our differential system for modal coordinates was reduced to "r" number of equations, which were then solved using a numerical integration technique

$$\ddot{X}_i^R + \alpha \dot{X}_i^R + \omega_i^2 X^R = f_i^R \ , \qquad i = 1, \ldots, r \tag{21}$$

where the superscript R represents the reduced system. We then transferred the modal coordinates back to the original coordinates using the following transformation:

$$U(t)_{n\times 1} = \Phi^R_{n\times r} X^R(t)_{r\times 1}. \tag{22}$$

i) Dynamic analysis using numerical integration: Numerical integration techniques are typically used to solve the differential equations that arise from FEM. Various integration schemes based on finite difference equations have been suggested in the literature for the dynamic analysis of FEM [25], [26]. In our case, real-time performance and the stability of solutions for various loading, initial, and boundary conditions are both equally important. For example, the central difference method appears to be fast and simple to implement, the solutions becomes unstable if the integration step (Δt) is greater than (T_n/π), where T_n is the shortest period of vibration. Bathe [26] suggests Newmark numerical integration procedure due to its optimum stability and accuracy characteristics. The Newmark method is implicit and also known as the "average acceleration" method.

Using the Newmark method, we first formulated the displacement and velocity of each reduced modal coordinate at $t + \Delta t$ as

$$^{t+\Delta t}\dot{X}^R = {}^t\dot{X}^R + \left[(1-\delta)\ {}^t\ddot{X}^R + \delta\ {}^{t+\Delta t}\ddot{X}^R\right]\Delta t \tag{23}$$

$$^{t+\Delta t}X^R = {}^tX^R + {}^t\dot{X}^R\Delta t + \left[\left(\frac{1}{2}-\eta\right)\ {}^t\ddot{X}^R + \eta\ {}^{t+\Delta t}\ddot{X}^R\right]\Delta t^2 \tag{24}$$

where η and δ are parameters that can be determined to obtain integration accuracy and stability (solutions become unconditionally stable for $\eta = 1/4$ and $\delta = 1/2$). Then, the equilibrium equation for each reduced modal coordinate was formulated at $t + \Delta t$ as

$$^{t+\Delta t}\ddot{X}^R + \alpha\ {}^{t+\Delta t}\dot{X}^R + \omega^2\ {}^{t+\Delta t}X^R = {}^{t+\Delta t}\ f^R. \tag{25}$$

Finally, we substituted the displacement and velocity formulations into the equilibrium equation derived for $t + \Delta t$ and obtained a system that looks quite similar to the static analysis (Please note that we assume the mass matrix is diagonal and the damping matrix is linearly proportional with the mass matrix such that $B = \alpha M$.):

$$\hat{F}\hat{U} = \hat{K} \tag{26}$$

Fig. 8. The catheter was modeled as a set of particles that were uniformly distributed along its centerline. In this model, each particle, moving under the influence of internal and external forces, has a mass and is connected to its neighbors with linear and torsional springs and a damping element. The graphical deformations of the catheter were simulated using a spline-based model coupled with the particle-based model.

where $\hat{F}$, $\hat{U}$, $\hat{K}$ are modified force and displacement vectors and modified stiffness matrix that are defined in Appendix B of this paper. The steps for solving this system are also given in Appendix B.

2) Modeling of Flexible Catheter Using Particles: Particle systems (also known as mass-spring models) consist of a set of point masses, connected to each other through a network of springs and dampers, moving under the influence of internal and external forces (Fig. 8). Particles simulate the physics-based behavior of deformable objects reasonably well [27], [28]. Particle systems have been extensively used in computer graphics to simulate the behavior of clothes and fluid flow. Although it has some disadvantages (see the course notes in Basdogan, [29] for detailed discussions), this technique is easy to implement since the developer does not necessarily need to construct the equations of motion explicitly. In addition, the particle approach can be efficient in modeling the haptic interactions between 3-D objects. For example, using the particles that were uniformly distributed along the centerline of the catheter, the complex haptic interactions between the 3-D catheter and 3-D cystic duct were simplified to a set of point-based haptic interactions in our simulations.

In our particle-based model, the total force applied to each particle was decomposed to *spring, bending, dissipative*, and *external* forces.

$$F_{\mathrm{total}} = \left\{ \begin{array}{ll} F_{\mathrm{spring}} & = \sum k_s(l - l_0) \\ F_{\mathrm{bending}} & = \sum \frac{k_b(\Theta - \Theta_0)}{l} \\ F_{\mathrm{dissipative}} & = -bv \\ F_{\mathrm{external}} & \end{array} \right\}_{i^{\mathrm{th}}\mathrm{particle}} \tag{27}$$

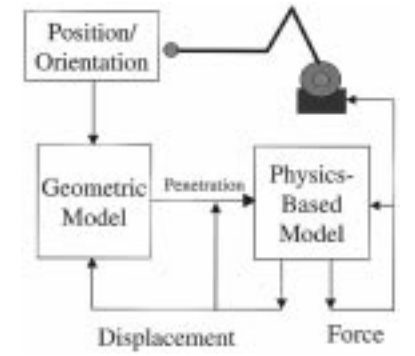
where

l	distance between the particle and its neighbor;
v	velocity of the particle;
θ	relative angle between neighboring segments;
k, b	stiffness and damping coefficients, respectively.

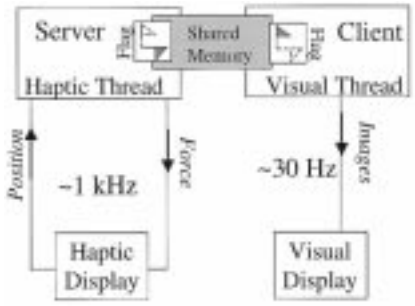
The acceleration, velocity, and position of each particle were updated using the Euler numerical integration method.

$$\begin{aligned} a_{\mathrm{t}} &= \frac{F_{\mathrm{total}}}{\mathrm{m}} \\ v_{\mathrm{t}+\Delta\mathrm{t}} &= v_{\mathrm{t}} + \Delta t\ a_{\mathrm{t}} \\ p_{\mathrm{t}+\Delta\mathrm{t}} &= p_{\mathrm{t}} + \Delta t\ v_{\mathrm{t}+\Delta\mathrm{t}} \end{aligned} \tag{28}$$

where m represents the mass of the particle. A spline-based model was used to simulate the graphical deformations of the catheter based on the movements of the particles.



(a)



(b)

Fig. 9. (a) The computational architecture for simulating force-reflecting deformable objects in VE's. The physics-based model generates forces and displacements simultaneously, which are then fed back to the model to generate the new values. (b) Integration of vision and touch: multi-threading techniques were used to synchronize visual and haptic loops for better performance.

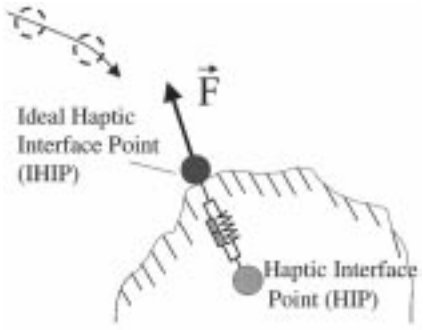
VI. Software Architecture for Physically-Based Simulations

The graphical and haptic rendering of deformable surfaces for surgical simulation involves the computation of (i.e., magnitude and direction) organ displacements and interaction forces to provide the user with a realistic sense of physical contact. We have developed a computational architecture to solve the governing system equations by iteration [Fig. 9(a)]. During the real-time computations, the force and displacement values computed in the previous step were supplied back to the physics-based model to generate the new values in the next step.

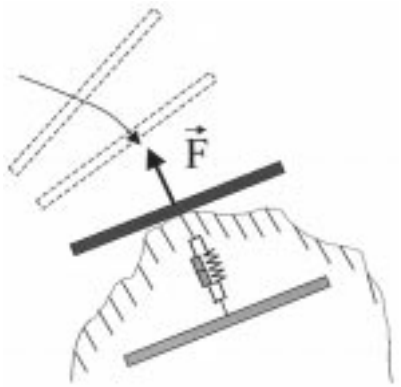
This architecture requires the synchronization of visual and haptic modalities which, was achieved in an efficient manner using shared memory and multi-threading techniques [Fig. 9(b)]. When displaying visual images, it is known that the update rate should be around 30 Hz or more to appear continuous. On the other hand, to create a satisfying haptic display, the update rate for sending the force commands to the haptic interface needs to be about 1000 Hz (In fact, this requirement is not stringent in rendering deformable surfaces. If the high haptic update rate is not satisfied, the objects actually feel softer!). To achieve high update rates and for the optimal use of CPU time, we created one thread for each modality and synchronized the data exchange between threads.

VII. Simulation of Visual and Haptic Interactions

Several haptic rendering techniques have been developed recently to render 3-D objects in virtual environments (reviewed in Basdogan and Srinivasan [30] and Srinivasan and Basdogan, [31]). The existing haptic rendering techniques can be distinguished based on the way the probing object is modeled: 1) a point [32], [33], and [17]; 2) a line segment [34], [35]; or 3) a 3-D object made of group of points polygons, line segments and polygons [36]. In point-based haptic interactions, only the end point of haptic device, also known as the haptic interface point (HIP), interacts with objects. Since a virtual surface has a finite stiffness, the end point of the haptic device penetrates into the object after collision. Each time the user moves the generic probe of the haptic device, the collision detection algorithm checks to see if the end point is inside the virtual object. If so, the depth of indentation is calculated as the distance between the current HIP and the corresponding surface point, also known as the ideal haptic interface point IHIP, or proxy point [Fig. 10(a)]. In ray-based interactions, the generic probe of the haptic device is modeled as a finite line whose orientation is taken into account, and the collisions are checked between the finite line and the objects. This approach enables us to touch multiple objects simultaneously and to reflect forces as well as torques to the user, both of which are not possible with the point-based approach [Fig. 10(b)]. Of course, the simulation of haptic interactions between two 3-D polyhedra is the



(a)



(b)

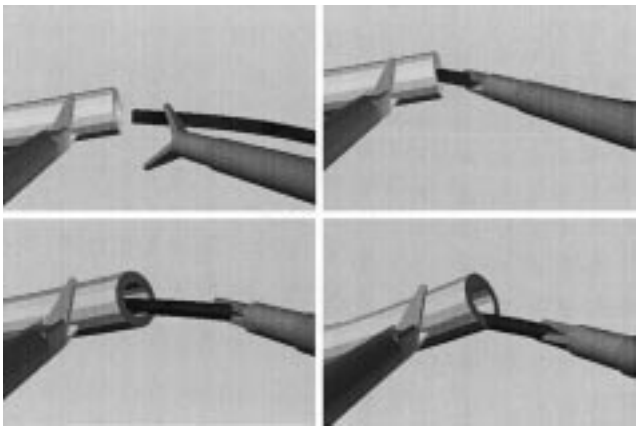
Fig. 10. (a) Point-based interactions: The haptic interface point is free to move in 3-D space and utilized to detect collisions with virtual objects. The IHIP (i.e., proxy point) represents the ideal location of the HIP that is constrained to stay on the surface of the object. A simple spring and damper model can be used to compute the reaction forces (F_x, F_y, F_z) that will be sent to the user for rigid objects. In simulating deformable objects, physics-based model computes the displacements and forces in tandem. (b) Ray-based interactions: the same concept of computing interaction forces applies to the simulation of interactions between a line segment and a 3-D object as well. In addition to displaying forces, interaction torques can also be displayed to the user using this technique since multiple collisions can be detected simultaneously using a finite line.

most desirable but this is computationally more expensive than the point-based and ray-based interactions.

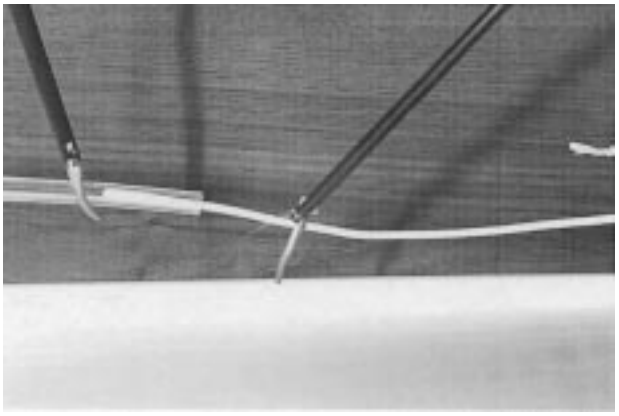
Independent of the selected computational model for the haptic probe (i.e., point, line segment, or 3-D polyhedron), the key component of any haptic interaction paradigm is the modeling of *collision response* phase: the response to the detection of collisions in terms of how the forces are reflected to the user. During the simulation of haptic interactions, the virtual probe (i.e., the computational model of the probe) penetrates into the object. Then, force/torque commands are sent to the motors to prevent the probe from further penetrating into the object. However, it is important to observe that the probing object is always inside the contacted object during the haptic interactions. Hence, the depth of penetration and the history of contacts are important in haptic rendering to properly compute the direction and magnitude of reaction forces and torques (see Ho *et al.* [17] for the details).

In our simulations, the contact interactions between the catheter and the duct were modeled using a point-based haptic rendering technique while the ones between the instrument and flexible objects were modeled using the ray-based rendering technique. To simulate haptic interactions between the instruments and the flexible objects in the scene (i.e., catheter and duct), the forceps were modeled as three connected line segments. Following the computation of collisions, the index of the collided node and the depth of penetration was sent to the physics-based model to compute nodal displacements and the interaction forces. During the simulations, the simulated forces acting on the tip of the generic instruments were displayed to the trainee through haptic devices. In addition, the trainee felt the coupling moments generated by the instruments. Since the laparoscopic forceps were inserted into the training box through a pair of ports whose position was fixed, the interaction forces between the physical instruments and the actual ports and the forces acting on the tool tip due to virtual interactions generated coupling moments. Moreover, the trainee felt the gripping forces as he/she squeezed the tool handle during the simulations. During the simulation of grasping, the simulated arms of the forceps were slowly rotated toward each other until a collision was detected. Following the detection of collision, the grasping forces were calculated and reflected to the user via the actuator attached to the distal end of the surgical instrument.

To simulate contact interactions between the catheter and the cystic duct, we made modeling simplifications. In general, simulating the visual and haptic interactions between two flexible 3-D objects is quite challenging. Not only the detection of object-object collisions is computationally expensive, but also the calculations during "collision response" phase are complex. For this reason, we modeled the catheter using a group of particles. Instead of detecting the collisions extensively between the 3-D geometric models of the catheter and the duct, we only check the collisions between the particles (see the section on particle-based modeling of the catheter) of the catheter and the inner surface of the duct. In other words, the 3-D interactions between the catheter and the duct were reduced to a group of point-based interactions to achieve acceptable haptic rendering rates. Since the particles were distributed along the centerline of the catheter and the collisions must be detected before the catheter penetrates out of the duct, we define a sphere of influence for each particle. The forces acting on each particle was computed based on the depth of penetration of each sphere to the polygonal surface of the duct and its interactions with the neighboring particles (recall that the particles were connected to each other with springs and dampers).



(a)



(b)

Fig. 11. (a) Simulation of the catheter insertion in virtual environments: The cystic duct was modeled using finite element methods while the catheter was modeled using a particle based technique. (b) A scene from current training method: the surgical residents are trained with a laparoscopic training box in hospitals today. However, the laparoscopic training that relies on standard laparoscopic training box and the associated procedures has limitations: 1) the simulated surgical procedures are usually poor imitations of the actual ones; 2) the training system cannot be easily customized to the needs of the trainee; 3) the trainee's performance cannot be measured easily.

VIII. Discussion

We have developed a computer-based training system to simulate laparoscopic procedures in virtual environments for training surgical residents. The simulator provides the user with visual and haptic feedback. We opted to simulate insertion of a flexible catheter into the cystic duct to demonstrate the feasibility of the developed system and the proposed modeling concepts (Fig. 11). The simulation of this procedure involves the real-time rendering of computer generated graphical images of laparoscopic forceps, a catheter and the cystic duct, physically based modeling of soft tissue behavior, and display of touch and force sensations to the trainee through the simulation of haptic interactions among the forceps, the catheter and the

cystic duct. The development phases of the simulator along with integration issues were discussed step by step in the paper.

Several simplifications and assumptions were made in the modeling and implementation phases in order to simulate the visual and haptic interactions between the virtual objects in real-time. To reduce the number of collision computations, simplified models of the instruments and objects were used. For example, the forceps were modeled as three connected line segments in order to detect the collisions between the simulated forceps and the flexible objects in real-time. Similarly, the catheter was modeled as a set of particles, connected to each other via springs and dampers, to detect its collisions with the polygonal model of the cystic duct. As the catheter is inserted into the duct, increasingly more particles collide with the inner surface of the duct and the computational load increases. However, detecting multiple collisions between a set of particles and a 3-D polygonal surfaces is still computationally less expensive than detecting collisions between two 3-D polygonal surfaces. Although our selected medical procedure involves the simulation of interactions between two tubular-shaped objects, this approach can easily be extended to the simulation of two arbitrary shaped 3-D objects. A set of particles that are strategically distributed over the surface of a probing 3-D object (i.e., the object that is held by the user) can be used to detect collisions with a contacted 3-D polygonal object. Since the collision detection only involves checking if a set of moving 3-D points penetrates into a 3-D polyhedron, real-time rendering rates can be achieved.

A hybrid modeling approach that smoothly blends a high fidelity finite element model with a lower fidelity particle model was introduced to simulate the physically based behavior of the duct and the catheter, respectively [37]. The cystic duct exhibits the characteristics of a living tissue, which is complex and requires high-fidelity models such as finite elements to simulate its behavior. On the other hand, the catheter, made of plastic, was modeled using a set of particles connected to each other with springs and dampers. This hybrid approach, which enables the integration of two different modeling paradigms, can be generalized to simulate the physically-based behavior of deformable objects in real-time. First, the parts of the geometry, which require a higher resolution force and displacement response, can be distinguished from the parts that require a coarser analysis. Then, the low and high fidelity models, chosen for each region, are combined to simulate the behavior of the object. The challenge in here is to link the high and the low fidelity models successfully while satisfying the continuity requirements.

A modal analysis approach was implemented to run our *dynamic* finite elements in real-time. As the comparison in Appendix B demonstrates, the direct integration of the original differential system (DIOS) results in $O(n^2)$ floating-point operations. However, the solution using modal analysis (NIRMS) leads to $O(n \log n))$ flops. Therefore, the simulation using DIOS will be increasingly more difficult as n (i.e., degrees of freedom or the number of nodes of the object) increases. Although our finite element model can simulate the real-time dynamics of a deformable object, it only approximates the characteristics of living tissues due to the stringent requirements of real-time simulation (update rate should be around 30 Hz and 1 kHz for visual and haptic displays, respectively). Moreover, our restricted knowledge on the biomechanical properties of living tissues makes the modeling more difficult. Although an extensive amount of research has been conducted to study the soft tissue characteristics *in vitro,* there is very little literature available on *in vivo* measurement in the context of laparoscopic surgeries [38], [39].

In this paper, we also discussed the modeling paradigms for simulating haptic interactions between virtual objects. We demonstrated the implementations of our point and ray-based haptic rendering techniques in medical simulation. The developed techniques convey to the user a sense of touch and force while they interact with soft objects in virtual environments. The forceps were modeled as connected finite rays to simulate the haptic interactions that take place between the laparoscopic forceps and flexible objects. On the other hand, the haptic interactions between the catheter and the duct were modeled using the point-based haptic rendering technique. A set of points that are positioned along the centerline of the catheter was used to detect collisions with the inner polygonal surface of the duct. The interaction forces were calculated based on the penetration depth of each particle and the model that governs the physical interactions. Then, computed forces were reflected to the user through haptic devices attached to the distal end of the customized laparoscopic instruments.

A software architecture was developed to integrate the visual and haptic modalities. This architecture utilizes a multithreading technique to synchronize the visual and haptic threads and to control the data exchange. Since visual and haptic modalities have different sensory characteristics, such synchronization is essential for the optimum use of computational power.

APPENDIX A
FEM

A. Construction of Local Element Stiffness Matrix

If the thickness of the triangular element is assumed to be constant, the inplane stiffness matrix ($[k_m^e]$) can be separated into two parts; one the matrix due to normal stresses ($[k_n^e]$), and the other due to shear stresses ($[k_s^e]$) (see Rao [40]).

$$[k_m^e] = [k_n^e] + [k_s^e]$$

The components of these matrices can be given in the explicit form as shown by the first set of equations at the bottom of the next page, where

x_{ij} $= x_i - x_j$
y_{ij} $= y_i - y_j$
t thickness of the element;
E Young's modulus;
A area of the triangle;
nu Poisson's ratio.

Young's modulus determines the flexibility of the object, whereas the Poisson's ratio defines the relation between lateral strain and strain along the direction of loading (Hence, smaller values of Poisson's ratio indicates that the object will not stretch much under tension.)

Similarly, the bending stiffness matrix ($[k_b^e]$) can be expressed as the multiplication of three matrices (see Rao [40])

$$[k_b^e] = \left[N^{-1}\right]^{\mathrm{T}} [A] \left[N^{-1}\right]$$

where (see matrix equation at the bottom of the page) and A is a symmetric matrix ($A[i][j] = A[j][i], \quad i = 1,\ldots,9;\ j = 1,\ldots,9$) with elements described as

$$\begin{aligned}
A[1][1] =& 0\\
A[2][1] =& A[2][2] = 0\\
A[3][1] =& A[3][2] = A[3][3] = 0\\
A[4][1] =& A[4][2] = A[4][3] = 0\\
A[4][4] =& 2x_3y_2\\
A[5][1] =& A[5][2] = A[5][3] = A[5][4] = 0\\
A[5][5] =& x_3y_2(1-\nu)\\
A[6][1] =& A[6][2] = A[6][3] = A[6][5] = 0\\
A[6][4] =& 2\nu x_3y_2\\
A[6][6] =& 2x_3y_2\\
A[7][1] =& A[7][2] = A[7][3] = A[7][5] = 0\\
A[7][4] =& 2x_3^2y_2\\
A[7][6] =& 2\nu x_3^2y_2\\
A[7][7] =& 3x_3^3y_2\\
A[8][1] =& A[8][2] = A[8][3] = 0\\
A[8][4] =& \frac{2}{3}x_3y_2(\nu x_3 + (y_2+y_3))\\
A[8][5] =& \frac{2}{3}(1-\nu)x_3y_2(x_3y_2 + (y_2+y_3))\\
A[8][6] =& \frac{2}{3}x_3y_2(x_3y_2 + \nu(y_2+y_3))\\
A[8][7] =& \frac{1}{2}x_3^2y_2\{2\nu x_3y_2 + (y_2+2y_3)\}\\
A[8][8] =& \frac{1}{3}x_3y_2\left\{(3-2\nu)\left(x_3^2y_2\right)\right.\\
&\qquad + (2-\nu)(x_3y_2)(y_2+2y_3)\\
&\qquad \left. +(3-2\nu)\left(y_2^2+y_2y_3+y_3^2\right)\right\}\\
A[9][1] =& A[9][2] = A[9][3] = A[9][5] = 0\\
A[9][4] =& 2\nu x_3y_2(y_2+y_3)\\
A[9][6] =& 2x_3y_2(y_2+y_3)\\
A[9][7] =& \frac{3}{2}(\nu {x_3}^2y_2)(y_2+2y_3)\\
A[9][8] =& \frac{1}{2}\left(x_3^2y_2\right)(y_2+2y_3) + \nu x_3y_2\left(y_2^2+y_2y_3+y_3^2\right)\\
A[9][9] =& 3x_3y_2\left(y_2^2+y_2y_3+y_3^2\right)
\end{aligned}$$

B. Computer Implementation of Assembly of Overall Stiffness Matrix (Adapted from Rao [40])

If K is stored as a symmetric square matrix

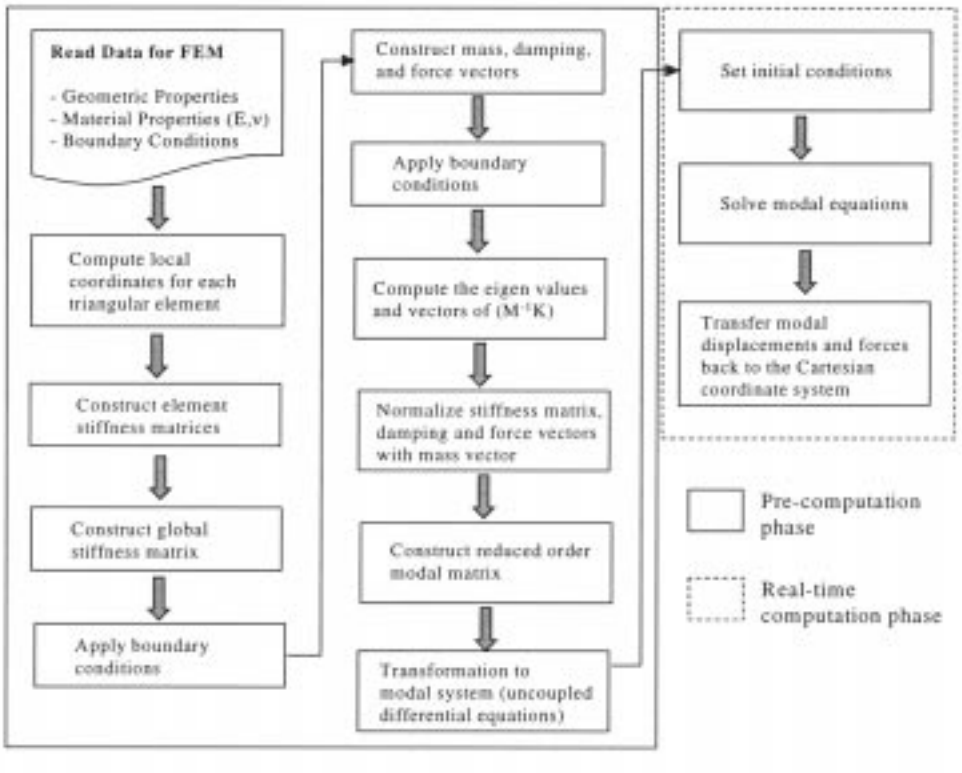
```
initialize K (set the elements of the K
   matrix to zero)
```

$$[k_n^e] = \frac{Et}{4A(1-\nu^2)}\begin{bmatrix}
y_{32}^2 & & & & & \\
-\nu y_{32}x_{32} & x_{32}^2 & \text{Symmetric} & & & \\
-y_{32}y_{31} & \nu x_{32}y_{31} & y_{31}^2 & & & \\
\nu y_{32}x_{31} & -x_{32}x_{31} & -\nu y_{31}x_{31} & x_{31}^2 & & \\
y_{32}x_{21} & -\nu x_{32}y_{21} & -y_{31}y_{21} & \nu x_{31}y_{21} & y_{21}^2 & \\
-\nu y_{32}x_{21} & x_{32}x_{21} & \nu y_{31}x_{21} & -x_{31}x_{21} & -\nu y_{21}x_{21} & x_{21}^2
\end{bmatrix}$$

$$[k_s^e] = \frac{Et}{8A(1+\nu)}\begin{bmatrix}
x_{32}^2 & & & & & \\
-x_{32}y_{32} & y_{32}^2 & \text{Symmetric} & & & \\
-x_{32}x_{31} & y_{32}x_{31} & x_{31}^2 & & & \\
x_{32}y_{31} & -y_{32}y_{31} & -x_{31}y_{31} & y_{31}^2 & & \\
x_{32}x_{21} & -y_{32}x_{21} & -x_{31}x_{21} & y_{31}x_{21} & x_{21}^2 & \\
-x_{32}y_{21} & y_{32}y_{21} & x_{31}y_{21} & -y_{31}y_{21} & -x_{21}y_{21} & y_{21}^2
\end{bmatrix}$$

$$[N] = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & 0 & y_2 & 0 & 0 & y_2^2 & 0 & 0 & y_2^3\\
0 & 0 & 1 & 0 & 0 & 2y_2 & 0 & 0 & 3y_2^2\\
0 & -1 & 0 & 0 & -y_2 & 0 & 0 & -y_2^2 & 0\\
1 & x_3 & y_3 & x_3^2 & x_3y_3 & y_3^2 & x_3^3 & \left(x_3^2y_3+x_3y_3^2\right) & y_3^3\\
0 & 0 & 1 & 0 & x_3 & 2y_3 & 0 & \left(2x_3y_3+x_3^2\right) & 3y_3^2\\
0 & -1 & 0 & -2x_3 & -y_3 & 0 & -3x_3^2 & -\left(y_3^2+2x_3y_3\right) & 0
\end{bmatrix}$$

TABLE I
FLOWCHART OF THE FINITE ELEMENT COMPUTATIONS




```
for i = 1: number of triangles
          for j = 1:3
          ij   = the index of the j-th node
    of the i-th triangle
               for k = 1:3
               ik    = the index of the k-th
    node of the i-th triangle
               K[ij][ik] = K[ij][ik] + K^i[j][k]
               end
          end
end
```

C. Flow Chart of the Finite Element Computations

See Table I.

APPENDIX B
NUMERICAL INTEGRATION USING NEWMARK PROCEDURE
(Adapted from Bittnar and Sejnoha [25])

Note that

- the damping matrix is assumed to be linearly proportional with mass matrix in the original equations, $B = \alpha M = \text{diag}\,(\alpha m_i)$
- the steps for "Direct Numerical Integration of Original System" (DNIOS) and the "Numerical Integration of Reduced Modal System" (NIRMS) are slightly different from each other. These differences were clearly marked in the algorithm below.

A. Precomputation Phase

a) Construct stiffness and mass matrices: K and $M = \text{diag}(m_i)$

b) Define the initial conditions:

$$U_0, \dot{U}_0, \ddot{U}_0 \qquad \text{(DNIOS)}$$
$$X_0 = \Phi U_0, X = \Phi \dot{U}_0, X = \Phi \ddot{U}_0 \qquad \text{(NIRMS)}$$

c) Calculate the constants of "Newmark" integration

$$a_0 = \frac{4 + 2\alpha\Delta t}{\Delta t^2} \quad a1 = a0 \quad a2 = \frac{4}{\Delta t} + \alpha \quad a_3 = 1$$
$$a_4 = \frac{4}{\Delta t^2} \quad a_5 = -a_4 \quad a_6 = a_5(\Delta t) \quad a_7 = -1$$
$$a_8 = \frac{\Delta t}{2} \quad a_9 = \frac{\Delta t^2}{4} \quad a_{10} = a_9$$

d) Construct the modified stiffness matrix:

$$\hat{K} = K + a_0 M \qquad \text{(DNIOS)}$$
$$\hat{K} = K + a_0 I = \text{diag}\,(\omega_i^2 + a_0) \qquad \text{(NIRMS)}$$

e) Decompose the modified stiffness matrix using Cholesky decomposition (see [41]) (Only for DNIOS)

f) Compute the modal force vector (Only for NIRMS)

$$^{t+\Delta t}f^R = (\Phi^R)^{\mathrm{T}} \; {}^{t+\Delta t}F$$

B. Real-Time Simulation Phase (Computational Steps for Direct Integration of Equilibrium Equations)

Step 1) Construct modified force vector

$$\hat{F} = {}^{t+\Delta t}F + M\left(a_1{}^tU + a_2{}^t\dot{U} + a_3{}^t\ddot{U}\right) \quad \text{(DNIOS)}$$

$$\hat{f}^R = {}^{t+\Delta t}f^R + a_1{}^tX^R + a_2{}^t\dot{X}^R + a_3{}^t\ddot{X}^R \quad \text{(NIRMS)}$$

Step 2) Compute modified displacements using

$$\hat{K}\hat{U} = \hat{F} \quad \text{"Cholesky backsubstitution" (see [41])} \quad \text{(DNIOS)}$$

$$\hat{X}^R = \frac{\hat{f}^R}{\hat{K}} \quad \text{(NIRMS)}$$

Step 3) Compute accelerations, velocities, and displacements at time $t + \Delta t$

$$\begin{aligned} {}^{t+\Delta t}\ddot{U} &= a_4\hat{U} + a_5{}^tU + a_6{}^t\dot{U} + a_7{}^t\ddot{U} \\ {}^{t+\Delta t}\dot{U} &= {}^t\dot{U} + a_8\left({}^t\ddot{U} + {}^{t+\Delta t}\ddot{U}\right) \\ {}^{t+\Delta t}U &= {}^tU + \Delta t{}^t\dot{U} + a_9{}^t\ddot{U} + a_{10}{}^{t+\Delta t}\ddot{U} \end{aligned} \quad \text{(DNIOS)}$$

$$\begin{aligned} {}^{t+\Delta t}X^R &= a_4\hat{X}^R + a_5{}^tX^R + a_6{}^t\dot{X}^R + a_7{}^t\ddot{X}^R \\ {}^{t+\Delta t}\dot{X}^R &= {}^t\dot{X}^R + a_8\left({}^t\ddot{X}^R + {}^{t+\Delta t}\ddot{X}^R\right) \\ {}^{t+\Delta t}\ddot{X}^R &= {}^tX^R + \Delta t\ {}^t\dot{X}^R + a_9{}^t\ddot{X}^R + a_{10}\ {}^{t+\Delta t}\ddot{X}^R \\ {}^{t+\Delta t}U &= (\Phi^R)\ {}^{t+\Delta t}X^R \end{aligned} \quad \text{(NIRMS)}$$

Step 4) Compute the forces that will be reflected to the user

$${}^{t+\Delta t}F = M\ {}^{t+\Delta t}\ddot{U} + B\ {}^{t+\Delta t}\dot{U} + K\ {}^{t+\Delta t}U \quad \text{(DNIOS)}$$

$$\begin{cases} {}^{t+\Delta t}f^R = {}^{t+\Delta t}\ \ddot{X}^R + \alpha\ {}^{t+\Delta t}\dot{X}^R + \Omega^2\ \ {}^{t+\Delta t}X^R \\ {}^{t+\Delta t}F = (\Phi^R)\ \ {}^{t+\Delta t}f^R \\ \\ \text{where,} \Omega^2 = \text{diag}(\omega_i^2) \end{cases} \quad \text{(NIRMS)}$$

Step 5) Goto Step 1.

C. Comparison of Direct Numerical Integration of Original System (DNIOS) with Numeric Integration of Reduced Modal System (NIRMS)

One way to quantify the computational differences between the numerical integration schemes is with the notation of a *flop* (Golub and Van Loan [42]). A flop is a floating-point operation. For example, a dot product of two vectors of length "n" involves "$2n$" flops since there are n multiplications and n additions. Similarly, the multiplication of a matrix of size $m \times n$ with a vector of length n involves "$2mn$" flops.

In general, the integration of equilibrium equations for force-reflecting deformable body involves the computation of forces and displacements at each time step. If we assume that both techniques (DNIOS and NIRMS) follow the same integration scheme (Newmark method) to compute the forces and displacements, we can compare the number of computations in terms of flops. Since precompuations are done in advance, we only compare the flops of real-time computations step by step according to the integration scheme presented in the Appendix B.

Step 1) $7n$ flops for DNIOS and $6r$ flops for NIRMS, where n represents the dofs and r represents the significant modes.

Step 2) Computation of $\hat{U}$, using Cholesky backsubstitution, involves $\mathbf{0}(n^2)$ flops for DNIOS. Computation of $\hat{X}$ involves r number of flops for NIRMS (Note that we do not need Cholesky decomposition in NIRMS since the equations are decoupled from each other as a result of the modal transformation. Hence, a simple division operation—dividing the modified force vector with the diagonal $\hat{K}$ matrix—is sufficient for computing $\hat{X}$.)

Step 3) Computation of accelerations, velocities, and positions involve total of $16n$ $(7n + 3n + 6n)$ flops for DNIOS and $16r + 2nr$ for NIRMS. Note that the last line of step 3 in NIRMS involves a multiplication of reduced modal matrix (size of $n \times r$) with the modal displacement vector (length of r) which accounts for the additional $2nr$ flops.

Step 4) Computation of force vector at $t+\Delta t$ involves $2n^2 + 4n$ flops for DNIOS and $2nr + 4r$ for NIRMS.

It is clear from this analysis that the number of computations increase with the order $O(n^2)$ for DNIOS. However, it will be in the order of $O(n \log n)$ for NIRMS as n goes to infinity.

ACKNOWLEDGMENT

Several people contributed to this paper in many different ways. The authors would like to acknowledge discussions with Dr. S. Small and Dr. D. Rattner of the Harvard Medical School about the medical procedures. They are also thankful to Dr. D. Rattner and Dr. S. Dawson for arranging visits to the operating room and the animal facility. C. Barlow of the Harvard Medical School was very helpful during their visits to the animal laboratory. She also provided them with a standard laparoscopic trainer, which helped them to design and test the catheter insertion procedure. They are also grateful to Dr. S. Small of the Harvard Medical School for providing them with books and videotapes on laparoscopic procedures. Their earlier discussions with S. De of MIT on finite element techniques were helpful. M. Ottensmeyer, E. Ben-Ur, and K. Salisbury of the MIT-Artificial Intelligence Lab provided us with the grippers that were attached to the distal end of the modified laparoscopic instruments for simulating object grasping.

REFERENCES

[1] M. R. Treat *et al.*, "Surgeon's perspective on the difficulties of laparoscopic surgery," in *Computer Assisted Surgery*, Taylor *et al.*, Eds. Cambridge, MA: MIT Press, 1996, pp. 559–565.

[2] E. Schippers and V. Schumpelick *et al.*, "Requirements and possibilities of computer-assisted endoscopic surgery," in *Computer Assisted Surgery*, Taylor *et al.*, Eds. Cambridge, MA: MIT Press, 1996, pp. 567–575.
[3] S. A. Cover, N. F. Ezquerra, J. O'Brien, R. Rowe, T. Gadacz, and E. Palm, "Interactively deformable models for surgery simulation," *IEEE Comput. Graphics Appl.*, pp. 65–78, Nov. 1993.
[4] C. Basdogan, J. P. Loan, J. M. Rosen, and S. L. Delp, "An interactive model of the human lower limb for simulation of surgical procedures in virtual environments," in *Winter Annu. Meeting ASME'96*, vol. BED-33, Atlanta, GA, Nov. 17–22, 1996, pp. 439–440.
[5] R. Playter, "A novel virtual reality surgical trainer with force feedback: Surgeon vs. medical student performance," in 2nd PHANToM User's Group Workshop, Dedham, MA, Oct. 19–22, 1997.
[6] C. Baur, D. Guzzoni, and O. Georg, "Virgy: A virtual reality and force feedback based endoscopic surgery simulator," in *Proc. Medicine Meets Virtual Reality (MMVR'6) Conf.*, San Diego, CA, 1998, pp. 385–391.
[7] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissues for surgery simulation," *IEEE Trans. Visual. Comput. Graphics*, vol. 5, pp. 62–73, 1999.
[8] F. Tendick, M. Downes, T. Goktekin, M. C. Cavusoglu, D. Feygin, X. Wu, R. Eyal, M. Hegarty, and L. W. Way, "A virtual environment testbed for training laparoscopic surgical skills," *Presence*, vol. 9, no. 3, pp. 236–255, 2000.
[9] H. K. Cakmak, H. Maass, U. Kuhnaphel, and G. Bretthauer, "KisMo: A virtual reality modeling tool for surgical education and training," in *Complexity in Medicine KBF Symp.*, Cologne, Germany, 2000, [Online]. Available: http://iregt1.iai.fzk.de/KISMET/kis_apps_med.html.
[10] J. Berkley, P. Oppenheimer, W. Weghorst, D. Berg, G. Raugi, D. Haynor, M. Ganter, C. Brooking, and G. Turkiyyah, "Creating fast finite element models from medical images," in *Proc. Medicine Meets Virtual Reality*, Irvine, CA, 2000, pp. 26–32.
[11] R. J. C. Steele, S. W. Hosking, and S. C. S. Chung, "Graded exercises for basic training in laparoscopic surgery," *J. Coll. Surg. Edinburg*, vol. 39, pp. 112–117, Apr. 1994.
[12] S. J. Shapiro, M. Paz-Partlow, L. Daykhovsky, and L. A. Gordon, "The use of a modular skills center for the maintenance of laparoscopic skills," *Surgical Endoscopy*, vol. 10, pp. 816–819, 1996.
[13] M. P. Ottensmeyer, E. Ben-Ur, and K. J. Salisbury, "Input and output for surgical simulation: Devices to measure tissue properties in vivo and a haptic interface for laparoscopy simulators," in *Proc. Medicine Meets Virtual Reality (MMVR'2000)*, Irvine, CA, Jan. 27–30, 2000, pp. 236–242.
[14] M. Lin, "Efficient collision detection for animation and robotics," Ph.D., Univ. California, Berkeley, 1993.
[15] S. Gottschalk, M. Lin, and D. Manocha, "OBB-Tree: A hierarchical structure for rapid interference detection," in *Proc. ACM SIGGRAPH*, Aug. 1996.
[16] P. Hubbard, "Collision detection for interactive graphics applications," *IEEE Trans. Visual. Comput. Graphics*, vol. 1, pp. 219–230, 1995.
[17] C. Ho, C. Basdogan, and M. A. Srinivasan, "An efficient haptic rendering technique for displaying 3D polyhedral objects and their surface details in virtual environments," *Presence: Teleoperators Virtual Environment*, vol. 8, no. 5, pp. 477–491, 1999.
[18] C. Basdogan, C. Ho, M. A. Srinivasan, S. D. Small, and S. L. Dawson, "Force interactions in laparoscopic simulations: Haptic rendering of soft tissues," in *Proc. Medicine Meets Virtual Reality (MMVR'6) Conf.*, San Diego, CA, Jan. 19–22, 1998, pp. 385–391.
[19] H. Ng and R. Grimsdale, "Computer graphics techniques for modeling cloth," *IEEE Comput. Graph. Appl.*, pp. 28–41, Sept. 1996.
[20] D. Terzopoulus, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," *SIGGRAPH Proc. Computer Graphics*, vol. 21, no. 4, pp. 205–214, 1987.
[21] M. Bro-Nielsen and S. Cotin, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *EUROGRAPHICS'96*, vol. 15, no. 3, pp. 57–66, 1996.
[22] O. C. Zienkiewicz, *The Finite Element Method*. New Delhi, India: McGraw-Hill, 1979.
[23] K. H. Huebner, E. A. Thornton, and T. G. Byrom, *The Finite Element Method for Engineers*. New York: Wiley, 1995.
[24] A. Pentland and J. Williams, "Good vibrations: Modal dyanmics for graphics and animation," *SIGGRAPH*, vol. 23, no. 3, pp. 215–222, 1989.
[25] Z. Bittnar and J. Sejnoha, *Numerical Methods in Structural Mechanics*. New York: ASCE, 1996.
[26] K. Bathe, *Finite Element Procedures*. Englewood Cliffs, NJ: Prentice-Hall, 1996.
[27] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," *SIGGRAPH*, pp. 55–62, 1995.
[28] A. Witkin, D. Barraff, and M. Kass, "An introduction to physically-based modeling," in *Conf. SIGGRAPH'98*, Orlando, FL.
[29] C. Basdogan, "Force-reflecting deformable objects for virtual environments," in Course Name: Haptics From Basic Principles to Advanced Applications SIGGRAPH '99, 26th Int. Conf. on Computer Graphics and Interactive Techniques, Course No: 38, Los Angeles, CA, Aug., 8–13 1999, [Online]. Available: http://eis.jpl.nasa.gov/~basdogan.
[30] C. Basdogan and M. A. Srinivasan, "Haptic rendering in virtual environments," in *Virtual Environments HandBook*, K. Stanney, Ed., 2001, [Online]. Available: http://www.eis.jpl.nasa.gov/~basdogan.
[31] M. A. Srinivasan and C. Basdogan, "Haptics in virtual environments: Taxonomy, research status, and challenges," *Comput. Graphics*, vol. 21, no. 4, pp. 393–404, 1997.
[32] C. B. Zilles and J. K. Salisbury, "A constraint-based god-object method for haptic display," in *IEEE Int. Conf. Intelligent Robots Systems, Human Robot Interaction, and Co-operational Robots*, vol. 3, 1995, pp. 146–151.
[33] D. C. Ruspini, K. Kolarov, and O. Khatib, "The haptic display of complex graphical environments," in *ACM SIGGRAPH*, July 1997, pp. 345–352.
[34] C. Basdogan, C. Ho, and M. A. Srinivasan, "A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments," in *Winter Annu. Meeting ASME'97*, vol. DSC-61, Dallas, TX, Nov. 16–21, 1997, pp. 77–84.
[35] C. Ho, C. Basdogan, and M. A. Srinivasan, "Modeling of force and torque interactions between a line segment and triangular surfaces for haptic display of 3D convex objects in virtual and teleoperated environments," *Int. J. Robot.*, vol. 19, no. 7, pp. 668–684, 1999.
[36] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy, "Six degree-of-freedom haptic rendering using voxel sampling," in *SIGGRAPH'99*, 1999, pp. 401–408.
[37] C. Basdogan, "Course name: Simulating minimally invasive surgical procedures in virtual environments: from tissue mechanics to simulation and training," in *Conf. Medicine Meets Virtual Reality MMVR'00*, Irvine, CA, Jan. 27–30, 2000, [Online]. Available: http://www.amainc.com/MMVR/MMVR2000pro.html.
[38] F. J. Carter, "Biomechanical testing of intra-abdominal soft tissue," in *Int. Workshop on Soft Tissue Deformation and Tissue Palpation*, Cambridge, MA, Oct. 1998.
[39] F. J. Rosen, B. Hannaford, M. MacFarlane, and M. Sinanan, "Force controlled and teleoperated endoscopic grasper for minimally invasive surgery—experimental performance evaluation," *IEEE Trans. Biomed. Eng.*, vol. 46, pp. 1212–1221, 1999.
[40] S. S. Rao, *The Finite Element Method in Engineering*. New York: Pergamon, 1988.
[41] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C; Art of Computing*. Cambridge, U.K.: Cambridge University Press, 1993.
[42] G. H. Golub and C. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1996.

Cagatay Basdogan received the Ph.D. degree in mechanical engineering from Southern Methodist University, Dallas, TX, in 1994.

Previously, he worked as a scientist at NASA-Johnson Space Center, Houston, Tx, and Northwestern University Research Park, Evanston, IL. He was also a research scientist with the Massachusetts Institute of Technology, Cambridge, for four years. He joined the Jet Propulsion Laboratory (JPL), of the California Institute of Technology, Pasadena, in 1999, where he is a senior member of technical staff. At JPL, he established the Virtual Environments Laboratory. He has written three book chapters, published more than 25 journal and conference articles, and organized sessions and tutorials in the areas of haptic interfaces, virtual environments, computer graphics, man–machine interactions, robotics, medical simulation, and biomechanical modeling of human movement. Currently, he leads the development of a virtual reality system for interacting with a fleet of rovers to plan and schedule planetary robotic missions effectively.

Chih-Hao Ho received the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, in 1999. During his doctoral studies, he developed novel haptic interaction techniques that enable the user to touch and feel 3-D objects in virtual environments.

He is a software engineer with Cambridge Research Associates, McLean, VA. His research interests include haptic rendering, computer graphics, robotics, real-time simulation, and virtual environments. Currently, he develops algorithms for real-time 3-D terrain visualization.

Mandayam Srinivasan received the Bachelor's degree in civil engineering from Bangalore University, India, in 1975, the Master's degree in aeronautical engineering from the Indian Institute of Science, in 1977, and the Ph.D. degree in applied mechanics from the Department of Mechanical Engineering, Yale University, New Haven, CT, in 1984 .

After receiving the Ph.D. degree, he became a member of the research faculty of the Department of Anesthesiology, Yale University School of Medicine, New Haven, CT, where he conducted research on mechanics and mechanisms of the primate sense of touch. In 1987, he joined the Massachusetts Institute of Technology (MIT), Cambridge, and founded the Laboratory for Human and Machine Haptics, informally known as the Touch Lab. He is currently the Director of MIT's Touch Lab and a principal research scientist with the Department of Mechanical Engineering and the Research Laboratory of Electronics, also at MIT. His research is focused on haptic computation, cognition, and communication in humans and machines, particularly to enhance human-machine interactions in virtual environment systems.