Computationally Efficient Techniques for Real Time Surgical Simulation with Force Feedback

Jung Kim, Suvranu De, Mandayam A. Srinivasan
Laboratory for Human and Machine Haptics (The Touch Lab)
Department of Mechanical Engineering and Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
jungkim@mit.edu, srini@mit.edu

Abstract

In this paper, we present computationally efficient algorithms for the real time simulation of minimally invasive surgical (MIS) procedures. To develop a surgical simulator for training medical personnel, due to high computational speed required for realistic simulation in real time, there is a fundamental trade-off between realism and processing speed of the simulator. Our research focuses on how we can optimally utilize computing resources in the presence of constraints related to real time performance. We first present a novel approach to rapid collision detection so as to maximize the available time for computing collision response. We then present a real time tissue model that computes visual rendering of deformations and haptic rendering of interaction forces by using a newly developed meshless numerical scheme. By integrating these techniques with a force-feedback device and a visual display connected to PC, we simulate a specific procedure (palpation) with update rate of 1KHz for force and 30Hz frame rate for graphics.

1. Introduction

Minimally invasive surgical procedures (MIS) are being widely used due to their significant advantages over traditional open surgeries. A major advantage of MIS procedures is that patients recover quickly, dramatically shortening the duration of stay in the hospital. However, the surgeons have to face several difficulties never experienced in open surgeries such as poor depth perception, limited field of view, improper hand-eye coordination and limited force feedback from instruments [1]. Therefore, MIS surgeons need long learning times and repetitive practice to reach a necessary skill level. But their opportunities to practice procedures are limited.

This calls for a training system for surgical residents or medical students. Similar to flight simulators, surgical simulators promise a safer and cheaper way of training [2] [3]. The surgeons can be trained in computer-generated environments where they would interact with 3-dimensional realistic organ models. The simulator updates multimodal information seamlessly and responds to users' actions in real time. While the graphical frames are typically updated at around 30 - 40Hz, the haptic information needs to be updated at a rate of about 500 Hz – 1KHz for stable haptic interaction [4].

The current computational speed of even high-end PCs is not sufficient for simulating procedures perfectly in real time. Therefore real time computing is one of the main concerns in the development of the simulators. Our paper presents new computationally efficient approaches for component tasks so as to optimally manage the limited computational resources during simulation.

2. Run Time Processes in Surgical Simulation

Once the simulator starts, it mainly repeats two processes within one cycle time. The first process is the collision detection part that detects occurrence of contacts between the virtual tools and the virtual organs. The second one is the collision response that computes proper response of organs such as deformation fields and interaction forces. Fig. 1 shows these two important processes during run time. At every cycle controlled by an external timer, the simulator reads the tip position of the force-feedback device and checks for tool-tissue collisions. If collision occurs, the tissue model computes the response of the tissues in terms of the deformation fields as well as the interaction forces.

The deformation field is displayed graphically and haptic interface devices display the interaction forces to the users in real time. In this context, for a given computer, the processing speed can be denoted by the computing time $\mathbf{C}_{\mathbf{Limit}}$ available for each cycle. If **Cd** is the computing time for collision detection and **Cr**, then the sum of **Cd** and **Cr** cannot exceed $\mathbf{C}_{\mathbf{Limit}}$.

0-7695-1489-8/02 $17.00 © 2002 IEEE

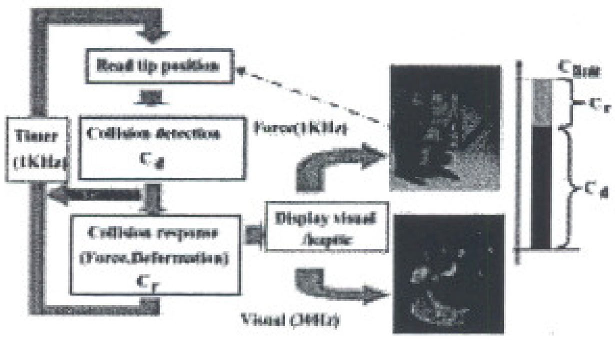

Figure 1. Two important tasks in real time medical simulations, collision detection Cd and collision response Cr

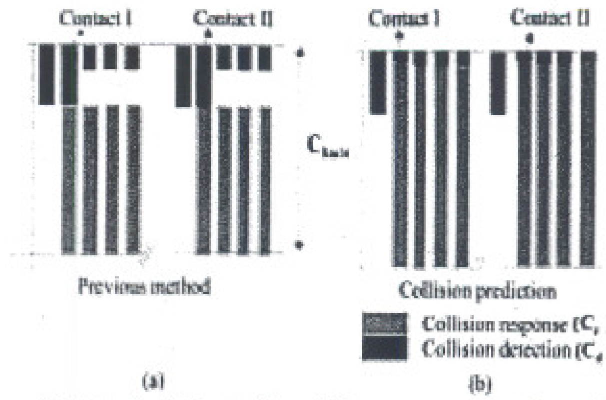

Figure 2. Schematic of the processing time in surgical simulation showing two successive contacts I and II separated by break in contact. The vertical axis represents the processing time and each column represents the one cycle time.

3. Collision Detection

The objective of collision detection in virtual environments can be summarized as finding the occurrence and location of collision points between the virtual organs and the tools. Usually, the collision points are determined by tool trajectories and the geometrical models. There is substantial literature in computational geometry, computer graphics and virtual environments on fast determination of collision points [5, 6]. In most of the techniques, the objects are partitioned into a group of small convex cells and stored in the database hierarchically. For example, the H-COLLIDE algorithm uses oriented bounding box hierarchy trees [7]. The simulator finds intersection points with these small cells and lines representing tool trajectory. Ray-based rendering, developed previously in our laboratory, models the tool as a line instead of a point and is capable of handling tool-tissue interactions not only at the tool tip, but also its sides [8]. Currently available techniques such as bounding box hierarchy are capable of finding the collision points for models containing 20-30 thousand triangles within a millisecond.

3.1 Collision Prediction

The collision detection, however, is not the only part within one cycle time. The simulator should also complete the computation of collision response part as in Fig. 1. Fig.2 (a) shows these processing timings of each sampling time during tool-tissue interaction. The collision detection process limits the maximum processing time of the collision response, which determines the realism of tissue behavior simulation.

Therefore, it is required to consider the relationship of both processes in one cycle time as well as the consideration of individual processing times. In Fig. 2(a), we observe that the maximum processing time for collision detection always occurs at the first contact because of the need for global search. Second, this initial high **Cd** limits available **Cr** for the rest of cycles until contact breaks. Since there is no collision response prior to tool-tissue contact, if we can utilize the $\mathbf{C}_{\mathbf{Limit}}$ available during this task phase to predict the collision point, we can subsequently increase **Cr** as shown in Fig. 2 (b).

The algorithm of collision prediction is as follows. Since the user's hands holding the force-feedback devices have low frequency motion (of the order of 10Hz or less) compared to the sampling frequency of the system (100 to 1000Hz), the approach direction of the tools can be computed from the trajectories. In other words, the simulator can predict subsequent tool positions from a set of previous positions. We refer to the vector connecting the current position of tool with its previous position as the "tool path vector". Since the physiological tremor in the hands transferred into the tool trajectory appears like noise, low-pass filtering is required to determine the "mean tool path" (mean of several sequential tool path vectors) along the tool approach direction.

At the beginning of each cycle time, the simulator computes the intersection points between the mean tool path and the organs. These points are located on the surface of the objects and move along the mean tool path. Fig. 3 shows the concept of the mean tool path and the predicted collision point. After the tools are located within a certain distance (5mm-10mm), the motion of the tool is assumed to be along a straight line to the organs.

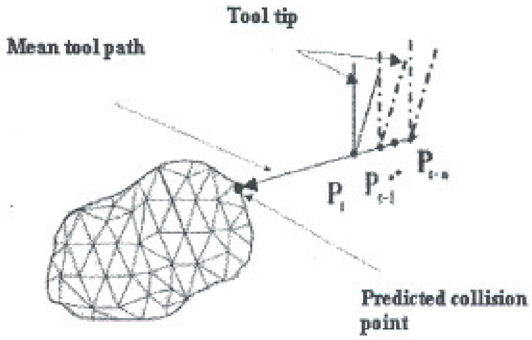

Figure 3. The concept of mean tool path and computation of the predicted collision point.

After that, the simulator only checks the distance from the organ to the tool until the tool reaches the predicted collision points. The computation time reduces to one distance computation and one comparison (Boolean computation) from the time of collision detection algorithms, which normally increases with complexity of organ geometry. Table. 1 summarizes the results. In case of typical collision detection, the computing time (number of overlap tests) increases with respect to number of polygons. Using a workstation with 600MHz Pentium III processor, for a 35K polygon model 0.3 millisecond is required to obtain the collision points when we used the fast point based collision detection algorithm in [5]. The collision prediction method, however, only requires microseconds for the computations at and after the first contact, thus allowing more time for collision response. This implies that the tissue behavior can be simulated more realistically, since increasing realism generally requires increasingly complex tissue models.

Another advantage of the collision prediction is that we can reduce real time computational burden by running the collision detection part with a slower update rate. In a technique without collision prediction, the collision detection part needs to be performed at the force update frequency to prevent latency of the collision response. But with the collision prediction technique, it is not necessary to perform the collision detection at the force update frequency because the collision point is determined prior to the first collision. Given that the hand speed of the user is limited especially in surgical tasks, we can run the collision prediction part at the slower frequencies. It also means the simulator can handle more complicated organ models and multi-organ collision detections without the major constraint of the required force update frequency.

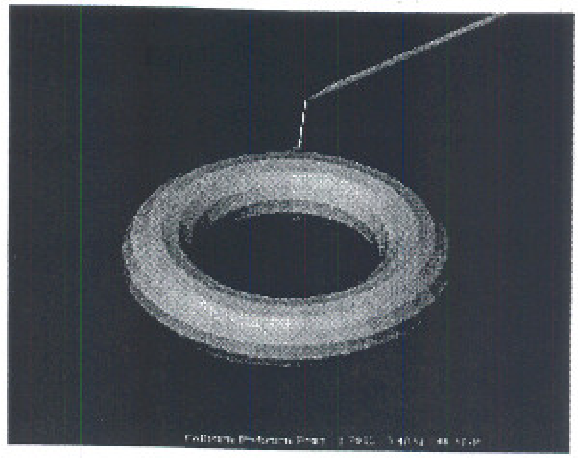

Figure 4. The collision prediction for a 3D object. The small ball on the surface represents the predicted point. The white line is the mean tool path.

Our collision prediction technique is not dependent on any specific collision detection algorithm so that we can select any kind of algorithm to be suitable for main requirements of target applications such as speed or memory usage.

3.2 Hand tremor removal

The prediction of collision points starts from the determination of the tool path vector;

$$\vec{\mathbf{P}}_{\text{tool path}} = \vec{\mathbf{P}}_{\text{current}} - \vec{\mathbf{P}}_{\text{previous}} \tag{1}$$

Although this method is very easy to be implemented and computationally efficient, the vibration of the tool imposed by physiological tremor affects the accuracy of the predicted collision point. According to literature, physiological tremor has a bandwidth of about 7Hz to 12 Hz [9]. To remove this, we take several points from the tool trajectory and average these points as follows.

$$\vec{\mathbf{P}}_{\text{mean tool path}} = \frac{1}{\mathbf{m}}\left(\sum_{i=1}^{m} \vec{\mathbf{P}}_{\text{current}} - \vec{\mathbf{P}}_{i}\right) \tag{2}$$

$\vec{\mathbf{P}}_i$ is the i-th previous position in the tool trajectory and **m** is the total number of previous points. This procedure effectively acts as a low-pass filter applied to the tool path vectors and it improves accuracy of the prediction from more than 5 mm deviation to within 1 mm.

Table 1. Computing time at the first collision[1]

Number of polygons	500	2500	17K	35K
Typical Collision detection	Number of overlap tests			
	9	12	14	16
Collision prediction	1 distance computation + 1 comparison (No overlap test)			

4. Collision Response: Simulation of Soft Tissues

Collision response in surgical simulation usually involves simulation of soft tissues at lower frequency and rendering of forces at higher frequency. It computes realistic or physically based behavior of soft tissues in response to tool interaction. Among various techniques found in literature, the finite element (FEM) and Boundary Element (BEM) methods are widely used in engineering analysis [10]. Several researchers have applied these methods for modeling deformable objects, including soft tissues for real time surgical simulation [11-13] [14]. In spite of accuracy and robustness, FE-based models are difficult to be applied in real time simulation because of high computational overhead and computationally expensive remeshing. To achieve real time performance, the current FE-based real time techniques highly rely on precomputation. Although they work well in initially well-established meshes, the changes of meshes during run time cause problems such as instability or stopping of system. Also, tissue properties based on old meshes should be recomputed at least locally from newly generated meshes without interrupting the simulation. With current computers, it is very difficult to do this process within a few millisecond cycle time.

We therefore take a different approach that does not require meshes for simulation of soft tissues. We apply a newly developed "meshless" technique named "the Method of Finite Spheres "(MFS) [15]. It uses a set of points instead of meshes for solving governing equations. When a surgical tool touches the tissues, a set of points is sprinkled locally around the tool tip and a sphere with a finite radius is located at each sprinkled point. Shape functions or interpolation functions are used to approximate deformation fields;

$$\mathbf{u_h}(\mathrm{x}) = \sum_{J=1}^{N} \mathbf{H_J}(\mathrm{x})\alpha_{\mathbf{J}} = \mathbf{H}(\mathrm{x})\mathbf{U} \tag{3}$$

[1] In case of typical overlap test, one test requires 15 comparisons and 60 addition/subtractions and 81 multiplications and 24 absolute value computations[6].

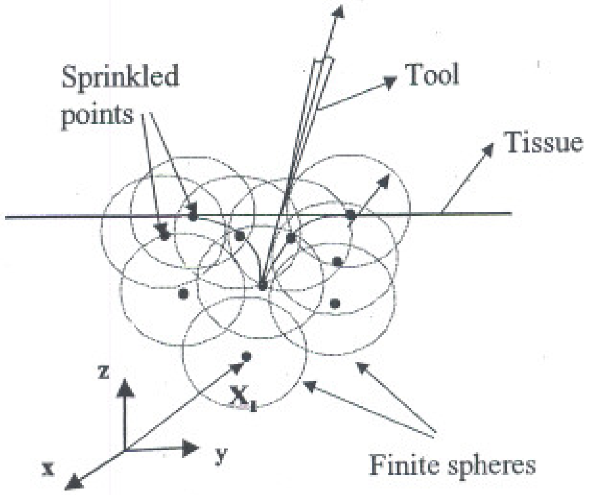

Figure 5. The concept of the method of finite spheres (MFS).

The matrix $\mathbf{H(x)}$ is the nodal shape function matrix corresponding to the J-th node and $\mathbf{h_J}$ is the diagonal term of $\mathbf{H(x)}$.

$$\mathbf{h_J}(\mathrm{x}) = \mathbf{W_J}(\mathrm{x})\mathbf{P}(\mathrm{x})^{\mathrm{T}}\mathbf{A}^{-1}(\mathrm{x})\mathbf{P}(\mathbf{x_J}) \tag{4}$$

$$\mathbf{A}(\mathrm{x}) = \sum_{I=1}^{N} \mathbf{W_I}(\mathrm{x})\mathbf{P}(\mathrm{x}_I)\mathbf{P}(\mathrm{x}_I)^{\mathrm{T}} \tag{5}$$

$\mathbf{P(x)}$ is a vector containing polynomials and we have chosen $\mathbf{P(x) = [1,x,y,z]}$. $\mathbf{W_J}$ is a weighting function at node **J**.

Under the assumption of linear elastic behavior, the discrete equations using a point collocation technique are given as [16].

$$\mathbf{KU = f} \tag{6}$$

where **K** is the stiffness matrix and **f** is the vector containing external forces. The size of the stiffness matrix is 3N x 3N where **N** is the number of modal points. Since **K** is built locally, the size of the matrix is not large. A point to note is that even though linear elastic behavior is assumed, the technique is very general (no precomputations are required) and a consistent linearization scheme may be adopted for iterative solution of problems involving nonlinear constitutive behavior.

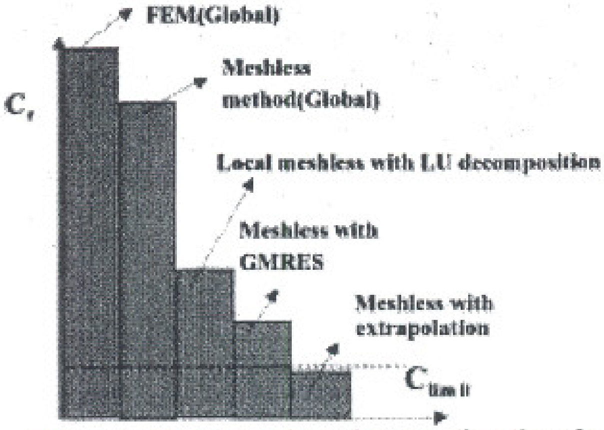

Figure 6. The reduction of computing time for the real time performance

Table 2. Computation times for FEM and MFS

	FE model	Meshless model
No. of nodes	4045	34
Model time (sec)	6.44	0.0021
Solution time (sec)	11.88	0.0015
Total time (sec)	18.32	0.0036

The local deformation assumption may be justified as follows. First, the viewport through which the surgeons view tissue deformation during the operation has limited field of view. Second, for given boundary conditions, the force behavior of soft tissues is determined by mainly local properties around tool-tissue collision points. Therefore, the local model can be used in the simulator as long as the deformations from the model provide realistic feeling to the users. The local model can also simulate complicated behavior of tissues such as due to material nonlinearity.

We have performed a comparison of computational times when the finite element technique as well as the MFS is used to solve the problem of indenting a hemisphere with 10 cm diameter. Table. 2 summarize the result of the comparisons when the local deformation of the two models matched well. The total time is assumed to be composed of the time to generate the stiffness matrix and time to solve the set of 1system of equations. Since the FE technique generates the discrete equations of the entire domain, it has about 4000 nodal points. In the MFS, only 34 points are used in the vicinity of the indenter.

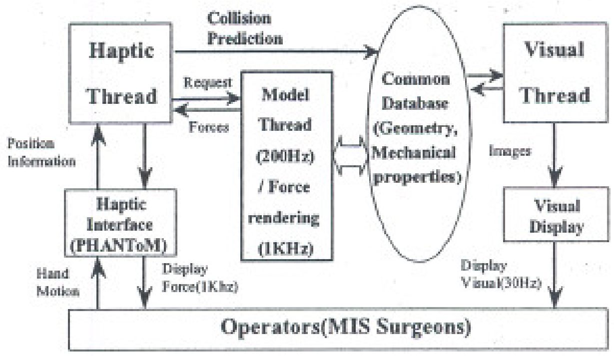

Figure 7. The software architecture of the surgical simulator.

While the FEM solution takes more than 10 seconds, the MFS solution takes only a few milliseconds. This fact explains why FEM model requires elaborate precomputation and the local MFS model does not. Certainly, the point collocation based MFS technique is much less accurate than the Finite Element technique in regions far away from the tool tip.

5. Implementation Issues

As mentioned before, the simulator should complete all real time processes within the time limit, usually a few milliseconds. Actually, the original form of the MFS [15] generates global solutions and computing time is as expensive as FEM (Fig. 6).

We simplify the MFS technique so that we are capable of conveying realism of the local deformation by removing computationally expensive numerical integration and using LU decomposition [17]. Moreover, GMRES [18], an iterative method for matrix inversion is used to reduce the solution time. Consequently, the reaction forces are computed at 200 Hz and an extrapolation technique is used to update the forces at 1 kHz frequency. Fig. 6 shows our strategy of reducing computing time to achieve real time performance.

Fig. 7 shows a schematic of multithreading software architecture used for MIS simulation. Since haptic and graphic loops have different frequencies, we design two independent threads running at different frequencies. These two threads share the information of organs such as geometry and material properties in a common database. We also design an independent thread to compute force and deformation fields from the model. It computes the response of tissues at about 200Hz. The haptic loop sends requests to the model thread and obtains extrapolated forces to display them at 1KHz. The model thread updates the visual information for the graphic loop running at 30Hz frame rate.

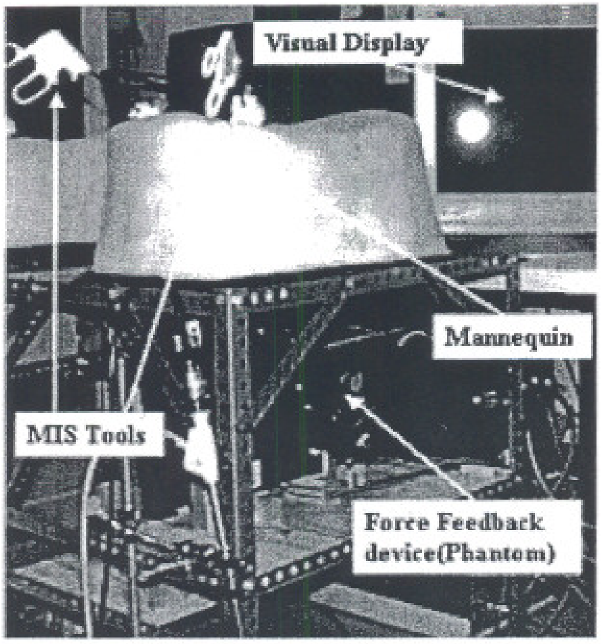

Figure 8. The setup for MIS procedure simulation

To compute the reaction forces to be delivered to the force feedback device, the tool tip is modeled as having point interaction with the tissues. The node placed at the interaction point bears the applied displacement $\mathbf{U}_{tooltip}$ and the stiffness matrix in (6) may be partitioned as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \tag{7}$$

corresponding to a partitioning of the vector of nodal parameters as $\mathbf{U} = [\mathbf{U}_{tooltip} \ \ \mathbf{U}_b]^T$ where $\mathbf{U}_b$ is the vector of unknowns which can be obtained from

$$\mathbf{U}_b = -\mathbf{K}_{bb}^{-1}\mathbf{K}_{ba}\mathbf{U}_{tooltip} \tag{8}$$

The force sent the force feedback device is

$$\mathbf{f}_{tooltip} = \mathbf{K}_{aa}\mathbf{U}_{tooltip} + \mathbf{K}_{ab}\mathbf{U}_b \tag{9}$$

The forces are updated at 1KHz by using extrapolation scheme.

It should be noted that the point-based interaction technique used here does not handle contact between the organ and side of the probe. But earlier work from our laboratory, Ray-based rendering, is capable of computing both forces and torques arising from both tool tip and side contacts.

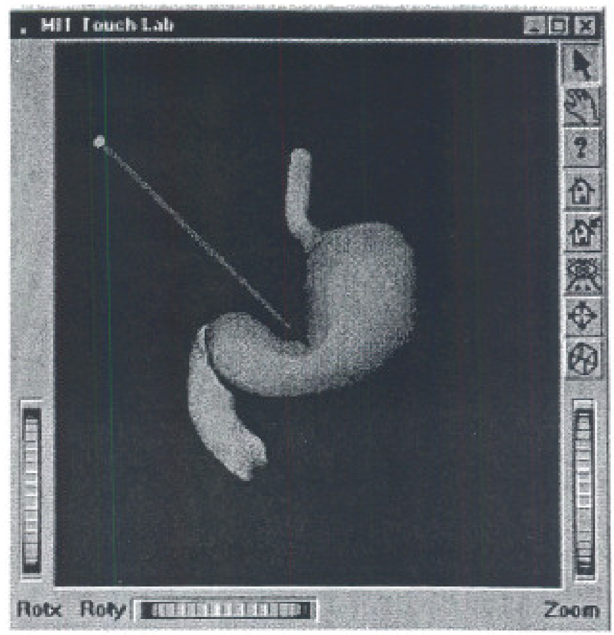

Figure 9. A snapshot of the palpation simulation.

6. Example of the Simulation

Fig. 8 shows our setup to simulate MIS procedures. It uses two MIS tools connected to Phantom force feedback devices, an abdominal mannequin and a visual display to provide realistic environments to the users.

The technical components described previously have been integrated into a palpation MIS simulator. The material properties are obtained from in-vivo experiments conducted by colleagues in our laboratory. The Young's modulus of the stomach is estimated to be around 10 kPa. The geometric model of the stomach has 20,000 triangles. Fig. 9 shows a snapshot of the stomach palpation simulation. The simulator has been implemented on Windows NT and dual Pentium 900MHz processors with one PHANToM force-feedback device [19, 20] from SensAble Technologies. The simulator has 1KHz force update frequency and 30 Hz visual frame update rate.

7. Concluding Remarks

In this paper, we have presented computationally efficient algorithms for the development of minimally invasive surgical simulators for training medical personnel. Various techniques for tool-tissue collision detection, and simulation of tissue behavior in real time have been described along with computational architecture for graphical and haptic rendering of MIS procedures. Although we focus on the simulation of MIS procedures, the proposed techniques are general and can be used in other types of medical simulations.

8. References

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

[2] C. Baur, D. Guzzoni, and O. Georg, "Virgy: A Virtual reality surgical trainer and force feedback based endoscopic surgery simulator", Medicine Meets Virtual Reality(MMVR 6), 1998.

[3] C. Bosdogan, C. Ho, and M. A. Srinivasan, "Virtual Environments for Medical Training: Graphical and Haptic Simulation of Laparoscopic Common Bile Duct Exploration", *IEEE/ASME Transactions on Mechatronics*, vol. 6, pp. 269-285, 2001.

[4] M. A. Srinivasan and C. Basdogan, "Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges", *Computer & Graphics*, vol. 21, pp. 393-404, 1997.

[5] C. Ho, C. Basdogan, and M. A. Srinivasan, "Efficient Point-Based Rendering Techniques for Haptic Display of Virtual Objects", *Presence*, vol. 8, pp. 477-491, 1999.

[6] M. Lin and S. Gottschalk, "Collision Detection between Geometric Models: A survey", Proc. of IMA Conference on Mathematics of Surfaces, 1998.

[7] A. Gregory, M. Lin, S. Gottschalk, and R. Talor, "H-COLLIDE: A Framework for Fask and Accurate Collision Detection for Haptic Interaction", IEEE Virtual Reality Conference, 1999.

[8] C. Bosdogan, C. Ho, and M. A. Srinivasan, "A Ray-Based Haptic Rendering Technique for Displaying Shape and Texture of 3D Objects in Virtual Environments", ASME Winter Annual Meeting, 1997.

[9] L. Jones, "Dextrous Hands: Human,Prosthetic and Robotic: A Survey", *Presence*, vol. 6, pp. 29-56, 1997.

[10] K. J. Bathe, *Finite Element Procedures*: Englewood Cliffs, NJ, Prentice Hall, 1996.

[11] S. Cotin, H. Delingette, and N. Ayache, "Real-time elastic deformations of soft tissue for surgery simulation", *IEEE Trans. On Visualization and computer graphics*, vol. 5, pp. 62-73, 1999.

[12] M. Bro-Nielsen, "Finite Element Modeling in Surgery Simulation", *Proceeding of IEEE*, vol. 86, pp. 490-503, 1998.

[13] D. James and D. K. Pai, "ArtDefo, Accurate Real Time Deformable Objects", Computer Graphics (ACM SIGGRAPH 99 Conference Proceedings), 1999.

[14] D. L. James and D. K. Pai, "A Unified Treatment of Elastostatic Contact Simulation for Real Time Haptics", *www.haptics-e.org*, vol. 2, 2001.

[15] S. De and K. J. Bathe, "The Method of Finite Spheres", *Computational Mechanics*, vol. 25, pp. 329-345, 2000.

[16] S. De, J. Kim, and M. A. Srinivasan, "A Meshless Numerical Technique for Physically Based Real Time Medical Simulations", Proceeding of MMVR 2001, 2001.

[17] W. H. Press, B. P. Flannery, and et al., *Numerical Recipes in C*. Cambridge: Cambridge University Press, 1987.

[18] Y. Saad and M. H. Schultz, "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems", *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856-869, 1986.

[19] J. K. Salisbury and M. A. Srinivasan, "Phantom-based haptic interaction with virtual objects", *IEEE Computer Graphics and Applications*, vol. 17, pp. 6-10, 1997.

[20] T. Massie and J. K. Salisbury, "The PHANToM Haptic Interface, A Device of Probing Virtual Objects", Proceeding of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environments and Teleoperator systems, 1994.