Suvranu De
Yi-Je Lim*
Advanced Computational Research Lab
Rensselaer Polytechnic Institute
Troy NY 12180

Muniyandi Manivannan
Mandayam A. Srinivasan
Touch Lab
Massachusetts Institute of Technology
Cambridge, MA 02139

Physically Realistic Virtual Surgery Using the Point-Associated Finite Field (PAFF) Approach

Abstract

The generation of multimodal virtual environments for surgical training is complicated by the necessity to develop heterogeneous simulation scenarios such as surgical incision, cauterization, bleeding, and smoke generation involving the interaction of surgical tools with soft biological tissues in real time. While several techniques ranging from rapid but nonphysical geometry-based procedures to complex but computationally inefficient finite element analysis schemes have been proposed, none is uniquely suited to solve the digital surgery problem. In this paper we discuss the challenges facing the field of realistic surgery simulation and present a novel point-associated finite field (PAFF) approach, developed specifically to cope with these challenges. Based upon the equations of motion dictated by physics, this technique is independent of the state of matter, geometry and material properties and permits different levels of detail. We propose several specializations of this scheme for various operational complexities. The accuracy and efficiency of this technique is compared with solutions using traditional finite element methods and simulation results are reported on segmented models obtained from the Visible Human Project.

1 Background

Surgical teaching has been based traditionally on the preceptor or apprenticeship model, in which the novice surgeon learns with small groups of peers and superiors, over time, in the course of patient care. The operating room and the patient, however, comprise the most common, the most readily available, and often the only setting where hands-on training takes place. The novice surgeon acquires skills by first observing experienced surgeons in action and then by progressively performing, under varying degrees of supervision, more of the surgical procedures, as his/her training advances and his/her skill level increases.

The drawbacks of this traditional training approach have been underscored in the report by the Institute of Medicine, entitled "To Err Is Human: Building a Safer Health System," which points out that about 100,000 deaths per

*Correspondence to: limy@rpi.edu

Presence, *Vol. 15, No. 3, June 2006, 294–308*

year in the United States occur as a result of medical errors, making it the eighth leading cause of death.

While the importance of training has been established, the best or most effective method has not. Animal models are not considered good substitutes for human patients due to fundamental differences in anatomy and tissue consistency. Moreover, the use of animals for training purposes is expensive and controversial and requires dedicated facilities, special care, and housing. Moreover, only a limited number of trainees can practice on a single animal.

Therefore, relatively inexpensive inanimate training methodologies such as video tool box trainer are becoming more popular. The Society of American Gastrointestinal Endoscopic Surgery (SAGES) is the first organization that has officially adopted an inanimate training methodology and established the Fundamentals of Laparoscopic Surgery (FLS) course (Peters et al., 2003). The FLS has a basic instructional guide (CD ROM-based educational module) and a manual skill station.

However, both animate and inanimate training techniques suffer from the same drawbacks, the most important being the need for an instructor/supervisor, nonstandardized methods of feedback, inadequate provision to practice for rare medical conditions, and lack of well-defined subjective methods of performance evaluation. These training methods and trainers have access to a limited number of parameters and often time is the only performance measure.

These drawbacks have prompted the development of virtual environment (VE) based surgery simulators where the human user is able to interact with three-dimensional virtual models of organs using his/her sense of vision as well as actively manipulate them using his/her sense of touch through a haptic interface device such as a PHANToM. It is well recognized that VE-based simulation systems offer a unique way of objectively assessing performance while imparting training, providing real-time feedback, tracking the trainee's learning curve over extended periods of time, and providing quantitative scores while, at the same time, offering summative evaluation during examinations (Satava et al., 2001).

Several groups have been or are currently involved in research and development of multi-modal surgical simulation including the Stanford University Medical Media and Information Technologies Center (SUMMIT) and Biocomputation Research Center at Stanford University, the Center for Integration of Medicine and Innovative Technology (CIMIT) at Harvard Medical School, the Human Interface Technology Laboratory and Biorobotics Laboratory at University of Washington, the Center for Surgical Simulation at the University of Colorado, and the National Capital Area Medical Simulation Center at Uniformed Services University.

Commercial forays into the realm of surgery simulation include the Anastomosis Simulator (Boston Dynamics, Inc.), the Surgical Workbench (Hand Immersive, Inc.), the CathSim & PreOp Simulators (Immersion Medical, Inc.), the Patient Anesthesia Mannequins (Laerda, Inc., and MedSim, Inc.), the Melerit Medical Simulator (Melerit AB), the MIST & VIST (Mentice, Inc.), the Limb Trauma Simulator (MusculoGraphics, Inc.), the Reachin Laparoscopic Trainer (RLT) and Core (Reachin Technologies AB), the Simbionix Mentor Simulators (Simbionix Ltd.), the Telepresence Surgery System (SRI, International), and the LapSim (Surgical Science, Inc.).

However, in spite of significant advances in this area, VR-based surgery simulators are still not quite popular with the medical community. In fact, in a study (Torkington, Smith, Rees, & Darzi, 2001) comparing VR-based (using MIST from Mentice, Inc.) with conventional training, current computer-based simulators did not exhibit significant advantages in terms of performance of standardized assessment exercises. This failure is largely attributable to a lack of complete understanding of the complexity of the tasks involved.

To achieve a high degree of realism, the virtual environments need to support the following: (1) heterogeneous scenes composed of different states of matter (solids, liquids, and gases); (2) complex geometry and material properties of objects within the scene; (3) dynamic and real-time interaction (palpation, cutting, etc.) between virtual objects and tools physically manipulated by the user; and (4) multimodal (visual, auditory, and haptic) rendering of the results to the user. The complexity resulting from the four requirements above necessitate the dedicated techniques of modeling, simulation, and rendering.

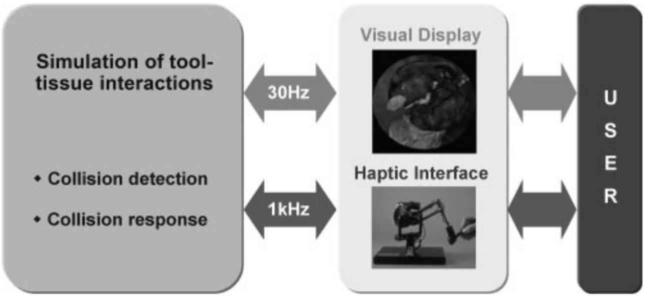

Figure 1. *A schematic of the surgery simulation process.*

In this paper we discuss the development of such techniques. In Section 2 we present some of the related literature and point out the drawbacks of current simulation strategies. In Section 3 we present the point-associated finite field (PAFF) approach as a technique uniquely adapted to modeling surgical procedures and outcomes. Various specializations are provided for models of varying degrees of complexity and states of matter (solids and fluids). In Section 4 the software architecture is presented. Finally, in Section 5, we provide realistic simulation examples, some of which are based on segmented image data from the Visible Human Project.

2 Related Work

The two most important issues in developing a multimodal surgery simulator are *collision detection* and *collision response* (Figure 1). In collision detection, one needs to ensure that there has indeed been a contact between the surgical tool and the objects in the scene. Collision detection problems have been extensively studied in many different research communities including robotics, computer graphics, computer-aided design, and computational geometry. Most collision detection algorithms approximate the objects in the scene using bounding volumes such as spheres (Quinlan, 1994; Hubbard, 1995; Palmer & Grimsdale, 1995), oriented bounding boxes (OBBs) (Gottschalk, Lin, & Manocha, 1996) and axis aligned bounding boxes (AABBs) (Zachmann & Felger, 1995; Zachmann, 1997).

Generally, the collision model in haptic rendering defines the graphical description of the surgical tools and the nature of tool–tissue interactions. The virtual tool may be modeled as a point (Ho, Basdogan, & Srinivasan, 1999), a ray (Basdogan, Ho, & Srinivasan, 1997), or as a 3D object (Ho, Sarma, & Adachi, 2001).

Once collision has been detected, the interaction of the surgical tool with the virtual organ model is computed during collision response. The deformed organ model is displayed on the computer screen. The reaction forces are fed back to the user through the haptic interface device(s).

One way to categorize the literature on the computation of deformation and the display of deformable objects is according to whether the technique is physics-based or is based on purely geometric considerations. In *geometry-based techniques,* the object or surrounding space is deformed by manipulating vertices or control points. These techniques are relatively faster and easier to implement. However, they do not capture the physics of the problem. The concept of *free form deformation* was originally suggested by Sederberg and Parry (1986) and extended by Hsu, Hughes, and Kaufman (1992) to direct free-form manipulation. The extension of this technique to haptic display of deformable objects with application to medical simulation (Basdogan, Ho, & Srinivasan, 1998), computer-aided design (CAD; Gupta, Sheridan, & Whitney, 1997; Gregory, Ehmann, & Lin, 2000; Jayaram, Vance, Gadh, Jayaram, & Srinivasan, 2001), and haptic sculpting (Dachille, Qin, Kaufman, & El-Sana, 1999; Edwards & Luecke, 1996) can be found in literature.

The *physics-based techniques,* on the other hand, aim to model the physics involved in the motion and dynamics of interactions. The current trend in the computer graphics literature is to use a surface/volume representation of the object and couple it with a mass-spring model or mesh-based computational scheme. One of the simplest physics-based models, and thus the most likely to achieve real-time interactivity, is the mass-spring elastic network. Mass-spring systems 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 (see Witkin, Baraff & Kass, 1996 for implementation details). This technique has been used extensively by computer graphics researchers in simulating soft tissue and cloth behavior (Cover et al., 1993; Terzopoulos & Waters, 1990; Ng & Grimsdale, 1996). Swarup (1995) demonstrated the application of this technique to haptic simulation of deformable objects. Recent work (Desbrun, Shroder, & Barr, 1999; Kang, Choi, & Cho, 2000) has focused on reducing computational cost per time step while preserving the stability of this system.

However, for the purpose of deformation modeling, these models suffer from many disadvantages. It is difficult, and sometimes impossible, to determine the parameters of hundreds of thousands of springs, dampers, and masses to represent the global behavior of the tissue especially if nonlinear and/or viscoelastic behavior is to be captured. It is difficult to enforce global properties like incompressibility when using such models and the problem is exacerbated when one tries to use a relatively few particles to reduce computational time. Relatively stiff springs are necessary to model hard tissues, jeopardizing the stability of the solution scheme, and requiring the numerical temporal integrator to take minute time steps. Finally, anisotropic distribution of mass points necessitates fine-tuning for individual organ geometry, difficulty in controlling the variation of forces and deformations across the geometry as well as integrating tissue properties into the model.

A few researchers have proposed the use of more robust but expensive finite element analysis procedures (Bathe, 1996) as an alternative to mass-spring models. Drastic modeling simplifications have to be made to implement real-time finite element models with haptic feedback (see Bro-Nielsen & Cotin, 1996; Cotin, Delingette, & Ayache, 1999; Girod, Keeve, & Girod, 1996; Ayache, Cotin, & Delingette, 1998; Berkley et al., 2000; Berkley, Turkiyyah, Berg, Ganter, & Weghorst, 2004; De & Srinivasan, 1998, 1999; Basdogan, Ho, & Srinivasan, 2001; Wu, Downes, Goktekin, & Tendick, 2001; Picinbono, Delingette, & Ayache, 2003; Masutani et al., 2004; Choi, Sun, & Heng, 2004).

In spite of accuracy and robustness, finite element techniques suffer from certain drawbacks in real time simulation. First, the need for numerical integration and volumetric meshing results in a slower-than-real-time performance unless extensive precomputations are performed. Furthermore, the contact between tool and tissues must occur only at nodal points. Hence, for a smooth visual and haptic display, a fine mesh needs to be utilized, resulting in extensive memory usage and high computational overhead. Large deformations and nonlinear response of tissues cause the finite elements to behave badly or totally fail unless remeshing is performed. Finally, change of topology, for example, during the simulation of surgical cutting, necessitates remeshing, which destroys any precomputed data, increases the number of computations on the fly, and seriously degrades real time performance.

There are, however, positive features of both mass-spring and finite element analysis schemes. Mass-spring models offer simplicity and speed and do not suffer from the deleterious effects of mesh distortion. Finite element schemes are used to solve the partial differential equations that govern the deformation and motion of soft tissues and only a limited number of empirically determined parameters are necessary. For example, to simulate the response of linear elastic (isotropic) tissue behavior, only two independent parameters (Young's modulus and Poisson's ratio) need to be determined from experiments.

Hence, an ideal combination of mass-spring and finite element-based techniques is desirable. Such an "ideal" scheme should solve the governing partial differential equations (as a finite element scheme), but not suffer from any of the problems associated with a mesh (e.g., mesh distortion and remeshing when large deformations or surgical cutting have to be modeled). It should be very flexible, in the sense that it should allow arbitrary local refinement and multiresolution capability to zoom into regions of "action" without having to unduly refine the discretization over the entire computational domain.

Finally, the ideal scheme should allow the simulation of matter, irrespective of its state of solid, liquid, or gas, within a single computational framework. This is essential since one needs to simulate dissection, bleeding, and, possibly, smoke generation within the same scenario without having to switch between various modeling schemes stitched together through tenuous nonphysical links. Neither finite element nor mass-spring

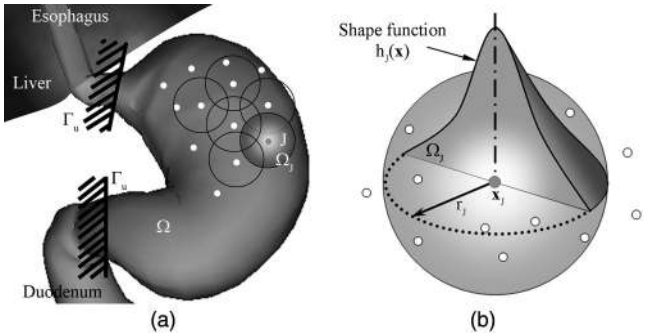

Figure 2. *The point-associated finite field (PAFF) technique. (a) A general three-dimensional body (e.g., stomach) discretized using a set of nodes. Each node has an associated spherical influence zone. (b) The approximation function $h_J(x)$ at node J is "bell shaped" and is nonzero only on the spherical influence zone of radius r_J centered at node J.*

models have the ability to provide such a general framework for the modeling of heterogeneous media.

3 The Point-Associated Finite Field (PAFF) Approach

With the specifications of the "ideal" computational scheme laid out in the previous section, we have developed the *point-associated finite field* (PAFF) approach. In this method, matter, irrespective of its state, is represented as a collection of particles or nodes that serve as the computational primitives (Figure 2), much like mass-spring models. But unlike mass-spring models, the governing partial differential equations are solved. However, the method is "meshless" since no direct link exists between the computational particles.

The particles possess a finite "region of influence" which smears out their effects and coordinates their motions during simulation. Since no mesh is used, none of the problems associated with a mesh are encountered. We will show shortly that it is straightforward to develop a multiresolution strategy whereby one can zoom into regions of interest without jeopardizing the computations. Finally, since the particles are not constrained as in a mesh, they can slide past each other, much like clouds, and this allows the modeling of very large deformations as well as surgical cutting without difficulty; they can flow like a liquid or escape like a gas. In the following paragraphs we will provide some technical details regarding the method.

In PAFF, the approximation u_h of the vector of variables u (i.e., the *x, y,* and *z* components of the displacement field), using *N* particles, may be written as:

$$\mathbf{u}_h(\mathbf{x}) \approx \sum_{J=1}^{N} \mathbf{H}_J(\mathbf{x})\boldsymbol{\alpha}_J = \mathbf{H}(\mathbf{x})\mathbf{U} \tag{1}$$

where $\boldsymbol{\alpha}_J = [u^J\ v^J\ w^J]$ is the vector of nodal unknowns at node *J*, u^J, v^J and w^J are the nodal variables corresponding to the *x, y,* and *z* directions at node *J* and $\mathbf{U} = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2\ \boldsymbol{\alpha}_3 \ldots]^T$ is the vector of all the nodal unknowns. The nodal shape function matrix corresponding to the *J*th node is

$$\mathbf{H}_J(\mathbf{x}) = \begin{bmatrix} h_J(\mathbf{x}) & 0 & 0 \\ 0 & h_J(\mathbf{x}) & 0 \\ 0 & 0 & h_J(\mathbf{x}) \end{bmatrix} \tag{2}$$

where

$$h_J(\mathbf{x}) = W_J(\mathbf{x})\mathbf{P}^T(\mathbf{x})\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x}_J) \quad J = 1 \ldots N \tag{3}$$

with

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{N} W_I(\mathbf{x})\mathbf{P}(\mathbf{x}_I)\mathbf{P}^T(\mathbf{x}_I) \tag{4}$$

is the "shape function" at node *J*.

The purpose of the vector $\mathbf{P}(\mathbf{x}) = [1\ x\ y\ z]^T$ is to ensure a first order accurate scheme in 3D, similar to bilinear finite elements. The choice of a radial weighting function $W_I(\mathbf{x})$ at node *I* determines the degree of continuity and differentiability of the approximation as well as the computational cost. It is clear from Eq. (3) that the continuity of the shape functions is determined by the smoothness of the weight functions. The point collocation method (Bathe, 1996), which we use to discretize the governing differential equations, requires that the first derivatives of the weight functions be continuous. To ensure this property, we choose a quartic spline weighting function having the following form:

$$W_I(s) = \begin{cases} 1-6s^2 + 8s^3-3s^4 & 0 \le s < 1 \\ 0 & s \ge 1 \end{cases} \tag{5}$$

with minimum overlap between the neighboring nodes, where $s = \| \mathbf{x}-\mathbf{x}_I \|_0 / \mathbf{r}_I$ with $\| \cdot \|_0$ and $\mathbf{r}_I$ denoting the usual Euclidean distance and radius of sphere at node *I*, respectively.

Let us assume that we are interested in solving the elasticity equation, for example, on a domain Ω with boundary Γ (Figure 2):

$$\boldsymbol{\delta}_{\varepsilon}^{\mathrm{T}}\boldsymbol{\tau}(\mathbf{u})+\mathbf{f}^{\mathrm{B}}(\mathbf{x})=0 \quad \text{in } \Omega \tag{6}$$

subject to the boundary conditions

$$\begin{array}{ll} \mathrm{N}\boldsymbol{\tau}(\mathbf{u})-\mathbf{f}^{\mathrm{s}}(\mathbf{x})=0 & \text{on } \Gamma_{\mathrm{f}} \\ \mathbf{u}(\mathbf{x})-\mathbf{u}^{\mathrm{s}}=0 & \text{on } \Gamma_{\mathrm{u}} \end{array} \tag{7}$$

In equations (6) and (7), $\mathbf{u} = \{\mathrm{u}(\mathbf{x})\ \mathrm{v}(\mathbf{x})\ \mathrm{w}(\mathbf{x})\}^{\mathrm{T}}$ and $\boldsymbol{\tau} = \{\tau_{xx}\ \tau_{yy}\ \tau_{zz}\ \tau_{xy}\ \tau_{yz}\ \tau_{zx}\}^{\mathrm{T}}$ are the displacement at any point $\boldsymbol{x}$ in the domain and the vector of stress, respectively; $\mathbf{f}^{\mathrm{S}}$ is the prescribed force vector on the boundary Γ_{f}, $\mathbf{u}^{\mathrm{S}}$ is the vector of prescribed displacements on the boundary Γ_{u} (note that the domain boundary $\Gamma = \Gamma_{\mathrm{f}} \cup \Gamma_{\mathrm{u}}$ and $\Gamma_{f} \cap \Gamma_{\mathrm{u}} = 0$), $\mathbf{f}^{\mathrm{B}}$ is the body force vector, δ_{ε} is a linear gradient operator, **N** is the matrix of direction cosine components of a unit normal to the domain boundary (positive outwards) and $\mathbf{x} = \{\mathrm{x}, \mathrm{y}, \mathrm{z}\}$. The stress vector is related to the strain vector, ε, through the constitutive relation (linear, elastic):

$$\boldsymbol{\tau}(\mathbf{u})=\mathbf{D}\boldsymbol{\varepsilon}(\mathbf{u}) \tag{8}$$

where **D** is the constitutive matrix. In the simplest case of linear isotropic elasticity, there are two independent parameters in **D** (Young's modulus, **E,** and Poisson's ratio, ν) of the form:

$$\mathbf{D}=\frac{\mathrm{E}(1-\nu)}{(1+\nu)(1-2\nu)} \left[\begin{array}{cccccc} 1 & \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & & & \\ \frac{\nu}{1-\nu} & 1 & \frac{\nu}{1-\nu} & & & \\ \frac{\nu}{1-\nu} & \frac{\nu}{1-\nu} & 1 & & & \\ & & & \frac{1-2\nu}{2(1-\nu)} & & \\ \multicolumn{3}{c}{\text{Elements not}} & & \frac{1-2\nu}{2(1-\nu)} & \\ \multicolumn{3}{c}{\text{shown are zeros}} & & & \frac{1-2\nu}{2(1-\nu)} \end{array}\right] \tag{9}$$

In PAFF, the approximation in Eq. (1) is used to solve the governing partial differential equations of motion from Eq. (6) using a method known as point collocation. In this technique, the partial differential equations, as well as the boundary conditions, are satisfied at the nodal points. This method is vastly simplified compared to the finite element method since no computationally intensive numerical integration is used.

The resulting set of discretized equations can be written in compact form:

$$\mathbf{KU}=\mathbf{f} \tag{10}$$

where **K** is the stiffness matrix and **f** is the vector containing nodal loads. Solving this set of equations, one computes the numerical solution from Eq. (6). In this paper we are interested in linear elastic tissue response. However, extension to nonlinear tissue models is straightforward and is discussed in Lim and De (2005).

However, a naive solution of Eq. (10) does not, of course, lead to real time performance. We present two techniques of obtaining real time interactivity for linear elastic material models. In Section 3.1 we present a fast global analysis technique (GPAFF) that depends on precomputed data but scales linearly as the number of unknowns. However, for models with over a few thousand nodes, this technique hits the limit of allowable operational complexity on a serial computer. In Section 3.2 we present a localized version of the PAFF algorithm (LPAFF) for larger models.

The key assumption in both accelerating techniques is that the surgical tool–soft tissue interaction is local (see Figure 3) and the deformation field dies off rapidly with increase in distance from the surgical tool tip. We will, henceforth, refer to this as the localization assumption.

The localization assumption follows naturally from the physics that governs the deformation of elastic bodies as well as the psychophysics of human perception. The well-known Boussinesq solution (Johnson, 1996) for the displacement field produced by a concentrated point force, acting normally on an elastic half-space, dies off inversely as the distance from the point of application of the load. One needs to couple this with the well-known observation (Weber's law) that the human sensory system requires a finite change in stimulus intensity

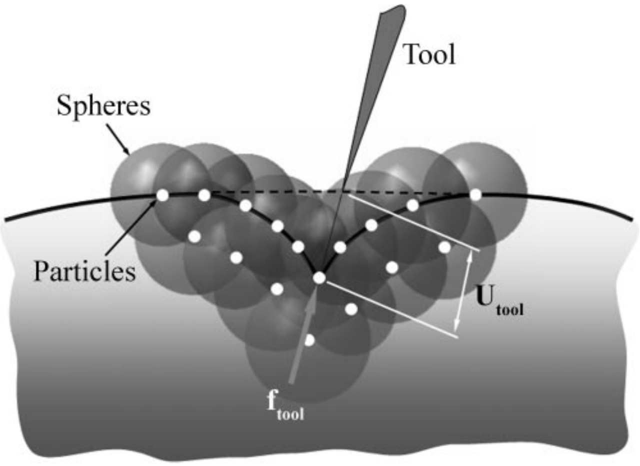

Figure 3. *A schematic of the PAFF technique. U_{tool} and f_{tool} are the prescribed displacement and reaction force at the tool tip, respectively.*

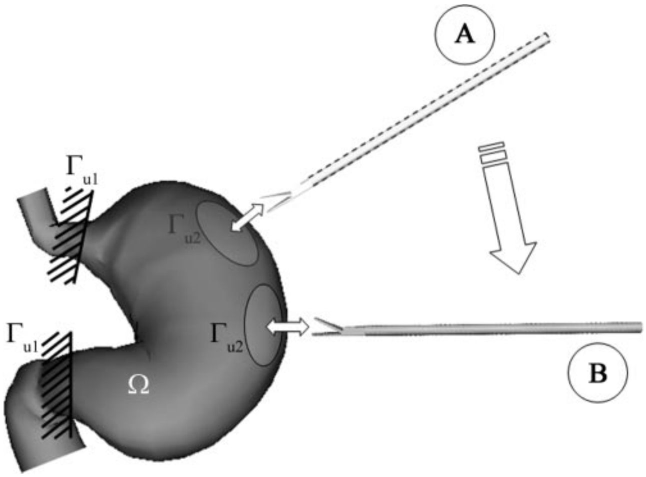

Figure 4. *The global PAFF (GPAFF) scheme. When the surgical tool moves from location A to location B, only a very small portion of the boundary (Γ_{u2}) is affected.*

over the background intensity (known as the just noticeable difference or JND) to recognize the change (for example, we will possibly not visually perceive the difference between a 1 mm deformation and 1.1 mm deformation; Surdick, Davis, King, & Hodges, 1997).

3.1 Real Time Global PAFF (GPAFF)

Mathematically, the localization assumption (in the previous section) translates to the condition that the prescribed boundary condition changes on only a very small portion of the boundary (Γ_{u2}) where the surgical tool interacts with the virtual organ (Figure 4). Hence we do not need to solve the entire problem over and over again. Instead it is possible to make incremental corrections to a previously computed solution, which results in a vastly accelerated solution procedure.

We therefore define a specialized problem (Figure 4) where we assume that zero displacements are prescribed on a portion of the boundary (Γ_{u1}) and the virtual tool interacts with only a very small portion of the boundary (Γ_{u2}), that is, $\mathbf{u}(\mathbf{x}) = \mathbf{U}_{\text{tool}}$ on Γ_{u2}, where $\mathbf{U}_{\text{tool}}$ is the displacement applied to the virtual organ through the surgical tool (assumed known). During the process of interaction of the surgical tool with the soft tissue only the boundary conditions on Γ_{u2} change.

Let $\tilde{\mathbf{K}}$ be the stiffness matrix obtained from $\mathbf{K}$ in Eq. (10) after incorporating the zero displacement boundary conditions on Γ_{u1}. Then the matrix $\tilde{\mathbf{C}} = \tilde{\mathbf{K}}^{-1}$ can be partitioned as:

$$\tilde{\mathbf{C}} = \begin{bmatrix} \tilde{\mathbf{C}}_{nn} & \tilde{\mathbf{C}}_{nu} \\ \tilde{\mathbf{C}}_{un} & \tilde{\mathbf{C}}_{uu} \end{bmatrix} \tag{11}$$

corresponding to a partitioning of the vector of nodal parameters as $\mathbf{U} = [\mathbf{U}_{\text{tool}}\mathbf{U}_{\text{u}}]^{\text{T}}$ where $\mathbf{U}_{\text{u}}$ is the vector of unknown nodal displacements, and $\mathbf{U}_{\text{tool}}$ is the vector of known nodal displacements. The reaction force vector at the tool $\mathbf{f}_{\text{tool}}$ is obtained from $\mathbf{f}_{\text{tool}} = \tilde{\mathbf{C}}_{nn}{}^{-1}\mathbf{U}_{\text{tool}}$. The unknown nodal displacement may be obtained as $\mathbf{u}_{\text{u}} = \tilde{\mathbf{C}}_{un}\mathbf{f}_{\text{tool}}$.

If the matrix $\tilde{\mathbf{C}}$ is precomputed and stored, then the cost of computation of the tool reaction forces is only of the order of m^2 (written as $O(m^2)$) where m nodes are in contact with the tool. The cost of computation of the displacement field is $O(m(N–m))$. Therefore, the overall computational complexity of such a procedure is $O(mN)$ which is essentially $O(N)$ when $m << N$.

To compare the performance of GPAFF with finite elements we consider a test problem of a hemisphere (25 mm radius) indented at its pole (Figure 5a). A PAFF discretization using 1026 particles and a FEM

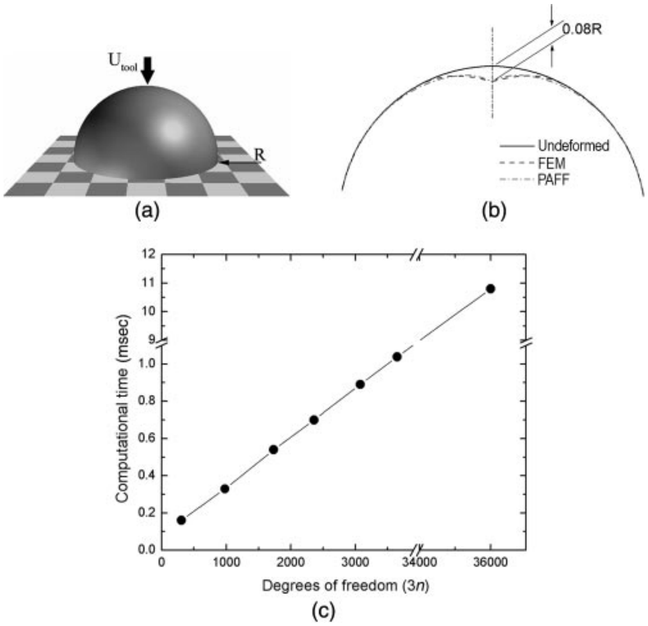

Figure 5. *(a) The test problem of a hemisphere indented at the pole. (b) The deformation field computed when the GPAFF technique (U_{tool} = 0.08 R) is used for the simulation of a surgical tool tip interacting with a hemisphere (of radius R = 25mm). The undeformed surface as well as the deformed profile obtained when a finite element software package is used to solve the problem is also shown. (c) The computational cost for GPAFF scheme is seen to increase linearly with increase in the degrees of freedom of the model.*

Table 1. *Comparison of Computational Times for Solving the Hemisphere Model (Figure 5a) Using Finite Elements (FEM), GPAFF, and LPAFF*[a]

	FEM	GPAFF	LPAFF
Number of nodal points (n)	1715	1026	28[b]
Degrees of freedom ($3n$)	5145	3078	84
Solution times (sec)	4.02	0.0009	0.0137[c]

[a] The solution time refers to the time to solve the system matrices. A 2.2 GHz P IV machine was used for the simulations.

[b] Number of nodes in influence region.

[c] Computational time for building stiffness matrix and nodal solution.

discretization using 1715 nodal points provided solutions with comparable accuracy (see Figure 5b). However, the use of precomputations in the GPAFF method results in a solution time which is four orders of magnitude smaller than that required by FEM (see Table 1). The finite element method using precomputations will provide speedups comparable to GPAFF. However, other advantages of meshfree analysis will be unavailable.

Figure 5c shows that the computational cost of the GPAFF scheme increases linearly with increase in the degrees of freedom (DOF) of the global model. The maximum size of the model (in terms of the number of nodes that can be simulated) is constrained by the hardware of the system. For GPAFF we calculate this "hard limit" to be in the range of 11,600 nodes.

Since a finer discretization may be necessary for complex organ models, we have developed a localized version of the PAFF algorithm (LPAFF), which we describe in the next section.

3.2 Real Time Local PAFF (LPAFF)

The localized PAFF (LPAFF) idea essentially implies that only local discretization is performed as a "hive" of nodal points travels with the tool tip (Figure 6). This technique results in a dramatic reduction in the solution time for massively complex organ geometries.

The localization results in zero displacements being automatically assumed on the periphery of the "region of influence" (ROI) of the surgical tool-tip, which is chosen after careful observation of videos of actual surgical procedures.

Figure 7 shows the placements of nodes around the collision point. We use a bounding box hierarchy with local neighborhood search algorithm (Ho et al., 1999) for real time collision detection. Hierarchical databases for geometrical connectivity information of the objects permit localized search when the surgical tool moves. The neighboring polygons and vertices of a collision point within the ROI can be rapidly computed using this local-search approach and the normal vectors of neighborhood vertices are obtained from the normals of their neighboring polygons. Then PAFF particles are placed by projecting along these normal vectors.

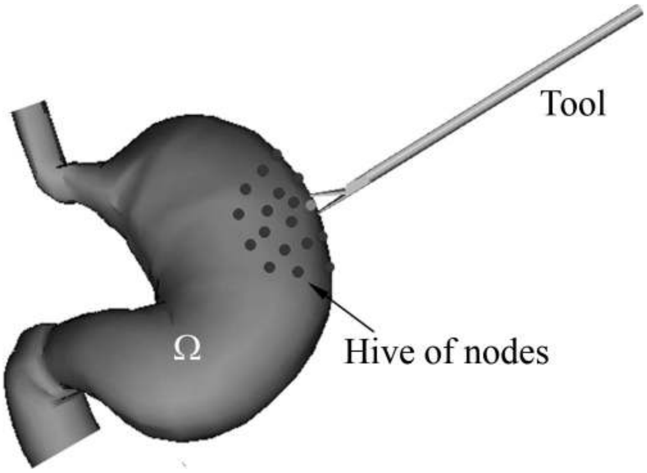

Figure 6. *A schematic of the localized PAFF (LPAFF) scheme. In LPAFF, the computations are performed on a local "hive" of nodal points which moves with the tool tip.*

The stiffness matrix, **K,** in Eq. (10) may be partitioned as:

$$\mathbf{K}=\begin{bmatrix}\mathbf{K}_{\mathrm{nn}} & \mathbf{K}_{\mathrm{nu}}\\ \mathbf{K}_{\mathrm{un}} & \mathbf{K}_{\mathrm{uu}}\end{bmatrix} \tag{12}$$

corresponding to a partitioning of the vector of nodal parameters as $\mathbf{U} = [\mathbf{U}_{\mathrm{tool}}\ \mathbf{U}_{\mathrm{u}}]^{\mathrm{T}}$ where $\mathbf{U}_{\mathrm{tool}}$ is the vector of known nodal displacements at the surgical tool tip and $\mathbf{U}_{\mathrm{u}}$ is the vector of unknown displacements which can be obtained from $\mathbf{U}_{\mathrm{u}} = -\mathbf{K}_{\mathrm{uu}}^{-1}\ \mathbf{K}_{\mathrm{un}}\ \mathbf{U}_{\mathrm{tool}}$. The force vector delivered to the force feedback device is computed as $\mathbf{f}_{\mathrm{tool}} = \mathbf{K}_{\mathrm{nn}}\ \mathbf{U}_{\mathrm{tool}} + \mathbf{K}_{\mathrm{nu}}\ \mathbf{U}_{\mathrm{u}}$.

Since **K** is built locally, the size of the matrix is quite small. The major advantage of LPAFF is that it is not limited to the computation of linear elastic tissue and real time performance may be obtained without using any pre-computations (Lim & De, 2005).

Table 1 presents a comparison of solution times for the hemisphere problem (Figure 5a). The total time is assumed to be composed of the time to generate the stiffness matrix and time to solve the system of equations. While the FEM solution takes more than 4 seconds, the LPAFF solution takes only 14 milliseconds. This is, of course, not a very fair comparison since the entire volume of the sphere is meshed using volumetric finite elements while only a few LPAFF nodes are sprinkled around the tool tip.

However, the solution accuracy of the two techniques is quite comparable, at least in the vicinity of the tool tip, emphasizing the fact that the deformation profile is truly local. As expected, the LPAFF solution drifts from the FEM solution far from the tool tip.

Figure 8 shows the computational cost versus the degrees of freedom in the LPAFF model.

3.3 Smoke Generation During Cauterization Using PAFF

As we have pointed out, the PAFF technique provides a very general and powerful framework for simulation. In this section we will demonstrate how it can be used to generate smoke during cauterization.

A variety of mechanisms have been used to dissect tissue and enable haemostasis in laparoscopic surgery. The dissection technique requires a modality that can accomplish meticulous haemostasis without causing inadvertent tissue damage. Electrosurgery has become the most widely used cutting and coagulating technique in minimally invasive surgery. During laparoscopic electrosurgery, tissue is burnt or hemorrhaging is stopped with the help of an electrocautery. This procedure results in copious generation of smoke.

It is quite straightforward to generate interpolations with no connectivity information such that the smoke can drift away and diffuse. In the absence of pressure gradients, the Lagrangian form of the Navier Stokes equations which governs the motion of any fluid is

$$\frac{d\mathbf{v}}{dt} = \frac{1}{\rho}(\mu\nabla^2\mathbf{v} + \mathbf{f}) \tag{13}$$

In this equation, **v** represents the smoke velocity (in 3D) at time instant *t*. ρ and μ are the smoke density and viscosity, respectively, and **f** is the external force acting on the smoke per unit volume. This equation is integrated in time (using Euler integration) to obtain the velocity of any smoke particle at any instant of time $t + \Delta t$ according to the equation:

$$\mathbf{v}_{t+\Delta t} = \mathbf{v}_t + \frac{\Delta t}{\rho}(\mu\nabla^2\mathbf{v} + \mathbf{f})_{\mathrm{t}} \tag{14}$$

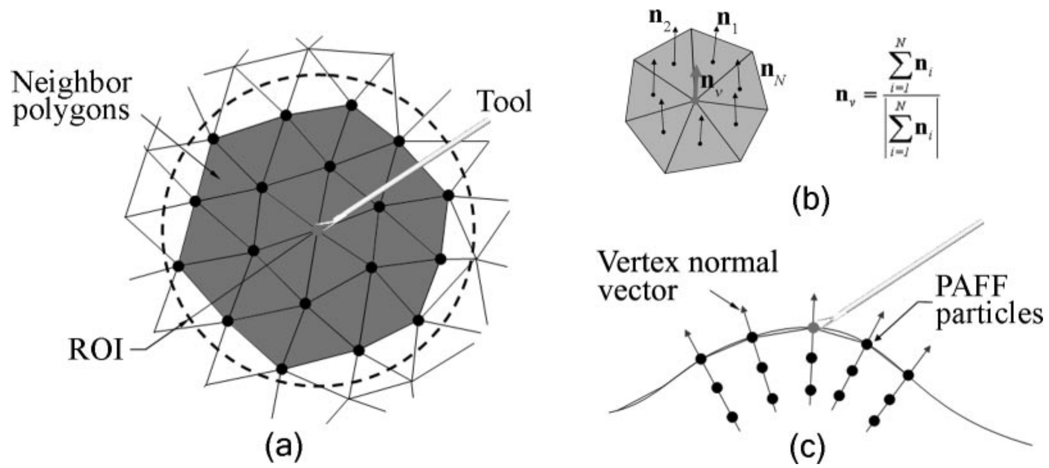

Figure 7. *Placement of the PAFF particles after a collision point has been determined. (a) A local-search algorithm is used to find neighboring polygons and vertices. (b) Normal vectors defined. (c) PAFF particles. PAFF particles are placed by projecting the neighboring vertices along their normal vectors.*

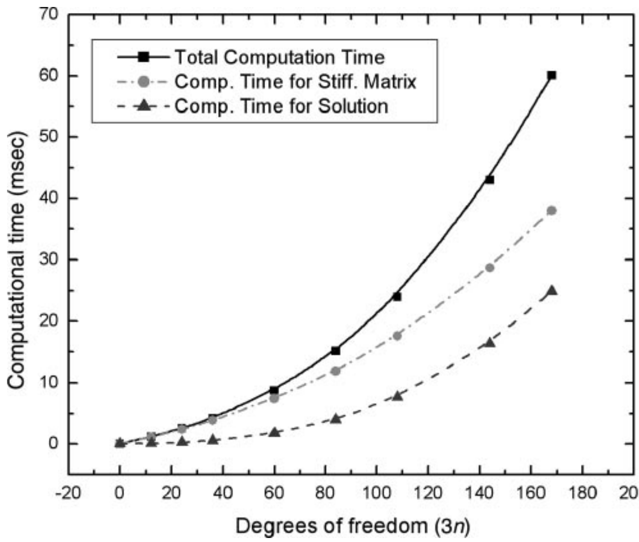

Figure 8. *Computational cost of the LPAFF scheme.*

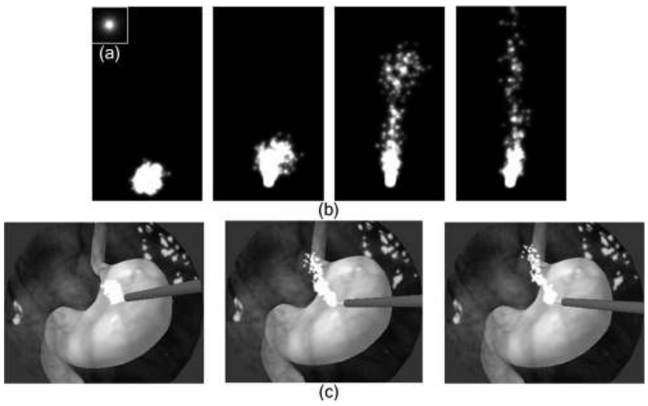

Figure 9. *Smoke generation using PAFF. (a) A texture image used for smoke particles. We use a single image frame of size 32- by 32-bit. (b) A sequence of snapshots for smoke generation. (c) Smoke generation during the simulation cauterization on a model of the stomach.*

where Δt is the time step. The second derivative of the velocity field is obtained as follows

$$\nabla^2 \mathbf{v}_t = \sum_{J=1}^{N} (\nabla^2 \mathbf{H}_J(\mathbf{x}))_t \boldsymbol{\alpha}_J \qquad (15)$$

Figures 9a,b show how smoke may be generated using this algorithm and Figure 9c presents an example of the simulation of cauterization on a stomach model. In our simulation, however, the external force **f** acting on the smoke particles was assumed to be random, as opposed to pressure forces due to insufflations in actual surgery which tends to diffuse the smoke particles to a greater extent.

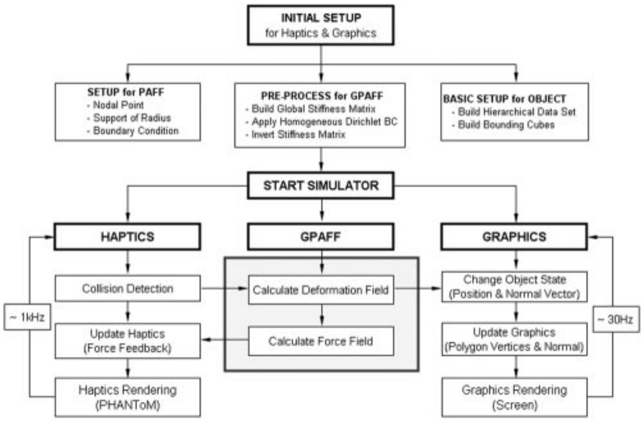

Figure 10. *The software architecture for GPAFF.*

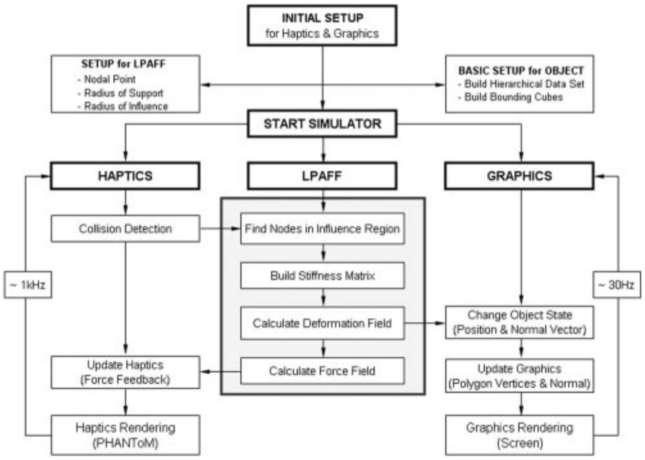

Figure 11. *The software architecture for LPAFF.*

4 Implementation Details

Figures 10 and 11 show the implementation steps in GPAFF and LPAFF schemes, respectively. Three major computational steps need to be performed in real time:

- *Pre-processing:* This step involves an initial setup for graphics and haptics rendering, generation of database of organ geometries, and setting up of the computational primitives for PAFF.
- *Response analysis:* This includes collision detection and collision response computations. In our implementation, we have used a point-based representation of the surgical tool and have employed a bounding box hierarchy with local neighborhood search algorithm developed by Ho et al. (1999) for real time collision detection.
- *Graphics and haptics rendering:* In this step the force vector is updated and is fed back to the user through the haptic interface device and the surface polygon vertices are updated and rendered.

In the GPAFF model, the precomputation of the global linear stiffness matrix, the application of the fixed boundary conditions, and the inversion of this matrix are executed in the pre-processing stage. After starting the simulator, the stored matrix is retrieved and the deformation field and interaction forces are calculated in real time.

In the LPAFF scheme, the size of the influence region is defined after a careful observation of real surgical videos and particles are placed around the collision point. The computation of the deformation of the organ and the reaction force at the tool-tip are performed during simulation.

5 Simulation Examples

We use a WinNT-based personal computer (Pentium 4 2.2 GHz processor) with a high end graphics accelerator (NVIDIA Quadro4 XGL) and the PHANToM (Premium 1.0A) force feedback devices from SensAble Technologies Inc. The source code is written in C++ using the OpenGL library for graphics rendering and GHOST SDK for haptic rendering.

Figure 12 shows the successive deformations of a linear elastic sphere indented at its pole. In Figure 12a the GPAFF scheme is used, while in Figure 12b the LPAFF is used.

The rest of the examples are 3D organ models obtained from segmented image data from the National Library of Medicine's (NLM) Visible Human Project (VHP). A whole-body adult male model, called Visible Photographic Man, or VIP-Man (Xu, Chao, & Bozkurt, 2000), as shown in Figure 13, has been developed using the high-resolution transverse color photographic im-

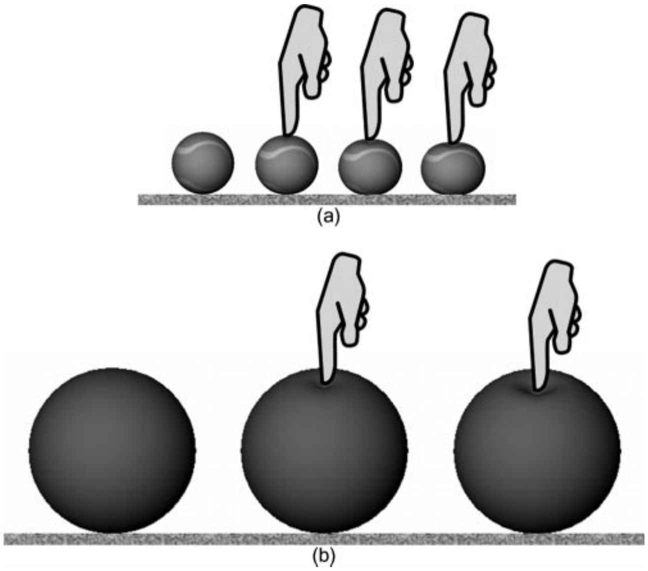

Figure 12. *Successive deformations of a linear elastic sphere indented at its pole. (a) In case of moderately sized objects (e.g., tennis ball indented by finger), the prescribed boundary condition changes on only a very small portion of the boundary as the indenter interacts with the object and the GPAFF scheme is suitable. (b) For larger objects (e.g., balloon indented by finger), it may be assumed that the deformation zone is localized within a "region of influence" in this situation, hence the LPAFF is more favorable.*

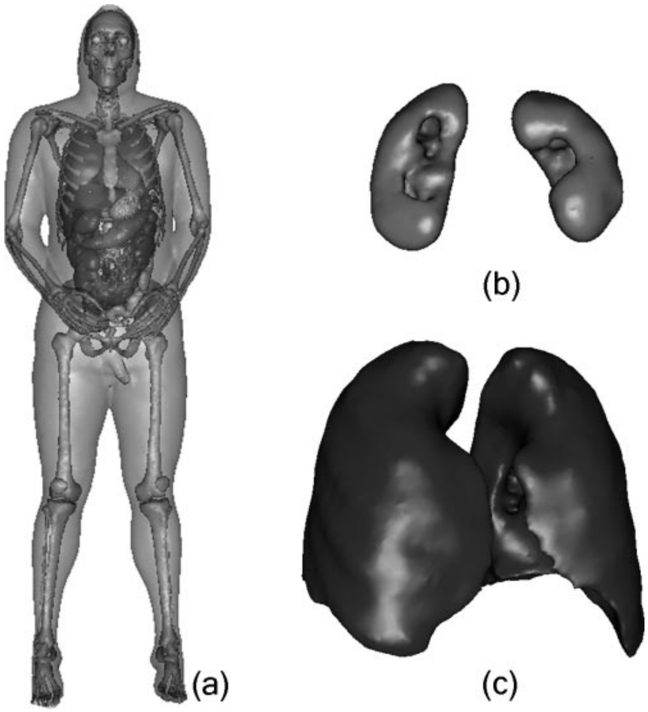

Figure 13. *VIP-Man showing (a) whole body skin, skeletal structures, and organs; (b) kidney and (c) lung model used in our simulation.*

ages from the VHP dataset. At a voxel size of 0.33 mm × 0.33 mm × 1 mm, the resolution of VIP-Man is at least 10 times finer than the tomographic models developed previously. The original color photographs for the male had been identified and segmented mostly by manual procedures to yield up to 1,400 anatomical structures by Spitzer, Whitlock, and Whitlock (1998). Once an organ or tissue has been identified, the associated voxels are arbitrarily colored for visualization. Then the marching cubes algorithm (Lorensen & Cline, 1987) is employed to reconstruct 3D volumetric models from segmented data.

We have used the 3D kidney (Figure 13b) and the lung (Figure 13c) models of VIP-Man to demonstrate our techniques. The visualization toolkit (VTK) is used for the surface rendering of the voxelized images.

In Figure 14a a kidney model having 1080 polygons is simulated using the GPAFF scheme. For a lung model having 1364 polygons (Figure 14b), the LPAFF is used for real-time simulation.

6 Concluding Remarks

Points as rendering primitives have been used in computer graphics (Levoy & Whitted, 1985; Zwicker, Pfister, van Baar, & Gross, 2002; Pfister & Gross, 2004). In this paper we present a scheme of using points as computational primitives in multimodal medical simulations. These points are not connected to each other by elastic members, but interact through diffuse elastic force fields. Their motion is dictated by the physics of the problem. Several strategies have been developed to accelerate the solution process. While localization of approximation is more suited to models having many nodes, the global solution strategy is more accurate and is preferable when massively parallel systems may be utilized for computations. We have, however, not followed the route to parallelism in this work since our philosophy has been to allow simulators to be run on PCs owned by medical personnel.

In this paper, we demonstrate the application of our

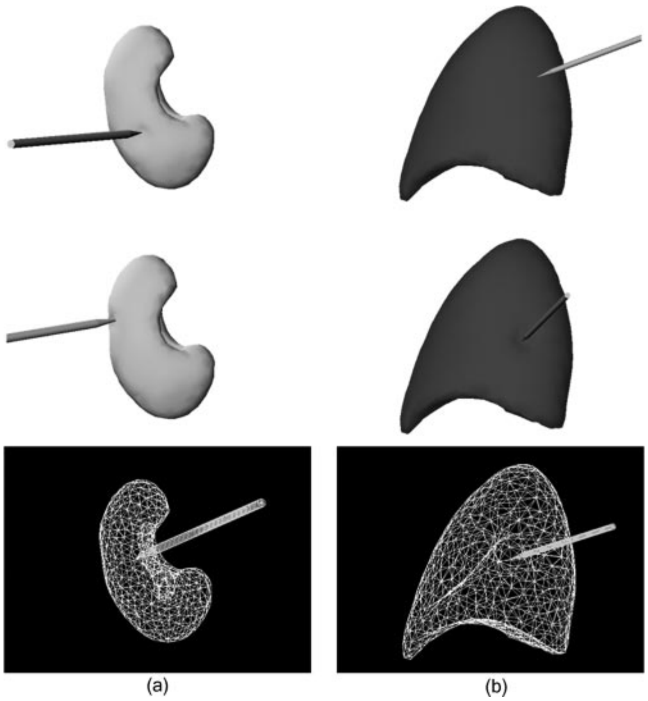

Figure 14. *Snapshots of the simulation results showing palpation task. (a) A kidney model having 1080 polygons using GPAFF technique. (b) A lung model having 1364 polygons using LPAFF scheme.*

method to linear elastic tissue behavior. In reality, soft biological tissues exhibit nonlinearities, time and rate dependence, as well as nonhomogeneity and anisotropy. The nonlinearity of tissue behavior may stem from either large deformations or simply from inherently nonlinear material response. A third type of nonlinearity arises due to change of contact area during deformation, for example, due to interaction of the surgical tool and the organ or between several organs as they press against each other. We have recently demonstrated the successful extension of the PAFF to the simulation of nonlinear tissue behavior (Lim & De, 2005).

We have confined our discussion to only tissue deformation and smoke generation. However, surgical procedures frequently involve bleeding. Simulation of bleeding in a physically realistic fashion is difficult since it requires the solution of the nonlinear Navier Stokes equations. Basdogan, Ho, and Srinivasan (1999) presented a method to simulate bleeding by calculating the depth of the fluid above a virtual surface based on a linearized Stokes model and rendered it through an auxiliary surface. While this method provides realistic bleeding simulation, it does not account for arterial bleeding. The application of PAFF to the simulation of tissue bleeding is left as future work.

Acknowledgments

The authors would like to thank Professor George Xu of RPI for providing the segmented VIP-Man models obtained from the Visible Human Project dataset. Numerous discussions with Professor C. Basdogan of Koc University, Turkey, are also acknowledged. This work was supported in part by grant R21 EB003547-01 from NIH.

References

Ayache, N., Cotin, S., & Delingette, H. (1998). Surgery simulation with visual and haptic feedback. *Robotics Research,* 311–316.

Basdogan, C., Ho, C., & Srinivasan, M. A. (1997). A ray-based haptic rendering technique for displaying shape and texture of 3D objects in virtual environments. *ASME Winter Annual Meeting.*

Basdogan, C., Ho, C., & Srinivasan, M. A. (1998). Force interaction in laparoscopic simulation haptic rendering of soft tissue. *Proceedings of MMVR'6 Conference.*

Basdogan, C., Ho, C., & Srinivasan, M. A. (1999). Simulation of tissue cutting and bleeding for laparoscopic surgery using auxiliary surfaces. *Proceedings of MMVR'7 Conference,* 38–44.

Basdogan, C., Ho, C., & Srinivasan, M. A. (2001). Virtual environments in medical training: Graphical and haptic simulation of laparoscopic common bile duct exploration. *IEEE/ASME Transactions on Mechatronics.*

Bathe, K. J. (1996). *Finite element procedures.* Englewood Cliffs, NJ: Prentice Hall.

Berkley, J., Oppenheimer, P., Weghorst, W., Berg, D., Raugi, G., Haynor, D., et al. (2000). Creating fast finite element models from medical images. *Proceedings of MMVR'8 Conference,* Irvine, CA, 26–32.

Berkley, H., Turkiyyah, G., Berg, D., Ganter, M., & Weghorst, S. (2004). Real-time finite element modeling for surgery simulation: An application to virtual suturing. *IEEE*

Transactions on Visualization and Computer Graphics, 10(3), 314–325.

Bro-Nielsen, M., & Cotin, S. (1996). Real-time volumetric deformable models for surgery simulation using finite elements and condensation. *Computer Graphics Forum, 15*(3), 57–66.

Choi, K. S., Sun, H., & Heng, P. A. (2004). Deformable simulation using force propagation model with finite element optimization. *Computer & Graphics, 28*(4), 559–568.

Cotin, S., Delingette, H., & Ayache, N. (1999). Real time elastic deformations of soft tissues for surgery simulation. *IEEE Transactions on Visualization and Computer Graphics, 5*(1), 62–73.

Cover, S. A., Ezquerra, N. F., O'Brien, J. F., Rowe, R., Gadacz, T., & Palm, E. (1993). Interactively deformable models for surgery simulation. *IEEE Computer Graphics and Applications,* 68–75.

Dachille, F., Qin, H., Kaufman, A., & El-Sana, J. (1999). Haptic sculpting of dynamics surfaces. *ACM Symposium on Interactive 3D Graphics.*

De, S., & Srinivasan, M. A. (1998). Rapid rendering of tool–tissue interactions in surgical simulations: Thin walled membrane models. The Third PHANToM User's Group Workshop, Dedham, MA.

De, S., & Srinivasan, M. A. (1999). Thin walled models for haptic and graphical rendering of soft tissues in surgical simulations. *Proceedings of MMVR'7 Conference,* San Francisco.

Desbrun, M., Schroder, P., & Barr, A. (1999). Interactive animation of structured deformable objects. *Graphics Interface,* 1–8.

Edwards, J., & Luecke, G. (1996). Physically based models for use in a force feedback virtual environment. *Japan/USA Symposium on Flexible Automation (ASME),* 221–228.

Girod, B., Keeve, E., & Girod, S. (1996). Craniofacial surgery simulation. *Proceedings of the 4th Conference on Visualization in Biomedical Computing (VBC'96),* 541–548.

Gottschalk, S., Lin, M. C., & Manocha, D. (1996). OBB-tree: A hierarchical structure for rapid interference detection. *Proceedings of SIGGRAPH'96,* 171–180.

Gregory, A., Ehmann, S., & Lin, M. (2000). *inTouch:* Interactive multiresolution modeling and 3D painting with a haptic interface. *Proceedings of IEEE Virtual Reality 2000,* 45–54.

Gupta, R., Sheridan, T., & Whitney, D. (1997). Experiments using multimodal virtual environments in design for assembly analysis. *Presence: Teleoperators and Virtual Environments, 6*(3), 318–338.

Ho, C., Basdogan, C., & Srinivasan, M. A. (1999). Efficient point-based rendering techniques for haptic display of virtual objects. *Presence: Teleoperators and Virtual Environments, 8*(5), 477–491.

Ho, S., Sarma, S., & Adachi, Y. (2001). Real-time interference analysis between a tool and an environment. *Computer Aided Design, 33*(13), 935–947.

Hsu, W. M., Hughes, W. M., & Kaufman, H. (1992). Direct manipulation of free-form deformations. *SIGGRAPH Proceedings on Computer Graphics, 26*(2), 177–184.

Hubbard, P. M. (1995). Collision detection for interactive graphics applications. *IEEE Transactions on Visualization and Computer Graphics, 1*(3), 218–230.

Jayaram, S., Vance, J., Gadh, R., Jayaram, U., & Srinivasan, H. (2001). Engineering applications of virtual reality environments. *ASME Transactions Journal of Computing and Information Sciences in Engineering, 1*(1), 72–83.

Johnson, K. L. (1996). *Contact mechanics.* Cambridge, UK: Cambridge University Press.

Kang, Y., Choi, J., & Cho, C. (2000). Fast and stable animation of cloth with an approximated implicit method. *Proceedings of Computer Graphics International 2000.*

Levoy, M., & Whitted, T. (1985). The use of points as display primitives (Technical Report TR 85-022). Chapel Hill, NC: UNC at Chapel Hill, Department of Computer Science.

Lim, Y.-J., & De, S. (2005). Nonlinear tissue response modeling for physically realistic virtual surgery using PAFF. World Haptics 2005 Conference.

Lorensen, W. E., & Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm. *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques,* 163–169.

Masutani, Y., Inoue, Y., Ishii, K., Kumai, N., Kimura, F., & Sakuma, I. (2004). Development of surgical simulator based on FEM and deformable volume-rendering. *Proceedings of the SPIE, 5367,* 500–507.

Ng, H., & Grimsdale, R. (1996). Computer graphics techniques for modeling cloth. *IEEE Computer Graphics and Applications, 16*(5), 91–108.

Palmer, I. J., & Grimsdale, R. L. (1995). Collision detection for animation using sphere-trees. *Proceedings of Eurographics, 14*(2), 105–116.

Peters, J., Fried, G., Swanstrom, L., Soper, N., Sillin, L., Schirmer, B., et al. (2003). Development and validation of a comprehensive program of education and assessment of the basic fundamentals of laparoscopic surgery. *Surgery, 135,* 21–27.

Pfister, H., & Gross, M. (2004). Point-based computer graphics. *IEEE Computer Graphics and Applications, 24*(4), 22–23.

Picinbono, G., Delingette, H., & Ayache, N. (2003). Nonlinear anisotropic elasticity for real-time surgery simulation. *Graphical Models, 65*(5), 305–321.

Quinlan, S. (1994). Efficient distance computation between non-convex objects. *Proceedings of International Conference on Robotics and Automation,* 3324–3329.

Satava, R., Magee, H., Cuschieri, A., Jakimowitz, J., Darzi, A., Buess, G., et al. (2001). Metrics final report: Developing quantitative measurements through surgical simulation. *Metrics for Objective Assessment of Surgical Skills Workshop.* Available at http://www.tatrc.org/website_metrics.

Sederberg, T. W., & Parry, S. R. (1986). Free-form deformation of solid models. *SIGGRAPH Proceedings on Computer Graphics, 20*(4), 151–160.

Spitzer, V. M., Whitlock, D. G., & Whitlock, D. Z. (1998). *Atlas of the visible human male: Reverse engineering of the human body.* Sudbury, MA: Jones & Bartlett Pub.

Surdick, R., Davis, E., King, R., & Hodges, L. (1997). The perception of distance in simulated visual displays: A comparison of the effectiveness and accuracy of multiple depth cues across viewing distances. *Presence: Teleoperators and Virtual Environments, 6*(5), 513–531.

Swarup, N. (1995). *Haptic interaction with deformable objects using real-time dynamic simulation.* Master's thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology.

Terzopoulos, D., & Waters, K. (1990). Physically-based facial modeling, analysis and animation. *Journal of Visualization and Computer Animation, 1,* 73–80.

Torkington, J., Smith, S., Rees, B., & Darzi, A. (2001). Skill transfer from virtual reality to a real laparoscopic task. *Surgical Endoscopy, 15,* 1076–1079.

Witkin, A., Baraff, D., & Kass, M. (1996). Tutorial notes on "An introduction to physically-based modeling." *Proceedings of the SIGGRAPH.*

Wu, X., Downes, M., Goktekin, T., & Tendick, F. (2001). Adaptive nonlinear finite elements for deformable body simulation using dynamic progressive meshes. *Proceedings of Eurographics,* 349–358.

Xu, X. G., Chao, T. C., & Bozkurt, A. (2000). VIP-Man: An image-based whole-body adult male model constructed from color photographs of the visible human project for multi-particle Monte Carlo calculations. *Health Physics, 78*(5), 476–486.

Zachmann, G., & Felger, W. (1995). The Box Tree: Enabling real time and exact collision detection of arbitrary polyhedra. *Proceedings of Simulation and Interaction in Virtual Environments,* 104–113.

Zachmann, G. (1997). Real-time and exact collision detection for interactive virtual prototyping. *Proceedings of DETC'97 (ASME Design Engineering Technical Conferences).*

Zwicker, M., Pfister, H., van Baar, J., & Gross, M. (2002). EWA splatting. *IEEE Transactions on Visualization and Computer Graphics, 8*(3), 223–238.