M. Harders*
Sternwartstrasse 7
ETF C 107
ETH Zurich
Switzerland

D. Bachofen
M. Grassi
Zurich University of Applied
Science

M. Bajka
University Hospital Zurich

U. Spaelter
Ecole Polytechnique Federale de
Lausanne

M. Teschner
University of Freiburg
Germany

B. Heidelberger
R. Sierra
D. Steinemann
S. Tuchschmid
J. Zatonyi
G. Székely
ETH Zurich

Virtual Reality Based Simulation of Hysteroscopic Interventions

Abstract

Virtual reality based simulation is an appealing option to supplement traditional clinical education. However, the formal integration of training simulators into the medical curriculum is still lacking. Especially, the lack of a reasonable level of realism supposedly hinders the widespread use of this technology. Therefore, we try to tackle this situation with a reference surgical simulator of the highest possible fidelity for procedural training. This overview describes all elements that have been combined into our training system as well as first results of simulator validation. Our framework allows the rehearsal of several aspects of hysteroscopy—for instance, correct fluid management, handling of excessive bleeding, appropriate removal of intrauterine tumors, or the use of the surgical instrument.

1 Introduction

Virtual reality based simulation is an appealing option to supplement traditional clinical education (Liu, Tendick, Cleary, & Kaufmann, 2003). The first attempts at using computer-based surgical simulation for training prospective surgeons were carried out at the beginning of the 1990s. While the great potential of this approach has consistently been recognized, the formal integration of training simulators into the medical curriculum is still lacking. Several factors are likely contributing to this unsatisfactory situation.

- Lack of identification of skill categories to be taught. For example, a clear separation between motor and cognitive skills, as well as other skill factors, is often not taken into account.
- Inappropriate evaluation and validation methods for existing systems. Often only the performance improvement on the simulator itself is taken as an indication for successful training, while inside the OR only semi-quantitative measures are available.
- The relation to and advantage over conventional methods is not shown. This is especially disturbing, if only basic motor skills are trained, for example, placing clips on a tube.
- Effects of negative training, such as due to reduced performance in a real situation due to an incorrect representation in the simulated environment, are not examined. In the extreme case, incorrectly trained behavior could lead to errors with potentially severe consequences for the patient.

*Correspondence to mharders@vision.ee.ethz.ch.

- The correlation between level of realism and training effect concerning the specific skills to be trained is still unknown. This is an important factor, since realism usually strongly correlates with complexity (and thus with overall cost).

We hypothesize that, especially for the training of higher level cognitive skills, the lack of a reasonable level of realism seriously hinders the widespread use of this potentially very useful technology. Therefore, we try to tackle this situation with a reference surgical simulator of the highest possible fidelity for procedural training. A key target is to go beyond rehearsal of basic manipulative skills, and enable the training of procedural skills such as decision making and problem solving. Moreover, with this system we intend to identify the necessary level of realism for achieving a specific training effect, by stepwise reduction of fidelity. Thereby, we hope to answer some of the prevalent questions listed above.

1.1 Hysteroscopic Interventions

The driving application for our endeavors is the simulation of hysteroscopic interventions. Therapeutic hysteroscopy has become a common technique in gynecological practice (ACOG, 1994). Nevertheless, a number of potentially dangerous complications exist. Some of these will be reviewed in more detail, in order to show the necessity of appropriate procedural training.

1.1.1 Wall Perforation. In Pinion et al. (1994) the rate of complications for therapeutic interventions is reported as 17%. According to Mencaglia and Hamou (2001), 97% of hysteroscopic interventions are performed with resectoscopes, with up to 9% of them leading to perforations. The most critical situation is wall perforation with resectoscope electrodes during cutting procedures, likely to cause lesions of intra-abdominal organs. In these cases, hysteroscopy usually has to be stopped and an emergency abdominal intervention must be performed.

1.1.2 Fluid Mismanagement. A fundamental prerequisite of hysteroscopy is the proper distension of

the uterine cavity, also known as hydrometra. In- and outflow of the distension fluid is accomplished via the endoscopic tool and controlled with valves, while the pressure of the liquid is provided by a pump. It is essential to select correct pressure settings for the hydrometra according to muscle tone and uterine wall thickness (Petrozza, 2008).

1.1.3 Intravasation. In the majority of hysteroscopic surgeries, a unipolar current is applied inside the cavity for electric resection. Therefore, the usage of a nonconductive distension media is indispensable. Unfortunately, this might cause another critical complication: fluid overload due to absorption in tissue or intravasation through sectioned vessels (Mencaglia & Hamou, 2001). This can have serious adverse effects, possibly leading to cerebral edema or even death. Therefore, the fluid balance needs to be closely monitored during the intervention. In addition, total operation time is usually limited.

1.1.4 Intra-uterine Bleeding. Hemorrhagic complications can also occur during hysteroscopic interventions, for instance intensive bleeding. After location of the source, the bleeding can be stopped via unipolar coagulation. As an alternative, small local bleeding can also be controlled by increasing the distension fluid pressure. However, this potentially aggravates the previously mentioned intravasation problems.

1.1.5 Reduced Visibility. Obscurations during interventions can be caused by endometrial bleeding, floating tissue fragments, or air bubbles. The flow of the distension fluid ensures clear visibility in the cavity. The gynecologist has to learn to correctly dose the in- and outflow through the valves ensuring the proper visibility of the scene. This is necessary to maintain orientation inside of the cavity.

1.2 System Overview

In contrast to existing systems and products (Levy, 1996; Montgomery et al., 2001; Muller-Wittig et al., 2001; Immersion, 2005; Lim, Brown, McColl,
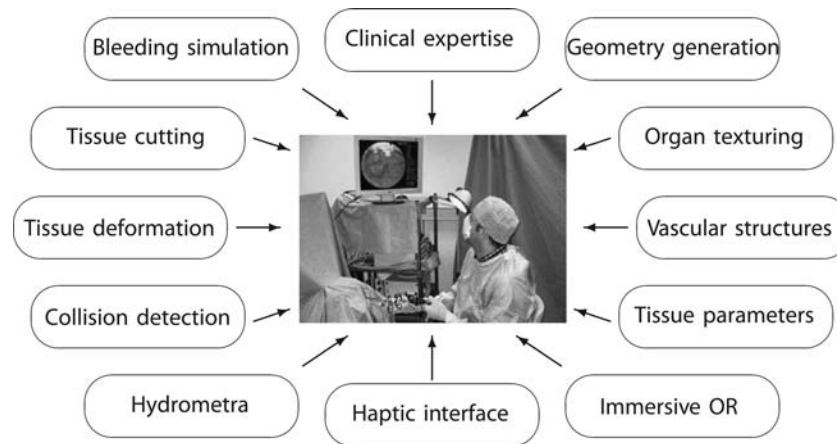
Figure 1. *Modules of the hysteroscopy simulator.*

Seligman, & Alsaraira, 2006), our work aims at achieving the highest possible realism. In order to develop the envisioned high fidelity training simulator, several components have to be integrated. Figure 1 depicts all elements that have been combined into our prototype. The components belong to four major categories.

First and foremost is the clinical background, which defines the problem area, identifies the training needs, and influences the system development at all stages. A successful collaboration with the OB/GYN department of the University Hospital Zurich was established in our project, which was also crucial for intra-operatively collecting necessary input data, ranging from tissue measurements to video and audio recordings. The next category focuses on the generation of surgical scenes. This includes the generation of geometric models of healthy and pathological anatomy (Sierra, Zsemlye, Szekely, & Bajka, 2006), organ textures (Paget, Harders, & Szekely, 2005), vessel structures (Szczerba & Szekely, 2002), and the determination of tissue deformation parameters (Bianchi, Solenthaler, Szekely, & Harders, 2004). In the former step, high resolution triangular meshes are obtained for visualization purposes, as well as lower resolution tetrahedral meshes for representation of objects in the physics modules. The complete scenarios are generated before the interactive simulation session. In this context, the ability to present variable scenes is a crucial component to simulate the day-to-day

demand on a surgeon. The training scenario definition will not be covered in this paper; further details can be found in Sierra, Bajka, Karadogan, Szekely, & Harders (2004). Finally, the real-time simulation is composed of elements of the last two categories. The first comprises the hardware and interface elements controlling the simulation, for instance the haptic device, or the immersive training environment. The second denotes the real-time software modules, which are combined into the virtual representation of the surgery.

In the following, we will introduce all elements of the real-time simulation system, namely simulation of the hydrometra, soft tissue deformation models, collision detection and response, cutting procedures, simulation of bleeding, scene visualization, and integration of all elements into a coherent framework. Moreover, we will provide a description of the simulator hardware and the immersive environment. Finally, we conclude with the derivation of performance metrics and a first evaluation study of the presented system.

2   Hydrometra Simulation

During hysteroscopy, a hydrometra is maintained, that is, the uterus is distended with liquid media to access and visualize the uterine cavity. The pressure and flow induced by the liquid are crucial manipulators for

the gynecologists during surgery, for instance to obtain a clear view of the operation site. Two different aspects of hydrometra are modeled in our system, namely the distension of the uterine muscle and the liquid flow simulation in the cavity (Sierra, Zatonyi, Bajka, Szekely, & Harders, 2005).

2.1 Cavity Deformation

The almost instantaneous response of the uterine cavity to pressure adjustments of the distension fluid exceeds the real-time capabilities of most known deformation models. In order to achieve physically realistic real-time deformation of the uterine cavity, we combine accurate FEM computations with freeform deformation approaches. The underlying idea is to first carry out off-line precomputations to obtain the response of the tissue model to the fluid pressure. These data are then used during real-time interaction for model adjustments based on given pressure states.

Three major steps are needed for our approach: In-vivo acquisition of material laws from real organs, offline computations with a highly accurate FE approach, and real-time replay of these data during the intervention. Material parameters for the offline FE deformation computations are obtained in vivo with a tissue aspiration device (Nava, Mazza, Kleinermann, Avis, McClure, & Bajka, 2004). In the second step, the offline computation enables the use of more advanced, and thus more accurate, tissue modeling techniques. We apply the FEM presented in Weiss, Bajka, Nava, Mazza, & Niederer (2004), where homogeneous, isotropic, and nonlinear hyper-elastic material laws are used. Based on the in vivo aspiration experiment, a polynomial strain-energy function is derived to describe the mechanical behavior of the particular soft tissue. The resulting equations are solved with a commercial software package. The organ model deformations are obtained for 100 different pressure settings. In the third step, the precomputed deformation states are loaded into the simulator and piecewise linearly interpolated in real time according to the applied fluid pressures. Figure 2 illustrates results obtained with the hydrometra deformation approach. First, vertices of a uterus mesh in its collapsed

state are depicted. Pressure is applied on the vertices inside the uterine cavity. The vertices near the cervix are fixed, and the remaining ones are free to move. The second image shows the front surface in wireframe mode to visualize the inner cavity. The uterus in the third image illustrates the result of the FEM-based hydrometra simulation.

It should be noted that in our method it is assumed that the boundary conditions of the deformation model do not change during the intervention. This is usually the case in hysteroscopic procedures, since the surrounding tissues are not directly accessible. Moreover, extreme modifications of the cavity wall are avoided, since cutting into the myometrium is limited to a relatively thin layer during surgery. Otherwise, a perforation of the wall could result, leading to the immediate termination of the intervention. Hence we can assume that the uterus' response to pressure differences stays relatively constant throughout the intervention.

2.2 Flow Simulation

The motion of the distension fluid is described by the Navier-Stokes equations. For an incompressible liquid they take the form:

$$\nabla \cdot \mathbf{v}_f = 0 \qquad (1)$$

$$\frac{\partial \mathbf{v}_f}{\partial t} = -(\mathbf{v}_f \cdot \nabla)\mathbf{v}_f + \nu \nabla^2 \mathbf{v}_f - \frac{1}{\rho}\nabla p + \mathbf{f} \qquad (2)$$

where $\mathbf{v}_f$ is the velocity vector, p the pressure, ρ the density of the fluid, ν the kinematic viscosity coefficient, and $\mathbf{f}$ the acceleration due to external body forces per unit mass. Density and temperature of the fluid are considered constant. Equation 1 represents the mass conservation in the fluid, while Equation 2 is the momentum equation (Anderson, 1995). These equations are solved numerically on a uniform Cartesian grid to evaluate the time dependent behavior of the fluid. We use the semi-Lagrangian solver described in Stam (1999) in order to obtain an unconditionally stable solution in real time. Furthermore, the method is adapted to cope with the complex dynamic boundary conditions. For hysteroscopy these are the endometrium as well as pa-
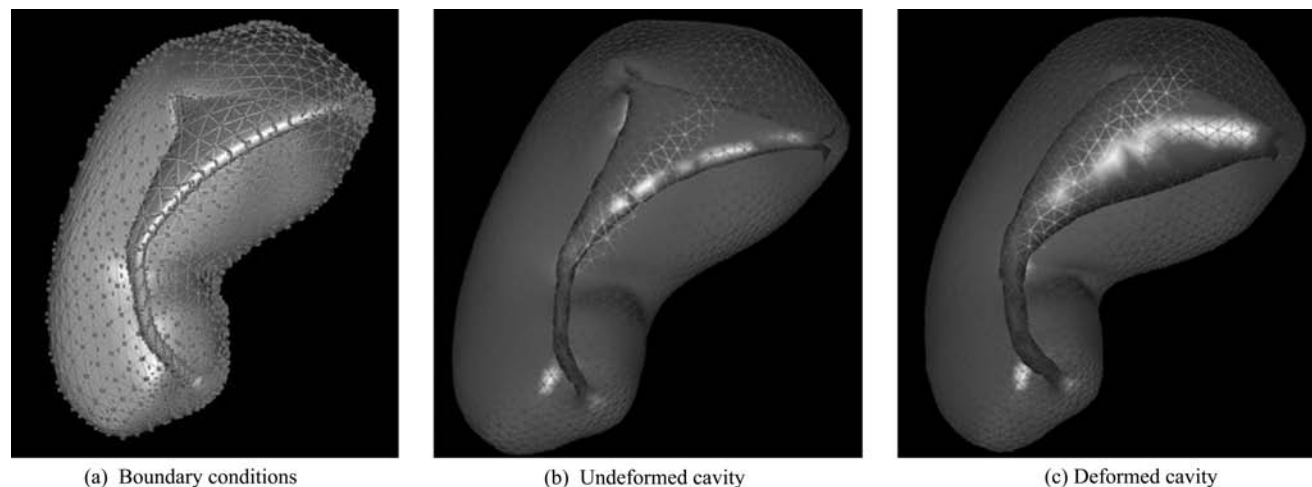
(a) Boundary conditions   (b) Undeformed cavity   (c) Deformed cavity

Figure 2. *Uterus deformation during hydrometra.*

thologies present in the scene, the in- and outflow according to the tool position, and their intensity according to the valve states. By labeling the voxels of a grid overlaid on the scene, these elements can be incorporated consistently into the numerical solver. As both the hydrometra and the pathologies may frequently change during hydrometra simulation, the labeling of the voxels has to be updated accordingly. Therefore, all voxels are relabeled in real time whenever the surfaces are modified.

Considering the small size of a normal cavity in the order of a few cm^3, the liquid flow can be highly turbulent, with a Reynolds number of about 30,000. Therefore, for higher flow rates, artificial vortices are added. While this combination is not physically accurate, it provides a higher level of visual fidelity while maintaining real-time performance. The resulting flow velocity can be computed by superimposing the flow provided by the fluid solver and the velocity originating from the various vortices. In Figure 3, the flow field is visualized by pathlines. Two different stages of hydrometra are shown in the first two images, while the third image illustrates the flow in the presence of 20 vortices. The arrows indicate the in- and outflow directions. As can be seen, the pathlines clearly indicate the main flow streams from the inlet position to the outflow. The vortices visi-

bly perturb the local behavior as desired, but do not disrupt the overall flow.

Average calculation time of the 3D numerical solver for a $40 \times 40 \times 40$ grid is 38 ms using three parallel threads on a Sun Fire 6800 HPC server with Ultra Sparc III processors. Similar results for real-time simulation were obtained using a $4 \times$ AMD Opteron 275 CPU 2.2 GHz system.

3 Tissue Deformation Modeling

Apart from the almost instantaneous deformation of the uterine cavity due to pressure changes, we also have to simulate soft tissue behavior for other structures in the scene. This mainly includes the movement of polyps and myomas. This is handled with more conventional interactive deformation schemes. We have developed a unified method suitable for modeling elastically deformable tetrahedral meshes, which extends existing techniques by incorporating efficient ways for volume and surface area preservation (Teschner, Heidelberger, Mueller, & Gross, 2004). The computational efficiency of the approach is similar to simple mass-spring systems. Thus, environments of up to several thousand deforming primitives can be handled at interactive speed.
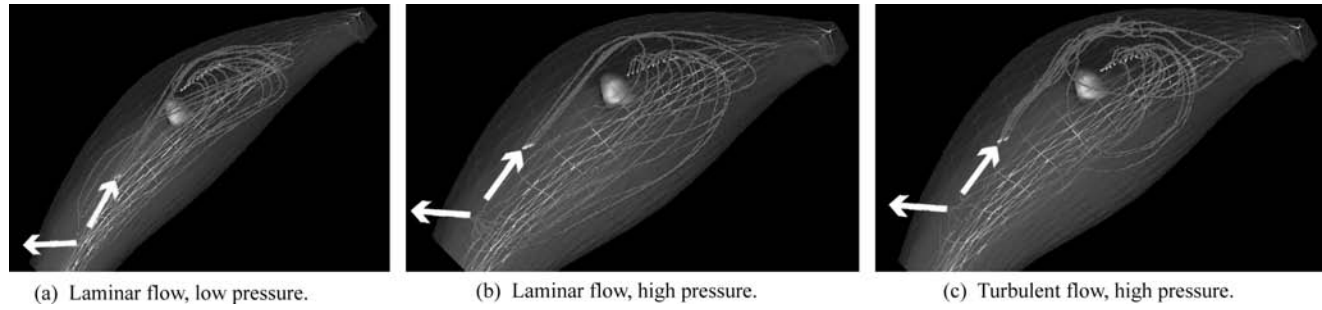
(a) Laminar flow, low pressure. (b) Laminar flow, high pressure. (c) Turbulent flow, high pressure.

Figure 3. *Simulation of distension fluid flow in the uterine cavity.*

We consider deformable solids that are discretized into tetrahedra. Object mass points are given by the nodes of the tetrahedral elements. In order to compute the dynamic behavior of objects, we derive forces at the mass points from potential energies. These forces aim at preserving distances between mass points, surface areas of the object, and volumes of tetrahedra. In order to represent deformations of objects, we consider constraints of the form $C(\mathbf{p}_0, \ldots, \mathbf{p}_{n-1})$. These scalar functions depend on mass point positions $\mathbf{p}_i$. They are zero if the object is undeformed. In order to compute forces based on these constraints, we consider three different potential energies of the form

$$E(\mathbf{p}_0, \ldots, \mathbf{p}_{n-1}) = \tfrac{1}{2}kC^2 \qquad (3)$$

with k denoting a material stiffness coefficient that has to be defined for each type of potential energy.

The first potential energy E_D ensures distance preservation and considers all pairs of mass points that are connected by tetrahedral edges. It represents energy based on the difference of the current distance of two points and the initial or rest distance D_0.

$$E_D(\mathbf{p}_i, \mathbf{p}_j) = \tfrac{1}{2}k_D\left(\frac{|\mathbf{p}_j - \mathbf{p}_i| - D_0}{D_0}\right)^2 \qquad (4)$$

The second energy E_A denotes surface area preservation, and considers triples of mass points that build surface triangles. E_A represents energy based on the difference of the current area of a surface triangle and its initial area A_0.

$$E_A(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k) = \tfrac{1}{2}k_A\left(\frac{\tfrac{1}{2}|(\mathbf{p}_j - \mathbf{p}_i) \times (\mathbf{p}_k - \mathbf{p}_i)| - A_0}{A_0}\right)^2 \qquad (5)$$

The third potential energy E_V guarantees volume preservation, and considers sets of four mass points that build tetrahedra. E_V represents energy based on the difference of the current volume of a tetrahedron and its initial volume V_0.

$$E_V(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l)$$
$$= \tfrac{1}{2}k_V\left(\frac{\tfrac{1}{6}(\mathbf{p}_j - \mathbf{p}_i) \cdot ((\mathbf{p}_k - \mathbf{p}_i) \times (\mathbf{p}_l - \mathbf{p}_i)) - V_0}{V_0}\right)^2 \qquad (6)$$

The preservation of the signed volume as it is calculated with the mixed product in Equation 6 is of major importance to our deformation model, since forces based on E_V preserve the initial orientation of the vectors in the mixed product. If a tetrahedron is inverted and the orientation of these vectors changes, the sign of the volume represented with the mixed product changes accordingly. Thus, inverting a tetrahedron results in forces that restore its original orientation.

The potential energies of our model are independent of rigid body modes of the object, that is, they conserve linear and angular momentum. The overall potential energy derived from our constraints can be interpreted as deformation energy of the object. The force at a mass point is given as the sum of all forces based on potential energies that consider a mass point. Thus, forces at mass points $\mathbf{p}_i$ with a given potential energy are derived as

$$\mathbf{F}^i(\mathbf{p}_0, \ldots, \mathbf{p}_{n-1}, \mathbf{v}_0, \ldots, \mathbf{v}_{n-1})$$

$$= \left(-kC - k_d \sum_{0 \le j < n} \frac{\partial C}{\partial \mathbf{p}_j} \mathbf{v}_j \right) \frac{\partial C}{\partial \mathbf{p}_i} \quad (7)$$

with $\mathbf{v}_i$ denoting the velocity of a mass point and k_d a damping coefficient. In contrast to similar approaches (Bridson, Fedkiw, & Anderson, 2002; Caramana, Burton, Shashkov, & Whalen, 1998; Provot, 1995), we do not explicitly bound potential energies or forces resulting from the energies. The direction of a force $\mathbf{F}$ based on a potential energy E corresponds to the negative gradient of E, that is, a dynamic simulation resulting from these forces reduces the deformation energy of an object. Damping significantly enhances the robustness of the dynamic simulation. While it is very useful to improve the stability of the numerical integration process, experiments have shown no improvement if damping is applied to forces resulting from E_A and E_V. Therefore, these are computed without a damping term.

In order to compute the dynamic behavior of our deformable models, Newton's equation of motion is applied to all mass points. Based on initial values for positions and velocities, internal forces at mass points resulting from our deformation energies and external forces, such as gravity, piecewise linear trajectories for all mass points are calculated employing a numerical integration scheme. Based on performance comparisons, we have chosen the Verlet scheme for the integration (Verlet, 1968). This technique has been very popular in molecular dynamics for decades and has recently been proposed in the context of physically based simulation of cloth, general mass-spring systems, and rigid bodies (Hauth, Etzmuss, Eberhardt, Klein, Sarlette, et al., 2002; Fuhrmann, Groβ, & Luckas, 2003; Kacic-Alesi, Nordenstam, & Bullock, 2003). The Verlet method has several advantages in environments with interacting, dynamically deforming objects. First, only one force computation is required. This is essential, since force computation is the most expensive part in the calculation of an integration step. Second, the integration results in positions with local discretization error of $O(h^4)$. This high accuracy allows for comparatively large time steps.

4   Collision Handling

In order to realistically process the interaction between deformable objects, efficient collision detection algorithms are required. Furthermore, the information provided by the collision detection approach should allow for an efficient and physically correct collision response. These two steps will be discussed in the following.

4.1 Collision Detection

We have developed an algorithm for the detection of collisions and self-collisions of deformable objects based on spatial hashing (Heidelberger, Teschner, Frauenfelder, & Gross, 2004). The algorithm implicitly subdivides R^3 into small cells and classifies all object primitives with respect to these cells.

In the first pass, the positions of all vertices are discretized with respect to a user-defined cell size. A hash function maps 3D vertices to a 1D index. Vertex and object information is stored in a hash table at this index. In addition to generating a hash value for each vertex, axis-aligned bounding boxes (AABB) of all tetrahedra are also computed based on their current deformed state. The second pass of the algorithm traverses all tetrahedra. Hash values are computed for all cells affected by the AABB of a tetrahedron. If a tetrahedron intersects with a cell, all vertices that have been associated with this cell in the first pass are checked for collision with this tetrahedron. Since the hash table index can contain more than one primitive from the same object, as well as primitives from different objects, collisions and self-collisions can both be detected. If a tetrahedron and a vertex are mapped to the same hash index, a penetration test taking into account the AABBs is performed. If it is positive, a collision has occurred and the contact response has to be determined for the colliding points.

Using a hash function for spatial subdivision is very efficient. While traditional spatial subdivision usually requires one preprocessing pass through all object primitives to estimate the global bounding box and the cell size, this pass can be omitted in our approach. Unfortu-
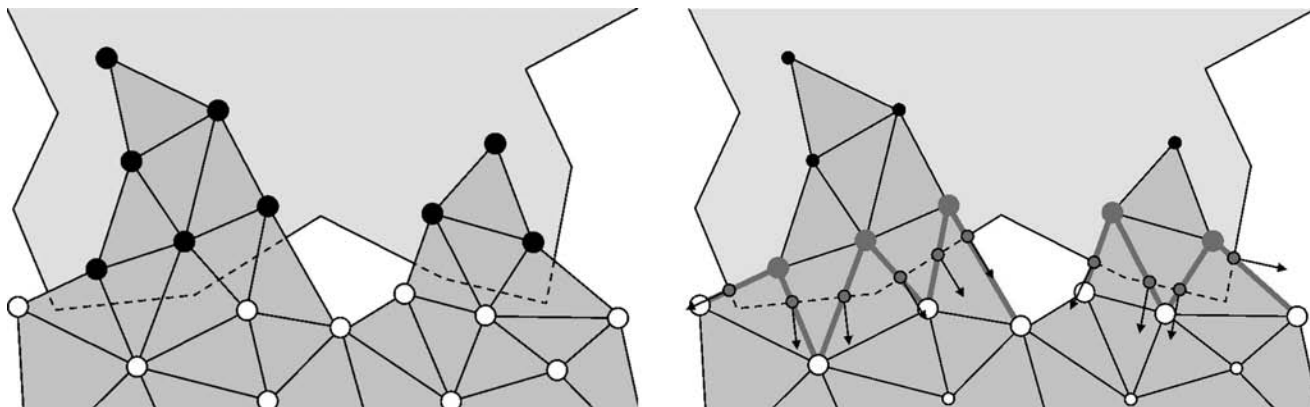
Figure 4. *First two stages of collision response. (a) Left: points classified as colliding (solid) or noncolliding (open). (b) Right: finding all intersection edges (thick line). Surface intersection points and normals are also computed.*

nately, the hash mechanism does not always provide a unique mapping to hash table entries. If different 3D cells are mapped to the same index, the performance of the proposed algorithm decreases. In order to reduce the number of index collisions, parameters of the collision detection algorithm have to be optimized. This includes the hash function, hash table size, and 3D cell size.

4.2 Collision Response

In order to simulate the behavior of the colliding objects, the appropriate collision response has to be determined. We have developed a method to compute consistent penetration depths and directions for colliding tetrahedral meshes with triangulated surfaces (Heidelberger, Teschner, Keiser, Mueller, & Gross, 2004). This information is used for determining penalty based separation forces. The algorithm is composed of three main stages.

The first stage identifies all colliding points with at least one adjacent noncolliding vertex as *border points* (Figure 4a). Thereafter, all *intersecting edges* that connect one noncolliding and one border point are found. Furthermore, the exact *intersection point* of each of these edges with the surface, as well as the corresponding surface normal is determined (Figure 4b). In order

to efficiently compute this information, the original spatial hashing approach has been extended to handle collisions between edges and surfaces.

The second stage approximates the penetration depth and direction for all border points based on the adjacent intersection points and surface normals. The penetration depth of a border point $\mathbf{p}_B$ is computed according to

$$d(\mathbf{p}_B) = \frac{\sum_{i=1}^{k}(\omega(\mathbf{x}_i,\mathbf{p}_B) \cdot (\mathbf{x}_i - \mathbf{p}_B) \cdot \mathbf{n}_i)}{\sum_{i=1}^{k}\omega(\mathbf{x}_i,\mathbf{p}_B)} \qquad (8)$$

with $\mathbf{n}_i$ denoting the unit surface normal of the penetrated object surface at the intersection point $\mathbf{x}_i$, k the number of intersection points adjacent to the border point, and ω a weighting function describing the influence of all adjacent intersection points on a border point. Finally, the normalized penetration direction of a border point is computed as a weighted average of the surface normals.

$$\mathbf{r}(\mathbf{p}) = \frac{\hat{\mathbf{r}}(\mathbf{p})}{\|\hat{\mathbf{r}}(\mathbf{p})\|} \text{ with } \hat{\mathbf{r}}(\mathbf{p}) = \frac{\sum_{i=1}^{k}(\omega(\mathbf{x}_i,\mathbf{p}) \cdot \mathbf{n}_i)}{\sum_{i=1}^{k}\omega(\mathbf{x}_i,\mathbf{p})} \qquad (9)$$

At the end of this stage, consistent penetration depths and directions have been computed for all border points (see Figure 5a). In contrast to existing approaches that consider only one distance, the
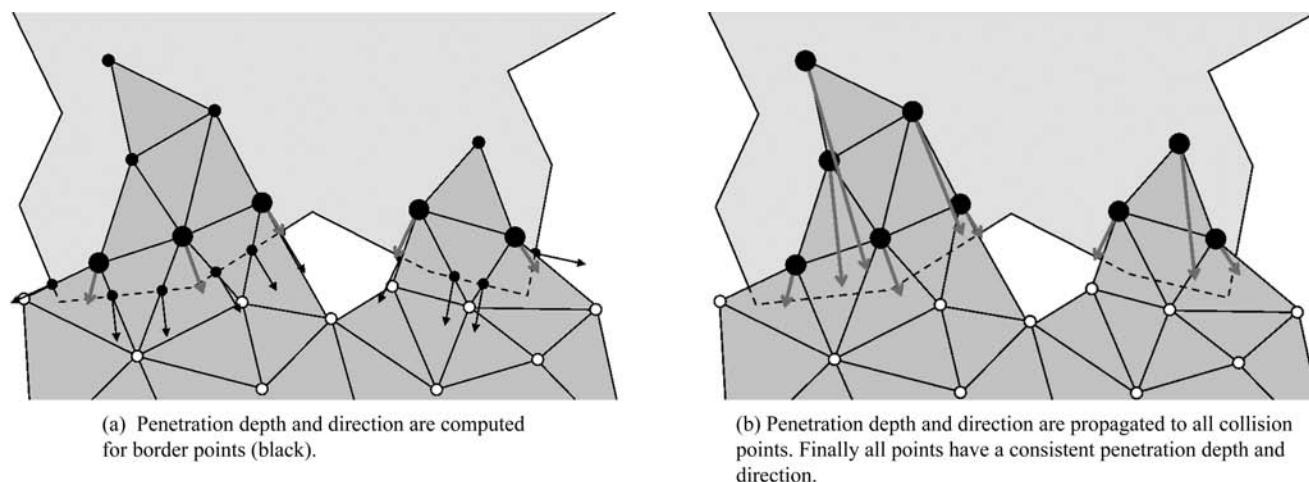
(a) Penetration depth and direction are computed for border points (black).

(b) Penetration depth and direction are propagated to all collision points. Finally all points have a consistent penetration depth and direction.

Figure 5. *Final stage and end result of collision response.*

weighted averaging of distances and directions provides a continuous behavior of the penetration depth function for small displacements and for colliding points that are adjacent to each other. Nonplausible penetration directions due to the surface discretization are avoided.

Based on the computed penetration depth information for border points, the third stage propagates the information to all other colliding points that are not border points. This is in contrast to existing approaches that compute the penetration depth for all points independently. This scheme tries to avoid problems with large penetrations. The propagation is an iterative process that consists of two substeps. First, the current border points are marked as *processed*. Second, a new set of border points is identified from all colliding points that are adjacent to one or more processed points. The iteration is aborted if no new border points are found. Otherwise, the penetration depth and direction for the new border points is computed based on the information available from all adjacent processed points. Finally, the normalized penetration direction is calculated as a weighted average of the penetration direction of the already processed adjacent points. At the end, all colliding points have a consistent penetration depth and direction assigned (Figure 5b).

5   Cutting

One key element of hysteroscopy training is the tissue ablation process. We focus on the resection of intra-uterine neoplasms, for instance myomas. Myomectomy with loop electrodes has to be carried out by stepwise shaving. The process is depicted in Figure 6. The loop electrode is positioned behind the pathology and then advanced toward the endoscopic camera from the back, cutting off tissue parts. As a consequence, the actual cutting process cannot be seen by the surgeon. Moreover, there is little, if any, resistance to the cut tool movement through the tissue. Therefore, in the current stage, we only use a simplified model of the interaction between the electrode loop and the deformable object during a cut.

5.1 Hybrid Cutting Approach

Including cutting procedures into our simulator system necessitates stable real-time updates of the underlying mesh representations. To this end, we have developed a new hybrid cutting approach for hysteroscopic resection (Steinemann, Harders, Szekely, & Gross, 2006). It combines existing techniques of decomposing tetrahedra into smaller elements and cutting entirely along existing edges and faces. This enables us
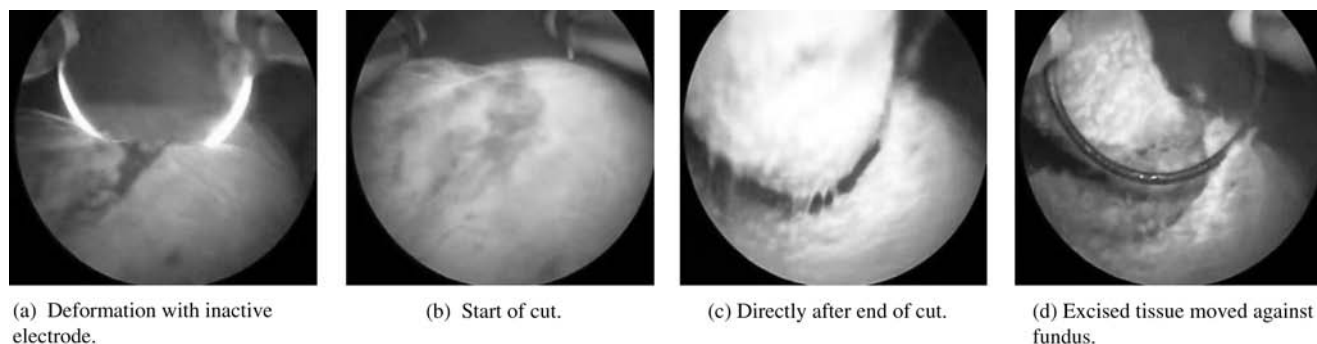
(a) Deformation with inactive electrode.

(b) Start of cut.

(c) Directly after end of cut.

(d) Excised tissue moved against fundus.

Figure 6. *Steps of shaving process with loop electrode during myomectomy.*

to approximate a given cutting surface more closely while avoiding the creation of degenerate tetrahedra, thus keeping the physical simulation stable.

Since the cutting process is not fully visible, nonprogressive cutting is a reasonable approximation for our purpose. Object meshes are decomposed only once they have been completely traversed. A cut surface is defined according to the movement of the activated loop electrode through the soft tissue. Once it has been obtained, we determine which of the traversed tetrahedra can be simply separated from each other and which should be actually decomposed into smaller subelements. This differentiation is performed with regard to the distance of the cut surface to the vertices of the tetrahedra. There are five rotationally invariant schemes to decompose a tetrahedron that has at least one of its edges cut. In addition, there are three schemes where one, two, or three existing nodes are separated. After applying the improved decomposition scheme, it is still possible that elements with very different sizes are created, which decreases the stability of the simulation. Therefore, to further improve mesh quality after a cut, we perform a sequence of local mesh enhancement steps. To this end, a set of nodes in the neighborhood of a cut is selected for adjustment. To preserve the shape of the model, only interior nodes are free to move in all directions. Nodes on the cut surface can translate only along the surface, while those on the original surface or surfaces resulting from an earlier cut are fixed. All edges incident to a node are considered as springs whose rest lengths equal the average length of those

edges. All nodes are considered as mass points with equal masses. To enforce an even spacing between nodes, forces resulting from the Lennard-Jones potential energy function (Lennard-Jones, 1931) are applied. The movement of a node is governed by Newton's equation of motion. To avoid oscillations, the system is highly damped. Improvement of the mesh quality requires only a few iteration steps. Finally, after a cut has been performed, we update the mesh parameters to keep the underlying deformation simulation stable. An example of the cutting process in the simulation is depicted in Figure 7.

6 Bleeding

Bleeding plays a key role in hysteroscopy, since it primarily determines the visibility of the surgical scene and can cause critical complications. Therefore, a realistic bleeding representation not only enhances the visual realism of the simulator, but offers the possibility for the trainee to practice how to handle such challenges properly. We use a marker particle approach similar to Foster and Metaxas (1996) to represent the blood flow in the uterine cavity.

6.1 Physical Model

The primary objective is to achieve a visually convincing appearance of the interaction of blood and the distension fluid. Hence, only the most important factors
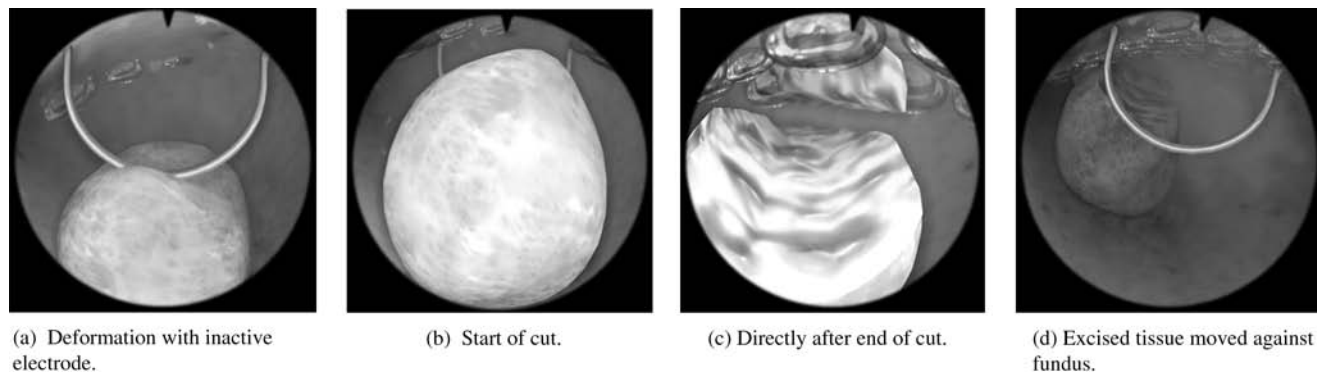
(a) Deformation with inactive electrode.

(b) Start of cut.

(c) Directly after end of cut.

(d) Excised tissue moved against fundus.

Figure 7. *Steps of myoma shaving process in virtual reality.*

need to be captured and physical details such as the non-Newtonian properties of the blood are neglected. The following key characteristics have to be considered by the model: the speed of bleeding, which can range from slow oozing to fast spurts; the recognizability of the blood flow trace, which can be from focal to diffuse; the speed of blood diffusion and dissolving; the contribution of the blood flow to the overall haze in the operation scene; and the influence of the irrigation fluid flow on the bleeding. All these features have to be accomplished efficiently from both a computational and a rendering point of view. As a solution, we have chosen to use particles with dynamically textured billboards. The particles allow for the realistic movement, taking the actual flow of the distension fluid and the interaction with the environment into account, while the billboards with changing textures ensure an efficient rendering performance with configurable and versatile visual appearance.

6.2 Implementation

A bleeding source is considered as an emitter, which creates noninteracting particles during the period of bleeding. In general, the direction of the initial velocity of a particle is identical to the surface normal at the emitter point. The emission rate and velocity magnitude of the particles are in correlation with the intensity of bleeding defined by local physiology. In the scene, the particles are represented by billboards, that is, rectangles that always face the camera. As a particle ages, the size

of its billboard is gradually increased, and the applied texture is repeatedly exchanged. Generally speaking, the texture depicts ranges from an opaque spot to a spread-out, semi-transparent circular patch. The adjustment of the timing of the texture update on the billboard determines primarily the type of bleeding. The size and the texture alteration of the billboards together are responsible for representing blood diffusion. The different stages of a growing, circularly symmetric 2D blob are stacked into a 3D texture block and stored in the graphics card texture memory. With this solution, changing the third texture coordinate of a billboard results in the desired dynamic effect. The intensity and alpha values of the 2D patches drop off in a Gaussian manner from the center. This implies that the transparency of the billboard is increasing from the center, which provides good blending in areas where the textured rectangles partially overlap. In the final phase of the rendering, the textured billboards are alpha-blended. This results in a volumetric effect with relatively low burden on the rendering engine. In addition, volumetric fog effects are used to augment the visual representation of the staining of the liquid with blood during heavy bleeding (Figure 8). The influence of the emitter with its initial velocity on a particle is decaying over time, while the influence of the external velocity field, induced by the flow of the distension fluid, is increasing. In each time-step, the particles with positions $\mathbf{p}_{flow}^{t}$ are advected by a simple Euler-step: $\mathbf{p}_{flow}^{t+1} = \mathbf{p}_{flow}^{t} + \mathrm{d}t \cdot \mathbf{v}_{particle}$. The $\mathbf{v}_{particle}$ is the weighted sum of the emitter and external flow field

(a) Traces of blood flowing from cut vessels.

(b) Scene in wireframe showing outlines of blood billboards.

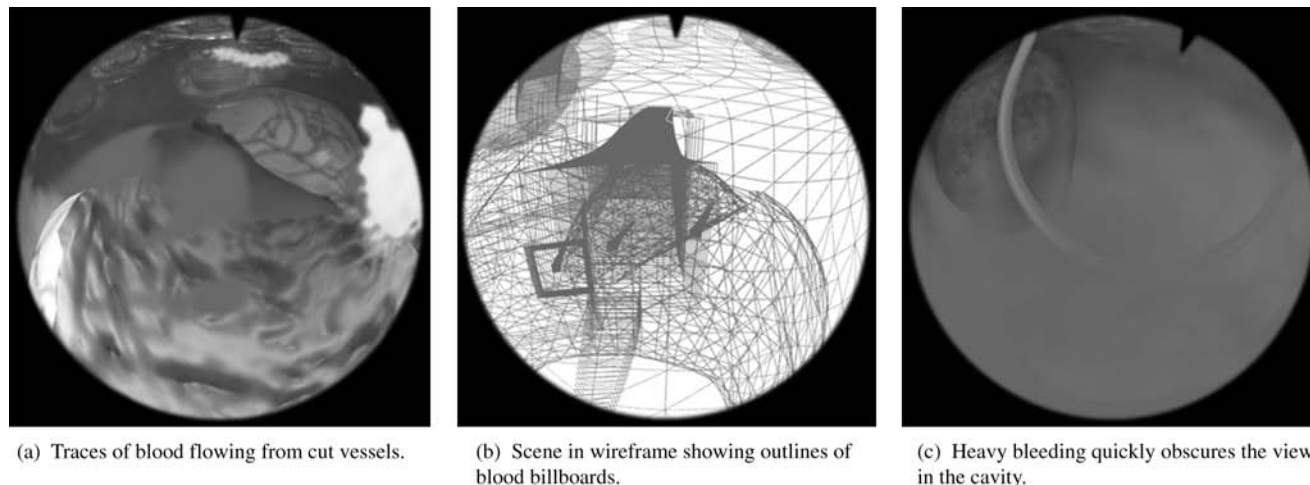(c) Heavy bleeding quickly obscures the view in the cavity.

Figure 8. *Example of bleeding in the uterine cavity.*

velocity components, with weights changing during the lifetime of a particle. The way a bleeding stops can be different depending on its type. In the most simple case, it stops after a given time; however, it can also be affected by coagulation procedures or pressure increase during the change of the hydrometra.

7   Scene Visualization

For visualization of the surgical scene, we have developed a graphics engine that handles a vast number of visual cues (Bachofen, Zatonyi, Harders, Frueh, & Thaler, 2006). A scene graph providing an organized hierarchy of the scene's triangle surfaces and visual properties is responsible for data handling. The tree is fully dynamic, allowing a runtime update of the data structures whenever necessary, for example, after cutting. The system makes use of graphics hardware support; however, if specific functionality is not supported by a configuration, the respective effect is automatically disabled.

Mesh geometries as well as object textures are generated with our scenario definition framework (Sierra et al., 2004) based on real patient data. New textures are synthesized and automatically mapped onto the object geometries. A further element of the graphics engine is the endoscopic camera model, which is composed of

scene lighting, lens distortion, and camera focus. The lighting is calculated by combining different rendering techniques, for example, bump mapping for rough surfaces or environment mapping for specular materials (Figure 9). Real-time distortion is achieved by rendering the scene to a texture which is then mapped on a distorted screen-sized quadrangle. For simulating the focus effect, we apply a depth-of-field technique to blur the distorted quadrangle depending on the focal distance. Figures 10(a) and 10(b) show a scene without and with distortion and blurring. Another visualization component is the proper handling of different floating objects inside of the fluid-filled cavity. These have to change their behavior according to the flow of the distension fluid, that is adjusted by the user via the valves of the hysteroscope. These objects are endometrial material, floating tissue fragments, and bubbles. Instead of rendering these as 3D objects, they are represented by billboards. A view of a complete virtual endoscopic image generated by this engine is shown in Figure 10c.

8   Haptic Interface

The haptic module serves two major functions—it provides the interface between surgeon and simulation to track the surgical gestures with the instrument, and to dis-
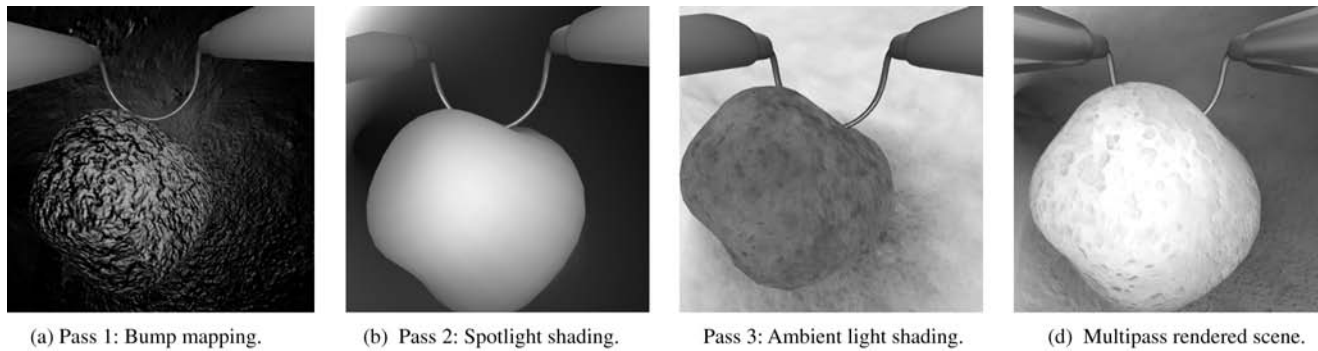
(a) Pass 1: Bump mapping.   (b) Pass 2: Spotlight shading.   Pass 3: Ambient light shading.   (d) Multipass rendered scene.

Figure 9. *Multipass rendering with all rendering passes.*

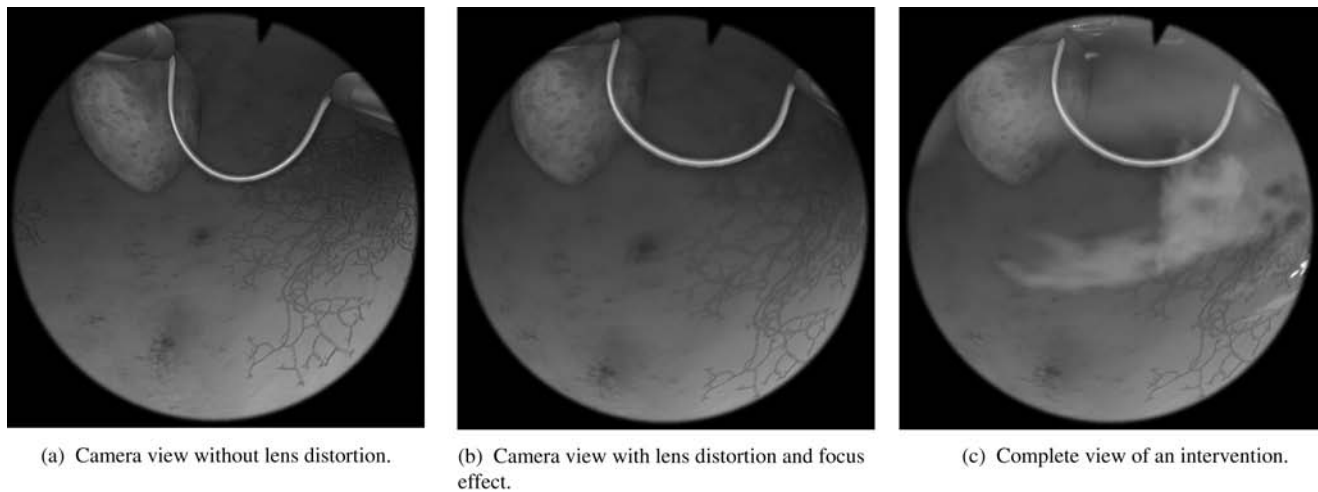(a) Camera view without lens distortion.   (b) Camera view with lens distortion and focus effect.   (c) Complete view of an intervention.

Figure 10. *Different surgical scene visualization effects.*

play force feedback to the trainee (Harders, Spaelter, Leskovsky, Szekely, & Bleuler, 2007). To enable user immersion into the training environment, the interaction metaphors should be the same as during real surgery. Therefore, an actual surgical instrument has been adapted in order to allow natural control of the intervention. A haptic mechanism (Spaelter, Moix, Ilic, Bajka, & Bleuler, 2004), allowing the complete removal of the instrument, provides force feedback calculated by the haptic rendering module.
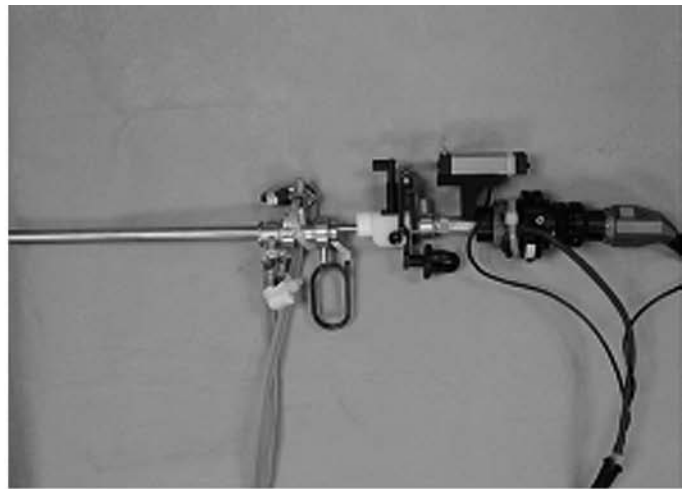
8.1 Haptic Interface

During a training session, the surgeon manipulates an original resectoscope for hysteroscopy (Figure 11),

which was slightly modified to integrate sensors for the inlet and outlet valve positions, camera angle, camera focus, and cutting tool position. Signal and power cables of the sensors are hidden in the unused fluid tubes or in standard instrument cables. The tool can be completely disassembled and reassembled—for instance, at the start of a training session, the fluid flow tubes have to be fitted to the instrument shaft. Force feedback is generated by a haptic mechanism into which the surgical instrument can be seamlessly inserted. The device allows us to insert and completely remove the surgery tool at simulation start and during a training session in order to fully match the actual interventions. The treatment of the uterus demands a large workspace ($\pm 60°$
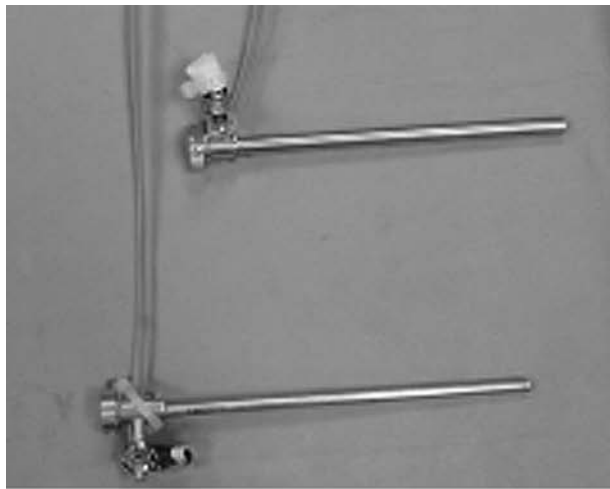
(a) Hysteroscope in- and outflow tubes and valves.

(b) Assembled tool with cutting mechanism, camera, and instument shaft.

Figure 11. *Haptic interface based on real resectoscope.*
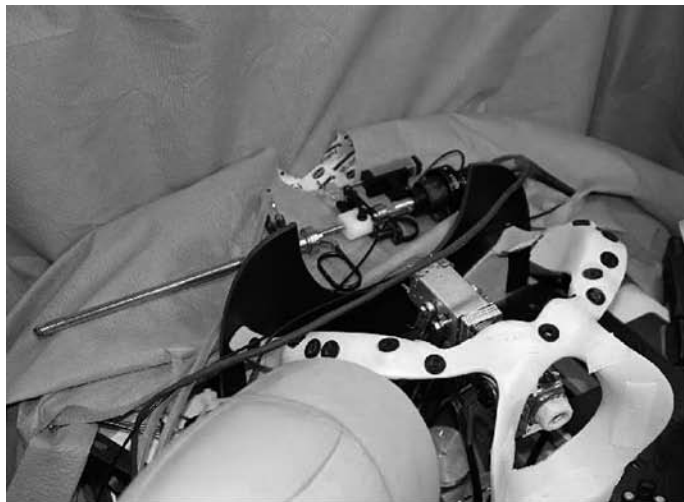
(a) Hysteroscope removed from frame.

(b) Inserted surgical tool.

Figure 12. *Haptic interface for hysteroscopy.*

pitch and yaw), especially as the anatomy of the uterus can vary within a wide range between individuals. At the same time, the device has to be compact due to the confined space within the female dummy torso (Figure 12). The base linkage has two degrees-of-freedom (DOF) for spherical displacement around a virtual pivot point. Inertia is reduced by fixing the actuators of the parallel

structure to the base. The virtual pivot can be placed in free space without mechanical connection to linkages. This allows us to hide the mechanical structure inside the patient dummy torso. A serially attached head with a two DOF friction drive provides tracking and force feedback for the tool translation along and rotation around the tool axis. The manipulator can transmit

pitch and yaw torques up to 0.5 Nm, and roll torques of 0.02 Nm. For the translational DOF, which is relevant for tissue probing, forces within the range 0.04–2 N can be generated (Spaelter, Samur, Ilic, Moix, & Bleuler, 2006).

8.2 Haptic Rendering

Displacement and force data are exchanged via UDP socket communication between the low-level impedance control of the haptic device and the VR environment. Problems due to out-of-order packets were not experienced and potentially also reduced due to low-pass filtering. The control loop for stable and transparent haptic interaction runs at >1 kHz under the real-time operating system RTAI-Linux. Although active human motion control capabilities rarely exceed 10 Hz, the haptic rendering of contact with virtual objects, as well as the sensitivity of the human hand to vibrations at much higher frequencies, make comparably higher update rates necessary (Sharpe, 1988). Virtual coupling techniques provide a data exchange between the fast haptic control loop and the slower virtual environment. In order to increase haptic realism, the gravitation, inertia, and friction of the haptic device are actively compensated for or reduced by control (Spaelter et al., 2004, 2006). In order to generate force feedback, we follow a point-based haptic proxy paradigm (Ruspini, Kolarov, & Khatib, 1997). We apply this technique to single as well as multiple interaction points. During the simulation we make full use of the dual object representation. Tetrahedral meshes are employed for collision detection and the calculation of tissue deformation, while surface meshes are used for visualization and local proxy updates.

Surgical tools are approximated by a collection of points. A proxy point is updated according to the movement of the tip of the surgical tool and, in case of a collision, it is restricted to the surface of the virtual object, while locally minimizing the distance to the tip. Collisions are detected via the spatial hashing algorithm. In case of a collision, the force applied to the deformable object and the corresponding haptic feedback is determined. In the haptic proxy model, a linear spring connects the proxy and the tip. Thus, based on the penetration depth, an interaction force can be obtained. It is distributed to the nodes of the involved surface triangle, resulting in the external force vectors for the computation of object deformation.

9   System Framework

As discussed in the previous sections, a number of specialized individual components have been developed for the hysteroscopy simulator. While these separate modules already have a high level of complexity, integration and control of all elements in a coherent system is also not straightforward. Difficulties arise due to varying requirements for update rates, different model representations, or necessary synchronization of parallel threads. At the same time, we have to comply with boundary conditions arising from a software engineering point of view (e.g., maintainability, extensibility, scalability). In order to handle this highly modular real-time system, and achieve robustness and reliability without sacrificing performance, specialized approaches were taken (Tuchschmid et al., 2006).

An object-oriented domain model was defined, which comprises instances of actual simulation objects and the simulation engines. An overview of the integrated framework is depicted in Figure 13. The interaction between all components is scheduled via a site manager. A central element of the model is the dual object representation, which has to be taken into account by all simulation engines. The high-resolution triangular surface is mapped to the lower resolution tetrahedral model and updated according to its changes. All modules involved in the mechanics of tool-tissue interaction build the core of the simulator, and are thus part of the main loop. It has the highest priority, since it is involved in the generation of haptic feedback. Further units within the framework address the flow simulation and the graphical rendering.

In order to simplify interaction between the different engines and to relieve the engine programmer from dealing with low-level synchronization issues, we developed control mechanisms to prevent deadlocks, starva-
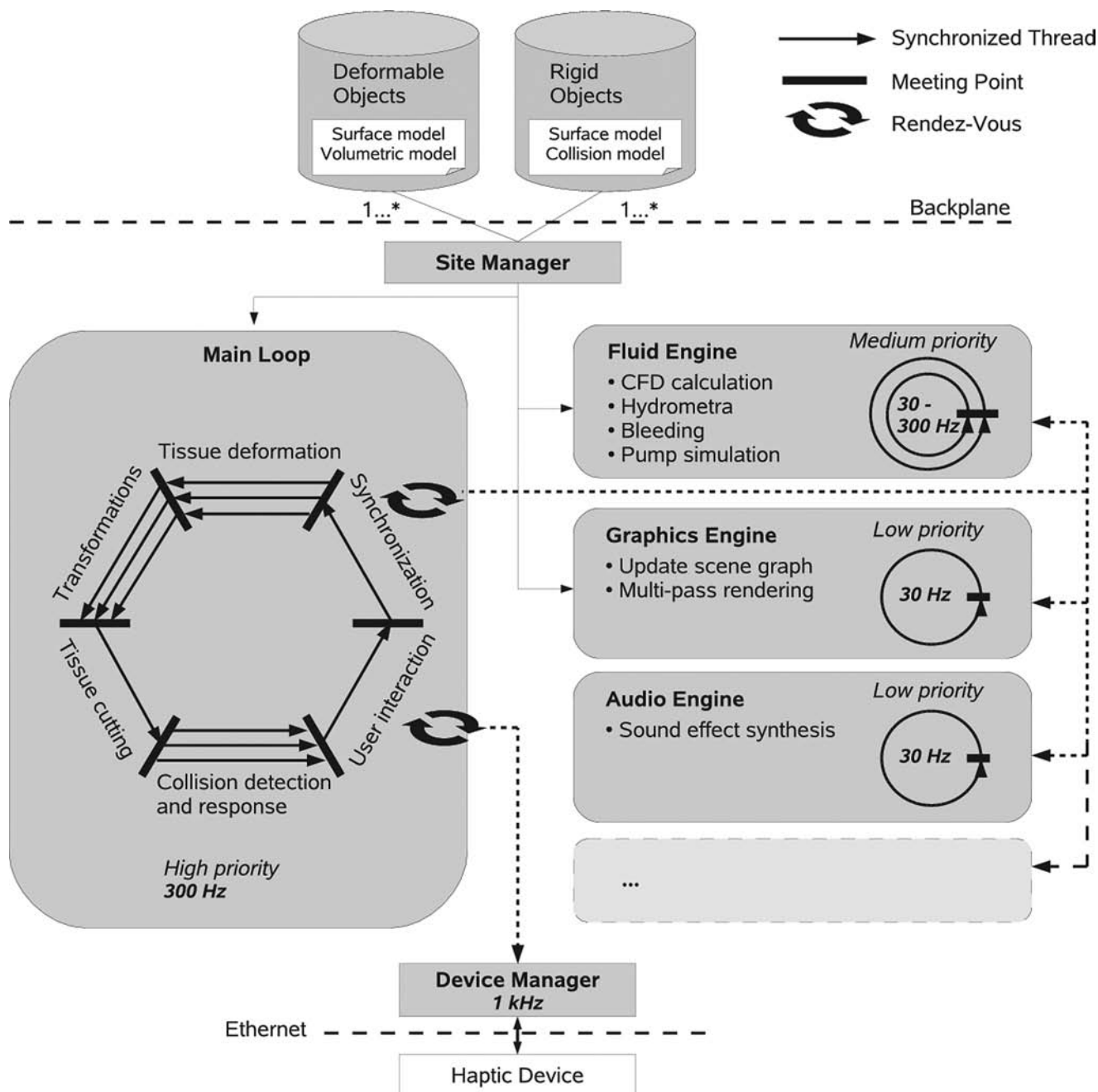
Figure 13. *Overview of the software framework (from Tuchschmid et al., 2006, with kind permission of Springer Science and Business Media).*

tion, or priority inversion. Meeting points, rendezvous, and synchronized threads are concepts stemming from the domain of parallel programming, which we adapted to our needs. Meeting points allow us to coordinate groups of worker threads and build arbitrary sequences of parallel blocks where each block has an arbitrary amount of parallel acting workers. The rendezvous tool supports the controlled exchange of information, for

(a) View of the complete OR.

(b) Close up of simulator prototype.

Figure 14. *Immersive environment.*

instance, the appearance data of all displayable objects between independent threads. Finally, synchronized threads are used to encapsulate synchronization policies. The implementation of these tools is based on the POSIX library.

Since several synchronized threads can be grouped to run in parallel, we also use them as the main scheme for intramodular parallelization. To achieve this, we first identify slow, parallelizable code sections. A domain decomposition is then carried out for an appropriate number of worker threads. A meeting point is added to coordinate the workers, while communication takes place via a rendezvous. Using this scheme, we parallelized time intensive modules of the simulation, that is, the fluid computation and the collision detection. For larger datasets, the parallelization scales almost perfectly.

10 Prototype System

A key target of our simulator is to enable higher level skill training. In this respect, the sense of presence (Slater & Steed, 2000) plays an important role in the training effect which can be achieved. To enable user immersion into the training environment, the surrounding and interaction metaphors should be the same as during real interventions. To this end, we replicated an

operating room (OR) in our lab, provided standard hysteroscopic tools for interaction, and generated a new virtual patient for every session (Harders, Bajka, Spaelter, Tuchschmid, & Szekely, 2006). In this setting, the training starts when the trainee enters the OR, and it ends when the trainee leaves the room. The first element of our immersive setup is the training environment. We outfitted our OR with real tools and equipment (Figure 14a). The female pelvic region is represented by a Limbs&Things model. The first step of an intervention is to enable access to the cervix with the speculum, and then fixate it with surgical forceps. Thereafter, the hysteroscope is inserted into the uterine cavity. The tool is advanced through the cervix and connects with the haptic mechanism inside of the pelvic dummy. The connection is seamless, without any noticeable mechanical constraints. A further element in the immersive environment is the auditory feedback. Characteristic sound sources are the electrocardiogram, the hysteroscopy pump, the artificial respiration, and warning sounds for electrotomy and coagulation. These provide important cues to the gynecologist, and thus have to be integrated into a training system. The sounds were obtained from real surgical equipment, partly during actual interventions. In the simulation, audio cues are synthesized in real time and adapted to the actual surgical action, like the regulation of the inflow/out-
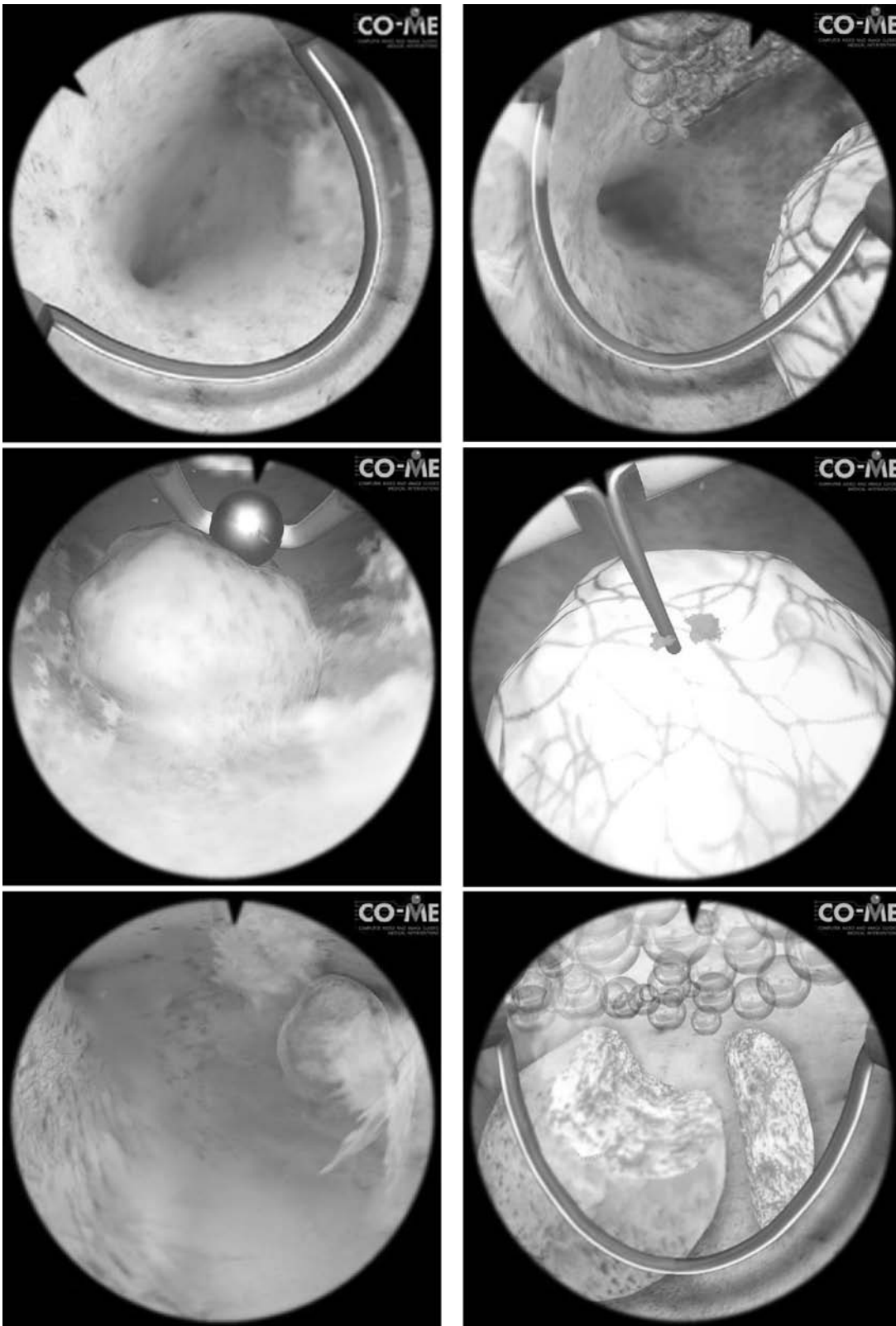
Figure 15. *Additional screenshots showing various virtual interventions.*

(a) Novice 1st trial.   (b) Novice 5th trial.   (c) Expert 1st trial.
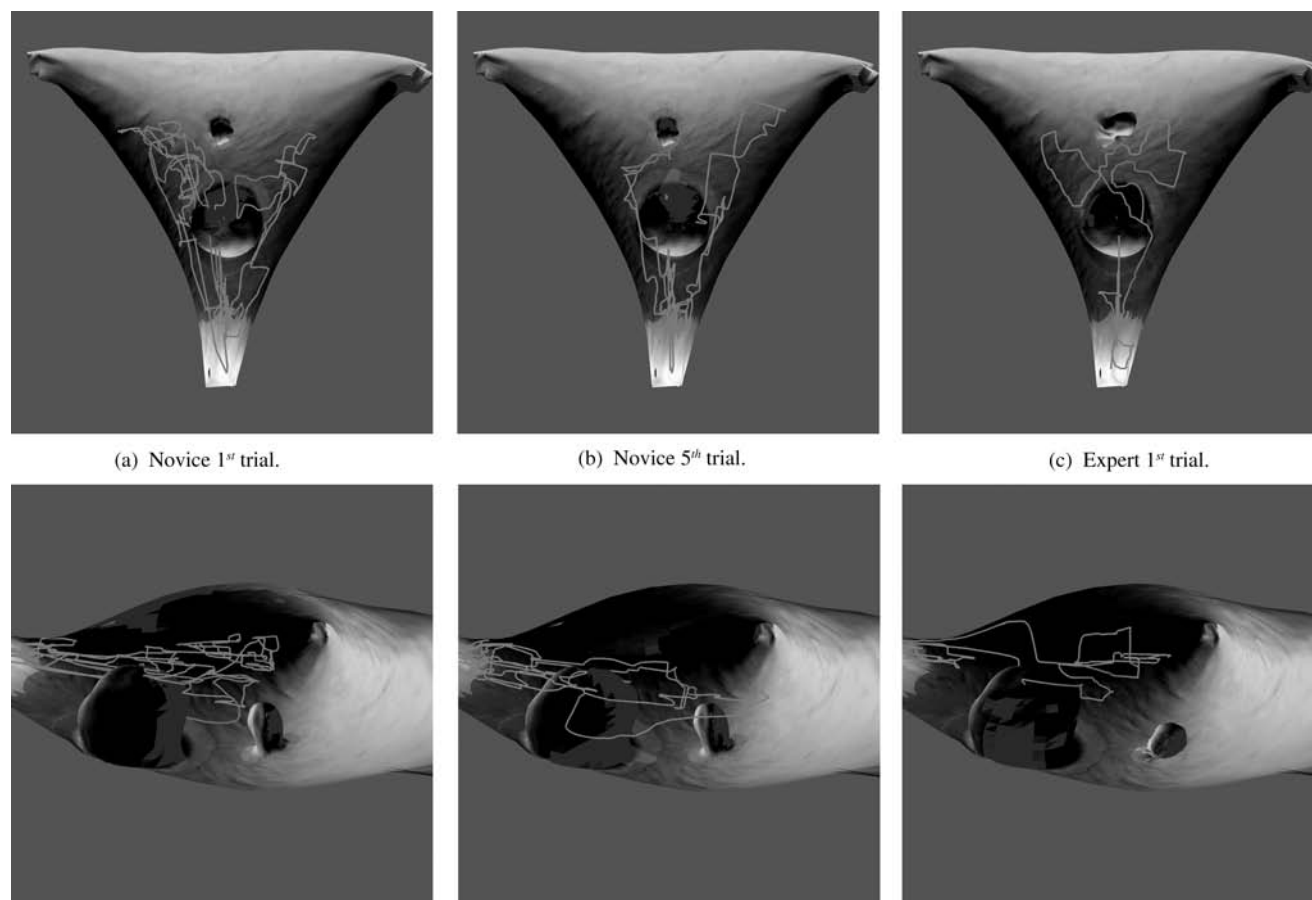
Figure 16. *Visualized uterus surface, camera navigation path, and wall collisions during novice and expert interventions (top: view from above, bottom: view from left patient side).*

flow. The complete prototype system is depicted in Figure 14b. Further details on our hysteroscopy simulator are available on the project Web page (http://www.hystsim.ethz.ch). Additional screenshots of the virtual intervention are depicted in Figure 15.

11   Simulator Evaluation

In order to obtain an objective, structured, and automated assessment of surgical performance in virtual diagnostic hysteroscopy, we implemented robust evaluation methods (Tuchschmid, Bajka, Bachofen, Szekely, & Harders, 2007). To this end, we first carried out a hierarchical task decomposition to break down the sur-

gery into its elements. A panel of expert surgeons split up the procedure of diagnostic hysteroscopy into surgical steps (e.g., inspect uterine cavity), substeps (e.g., identify right tubal orifice), tasks (e.g., centralize tubal orifice in view), and low-level activities such as tool motions. The latter are only used to evaluate the completion of a task or substep, while acquisition of the former aims at measuring procedural skills on a higher level.

Thereafter, we implemented surgery-specific performance metrics matching the hierarchical decomposition and integrated these into the simulator setup. These metrics include, for instance, the amount of distension media used, the quality of the endoscopic view, the safe handling of the hysteroscope, tool-wall collisions, the economy of movement, or the quantification of the

properly visualized surface. To demonstrate the reliability and validity of the implemented metrics, both novice residents and experienced clinicians carried out simulated diagnostic hysteroscopies. For each virtual intervention, a comprehensive report summarizing the quantitative metrics was created and used for providing instructive feedback. As an example, Figure 16 shows the movement of the endoscopic tool in the uterine cavity. The navigation path of the camera is color-coded with respect to the distance to the uterine surface. It can be seen that in the first trial a novice exhibits a worse economy of movement as compared to an experienced surgeon (>200 hysteroscopies performed). However, after a few sessions with the system, the performance of the former shows a significant progressive improvement. While these early qualitative findings already indicate the training effect of the system, further studies are needed. At the moment, a structured study is carried out with novices and experienced surgeons of several Swiss Clinics of Gynecology to quantify the performance improvement.

12 Conclusion

Our current research targets the development of a high-fidelity training simulator for hysteroscopic interventions. We provided an overview of all components of our current prototype system, including the real-time interactive simulation modules, the simulator hardware, and the evaluation framework. Our simulator allows for the rehearsal of several aspects of hysteroscopy, for instance, correct fluid management, handling of excessive bleeding, appropriate removal of intra-uterine tumors, and correct handling of surgical instruments. Early results of an evaluation study demonstrate the training effect of the system. Future work will focus on the quantification of the effect of simulation fidelity on the training outcome. Moreover, the integration of the simulator in the medical curriculum will be pushed forward. Other extensions are the combination of the simulator with an anesthesia simulation system to enable complete OR team training. Finally, the application of promising alternatives to currently used techniques will be examined, for example, the usage of meshless representations for deformation models.

References

ACOG. (1994). Hysteroscopy. *International Journal of Gynecological Obstetrics Tech. Bulletin, 45*(2), 175–180.

Anderson, J. D. (1995). *Computational Fluid Dynamics: The Basics with Applications.* New York: McGraw-Hill.

Bachofen, D., Zatonyi, J., Harders, M., Frueh, P., & Thaler, M. (2006). Enhancing the visual realism of hysteroscopy simulation. *Proceedings of Medicine Meets Virtual Reality,* 31–37.

Bianchi, G., Solenthaler, B., Szekely, G., & Harders, M. (2004). Simultaneous topology and stiffness identification for mass-spring models based on FEM reference deformations. *Proceedings of MICCA,* 293–301.

Bridson, R., Fedkiw, R., & Anderson, J. (2002). Robust treatment of collisions, contact and friction for cloth animation. *SIGGRAPH '02: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques,* 594–603.

Caramana, E. J., Burton, D. E., Shashkov, M. J., & Whalen, P. P. (1998). The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics 146*(1), 227–262.

Foster, N., & Metaxas, D. (1996). Realistic animation of liquids. *Graphical Models and Image Processing, 58*(5), 471–483.

Fuhrmann, A., Groβ, C., & Luckas, V. (2003). Interactive animation of cloth including self collision detection. *International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision,* 141–148.

Harders, M., Bajka, M., Spaelter, U., Tuchschmid, S., & Szekely, G. (2006). Highly-realistic, immersive training environment for hysteroscopy. *Proceedings of Medicine Meets Virtual Reality,* 176–181.

Harders, M., Spaelter, U., Leskovsky, P., Szekely, G., &

Bleuler, H. (2007). Haptic interface module for hysteroscopy simulator system. *Proceedings of Medicine Meets Virtual Reality,* 167–169.

Hauth, M., Etzmuss, O., Eberhardt, B., Klein, R., Sarlette, R., et al. (2002). Cloth animation and rendering. *Proceedings of Eurographics*, tutorial notes.

Heidelberger, B., Teschner, M., Frauenfelder, T., & Gross, M. (2004). Collision handling of deformable anatomical models for real-time surgery simulation. *Journal of Technology and Health Care, 12*(3), 235–243.

Heidelberger, B., Teschner, M., Keiser, R., Mueller, M., & Gross, M. (2004). Consistent penetration depth estimation for deformable collision response. *Proceedings of Vision, Modeling, Visualization,* 339–346.

Immersion Medical. (2005). *AccuTouch System*. Available at www.imersion.com. Accessed March, 2005.

Kacic-Alesi, Z., Nordenstam, M., & Bullock, D. (2003). A practical dynamics system. *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation,* 7–16.

Lennard-Jones, J. (1931). Cohesion. *Proceedings of the Physical Society, 43,* 461–482.

Levy, J. (1996). Virtual reality hysteroscopy. *Journal of the American Association of Gynecological Laparoscopists, 3*(4), 25–26. Suppl.

Lim, F., Brown, I., McColl, R., Seligman, C., & Alsaraira, A. (2006). A visual graphic/haptic rendering model for hysteroscopic procedures. *Australasian Physical & Engineering Sciences Medicine, 29*(1), 57–61.

Liu, A., Tendick, F., Cleary, K., & Kaufmann, C. (2003). A survey of surgical simulation: Applications, technology, and education. *Presence: Teleoperators and Virtual Environments, 12*(6), 599–614.

Mencaglia, L., & Hamou, E. (2001). *Manual of gynecological hysteroscopy—Diagnosis and surgery*. Germany: Endo-Press.

Montgomery, K., Heinrichs, L., Bruyns, C., Wildermuth, S., Hasser, C., et al. (2001). Surgical simulator for hysteroscopy: A case study of visualization in surgical training. In *IEEE Visualization,* 449–452.

Muller-Wittig, W., Bisler, A., Bockholt, U., Arcos, J. L., Oppelt, P., et al. (2001). Lahystotrain development and evaluation of a complex training system for hysteroscopy. *Proceedings of Medicine Meets Virtual Reality,* 336–340.

Nava, A., Mazza, E., Kleinermann, F., Avis, N., McClure, J., & Bajka, M. (2004). Evaluation of the mechanical properties of human liver and kidney through aspiration experiments. *Technology and Health Care, 12*(3), 269–280.

Paget, R., Harders, M., & Szekely, G. (2005). A framework for coherent texturing in surgical simulators. *13th Pacific Conference on Computer Graphics and Applications,* 112–114.

Petrozza, J. (2008). *Hysteroscopy*. Available at http://www.emedicine.com/med/topic3314.htm.

Pinion, S., Parkin, D., Abramovich, D., Naji, A., Alexander, D., et al. (1994). Randomised trial of hysterectomy, endometrial laser ablation, and transcervical endometrial resection for dysfunctional uterine bleeding. *British Medical Journal, 309,* 979–983.

Provot, X. (1995). Deformation constraints in a mass-spring model to describe rigid cloth behavior. *Graphics Interface '95,* 147–154.

Ruspini, D., Kolarov, K., & Khatib, O. (1997). The haptic display of complex graphical environments. *Computer Graphics (SIGGRAPH '97 Conference Proceedings),* 345–352.

Sharpe, J. (1988). Technical and human operational requirements for skill transfer in teleoperations. *Proceedings of the International Symposium on Teleoperation and Control,* 175–187.

Sierra, R., Bajka, M., Karadogan, C., Szekely, G., & Harders, M. (2004). Coherent scene generation for surgical simulators. *Proceedings of MICCAI,* 221–229.

Sierra, R., Zatonyi, J., Bajka, M., Szekely, G., & Harders, M. (2005). Hydrometra simulation for VR-based hysteroscopy training. *Proceedings of MICCAI,* 575–582.

Sierra, R., Zsemlye, G., Szekely, G., & Bajka, M. (2006). Generation of variable anatomical models for surgical training simulators. *Medical Image Analysis, 10*(2), 275–285.

Slater, M., & Steed, A. (2000). A virtual presence counter. *Presence: Teleoperators and Virtual Environments, 9*(5), 413–434.

Spaelter, U., Moix, T., Ilic, D., Bajka, M., & Bleuler, H. (2004). A 4-DOF haptic device for hysteroscopy. *Proceedings of IEEE IROS,* 644–667.

Spaelter, U., Samur, E., Ilic, D., Moix, T., & Bleuler, H. (2006). A 2-DOF friction drive for haptic surgery simulation of hysteroscopy. *8th International IFAC Symposium on Robot Control,* 3257–3263.

Stam, J. (1999). Stable fluids. *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques,* 121–128.

Steinemann, D., Harders, M., Szekely, G., & Gross, M. (2006). Hybrid cutting of deformable solids. *Proceedings of IEEE Computer Society Conference on Virtual Reality,* 35–42.

Szczerba, D., & Szekely, G. (2002). Macroscopic modeling of vascular systems. *Medical Image Computing and*

Computer-Assisted Intervention, MICCAI 2002, vol. 2489 of *Lecture Notes in Computer Science,* 284–292.

Teschner, M., Heidelberger, B., Mueller, M., & Gross, M. (2004). A versatile and robust model for geometrically complex deformable solids. *Proceedings of Computer Graphics International,* 312–319.

Tuchschmid, S., Bajka, M., Bachofen, D., Szekely, G., & Harders, M. (2007). Objective assessment of surgical performance in hysteroscopy simulation. *Medicine Meets Virtual Reality,* 473–478.

Tuchschmid, S., Grassi, M., Bachofen, D., Frueh, P., Thaler, M., Szekely, G., et al. (2006). A flexible framework for highly-modular surgical simulation systems. *Proceedings of ISBMS'06,* 84–92.

Verlet, L. (1968). Computer "experiments" on classical fluids. II. Equilibrium correlation functions. *Physical Review, 165,* 201–214.

Weiss, S., Bajka, M., Nava, A., Mazza, E., & Niederer, P. (2004). A finite element model for the simulation of hydrometra. *Technology and Health Care, 12,* 259–267.