Real-Time Haptic Cutting of High-Resolution Soft Tissues

Jun WU [1], Rüdiger WESTERMANN and Christian DICK
Computer Graphics & Visualization Group, Technische Universität München, Germany

Abstract. We present our systematic efforts in advancing the computational performance of physically accurate soft tissue cutting simulation, which is at the core of surgery simulators in general. We demonstrate a real-time performance of 15 simulation frames per second for haptic soft tissue cutting of a deformable body at an effective resolution of 170,000 finite elements. This is achieved by the following innovative components: (1) a linked octree discretization of the deformable body, which allows for fast and robust topological modifications of the simulation domain, (2) a composite finite element formulation, which thoroughly reduces the number of simulation degrees of freedom and thus enables to carefully balance simulation performance and accuracy, (3) a highly efficient geometric multigrid solver for solving the linear systems of equations arising from implicit time integration, (4) an efficient collision detection algorithm that effectively exploits the composition structure, and (5) a stable haptic rendering algorithm for computing the feedback forces. Considering that our method increases the finite element resolution for physically accurate real-time soft tissue cutting simulation by an order of magnitude, our technique has a high potential to significantly advance the realism of surgery simulators.

Keywords. Surgery Simulation, Deformable Bodies, Cutting, Finite Elements, Collision Detection, Haptic Rendering

1. Introduction

The physically-based yet efficient and robust simulation of soft tissue cutting is a fundamental task in developing realistic surgery simulators. While deformable body simulation has been widely investigated and open source frameworks are available (e.g., [1,2]), few can support real-time cutting of high-resolution models.

From a technical point of view, simulating soft tissue cutting involves a) the simulation of the deformable body based on a computational model, b) the incorporation of cuts into the computational model, and c) collision detection and response. In particular, the physically accurate simulation of a deformable body based on the finite element method in combination with an implicit time integration scheme leads to a large, sparse linear system of equations that has to be solved in each time step.

[1]Corresponding Author: Jun Wu, Technische Universität München, Informatik 15 (Computer Graphik & Visualisierung), Boltzmannstraße 3, 85748 Garching, Germany; E-mail: jun.wu@tum.de.

Performing these operations in real-time is highly challenging, as for the simulation of cutting most simulation components must be updated or re-computed in each time step, with pre-computation in general not being applicable. This includes the update of the finite element model, where elements that are touched by the cutting blade have to be split, the update of the corresponding finite element matrices, the re-assembly of the element matrices into the linear system of equations, the re-initialization of the numerical solver, and the re-computation of acceleration data structures for collision detection. Moreover, if a high-resolution render surface is employed, cuts have to be incorporated into this surface as well, i.e., intersected triangles have to be split, and the newly created cutting surface has to be triangulated. These operations are computationally very intensive, and–considering the incorporation of cuts into volumetric and surface meshes, where ill-shaped elements can easily occur–difficult to realize in a robust way.

As a consequence, in order to meet the real-time requirements, most existing approaches for soft tissue cutting [3,4,5] use physically simplified models, e.g., mass-spring models, or rather low-resolution finite element models. Cutting is typically realized by completely removing the elements that are cut [5], or by strictly enforcing that cuts are aligned along existing element faces [6]. In addition, many approaches are based on an explicit time integration scheme [7] to circumvent solving a linear system of equations, which however (according to the CFL condition) requires using excessively small time step sizes (with the consequence of a non-realistic temporal behavior of the simulation), and/or excessively soft (and thus non-realistic) material properties.

In this paper we summarize our systematic efforts in advancing the simulation of soft tissue cutting. The demonstration of real-time performance for high-resolution soft tissue cutting clearly distinguishes our work from previous implementations.

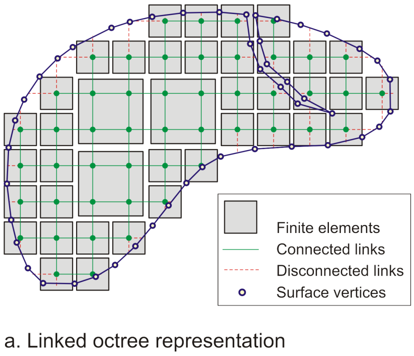
2. Methods

Key to our approach is the use of a *semi-regular hexahedral* finite element grid, in lieu of the commonly used *irregular tetrahedral* finite element grids [3]. A semi-regular hexahedral grid enables us to update the finite element model in a fast and robust way, without creation of ill-shaped elements, and allows us to efficiently generate geometric hierarchies as they are required for efficient numerical simulation [8,9] and collision detection [10]. In the following, we shortly present the individual components of our method.

2.1. Linked Octree Discretization

To represent the topology of the deformable body, we employ a linked volume representation. Face-adjacent hexahedral finite elements are connected via links, with six links emanating from each element. To model cuts, instead of deleting elements which leads to a volume loss [5], we mark links as disconnected at the exact intersection points of the links with the virtual blade. To allow for fine and detailed cuts, we use a hexahedral octree grid by starting with a uniform grid that is adaptively refined along cuts, down to a certain finest level (see Figure 1 left for an illustration).

A slight disadvantage of hexahedral elements is the jagged visual appearance for rendering. We solve this problem by reconstructing (during cutting, in each simulation frame) a smooth object surface mesh from the linked volume representation by means



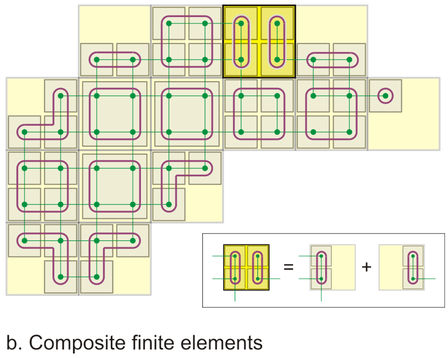


Figure 1. Left: A liver model is discretized into finite elements by means of an adaptive octree grid (gray-shaded cells; the grid resolution employed in the application is significantly higher than depicted). Finite elements are connected by links (solid green lines). Cutting is modeled by disconnecting links (dashed red lines). Surface vertices (blue-bordered dots) are computed from the dual grid of links. Note that adjacent elements share their vertices only if these elements are topologically connected. Right: Composite elements (light-yellow-shaded cells) are constructed by recursively merging topologically connected hexahedral elements (violet-framed subsets of element centers [green dots]) within 2^3 cell blocks of the underlying grid. A cut leads to duplicated composite elements (dark yellow), which are located at the same place but represent physically disconnected material parts.

of a dual contouring approach. The dual grid is the grid that is formed by the links used to model the connectivity between the cells. For each link that is cut by the blade, the distances from the intersection point to the link's endpoints as well as the normal of the blade at the intersection point are utilized to optimally place the surface vertices. Note that the original object surface is also considered as a cutting surface. The meshing procedure establishes a correspondence between the surface vertices and the vertices of the hexahedral finite element grid. Given this correspondence, the surface vertices can be bound to the simulation vertices and be moved according to the soft tissue deformation.

2.2. *Composite Finite Elements (CFEs)*

For a high-resolution adaptive octree discretization as we are using in our implementation, the number of hexahedral finite elements exceeds the real-time capabilities of current desktop PC/workstation platforms. To address this issue, we employ a composite finite element formulation. The basic idea is to subsume multiple finite elements into a single composite finite element, with the degrees of freedom (DOFs) of the individual elements being substituted with the DOFs of the composite elements by means of interpolation. In this way, the number of DOFs is reduced, allowing us to carefully balance performance and accuracy [9]. Based on our hexahedral discretization, constructing such a composite finite element grid is robust and efficient: composite elements are created by successively merging topologically connected hexahedral elements within each 2^3 cell block of the underlying grid, as illustrated in Figure 1 (right). Note that merging elements by only considering their spatial location would possibly produce a single composite element for mechanically separated material parts. To accurately represent the topology of the deformable body in the presence of cuts, we therefore analyze the connectivity between elements (represented by the links between elements) to possibly create multiple composite elements at the same location, each representing a separated material part on either side of a cut. The stiffness and mass matrices for each composite element are assembled from its associated hexahedral elements. It is worth noting that by using composite elements, although the numbers of DOFs is reduced, the computational model still considers both material properties and topology at the high resolution of the underlying hexahedral finite element grid.

To accurately simulate deformations with large rotations using the linear theory of elasticity, we employ the corotational formulation of strain. Since in the linear theory the linear infinitesimal strain tensor is employed, which interprets rotations as strains, the solution significantly diverges from the correct solution with increasing rotations within the deformation. The basic idea underlying the corotational formulation of strain is to remove the rotations before the linear strain is computed, and later re-apply the rotations to the resulting stresses.

2.3. Geometric Multigrid Solver

We apply the implicit Newmark time integration scheme to the spatially discretized equation of motion, which leads to a large, sparse linear system of equations in every simulation time step. An implicit time integration scheme is advantageous over an explicit scheme, as it allows for using considerably larger time steps and employing stiffer material properties. To solve the linear system of equations, we have developed a highly efficient geometric multigrid method [8]. A multigrid solver is optimal in the sense that it exhibits asymptotic linear runtime in the number of unknowns. Considering the error between the current approximate solution and the exact solution in an iterative solution scheme, standard relaxation schemes such as Jacobi or Gauss-Seidel relaxation effectively reduce high-frequency error components, but they are inefficient in reducing low-frequency error components. This causes the error reduction to stall after a few relaxation steps. The basic idea of multigrid is to improve the convergence by solving the system on a hierarchy of successively coarser grids, such that each frequency range of the error can effectively be reduced on the appropriate scale. Using a hexahedral discretization allows us to efficiently re-compute the coarse grid hierarchy in each simulation frame when a cut is being applied.

2.4. Collision Detection for CFEs

Collision detection in soft tissue cutting is especially time consuming, since new geometric primitives are created on-the-fly. This makes pre-computation of acceleration data structures (e.g., bounding volume hierarchies) impractical. As a consequence of cutting, connected parts of the same organ may be separated. It is therefore necessary to consistently detect both inter- and intra- collisions. Moreover, a quantitative measure of the penetration is desired for robust collision response.

We have developed a collision detection algorithm [10] that exploits the specific characteristics of CFEs, i.e., exhibiting a reduced number of simulation DOFs. The workflow of our collision detection is to first identify, in the deformed configuration, potentially overlapping pairs of a vertex and a composite element. The vertex can be from the same organ or from other objects (e.g., the scalpel), thus both self-collisions as well as scalpel-organ collisions are handled. The runtime of this broad phase depends on the number of CFEs, instead of the number of hexahedral elements. This leads to a speedup factor which scales exponentially with respect to the level of composition. In the narrow phase, each vertex/composite finite element pair is then transformed from the deformed configuration into the reference configuration, where the penetration depth and direction of the vertex is determined by means of a signed distance field. A problem in evaluating the penetration depth based on a signed distance field in the presence of cuts is that the voxels on both sides of the cut are flagged as inside. To this end, we have developed a topology-aware interpolation scheme. By flipping the signs of distance field voxels based on analyzing their connectivity, the error of penetration depth interpolation near cutting surfaces is reduced to sub-voxel level.

2.5. Haptic Rendering of Cutting

To employ our method for virtual surgery training, we have integrated haptic feedback into the simulation. To obtain an intuitive feedback force, we use a damping force model: the force direction is opposite to the direction of movement, and the force magnitude is proportional to the speed of movement and the contact volume between the scalpel and the soft tissue. To compute the force, the scalpel is discretized into small segments. We first compute an elementary cutting force for each segment. The overall cutting force is then obtained as the sum of these forces.

The velocity of the manipulated scalpel is estimated from the position information provided by the haptic device. This information is known to be noisy, and may lead to vibration of the device. To improve the stability of the damping force model, we apply the virtual coupling approach [11], an artificial spring-damper link between the motion of the haptic device in the real world and the motion of the scalpel in the virtual environment.

3. Results

Figure 2 (left) shows our experiment setup. The adaptive octree finite element model of the liver consists of 40,080 hexahedral elements, corresponding to 173,843 elements on a $82 \times 83 \times 101$ uniform grid. Applying three levels of composition ($8^3 : 1$) leads to 647 composite elements with 2,928 DOFs (see Figure 3). The surface mesh re-constructed from the hexahedral grid has 58,920 triangles. At that resolution, very fine details are visible on the cutting surface. In each simulation time step, the composite finite element simulation takes 23.3 ms. The detection of both inter- and intra-collisions takes 3.2 ms. In total, the simulation runs at 38 simulation frames per second. Using composite finite elements, as opposed to running the simulation directly on the hexahedral finite element grid, a speed-up factor of about 98.2 is obtained.

Cutting takes additional time for octree subdivision and re-creation of the coarse grid hierarchy and the render surface (12 ms), reassembly of element matrices (25 ms), as well as updating the signed distance field for collision detection (2 ms). In total, during cutting, the simulation is running at 15 simulation frames per second.

Note that the haptic rendering loop is decoupled from the visual rendering loop: the collision detection between the scalpel and the organ as well as the computation of the feedback force are performed at 1 kHz, whereas soft tissue cutting is performed at 15 Hz. In our example, the blade is represented by a set of triangles with a total of 30 vertices. The collision detection between this small set of vertices and the organ takes less than 1 ms.

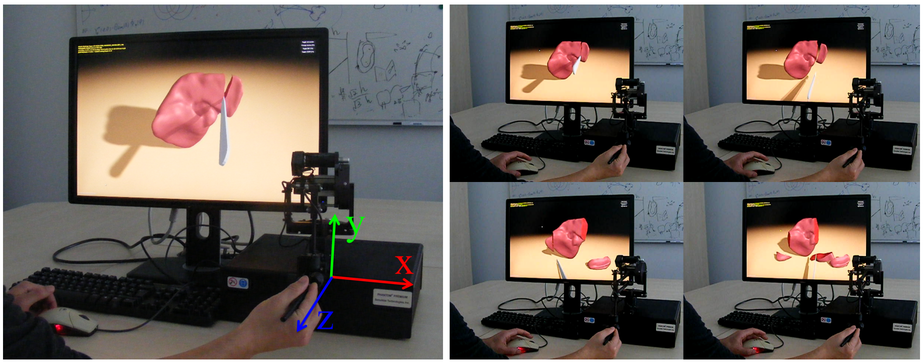


Figure 2. Left: Experiment setup. A user manipulates a haptic device that is mapped to a scalpel, in order to cut a liver model in the virtual environment. Right: A sequence of images from a live recording (also shown in the accompanying video, additionally available at http://wwwcg.in.tum.de/research/research/projects/real-time-haptic-cutting.html).

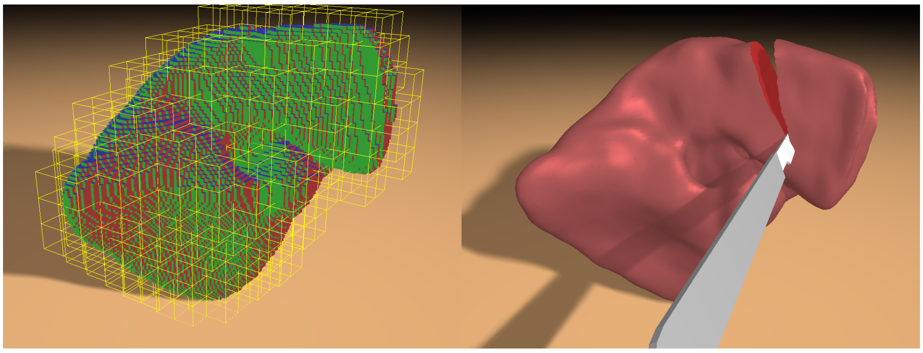


Figure 3. Left: Composite elements (yellow grid) and high-resolution hexahedral elements (color-shaded voxels). Right: Surface mesh reconstructed from the links between hexahedral elements.

For a demonstration of the quality of our haptic cutting approach, please see the accompanying video. This live recording shows cutting of a liver model by using a haptic device, which is mapped to a virtual scalpel. Figure 2 (right) shows a sequence of images from this recording. Feedback forces are depicted in Figure 4.

4. Conclusions & Discussion

We have presented a highly efficient and physically accurate real-time cutting simulation method. We have demonstrated that by using our method, cutting of soft tissues at an effective resolution of about 170,000 finite elements can be simulated at rates of 15 frames per second, while smooth and responsive haptic feedback is obtained.

In collaboration with medical experts, we are highly interested in applying this general cutting simulation tool to specific surgery planning and training scenarios.

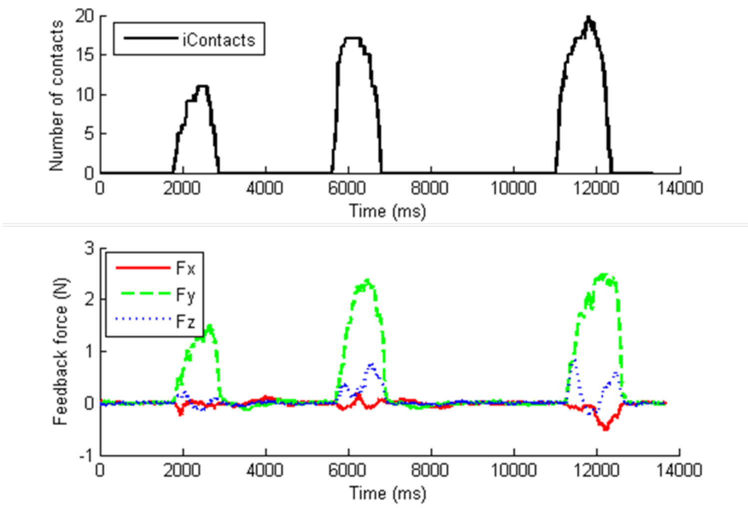


Figure 4. Top: Number of contacts between the scalpel and the liver model. Bottom: Computed feedback forces while making three cuts. (Refer to Figure 2 for the definition of the coordinate system.)

References

[1] M. Cavusoglu, T. Goktekin, and F. Tendick, Gipsi: a framework for open source/open architecture software development for organ-level surgical simulation, *IEEE Transactions on Information Technology in Biomedicine* **19** (2006), 312–322.

[2] J. Allard, S. Cotin, F. Faure, P.-J. Bensoussan, F. Poyer, C. Duriez, H. Delingette, and L. Grisoni, Sofa-an open source framework for medical simulation, *Studies in health technology and informatics* **125** (2007), 13–18. Published by IOS Press.

[3] L. Jeřábková and T. Kuhlen, Stable cutting of deformable objects in virtual environments using XFEM, *IEEE Computer Graphics & Applications* **29** (2009), 61–71.

[4] H. Courtecuisse, H. Jung, J. Allard, C. Duriez, D. Lee, and S. Cotin, GPU-based real-time soft tissue deformation with cutting and haptic feedback, *Progress in Biophysics and Molecular Biology* **103** (2010), 159–168.

[5] L. Jeřábková, G. Bousquet, S. Barbier, F. Faure, and J. Allard, Volumetric modeling and interactive cutting of deformable bodies, *Progress in Biophysics and Molecular Biology* **103** (2010), 217–224.

[6] A. Lindblad and G. Turkiyyah, A physically-based framework for real-time haptic cutting and interaction with 3d continuum models, *Proceedings of ACM symposium on Solid and Physical Modeling* (2007), 421–429.

[7] Z. Taylor, M. Cheng, and S. Ourselin, High-speed nonlinear finite element analysis for surgical simulation using graphics processing units, *IEEE Transactions on Medical Imaging* **27** (2008), 650–663.

[8] C. Dick, J. Georgii, and R. Westermann, A hexahedral multigrid approach for simulating cuts in deformable objects, *IEEE Transactions on Visualization and Computer Graphics* **17** (2011), 1663–1675.

[9] J. Wu, C. Dick, and R. Westermann, Interactive high-resolution boundary surfaces for deformable bodies with changing topology, *Proceedings of Workshop on Virtual Reality Interaction and Physical Simulation* (2011), 29–38.

[10] J. Wu, C. Dick, and R. Westermann, Efficient collision detection for composite finite element simulation of cuts in deformable bodies, *The Visual Computer* **29** (2013), 739–749.

[11] J. Wu, D. Wang, C. C. L. Wang, and Y. Zhang, Toward stable and realistic haptic interaction for tooth preparation simulation, *Journal of Computing and Information Science in Engineering* **10** (2010), 021007:1–9.