Surgical Cutting on a Multimodal Object Representation

Lenka Jeřábková and Torsten Kuhlen

Virtual Reality Group,
RWTH Aachen University, 52074 Aachen
Email: jerabkova@rz.rwth-aachen.de

Abstract. In this paper, we present the design of our surgery simulator under the aspects of multimodal object representation and parallelization on multicore architectures. Special focus is put on cutting. Surgical incisions can be accomplished interactively with force feedback.

1 Introduction

Surgical simulation is an important field of application in virtual reality (VR). A virtual surgery trainer can not only help to improve the skills of the surgeons, it also solves ethical issues related to training on animals or humans. Numerous surgical training systems have been developed in the last decade. The main requirement for a surgery simulator is the plausible deformation of the soft tissue in realtime and its interactive manipulation using a number of surgical instruments. An interactive cutting simulation is an essential feature of a surgery trainer. However, the interactive progressive cutting of a deformable object is still a challenging problem.

The majority of current surgical simulators use a polygonal approximation of the surface of the simulated object for visualization. The surface polygons are used to create a volumetric tetrahedral mesh needed by the finite elements method (FEM) in order to compute the tissue deformation. All vertices of the visualized geometry are placed at corresponding FEM mesh nodes. The interactive surgery simulation involves not only the simulation of the soft tissue deformation and its visualization, but also the processing of user interaction consisting of collision detection and response. In addition to that, force feedback is an indispensable part of user interaction. However, each of these tasks requires a different view of the data and different update rates in order to work efficiently.

An advantage of using dedicated representations of the simulated objects and the tools for each task enables the concurrent execution of the tasks. Parallelization used to be a domain of high performance computing on specialized computer architectures. However, the increase of performance of PCs due to the increase of the processor clock frequency reached its top. One of the most promising strategies for raising the performance of PCs is increasing the number of processor cores on a chip and on a board (multicore and multiprocessor architectures). However, the additional power can only be used if the application has been designed to run on a parallel architecture. Therefore, parallelization is becoming a crucial topic for all computationally intensive applications.

2 Related work and contribution

In this paper, we present the design of our surgery simulator under the aspects of multimodal object representation and parallelization. Special focus is put on cutting. The methods for surgical cutting published so far require the FEM elements to be aligned with the cut. This is achieved either by constraining the cut to the borders of existing elements [1] at the cost of creating unpleasing visual artifacts, or by splitting the elements along the cut [2], [3] or by snapping of the elements' borders to the cut [4] or by a combination of these methods [5].

All here mentioned approaches use the the polygonal boundary of the simulation mesh for visualization. As the tetrahedra are split along a cut, the visualization surface is updated to correspond to the new tetrahedra boundary. However, recent publications on interactive animation of deformable objects in computer graphics propose using a rougher resolution for the FEM simulation and a finer resolution for visualization [6], [7]. Moreover, the number and quality of the newly created FEM elements have a direct impact on the simulation performance and stability. Similarly, the newly created surface polygons influence the performance and quality of the visualization. Therefore, the main effort is to represent the cut using a small number of well shaped elements.

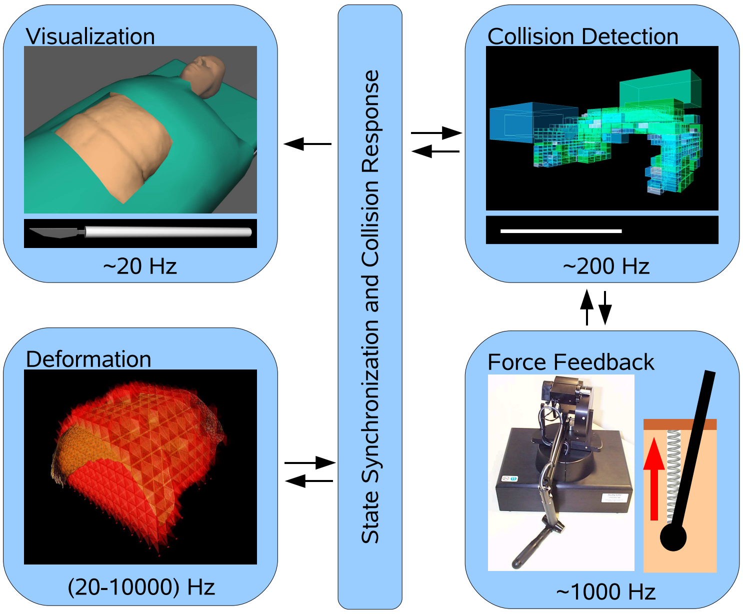
Our cutting approach is based on the discontinuous enrichment of the finite elements inspired by a method for crack modeling proposed by [8]. This method can effectively model discontinuity regions within an FEM mesh without remeshing. Our surgery simulator is based on four building blocks, each of them fulfilling a specific tasks (visualization, deformation, collision detection and force feedback) using dedicated representations of the simulated object and the surgical tools. The independence of the tasks allows their concurrent execution at task-specific update rates. In this paper, we describe the data coupling between collision detection, FEM simulation and the visual representation during an incision.

3 Methods

The structure of our surgery simulator is depicted in Figure 1. The visualization, deformation, collision detection and force feedback are four tasks run in parallel with different update rates.

- The visualization renders the polygonal approximation of the surface of the simulated object. Additionally, the surgical tools and an environment are rendered. A sufficient update rate for a smooth animation is about 20 Hz.
- The deformation process uses a mesh of tetrahedral elements approximating the volumetric object. Depending on the simulation time step, it has to be run up to several thousand times per second. The FEM deformation

Fig. 1. The structure of our surgery simulator. Each task uses a dedicated representation of the simulated object and surgical tool



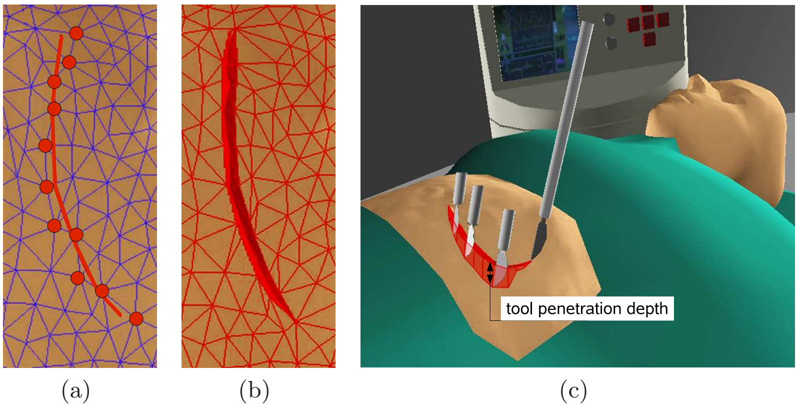
is the most computationally expensive task in this system. Therefore, it is parallelized internally.

- The collision detection uses primitives (e.g. bounding boxes) to approximate the tested surface and to quickly identify parts of the surface that the tool collides with. The tools are approximated using one or more line segments. The collision detection is processed about two hundred times per second.
- The force feedback process is usually run on a dedicated computer with a force feedback device attached to it. Here, a simplified local model of the simulated object is created. The force is proportional to the penetration depth and has to be updated about thousand times per second in order to provide smooth feedback without vibrations.

The multimodal object representation has to be kept consistent as the object undergoes deformation and topological changes. As the user perceives the system visually and haptically, it is crucial to provide the visual and haptical output at the specified rates. We synchronize the visualization, deformation and collision detection at the rate of the visualization, i.e. at about 20 Hz, so that the visualization renders the most current state of the objects. Any user interaction detected in the time between two synchronization steps is queued and processed simultaneously before the resulting changes are visualized. In the following, we will describe an interactive creation of an incision.

During a contact between the simulated object and a tool, the collision detection determines discrete points of contact and the tool penetration depth at

Fig. 2. Creating the visual representation of an incision. (a) the triangular mesh of the object surface (blue) with the desired cutting path (red line) and the vertices to be snapped to the (red points), (b) the situation with the cut completed, (c) the tool penetration depth is used to model the depth of the wound



these points. This data is used by the haptics in order to give the user the feeling of a resistant surface. Moreover, the collision data is used to generate interaction forces leading to a local deformation of the simulated object. In the FEM, the deformation field is given by an interpolation of the nodal displacements within the elements using shape funcions. For a linear tetrahedron, this corresponds to the linear interpolation using barycentric coordinates. Each vertex of the detailed geometry surface is assigned to a tetrahedron and the barycentric coordinates of the vertex are computed in the non deformed state. The barycentric coordinates remain constant during the simulation and as the FEM mesh deforms, the positions of the geometry vertices are interpolated accordingly.

When a cut has to be created, the collision data is used to create a sequence of cutting planes aligned with the movement of the tool. The intersected FEM elements are enriched by a discontinuous function defining the location of the cut. The major advantage of the element enrichment is, that the structure of the simulation mesh does not change during the simulation. Each FEM element involved in a cut stores the definition of the respective cutting plane.

The incision has to be modeled in the geometry mesh as well. This can be achieved either by subdividing the triangles intersected by the cut or by snapping the existing vertices to the cut. The first method is more suitable for simulating fractures with small material slivers. For surgery simulation we prefer the latter method as it does not generate unnecessary faces along the cut. Both methods lead to triangles, that are either completely above or completely below the discontinuity. Figure 2a shows the geometry mesh before the cut with the desired cut and the vertices, that are going to be snapped to the cut marked in red. After the vertices have been snapped to the cut, their barycentric coordinates within the FEM mesh have to be updated. Further, these vertices are doubled and the copies are assigned to the respective side of the cut in order to create an opening. The copied vertices are still on the same positions as the originals, the created hole is opened by the physical simulation (Fig. 2b). In surgery simulation, volumetric objects are simulated. Therefore, the hole in the surface has to be closed properly by modeling a wound. Once the cut path on the surface and its depth below the surface are known, the wound can be modeled by moving a copy of the surface vertices involved in the cut according to the penetration depth (Fig. 2c). New faces are created between the vertices on the surface and below the surface.

4 Results and discussion

The structure of a surgical simulator as described here takes advantage of the current development in the PC and workstations domain, utilizing multicore architectures. Dedicated representations of the simulated objects and the surgical tools are used in concurrently running tasks.

Surgical incisions can be accomplished interactively with force feedback. In the FEM mesh, the cut is modeled using discontinuous enrichment of the involved elements. As no new elements are created, the impact of an incision on performance and stability of the simulation is minimized.

The current prototype of our surgery simulator only allows for continuous cuts without branching, which has to be improved. Moreover, in order to increase realism of the surgery, different materials and internal organs have to be considered.

References

1. Cotin S, Delingette H, Ayache N. A hybrid elastic model for real-time cutting, deformations and force feedback for surgery training and simulation. The Visual Computer 2000;16(7):437–452.
2. Bielser D, Maiwald VA, Gross MH. Interactive cuts through 3-dimensional soft tissue. Procs Eurographics 1999;18(3):31–38.
3. Mor AB, Kanade T. Modifying soft tissue nodels: Progressive cutting with minimal new element creation. Procs MICCAI 2000; 598–607.
4. Nienhuys HW, van der Stappen AF. Supporting cuts and finite element deformation in interactive surgery simulation. Utrecht; 2001.
5. Steinemann D, Harders M, Gross M, Szekely G. Hybrid cutting of deformable solids. In: Procs IEEE VR; 2006. 35–42.
6. Müller M, Gross M. Interactive virtual materials. In: Procs Conference on Graphics Interface; 2004. 239–246.
7. Molino N, Bao ZH, Fedkiw R. A virtual node algorithm for changing mesh topology during simulation. ACM Transactions on Graphics 2004;23(3):385–392.
8. Belytschko T, Black T. Elastic crack growth in finite elements with minimal remeshing. Int J Numerical Methods in Engineering 1999;45(5):601–620.