A Continuous Material Cutting Model with Haptic Feedback for Medical Simulations

Maximilian Kaluschke* René Weller* Gabriel Zachmann* Mario Lorenz†‡

∗ Computer Graphics and Virtual Reality, University of Bremen
† Institute for Machine Tools and Production Processes, Chemnitz University of Technology
‡ Department of Orthopedics, Trauma and Plastic Surgery, University of Leipzig Medical Center

Abstract

We present a novel haptic rendering approach to simulate material removal in medical simulations at haptic rates. The core of our method is a new massively-parallel continuous collision detection algorithm in combination with a stable and flexible 6-DOF collision response scheme that combines penalty-based and constraint-based force computation.

Index Terms: Computing methodologies—Modeling and simulation—Simulation types and techniques—Massively parallel and high-performance simulations

1 Introduction

Medical procedures such as orthopedic surgery are highly dependent on the surgeons skill and experience. Therefore it is important for medical students to acquire a maximum of experience before they have to perform in real emergencies. However, learning these skills is usually very expensive and the physical properties of the training equipment often differs from the real procedures which limits the training success dramatically. For instance, many orthopedic procedures are simulated on simple dummies that exhibit very different physical properties and dentistry students traditionally practice tooth cutting on plastic teeth.

Medical VR simulations are able to bridge this gap. They are a good solution for preparing medical staff for difficult procedures. The most complicated type of procedures to simulate involve cutting of material, such as bone or tooth enamel. Material cutting is computationally cumbersome because a lot of data needs to be maintained and updated in real-time. Actually, haptic feedback requires the simulation to operate at a very high frequency of ideally 1 kHz or more. However, in many procedures the ability to experience and learn the *feeling* of an operation is paramount to the students ability to learn. [2] showed that practice in a virtual setting could significantly improve students motor skills compared to traditional methods.

We have developed a material cutting model that is well suited for medical simulations for a wide variety of materials and procedures, such as reaming, milling or drilling of bones that is required in hip- and knee arthroplasty and root canal opening of teeth.

2 Hybrid Force- and Torque Rendering

Our model to calculate forces and torques in the haptic rendering loop consists of a novel hybrid algorithms that combines methods from constraint- and penalty-based approaches: The linear force is calculated from the first contact point of the tool with the bone. This defines a collision-free position on the bone's surface. We attach a dampened spring of stiffness k and damping factor ζ along the displacement vector from the haptic device position p_{HIP} to the surface position p_{proxy}.

$$F = (p_{proxy} - p_{HIP}) \cdot k - v_{HIP} \cdot \zeta \tag{1}$$

The torque is calculated using a fast penalty-based approach. We represent all objects, i.e. the bone as well as the haptic tool by a polydisperse sphere packing [4]. In each frame, the sphere s that

*e-mail: mxkl@cs.uni-bremen.de

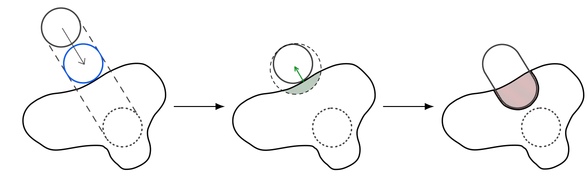

Figure 1: Overview of multi-pass GPU collision detection algorithm. Illustrations are highly exaggerated, each simulation frame only does tiny, incremental changes.

is responsible for the linear constraint at p_s with linear force F_s generates a torque

$$\tau = (p_s - p_{COM}) \times F_s \tag{2}$$

where p_{COM} is the tool's center of mass. We display the torque similarly as shown in [1] via virtual coupling.

The advantage of continuous collision detection, is that we do not need to pre-compute normals for the inside of the object, as these would eventually become outdated with enough material removal. Additionally, we avoid the *pop-through* effect with thin objects or high velocities that typically occur with pure penalty-based approaches such as VPS [3], which is often used for material cutting simulations.

Our method is especially suited for drilling, milling and reaming applications: the continuous collision detection enables us to drill along the movement path which results in a continuous drilling geometry. Most other methods evaluate the material removal at distinct states of the simulation that often results in scraps of material along the path.

2.1 Collision Detection and Material Removal

The computational bottleneck of haptic rendering is usually the collision detection. We present a novel massively-parallel continuous collision detection approach that runs completely on the GPU and supports material removal. It reuses the polydisperse sphere packing representation of the objects we have introduced in the previous section. Our approach consists mainly of three basic steps: first, finding the first point of contact between the tool and the bone, second, computing a collision normal for this point, and third, removal of material on the path of the tool. We will briefly describe these individual steps:

Pass 1: Surface Contact In order to determine the contact point on the bone's surface we first perform continuous collision detection. We iterate over each individual sphere of the tool and check for an intersection along the path from the last free state and the device's current pose.

The contact distance $d_{t,b}$ for a tool sphere t and bone sphere b along the linear movement m of the tool sphere t is given by:

$$d_{t,b} = \frac{2(\Delta_c \cdot m) - \sqrt{4(m\Delta_c)^2 - 4m^2(\Delta_c^2 - (r_t + r_e)^2)}}{2m^2} \tag{3}$$

where r_t, r_e are the spheres' radii and Δ_c connects the spheres' centers. A collision occurs in case of $4(m\Delta_c)^2 - 4m^2(\Delta_c^2 - (r_t +$

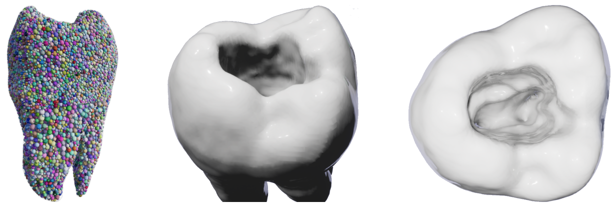

Figure 2: Drill-able tooth representation. *Left*: Sphere packing of the tooth with 36 k spheres. *Mid*: Mesh rendering after removing material. *Right*: Same modified mesh viewed from the top.

$r_e)^2) \geq 0$. The global minimum of all positive distances $d_{b,t}$ delivers the contact point p_c on m.

Pass 2: Normal Estimation In order to generate continuous contact normals we do not simply rely on a single contact point, but temporarily enlarge the contact sphere according to it's radius.

For all tool spheres t_i within this radius around the contact sphere we search all overlapping bone spheres b_j to determine the contribution to the normal, weighted by the intersection volume V_{t_i,b_j} and the bone sphere's density ρ_{b_j}:

$$n = \sum_{t_i} \sum_{b_j} \frac{c_{t_i} - c_{b_j}}{|c_{t_i} - c_{b_j}|} \cdot V_{t_i,b_j} \cdot \rho_{b_j} \tag{4}$$

$$\text{where } V_{t_i,b_j} = \frac{\pi(r_{t_i} + r_{b_j} - d)(d^2 + 2d(r_{t_i} + r_{b_j}) - 3(r_{b_j} - r_{t_i}))}{12 \cdot d}$$

c_s, r_s are the center and radius of sphere s, and $d = |c_{t_i} - c_{b_j}|$.

Pass 3: Material Removal Let p_e be the end position of the contact sphere assuming that it was not stopped at the bone's surface. This defines a capsule along the movement direction m. Inside this capsule we remove all bone material. To do that, we compute for each colliding bone sphere the shortest distance d_c from the spheres' centers to the capsules center line and shrink accordingly:

$$\forall b_j : c_{b_j} := c_{b_j} - \frac{d_c}{2|d_c|},\; r_{b_j} := r_{b_j} - \frac{|d_c|}{2} \tag{5}$$

3 Applications

We have implemented our method as a plugin that is supports two major game engines, Unreal and Unity. The plugin runs asynchronously to the game engines to avoid the frame limit of the main rendering thread.

We have developed two example applications. First, we developed a training simulator for dental students. The students can cut a virtual tooth to create an opening to the pulp (see Fig. 2), which is an important step in a root canal procedure and a major factor in the stability of the tooth post operation. In case of very thin walls, the tooth is at risk of breaking. In this scenario we use a Phantom Omni device for the haptic feedback.

Second, we have implemented a hip replacement simulator for the training of orthopedic surgeons (see Fig. 3). The surgical students prepare the hip-socket by removing old cartilage and bone to fit a prosthetic hip joint inside. This step in the procedure is crucial because a bad fit would risk future dislocation leading to future complications for the patient.

Actually, visibility of the hip in a real operation is extremely bad or non-existent during the milling process. Therefore the surgeon has to operate just by feeling, making haptics the ideal technology for this application. Another challenge in this application and the reason we use an industrial robot as haptic device is the demand for high forces (up to 200 N) that occur in the real operation.

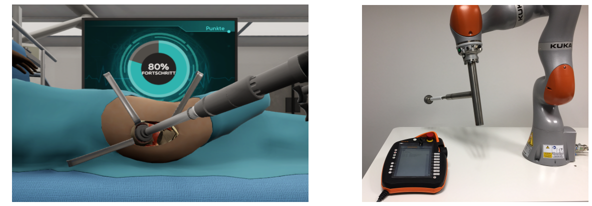

Figure 3: Hip surgery simulator with haptic feedback. *Left*: In-game screenshot. *Right*: KUKA robot that renders the haptic feedback.

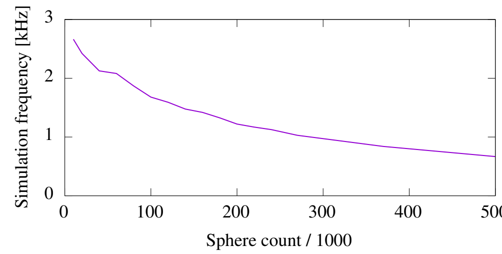

Figure 4: Performance graph for a recording of realistic drilling interaction with varying degrees of packing accuracy.

4 Results

In order to test the performance of our algorithm, including the collision detection, force and torque computation and the material removal, we recorded a typical drilling interaction from the dental simulator using a Phantom Omni device (Fig. 2).

We ran the simulation on a PC with i7-4770K CPU, 16 GB DDR3 RAM and GTX 1080 Ti GPU. Obviously, the accuracy depends on the density of the sphere packing. Consequently, we tested several different sphere packings. Figure 4 shows the preliminary results with a not yet finally optimized implementation. The $x-$axis shows the number of spheres with respect to the simulation frequency on the $y-$axis. Our experiment shows that we achieved haptic rates for up to 300 k spheres. Actually, a preliminary user test suggests that about 10 k spheres are sufficient to accurately model a human tooth.

5 Conclusion

We have presented a novel for haptics-enabled medical simulation with support for material removal. Our material removal model is, to the best of our knowledge, the first that supports continuous feedback for arbitrary tool models. Our method can easily be parallelized and runs completely on the GPU. Our results show that we achieve haptic rates even for very detailed models.

In the future we plan to further optimize our GPU implementation and to investigate the accuracy with respect to the sphere packing by extensive user tests in different medical simulation. Another avenue for future work is the support for deformable tools and materials.

References

[1] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. In *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, volume 3, pages 140–145 vol.3, Aug 1995.

[2] J. S. Henn, G. M. Lemole, M. A. T. Ferreira, L. F. Gonzalez, M. Schornak, M. C. Preul, and R. F. Spetzler. Interactive stereoscopic virtual reality: a new tool for neurosurgical education. *Journal of Neurosurgery*, 96(1):144 – 149, 2002.

[3] W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 401–408, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[4] R. Weller and G. Zachmann. Inner sphere trees and their application to collision detection. pages 181–201, 01 2008.