GPU-based Penetration Depth Computation for Haptic Texture Rendering

Department of Computer Science University of North Carolina at Chapel Hill November 2004

Figure 1: 6-DoF haptic texture rendering of the interaction between a hammer and a CAD part.

Highlights

- First 6-DoF haptic texture rendering algorithm
- Objects are described as simplified polygonal representations with texture images that store fine geometric detail
- Novel perceptually inspired force model that captures the interaction between textured surfaces
- Image-based algorithm for computing penetration depth
- Fast, parallel computation of penetration depth on GPU

The Challenge

Six-Degree-of-Freedom (6-DoF) haptic rendering refers to the computation and display of contact forces between interacting objects. Surface texture is one of the main haptic cues, but previous 6-DoF haptic rendering methods could not capture texture effects, due to sampling limitations. Exact collision detection methods cannot handle textured objects with hundreds of thousands of polygons at the force update rates required by haptic rendering (hundreds of Hz).

Haptic Texture Rendering

In a first step, we perform object-space collision detection between simplified polygonal meshes. We identify intersecting patches and a penetration direction at each contact.

In a second step, we refine the directional penetration depth (PD) at each contact using fine geometric detail stored in texture images. We have designed an image-space algorithm and developed a GPU-based implementation that perform fast computation at force update rates.

We compute per-contact forces and torques using a novel force model for texture rendering, inspired by perceptual studies. Finally, the net force and torque are rendered to the user.

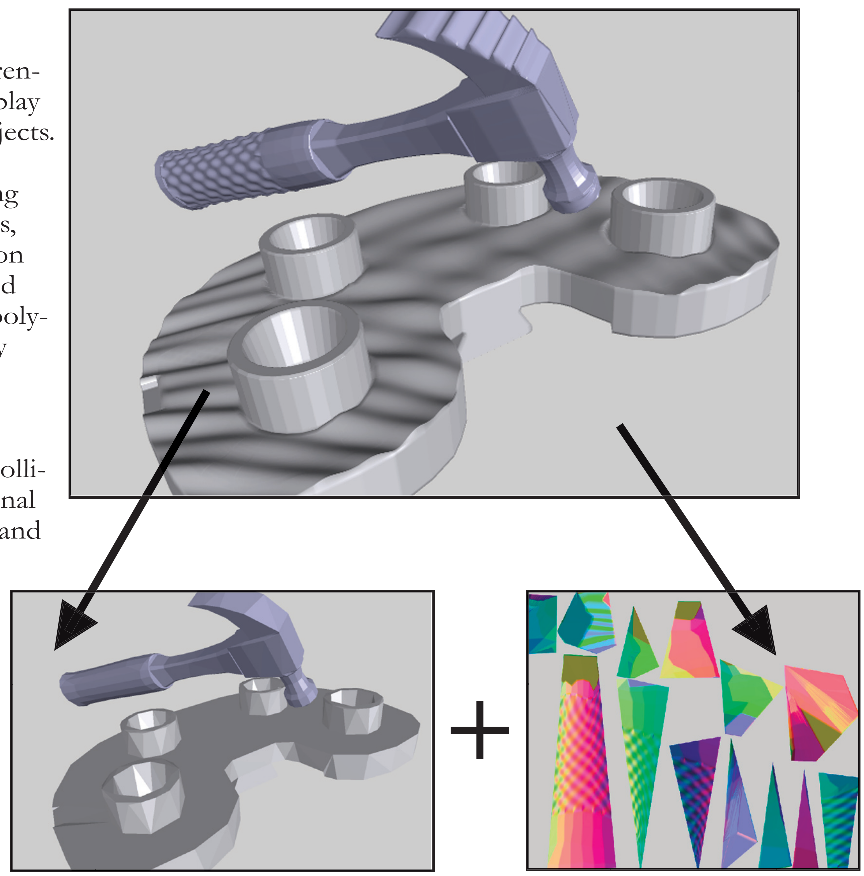

Figure 2: Object representation: simplified polygonal meshes with texture images that store fine geometric detail.

PD Computation on GPU

We have developed an image-based algorithm for computing directional penetration depth (PD) between textured polygonal models that is well suited for implementation on GPUs. We assume that, in the regions of contact, the surfaces can be described as height fields. In such cases, the directional penetration depth can be defined as the maximum height field difference between the intersecting patches.

As a preprocessing step, we parameterize the low-resolution surfaces used in collision detection, and create texture images that store the position of the full-resolution textured surfaces.

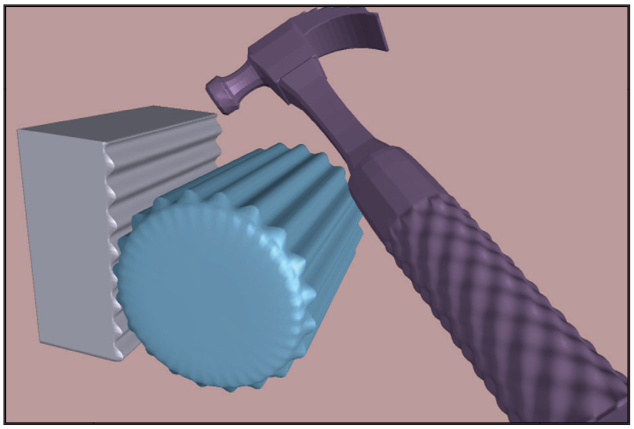

Figure 3: Hammer and blocks. We achieve haptic rendering rates of 500Hz between the blocks.

At runtime, we render the intersecting low-resolution surface patches through an orthographic projection along the penetration direction. At each fragment, we can obtain the original surface position values by looking up the texture images. We then perform a second pass over the intersecting region, where we subtract the height fields of both surfaces, and the result is copied to the depth buffer. Finally, we obtain the maximum penetration depth by performing a binary search on the depth buffer through occlusion queries, thus avoiding expensive buffer read-backs to the CPU.

Our texture force rendering algorithm requires the computation of penetration depth and its gradient at each contact location. In practice, this involves computing multiple instances of the penetration depth. We take advantage of parallelization and tiling to accelerate the GPU computations.

Results

We have tested our new haptic rendering algorithm on polygonal models of high complexity (hundreds of thousands of triangles) with rich surface texture. The input models are simplified down to a few hundreds of triangles, which is roughly the size that can be handled by exact haptic rendering techniques. Using texture images, we compute the penetration depth that captures the original high-frequency geometry, and we render texture effects that would otherwise be missed.

The GPU-based implementation of penetration depth computation enables haptic rendering rates of 500Hz on benchmarks such as the blocks and gears shown in Fig. 3, and update rates between 100 and 200Hz for highly challenging contact scenarios like the hammer and CAD part shown in Fig. 3 or the file and CAD part in Fig. 4. We have used a dual Pentium-4 2.4GHz processor PC and an NVidia GeForce FX5950 graphics card. The rendering time is linear in the number of contacts. With the performance growth curve of GPUs exceeding Moore's Law, we expect to achieve rendering rates of 1kHz in a year or two without further optimizations.

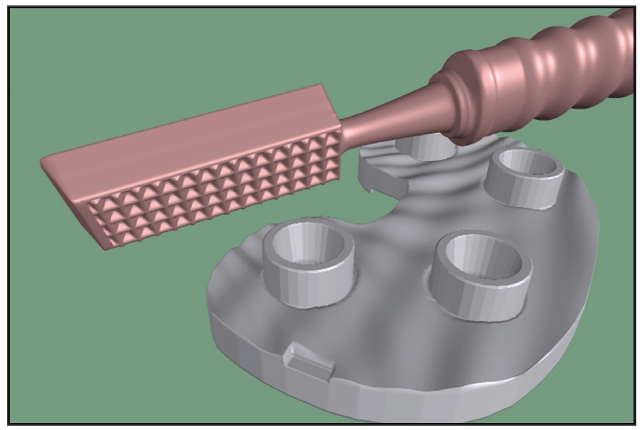

Figure 4: File and CAD part. In this complex scenario we achieve haptic rendering rates of 100Hz-200Hz.

Project Leader

Ming Lin, professor

Graduate Research Assistants

Miguel A. Otaduy, Nitin Jain, Avneesh Sud

Research Sponsors

Intel Corporation
National Science Foundation
Office of Naval Research
U.S. Army Research Office
Defense Advanced Research Projects Agency

Selected Publications

Otaduy, M.A., N. Jain, A. Sud, and M.C. Lin. Haptic Display of Interaction between Textured Models. *Proc. of IEEE Visualization Conference*, 2004.

Otaduy, M.A., and M. Lin. A Perceptually-Inspired Force Model for Haptic Texture Rendering. *Proc. of Symposium on Applied Perception in Graphics and Visualization*, 2004.

Key Words

Haptics, Texture, HCI, Image-based Algorithms

For More Information

http://gamma.cs.unc.edu/HTextures

http://gamma.cs.unc.edu/HTextures