Editors: Lawrence Rosenblum and
Michael Macedonia

Sensation-Preserving Haptic Rendering

Ming C. Lin

University of North Carolina at Chapel Hill

Miguel A. Otaduy

Swiss Federal Institute of Technology (ETH) Zurich

Haptic, from the Greek word *haptesthai*, means relating to or based on the sense of touch. Haptic is to touching as visual is to seeing and auditory is to hearing. The sense of touch is one of the most important sensory channels. The human tactile system provides a unique and bidirectional communication between humans and their physical environment. We can divide the tactile system into cutaneous, kinesthetic, and haptic systems, based on the underlying neural inputs. The cutaneous system employs receptors embedded in the skin, while the kinesthetic system uses receptors located in muscles, tendons, and joints. The haptic sensory system uses cutaneous and kinesthetic receptors, but it differs from the other systems because it's associated with an active exploration of the surroundings.

To date, most human–computer interactive systems focus primarily on the graphical rendering of visual information and, to a lesser extent, on the display of auditory information. Haptic interfaces have the potential to increase the quality of human–computer interaction by accommodating the sense of touch. They provide an attractive augmentation to visual display and enhance the level of understanding of complex data sets. In addition, researchers have effectively used haptic interfaces for a number of applications including molecular docking, manipulation of nanomaterials, surgical training, virtual prototyping, and digital sculpting.

Haptic rendering, or force display of interaction with virtual objects, is poorly understood and practically unknown to the general public. Haptic display is often rendered through what is essentially a small robot arm. Such devices are now commercially available for a variety of configurations. A haptic rendering system generates contact or restoring forces to prevent penetration into the virtual objects and create a sense of touch. The system computes contact forces by first detecting if a collision or penetration has occurred. Then, the system determines the (projected) contact points on the model surface. Finally, it computes restoring forces based on the amount of penetration.

Despite the huge body of literature in collision detection and contact handling, the existing accurate algorithms cannot run at the desired update rates (at least hundreds of hertz but preferably several kilohertz) for haptic rendering of complex models. This problem results from the fact that the performance of any collision detection algorithm intrinsically depends on both the number of geometric primitives (such as polygons) and the configuration of objects when they collide. While we can render millions of polygons at interactive rates, we can barely create a force display of an environment consisting of just tens of thousands of polygons at the force update frequency.

Researchers have recently investigated the problem of rendering the contact forces and torques between 3D virtual objects. This problem is known as six-degrees-of-freedom (6-DOF) haptic rendering, as the computed output includes both 3-DOF forces and 3-DOF torques. This article presents an overview of our work in this area. We suggest different approximation methods based on the principle of preserving the dominant perceptual factors in haptic exploration. References to related research in this article are available at http://gamma.cs.unc.edu/CGA05/ and elsewhere.[1-3]

Haptic perception

Psychophysics of touch has laid the foundation for many fundamental results in haptic rendering. Our work is also inspired by findings in haptic perception.

Klatzky and Lederman conducted and documented studies on identifying objects using *haptic glance*, a brief haptic exposure that placed several temporal and spatial constraints on stimulus processing.[1] They showed that a larger contact surface area helped identify textures or patterns. However, they found it was better to have a stimulus of the size comparable or just slightly smaller than that of the contact area when exploring geometric surface features. Okamura and Cutkosky defined a fine (geometric) surface feature based on the ratio of its curvature to the radius of the fingertip acquiring the surface data.[1]

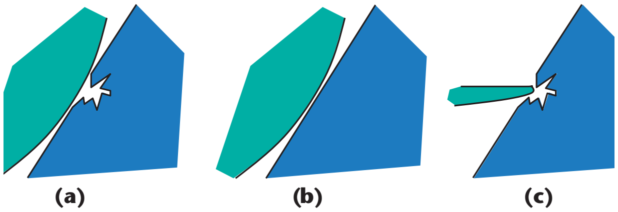
We draw the following key observation from these different studies relevant to 6-DOF haptic rendering: Human haptic perception of the geometric surface features depends on the ratio between the contact area and the size of the feature, not the absolute size of the feature itself.

As Figure 1 illustrates, Okamura and Cutkosky's observation for tactile exploration can extend to haptic rendering of contact forces between rigid bodies.[1,2] The resolution at which the models are represented affects the number of contact points used to describe object interaction. However, increasing the resolution beyond a sufficiently large value does not affect the computed net force much, as Figures 1a and 1b show. In contrast,

Published by the IEEE Computer Society

Figure 1c shows that a small contact area requires a high-resolution model.

We propose a new model of acceptable haptic error metrics that differs notably from that of human visual perception. In visual rendering, a combination of surface deviation (also commonly measured using the Hausdorff distance) and the viewing distance from the object determines whether the simplified object representations require higher resolution. In haptic rendering, on the other hand, the error metric is governed by the relationship among the surface deviation, the resolution of the simplified model, and the contact surface area.



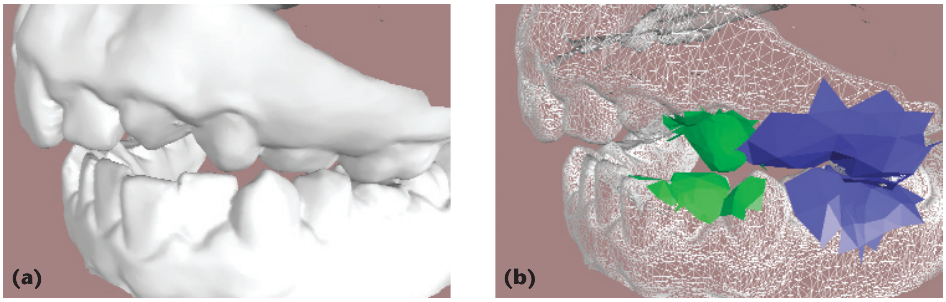
1 Contact area and resolution: (a) high-resolution model with large contact area, (b) low-resolution model with large contact area, and (c) high-resolution model with small contact area.[1] (©2003 ACM Press. Reprinted with permission.)

Sensation-preserving simplification

So how does the new haptic error metric help in achieving interactive haptic rendering of complex environments? We introduce the notion of sensation-preserving simplification to accelerate collision queries between two complex 3D polyhedral models in haptic rendering. Given a polyhedral representation of an object, we compute a series of approximations at different resolutions and a bounding volume hierarchy for efficient collision detection. We combine the multiresolution representation and the bounding volume hierarchy into one dual-hierarchical data structure that enables time-critical contact force computation in haptic rendering.

At runtime, the hierarchy is locally refined based on the haptic error metric, preserving perceivable contact details and making the simplified models feel almost the same as the original ones. The multiresolution hierarchy enables selection of the appropriate geometric resolution based on the contact configuration. As Figure 2 shows, we can select different approximation models at each contact location, depending on the ratio between feature size and contact area.

By using simplified models of drastically lower geometric complexity (for example, fewer polygons) we can significantly reduce the overall computational cost of contact queries, thereby achieving the desired force update rates otherwise not possible. However, high-resolution, fine geometric features, which are removed by our haptic error metric, produce relevant tactile sensations under sliding motion. We need to find a way to compensate for these missing surface texture effects.

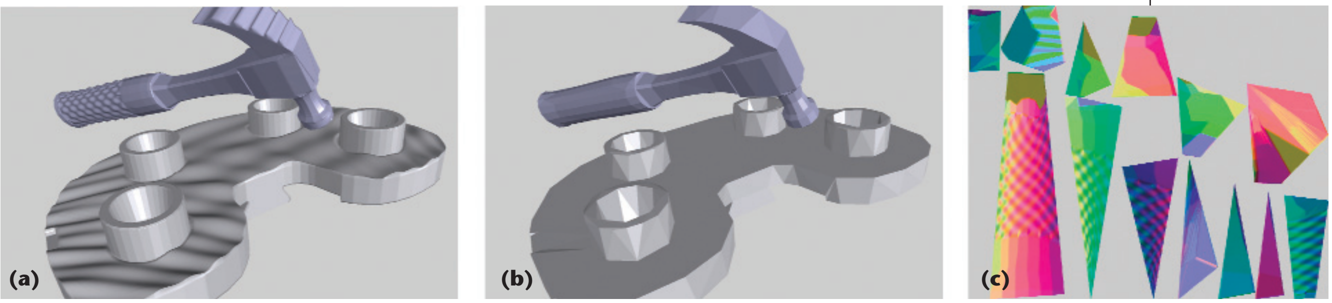


2 Adaptive resolution selection. (a) Moving jaws in contact, rendered at their highest resolution. (b) The appropriate resolution (shown in blue and green) is selected adaptively for each contact location, while the finest resolution is displayed in wireframe.[1] (©2003 ACM Press. Reprinted with permission.)

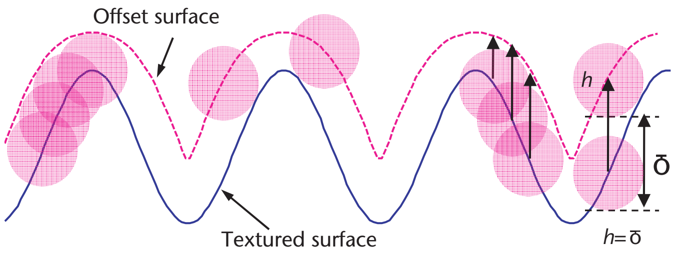
Haptic textures

Geometric surface texture is among the most salient haptic characteristics of objects. It can be a compelling cue to object identity and can strongly influence forces during manipulation.

In computer graphics, texture mapping is commonly used in real-time rendering and computer games. Objects with high polygon counts are described by coarse representations (for example, just a few flat-shaded polygons) with texture images that replace the original fine geometric detail. We adopt a similar principle for haptic rendering and refer to the fine geometric detail stored in texture images as haptic textures. Figure 3a shows highly complex objects based on the simpli-



3 Haptic display of interaction between textured models: (a) high-resolution textured hammer (433,000 polygons) and CAD part (658,000 polygons), (b) low-resolution models (518 and 720 polygons), and (c) hammer texture with fine geometric detail.[2] (©2004 IEEE.)



4 **Offset surface computed as the convolution of a surface with a sphere (left). Sphere whose trajectory traces an offset surface (center). Correspondence between penetration depth, δ, and height of the offset surface, *h* (right).[2] (©2004 IEEE.)**

fied models of Figure 3b along with the haptic textures of Figure 3c.

Most of the existing haptic texture-rendering algorithms focus primarily on rendering the interaction between the point tip of a haptic probe and a textured surface. Prior to our work, no technique has displayed interaction forces and torques between two textured models. So, one of our biggest challenges was to design a new force model for 6-DOF haptic texture rendering.

6-DOF haptic texture rendering

Through a series of experiments, Klatzky and Lederman observed that when perceived through a rigid, spherical probe, roughness initially increases as texture spacing increases, but after reaching a maximum value, it decreases again. Their studies showed that the perception of roughness is intimately related to the trajectory traced by the probe. They identified the radius of the probe, the normal forces, and the speed of exploration as some of the factors affecting the probe trajectory for a given texture spacing.

For a spherical probe, and in the absence of dynamic effects, the surface traced by the probe during exploration constitutes an offset surface (see Figure 4). The oscillation of the offset surface produces the vibratory motion that encodes roughness.

How can we generalize the concept of offset surface to the interaction between two arbitrary textured surfaces? Consider the case, shown in Figure 4, of a spherical probe whose center moves along a textured surface. In this situation, the spherical probe penetrates the textured surface. The vertical penetration depth δ is the vertical translation required to separate the probe from the textured surface. Vertical penetration depth equals the height of the offset surface.

We can generalize the concept of directional penetration depth to the interaction between arbitrary surfaces. Using the contact normal between low-resolution approximations of the surfaces, we can define a directional penetration depth between the full-resolution surfaces. We can then interpret the gradient of the penetration depth as texture-induced motion. Minsky and others have already verified the validity of the penetration depth's gradient as a descriptor for texture-induced motion for a point haptic probe interacting with a texture surface.[2]

Generalizing the observations of earlier work, we developed a novel force model for 6-DOF haptic texture rendering based on the gradient of directional penetration depth that accounts for important factors identified by psychophysics studies. But, the computational bottleneck remains, and we need a way to compute the penetration depth and its gradient fast enough to perform 6-DOF haptic texture rendering.

GPU-accelerated computation

For years, the performance and functionality of graphics processing units (GPUs) have increased at a faster pace than Moore's law. Fortunately, today's GPUs can potentially serve as coprocessors for general-purpose computations. Recently, researchers have proposed many new algorithms and applications that exploit the inherent parallelism and vector processing capabilities of GPUs.

We designed a penetration-depth estimation algorithm between textured polygonal models by taking advantage of the GPUs' computational power. We assume that, in the regions of contact, the surfaces can be described as height fields. In such cases, we can define the directional penetration depth as the maximum height difference between the intersecting patches. As a preprocessing step, we parameterize the low-resolution surfaces used in collision detection and create haptic textures that store the position of the full-resolution textured surfaces.

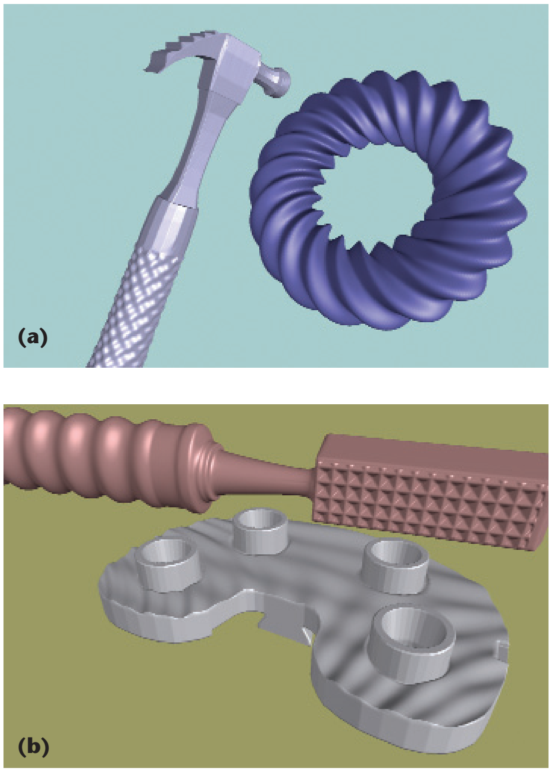
At runtime, we render the intersecting low-resolution surface patches through an orthographic projection along the penetration direction. At each pixel, we can obtain the original surface position by looking up the haptic textures. We then subtract the heights of both surfaces over the intersection region and find the maximum value, which corresponds to the directional penetration depth. We obtain the maximum by performing a binary search on the depth buffer using occlusion queries, thus avoiding expensive buffer readbacks from the GPU to the CPU. Our texture-force rendering algorithm requires the computation of penetration depth and its gradient at each contact location. In practice, this step involves computing the penetration depth at multiple configurations and applying divided differences. We take advantage of parallelization and tiling to accelerate the GPU computations. Please see Otaduy et al.[2] for more details.

Putting it all together

Our haptic rendering algorithm starts by first performing object–space collision detection between low-resolution polygonal meshes, computed by our sensation-preserving simplification method. We identify intersecting surface patches and a penetration direction at each contact. We refine the directional penetration depth at each contact using haptic textures and the image–space algorithm implemented on the GPU, thus performing fast computations at haptic rates.

We compute per-contact force and torque using our novel texture force model and, finally, the net force and torque that preserve the pertinent tactile sensation are displayed to the user.

Elsewhere, we present a detailed description and implementation of the entire haptic rendering pipeline using implicit integration (to achieve greater stability).[3] We tested our new sensation-preserving haptic rendering algorithm on polygonal models consisting of hundreds of thousands of polygons with rich surface texture, as Figure 5 shows.



5 **Benchmark models: (a) hammer and torus and (b) file and CAD part.[2] (©2004 IEEE.)**

The results in this article are only a few examples of what we can achieve using sensation-preserving haptic rendering. We're in the process of studying how to extend our approach to deformable models and a broader class of geometric representations (such as point-based, voxel-based, high-order primitives). We're also investigating how we can apply our approach to medical training, rapid prototyping, scientific visualization, and other areas. We believe that sensation-preserving haptic rendering has promising potential in providing a richer 3D interface for a wide range of applications. ■

Acknowledgments

We thank Naga Govindaraju, Nitin Jain, and Avneesh Sud for their help on the GPU implementation and the University of North Carolina's Geometric Algorithms for Modeling, Motion, and Animation (GAMMA) Group for helpful discussions about this work. This research is supported in part by the US Army Research Office, Intel, National Science Foundation, Office of Naval Research, and fellowships from the government of Basque Country and UNC Chapel Hill.

References

1. M.A. Otaduy and M.C. Lin, "Sensation Preserving Simplification for Haptic Rendering," *ACM Trans. Graphics* (Proc. Siggraph), vol. 23, no. 3, 2003, pp. 543-553.
2. M.A. Otaduy et al., "Haptic Display of Interaction between Textured Models," *Proc. IEEE Visualization Conf.*, IEEE CS Press, 2004, pp. 297-304.
3. M.A. Otaduy and M.C. Lin, "Stable and Responsive Six-Degree-of-Freedom Haptic Manipulation Using Implicit Integration," *Proc. IEEE World Haptics Conf.*, IEEE Press, 2005, pp. 247-256.

Readers may contact the authors at lin@cs.unc.edu or otaduy@inf.ethz.ch.

Readers may contact the department editors at lrosenbl@nsf.gov or michael_macedonia@peostri.army.mil.