Interactive Haptic Rendering of High-Resolution Deformable Objects

Nico Galoppo[1], Serhat Tekin[1], Miguel A. Otaduy[2], Markus Gross[2],
and Ming C. Lin[1]

[1]Department of Computer Science, University of North Carolina, Chapel Hill,
North Carolina, 27599-3175, U.S.A.

[2]Computer Graphics Laboratory, ETH Zentrum, Haldeneggsteig 4,
CH - 8092 Zurich, Switzerland

{nico, tekin, lin}@cs.unc.edu
{otaduy, grossm}@inf.ethz.ch

Abstract. We present an efficient algorithm for haptic rendering of deformable bodies with highly detailed surface geometry using a fast contact handling algorithm. We exploit a layered deformable representation to augment the physically based deformation simulation with efficient collision detection, contact handling and interactive haptic force feedback.

Keywords: Haptic Display, Collision Detection, Deformable Models.

1 Introduction

Haptic rendering of forces and torques between interacting objects, also known as 6 degree-of-freedom (DoF) haptics [11], has been demonstrated to improve task performance in applications such as molecular docking, nano-manipulation, medical training, and mechanical assembly in virtual prototyping. Haptic display of complex interaction between two deformable models is considered especially challenging, due to the computational complexity involved in computing contact response and performing proximity queries. Such queries include collision detection, separation distance, and penetration depth, between two deformable models at force update rates [11] and [14].

Dynamic contact scenarios arise when objects with rich surface geometry are rubbed against each other while they bounce, roll or slide through the scene. They are computationally costly to simulate in real time and particularly difficult to perform with 6-DoF haptic manipulation. We present an efficient algorithm for 6-DoF haptic rendering of deformable bodies with high-resolution surface geometry using approximate but stable contact forces to achieve interactive haptic force feedback rendering. We propose a two-level layered model representation for deformable objects in the scene to augment a fast physically based deformable model simulator with efficient collision detection, contact handling and stable force feedback rendering.

A fast collision detection module is essential to compute contact force at sufficiently high rates that enable haptic rendering. Our method exploits the layered model representation in a fast image-based collision detection algorithm using parallelized operations on graphics hardware, such that the actual cost depends on the size of the contact area, not directly on the resolution of the deformable mesh.

We have developed novel and efficient solutions for contact force computation that interacts with haptic force feedback at haptic update rates in a stable manner using virtual coupling. We demonstrate our haptic rendering algorithm on various scenarios, such as those shown in Fig. 1.

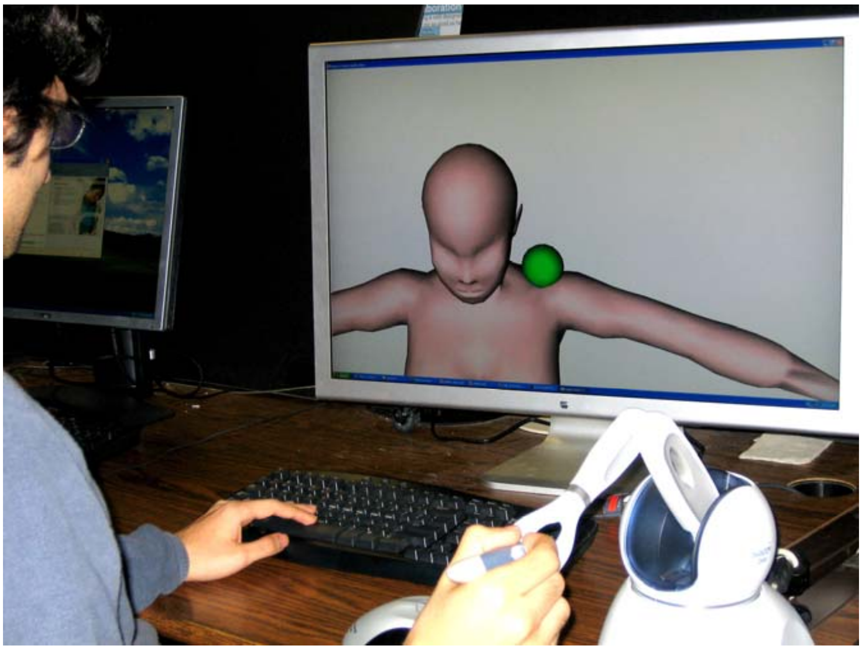

Figure 1. A user interacting with a soft body using a haptic device.

The rest of the paper is structured as follows. Section 2 discusses the related work on dynamic simulation and haptic rendering of deformable models. We introduce our novel representation and give an overview of our approach in Section 3. Section 4 presents our proximity query algorithm and Section 5 describes our penalty-based method for rendering collision response. Finally, we show results in Section 6 and conclude with possible future research directions in Section 7.

2 Related Work

Physically-based simulation of deformable bodies has been widely studied in computer graphics and haptic rendering during the last two decades [17]. We focus our discussion here on work directly related to our approach.

Continuum mechanics methods are commonly used to simulate deformable objects and some of these methods have been used for haptic rendering or other interactive applications. Some of the examples include linear FEM with matrix condensation [1], [5], and [6], corotational linear FEM for stable large deformations [12], the boundary element method [9], or quasi nonlinear elasticity with precomputation of response functions [4]. For contact handling, penalty methods are easy to implement but rely on existence of interpenetration and suffer from loss of passivity, although recent local models alleviate the latter problem [10]. Constraint-based methods, which have a higher computational cost but handle nonpenetration accurately, have been applied in a variety of formulations. Cotin et al. [4] applied equality position constraints at contacts, and solved for contact forces using Lagrange multipliers. Duriez et al. [5] and [6] eliminated sticking problems by adopting Signorini's contact model, and formulating a linear complementarity problem (LCP). This approach proves to model contact effectively for deformable bodies, but it is not directly applicable to rigid bodies, because the system may be over-constrained.

3 Overview

In this section, we first introduce the representation we use in this work, and then give an overview of our haptic rendering pipeline.

3.1. Representations and Key Concepts

Our haptic rendering algorithm exploits the following key concepts:

- A two-level layered model representation for deformable objects, illustrated in Figure 2. The low-resolution proxies are used to accelerate collision detection and haptic force rendering, whereas a high-resolution tetrahedral mesh is used to achieve highly detailed deformations. See Fig. 2 for an example.

- Using a two-stage collision detection algorithm for layered deformable models, our proximity queries are scalable and output-sensitive, i.e. the performance of the queries does not directly depend on the complexity of the surface meshes but rather on the area of the contact patch.

- Using penalty-based collision response, implicit integration of motion equations and virtual coupling of contact forces to the haptic tool, we provide fast and responsive contact handling, alleviating the time-step restrictions of previous impulsive methods.

We simulate the deformable material with a fast rotationally invariant FEM simulator with implicit integration to guarantee stability [12]. Our mathematical formulation of dynamic simulation and contact processing, along with the use of dynamic deformation textures, is especially well suited for realization on commodity SIMD or parallel architectures, such as graphics processing units (GPU), Cell processors, and physics processing units (PPU).

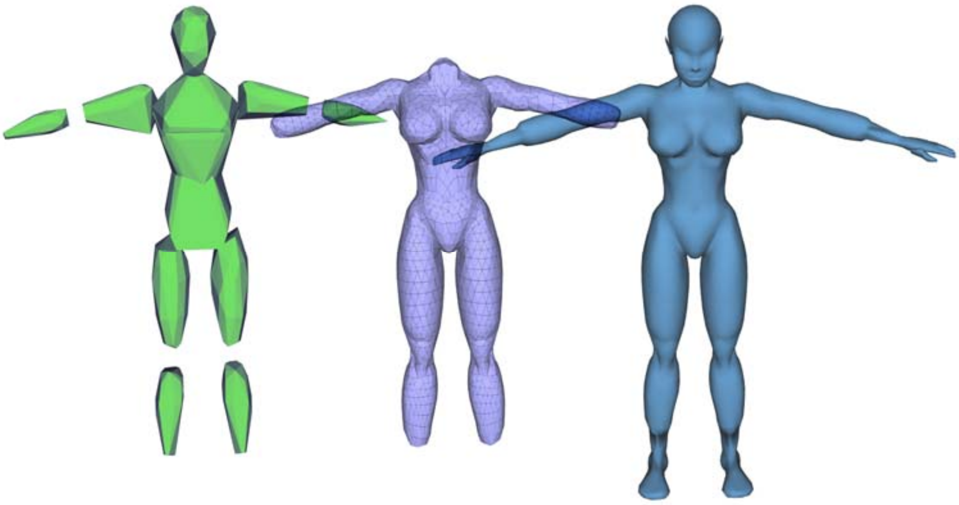

Figure 2. Our layered representation of a human model. In green on the left are low-resolution proxies used for collision detection and haptic interaction; in the middle the deformable tetrahedral mesh; on the right, the highly detailed surface mesh with deformable skin.

3.2. Haptic Rendering Pipeline

We use the haptic rendering pipeline and the multi-rate architecture similar to the one presented by Otaduy and Lin [15], along with virtual coupling introduced by Colgate et al. [2] and [3]. Fig. 3 shows the overall haptic rendering pipeline of our system.

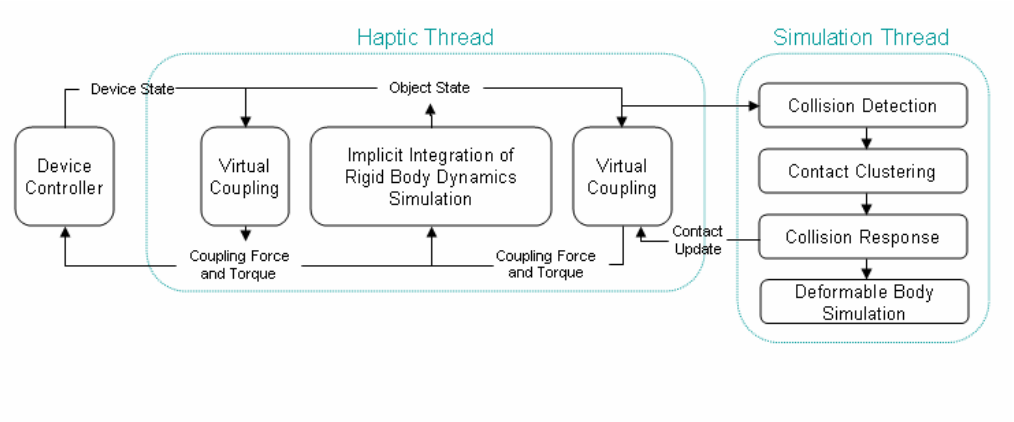

Figure 3. The overall haptic rendering pipeline of our system.

4 Accelerated Collision Queries

We propose a fast image-based algorithm that exploits our representation of soft characters. Our accelerated collision queries are performed in three steps:

1. Identify potentially colliding contact patches using low-resolution proxies.
2. Compute localized penetration depth fields.
3. Collect high-resolution skin surface collisions and directional penetration depths.

The last two steps are performed using image-space techniques with the aid of graphics hardware, achieving fast collision detection results at haptic update rates. Our method shares the two-step approach of others used for rigid bodies [13] or rigid bodies with a deformable surface layer [8]. But unlike these methods, our collision query algorithm also performs hierarchical pruning to eliminate large portions of the objects from collision queries, by exploiting the skeletal nature of the models. The worst-case cost of our collision detection is $O(n)$ for a pair of tested objects with n surface nodes; the actual cost depends on the size of the contact area.

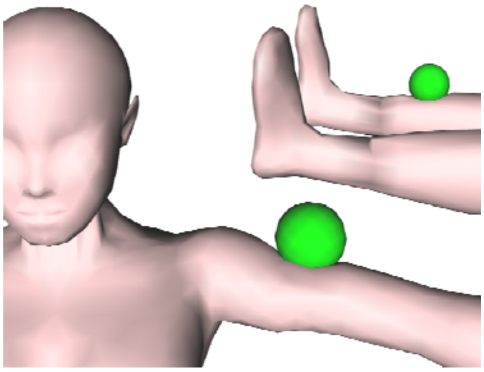

Figure 4. Different collision scenarios with contact response on the human skin model. The contact areas are associated with the low-resolution detected areas in Figure 5.

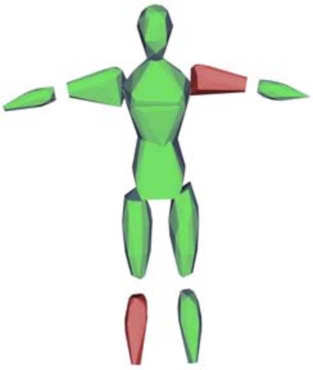

Figure 5. Collision pruning. Only the area around the bone proxies colored in red are considered for high-resolution collision detection.

Our accelerated proximity query algorithm starts by performing object-space collision detection between low-resolution polygonal proxies. In a pre-process, we first perform an approximate convex decomposition of the input models, creating a set of coarse, convex proxies with a few tens of vertices each, as shown in Figure 3. The union of all proxies represents an approximate bounding volume of the model, where each deformable skin vertex is associated with at least one proxy. At runtime we use the maximum skin displacement of all vertices to compute a conservative bound of the proxy. We identify potentially intersecting surface patches and a penetration direction for each contact region, using a fast collision detection algorithm for convex objects [7].

We then refine the query result by considering a localized height-field representation of the deformable geometry parameterized on a 2D domain. We have developed an image-space algorithm for computing directional penetration depth between deformable models that is well suited for parallel implementation on GPUs that achieves fast computation at haptic update rates. We assume that, within regions of contact, the surfaces can be described as height fields. The directional penetration depth can then be defined as the height field difference between the intersecting patches, in the local direction of penetration [13]. Given a contact region between two skin surface patches, we identify a contact plane *D* as the plane passing between the contact points and oriented according to the contact normal **n**. In a first pass we rasterize the high-resolution surface geometry using an orthogonal projection *P* aligned with the normal **n** of contact plane *D* and use a simple fragment program to write out depth to the frame buffer. In the next pass, we subtract the depth, which yields positive values where the patches intersect in the image plane. The resulting localized penetration depth field is stored in the graphics hardware texture memory.

Finally, we check for high-resolution skin surface collisions and collect colliding skin vertices, again with the aid of graphics hardware. For each vertex of the skin mesh, we use the projection matrix associated with *P* to determine its position in the image plane and read back its penetration depth value from GPU texture memory to the CPU. Note that in practice this last step is performed in a forward algorithm using a fragment program on the GPU, because reading back individual values to the CPU is potentially very slow. Moreover, in this way we exploit the parallelism in the GPU hardware to do the projection transforms. The end result of this final stage is a list of colliding skin vertices, along with their local directional penetration depths, as shown in Figure 4. The penetration depths can then be used for haptic rendering of net reaction forces and torques using a penalty-based approach, which we will briefly describe next.

5 Penalty-Based Collision Response

The collision query returns the contact normal and penetration depth for each colliding point while preserving the original high-frequency geometry. As a result, penalty-based collision response forces can be computed on a per point basis. We compute the reaction forces based on a force-displacement relationship between the surface points s_1, s_2 and the penetration depth δ. A spring of zero rest length is attached between s_1, s_2 which results in the force $\mathbf{F_p} = -k\,\delta$, where k is the stiffness of the spring. The stability problems introduced by stiff springs can be countered by adding a damping term, which gives $\mathbf{F_p} = -(k\,\delta + d\,\delta')$. Here d is the damping constant and δ' is the relative velocity in the direction of contact normal. Once given the penetration depth, to compute the force feedback is very fast and takes little time to display the resulting forces and torques to the user.

6 Results

We have tested our novel haptic rendering algorithm on deformable models of high complexity (consisting of many tens of thousands of surface elements) with rich surface deformation. The low-resolution proxies are simplified down to a few hundred of triangles, which is roughly the size that can be handled by existing collision detection techniques [14]. In the case of the head model in Fig. 6, which has 44,000 deformable vertices, we were able to obtain per-vertex penetration depth information within 2 msec for touching one and within 6 msec for two of them in contact. The gear model in Fig. 7 has 29,000 vertices each and takes about 3 msec for two gears in contact to compute the penetration depth between them. The pumpkin model (Fig. 8) consists of 30,000 vertices each and also takes roughly 3 msec to perform the proximity queries between two such models in contact. These timings include the transfer of the contact information to the deformation domain, where it is directly available for dynamics computation.

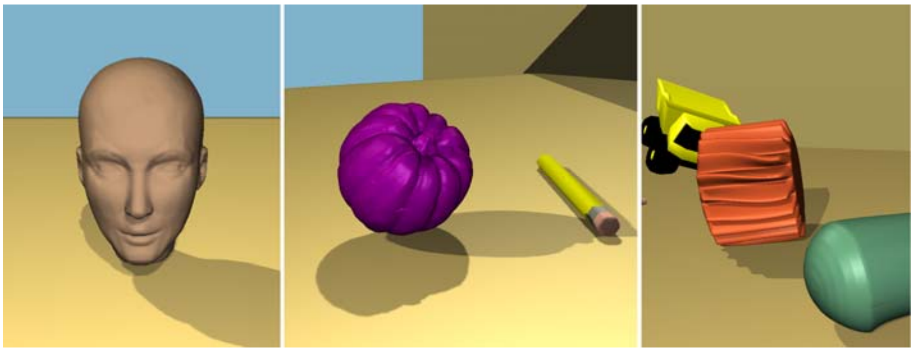

Figure 6. Models used for benchmarking collision detection performance. Shown are a head model, a pumpkin model and a deformable gear with respectively 44K, 30K and 29K vertices.

Using our scalable and output-sensitive collision detection algorithm, we compute object penetration depth that captures the original high-frequency geometry, and we then display dynamic effects due to surface deformation that would otherwise be missed, such as the deformation on the bottom of the pumpkins and the dynamic rolling behavior of the gears due to the deformation of its teeth.

In Fig. 1, we show a scene where the user uses a haptic device, PHANTOM 1.5, to push a human body model with high-resolution features on its surfaces. This model consists of 3563 vertices and 10,340 tetrahedral elements (see Fig. 2, 4, and 5). Figure 7 shows two more screenshots of the user touching the model's upper leg and its arm/shoulder respectively. The proximity queries for haptic rendering take less than 2 msec on this model. All contacts on the surface have global effect on the entire deformable layer, they are processed simultaneously and robustly. Without any precomputation of dynamics or significant storage requirement, we were able to haptically render the human model and other benchmarks shown in this paper at interactive rates, on a 3.2 GHz Pentium 4 PC with NVidia GeForce 7800.

Our approach is considerably faster than other methods that enable large time steps, such as those that focus on the surface deformation [1] and corotational methods that compute deformations within the entire volume [12], with more stable collision response. Our approach can also handle many more contact points than novel quasi-rigid dynamics algorithms using LCP [16], while producing richer deformations for haptic rendering.

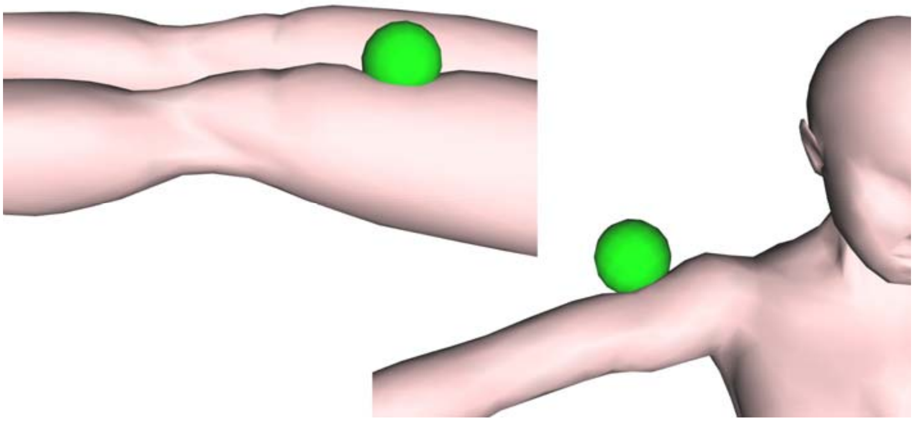

Figure 7. The haptic tool (the green sphere) is in continuous contact with the human skin. Our algorithm computes skin deformation and force feedback to the user at interactive rates.

7 Conclusion and Future Work

We have presented an interactive 6-DoF haptic rendering algorithm for deformable bodies with high-resolution surface geometry and large contact regions. We have overcome the computational challenges of this problem by adapting a novel layered representation of deformable models where physically based deformation and haptic force feedback are computed in a multi-level algorithm. Based on this representation, we have also presented efficient collision detection which enables fast contact response for haptic manipulation of deformable models.

The use of a layered representation obviously poses some limitations on the type of deformations that can be modeled, but they are well suited for many real-world models, such as human bodies and soft objects that deform up to 30-40% of its radius. Our representation can be extended to articulated, flexible bodies that undergo skeletal deformations, by augmenting the generalized coordinate set of the core representation to include multi-body systems.

We plan to apply our algorithm and representation to the simulation of virtual human with skeletal control and plausible collision avoidance. Further optimizations of the algorithm and the steady growth of parallel processors offer possibilities for interactive, detailed physically-based haptic rendering of skin deformations in surgical simulator.

Acknowledgements. This work was supported in part by ARO Contracts W911NF-04-1-0088, NSF awards 0400134, 0429583 and 0404088, DARPA/RDECOM Contract N61339-04-C-0043 and Disruptive Technology Office.
This research was also funded in part by the NCCR CoMe of the Swiss NSF. The authors would also like to thank the session chair, Dr. Lawrence Rosenblum, for organizing this special session.

References

1. M. Bro-Nielsen and S. Cotin. Real-time volumetric deformable models for surgery simulation using finite elements and condensation. In Computer Graphics Forum, 15(3) (Eurographics'96), pp. 57–66 (1996).

2. J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. IEEE International Conference on Robotics and Automation, pp. 3205–3210 (1994).

3. J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 140–145 (1995).

4. S. Cotin, H. Delingette, and N. Ayache. Realtime elastic deformation of soft tissues for surgery simulation. IEEE Trans. on Visualization and Computer Graphics, 5:1 (1999).

5. C. Duriez, C. Andriot, and A. Kheddar. A multithreaded approach for deformable/rigid contacts with haptic feedback. Proc. of Haptics Symposium (2004).

6. C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. Proc. of IEEE TVCG, 12: 1 (2006).

7. S. A. Ehmann, M. C. Lin. Accelerated proximity queries between convex polyhedra by multi-level Voronoi marching. Proc. of International Conference on Intelligent Robots and Systems (2000).

8. N. Galoppo, M. A. Otaduy, P. Mecklenburg, M. Gross, M. C. Lin. Fast simulation of deformable models in contact using dynamic deformation textures. Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation (2006), pp. 73—82.

9. D. L. James and D. K. Pai. Artdefo: Accurate real time deformable objects. Proc. of ACM SIGGRAPH (1999).

10. M. Mahvash and V. Hayward. High-fidelity passive force-reflecting virtual environments. IEEE Transactions on Robotics, 21(1):38–46 (2005).

11. W. McNeely, K. Puterbaugh, and J. Troy. Six degree-of freedom haptic rendering using voxel sampling. Proc. of ACM SIGGRAPH, pp. 401–408 (1999).

12. M. Muller, J. Dorsey, L. McMillan, R. Jagnow, and B. Cutler. Stable real-time deformations. Proc. of ACM SIGGRAPH Symposium on Computer Animation (2002).

13. M. A. Otaduy, N. Jain, A. Sud, M. C. Lin. Haptic display of interaction between textured models. In *Proc. of IEEE Visualization* (2004).

14. M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. Proc. of ACM SIGGRAPH, pp. 543—553 (2003).

15. M. A. Otaduy and M. C. Lin. Stable and responsive six degree-of-freedom haptic manipulation using implicit integration. Proc. of World Haptics Conference (2005).

16. M. Pauly, D. K. Pai, L. J. Guibas.: Quasirigid objects in contact. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2004).

17. D. Terzopoulos, A. Witkin. Physically based models with rigid and deformable components. *IEEE Computer Graphics and Applications 8*, 6 (1988).