Sensation Preserving Simplification for Haptic Rendering

Miguel A. Otaduy   Ming C. Lin
Department of Computer Science
University of North Carolina at Chapel Hill
http://gamma.cs.unc.edu/LODHaptics/

Abstract

We introduce a novel "sensation preserving" simplification algorithm for faster collision queries between two polyhedral objects in haptic rendering. Given a polyhedral model, we construct a multiresolution hierarchy using "filtered edge collapse", subject to constraints imposed by collision detection. The resulting hierarchy is then used to compute fast contact response for haptic display. The computation model is inspired by human tactual perception of contact information. We have successfully applied and demonstrated the algorithm on a time-critical collision query framework for haptically displaying complex object-object interaction. Compared to existing exact contact query algorithms, we observe noticeable performance improvement in update rates with little degradation in the haptic perception of contacts.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Hierarchy and Geometric Transformations; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality

Keywords: Level-of-Detail Algorithms, Haptics, Collision Detection

1 Introduction

Haptic rendering, or force display, is emerging as an alternative form or an augmentation for information presentation, in addition to visual and auditory rendering. The sense of touch is one of the most important sensory channels, yet it is relatively poorly understood as a form of human-machine interface. Coupled with graphical rendering, force feedback can enhance the user's ability to interact intuitively with complex synthetic environments and increase the sense of presence in exploring virtual worlds [Brooks, Jr. et al. 1990; Mark et al. 1996; Hollerbach et al. 1997; Salisbury 1999].

The first step in displaying force and torque between two 3D virtual objects is collision query and contact handling. Collision detection has been well studied, and many practical techniques and theoretical advances have been developed (see surveys by Lin and Gottschalk [1998] and Klosowski et al. [1998]). Yet, despite the huge body of literature in this area, the existing algorithms cannot run at the desired force update rates (at least hundreds of Hz but preferably several kHz) for haptic rendering of complex models. This is mainly due to the fact that the optimal running time of any collision detection algorithm intrinsically depends on both the input and output sizes of the problem. Those in turn depend on both the combinatorial complexity and the contact configuration of the objects involved in the queries. 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 desired update rates.

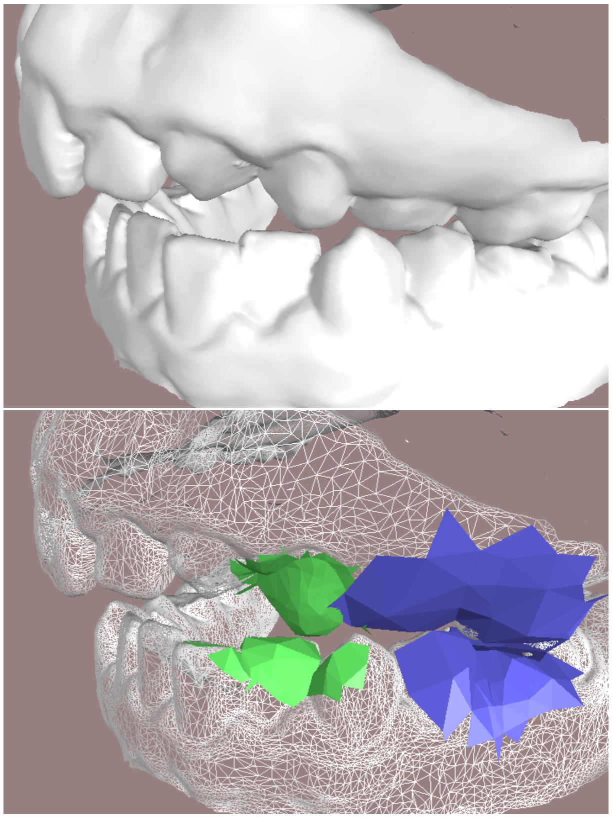

Figure 1: **Adaptive Resolution Selection.** *Top: Moving jaws in contact, rendered at their highest resolution; Bottom: The appropriate resolution (shown in blue and green) is selected adaptively for each contact location, while the finest resolution is displayed in wireframe.*

Inspired by the large body of research in digital geometry processing and mesh simplification, we propose an algorithm based on multiresolution hierarchies of object geometry to perform time-critical collision queries for haptic rendering. In addition, our method is influenced by findings from tactual perception and spatial recognition to preserve pertinent contact information for haptic display.

Main Contribution: 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 generate a series of approximations for the object using *filtered edge collapse* operations, which smooth away high-frequency detail in low-resolution approximations while respecting the convexity constraints imposed by collision queries. Our goal is to generate a multiresolution hierarchy that can also be used as a bounding volume hierarchy for time-critical contact force computation in haptic rendering. Our computation model is based on a criterion that preserves perceivable contact details, effectively making the simplified model feel practically the same as the original. The resulting multiresolution hierarchy enables use of varying resolution approximations in contact queries at different locations across the surfaces of the objects in contact, depending on each contact configuration, as shown in Fig. 1. The key results in this paper include:

- A novel simplification algorithm, based on a formal definition of resolution, to generate representations for contact queries;
- A collision detection framework that dynamically selects adaptive levels of detail at each contact location;
- Application of this framework to real-time haptic rendering of complex object-object interaction.

This approach allows us to bound both the input and output sizes of the problem, thus achieving the desired contact query performance for force display. We have applied our approach to haptic rendering of complex models in contact configurations that are particularly challenging to collision detection algorithms. Compared to existing exact contact query algorithms, we are able to achieve up to two orders of magnitude performance improvement with little degradation in the haptic perception of contacts.

Organization: The rest of the paper is organized as follows. In Section 2, we give a brief survey of related work. Section 3 presents the haptic perception characteristics central to the design of our computational model and the overview of our approach. We describe the construction of the multiresolution hierarchy in Section 4 and sensation preserving contact queries using the hierarchy in Section 5. We address implementation issues and present results in Section 6. We conclude with a discussion and analysis of our approach and implementation, as well as future research directions.

2 Previous Work

Our research draws on a large collection of knowledge in mesh simplification, signal processing for digital geometry, collision detection, and haptic display. We briefly survey related work here.

2.1 Polygonal Simplification

Polygonal simplification has been an active research topic for the last decade. Numerous approaches have been proposed. We refer readers to an excellent new book on this subject [Luebke et al. 2002]. However, note that the growing interest in perception-based simplification for interactive rendering, e.g. [Luebke and Hallen 2001], has been based on human visual perceptual metrics. Our approach differs in many ways from existing work in this area, and our target application, haptic rendering, has a much higher performance requirement than visual display. Although we precompute the level of detail (LOD) hierarchy offline, the way we select the appropriate LOD on the fly is "contact-dependent" at each contact location across the object surfaces. Our approach bears a closer resemblance to view-dependent simplification [Luebke et al. 2002], which uses higher resolution representations on the silhouette of the object and much coarser approximations on the rest of the object that is not as noticeable to the viewpoint.

2.2 Signal Processing for Digital Geometry

Much of the work presented in this paper takes advantage of observations made in signal processing of meshes, since many concepts in multiresolution representations can be analyzed using frequency domain analysis. By generalizing discrete Fourier analysis to meshes, Taubin [1995] introduced a novel linear-time low-pass filtering algorithm for surface smoothing. This algorithm can be extended to accommodate different types of geometric constraints as well. Through a non-uniform relaxation procedure, whose weights depend on the geometry instead of connectivity, Guskov et al. [1999] generalized signal processing tools to irregular triangle meshes. Our work borrows some ideas from the relaxation techniques proposed in this paper.

2.3 Collision Detection

Hierarchical data structures have been widely used to design efficient algorithms for interference detection between geometric models (see surveys by Lin and Gottschalk [1998] and Klosowski et al. [1998]). Typical examples of bounding volumes include axis-aligned boxes and spheres, chosen for their simplicity in performing overlap tests between two such volumes. Other hierarchies include k-d trees and octrees, OBBTree, cone-trees, R-trees and their variants, trees based on S-bounds, etc. [Lin and Gottschalk 1998; Klosowski et al. 1998]. Additional spatial representations are based on BSP's and their extensions to multi-space partitions, space-time bounds or four-dimensional tests (see a brief survey by Redon et al. [2002]), and many more.

Hubbard [1994] first introduced the concept of time-critical collision detection using sphere-trees. Collision queries can be performed as far down the sphere-trees as time allows, without traversing the entire hierarchy. This concept can be applied to any type of bounding volume hierarchy (BVH). However, no tight error bounds have been provided using this approach. An error metric is often desirable for interactive applications to formally and rigorously quantify the amount of error introduced. Approaches that exploit motion coherence and hierarchical representations for fast distance computation between convex polytopes have been proposed [Guibas et al. 1999; Ehmann and Lin 2000]. However, these techniques are only applicable to convex polyhedra.

O'Sullivan and Dingliana [2001] studied LOD techniques for collision simulations and investigated different factors affecting collision perception, including eccentricity, separation, distractors, causality, and accuracy of simulation results. Based on a model of human visual perception validated by psychophysical experiments, the feasibility of using these factors for scheduling interruptible collision detection among large numbers of visually homogeneous objects is demonstrated. Instead of addressing the scheduling of multiple collision events among many objects, we focus on the problem of contact queries between two highly complex objects. Our approach, guided by a completely different tactual perception for haptic rendering, has distinct goals differing significantly from theirs.

Recently, GPU accelerated techniques have also been proposed for collision queries [Lombardo et al. 1999; Hoff et al. 2001]. Though fast, these approaches are not currently suitable for haptic rendering, since the readback from framebuffer and depth buffer cannot be done fast enough to perform queries at haptic update rates.

2.4 Haptics

Over the last few years, haptic rendering of geometric models has received much attention. Most previous work addresses issues related to rendering the interaction between a probe point and 3D objects [Ruspini et al. 1997]. This problem is characterized by high spatial coherence, and its computation can be localized. By contrast, we attack the problem of force rendering for arbitrary 3D polyhedral object-object interaction, which involves a substantially higher computational complexity. Force rendering of object-object interaction also makes it much more challenging to correctly cache results from previous computations.

McNeely et al. [1999] proposed "point-voxel sampling", a discretized approximation technique for contact queries that generates points on moving objects and voxels on static geometry. This approximation algorithm is the first to offer run-time performance independent of the environment's input size by sampling the object geometry at a resolution that the given processor can handle. A recent approach proposed by Gregory et al. [2000] is limited to haptic display of object-object interaction for relatively simple models that can be easily represented as unions of convex pieces. Kim et al. [2002] attempt to increase the stability of the force feedback using contact clustering, but their algorithm for contact queries suffers from the same computational complexity.

The idea of using multiresolution representations for haptic rendering has been recently investigated by several researchers. Pai and Reissel [1997] investigated the use of multiresolution image curves for 2D haptic interaction. El-Sana and Varsheny [2000] proposed the construction of a multiresolution hierarchy of the model during preprocessing. At run-time, a high-detail representation is used for regions around the probe pointer and a coarser representation farther away. The proposed approach only applies to haptic rendering using a point probe exploring a 3D model. It does not extend naturally to force display of two interacting 3D objects, since multiple disjoint contacts can occur simultaneously at widely varying locations without much spatial coherence. The latter problem is the focus of our paper.

3 Overview

In this section, we first present important findings from studies on tactual perception that guide our computational model. Then, we describe the requirements for haptic rendering and our design goals.

3.1 Haptic Perception of Surface Detail

From a perceptual perspective, both formal studies and experimental observations have been made regarding the impact of contact areas and relative size (or curvature) of features to the size of the contact probe (or finger) on identifying fine surface features.

Klatzky and Lederman [1995] conducted and documented studies on identification of objects using "haptic glance", a brief haptic exposure that placed several temporal and spatial constraints on stimulus processing. They showed that a larger contact surface area helped in the identification of textures or patterns, though 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 [1999] defined a fine (geometric) surface feature based on the ratio of its curvature to the radius of the fingertip acquiring the surface data. Their paper gives examples on how a larger fingertip, and thus a larger surface contact area, can miss some surface detail.

In this paper, we mainly focus on geometric surface features, not microscopic surface roughness or friction. We draw the following key observation from these studies relevant to our computational model:

> *Human haptic perception of the existence of 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.*

Here we broadly define the size of a given feature in all three dimensions, namely width, length, and height. The width and length of a feature can be intuitively considered as the "inverse of resolution" (formally defined in Sec. 4) of the simplified model. That is, higher resolution around a local area implies that the width and length of the geometric surface features in that neighborhood are smaller, and vice versa. We extend the concept of "height" to include a perceivable amount of surface deviation introduced in the simplification process, according to haptic perception.

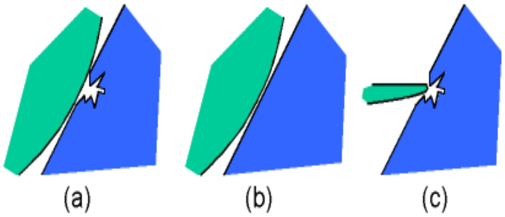

Figure 2: **Contact area and resolution:** *(a) high resolution model with large contact area; (b) low resolution model with large contact area; (c) high resolution model with small contact area.*

As illustrated in Fig. 2, the observation drawn by Okamura and Cutkosky [1999] for tactile feedback can extend to haptic rendering of contact forces between rigid bodies. 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 shown in Fig. 2(a) and (b).

Our proposed model of acceptable error metrics differs notably from that of human visual perception in both the current mesh simplification literature and visual collision perception. In visual rendering, a combination of surface deviation (or Hausdorff distance) and the viewing distance from the object is used to determine if the representation of the objects requires higher resolution. In haptic rendering, on the other hand, this is governed by the relationship among the surface deviation, the resolution of the simplified model, and the contact surface area. We will later show how this relationship lays the foundation of our algorithmic design and contact query process for haptic rendering in Sec. 4 and Sec. 5.

3.2 Requirements and Design Desiderata

We aim to create multiresolution representations where geometric surface detail is filtered when it cannot be perceived by the sense of touch. The resulting multiresolution hierarchies can be used to perform time-critical contact queries that stop when the reported result is accurate up to some tolerance value. This helps to automatically speed up the contact query computation for haptic rendering.

In our haptic rendering framework, we have chosen BVHs of convex hulls, because overlap tests between convex hulls can be executed rapidly in expected constant time with motion coherence [Guibas et al. 1999]. Furthermore, convex hulls provide at least equally good, if not superior, fitting to the underlying geometry as OBBs [Gottschalk et al. 1996] or k-dops [Klosowski et al. 1998].

We integrate BVHs of convex hulls with multiresolution representations so that the hierarchies, while being used for effective collision detection, can themselves be used to report contact points and normals with bounded errors at different levels of resolution. To summarize, our goal is to design **multiresolution hierarchies that:**

1. **Minimize perceptible surface deviation.** We achieve this goal by filtering the detail at appropriate resolutions and by using a novel sensation preserving refinement test for collision detection;

2. **Reduce the polygonal complexity of low resolution representations.** This objective is achieved by incorporating mesh decimation during the creation of the hierarchy;
3. **Are themselves BVHs of convex hulls.** We perform a surface convex decomposition on the given triangular mesh and maintain it across the hierarchy. The convex surface decomposition places both local and global convexity constraints on the mesh decimation process.

Our algorithm assumes that the input models can be represented as oriented 2-manifold triangular meshes with boundaries.

3.3 Notation and Terminology

We use bold-face letters to distinguish a vector (e.g. a point, normal, etc.) from a scalar value. In Table 1, we enumerate the notations we use throughout the paper.

Notation	Meaning
r, r_i, r_j	Different resolutions
M^k	An LOD of a mesh M with a resolution r_k
c_i	A convex surface patch
C_i	A convex piece constructed as the convex hull of a patch c_i
$e(\mathbf{v_1}, \mathbf{v_2})$	An edge between two vertices $\mathbf{v_1}$ and $\mathbf{v_2}$
s, s_a, s_b	Surface deviations
D, D_a, D_b	Contact areas
q	A distance query between two convex pieces
Q	A contact query between two objects that consists of multiple distance queries q

Table 1: **Notation Table**

4 Multiresolution Hierarchy

In this section we describe the hierarchical representations of triangulated models used to perform sensation preserving contact queries for haptic rendering.

We create a hierarchy of static levels of detail (LOD), each level representing an approximation to the original triangular mesh at a different resolution (i.e. spatial scale), to be formally defined next. Because our goal is to provide an error bound arising from contact queries using simplified models, we must design a multiresolution hierarchy that computes error metrics between each LOD and the original model.

Conceptually, an LOD at resolution r_j of a mesh M, M^j, can be obtained from an LOD at a lower resolution r_i, M^i, by adding detail at resolutions in the range $[r_i, r_j]$. Our approach for generating LODs reverses this definition, so LODs at low resolution are obtained by removing detail at high resolution. While the detail is being removed, we quantify it and compute the surface deviation.

Following the LOD generation, we obtain a hierarchy where an LOD at resolution r_j preserves the lower resolution geometric information, while the higher resolution detail might have been culled away.

> We generate each LOD by a sequence of **filtered edge collapse** operations (to be defined in Sec. 4.2) that perform filtering and mesh decimation,
>
> *subject* to both **local and global convexity constraints** imposed by the collision detection module of the haptic rendering framework.

4.1 Definition and Computation of Resolution

Before we explain how we generate each LOD, we must first formally define what we consider as a resolution in our hierarchical representation. We follow the framework of signal processing for irregular meshes. Our definition of resolution for irregular meshes assumes that a triangular mesh M can be considered as a sampled version of a smooth surface S, which has been later reconstructed via linear interpolation. The vertices of the mesh are samples of the original surface, while edges and faces are the result of the reconstruction.

Our formal definition of sampling resolution for irregular meshes is based on the 1D setting. For a 1D function $F(x)$, the sampling resolution r is the inverse of the distance between two subsequent samples on the real line. This distance can also be interpreted as the projection of the segment between two samples v_1 and v_2 of the function on the average value. The average value is the low resolution representation of the function itself, and can be obtained by lowpass filtering.

Extrapolating this idea to irregular meshes, the sampling resolution of an edge $(\mathbf{v_1}, \mathbf{v_2})$ of the mesh M at resolution r_j, M^j, can be estimated as the inverse of the projected length of the edge onto a low resolution representation of the mesh, M^i.

We locally compute the low resolution mesh M^i by filtering the mesh M^j, applying the filtered edge collapse operation to the edge $(\mathbf{v_1}, \mathbf{v_2})$. Then we compute the normal $\mathbf{N}$ of the resulting vertex $\tilde{\mathbf{v}}_\mathbf{3}$ by averaging the normals of incident triangles. Finally, we project the edge on the tangent plane Π defined by $\mathbf{N}$. The resolution r is computed as the inverse of the length of the projected edge.

$$r = \frac{1}{\|(\mathbf{v_1} - \mathbf{v_2}) - ((\mathbf{v_1} - \mathbf{v_2}) \cdot \mathbf{N}) \cdot \mathbf{N}\|} \tag{1}$$

4.2 Filtered Edge Collapse

As stated, our multiresolution hierarchy is obtained through mesh simplification. We have selected edge collapse as the atomic decimation operation for two main reasons:

1. Under the required self-intersection tests, edge collapse can guarantee preservation of topology, a requirement for maintaining a surface convex decomposition of the object during the hierarchy construction.
2. Topologically, an edge collapse can be regarded as a local downsampling operation, where two samples (i.e. vertices) are merged into a single one.

In the construction of the hierarchy, we aim to:

1. Generate multiresolution representations with low polygonal complexity at low resolution for accelerating contact queries;
2. Filter detail as we compute low resolution LODs. This approach allows more aggressive simplification and enables faster merging of convex pieces to build the hierarchy.

These two goals are achieved by merging downsampling and filtering operations in one atomic operation, which we call *filtered edge collapse*.

In the filtered edge collapse operation, an edge $(\mathbf{v_1}, \mathbf{v_2})$ is first topologically collapsed to a vertex $\hat{\mathbf{v}}_\mathbf{3}$. This step provides the downsampling. Then, given its connectivity, $\hat{\mathbf{v}}_\mathbf{3}$ is relaxed to a position $\tilde{\mathbf{v}}_\mathbf{3}$, which provides the filtering. In our implementation, we used a relaxation operation based on the minimization of second order divided differences [Guskov et al. 1999]. Intuitively, this resembles the minimization of dihedral angles, without much affecting the shape of the triangles. We also tried other filtering techniques, such as those proposed by Taubin [1995], with very similar results. However, linear functions are invariant under the minimization of second order differences. This is consistent with the selection of the tangent plane of the filtered mesh as the low resolution representation for the computation of resolutions.

In order to apply the relaxation to $\hat{\mathbf{v}}_3$, we need to compute a local parameterization. This local parameterization requires an initial position of $\hat{\mathbf{v}}_3$, which is computed using quadric error metrics, proposed by Garland and Heckbert [1997].

To sum up, the goal of our simplification and filtering process is to create multiresolution hierarchies for contact queries. As mentioned earlier, the collision detection module imposes convexity constraints on filtered edge collapse. Next, we will describe how the convexity constraints are satisfied.

4.3 Convexity Constraints

Due to the requirements of haptic rendering, we have chosen to perform collision detection using the Voronoi marching algorithm and surface convex decomposition as described by Ehmann and Lin [2001], for this approach provides us with both the distance and contact information needed for force display and its implementation is available to the public. A surface convex decomposition is first computed for the original mesh, and then a hierarchy of convex pieces is created.

The surface convex decomposition yields a set of convex surface patches $\{c_1, c_2, ..., c_n\}$ [Chazelle et al. 1997; Ehmann and Lin 2001]. For the correctness of the collision detection algorithm, the convex hulls of these patches are computed, resulting in convex pieces $\{C_1, C_2, ..., C_n\}$.

The initial convex decomposition can be created using techniques presented by Chazelle et al. [1997]. However, our hierarchy is created in a novel way. Instead of creating convex hulls of pairs of convex pieces and joining them into a single convex piece, we merge neighboring sets of convex patches as long as they represent a single *valid* convex patch. The implications of this procedure for the collision detection algorithm are explained in Sec. 5.

Let c_1 and c_2 be two convex patches of LOD M^j. Let $c = c_1 \cup c_2$ be a surface patch of M^j. After a filtered edge collapse operation is applied to M^j, c_1 and c_2 will be merged if c constitutes a valid convex patch. The convex hull of c, C, becomes the parent of C_1 and C_2 in the hierarchy of convex pieces for collision detection.

When a filtered edge collapse takes place, the convex patches in the neighborhood of the collapsed edge may be affected. Their boundary has to be updated accordingly.

A surface convex decomposition for the collision detection algorithm must meet several constraints:

1. All the interior edges of a convex patch must themselves be convex.
2. No vertex in a convex patch may be visible from any face in the patch, except the ones incident on it.
3. The virtual faces added to complete the convex hulls of the convex patches cannot intersect the mesh.

We consider the first constraint as a local constraint and the other two as global constraints. Before a filtered edge collapse operation is applied, we must check that the convexity constraints are preserved for all the convex pieces. Local and global convexity constraints are treated separately.

4.3.1 Local Convexity Constraints

Let $e \equiv (\mathbf{v}_1, \mathbf{v}_2)$ be a candidate edge that will be tested for a filtered edge collapse. Let $\mathbf{v}_3$ represent the vertex resulting from the edge collapse, as well as its associated position. The edges in the 1-ring neighborhood of $\mathbf{v}_3$ are susceptible to changing from convex to reflex and vice versa. Interior edges of convex patches are convex before the filtered edge collapse and must remain convex after it. These constraints can be expressed as linear constraints in the position of $\mathbf{v}_3$.

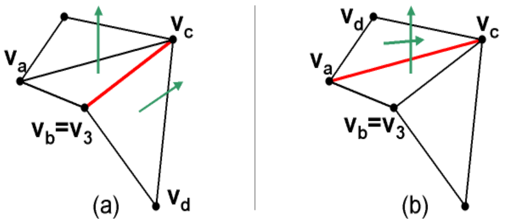

Figure 3: **Local Convexity Constraints.**

Given e, the edge to be collapsed, two possible types of interior edges of convex patches exist: edges incident to $\mathbf{v}_3$ and edges opposite to $\mathbf{v}_3$, as shown in Fig. 3. However, both cases can be treated equally. Assigning $\mathbf{v_a}$, $\mathbf{v_b}$, $\mathbf{v_c}$ and $\mathbf{v_d}$ vertices as in Fig. 3, the convexity constraint of an edge can be expressed as a negative volume for the parallelepiped defined by the adjacent triangles:

$$((\mathbf{v_b} - \mathbf{v_a}) \times (\mathbf{v_c} - \mathbf{v_a})) \cdot (\mathbf{v_d} - \mathbf{v_a}) \leq 0 \qquad (2)$$

To satisfy the convexity constraints, we have opted for formulating an optimization program, where $\mathbf{v}_3$ is constrained to the segment between $\hat{\mathbf{v}}_3$ and $\tilde{\mathbf{v}}_3$, and the objective function is the distance to $\tilde{\mathbf{v}}_3$. This optimization program is unidimensional. Because distance in one dimension is linear, it is a simple linear program in one dimension.

The position of the result of the constrained filtered edge collapse can be written as a linear interpolation between the initial position and the goal position:

$$\mathbf{v}_3 = u \cdot \hat{\mathbf{v}}_3 + (1-u) \cdot \tilde{\mathbf{v}}_3 \qquad (3)$$

The limit constraints can be expressed as $u \geq 0$ and $u \leq 1$.
The convexity constraints in Eq. 2 can be rewritten as:

$$\begin{aligned} A \cdot u + B \geq 0, \quad \text{where} \qquad (4) \\ A = ((\mathbf{v_d} - \mathbf{v_a}) \times (\mathbf{v_c} - \mathbf{v_a})) \cdot (\hat{\mathbf{v}}_3 - \tilde{\mathbf{v}}_3) \\ B = ((\mathbf{v_d} - \mathbf{v_a}) \times (\mathbf{v_c} - \mathbf{v_a})) \cdot (\tilde{\mathbf{v}}_3 - \mathbf{v_a}) \end{aligned}$$

$\mathbf{v}_3$ is computed for the minimum value of u that meets all the constraints. When $\tilde{\mathbf{v}}_3$ is not a feasible solution but a solution exists, the constrained filtered edge collapse can be regarded as a partial filter.

4.3.2 Global Convexity Constraints

The global convexity constraints are too complicated to be expressed explicitly, so they cannot be incorporated into the filtering process. Instead, they have to be verified after the filtering has been performed. We conduct this verification by computing the affected convex pieces after the edge collapse and performing the required intersection tests, using OBBs [Gottschalk et al. 1996] and spatial partitioning.

If a position $\mathbf{v}_3$ that meets the local convexity constraints has been found, we check the global constraints. If they are met, the edge collapse is valid. If they are not met, then we check them at $\hat{\mathbf{v}}_3$. If they are not met at $\hat{\mathbf{v}}_3$ either, the edge collapse is considered invalid and we disallow it. If $\hat{\mathbf{v}}_3$ meets the global constraints, we perform a bisection search between $\hat{\mathbf{v}}_3$ and $\mathbf{v}_3$ of up to K iterations (in our current implementation $K = 3$), searching for the position closest to $\tilde{\mathbf{v}}_3$ that meets the global convexity constraints, as shown in Fig. 4. $\mathbf{v}_3$ is reassigned to this position.

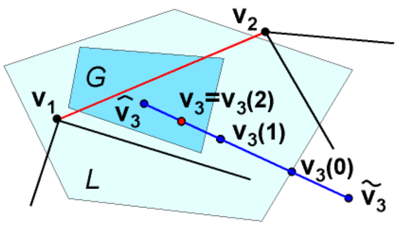

Figure 4: **Filtered Edge Collapse with Convexity Constraints.** *The figure shows a filtered edge collapse where bisection search is required to find a position that meets the convexity constraints. G and L represent feasible regions of global and local constraints respectively.*

4.4 Multiresolution Hierarchy Generation

The hierarchy of LODs is created by applying successive filtered edge collapses on the given mesh, while performing a surface convex decomposition and merging convex pieces. First we compute the convex decomposition of the initial mesh. We then compute the value of resolution for all edges, and set them as valid for collapse. The edges are inserted in a priority queue, where edges with higher resolution have higher priority.

The main processing loop always tries to filter and collapse the edge with highest priority. If the filtered edge collapse is successful, the affected edges update their resolution and priority, and they are reset as valid for collapse. Moreover, the filtering and simplification may have relaxed some convexity constraints in the neighborhood of the collapsed edge, so we attempt to merge convex pieces in the process as well. If the filtered edge collapse fails, the edge is set as invalid for collapse. The process continues until no edges are valid for collapse.

This process must yield a hierarchy of static LODs. We have decided to generate a new LOD every time the number of convex pieces is halved. All the pieces in LOD M^j that are merged to a common piece $C \in M^{j+1}$ during the processing will have C as their parent in the BVH.

Ideally, the process will end with one single convex piece, which serves as the root for the hierarchy to be used in the collision detection. However, this result is rarely achieved in practice, due to topological and geometric constraints that cannot be removed by a local operation such as filtered edge collapse. In such cases, the hierarchy is completed using a pairwise convex hull merging step. We call these remaining completing LODs “free” LODs.

During the process, we assign to each LOD M^j an associated resolution r_j. This resolution is the smallest resolution of an edge that has been collapsed before the LOD M^j is generated. Geometrically it means that the LOD M^j preserves all the detail of the original mesh at a resolution lower than r_j. In our sensation preserving simplification for haptic rendering, we wish to maximize the resolution at which LODs are generated. As will be explained in Sec. 5, the perceptual error for haptic rendering is measured by taking into account the resolution of the surface detail culled away. By maximizing the resolution at which LODs are generated, the contact queries can be completed faster. This is the basis for selecting edge resolution as the priority for filtered edge collapses. The pseudo code for the entire process of hierarchy construction is given in Appendix A on the conference proceedings CD.

Fig. 5 shows several of the LODs obtained when processing a model of a lower jaw (see Sec. 6 for statistics on this model). The LODs 3 and 6 shown in the figure are obtained from the original model by our simplification process. The convex pieces shown for the original model are successively merged to create the BVH during the process of simplification. Thus, the multiresolution hierarchy itself serves as BVH for collision detection. Unlike other types of BVHs, with our simplification processing the different levels of the BVH only bound their associated LOD; they do not necessarily bound the original surface. This fact has some implications for the contact queries, described in Sec. 5.3. The free LODs 11 and 14 in the figure are obtained through pairwise merging of convex hulls. They serve to complete the BVH, but cannot be considered as LODs of a multiresolution hierarchy. Fig. 6 shows a more detailed view of the simplification and merging process. Notice how in the creation of the first LOD, most of the simplification and merging takes place at the gums. The gums are, indeed, the locations with detail at the highest resolution. When the processing reaches LOD 7, one tooth in particular is covered by a single convex patch, thus showing the success of the processing.

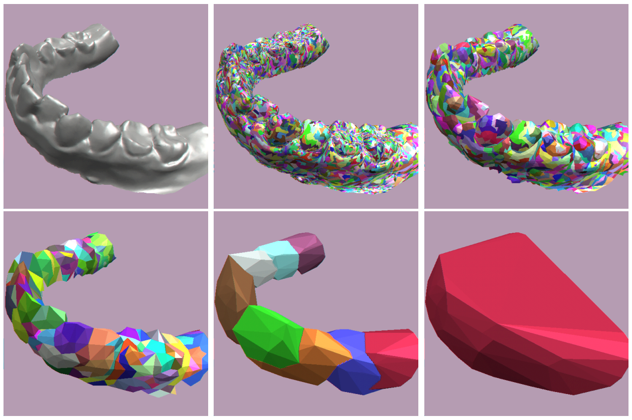

Figure 5: **Hierarchy of the Lower Jaw.** *From left to right and top to bottom, original mesh, LOD_0, and convex decompositions of LOD_0, LOD_3, LOD_6, LOD_{11} and LOD_{14}.*

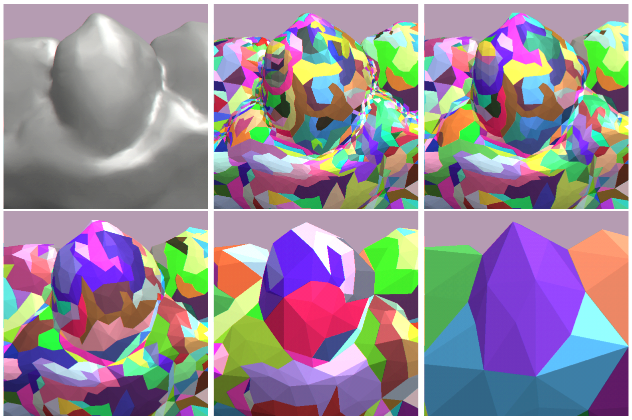

Figure 6: **Detail View of the Hierarchy.** *From left to right and top to bottom, original mesh, LOD_0, and convex decompositions of LOD_0, LOD_1, LOD_2, LOD_4 and LOD_7.*

4.5 Error Metrics

In this section, we present the parameters that must be computed after the hierarchy is created, in order to quantify the error for sensation preserving haptic rendering. The utilization of these parameters during the contact queries is explained in Sec. 5. To perform sensation preserving haptic rendering using a multiresolution hierarchy, we must measure the error that is introduced in the contact query and force computation and refine the query if the error is above a given tolerance. Once the hierarchies of LODs are created, with the resolution r computed for each LOD, we must compute several additional parameters for measuring the error:

1. The surface deviation, s, between every convex patch c and the original mesh. This is an upper bound on the size of the geometric surface detail lost during the simplification and filtering process.
2. A support area, D, for every vertex in the hierarchy. This value is used to calculate contact area at run-time. The support area D is computed for every vertex $\mathbf{v}$ of the initial mesh M as the projected area onto the tangent plane of $\mathbf{v}$ of the faces incident to $\mathbf{v}$, such that they are within a distance tolerance from $\mathbf{v}$ along the direction of the normal $\mathbf{N}$ of $\mathbf{v}$, and their normal lies inside the normal cone of $\mathbf{N}$. When an edge $(\mathbf{v_1},\mathbf{v_2})$ is collapsed to a vertex $\mathbf{v_3}$, we assign to $\mathbf{v_3}$ the minimum of the two support areas of $\mathbf{v_1}$ and $\mathbf{v_2}$. We have typically used the same tolerance used in the contact queries (see Sec. 5) as the distance tolerance for this computation as well.
3. The maximum directed Hausdorff distance, h, computed for every convex piece C, from the descendant pieces of C.

The use of the surface deviation s, the support area D, and the resolution r of the LODs (whose computation is explained in Sec. 4.4) during the contact queries is described in Sec. 5.4. And the run-time use of the Hausdorff distance h is described in Sec. 5.3.

5 Contact Computation for Haptics

In this section, we describe how our collision detection algorithm uses the new multiresolution hierarchy described in Sec. 4 to compute contact response for haptic rendering. First, we describe the requirements of our collision detection system. Then, we present and analyze the data structures and algorithms. Finally, we show how to perform sensation preserving contact queries for force display.

5.1 Basic Haptic Rendering Framework

Our haptic rendering system uses a penalty-based force computation model, in which the amount of force displayed is proportional to the penetration depth or separation distance. Contact features within a given tolerance value are all considered as "contacts" for the purpose of force display. For more information about our haptic rendering framework, we refer readers to Appendix B on the conference proceedings CD.

We define the contact query between two objects A and B as $Q(A,B,\delta)$. From $Q(A,B,\delta)$, we obtain all local minima of the distace function between A and B that are closer than a distance tolerance δ, as well as the associated contact information (i.e. distance, contact normal, etc.). $Q(A,B,\delta)$ is performed by recursively traversing the bounding volume hierarchies (BVH) of A and B and performing "distance queries" for pairs of convex pieces. We define the distance query between two convex pieces $a \in A$ and $b \in B$, $q(a,b,\delta)$, as a boolean query that returns whether a and b are closer than δ.

5.2 The Bounding Volume Test Tree

We use the concept of the *Bounding Volume Test Tree (BVTT)* [Larsen et al. 2000] to describe the algorithm and data structures used in our collision detection system. A node ab in the BVTT encapsulates a pair of pieces $a \in A$ and $b \in B$, which might be tested with a query $q(a,b,\delta)$. Performing a contact query $Q(A,B,\delta)$ can be understood as descending along the BVTT as long as the distance query q returns "true". In the actual implementation, the BVTT is constructed dynamically while the contact query is performed. If the distance query result is "true" for a given pair, then the piece whose children have the lowest resolution is split. This splitting policy yields a BVTT where the levels of the tree are sorted according to their resolution, as shown in Fig. 7. Nodes of the BVTT at coarser resolution are closer to the root. This is a key issue for optimizing our sensation preserving haptic rendering, because we obtain a BVTT where LODs with lower resolution and larger error are stored closer to the root. Descending the BVTT has the effect of selecting finer LODs.

As pointed out in Sec. 4.4, the top levels of the BVHs are "free" LODs, which are not obtained using our simplification algorithm, but pairwise convex hull merging. Therefore, the top levels of the BVTT have no associated metric of resolution. The boundary between "free" and regular LODs is indicated in Fig. 8 by the line λ.

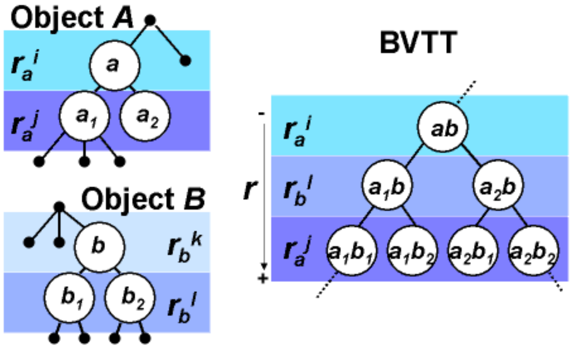

Figure 7: **Bounding Volume Test Tree.**

Instead of starting the contact query Q at the root of the BVTT every time, temporal coherence can be exploited using "generalized front tracking" [Ehmann and Lin 2001]. We store the "front" of the BVTT, $\mathcal{F}$, where the result of the distance query q switches from "true" to "false", as shown in Fig. 8. The front is recorded at the end of a contact query Q_i, and the next query Q_{i+1} proceeds by starting recursive distance queries q at every node in the front $\mathcal{F}$.

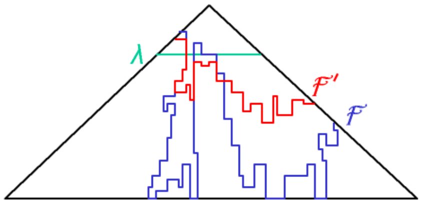

Figure 8: **Generalized Front Tracking of the BVTT.** *The front of BVTT for the original model, $\mathcal{F}$, is raised up to the new front $\mathcal{F}'$ using mesh simplification, since the contact queries can stop earlier using the sensation preserving selective refinement criterion. λ indicates the portion of the hierarchy constructed using the pairwise convex piece merging strategy, instead of mesh simplification, to form bounding volumes in the hierarchy.*

5.3 Distance Query for Convex Pieces

In a contact query Q using BVHs, we need to ensure that if the distance query q is "true" for any node of the BVTT, then it must be "true" for all its ancestors. To guarantee this result with our multiresolution hierarchy, given a distance tolerance δ for a contact query $Q(A,B,\delta)$, the distance tolerance δ_{ab} for a distance query $q(a,b,\delta_{ab})$ must be computed as:

$$\delta_{ab} = \delta + h(a^i,a) + h(b^j,b) \qquad (5)$$

where $h(a^i,a)$ and $h(b^j,b)$ are maximum directed Hausdorff distances from the descendant pieces of a and b to a and b respectively. As explained in Sec. 4.5, these Hausdorff distances are precomputed.

5.4 Sensation Preserving Selective Refinement

The time spent by a collision query Q depends directly on the number of nodes visited in the BVTT. Generalized front tracking considerably reduces the running time of Q when temporal coherence is high, which is the case in haptic rendering. Then, the time spent by Q is proportional to the size of the front $\mathscr{F}$. However, the cost is still $O(nm)$ in the worst case, where n and m are the number of convex pieces of the objects.

In our system, we further take advantage of the multiresolution hierarchies to accelerate the performance of the query. The core idea of our sensation preserving selective refinement is that the nodes of the BVTT are only refined if the missing detail is perceptible. Note that the selective refinement does not apply to the "free" levels of the BVTT. Those levels must always be refined if the distance query q returns "true".

As discussed in Sec. 3, the perceptibility of surface features depends on their size and the contact area. We have formalized this principle by devising a heuristic that assigns a functional ϕ to surface features which is averaged over the contact area.

Given a node ab of the BVTT for which the distance query result q is "true", we determine if the missing detail is perceptible by computing the functional of the missing detail and averaging it over the contact area of that node. For a convex piece a of the node ab, with resolution r_a and surface deviation from its descendent leaves s_a, we define the functional ϕ as:

$$\phi_a = \frac{s_a}{r_a^2} \tag{6}$$

This definition of the functional can be regarded as a measure of the maximum volume of features that have been culled away in the convex piece a.

The online computation of the contact area for a pair of convex pieces is too expensive, given the time constraints of haptic rendering. Therefore, we have estimated the contact area by selecting the maximum support areas of the contact primitives (i.e. vertex, edge or triangle). As explained in Sec. 4.5, the support area D is stored for all vertices in the hierarchy. For edge or triangle contact primitives, we interpolate the support areas of the end vertices, using the baricentric coordinates of the contact point.

Given functional values of ϕ_a and ϕ_b for the convex pieces a and b, as well as support areas D_a and D_b, we compute a weighted surface deviation, s^*_{ab}, as:

$$s^*_{ab} = \frac{\max(\phi_a, \phi_b)}{\max(D_a, D_b)} \tag{7}$$

Note that s^*_{ab} can be considered as the surface deviation weighted by a constant that depends both on the resolution and the contact area. If s^*_{ab} is above a threshold s_0, the node ab has to be refined. Otherwise, the missing detail is considered to be imperceptible. The selection of the value of s_0 is discussed in Sec. 6. As described in Sec. 4.4 and Sec. 4.5, the resolution r, the surface deviation s, and the support areas D are parameters computed as part of the preprocessing.

By using the described sensation preserving selective refinement of nodes of the BVTT, we achieve varying contact resolutions across the surfaces of the interacting objects, as shown in Fig. 1. In other words, every contact is treated independently, and its resolution is selected to cull away imperceptible local surface detail. As a consequence of the selective refinement, the active front of the BVTT, $\mathscr{F}'$, is above the original front $\mathscr{F}$ that separates nodes with "true" result for distance queries q from nodes with "false" result. The front does not need to reach the leaves of the BVTT as long as the missed detail is imperceptible, as depicted in Fig. 8. This approach results in a much faster processing of contact queries.

5.5 LOD Interpolation

A major issue in systems that use multiresolution hierarchies is the discontinuity that arises when the algorithm switches between different LODs. This problem is known as "popping" in multiresolution (visual) rendering. In haptic rendering its effects are discontinuities in the delivered force and torque, which are perceived by the user.

We have addressed the problems of discontinuities by interpolating contact information (e.g. contact normal and distance) from different LODs. When the sensation preserving selective refinement determines that no more refining is necessary, we perform a conservative refinement and compute contact information for the children of the current node of the BVTT. The contact information is interpolated between the two levels.

Naturally, LOD interpolation increases the number of nodes of the BVTT that are visited. However, for complex models and/or complex contact scenarios, the gain obtained from the selective refinement still makes sensation preserving simplification significantly outperform the exact technique, as presented in Sec. 6.

6 Implementation and Results

In this section we describe some of the models and experiments we have used to validate our sensation preserving simplification for haptic rendering.

6.1 System Demonstration

We have applied our sensation preserving simplification for haptic rendering on the models listed in Table 2. The complexity and surface detail of these models can be seen in Fig. 9.

Models	Lower Jaw	Upper Jaw	Ball Joint	Golf Club	Golf Ball
Orig. Tris	40180	47339	137060	104888	177876
Orig. Pcs	11323	14240	41913	27586	67704
Simp. Tris	386	1038	122	1468	826
Simp. Pcs	64	222	8	256	64
r_1	144.49	117.5	169.9	157.63	216.3
r_λ	12.23	19.21	6.75	8.31	7.16
Free LODs	6	8	3	8	6
LODs	15	15	17	16	18

Table 2: **Models and Associated Hierarchies.** *The number of triangles (Orig. Tris) and the number of convex pieces (Orig. Pcs) of the initial mesh of the models; the number of triangles (Simp. Tris) and the number of convex pieces (Simp. Pcs) of the coarsest LOD obtained through simplification; resolution (r_1 and r_λ) of the finest and coarsest LOD obtained through simplification; and "free" LODs and total number of LODs. The resolutions are computed for a radius of 1 for all the objects.*

As seen from the results in Table 2, we are able to simplify the models to LODs with only a couple hundred convex pieces or less. In other words, the sensation preserving selective refinement can be applied at earlier stages in the contact query, and this allows more aggressive culling of parts of the BVTT whenever the perceivable error is small.

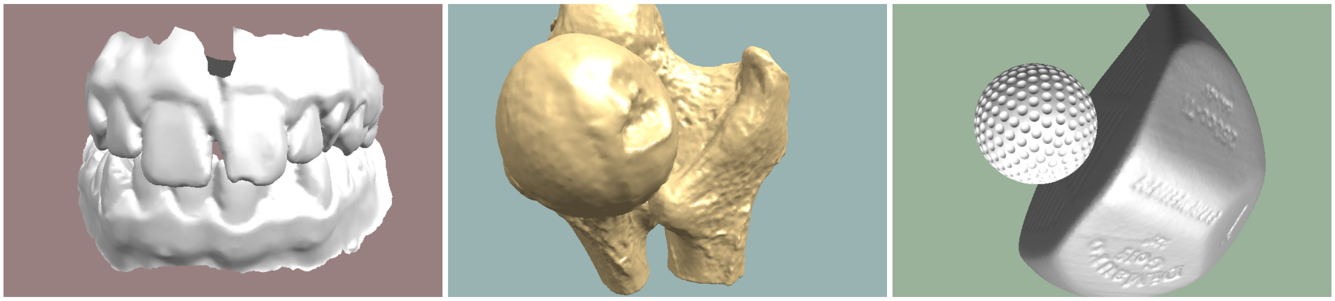

Figure 9: **Benchmark Models.** *From left to right, moving upper and lower jaws, interlocking ball joints and interacting golf club and ball.*

With the aforementioned models, we have performed the following proof-of-concept demonstrations:

- Moving upper and lower jaws.
- Golf club tapping a golf ball.
- Interlocking ball joints.

These demonstrations have been performed using our sensation preserving haptic rendering, a six-DOF *Phantom*^{TM} haptic device, a dual Pentium-4 2.4GHz processor PC with 2.0 GB of memory and a NVidia GeForce-4 graphics card, and Windows2000 OS. Our implementation, both for preprocessing and for the haptic rendering, has been developed using C++. For the force computation of the haptic rendering we have used penalty methods based on the contact tolerance δ [Kim et al. 2002]. We choose the value of δ so that the maximum force of the haptic device is exerted for a 0 contact distance with the optimal value of stiffness.

6.2 Studies on Perceivable Contact Information

The performance of the sensation preserving haptic rendering is heavily determined by the selection of the threshold of weighted surface deviation s_0. If the chosen value is too high, the perceived contact information will deviate too much from the exact contact information. On the other hand, if the value is too low and the simplified models used are moderately complex consisting of more than a thousand convex pieces, the contact query will no longer be executable at the required rate. This severely affects the realism of haptic perception.

We have designed a scenario where we could test the fidelity of the sensation preserving selective refinement. In this scenario, users can touch the model of the golf ball with an ellipsoid. The ellipsoid has varying curvature, implying the existence of a wide range of contact scenarios, where the selective refinement will stop at varying LODs.

12 users experimented with this scenario and reported that the perception of contact information hardly varied for values of s_0 in the range between 0.025 and 0.05 times the radius of the models. (For readers interested in the detail of experimental data, please refer to Appendix C on the conference proceedings CD).

6.3 Performance Demonstration

Based on the value of s_0 obtained from the studies, we have successfully applied our algorithm to haptic rendering of object-object interaction on the benchmarks listed in Sec 6.1. We have also performed an analysis on contact forces and running time for the demonstrations previously mentioned. We have compared force profiles and statistics of the contact query of interactive haptic demonstrations with offline executions, using smaller error tolerances and an exact method. By exact, we mean that the distance computation for force display is accurate [Ehmann and Lin 2001]. In particular, Fig. 10 shows the contact profile, including the force profile, the query time and the size of the front of the BVTT, for 200 frames of the moving jaws simulation. Fig. 11 shows the contact profile for 300 frames of the simulation on the golf scene. The contact profile of interlocking joints is quite similar to that of the interacting golf club and golf ball, thus omitted here.

For both scenarios, the simulation with $s_0 < 5\%$ of the radii of the models has been performed in real time, using a haptic device to control the motion of the upper jaw and the golf club respectively, and to display the contact forces to the user. The trajectories of the upper jaw and the golf club are recorded and played back to perform the rest of the simulations offline, since the exact method was too slow to be used to keep up with the force update. As shown in the figure, we observed a gain of two orders of magnitude in the query time between the interactive haptic rendering using our approach and the exact offline simulation. Note that the spikes in the contact query time present in Fig. 10 and Fig. 11 result from lack of coherence in the traversal of the BVTT. As reflected in the graphs, the query time is more susceptible to lack of coherence when the error tolerance is lower.

Using our approach, the force profiles of simulations with varying error tolerances less than 5% of the radii of the models exhibit similar and sometimes nearly identical patterns as that of the original models. This resemblance validates our hypothesis on the haptic perception of contacts, inferred from human tactual perception.

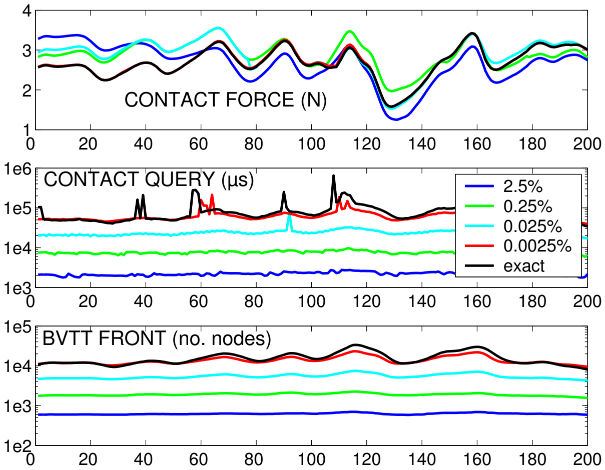

Figure 10: **Contact Profile for Moving Jaws.** *Top: The profiles of the contact forces displayed using simplification, with varying error tolerances up to* 2.5% *of the radii of the jaws, all show very similar patterns. This similarity implies that the sensations of shape provided to the user are nearly identical. Middle: A log plot of contact query time using simplification with various error tolerances shows up to two orders of performance improvement. Bottom: The number of nodes in the front of the BVTT is also reduced by more than a factor of 10.*

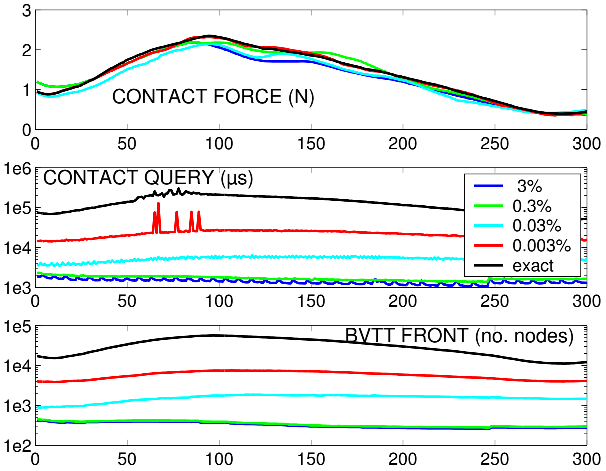

Figure 11: **Contact Profile for Golf Scene.** *Top: The profiles of the contact forces displayed using simplification, with varying error tolerances up to 3% of the radius of the ball, show nearly identical patterns. Middle: A log plot of contact query time using simplification with various error tolerances shows more than two orders of performance improvement. Bottom: The number of nodes in the front of the BVTT is reduced by nearly a factor of 100.*

7 Discussion and Analysis

In this section, we compare our approach with previous work in related areas and analyze the various algorithmic and performance issues.

7.1 Comparison with Related Work

Mesh Decimation and Filtering: The construction of our multiresolution hierarchy can be compared with both mesh decimation techniques and mesh filtering techniques. Representations obtained through these techniques might present better results than our hierarchies in certain aspects.

LODs created using our filtered edge collapse operation will have a larger surface deviation than LODs of traditional mesh decimation. This deviation inevitably results from combining decimation and filtering. In our framework, the detail at high resolution is filtered independently of its magnitude, while mesh decimation techniques will preserve detail to minimize the surface deviation. The elimination of the detail has beneficial consequences in the creation of the BVH and does not reflect on the output quality of the haptic rendering, since the filtered detail is quantified and taken into account in the sensation preserving refinement. Besides, multiresolution representations obtained through mesh decimation techniques are not valid by themselves to perform efficient contact queries.

Representations obtained through filtering appear smoother than our representations. The reduction in visual smoothness occurs because we use fewer samples (i.e. vertices) to represent meshes with the same frequency content. This approach is advantageous for our application, because it accelerates the contact queries. In addition, we have also presented a definition that allows comparing the resolution of the detail of the objects in contact.

Contact Queries for Haptic Rendering: As mentioned earlier, the running time of any contact query algorithm depends on both the input and output size of the problem. Given two polyhedra, characterized by their combinatorial complexity of n and m polygons, the contact query algorithm can have an output size and a run-time complexity as high as $O(nm)$.

The discretized approximation presented by McNeely et al. [1999] can avoid direct dependency on the input size of the problem by limiting the number of points sampled and the number of voxels generated. However, its performance on complex contact scenarios with many collisions is unknown. Both approaches by Gregory et al. [2000] and Kim et al. [2002] are limited to relatively simple models or modestly complex contact scenarios and do not scale well to highly complex object-object interaction.

In contrast, our approach, by reducing the combinatorial complexity of the input based on the contact configuration at each local neighborhood of (potential) collisions, automatically decreases the output size as well. In addition, its selection of LODs is contact-dependent to minimize the perceived difference in force display, while maximizing the amount of simplification and performance gain possible. This method is perhaps the first "contact-dependent simplification" algorithm for collision queries as well.

7.2 Generalization of the Algorithmic Framework

Due to the hard time constraints of haptic rendering, we have chosen a collision detection algorithm using BVHs of convex hulls and automatic convex surface decomposition. The choice of collision detection algorithm imposes convexity constraints on the simplification process and the hierarchy construction

These constraints are rather specific. However, the algorithmic framework for generating the multiresolution hierarchy for sensation preserving contact queries that we have developed and presented in this paper is general and applicable to other collision detection algorithms.

Furthermore, although we focus on contact determination for haptic rendering in this paper, our approach for sensation preserving simplification can also be applied to other types of proximity queries, such as penetration depth estimation. Our approach can be generalized to multiresolution collision detection by automatically identifying superfluous proximity information and thus cleverly selecting the appropriate resolutions for performing the queries at different locations across the objects' surfaces. This key concept can significantly accelerate the performance of any proximity query algorithm, as we have demonstrated in this paper.

A further analysis of the applicability of sensation preserving simplification to multiresolution collision detection for rigid body simulation has been conducted [Otaduy and Lin 2003]. More study is needed on the relationship between our sensation preserving error metrics and contact force models of rigid body simulations, but the preliminary results are promising.

7.3 Integration with Graphic Rendering

The LODs selected for haptic rendering are decoupled from the representation of the objects used for visual rendering. This difference in representations can potentially lead to some inconsistency between visual and haptic display, such as the existence of visual gaps when the displayed forces indicate that the objects are in contact. Future investigation is required for a better integration of multisensory cues in a multimedia environment.

7.4 Other Limitations

Our approach can handle triangular meshes with two-manifolds and boundaries. The current implementation is limited to polygonal models with connectivity information. As with all simplification algorithms that generate levels of detail offline, our approach has the similar memory requirement.

8 Summary and Future Work

We have presented a novel sensation preserving simplification to accelerate collision queries for force display of complex object-object interaction. The resulting multiresolution hierarchy constructed with a formal definition of resolution enables us to compute contact information at varying resolutions independently for different locations across the object surfaces. By selecting the most aggressive simplification possible on the fly based on the contact configuration, this approach considerably improves the run-time performance of contact queries. It makes haptic rendering of the interaction between highly complex models possible, while producing only relatively imperceptible changes in the force display. Our approach can also be easily extended to perform time-critical haptic rendering while optimizing the fidelity of the force display, using the technique described in [Otaduy and Lin 2003].

This new ability to perform force display of complex 3D object interaction enables many exciting applications, where haptic rendering of point-object interaction is insufficient. In addition to further optimizing and increasing the performance of sensation preserving haptic rendering, this research may be extended in several possible directions. These include haptic display of friction and textures exploiting our current framework, applications of 6-DOF haptic rendering to scientific visualization, engineering prototyping, and medical training, as well as formal user studies and task performance analysis.

Acknowledgments

This research is supported in part by a fellowship of the Government of the Basque Country, National Science Foundation, Office of Naval Research, U.S. Army Research Office, and Intel Corporation. We would like to thank Stephen Ehmann and Young Kim for their help on integrating SWIFT++ and DEEP with our 6-DOF haptic rendering framework. We are also grateful to Dinesh Manocha, Russell Taylor, Mark Foskey, and the anonymous reviewers for their feedback on the earlier drafts of this paper.

References

BROOKS, JR., F. P., OUH-YOUNG, M., BATTER, J. J., AND KILPATRICK, P. J. 1990. Project GROPE — Haptic displays for scientific visualization. In *Computer Graphics (SIGGRAPH '90 Proceedings)*, F. Baskett, Ed., vol. 24, 177–185.

CHAZELLE, B., DOBKIN, D., SHOURABOURA, N., AND TAL, A. 1997. Strategies for polyhedral surface decomposition: An experimental study. *Computational Geometry: Theory and Applications 7*, 327–342.

EHMANN, S., AND LIN, M. C. 2000. Accelerated proximity queries between convex polyhedra using multi-level voronoi marching. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

EHMANN, S., AND LIN, M. C. 2001. Accurate and fast proximity queries between polyhedra using convex surface decomposition. *Computer Graphics Forum (Proc. of Eurographics'2001) 20*, 3.

EL-SANA, J., AND VARSHNEY, A. 2000. Continuously-adaptive haptic rendering. *Virtual Environments 2000*, pp. 135–144.

GARLAND, M., AND HECKBERT, P. S. 1997. Surface simplification using quadric error metrics. In *Proc. of ACM SIGGRAPH*, 209–216.

GOTTSCHALK, S., LIN, M., AND MANOCHA, D. 1996. OBB-Tree: A hierarchical structure for rapid interference detection. *Proc. of ACM SIGGRAPH*, pp. 171–180.

GREGORY, A., MASCARENHAS, A., EHMANN, S., LIN, M. C., AND MANOCHA, D. 2000. 6-dof haptic display of polygonal models. *Proc. of IEEE Visualization Conference*.

GUIBAS, L., HSU, D., AND ZHANG, L. 1999. *H-Walk*: Hierarchical distance computation for moving convex bodies. *Proc. of ACM Symposium on Computational Geometry*.

GUSKOV, I., SWELDENS, W., AND SCHRODER, P. 1999. Multiresolution signal processing for meshes. *Proc. of ACM SIGGRAPH*, pp. 325 – 334.

HOFF, K., ZAFERAKIS, A., LIN, M., AND MANOCHA, D. 2001. Fast and simple geometric proximity queries using graphics hardware. *Proc. of ACM Symposium on Interactive 3D Graphics*.

HOLLERBACH, J., COHEN, E., THOMPSON, W., FREIER, R., JOHNSON, D., NAHVI, A., NELSON, D., AND II, T. T. 1997. Haptic interfacing for virtual prototyping of mechanical CAD designs. *CDROM Proc. of ASME Design for Manufacturing Symposium*.

HUBBARD, P. 1994. *Collision Detection for Interactive Graphics Applications*. PhD thesis, Brown University.

KIM, Y., OTADUY, M., LIN, M., AND MANOCHA, D. 2002. 6-dof haptic display using localized contact computations. *Proc. of Haptics Symposium*.

KLATZKY, R., AND LEDERMAN, S. 1995. Identifying objects from a haptic glance. *Perception and Psychophysics 57*, pp. 1111–1123.

KLOSOWSKI, J., HELD, M., MITCHELL, J., SOWIZRAL, H., AND ZIKAN, K. 1998. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE Trans. on Visualization and Computer Graphics 4*, 1, 21–37.

LARSEN, E., GOTTSCHALK, S., LIN, M., AND MANOCHA, D. 2000. Distance queries with rectangular swept sphere volumes. *Proc. of IEEE Int. Conference on Robotics and Automation*.

LIN, M., AND GOTTSCHALK, S. 1998. Collision detection between geometric models: A survey. *Proc. of IMA Conference on Mathematics of Surfaces*.

LOMBARDO, J. C., CANI, M.-P., AND NEYRET, F. 1999. Real-time collision detection for virtual surgery. *Proc. of Computer Animation*.

LUEBKE, D., AND HALLEN, B. 2001. Perceptually driven simplification for interactive rendering. *Rendering Techniques; Springer-Verlag*.

LUEBKE, D., REDDY, M., COHEN, J., VARSHNEY, A., WATSON, B., AND HUEBNER, R. 2002. *Level of Detail for 3D Graphics*. Morgan-Kaufmann.

MARK, W., RANDOLPH, S., FINCH, M., VAN VERTH, J., AND TAYLOR II, R. M. 1996. Adding force feedback to graphics systems: Issues and solutions. *Proc. of ACM SIGGRAPH*, 447–452.

MCNEELY, W., PUTERBAUGH, K., AND TROY, J. 1999. Six degree-of-freedom haptic rendering using voxel sampling. *Proc. of ACM SIGGRAPH*, 401–408.

OKAMURA, A., AND CUTKOSKY, M. 1999. Haptic exploration of fine surface features. *Proc. of IEEE Int. Conf. on Robotics and Automation*, pp. 2930–2936.

O'SULLIVAN, C., AND DINGLIANA, C. 2001. Collisions and perception. *ACM Trans. on Graphics 20*, 3, pp. 151–168.

OTADUY, M. A., AND LIN, M. C. 2003. CLODs: Dual hierarchies for multiresolution collision detection. *UNC Technical Report TR03-013*.

PAI, D. K., AND REISSEL, L. M. 1997. Haptic interaction with multiresolution image curves. *Computer and Graphics 21*, 405–411.

REDON, S., KHEDDAR, A., AND COQUILLART, S. 2002. Fast continuous collision detection between rigid bodies. *Proc. of Eurographics (Computer Graphics Forum)*.

RUSPINI, D., KOLAROV, K., AND KHATIB, O. 1997. The haptic display of complex graphical environments. *Proc. of ACM SIGGRAPH*, 345–352.

SALISBURY, J. K. 1999. Making graphics physically tangible. *Communications of the ACM 42*, 8.

TAUBIN, G. 1995. A signal processing approach to fair surface design. In *Proc. of ACM SIGGRAPH*, 351–358.

APPENDIX A: Pseudo Code for Multiresolution Hierarchy Generation

```
Compute surface convex decomposition
Dump initial LOD
$n$ = number of convex pieces
Compute resolution of edges
Initialize edges as valid
Create priority queue
while Valid(Top(queue)),
  if FilteredEdgeCollapse(Top(queue)) then
    PopTop(queue)
    Recompute resolution of affected edges
    Reset affected edges as valid
    Update priority of affected edges
    Attempt merge of convex pieces
  else
    Set Top(queue) as invalid
    Update priority of Top(queue)
  endif
  if Number of pieces $\leq n/2$ then
    Dump new LOD
    $n$ = number of convex pieces
  endif
endwhile
while Number of pieces $> 1$,
  Binary merge of pieces
endwhile
```

ALGORITHM 0.1: Creation of the Hierarchy

APPENDIX B: Overview of the Haptic Rendering Framework

In this appendix, we give an overview of our six-degree-of-freedom haptic rendering framework for displaying force and torque between two objects in contact. We compute the displayed force based on the following steps:

1. Perform a contact query between two objects A and B, collecting the set S of nodes of the front of the BVTT that are inside a distance tolerance δ, at the appropriate resolution.
2. For all nodes $ab \in S$, compute the contact information:
   (a) If the pair ab is disjoint, compute the distance, the contact normal and the closest points and features (i.e. vertex, edge or face).
   (b) If the pair ab is intersecting, compute the penetration depth, the penetration direction, and penetration features [1].
3. Cluster all contacts based on their proximity.
4. Compute a representative contact for each cluster, averaging contact information weighted by the contact distance.
5. Compute a penalty-based restoring force at the representative contact of each cluster.
6. Apply the net force and torque to the haptic probe.

APPENDIX C: Studies on Perceivable Contact Details

In this appendix we briefly describe an informal user study that has been conducted to test the fidelity of the sensation preserving simplification for haptic rendering, and also to help us identify what are the error tolerances for which the missing surface detail is not perceptible to the users.

The scenario of our experiment consists of a golf ball (please refer to the paper for statistics of the model) that is explored with an ellipsoid as shown in Fig. 1. The ellipsoid consists of 2000 triangles and one single convex piece. For simplicity, we have only applied the simplification to the golf ball and left the ellipsoid invariant. Thus, the fidelity of the sensation preserving simplification relies on the adequacy of the resolution of the golf ball that is selected at each contact.

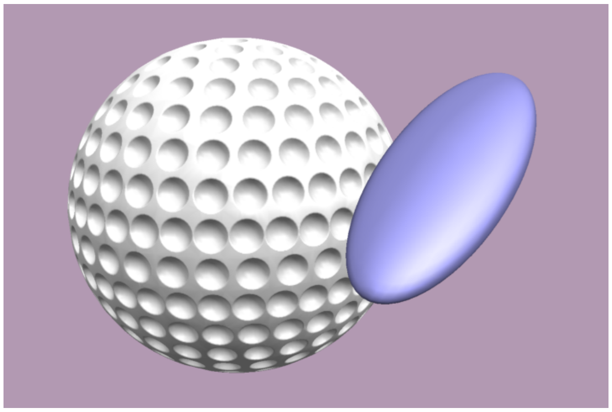

Figure 1: **Scenario of the User Study.**

12 users have experimented with this scenario. They were asked to identify at what value of the threshold s_0 of the sensation preserving selective refinement they started perceiving a deviation in the perception of the surface detail of the golf ball. The values of s_0 were in the range from 0.05% to 20% of the radius of the ball. In Table 1 we indicate how many users picked each threshold value. The users also reported that the main characteristic that they explored was the perceptibility of the dimples of the golf ball.

s_0	$\geq$ 10%	5%	2.5%	1%	$\leq$ 0.5%
no. users	0	4	7	1	0

Table 1: **Results of the User Study.**

Based on the results from this informal user study, we selected values of s_0 in the range of 2.5% to 5% of the radii of the models for the demonstrations presented in the paper.

[1] By the penetration features, we mean a pair of features on both objects whose supporting planes realize the penetration depth.