Introduction to Haptic Rendering

Miguel A. Otaduy   Ming C. Lin

Department of Computer Science
University of North Carolina at Chapel Hill

1 Why Haptic Rendering?

For a long time, human beings have dreamed of a virtual world where it is possible to interact with synthetic entities as if they were real. To date, the advances in computer graphics allow us to *see* virtual objects and avatars, to *hear* them, to *move* them, and to *touch* them. It has been shown that the ability to touch virtual objects increases the sense of presence in virtual environments [Insko 2001].

Haptic rendering offers important applicability in engineering and medical training tasks. In this chapter we introduce the concept of haptic rendering, and we briefly describe some of the basic techniques and applications. In the first section we define some terminology, discuss the evolution of the research in haptic rendering, and introduce practical applications.

1.1 Definitions

The term *haptic* (from the Greek *haptesthai*, meaning "to touch") is the adjective used to describe something relating to or based on the sense of touch. Haptic is to touching as visual is to seeing and as auditory is to hearing [Fisher et al. 2004].

As described by Klatzky and Lederman [Klatzky and Lederman 2003], touch is one of the main avenues of sensation, and it can be divided into cutaneous, kinesthetic, and haptic systems, based on the underlying neural inputs. The cutaneous system employs receptors embedded in the skin, while the kinesthetic system employs receptors located in muscles, tendons, and joints. The haptic sensory system employs both cutaneous and kinesthetic receptors, but it differs in the sense that it is associated with an active procedure. Touch becomes active when the sensory inputs are combined with controlled body motion. For example, cutaneous touch becomes active when we explore a surface or grasp an object, while kinesthetic touch becomes active when we manipulate an object and touch other objects with it.

Haptic rendering is defined as the process of computing and generating forces in response to user interactions with virtual objects [Salisbury et al. 1995]. Several haptic rendering algorithms consider the paradigm of touching virtual objects with a single contact point. Rendering algorithms that follow this description are called 3-DoF haptic rendering algorithms, because a point in 3D has only three DoFs. Other haptic rendering algorithms deal with the problem of rendering the forces and torques arising from the interaction of two virtual objects. This problem is called 6-DoF haptic rendering, because the grasped object has six DoFs (position and orientation in 3D), and the haptic feedback comprises 3D force and torque. When we eat with a fork, write with a pen, or open a lock with a key, we are moving an object in 3D, and we feel the interaction with other objects. This is, in essence, 6-DoF object manipulation with force-and-torque feedback. Fig. 1 shows an example of a user experiencing haptic rendering. When we manipulate an object and touch other objects with it, we perceive cutaneous feedback as the result of grasping, and kinesthetic feedback as the result of contact between objects.

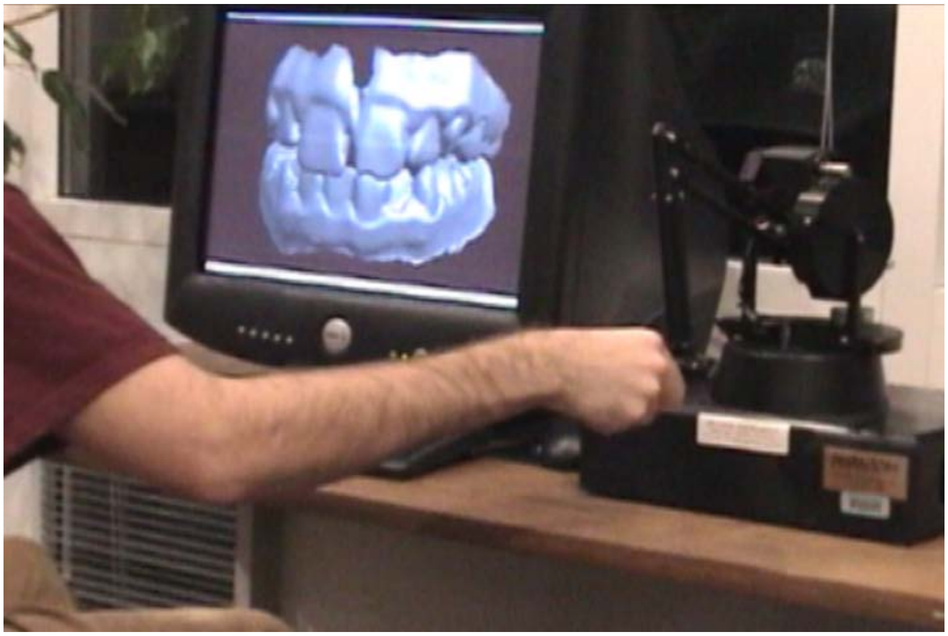

Figure 1: **Example of Haptic Rendering.** *A person manipulates a virtual jaw using a haptic device (shown on the right of the image), and the interaction between jaws is displayed both visually and haptically.*

1.2 From Telerobotics to Haptic Rendering

In 1965, Ivan Sutherland [Sutherland 1965] proposed a multimodal display that would incorporate haptic feedback into the interaction with virtual worlds. Before that, haptic feedback had already been used mainly in two applications: flight simulators and master-slave robotic teleoperation. The early teleoperator systems had mechanical linkages between the master and the slave. But, in 1954, Goertz and Thompson [Goertz and Thompson 1954] developed an electrical servomechanism that received feedback signals from sensors mounted on the slave and applied forces to the master, thus producing haptic feedback.

From there, haptic interfaces evolved in multiple directions, but there were two major breakthroughs. The first breakthrough was the idea of substituting the slave robot by a simulated system, in which forces were computed using physically based simulations. The GROPE project at the University of North Carolina at Chapel Hill [Brooks, Jr. et al. 1990], lasting from 1967 to 1990, was the first one to address the synthesis of force feedback from simulated interactions. In particular, the aim of the project was to perform real-time simulation of 3D molecular-docking forces. The second breakthrough was the advent of computer-based Cartesian control for teleoperator systems [Bejczy and Salisbury 1980], enabling a separation of the kinematic configurations of the master and the slave. Later, Cartesian control was applied to the manipulation of simulated slave robots [Kim and Bejczy 1991].

Those first haptic systems were able to simulate the interaction of simple virtual objects only. Perhaps the first project to target computation of forces in the interaction with objects with rich geometric information was Minsky's *Sandpaper* [Minsky et al. 1990]. Minsky et al. developed a planar force feedback system that allowed the exploration of textures. A few years after Minsky's work, Zilles and Salisbury presented an algorithm for 3-DoF haptic rendering of polygonal models [Zilles and Salisbury 1995]. Almost in parallel with Zilles and Salisbury's work, Massie and Salisbury [Massie and Salisbury 1994] designed the PHANToM, a stylus-based haptic interface that was later commercialized and has become one of the most commonly used force-feedback devices. But in the late '90s, research in haptic rendering revived one of the problems that first inspired virtual force feedback: 6-DoF haptic rendering or, in other words, grasping of a virtual object and synthesis of kinesthetic feedback of the interaction between this object and its environment.

Research in the field of haptics in the last 35 years has covered many more areas than what we have summarized here. [Burdea 1996] gives a general survey of the field of haptics and [McLaughlin et al. 2002] discuss current research topics.

1.3 Haptic Rendering for Virtual Manipulation

Certain professional activities, such as training for high-risk operations or pre-production prototype testing, can benefit greatly from simulated reproductions. The fidelity of the simulated reproductions depends, among other factors, on the similarity of the behaviors of real and virtual objects. In the real world, solid objects cannot interpenetrate. Contact forces can be interpreted mathematically as constraint forces imposed by penetration constraints. However, unless penetration constraints are explicitly imposed, virtual objects are free to penetrate each other in virtual environments. Indeed, one of the most disconcerting experiences in virtual environments is to pass through virtual objects [Insko et al. 2001; Slater and Usoh 1993]. Virtual environments require the simulation of non-penetrating rigid body dynamics, and this problem has been extensively explored in the robotics and computer graphics literature [Baraff 1992; Mirtich 1996].

It has been shown that being able to touch physical replicas of virtual objects (a technique known as *passive haptics* [Insko 2001]) increases the sense of presence in virtual environments. This conclusion can probably be generalized to the case of synthetic cutaneous feedback of the interaction with virtual objects. As reported by Brooks et al. [Brooks, Jr. et al. 1990], kinesthetic feedback radically improved situation awareness in virtual 3D molecular docking. Kinesthetic feedback has proved to enhance task performance in applications such as telerobotic object assembly [Hill and Salisbury 1977], virtual object assembly [Unger et al. 2002], and virtual molecular docking [Ouh-Young 1990]. In particular, task completion time is shorter with kinesthetic feedback in docking operations but not in prepositioning operations.

To summarize, haptic rendering is especially useful in particular examples of training for high-risk operations or pre-production prototype testing activities that involve intensive object manipulation and interaction with the environment. Such examples include minimally invasive or endoscopic surgery [Edmond et al. 1997; Hayward et al. 1998] and virtual prototyping for assembly and maintainability assessment [McNeely et al. 1999; Chen 1999; Andriot 2002; Wan and McNeely 2003]. Force feedback becomes particularly important and useful in situations with limited visual feedback.

1.4 3-DoF and 6-DoF Haptic Rendering

Much of the existing work in haptic rendering has focused on 3-DoF haptic rendering [Zilles and Salisbury 1995; Ruspini et al. 1997; Thompson et al. 1997; Gregory et al. 1999; Ho et al. 1999]. Given a virtual object A and the 3D position of a point $\mathbf{p}$ governed by an input device, 3-DoF haptic rendering can be summarized as finding a contact point $\mathbf{p}'$ constrained to the surface of A. The contact force will be computed as a function of $\mathbf{p}$ and $\mathbf{p}'$. In a dynamic setting, and assuming that A is a polyhedron with n triangles, the problem of finding $\mathbf{p}'$ has an $O(n)$ worst-case complexity. Using spatial partitioning strategies and exploiting motion coherence, however, the complexity becomes $O(1)$ in many practical situations [Gregory et al. 1999].

This reduced complexity has made 3-DoF haptic rendering an attractive solution for many applications with virtual haptic feedback, such as: sculpting and deformation [Dachille et al. 1999; Gregory et al. 2000a; McDonnell et al. 2001], painting [Johnson et al. 1999; Gregory et al. 2000a; Foskey et al. 2002], volume visualization [Avila and Sobierajski 1996], nanomanipulation [Taylor et al. 1993], and training for diverse surgical operations [Kuhnapfel et al. 1997; Gibson et al. 1997]. In each of these applications, the interaction between the subject and the virtual objects is sufficiently captured by a point-surface contact model.

In 6-DoF manipulation and exploration, however, when a subject grasps an object and touches other objects in the environment, the interaction generally cannot be modeled by a point-surface contact. One reason is the existence of multiple contacts that impose multiple simultaneous non-penetration constraints on the grasped object. In a simple 6-DoF manipulation example, such as the insertion of a peg in a hole, the grasped object (i.e., the peg) collides at multiple points with the rest of the scene (i.e., the walls of the hole and the surrounding surface). This contact configuration cannot be modeled as a point-object contact. Another reason is that the grasped object presents six DoFs, 3D translation and rotation, as opposed to the three DoFs of a point. The feasible trajectories of the peg are embedded in a 6-dimensional space with translational and rotational constraints, that cannot be captured with three DoFs.

Note that some cases of object-object interaction have been modeled in practice by ray-surface contact [Basdogan et al. 1997]. In particular, several surgical procedures are performed with 4-DoF tools (e.g., laparoscopy), and this constraint has been exploited in training simulators with haptic feedback [Çavuşoğlu et al. 2002]. Nevertheless, these approximations are valid only in a limited number of situations and cannot capture full 6-DoF object manipulation.

2 The Challenges

Haptic rendering is in essence an interactive activity, and its realization is mostly handicapped by two conflicting challenges: high required update rates and the computational cost. In this section we outline the computational pipeline of haptic rendering, and we discuss associated challenges.

2.1 Haptic Rendering Pipeline

Haptic rendering comprises two main tasks. One of them is the computation of the position and/or orientation of the virtual probe grasped by the user. The other one is the computation of contact force and/or torque that are fed back to the user. The existing methods for haptic rendering can be classified into two large groups based on their overall pipelines.

In *direct rendering* methods [Nelson et al. 1999; Gregory et al. 2000b; Kim et al. 2003; Johnson and Willemsen 2003; Johnson and Willemsen 2004], the position and/or orientation of the haptic device are applied directly to the grasped probe. Collision detection is performed between the grasped probe and the virtual objects, and collision response is applied to the grasped probe as a function of object separation or penetration depth. The resulting contact force and/or torque are directly fed back to the user.

In *virtual coupling* methods [Chang and Colgate 1997; Berkelman 1999; McNeely et al. 1999; Ruspini and Khatib 2000; Wan and McNeely 2003], the position and/or orientation of the haptic device are set as goals for the grasped probe, and a virtual viscoelastic coupling [Colgate et al. 1995] produces a force that attracts the grasped probe to its goals. Collision detection and response are performed between the grasped probe and the virtual objects. The coupling force and/or torque are combined with the collision response in order to compute the position and/or orientation of the grasped probe. The same coupling force and/or torque are fed back to the user.

In Sec. 8, I describe the different existing methods for 6-DoF haptic rendering in more detail, and I discuss their advantages and disadvantages. Also, as explained in more detail in Sec. 4, there are two major types of haptic devices, and for each type of device the rendering pipeline presents slight variations. Impedance-type devices read the position and orientation of the handle of the device and control the force and torque applied to the user. Admittance-type devices read the force and torque applied by the user and control the position and orientation of the handle of the device.

2.2 Force Update Rate

The ultimate goal of haptic rendering is to provide force feedback to the user. This goal is achieved by controlling the handle of the haptic device, which is in fact the end-effector of a robotic manipulator. When the user holds the handle, he or she experiences kinesthetic feedback. The entire haptic rendering system is regarded as a mechanical impedance that sets a transformation between the position and velocity of the handle of the device and the applied force.

The quality of haptic rendering can be measured in terms of the dynamic range of impedances that can be simulated in a stable manner [Colgate and Brown 1994]. When the user moves the haptic device in free space, the perceived impedance should be very low (i.e., small force), and when the grasped virtual object touches other rigid objects, the perceived impedance should be high (i.e., high stiffness and/or damping of the constraint). The quality of haptic rendering can also be measured in terms of the responsiveness of the simulation [Brooks, Jr. et al. 1990; Berkelman 1999]. In free-space motion the grasped probe should respond quickly to the motion of the user. Similarly, when the grasped probe collides with a virtual wall, the user should stop quickly, in response to the motion constraint.

With impedance-type devices, virtual walls are implemented as large stiffness values in the simulation. In haptic rendering, the user is part of a closed-loop sampled dynamic system [Colgate and Schenkel 1994], along with the device and the virtual environment, and the existence of sampling and latency phenomena can induce unstable behavior under large stiffness values. System instability is directly perceived by the user in the form of disturbing oscillations. A key factor for achieving a high dynamic range of impedances (i.e., stiff virtual walls) while ensuring stable rendering is the computation of feedback forces at a high update rate [Colgate and Schenkel 1994; Colgate and Brown 1994]. Brooks et al. [Brooks, Jr. et al. 1990] reported that, in the rendering of textured surfaces, users were able to perceive performance differences at force update rates between 500Hz and 1kHz.

A more detailed description of the stability issues involved in the synthesis of force feedback, and a description of related work, are given in Sec. 4. Although here we have focused on impedance-type haptic devices, similar conclusions can be drawn for admittance-type devices (See [Adams and Hannaford 1998] and Sec. 4).

2.3 Contact Determination

The computation of non-penetrating rigid-body dynamics of the grasped probe and, ultimately, synthesis of haptic feedback require a model of collision response. Forces between the virtual objects must be computed from contact information. Determining whether two virtual objects collide (i.e., intersect) is not enough, and additional information, such as penetration distance, contact points, contact normals, and so forth, need to be computed. Contact determination describes the operation of obtaining the contact information necessary for collision response [Baraff 1992].

For two interacting virtual objects, collision response can be computed as a function of object separation, with worst-case cost $O(mn)$, or penetration depth, with a complexity bound of $\Omega(m^3n^3)$. But collision response can also be applied at multiple *contacts* simultaneously. Given two objects A and B with m and n triangles respectively, contacts can be defined as pairs of intersecting triangles or pairs of triangles inside a distance tolerance. The number of pairs of intersecting triangles is $O(mn)$ in worst-case pathological cases, and the number of pairs of triangles inside a tolerance can be $O(mn)$ in practical cases. In Sec. 5, we discuss in more detail existing techniques for determining the contact information.

The cost of contact determination depends largely on factors such as the convexity of the interacting objects or the contact configuration. There is no direct connection between the polygonal complexity of the objects and the cost of contact determination but, as a reference, existing exact collision detection methods can barely execute contact queries for force feedback between pairs of objects with $1,000$ triangles in complex contact scenarios [Kim et al. 2003] at force update rates of 1kHz.

Contact determination becomes particularly expensive in the interaction between textured surfaces. Studies have been done on the highest texture resolution that can be perceived through cutaneous touch, but there are no clear results regarding the highest resolution that can be perceived kinesthetically through an intermediate object. It is known that, in the latter case, texture-induced roughness perception is encoded in vibratory motion [Klatzky and Lederman 2002]. Psychophysics researchers report that 1mm textures are clearly perceivable, and perceived roughness appears to be even greater with finer textures [Lederman et al. 2000]. Based on Shannon's sampling theorem, a 10cm $\times$ 10cm plate with a sinusoidal texture of 1mm in orthogonal directions is barely correctly sampled with $40,000$ vertices. This measure gives an idea of the triangulation density required for capturing texture information of complex textured objects. Note that the triangulation density may grow by orders of magnitude if the textures are not sinusoidal and/or if information about normals and curvatures is also needed.

3 Psychophysics of Haptics

In the design of contact determination algorithms for haptic rendering, it is crucial to understand the psychophysics of touch and to account for perceptual factors. The structure and behavior of human touch have been studied extensively in the field of psychology. The topics analyzed by researchers include characterization of sensory phenomena as well as cognitive and memory processes.

Haptic perception of physical properties includes a first step of stimulus registration and communication to the thalamus, followed by a second step of higher-level processing. Perceptual measures can be originated by individual mechanoreceptors but also by the integration of inputs from populations of different sensory units [Klatzky and Lederman 2003]. Klatzky and Lederman [Klatzky and Lederman 2003] discuss object and surface properties that are perceived through the sense of touch (e.g., texture, hardness, and weight) and divide them between geometric and material properties. They also analyze active exploratory procedures (e.g., lateral motion, pressure, or unsupported holding) typically conducted by subjects in order to capture information about the different properties.

Knowing the exploratory procedure(s) associated with a particular object or surface property, researchers have studied the influence of various parameters on the accuracy and magnitude of sensory outputs. Perceptual studies on tactile feature detection and identification, as well as studies on texture or roughness perception are of particular interest for haptic rendering. In this section we summarize existing research on perception of surface features and perception of roughness, and then we discuss issues associated with the interaction of visual and haptic modalities.

3.1 Perception of Surface Features

Klatzky and Lederman describe two different exploratory procedures followed by subjects in order to capture shape attributes and identify features and objects. In *haptic glance* [Klatzky and Lederman 1995], subjects extract information from a brief haptic exposure of the object surface. Then they perform higher-level processing for determining the identity of the object or other attributes. In *contour following* [Klatzky and Lederman 2003], subjects create a spatiotemporal map of surface attributes, such as curvature, that serves as the pattern for feature identification. Contact determination algorithms attempt to describe the geometric interaction between virtual objects. The instantaneous nature of haptic glance [Klatzky and Lederman 1995] makes it strongly dependent on purely geometric attributes, unlike the temporal dependency of contour following.

Klatzky and Lederman [Klatzky and Lederman 1995] conducted experiments in which subjects were instructed to identify objects from brief cutaneous exposures (i.e., haptic glances). Subjects had an advance hypothesis of the nature of the object. The purpose of the study was to discover how, and how well, subjects identify objects from brief contact. According to Klatzky and Lederman, during haptic glance a subject has access to three pieces of information: roughness, compliance, and local features. Roughness and compliance are material properties that can be extracted from lower-level processing, while local features can lead to object identification by feature matching during higher-level processing. In the experiments, highest identification accuracy was achieved with small objects, whose *shapes* fit on a fingertip. Klatzky and Lederman concluded that large contact area helped in the identification of textures or patterns, although it was better to have a stimulus of a size comparable to or just slightly smaller than that of the contact area for the identification of geometric surface features. The experiments conducted by Klatzky and Lederman posit an interesting relation between feature size and contact area during cutaneous perception.

Okamura and Cutkosky [Okamura and Cutkosky 1999; Okamura and Cutkosky 2001] analyzed feature detection in robotic exploration, which can be regarded as a case of object-object interaction. They characterized geometric surface features based on the ratios of their curvatures to the radii of the robotic fingertips acquiring the surface data. They observed that a larger fingertip, which provides a larger contact area, can miss small geometric features. To summarize, the studies by Klatzky and Lederman [Klatzky and Lederman 1995] and Okamura and Cutkosky [Okamura and Cutkosky 1999; Okamura and Cutkosky 2001] lead to the observation that human haptic perception of the existence of a geometric surface feature depends on the ratio between the contact area and the size of the feature, not the absolute size of the feature itself. This observation has driven the design of multiresolution contact determination algorithms for haptic rendering [Otaduy and Lin 2003b].

3.2 Perception of Texture and Roughness

Klatzky and Lederman [Klatzky and Lederman 2003] describe a textured surface as a surface with protuberant elements arising from a relatively homogeneous substrate. Interaction with a textured surface results in perception of roughness. Existing research on the psychophysics of texture perception indicates a clear dichotomy of exploratory procedures: (a) perception of texture with the bare skin, and (b) perception through an intermediate (rigid) object, a probe.

Most of the research efforts have been directed towards the characterization of cutaneous perception of textures. Katz [Katz 1989] suggested that roughness is perceived through a combination of spatial and vibratory codes during direct interaction with the skin. More recent evidence demonstrates that static pressure distribution plays a dominant role in perception of coarse textures (features larger than 1mm) [Lederman 1974; Connor and Johnson 1992], but motion-induced vibration is necessary for perceiving fine textures [LaMotte and Srinivasan 1991; Hollins and Risner 2000].

As pointed out by Klatzky and Lederman [Klatzky and Lederman 2002], in object-object interaction roughness is encoded in vibratory motion transmitted to the subject. In the last few years, Klatzky and Lederman have directed experiments that analyze the influence of several factors on roughness perception through a rigid probe. Klatzky et al. [Klatzky et al. 2003] distinguished three types of factors that may affect the perceived magnitude of roughness: interobject physical interaction, skin- and limb-induced filtering prior to cutaneous and kinesthetic perception, and higher-level factors such as efferent commands. The design of contact determination and collision response algorithms for haptic texture rendering is mostly concerned with factors related to the physical interaction between objects: object geometry [Lederman et al. 2000; Klatzky et al. 2003], applied force [Lederman et al. 2000], and exploratory speed [Lederman et al. 1999; Klatzky et al. 2003]. The influence of these factors has been addressed in the design of haptic texture rendering algorithms [Otaduy et al. 2004].

The experiments conducted by Klatzky and Lederman to characterize roughness perception [Klatzky and Lederman 2002] used a common setup: subjects explored a textured plate with a probe with a spherical tip, and then they reported a subjective measure of roughness. Plates of jittered raised dots were used, and the mean frequency of dot distribution was one of the variables in the experiments. The resulting data was analyzed by plotting subjective roughness values vs. dot interspacing in logarithmic graphs.

Klatzky and Lederman [Klatzky and Lederman 1999] compared graphs of roughness vs. texture spacing (a) with finger exploration and (b) with a rigid probe. They concluded that, in the range of their data, roughness functions were best fit by linear approximations in finger exploration and by quadratic approximations in probe-based exploration. In other words, when perceived through a rigid spherical probe, roughness initially increases as texture spacing increases, but, after reaching a maximum roughness value, it decreases again. Based on this finding, the influence of other factors on roughness perception can be characterized by the maximum value of roughness and the value of texture spacing at which this maximum takes place.

Lederman et al. [Lederman et al. 2000] demonstrated that the diameter of the spherical probe plays a crucial role in the maximum value of perceived roughness and the location of the maximum. The roughness peak is higher for smaller probes, and it occurs at smaller texture spacing values. Lederman et al. [Lederman et al. 2000] also studied the influence of the applied normal force during exploration. Roughness is higher for larger force, but the influence on the location of the peak is negligible. The effect of exploratory speed was studied by Lederman et al. [Lederman et al. 1999]. They found that the peak of roughness occurs at larger texture spacing for higher speed. Also, with higher speed, textured plates feel smoother at small texture spacing, and rougher at large spacing values. The studies reflected that speed has a stronger effect in passive interaction than in active interaction.

3.3 Haptic and Visual Cross-modal Interaction

Haptic rendering is often presented along with visual display. Therefore, it is important to understand the issues involved in cross-modal interaction. Klatzky and Lederman [Klatzky and Lederman 2003] discuss aspects of visual and haptic cross-modal integration from two perspectives: attention and dominance.

Spence et al. [Spence et al. 2000] have studied how visual and tactile cues can influence a subject's attention. Their conclusions are that visual and tactile cues are treated together in a single attentional mechanism, and wrong attention cues can affect perception negatively.

Sensory dominance is usually studied by analyzing perceptual discrepancies in situations where cross-modal integration yields a unitary perceptual response. One example of relevance for this dissertation is the detection of object collision. During object manipulation, humans determine whether two objects are in contact based on a combination of visual and haptic cues. Early studies of sensory dominance seemed to point to a strong dominance of visual cues over haptic cues [Rock and Victor 1964], but in the last decades psychologists agree that sensory inputs are weighted based on their statistical reliability or relative appropriateness, measured in terms of accuracy, precision, and cue availability [Heller et al. 1999; Ernst and Banks 2001; Klatzky and Lederman 2003].

The design of contact determination algorithms can also benefit from existing studies on the visual perception of collisions in computer animations. O'Sullivan and her colleagues [O'Sullivan et al. 1999; O'Sullivan and Dingliana 2001; O'Sullivan et al. 2003] have investigated different factors affecting visual collision perception, including eccentricity, separation, distractors, causality, and accuracy of simulation results. Basing their work on a model of human visual perception validated by psychophysical experiments, they demonstrated the feasibility of using these factors for scheduling interruptible collision detection among large numbers of visually homogeneous objects.

4 Stability and Control Theory Applied to Haptic Rendering

In haptic rendering, the human user is part of the dynamic system, along with the haptic device and the computer implementation of the virtual environment. The complete human-in-the-loop system can be regarded as a sampled-data system [Colgate and Schenkel 1994], with a continuous component (the user and the device) and a discrete one (the implementation of the virtual environment and the device controller). Stability becomes a crucial feature, because instabilities in the system can produce oscillations that distort the perception of the virtual environment, or uncontrolled motion of the device that can even hurt the user. In Sec. 2.2, we have briefly discussed the importance of stability for haptic rendering, and we have introduced the effect of the force update rate on stability. In this section we review and discuss in more detail existing work in control theory related to stability analysis of haptic rendering.

4.1 Mechanical Impedance Control

The concept of mechanical impedance extends the notion of electrical impedance and refers to the quotient between force and velocity. Hogan [Hogan 1985] introduced the idea of impedance control for contact tasks in manipulation. Earlier techniques controlled contact force, robot velocity, or both, but Hogan suggested controlling directly the mechanical impedance, which governs the dynamic properties of the system. When the end effector of a robot touches a rigid surface, it suffers a drastic change of mechanical impedance, from low impedance in free space, to high impedance during contact. This phenomenon imposes serious difficulties on earlier control techniques, inducing instabilities.

The function of a haptic device is to display the feedback force of a virtual world to a human user. Haptic devices present control challenges very similar to those of manipulators for contact tasks. As introduced in Sec. 2.1, there are two major ways of controlling a haptic device: impedance control and admittance control. In impedance control, the user moves the device, and the controller produces a force dependent on the interaction in the virtual world. In admittance control, the user applies a force to the device, and the controller moves the device according to the virtual interaction.

In both impedance and admittance control, high control gains can induce instabilities. In impedance control, instabilities may arise in the simulation of stiff virtual surfaces. The device must react with large changes in force to small changes in the position. Conversely, in admittance control, rendering a stiff virtual surface is not a challenging problem, because it is implemented as a low controller gain. In admittance control, however, instabilities may arise during free-space motion in the virtual world, because the device must move at high velocities under small applied forces, or when the device rests on a stiff physical surface. Impedance and admittance control can therefore be regarded as complementary control techniques, best suited for opposite applications. Following the unifying framework presented by Adams and Hannaford [Adams and Hannaford 1998], Contact determination and force computation algorithms are often independent of the control strategy.

4.2 Stable Rendering of Virtual Walls

Since the introduction of impedance control by Hogan [Hogan 1985], the analysis of the stability of haptic devices and haptic rendering algorithms has focused on the problem of rendering stiff virtual walls. This was known to be a complex problem at early stages of research in haptic rendering [Kilpatrick 1976], but impedance control simplified the analysis, because a virtual wall can be modeled easily using stiffness and viscosity parameters.

Ouh-Young [Ouh-Young 1990] created a discrete model of the Argonne ARM and the human arm and analyzed the influence of force update rate on the stability and responsiveness of the system. Minsky, Brooks, et al. [Minsky et al. 1990; Brooks, Jr. et al. 1990] observed that update rates as high as 500Hz or 1kHz might be necessary in order to achieve stability.

Colgate and Brown [Colgate and Brown 1994] coined the term Z-Width for describing the range of mechanical impedances that a haptic device can render while guaranteeing stability. They concluded that physical dissipation is essential for achieving stability and that the maximum achievable virtual stiffness is proportional to the update rate. They also analyzed the influence of position sensors and quantization, and concluded that sensor resolution must be maximized and the velocity signal must be filtered.

Almost in parallel, Salcudean and Vlaar [Salcudean and Vlaar 1994] studied haptic rendering of virtual walls, and techniques for improving the fidelity of the rendering. They compared a continuous model of a virtual wall with a discrete model that accounts for differentiation of the position signal. The continuous model is unconditionally stable, but this is not true for the discrete model. Moreover, in the discrete model fast damping of contact oscillations is possible only with rather low contact stiffness and, as indicated by Colgate and Brown too [Colgate and Brown 1994], this value of stiffness is proportional to the update rate. Salcudean and Vlaar proposed the addition of braking pulses, proportional to collision velocity, for improving the perception of virtual walls.

4.3 Passivity and Virtual Coupling

A subsystem is *passive* if it does not add energy to the global system. Passivity is a powerful tool for analyzing stability of coupled systems, because the coupled system obtained from two passive subsystems is always stable. Colgate and his colleagues were the first to apply passivity criteria to the analysis of stability in haptic rendering of virtual walls [Colgate et al. 1993]. Passivity-based analysis has enabled separate study of the behavior of the human subsystem, the haptic device, and the virtual environment in force-feedback systems.

4.3.1 Human Sensing and Control Bandwidths

Hogan discovered that the human neuromuscular system exhibits externally simple, springlike behavior [Hogan 1986]. This finding implies that the human arm holding a haptic device can be regarded as a passive subsystem, and the stability analysis can focus on the haptic device and the virtual environment.

Note that human limbs are not passive in all conditions, but the bandwidth at which a subject can perform active motions is very low compared to the frequencies at which stability problems may arise. Some authors [Shimoga 1992; Burdea 1996] report that the bandwidth at which humans can perform controlled actions with the hand or fingers is between 5 and 10Hz. On the other hand, sensing bandwidth can be as high as 20 to 30Hz for proprioception, 400Hz for tactile sensing, and 5 to 10kHz for roughness perception.

4.3.2 Passivity of Virtual Walls

Colgate and Schenkel [Colgate and Schenkel 1994] observed that the oscillations perceived by a haptic user during system instability are a result of active behavior of the force-feedback system. This active behavior is a consequence of time delay and loss of information inherent in sampled-data systems, as suggested by others before [Brooks, Jr. et al. 1990]. Colgate and Schenkel formulated passivity conditions in haptic rendering of a virtual wall. For that analysis, they modeled the virtual wall as a viscoelastic unilateral constraint, and they accounted for the continuous dynamics of the haptic device, sampling of the position signal, discrete differentiation for obtaining velocity, and a zero-order hold of the output force. They reached a sufficient condition for passivity that relates the stiffness K and damping B of the virtual wall, the inherent damping b of the device, and the sampling period T:

$$b > \frac{KT}{2} + B. \tag{1}$$

4.3.3 Stability of Non-linear Virtual Environments

After deriving stability conditions for rendering virtual walls modeled as unilateral linear constraints, Colgate and his colleagues considered more complex environments [Colgate et al. 1995]. A general virtual environment is non-linear, and it presents multiple and variable constraints. Their approach enforces a discrete-time passive implementation of the virtual environment and sets a multidimensional viscoelastic *virtual coupling* between the virtual environment and the haptic display. In this way, the stability of the system is guaranteed as long as the virtual coupling is itself passive, and this condition can be analyzed using the same techniques as those used for virtual walls [Colgate and Schenkel 1994]. As a result of Colgate's virtual coupling [Colgate et al. 1995], the complexity of the problem was shifted towards designing a passive solution of virtual world dynamics. As noted by Colgate et al. [Colgate et al. 1995], one possible way to enforce passivity in rigid body dynamics simulation is to use implicit integration with penalty methods.

Adams and Hannaford [Adams and Hannaford 1998] provided a framework for analyzing stability with admittance-type and impedance-type haptic devices. They derived stability conditions for coupled systems based on network theory. They also extended the concept of virtual coupling to admittance-type devices.

Miller et al. [Miller et al. 1990] extended Colgate's passivity analysis techniques, relaxing the requirement of passive virtual environments but enforcing *cyclo-passivity* of the complete system. Hannaford and his colleagues [Hannaford et al. 2002] investigated the use of adaptive controllers instead of the traditional fixed-value virtual couplings. They designed passivity observers and passivity controllers for dissipating the excess of energy generated by the virtual environment.

4.4 Multirate Approximation Techniques

Multirate approximation techniques, though simple, have been successful in improving the stability and responsiveness of haptic rendering systems. The idea is to perform a full update of the virtual environment at a low frequency (limited by computational resources and the complexity of the system) and to use a simplified approximation for performing high-frequency updates of force feedback.

Adachi [Adachi et al. 1995] proposed an *intermediate representation* for haptic display of complex polygonal objects. In a slow collision detection thread, he computed a plane that served as a unilateral constraint in the force-feedback thread. This technique was later adapted by Mark et al. [Mark et al. 1996], who interpolated the intermediate representation between updates. This approach enables higher stiffness values than approaches that compute the feedback force values at the rate imposed by collision detection. More recently, a similar multirate approach has been followed by many authors for haptic interaction with deformable models [Astley and Hayward 1998; Çavuşoğlu and Tendick 2000; Duriez et al. 2004]. Ellis et al. [Ellis et al. 1997] produce higher-quality rendering by upsampling directly the output force values.

5 Collision Detection

Collision detection has received much attention in robotics, computational geometry, and computer graphics. Some researchers have investigated the problem of interference detection as a mechanism for indicating whether object configurations are valid or not. Others have tackled the problems of computing separation or penetration distances, with the objective of applying collision response in simulated environments. The existing work on collision detection can be classified based on the types of models handled: 2-manifold polyhedral models, polygon soups, curved surfaces, etc. In this section we focus on collision detection for polyhedral models. The vast majority of the algorithms used in practice proceed in two steps: first they cull large portions of the objects that are not in close proximity, using spatial partitioning, hierarchical techniques, or visibility-based properties, and then they perform primitive-level tests.

In this section, we first describe the problems of interference detection and computation of separation distance between polyhedra, with an emphasis on algorithms specialized for convex polyhedra. Then we survey algorithms for the computation of penetration depth, the use of hierarchical techniques, and multiresolution collision detection. we conclude the section by covering briefly the use of graphics processors for collision detection and the topic of continuous collision detection. For more information on collision detection, please refer to surveys on the topic [Lin and Gottschalk 1998; Klosowski et al. 1998; Lin and Manocha 2004].

5.1 Proximity Queries Between Convex Polyhedra

The property of convexity has been exploited in algorithms with sublinear cost for detecting interference or computing the closest distance between two polyhedra. Detecting whether two convex polyhedra intersect can be posed as a linear programming problem, searching for the coefficients of a separating plane. Well-known linear programming algorithms [Seidel 1990] can run in expected linear time due to the low dimensionality of the problem.

The separation distance between two polyhedra A and B is equal to the distance from the origin to the Minkowski sum of A and $-B$ [Cameron and Culley 1986]. This property was exploited by Gilbert et al. [Gilbert et al. 1988] in order to design a convex optimization algorithm (known as GJK) for computing the separation distance between convex polyhedra, with linear-time performance in practice. Cameron [Cameron 1997] modified the GJK algorithm to exploit motion coherence in the initialization of the convex optimization at every frame for dynamic problems, achieving nearly constant running-time in practice.

Lin and Canny [Lin and Canny 1991; Lin 1993] designed an algorithm for computing separation distance by tracking the closest features between convex polyhedra. Their algorithm "walks" on the surfaces of the polyhedra until it finds two features that lie on each other's Voronoi region. Exploiting motion coherence and geometric locality, *Voronoi marching* runs in nearly constant time per frame. Mirtich [Mirtich 1998b] later improved the robustness of this algorithm.

Given polyhedra A and B with m and n polygons respectively, Dobkin and Kirkpatrick [Dobkin and Kirkpatrick 1990] proposed an algorithm for interference detection with $O(\log m \log n)$ time complexity that uses hierarchical representations of the polyhedra. Others have also exploited the use of hierarchical convex representations along with temporal coherence in order to accelerate queries in dynamic scenes. Guibas et al. [Guibas et al. 1999] employ the inner hierarchies suggested by Dobkin and Kirkpatrick, but they perform faster multilevel walking. Ehmann and Lin [Ehmann and Lin 2000] employ a modified version of Dobkin and Kirkpatrick's outer hierarchies, computed using simplification techniques, along with a multilevel implementation of Lin and Canny's Voronoi marching [Lin and Canny 1991].

5.2 Penetration Depth

The penetration depth between two intersecting polyhedra A and B is defined as the minimum translational distance required for separating them. For intersecting polyhedra, the origin is contained in the Minkowski sum of A and $-B$, and the penetration depth is equal to the minimum distance from the origin to the surface of the Minkowski sum. The computation of penetration depth can be $\Omega(m^3 n^3)$ for general polyhedra [Dobkin et al. 1993].

Many researchers have restricted the computation of penetration depth to convex polyhedra. In computational geometry, Dobkin et al. [Dobkin et al. 1993] presented an algorithm for computing directional penetration depth, while Agarwal et al. [Agarwal et al. 2000] introduced a randomized algorithm for computing the penetration depth between convex polyhedra. Cameron [Cameron 1997] extended the GJK algorithm [Gilbert et al. 1988] to compute bounds of the penetration depth, and van den Bergen [van den Bergen 2001] furthered his work. Kim et al. [Kim et al. 2002a] presented an algorithm that computes a locally optimal solution of the penetration depth by walking on the surface of the Minkowski sum.

The fastest algorithms for computation of penetration depth between arbitrary polyhedra take advantage of discretization. Fisher and Lin [Fisher and Lin 2001] estimate penetration depth using distance fields computed with fast marching level-sets. Hoff et al. [Hoff et al. 2001] presented an image-based algorithm implemented on graphics hardware. On the other hand, Kim et al. [Kim et al. 2002b] presented an algorithm that decomposes the polyhedra into convex patches, computes the Minkowski sums of pairwise patches, and then uses an image-based technique in order to find the minimum distance from the origin to the surface of the Minkowski sums.

5.3 Hierarchical Collision Detection

The algorithms for collision detection between convex polyhedra are not directly applicable to non-convex polyhedra or models described as polygon soups. Brute force checking of all triangle pairs, however, is usually unnecessary. Collision detection between general models achieves large speed-ups by using hierarchical culling or spatial partitioning techniques that restrict the primitive-level tests. Over the last decade, bounding volume hierarchies (BVH) have proved successful in the acceleration of collision detection for dynamic scenes of rigid bodies. For an extensive description and analysis of the use of BVHs for collision detection, please refer to Gottschalk's PhD dissertation [Gottschalk 2000].

Assuming that an object is described by a set of triangles T, a BVH is a tree of BVs, where each BV C_i bounds a cluster of triangles $T_i \in T$. The clusters bounded by the children of C_i constitute a partition of T_i. The effectiveness of a BVH is conditioned by ensuring that the branching factor of the tree is O(1) and that the size of the leaf clusters is also O(1). Often, the leaf BVs bound only one triangle. A BVH may be created in a top-down manner, by successive partitioning of clusters, or in a bottom-up manner, by using merging operations.

In order to perform interference detection using BVHs, two objects are queried by recursively traversing their BVHs in tandem. Each recursive step tests whether a pair of BVs a and b, one from each hierarchy, overlap. If a and b do not overlap, the recursion branch is terminated. Otherwise, if they overlap, the algorithm is applied recursively to their children. If a and b are both leaf nodes, the triangles within them are tested directly. This process can be generalized to other types of proximity queries as well.

One determining factor in the design of a BVH is the selection of the type of BV. Often there is a trade-off among the tightness of the BV (and therefore the culling efficiency), the cost of the collision test between two BVs, and the dynamic update of the BV (relevant for deformable models). Some of the common BVs, sorted approximately according to increasing query time, are: spheres [Quinlan 1994; Hubbard 1994], axis-aligned bounding boxes (AABB) [Beckmann et al. 1990], oriented bounding boxes (OBB) [Gottschalk et al. 1996], k-discrete-orientation polytopes (k-DOP) [Klosowski et al. 1998], convex hulls [Ehmann and Lin 2001], and swept sphere volumes (SSV) [Larsen et al. 2000]. BVHs of rigid bodies can be computed as a preprocessing step, but deformable models require a bottom-up update of the BVs after each deformation. Recently, James and Pai [James and Pai 2004] have presented the BD-tree, a variant of the sphere-tree data structure [Quinlan 1994] that can be updated in a fast top-down manner if the deformations are described by a small number of parameters.

5.4 Multiresolution Collision Detection

Multiresolution analysis of a function decomposes the function into a basic low-resolution representation and a set of detail terms at increasing resolutions. Wavelets provide a mathematical framework for defining multiresolution analysis [Stollnitz et al. 1996].

Multiresolution representations of triangles meshes have drawn important attention in computer graphics. They have been defined in two major ways: following the mathematical framework of wavelets and subdivision surfaces [Lounsbery et al. 1997; Eck et al. 1995] or following level-of-detail (LOD) simplification techniques (please refer to [Luebke et al. 2002] for a survey on the topic). LOD techniques present the advantage of being applicable to arbitrary meshes, but they lack a well-defined metric of resolution. They construct the multiresolution representations starting from full-resolution meshes and applying sequences of local simplification operations. LOD techniques can be divided into those that produce a discrete set of representations (static LODs), and those that produce continuously adaptive representations (dynamic LODs). Multiresolution or LOD techniques have been used in applications such as view-dependent rendering [Hoppe 1997; Luebke and Erikson 1997], interactive editing of meshes [Zorin et al. 1997], or real-time deformations [Debunne et al. 2001]. The idea behind multiresolution techniques is to select the resolution or LOD of the representation in an adaptive manner based on perceptual parameters, availability of computational resources, and so forth.

Multiresolution collision detection refers to the execution of approximate collision detection queries using adaptive object representations. Hubbard [Hubbard 1994] introduced the idea of using sphere-trees [Quinlan 1994] for multiresolution collision detection, refining the BVHs in a breadth-first manner until the time allocated for collision detection expires. In a sphere-tree each level of the BVH can be regarded as an implicit approximation of the given mesh, by defining the surface as a union of spheres. Unlike LOD techniques, in which simplification operations minimize surface deviation, sphere-trees add extraneous "bumpiness" to the surface, and this characteristic can hurt collision response.

O'Sullivan and Dingliana [O'Sullivan and Dingliana 2001] have incorporated perceptual parameters into the refinement of sphere-trees. They insert pairs of spheres that test positive for collision in a priority queue sorted according to perceptual metrics (e.g., local relative velocity, distance to the viewer, etc.). In this way the adaptive refinement focuses on areas of the objects where errors are most noticeable.

The use of multiresolution representations for haptic rendering has also been investigated by several researchers. Pai and Reissel [Pai and Reissel 1997] investigated the use of multiresolution image curves for 2D haptic interaction. El-Sana and Varshney [El-Sana and Varshney 2000] applied LOD techniques to 3-DoF haptic rendering. They created a multiresolution representation of the haptically rendered object as a preprocessing step and, at runtime, they represented the object at high resolution near the probe point and at low resolution further away. Their approach does not extend naturally to the interaction between two objects, since multiple disjoint contacts can occur simultaneously at widely varying locations without much spatial coherence. Otaduy and Lin [Otaduy and Lin 2003b; Otaduy and Lin 2003a] introduced *contact levels of detail*, dual hierarchical representations for multiresolution collision detection, and they applied them to 6-DoF haptic rendering, producing a sensation-preserving simplified rendering.

5.5 Other Techniques for Collision Detection

We briefly cover two additional topics with potential applicability in haptic rendering: the use of graphics processors for collision detection, and continuous collision detection.

5.5.1 Use of Graphics Processors for Collision Detection

The processing capability of GPUs is growing at a rate higher than Moore's law [Govindaraju et al. 2003], and this circumstance has generated an increasing use of GPUs for general-purpose computation, including collision detection. Rasterization hardware enables high performance of image-based collision detection algorithms. Hoff et al. [Hoff et al. 2001] presented an algorithm for estimating penetration depth between deformable polygons using distance fields computed on graphics hardware. Others have formulated collision detection queries as visibility problems. Lombardo et al. [Lombardo et al. 1999] intersected a complex object against a simpler one using the view frustum and clipping planes, and they detected intersecting triangles by exploiting OpenGL capabilities. More recently, Govindaraju et al. [Govindaraju et al. 2003] have designed an algorithm that performs series of visibility queries and achieves fast culling of non-intersecting primitives in N-body problems with nonrigid motion.

5.5.2 Continuous Collision Detection

Continuous collision detection refers to a temporal formulation of the collision detection problem. The collision query attempts to find intersecting triangles and the time of intersection. Redon et al. [Redon et al. 2002] proposed an algorithm that assumes an arbitrary interframe rigid motion and incorporates the temporal dimension in OBB-trees using interval arithmetic. Continuous collision detection offers potential applicability to haptic rendering because it may enable constraint-based simulations without expensive backtracking operations used for computing the time of first collision.

6 Rigid Body Simulation

Computation of the motion of a rigid body consists of solving a set of ordinary differential equations (ODEs). The most common way to describe the motion of a rigid body is by means of the Newton-Euler equations, which define the time derivatives of the linear momentum, $\mathbf{P}$, and angular momentum, $\mathbf{L}$, as a function of external force $\mathbf{F}$ and torque $\mathbf{T}$:

$$
\begin{aligned}
\mathbf{F}(t) &= \dot{\mathbf{P}}(t) = m\,\ddot{\mathbf{x}}(t), \\
\mathbf{T}(t) &= \dot{\mathbf{L}}(t) = \omega(t) \times (M\omega(t)) + M\dot{\omega}(t).
\end{aligned} \tag{2}
$$

As shown in the equations, momentum derivatives can be expressed in terms of the linear acceleration of the center of mass $\ddot{\mathbf{x}}$, the angular velocity ω, the mass of the body m, and the mass matrix M. The complexity of rigid body simulation lies in the computation of force and torque resulting from contacts between bodies. Research in the field of rigid body simulation has revolved around different methods for computing contact forces and the resulting accelerations and velocities, ranging from approximate methods that consider each contact independently (such as penalty-based methods) to analytic methods that account concurrently for all non-penetration constraints. Important efforts have been devoted to capturing friction forces as well.

In this section we briefly describe the main methods for solving the motion of colliding rigid bodies, focusing on their applicability to haptic rendering. For further information, please refer to Baraff's or Mirtich's dissertations [Baraff 1992; Mirtich 1996], SIGGRAPH course notes on the topic [Baraff and Witkin 2001], or recent work by Stewart and Trinkle [Stewart and Trinkle 2000]. In the last few years, especially in the field of computer graphics, attention has been drawn towards the problem of simulating the interaction of many rigid bodies [Mirtich 2000; Milenkovic and Schmidl 2001; Guendelman et al. 2003]. For haptic rendering, however, one is mostly concerned with the dynamics of the object grasped by the user; therefore the interaction of many rigid objects is not discussed here.

6.1 Penalty-Based Methods

When two objects touch or collide, collision response must be applied to prevent object interpenetration. One method for implementing collision response is the insertion of stiff springs at the points of contact [Moore and Wilhelms 1988]. This method is inspired by the fact that, when objects collide, small deformations take place at the region of contact, and these deformations can be modeled with springs, even if the objects are geometrically rigid.

Given two intersecting objects A and B, penalty-based collision response requires the definition of a contact point $\mathbf{p}$, a contact normal $\mathbf{n}$ and a penetration depth δ. The penalty-based spring force and torque applied to object A are defined as follows:

$$
\begin{aligned}
&\mathbf{F}_A = -f(\delta)\mathbf{n}, \\
&\mathbf{T}_A = (\mathbf{p} - \mathbf{c}_A) \times \mathbf{F}_A,
\end{aligned} \tag{3}
$$

where $\mathbf{c}_A$ is the center of mass of A. Opposite force and torque are applied to object B. The function f could be a linear function defined by a constant stiffness k or a more complicated non-linear function. It could also contain a viscous term, dependent on the derivative of the penetration depth.

The basic formulation of penalty methods can be modified slightly in order to introduce repulsive forces between objects, by inserting contact springs when the objects come closer than a distance tolerance d. In this way, object interpenetration occurs less frequently. The addition of a tolerance has two major advantages: the possibility of using penalty methods in applications that do not allow object interpenetration, and a reduction of the cost of collision detection. As noted in Sec. 5, computation of penetration depth is notably more costly than computation of separation distance.

Penalty-based methods offer several attractive properties: the force model is local to each contact and computationally simple, object interpenetration is inherently allowed, and contact determination needs to be performed only once per simulation frame. This last property makes penalty-based methods best suited for interactive applications with fixed time steps, such as haptic rendering [McNeely et al. 1999; Kim et al. 2003; Johnson and Willemsen 2003; Otaduy and Lin 2005] and games [Wu 2000; Larsen 2001]. But penalty-based methods also have some disadvantages. There is no direct control over physical parameters, such as the coefficient of restitution. Non-penetration constraints are enforced by means of very high contact stiffness, and this circumstance leads to instability problems if numerical integration is executed using fast, explicit methods. The solution of penalty-based simulation using implicit integration, however, enhances stability in the presence of high contact stiffness [Wu 2000; Larsen 2001; Otaduy and Lin 2005].

Friction effects can be incorporated into penalty-based methods by means of localized force models that consider each contact point independently. Most local friction methods propose different force models for static or dynamic situations [Karnopp 1985; Hayward and Armstrong 2000]. Static friction is modeled by fixing adhesion points on the surfaces of the colliding objects and setting tangential springs between the contact points and the adhesion points. If the elastic friction force becomes larger than a threshold determined by the normal force and the friction coefficient, the system switches to dynamic mode. In the dynamic mode, the adhesion point follows the contact point. The system returns to static mode if the velocity falls under a certain threshold.

So far, we have analyzed contact determination and collision response as two separate problems, but the output of the contact determination step has a strong influence on the smoothness of collision response and, as a result, on the stability of numerical integration. As pointed out by Larsen [Larsen 2001], when a new contact point is added, the associated spring must be unstretched. In other words, the penetration depth value must be zero initially and must grow smoothly. The existence of geometry-driven discontinuities is an inherent problem of penalty-based simulations with fixed time steps. Some authors [Hasegawa and Sato 2004] have proposed sampling the intersection volume to avoid geometric discontinuities in the application of penalty-based methods to rigid body simulation and haptic rendering, but this approach is applicable only to very simple objects.

6.2 Constraint-Based Simulation

Constraint-based methods for the simulation of rigid body dynamics handle all concurrent contacts in a single computational problem and attempt to find contact forces that produce physically and geometrically valid motions. Specifically, they integrate the Newton-Euler equations of motion (see Eq. 2), subject to geometric constraints that prevent object interpenetration. The numerical integration of Newton-Euler equations must be interrupted before objects interpenetrate. At a collision event, object velocities and accelerations must be altered, so that non-penetration constraints are not violated and numerical integration can be restarted. One must first compute contact impulses that produce constraint-valid velocities. Then, one must compute contact forces that produce valid accelerations.

The relative normal accelerations $\mathbf{a}$ at the points of contact can be expressed as linear combinations of the contact forces $\mathbf{F}$ (with constant matrix A and vector $\mathbf{b}$). Moreover, one can impose non-penetration constraints on the accelerations and non-attraction constraints on the forces:

$$\begin{aligned} \mathbf{a} &= A\mathbf{F} + \mathbf{b}, \\ \mathbf{a} &\geq 0, \quad \mathbf{F} \geq 0. \end{aligned} \tag{4}$$

Baraff [Baraff 1989] pioneered the application of constraint-based approaches to rigid body simulation in computer graphics. He posed constrained rigid body dynamics simulation as a quadratic programming problem on the contact forces, and he proposed a fast, heuristic-based solution for the frictionless case. He defined a quadratic cost function based on the fact that contact forces occur only at contact points that are not moving apart:

$$\min\left(\mathbf{F}^T\mathbf{a}\right) = \min\left(\mathbf{F}^T A\mathbf{F} + \mathbf{F}^T\mathbf{b}\right). \tag{5}$$

The quadratic cost function suggested by Baraff indicates that either the normal acceleration or the contact force should be 0 at a resting contact. As indicated by Cottle et al. [Cottle et al. 1992], this condition can be formulated as a linear complementarity problem (LCP). Baraff [Baraff 1991; Baraff 1992] added dynamic friction to the formulation of the problem and suggested approaches for static friction, as well as a solution following an algorithm by Lemke [Lemke 1965] with expected polynomial cost in the number of constraints. Earlier, Lötstedt had studied the problem of rigid body dynamics with friction in the formulation of the LCP [Lötstedt 1984]. Later, Baraff himself [Baraff 1994] adapted an algorithm by Cottle and Dantzig [Cottle and Dantzig 1968] for solving frictionless LCPs to the friction case, and achieved linear-time performance in practice.

Stewart and Trinkle [Stewart and Trinkle 1996] presented an implicit LCP formulation of constraint-based problems. Unlike previous algorithms, which enforced the constraints only at the beginning of each time step, their algorithm solves for contact impulses that also enforce the constraints at the end of the time step. This formulation eliminates the need to locate collision events, but it increases the number of constraints to be handled, and it is unclear how it behaves with complex objects.

Stewart and Trinkle [Stewart and Trinkle 1996] mention the existence of geometry-driven discontinuities, similar to the ones appearing with penalty methods, in their implicit formulation of the LCP. After numerical integration of object position and velocities, new non-penetration constraints are computed. If numerical integration is not interrupted at collision events, the newly computed non-penetration constraints may not hold. Constraint violation may produce unrealistically high contact impulses and object velocities in the next time step. This phenomenon is equivalent to the effect of prestretched penalty-based springs described by Larsen [Larsen 2001]. Stewart and Trinkle suggest solving a non-linear complementarity problem, with additional cost involved.

If numerical integration is interrupted at collision events, the effects of geometry-driven discontinuities can be alleviated by capturing all the contact points that bound the contact region. Baraff [Baraff 1989] considers polygonal contact regions between polyhedral models and defines contact constraints at the vertices that bound the polygonal regions. Similarly, Mirtich [Mirtich 1998a] describes polygonal contact areas as combinations of edge-edge and vertex-face contacts.

6.3 Impulse-Based Dynamics

Mirtich [Mirtich and Canny 1995; Mirtich 1996] presented a method for handling collisions in rigid body dynamics simulation based solely on the application of impulses to the objects. In situations of resting, sliding, or rolling contact, constraint forces are replaced by trains of impulses. Mirtich defined a collision matrix that relates contact impulse to the change in relative velocity at the contact. His algorithm decomposes the collision event into two separate processes: compression and restitution. Each process is parameterized separately, and numerical integration is performed in order to compute the velocities after the collision. The parameterization of the collision event enables the addition of a friction model to instantaneous collisions.

The time-stepping engine of impulse-based dynamics is analogous to the one in constraint-based dynamics: numerical integration must be interrupted before interpenetration occurs, and valid velocities must be computed. One of the problems of impulse-based dynamics emerges during inelastic collisions from the fact that accelerations are not recomputed. The energy loss induced by a train of inelastic collisions reduces the time between collisions and increases the cost of simulation per frame. In order to handle this problem, Mirtich suggested the addition of unrealistic, but visually imperceptible, energy to the system when the microcollisions become too frequent. As has been pointed out by Mirtich, impulse-based approaches are best suited for simulations that are collision-intensive, with multiple, different impacts occurring frequently.

7 Haptic Texture Rendering

Although haptic rendering of textures was one of the first tackled problems [Minsky et al. 1990], it has been mostly limited to the interaction between a probe point and a textured surface. We begin this section with a description of Minsky's pioneering algorithm for rendering textures on the plane [Minsky 1995]. Then we discuss rendering of textures on 3D surfaces, covering basic 3-DoF haptic rendering, height-field-based methods, and probabilistic methods.

7.1 Rendering Textures on the Plane

Minsky [Minsky 1995] developed the *Sandpaper* system for 2-DoF haptic rendering of textures on a planar surface. Her system was built around a force model for computing 2D forces from texture height field information. Following energy-based arguments, her force model synthesizes a force $\mathbf{F}$ in 2D based on the gradient of the texture height field h at the location of the probe:

$$\mathbf{F} = -k\nabla h. \tag{6}$$

Minsky also analyzed qualitatively and quantitatively roughness perception and the believability of the proposed force model. One of the main conclusions of her work is to establish her initial hypothesis, that texture information can be conveyed by displaying forces tangential to the contact surface. This hypothesis was later exploited for rendering textured 3D surfaces [Ho et al. 1999].

7.2 3-Degree-of-Freedom Haptic Rendering

As described in Sec. 1.4, 3-DoF haptic rendering methods compute feedback force as a function of the separation between the probe point controlled with the haptic device and a contact point constrained to the surface of the haptically rendered object. Early 3-DoF haptic rendering methods set the contact point as the point on the surface of the object closest to the probe point. As has been addressed by Zilles and Salisbury [Zilles and Salisbury 1995], these methods lead to force discontinuities and possible "pop-through" problems, in which the contact point jumps between opposing sides of the object. Instead, Zilles and Salisbury proposed the *god-object* method, which defines the computation of the contact point as a constrained optimization problem. The contact point is located at a minimum distance from the probe point, but its interframe trajectory is constrained by the surface. Zilles and Salisbury solve the position of the contact point using Lagrange multipliers, once they define the set of active constraints.

Ruspini et al. [Ruspini et al. 1997] followed a similar approach. They modeled the contact point as a sphere of small radius and solved the optimization problem in the configuration space. Ruspini and his colleagues also added other effects, such as force shading for rounding of corners (by modifying the normals of constraint planes), or friction (by adding dynamic behavior to the contact point).

7.3 Methods Based on Height Fields

High-resolution surface geometry can be represented by a parameterized coarse mesh along with texture images storing detailed height field or displacement field information, similarly to the common approach of texture mapping in computer graphics [Catmull 1974]. Constraint-based 3-DoF haptic rendering methods determine a unique contact point on the surface of the rendered object. Usually, the mesh representation used for determining the contact point is rather coarse and does not capture high-frequency texture. Nevertheless, the parametric coordinates of the contact point can be used for accessing surface texture information from texture images.

Ho et al. [Ho et al. 1999] introduced a technique similar to bump mapping [Blinn 1978] that alters the surface normal based on the gradient of the texture height field. A combination of the original and refined normals is used for computing the direction of the feedback force.

Techniques for haptic texture rendering based on a single contact point can capture geometric properties of only one object and are not suitable for simulating full interaction between two surfaces. The geometric interaction between two surfaces is not limited to, and cannot be described by, a pair of contact points.

Moreover, the local kinematics of the contact between two surfaces include rotational degrees of freedom, which are not captured by point-based methods.

Ho et al. [Ho et al. 1999] indicate that a high height field gradient can induce system instability. Along a similar direction, Choi and Tan [Choi and Tan 2003b; Choi and Tan 2003a] have studied the influence of collision detection and penetration depth computation on 3-DoF haptic texture rendering. Discontinuities in the output of collision detection are perceived by the user, a phenomenon that they describe as *aliveness*. This phenomenon is a possible problem in 6-DoF haptic rendering too.

7.4 Probabilistic Methods

Some researchers have exploited statistical properties of surfaces for computing texture-induced forces that are added to the classic 3-DoF contact forces. Siira and Pai [Siira and Pai 1996] synthesized texture forces according to a Gaussian distribution for generating a sensation of roughness. In order to improve stability, they did not apply texture forces during static contact. Later, Pai et al. [Pai et al. 2001] presented a technique for rendering roughness effects by dynamically modifying the coefficient of friction of a surface. The roughness-related portion of the friction coefficient was computed according to an autoregressive process driven by noise.

Probabilistic methods have proved to be successful for rendering high-frequency roughness effects in point-surface contact. It is also possible, although this approach has yet to be explored, that they could be combined with geometric techniques for synthesizing high-frequency effects in 6-DoF haptic rendering.

8 6-Degree-of-Freedom Haptic Rendering

The problem of 6-DoF haptic rendering has been studied by several researchers. As introduced in Sec. 2.1, the existing methods for haptic rendering can be classified into two large groups based on their overall pipelines: *direct rendering* methods and *virtual coupling* methods. Each group of methods presents some advantages and disadvantages. Direct rendering methods are purely geometric, and there is no need to simulate the rigid body dynamics of the grasped object. However, penetration values may be quite large and visually perceptible, and system instability can arise if the force update rate drops below the range of stable values. Virtual coupling methods enable reduced interpenetration, higher stability, and higher control of the displayed stiffness. However, virtual coupling [Colgate et al. 1995] may introduce noticeable filtering, both tactile and visual, and it requires the simulation of rigid body dynamics.

The different 6-DoF haptic rendering methods propose a large variety of options for solving the specific problems of collision detection, collision response, and simulation of rigid body dynamics. In the presence of infinite computational resources, an ideal approach to the problem of 6-DoF haptic rendering would be to compute the position of the grasped object using constraint-based rigid body dynamics simulation [Baraff 1992] and to implement force feedback through virtual coupling. This approach has indeed been followed by some, but it imposes serious limitations on the complexity of the objects and contact configurations that can be handled interactively. We now discuss briefly the different existing methods for 6-DoF haptic rendering, focusing on those that have been applied to moderately complex objects and scenarios.

8.1 Direct Haptic Rendering Approaches

Gregory et al. [Gregory et al. 2000b] presented a 6-DoF haptic rendering system that combined collision detection based on convex decomposition of polygonal models [Ehmann and Lin 2001], predictive estimation of penetration depth, and force and torque interpolation. They were able to handle interactively dynamic scenes with several convex objects, as well as pairs of non-convex objects with a few hundred triangles and rather restricted motion. Kim et al. [Kim et al. 2003] exploited convex decomposition for collision detection and incorporated fast, incremental, localized computation of per-contact penetration depth [Kim et al. 2002a]. In order to improve stability and eliminate the influence of triangulation on the description of the contact manifold, they introduced a contact clustering technique. Their system was able to handle pairs of models with nearly one hundred convex pieces each interactively.

Earlier, Nelson et al. [Nelson et al. 1999] introduced a technique for haptic interaction between pairs of parametric surfaces. Their technique tracks contact points that realize locally maximum penetration depth during surface interpenetration. Tracking contact points, instead of recomputing them for every frame, ensures smooth penetration values, which are used for penalty-based force feedback. The contact points are solved in parametric space, and they are defined as those pairs of points for which their difference vector is collinear with surface normals.

Johnson and Willemsen [Johnson and Willemsen 2003] suggested a technique for polygonal models that defines contact points as those that satisfy a local minimum-distance criterion, according to Nelson's definition [Nelson et al. 1999]. Johnson and Willemsen exploit this definition in a fast collision culling algorithm, using spatialized normal cone hierarchies [Johnson and Cohen 2001]. The performance of their technique depends on the convexity and triangulation of the models, which affect the number of contact points. Recently, Johnson and Willemsen [Johnson and Willemsen 2004] have incorporated an approximate but fast, incremental contact-point-tracking algorithm that is combined with slower exact collision updates from their previous technique [Johnson and Willemsen 2003]. This algorithm handles models with thousands of triangles at interactive rates, but the forces may suffer discontinuities if the exact update is too slow.

8.2 Virtual Coupling with Object Voxelization

In 1999, McNeely et al. [McNeely et al. 1999] presented a system for 6-DoF haptic rendering that employs a discrete collision detection approach and virtual coupling. The system is intended for assembly and maintenance planning applications and assumes that only one of the objects in the scene is dynamic. The surfaces of the scene objects are voxelized, and the grasped object is point-sampled. The collision detection module checks for inclusion of the sample points in the scene voxels, and then a local force model is applied. Hierarchical culling of sample points is possible, but ultimately the computational cost depends on the number of contact points. This system has been integrated in a commercial product, VPS, distributed by Boeing.

McNeely and his colleagues introduced additional features in order to alleviate some of the limitations. Surface objects are voxelized only on the surface, therefore deep penetrations, which can occur if objects collide at high velocities, cannot be handled. They propose pre-contact braking forces, similar to the braking impulses suggested by Salcudean [Salcudean and Vlaar 1994], for reducing the contact velocity of the grasped object and thereby preventing deep penetrations. The existence of multiple contact points produces high stiffness values that can destabilize the simulation of rigid body dynamics. They propose averaging the effects of the different contact points before contact forces are applied to the grasped object, for limiting the stiffness and thereby ensuring stable simulation. The locality of the force model induces force discontinuities when contact points traverse voxel boundaries. They point out that force discontinuities are somewhat filtered by the virtual coupling. Renz et al. [Renz et al. 2001] modified McNeely's local force model to ensure continuity of the surface across voxel boundaries, but incurring more expensive force computation.

Using the same voxelization and point-sampling approach for collision detection, Wan and McNeely [Wan and McNeely 2003] have proposed a novel solution for computing the position of the grasped object. The early approach by McNeely et al. [McNeely et al. 1999] computed object dynamics by explicit integration of Newton-Euler equations. Instead, Wan and McNeely [Wan and McNeely 2003] presented a purely geometric solution that eliminates the instability problems that can arise due to high contact stiffness. Their algorithm formulates linear approximations of the coupling and contact force and torque in the space of translations and rotations of the grasped object. The state of the object is computed at every frame by solving for the position of quasi-static equilibrium. Deep penetrations are avoided by formulating the coupling force as a non-linear spring.

8.3 Rigid Body Dynamics with Haptic Feedback

Chang and Colgate [Chang and Colgate 1997] proposed a solution to 6-DoF haptic rendering by combining virtual coupling [Colgate et al. 1995] and rigid body simulation based on impulse dynamics [Mirtich 1996]. They found that impulses alone were not efficient in resting contact situations, and in those cases they suggested a combination of impulses and penalty forces. Recently, Constantinescu et al. [Constantinescu et al. 2004] have reached a similar conclusion. As has been addressed by Constantinescu, combining impulses and penalty forces requires a state machine in order to determine the state of the object, but it is not clear how to extend this solution to scenes with many contacts. Both Chang and Constantinescu have tested their implementations only on simple benchmarks.

One of the reasons for the simplicity of Chang and Constantinescu's benchmarks is the cost of collision detection for the simulation of rigid body dynamics. As has been discussed in Sec. 6, impulse- [Mirtich 1996] or constraint-based [Baraff 1992] methods must interrupt the integration before object interpenetration, and this leads to many collision queries per frame. Some researchers have integrated haptic interaction with constraint-based rigid body simulations [Berkelman 1999; Ruspini and Khatib 2000] in scenes with simple geometry.

As indicated in Sec. 6.1, non-penetration constraints can be relaxed using penalty-based methods. McNeely et al. [McNeely et al. 1999] employed penalty methods for rigid body simulation but, as explained earlier, they observed numerical instabilities due to high stiffness values, and large interpenetrations under high impact velocities. Those problems can be tackled with high-stiffness penalty contact forces along with implicit integration, an approach used in interactive rigid body simulations [Wu 2000; Larsen 2001]. Implicit integration requires the evaluation of the Jacobian of the Newton-Euler equations and the solution of a linear system of equations [Baraff and Witkin 1998]. As demonstrated by Otaduy and Lin [Otaduy and Lin 2005], implicit integration can be performed at force update rates under the assumption that only the grasped object is dynamic.

8.4 Multiresolution Techniques

The application of 6-DoF haptic rendering algorithms to complex models and complex contact scenarios becomes a challenging issue, due to the inherent cost of collision detection that induces slow force updates. Otaduy and Lin [Otaduy and Lin 2003b] have presented a sensation-preserving simplification technique for 6-DoF haptic rendering of complex polygonal models by selecting contact resolutions adaptively. Otaduy et al. [Otaduy et al. 2004] have also proposed a rendering algorithm for the interaction of textured surfaces. Their work is focused on the acceleration of collision detection and response using level-of-detail representations and texture images.

References

Adachi, Y., Kumano, T., and Ogino, K. 1995. Intermediate representation for stiff virtual objects. *Virtual Reality Annual International Symposium*, 203–210.

Adams, R. J., and Hannaford, B. 1998. A two-port framework for the design of unconditionally stable haptic interfaces. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Agarwal, P., Guibas, L., Har-Peled, S., Rabinovitch, A., and Sharir, M. 2000. Penetration depth of two convex polytopes in 3d. *Nordic J. Computing 7*, 227–240.

Andriot, C. 2002. Advances in virtual prototyping. *Clefs CEA Vol. 47, Research and Simulation*.

Astley, O. R., and Hayward, V. 1998. Multirate haptic simulation achieved by coupling finite element meshes through norton equivalents. *Proc. of IEEE International Conference on Robotics and Automation*.

Avila, R. S., and Sobierajski, L. M. 1996. A haptic interaction method for volume visualization. In *IEEE Visualization '96*, IEEE. ISBN 0-89791-864-9.

Baraff, D., and Witkin, A. 1998. Large steps in cloth simulation. *Proc. of ACM SIGGRAPH*, 43–54.

Baraff, D., and Witkin, A. 2001. *Physically-Based Modeling.* ACM SIGGRAPH Course Notes.

Baraff, D. 1989. Analytical methods for dynamic simulation of non-penetrating rigid bodies. In *Computer Graphics (SIGGRAPH '89 Proceedings)*, J. Lane, Ed., vol. 23, 223–232.

Baraff, D. 1991. Coping with friction for non-penetrating rigid body simulation. In *Computer Graphics (SIGGRAPH '91 Proceedings)*, T. W. Sederberg, Ed., vol. 25, 31–40.

Baraff, D. 1992. *Dynamic simulation of non-penetrating rigid body simulation.* PhD thesis, Cornell University.

Baraff, D. 1994. Fast contact force computation for nonpenetrating rigid bodies. In *Proceedings of SIGGRAPH '94*, A. Glassner, Ed., ACM SIGGRAPH, 23–34. ISBN 0-89791-667-0.

Basdogan, C., Ho, C.-H., and Srinivasan, M. A. 1997. A ray-based haptic rendering technique for displaying shape and texture of 3d objects in virtual environments. *DSC-Vol. 61, Proceedings of the ASME Dynamic Systems and Control Division*, pp. 77–84.

Beckmann, N., Kriegel, H., Schneider, R., and Seeger, B. 1990. The r*-tree: An efficient and robust access method for points and rectangles. *Proc. SIGMOD Conf. on Management of Data*, 322–331.

Bejczy, A., and Salisbury, J. K. 1980. Kinematic coupling between operator and remote manipulator. *Advances in Computer Technology Vol. 1*, 197–211.

Berkelman, P. J. 1999. *Tool-Based Haptic Interaction with Dynamic Physical Simulations Using Lorentz Magnetic Levitation.* PhD thesis, Carnegie Mellon University.

Blinn, J. F. 1978. Simulation of wrinkled surfaces. In *Computer Graphics (SIGGRAPH '78 Proceedings)*, 286–292.

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.

Burdea, G. 1996. *Force and Touch Feedback for Virtual Reality.* John Wiley and Sons.

Cameron, S., and Culley, R. K. 1986. Determining the minimum translational distance between two convex polyhedra. *Proceedings of International Conference on Robotics and Automation*, 591–596.

Cameron, S. 1997. Enhancing GJK: Computing minimum and penetration distance between convex polyhedra. *IEEE International Conference on Robotics and Automation*, 3112–3117.

Catmull, E. E. 1974. *A Subdivision Algorithm for Computer Display of Curved Surfaces.* PhD thesis, Dept. of CS, U. of Utah.

Çavuşoğlu, M. C., and Tendick, F. 2000. Multirate simulation for high fidelity haptic interaction with deformable objects in virtual environments. *Proc. of IEEE International Conference on Robotics and Automation*, 2458–2465.

Çavuşoğlu, M. C., Tendick, F., and Sastry, S. S. 2002. Haptic interfaces to real and virtual surgical environments. In *Touch in Virtual Environments*, M. L. McLaughlin, J. P. Hespanha, and G. S. Sukhatme, Eds. Prentice Hall PTR, Upper Saddle River, NJ, ch. 13, 217–237.

Chang, B., and Colgate, J. E. 1997. Real-time impulse-based simulation of rigid body systems for haptic display. *Proc. of ASME Dynamic Systems and Control Division*.

Chen, E. 1999. Six degree-of-freedom haptic system for desktop virtual prototyping applications. In *Proceedings of the First International Workshop on Virtual Reality and Prototyping*, 97–106.

Choi, S., and Tan, H. Z. 2003. Aliveness: Perceived instability from a passive haptic texture rendering system. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Choi, S., and Tan, H. Z. 2003. An experimental study of perceived instability during haptic texture rendering: Effects of collision detection algorithm. *Proc. of Haptics Symposium*, 197–204.

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

Colgate, J. E., and Schenkel, G. G. 1994. Passivity of a class of sampled-data systems: Application to haptic interfaces. *Proc. of American Control Conference*.

Colgate, J. E., Grafing, P. E., Stanley, M. C., and Schenkel, G. 1993. Implementation of stiff virtual walls in force-reflecting interfaces. *Virtual Reality Annual International Symposium*, 202–207.

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

Connor, C. E., and Johnson, K. O. 1992. Neural coding of tactile texture: Comparison of spatial and temporal mechanisms for roughness perception. *Journal of Neuroscience 12*, pp. 3414–3426.

Constantinescu, D., Salcudean, S. E., and Croft, E. A. 2004. Impulsive forces for haptic rendering of rigid contact. *Proc. of International Symposium on Robotics*, 1–6.

Cottle, R. W., and Dantzig, G. B. 1968. Complementarity pivot theory of mathematical programming. *Linear Algebra and its Applications*, 103–125.

Cottle, R. W., pang, J. S., and Stone, R. E. 1992. The linear complementarity problem. *Academic-Press, Inc.*.

Dachille, F., Qin, H., , Kaufman, A., and El-Sana, J. 1999. Haptic sculpting of dynamic surfaces. *Proc. of ACM Symposium on Interactive 3D Graphics*, 103–110.

Debunne, G., Desbrun, M., Cani, M. P., and Barr, A. H. 2001. Dynamic real-time deformations using space and time adaptive sampling. *Proc. of ACM SIGGRAPH*.

Dobkin, D. P., and Kirkpatrick, D. G. 1990. Determining the separation of preprocessed polyhedra – a unified approach. In *Proc. 17th Internat. Colloq. Automata Lang. Program.*, Springer-Verlag, vol. 443 of *Lecture Notes in Computer Science*, 400–413.

Dobkin, D., Hershberger, J., Kirkpatrick, D., and Suri, S. 1993. Computing the intersection-depth of polyhedra. *Algorithmica 9*, 518–533.

Duriez, C., Andriot, C., and Kheddar, A. 2004. A multi-threaded approach for deformable/rigid contacts with haptic feedback. *Proc. of Haptics Symposium.*

Eck, M., DeRose, T., Duchamp, T., Hoppe, H., Lounsbery, M., and Stuetzle, W. 1995. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH 95 Conference Proceedings*, Addison Wesley, R. Cook, Ed., Annual Conference Series, ACM SIGGRAPH, 173–182. held in Los Angeles, California, 06-11 August 1995.

Edmond, C., Heskamp, D., Sluis, D., Stredney, D., Wiet, G., Yagel, R., Weghorst, S., Oppenheimer, P., Miller, J., Levin, M., and Rosenberg, L. 1997. Ent endoscopic surgical simulator. *Proc. of Medicine Meets VR*, 518–528.

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*, 2101–2106.

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, 500–510.

El-Sana, J., and Varshney, A. 2000. Continuously-adaptive haptic rendering. *Virtual Environments 2000*, pp. 135–144.

Ellis, R. E., Sarkar, N., and Jenkins, M. A. 1997. Numerical methods for the force reflection of contact. *ASME Transactions on Dynamic Systems, Modeling and Control 119*, 768–774.

Ernst, M. O., and Banks, M. S. 2001. Does vision always dominate haptics? *Touch in Virtual Environments Conference*.

Fisher, S., and Lin, M. C. 2001. Fast penetration depth estimation for elastic bodies using deformed distance fields. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems 2001*.

Fisher, B., Fels, S., MacLean, K., Munzner, T., and Rensink, R. 2004. Seeing, hearing and touching: Putting it all together. In *ACM SIGGRAPH course notes.*

Foskey, M., Otaduy, M. A., and Lin, M. C. 2002. ArtNova: Touch-enabled 3D model design. *Proc. of IEEE Virtual Reality Conference*.

Gibson, S., Samosky, J., Mor, A., Fyock, C., Grimson, E., and Kanade, T. 1997. Simulating arthroscopic knee surgery using volumetric object representations, real-time volume rendering and haptic feedback. *First Joint Conference on Computer Vision, Virtual Reality and Robotics in Medicine and Medical Robotics and Computer-Assisted Surgery (CVMed-MRCAS)*, 368–378.

Gilbert, E. G., Johnson, D. W., and Keerthi, S. S. 1988. A fast procedure for computing the distance between objects in three-dimensional space. *IEEE J. Robotics and Automation vol RA-4*, 193–203.

Goertz, R., and Thompson, R. 1954. Electronically controlled manipulator. *Nucleonics*, 46–47.

Gottschalk, S., Lin, M., and Manocha, D. 1996. OBB-Tree: A hierarchical structure for rapid interference detection. *Proc. of ACM Siggraph'96*, 171–180.

Gottschalk, S. 2000. *Collision Queries using Oriented Bounding Boxes.* PhD thesis, University of North Carolina. Department of Computer Science.

GOVINDARAJU, N., REDON, S., LIN, M., AND MANOCHA, D. 2003. CULLIDE: Interactive collision detection between complex models in large environments using graphics hardware. *Proc. of ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware*, 25–32.

GREGORY, A., LIN, M., GOTTSCHALK, S., AND TAYLOR, R. 1999. H-COLLIDE: A framework for fast and accurate collision detection for haptic interaction. In *Proceedings of Virtual Reality Conference 1999*, 38–45.

GREGORY, A., EHMANN, S., AND LIN, M. C. 2000. *inTouch*: Interactive multiresolution modeling and 3d painting with a haptic interface. *Proc. of IEEE VR Conference*.

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*.

GUENDELMAN, E., BRIDSON, R., AND FEDKIW, R. 2003. Nonconvex rigid bodies with stacking. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH) 22*, 871–878.

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

HANNAFORD, B., RYU, J.-H., AND KIM, Y. S. 2002. Stable control of haptics. In *Touch in Virtual Environments*, M. L. McLaughlin, J. P. Hespanha, and G. S. Sukhatme, Eds. Prentice Hall PTR, Upper Saddle River, NJ, ch. 3, 47–70.

HASEGAWA, S., AND SATO, M. 2004. Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects. In *Proc. of Eurographics*.

HAYWARD, V., AND ARMSTRONG, B. 2000. A new computational model of friction applied to haptic rendering. *Experimental Robotics VI*.

HAYWARD, V., GREGORIO, P., ASTLEY, O., GREENISH, S., AND DOYON, M. 1998. Freedom-7: A high fidelity seven axis haptic device with applications to surgical training. *Experimental Robotics*, 445–456. Lecture Notes in Control and Information Sciences 232.

HELLER, M. A., CALCATERRA, J. A., GREEN, S. L., AND BROWN, L. 1999. Intersensory conflict between vision and touch: The response modality dominates when precise, attention-riveting judgements are required. *Perception and Psychophysics 61*, pp. 1384–1398.

HILL, J. W., AND SALISBURY, J. K. 1977. Two measures of performance in a peg-in-hole manipulation task with force feedback. *Thirteenth Annual Conference on Manual Control, MIT*.

HO, C.-H., BASDOGAN, C., AND SRINIVASAN, M. A. 1999. Efficient point-based rendering techniques for haptic display of virtual objects. *Presence 8*, 5, pp. 477–491.

HOFF, K., ZAFERAKIS, A., LIN, M., AND MANOCHA, D. 2001. Fast and simple 2d geometric proximity queries using graphics hardware. *Proc. of ACM Symposium on Interactive 3D Graphics*, 145–148.

HOGAN, N. 1985. Impedance control: An approach to manipulation, part i - theory, part ii - implementation, part iii - applications. *Journal of Dynamic Systems, Measurement and Control 107*, 1–24.

HOGAN, N. 1986. Multivariable mechanics of the neuromuscular system. *IEEE Annual Conference of the Engineering in Medicine and Biology Society*, 594–598.

HOLLINS, M., AND RISNER, S. 2000. Evidence for the duplex theory of tactile texture perception. *Perception & Psychophysics 62*, 695–705.

Hoppe, H. 1997. View dependent refinement of progressive meshes. In *ACM SIGGRAPH Conference Proceedings*, 189–198.

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

Insko, B., Meehan, M., Whitton, M., and Brooks, F. 2001. Passive haptics significantly enhances virtual environments. Tech. Rep. 01-010, Department of Computer Science, UNC Chapel Hill.

Insko, B. 2001. *Passive Haptics Significantly Enhance Virtual Environments*. PhD thesis, University of North Carolina. Department of Computer Science.

James, D. L., and Pai, D. K. 2004. Bd-tree: Output-sensitive collision detection for reduced deformable models. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*.

Johnson, D. E., and Cohen, E. 2001. Spatialized normal cone hierarchies. *Proc. of ACM Symposium on Interactive 3D Graphics*, pp. 129–134.

Johnson, D. E., and Willemsen, P. 2003. Six degree of freedom haptic rendering of complex polygonal models. In *Proc. of Haptics Symposium*.

Johnson, D. E., and Willemsen, P. 2004. Accelerated haptic rendering of polygonal models through local descent. *Proc. of Haptics Symposium*.

Johnson, D., Thompson II, T. V., Kaplan, M., Nelson, D., and Cohen, E. 1999. Painting textures with a haptic interface. *Proceedings of IEEE Virtual Reality Conference*.

Karnopp, D. 1985. Computer simulation of stick slip friction in mechanical dynamic systems. *Trans. ASME, Journal of Dynamic Systems, Measurement, and Control*.

Katz, D. 1989. *The World of Touch*. Erlbaum, Hillsdale, NJ. L. Krueger, Trans. (Original work published 1925).

Kilpatrick, P. J. 1976. *The use of a kinesthetic supplement in an interactive graphics system*. Ph.d. thesis, The University of North Carolina at Chapel Hill.

Kim, W., and Bejczy, A. 1991. Graphical displays for operator aid in telemanipulation. *IEEE International Conference on Systems, Man and Cybernetics*.

Kim, Y. J., Lin, M. C., and Manocha, D. 2002. DEEP: an incremental algorithm for penetration depth computation between convex polytopes. *Proc. of IEEE Conference on Robotics and Automation*, 921–926.

Kim, Y. J., Lin, M. C., and Manocha, D. 2002. Fast penetration depth computation using rasterization hardware and hierarchical refinement. *Proc. of Workshop on Algorithmic Foundations of Robotics*.

Kim, Y. J., Otaduy, M. A., Lin, M. C., and Manocha, D. 2003. Six-degree-of-freedom haptic rendering using incremental and localized computations. *Presence 12*, 3, 277–295.

Klatzky, R. L., and Lederman, S. J. 1995. Identifying objects from a haptic glance. *Perception and Psychophysics 57*, pp. 1111–1123.

Klatzky, R. L., and Lederman, S. J. 1999. Tactile roughness perception with a rigid link interposed between skin and surface. *Perception and Psychophysics 61*, pp. 591–607.

Klatzky, R. L., and Lederman, S. J. 2002. Perceiving texture through a probe. In *Touch in Virtual Environments*, M. L. McLaughlin, J. P. Hespanha, and G. S. Sukhatme, Eds. Prentice Hall PTR, Upper Saddle River, NJ, ch. 10, 180–193.

Klatzky, R. L., and Lederman, S. J. 2003. Touch. In *Experimental Psychology*, 147–176. Volume 4 in I.B. Weiner (Editor-in-Chief). Handbook of Psychology.

Klatzky, R. L., Lederman, S. J., Hamilton, C., Grindley, M., and Swendsen, R. H. 2003. Feeling textures through a probe: Effects of probe and surface geometry and exploratory factors. *Perception and Psychophysics 65(4)*, pp. 613–631.

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.

Kuhnapfel, U. G., Kuhn, C., Hubner, M., Krumm, H.-G., Maass, H., and Neisius, B. 1997. The karlsruhe endoscopic surgery trainer as an example for virtual reality in medical education. *Minimally Invasive Therapy and Allied Technologies Vol. 6*, 122–125.

LaMotte, R. H., and Srinivasan, M. A. 1991. Surface microgeometry: Tactile perception and neural encoding. In *Information Processing in the Somatosensory System*, O. Franzen and J. Westman, Eds. Macmillan Press, London, 49–58.

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*.

Larsen, E. 2001. A robot soccer simulator: A case study for rigid body contact. *Game Developers Conference*.

Lederman, S. J., Klatzky, R. L., Hamilton, C., and Ramsay, G. I. 1999. Perceiving roughness via a rigid stylus: Psychophysical effects of exploration speed and mode of touch. *Haptics-e*.

Lederman, S. J., Klatzky, R. L., Hamilton, C., and Grindley, M. 2000. Perceiving surface roughness through a probe: Effects of applied force and probe diameter. *Proceedings of the ASME DSCD-IMECE*.

Lederman, S. J. 1974. Tactile roughness of grooved surfaces: The touching process and the effects of macro- and microsurface structure. *Perception and Psychophysics 16*, pp. 385–395.

Lemke, C. E. 1965. Bimatrix equilibrium points and mathematical programming. *Management Science 11*, 681–689.

Lin, M. C., and Canny, J. F. 1991. Efficient algorithms for incremental distance computation. In *Proc. IEEE Internat. Conf. Robot. Autom.*, vol. 2, 1008–1014.

Lin, M., and Gottschalk, S. 1998. Collision detection between geometric models: A survey. *Proc. of IMA Conference on Mathematics of Surfaces*.

Lin, M. C., and Manocha, D. 2004. Collision and proximity queries. In *Handbook of Discrete and Computational Geometry, 2nd Ed.*, J. E. Goodman and J. O'Rourke, Eds. CRC Press LLC, Boca Raton, FL, ch. 35, 787–807.

Lin, M. 1993. *Efficient Collision Detection for Animation and Robotics.* PhD thesis, Department of Electrical Engineering and Computer Science, University of California, Berkeley.

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

LÖTSTEDT, P. 1984. Numerical simulation of time-dependent contact friction problems in rigid body mechanics. *SIAM Journal of Scientific Statistical Computing 5*, 370–393.

LOUNSBERY, M., DEROSE, T. D., AND WARREN, J. 1997. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph. 16*, 1 (Jan.), 34–73.

LUEBKE, D., AND ERIKSON, C. 1997. View-dependent simplification of arbitrary polygon environments. In *Proc. of ACM SIGGRAPH.*

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. In *SIGGRAPH 96 Conference Proceedings*, H. Rushmeier, Ed., Annual Conference Series, 447–452.

MASSIE, T. M., AND SALISBURY, J. K. 1994. The phantom haptic interface: A device for probing virtual objects. *Proc. of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems 1*, 295–301.

MCDONNELL, K., QIN, H., AND WLODARCZYK, R. 2001. Virtual clay: A real-time sculpting system with haptic interface. *Proc. of ACM Symposium on Interactive 3D Graphics*, 179–190.

MCLAUGHLIN, M., HESPANHA, J. P., AND SUKHATME, G. S. 2002. *Touch in Virtual Environments.* Prentice Hall.

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

MILENKOVIC, V. J., AND SCHMIDL, H. 2001. Optimization-based animation. *SIGGRAPH 01 Conference Proceedings*, 37–46.

MILLER, B. E., COLGATE, J. E., AND FREEMAN, R. A. 1990. Guaranteed stability of haptic systems with nonlinear virtual environments. *IEEE Transactions on Robotics and Automation 16*, 6, 712–719.

MINSKY, M., OUH-YOUNG, M., STEELE, O., BROOKS, JR., F. P., AND BEHENSKY, M. 1990. Feeling and seeing: Issues in force display. In *Computer Graphics (1990 Symposium on Interactive 3D Graphics)*, R. Riesenfeld and C. Sequin, Eds., vol. 24, 235–243.

MINSKY, M. 1995. *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display.* PhD thesis, Ph.D. Dissertation, Program in Media Arts and Sciences, MIT. Thesis work done at UNC-CH Computer Science.

MIRTICH, B., AND CANNY, J. 1995. Impulse-based simulation of rigid bodies. In *Symposium on Interactive 3D Graphics*, ACM Press.

MIRTICH, B. V. 1996. *Impulse-based Dynamic Simulation of Rigid Body Systems.* PhD thesis, University of California, Berkeley.

MIRTICH, B. V. 1998. Rigid body contact: Collision detection to force computation. Tech. Rep. TR98-01, Mitsubishi Electric Research Laboratory.

Mirtich, B. 1998. V-Clip: Fast and robust polyhedral collision detection. *ACM Transactions on Graphics 17*, 3 (July), 177–208.

Mirtich, B. 2000. Timewarp rigid body simulation. *SIGGRAPH 00 Conference Proceedings*, 193–200.

Moore, M., and Wilhelms, J. 1988. Collision detection and response for computer animation. In *Computer Graphics (SIGGRAPH '88 Proceedings)*, J. Dill, Ed., vol. 22, 289–298.

Nelson, D. D., Johnson, D. E., and Cohen, E. 1999. Haptic rendering of surface-to-surface sculpted model interaction. *Proc. of ASME Dynamic Systems and Control Division*.

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

Okamura, A. M., and Cutkosky, M. R. 2001. Feature detection for haptic exploration with robotic fingers. *International Journal of Robotics Research 20*, 12, 925–938.

O'Sullivan, C., and Dingliana, J. 2001. Collisions and perception. *ACM Trans. on Graphics 20*, 3, pp. 151–168.

O'Sullivan, C., Radach, R., and Collins, S. 1999. A model of collision perception for real-time animation. *Computer Animation and Simulation*, pp. 67–76.

O'Sullivan, C., Dingliana, J., Giang, T., and Kaiser, M. K. 2003. Evaluating the visual fidelity of physically based animations. *Proc. of ACM SIGGRAPH*, 527–536.

Otaduy, M. A., and Lin, M. C. 2003. CLODs: Dual hierarchies for multiresolution collision detection. *Eurographics Symposium on Geometry Processing*, 94–101.

Otaduy, M. A., and Lin, M. C. 2003. Sensation preserving simplification for haptic rendering. *ACM Trans. on Graphics (Proc. of ACM SIGGRAPH)*, 543–553.

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

Otaduy, M. A., Jain, N., Sud, A., and Lin, M. C. 2004. Haptic display of interaction between textured models. *Proc. of IEEE Visualization*, 297–304.

Ouh-Young, M. 1990. *Force Display in Molecular Docking.* PhD thesis, University of North Carolina, Computer Science Department.

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

Pai, D. K., van den Doel, K., James, D. L., Lang, J., Lloyd, J. E., Richmond, J. L., and Yau, S. H. 2001. Scanning physical interaction behavior of 3d objects. *Computer Graphics (Proc. of ACM SIGGRAPH)*.

Quinlan, S. 1994. Efficient distance computation between non-convex objects. In *Proceedings of International Conference on Robotics and Automation*, 3324–3329.

Redon, S., Kheddar, A., and Coquillart, S. 2002. Fast continuous collision detection between rigid bodies. *Proc. of Eurographics (Computer Graphics Forum)*.

Renz, M., Preusche, C., Pötke, M., Kriegel, H.-P., and Hirzinger, G. 2001. Stable haptic interaction with virtual environments using an adapted voxmap-pointshell algorithm. *Eurohaptics Conference*.

Rock, I., and Victor, J. 1964. Vision and touch: An experimentally created conflict between the two senses. *Science 143*, pp. 594–596.

Ruspini, D., and Khatib, O. 2000. A framework for multi-contact multi-body dynamic simulation and haptic display. *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Ruspini, D., Kolarov, K., and Khatib, O. 1997. The haptic display of complex graphical environments. *Proc. of ACM SIGGRAPH*, 345–352.

Salcudean, S. E., and Vlaar, T. D. 1994. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. *Proc. of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems*.

Salisbury, K., Brock, D., Massie, T., Swarup, N., and Zilles, C. 1995. Haptic rendering: Programming touch interaction with virtual objects. In *1995 Symposium on Interactive 3D Graphics*, P. Hanrahan and J. Winget, Eds., ACM SIGGRAPH, 123–130. ISBN 0-89791-736-7.

Seidel, R. 1990. Linear programming and convex hulls made easy. In *Proc. 6th Ann. ACM Conf. on Computational Geometry*, 211–215.

Shimoga, K. 1992. Finger force and touch feedback issues in dextrous manipulation. *NASA-CIRSSE International Conference on Inetelligent Robotic Systems for Space Exploration*.

Siira, J., and Pai, D. K. 1996. Haptic textures – a stochastic approach. *Proc. of IEEE International Conference on Robotics and Automation*, 557–562.

Slater, M., and Usoh, M. 1993. An experimental exploration of presence in virtual environments. Tech. Rep. 689, Department of Computer Science, University College London.

Spence, C., Pavani, F., and Driver, J. 2000. Crossmodal links between vision and touch in covert endogenous spatial attention. *Journal of Experimental Psychology: Human Perception and Performance 26*, pp. 1298–1319.

Stewart, D. E., and Trinkle, J. C. 1996. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering 39*, 2673–2691.

Stewart, D. E., and Trinkle, J. C. 2000. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. *IEEE International Conference on Robotics and Automation*, 162–169.

Stollnitz, E., DeRose, T., and Salesin, D. 1996. *Wavelets for Computer Graphics: Theory and Applications.* Morgan-Kaufmann.

Sutherland, I. 1965. The ultimate display. *Proc. of IFIP*, 506–508.

Taylor, R. M., Robinett, W., Chii, V., Brooks, F., and Wright, W. 1993. The nanomanipulator: A virtual-reality interface for a scanning tunneling microscope. In *Proc. of ACM SIGGRAPH*, 127–134.

Thompson, T. V., Johnson, D., and Cohen, E. 1997. Direct haptic rendering of sculptured models. *Proc. of ACM Symposium on Interactive 3D Graphics*, pp. 167–176.

Unger, B. J., Nicolaidis, A., Berkelman, P. J., Thompson, A., Lederman, S. J., Klatzky, R. L., and Hollis, R. L. 2002. Virtual peg-in-hole performance using a 6-dof magnetic levitation haptic device: Comparison with real forces and with visual guidance alone. *Proc. of Haptics Symposium*, 263–270.

van den Bergen, G. 2001. Proximity queries and penetration depth computation on 3d game objects. *Game Developers Conference*.

Wan, M., and McNeely, W. A. 2003. Quasi-static approximation for 6 degrees-of-freedom haptic rendering. *Proc. of IEEE Visualization*, 257–262.

Wu, D. 2000. Penalty methods for contact resolution. *Game Developers Conference*.

Zilles, C., and Salisbury, K. 1995. A constraint-based god object method for haptics display. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robotics and Systems*.

Zorin, D., Schröder, P., and Sweldens, W. 1997. Interactive multiresolution mesh editing. *Proc. of ACM SIGGRAPH*, 259–268.