DSC-3-4

HAPTIC RENDERING OF SURFACE-TO-SURFACE SCULPTED MODEL INTERACTION

Donald D. Nelson David E. Johnson Elaine Cohen
University of Utah, Computer Science Dept.
50 S Central Campus Dr Rm 3190
Salt Lake City, UT 84112-9205
Email: {dnelson,dejohnso,cohen}@cs.utah.edu

ABSTRACT

Previous work in haptics surface tracing for virtual prototyping and surface design applications has used a point model for virtual finger-surface interaction. We extend this tracing method for surface-to-surface interactions. A straightforward extension of the point-surface formulation to surface-surface can yield extraneous, undesirable solutions, although we rework the formulation to yield more satisfactory solutions. Additionally, we derive an alternative novel velocity formulation for use in a surface-surface tracing paradigm that exhibits additional stability beyond the Newton methods. Both methods require evaluating the surface point and first and second surface partial derivatives for both surfaces, an efficient kilohertz rate computation. These methods are integrated into a three step tracking process that uses a global minimum distance method, the local Newton formulation, and the new velocity formulation.

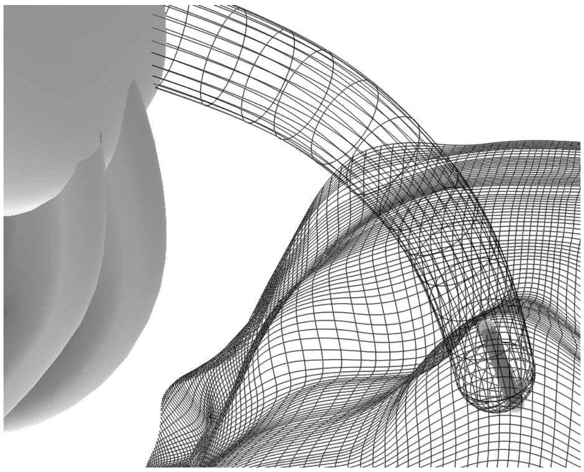

Figure 1: Well-behaved finger penetration into a surface shown by the “penetration cylinder”. The velocity method and modified Newton method return the maximum distance between the two surfaces upon penetration.

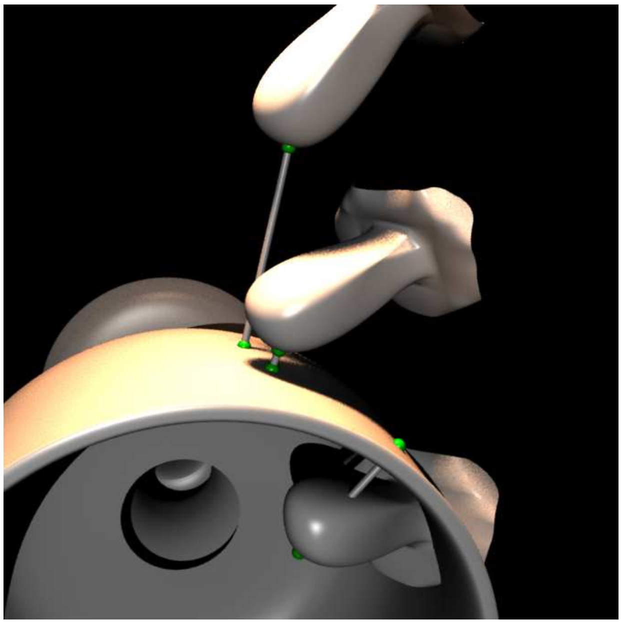

Figure 2: Virtual proxies are important in the penetrating case. The maximal distance is required by the haptics tracing algorithm. Global solution discontinuities such as “chopping through” an object are not desirable. Because the velocity formulation (shown) is the “most local,” it is the method of choice for the penetrating case.

1 INTRODUCTION

User interactions with surfaces with force feedback are an important design and visualization tool. Current work has mostly utilized a point position hand model for interaction, but a desirable extension is to allow more realistic geometry for the hand model. However, this extension creates some severe geometric computation challenges, as well much more complicated contact and response scenarios.

In this paper, we address one portion of the surface-surface haptic rendering problem, namely, computation of proper penetration depth between two surfaces. Once the penetration vector has been obtained, rendering force feedback will be done with established haptics techniques [Thompson, 1997].

This penetration depth computation will be placed within a framework for reliably finding and tracking multiple contact points between models. This framework breaks the haptic rendering problem into several phases — distant tracking using global minimum distance methods, nearby tracking using local Newton methods, and tracking during contact using a reformulated Newton's method or a novel velocity formulation.

1.1 Distance Extrema

Following [Baraff, 1990], the extremal distance may be defined as the minimum distance between the two models when they are disjoint, zero during tangential contact, and the locally maximum penetration depth when they are inter-penetrated. This measure is related to the minimum translation distance defined by Cameron [Cameron, 1997].

When a user touches a virtual surface with his virtual finger model, a curve $\gamma(t)$ embedded on the finger surface $\mathbf{f}$ and a curve $\zeta(t)$ on the embedded on a model surface $\mathbf{g}$ define the path of distance extrema required by the haptics tracing algorithm.

$$distance\ extrema = ||\mathbf{f}(\gamma(t)) - \mathbf{g}(\zeta(t))|| \tag{1}$$

Our goal is to find the piecewise continuous curves $\gamma(t)$ and $\zeta(t)$ for penetration depth computations in a real-time haptics tracing environment.

When the finger model is penetrated into the CAD model, the maximal distance is required for force computations as shown in Fig.1. When the finger is not penetrating the surface the curves γ and ζ may be discontinuous since the distance between surface models is non-differentiable in general. Both surfaces may be moving. A minimal distance measure between surfaces is useful in this case for a global monitoring and restarting mechanism.

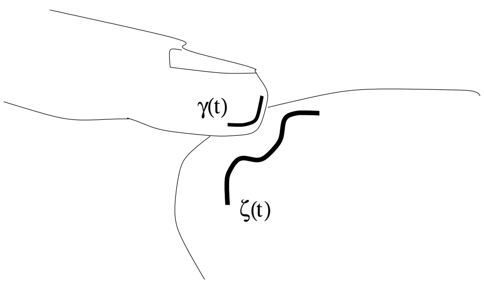

Figure 3: Curves of distance extrema embedded into the finger surface model and the CAD model.

2 BACKGROUND

The robotics community has considerable literature in the area of finding the minimal distance between a point and model or between model and model [Quinlan, 1994, Gilbert, 1988, Lin, 1994]. However, the minimum distance between models is zero during penetration; we desire the penetration depth for haptic force computations. The minimum distance between parametric surfaces $\mathbf{f}(u,v)$ and $\mathbf{g}(s,t)$ may be described by the following system of equations

$$(\mathbf{f}(u,v) - \mathbf{g}(s,t)) \cdot \mathbf{f}_u = 0 \tag{2}$$
$$(\mathbf{f}(u,v) - \mathbf{g}(s,t)) \cdot \mathbf{f}_v = 0 \tag{3}$$
$$(\mathbf{f}(u,v) - \mathbf{g}(s,t)) \cdot \mathbf{g}_s = 0 \tag{4}$$
$$(\mathbf{f}(u,v) - \mathbf{g}(s,t)) \cdot \mathbf{g}_t = 0 \tag{5}$$

which correctly describes that the line between the closest points is normal to each surface. This system has been solved with a search over the four-dimensional parameter space [Snyder, 1995], with global, high-dimensional resultant methods [Lin, 1994], and Euclidean space bounding methods [Johnson, 1998a].

Note that in haptic applications, we are often interested in the local solution to distance. Using a local solution constrains the force computation to a continuous solution, similar to a "God-object" [Zilles, 1995] or virtual proxy [Ruspini, 1997] as used in polygonal methods (see Fig.2).

The assumption that the situation upon penetration is local for haptics tracing is fortunate because the haptics controller often requires the greatest update rates precisely at the impact or penetrating event. Greater stability is achieved with the high update rates.

Local solution methods have been applied to point - surface analogs of Equations 2-5. A first-order solution to the minimum distance is described in [Thompson, 1997] and extended to a Newton formulation in [Johnson, 1998b, Stewart, 1997]. For a parametric surface $\mathbf{f}(u,v)$ and point $\mathbf{P}$, we can describe the minimum distance constraint equations as

$$(\mathbf{f}(u,v) - \mathbf{P}) \cdot \mathbf{f}_u = 0 \tag{6}$$
$$(\mathbf{f}(u,v) - \mathbf{P}) \cdot \mathbf{f}_v = 0. \tag{7}$$

Offset surfaces have recently been used to extend the point-surface finger model with a sphere-surface model [Ruspini, 1997, Thompson, 1999]. A surface that is offset from an original surface by a constant amount can be traced with the point-surface model. When the original surface is displayed, the point model is effectively performing sphere-surface tracing. A sphere is still a very simple finger model; rotating the finger has no effect. We develop a method in this work that allows more complicated finger models.

3 APPROACH

We have broken the haptic rendering of surface-surface interactions into three phases. In the first phase, the far phase, a global monitoring mechanism returns all portions of a surface within some distance of some part of the other surface. In the second phase, the near phase, local Newton methods determine the closest points between all portions of the surfaces returned from the far phase. The third phase, the penetration phase, uses these closest points to initiate a velocity formulation that maintains the proper penetration depth vector between surfaces.

3.1 Global Minimum Distance

The global minimum distance mechanism depends on the subdivision properties of the surface. For NURBS surfaces, the surface is made up of a patchwork of polynomial pieces. Each polynomial piece is contained with the convex hull of its defining control points. These pieces may be refined, such that each piece splits into several pieces that maintain the original surface shape, yet have additional control points. These additional control points converge quadratically to the surface as the refinement is increased.

We can exploit these properties to prune away portions of surfaces that are further away than some threshold, refine the remaining portions, and repeat [Johnson, 1998a]. Remaining areas may be used to find approximate minimum distances to initiate faster, local, Newton methods.

3.2 Local Minimum Distance

The local minimum distance phase uses Newton methods to quickly update the minimum distance between portions of the model. We use a reformulated extremal distance approach described in the following section for extra stability relative to the standard minimum distance formulation from the minimum distance formulation of Equations 2-5.

3.3 Local Penetration

We have developed two methods for maintaining the proper penetration depth vector for surface-surface interactions. The first is a reformulation of the Newton minimum distance method. The second is a velocity formulation. The Newton method has the advantage of being able to find the extremal distance given nearby starting locations on the surfaces. The velocity formulation is able to maintain the proper extremal distance and shows stability beyond that of the reformulated Newton method.

3.4 Extremal Distance Representation

The minimum distance equations 2-5 are not adequate to properly describe the extremal distance needed for surface-surface haptics. Along with a root at the extremal distance, a set of roots occurs along the curve of intersection between the two surfaces, where $(\mathbf{f}(u,v)-\mathbf{g}(s,t))$ goes to zero. The local methods may very easily "slide" into these solutions. Our reworked formulations avoid the zero distance roots.

The following methods are applicable to any parametric surface representation, including NURBS and subdivision surfaces, the most commonly used representations. Let $\mathbf{u} = \begin{bmatrix} u & v & s & t \end{bmatrix}^T$ designate the closest parametric contact coordinates between any two parametric surfaces $\mathbf{f}(u,v)$ and $\mathbf{g}(s,t)$. $\mathbf{f}$ and $\mathbf{g}$ denote surface evaluations, or mappings from parametric space to Cartesian space.

3.5 Newton Extremal Distance Formulation

The extremal distance between parametric surfaces $\mathbf{f}(u,v)$ and $\mathbf{g}(s,t)$ may be described by the following equation:

$$\mathbf{E}(u,v,s,t) = (\mathbf{f}(u,v)-\mathbf{g}(s,t))\cdot\mathbf{N} \tag{8}$$

where $\mathbf{N}$ is the surface normal of $\mathbf{f}$ at (u,v). We wish to find the extrema of $\mathbf{E}$, which may be found at simultaneous zeros of its partials. The partials are

$$\mathbf{f}_u\cdot\mathbf{N} + (\mathbf{f}(u,v)-\mathbf{g}(s,t))\cdot\mathbf{N}_u = 0 \tag{9}$$
$$\mathbf{f}_v\cdot\mathbf{N} + (\mathbf{f}(u,v)-\mathbf{g}(s,t))\cdot\mathbf{N}_v = 0 \tag{10}$$
$$-\mathbf{g}_s\cdot\mathbf{N} = 0 \tag{11}$$
$$-\mathbf{g}_t\cdot\mathbf{N} = 0. \tag{12}$$

Noting that the normal $\mathbf{N}$ is orthogonal to the tangent plane formed by the partials $\mathbf{f}_u$ and $\mathbf{f}_v$, we may remove the $\mathbf{f}_u\cdot\mathbf{N}$ and $\mathbf{f}_v\cdot\mathbf{N}$ terms. Additionally, the partials of $\mathbf{N}$ lie in the tangent plane of $\mathbf{f}$. The equivalent constraint may be formulated by replacing these partials with the partials of $\mathbf{f}$. These substitutions form a simplified set of equations.

$$\mathbf{N}\cdot\mathbf{g}_s = 0 \tag{13}$$
$$\mathbf{N}\cdot\mathbf{g}_t = 0 \tag{14}$$
$$(\mathbf{f}-\mathbf{g})\cdot\mathbf{f}_u = 0 \tag{15}$$
$$(\mathbf{f}-\mathbf{g})\cdot\mathbf{f}_v = 0. \tag{16}$$

The first two equations constrain the solution to collinear normals and the second two maintain collinearity of the closest points with the surface normals. This set of equations is analogous to those used in [Baraff, 1990] and [Snyder, 1995], however, we have expressed them in a form more suitable to the demands of haptic rate computation.

This system of equations may be locally solved through incremental updates of $\mathbf{u}$ using multi-dimensional Newton's method,

$$\Delta\mathbf{u} = \mathbf{J}^{-1}(-\mathbf{F}) \tag{17}$$

where $\mathbf{F}$ is the constraint violation defined by Eqs. 13-16, and $\mathbf{J}$ is the Jacobian of $\mathbf{F}$.

This formulation may result in extraneous zeros, since there may be multiple locations where the surfaces' tangent planes are parallel and are at a local distance extrema. However, these undesired roots are typically at polar opposites of the model and are less common than for the minimum distance formulation.

3.6 Velocity Extremal Distance Formulation

A different approach in the penetrating case is to take surface velocity into account. The relation of parametric contact differentials with the relative linear and angular velocity between the two surfaces can be used to provide incremental tracing updates. In the Appendix, the authors have extended the results of [Cremer, 1996, Montana, 1986] to arbitrary surface parameterizations, i.e. the contact velocity relations have been extended to surfaces whose partials are not everywhere perpendicular, making the relations useful for common models. The parametric contact coordinates $\mathbf{u}$ may be integrated through time using the following relation, derived in the Appendix,

$$\dot{\mathbf{u}} = \mathbf{A}\begin{bmatrix} \mathbf{v}_{x,y} \\ \omega_{x,y} \end{bmatrix} \tag{18}$$

where $[\mathbf{v}^T, \omega^T]^T$ are the relative linear and angular velocity between the surfaces relative to the frame $\mathbf{x}_g = \mathbf{g}_u/||\mathbf{g}_u||$, $\mathbf{z}_g = \mathbf{g}_u\times\mathbf{g}_v/||\mathbf{g}_u\times\mathbf{g}_v||$, and $\mathbf{y}_g = \mathbf{z}_g\times\mathbf{x}_g$ located at the contact point $\mathbf{g}(\mathbf{u})$ (Fig.6), and

$$\mathbf{A} = \begin{bmatrix} \mathbf{R}_\theta(\mathbf{E}^f - \beta\mathbf{F}^f_{-y,x}) & -\mathbf{E}^g \\ -(\mathbf{R}_\theta\mathbf{F}^f) & -\mathbf{F}^g \end{bmatrix}^{-1} \tag{19}$$

$$\mathbf{E}^f = \left[\begin{bmatrix} \mathbf{x}_f & \mathbf{y}_f \end{bmatrix}^T \mathbf{f}_\mathbf{u}\right] \tag{20}$$

$$\mathbf{F}^f = \begin{bmatrix} \mathbf{x}_f{}^T\mathbf{z}_\mathbf{u} \\ \mathbf{y}_f{}^T\mathbf{z}_\mathbf{u} \end{bmatrix} \tag{21}$$

$$\mathbf{R}_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ -sin\theta & -cos\theta \end{bmatrix} \tag{22}$$

where θ represents the angle between parametric axes $\mathbf{g}_s$ and $\mathbf{f}_u$, β is the distance between contacts, $\mathbf{x}_f = \mathbf{f}_u/||\mathbf{f}_u||$, $\mathbf{z}_f = \mathbf{f}_u\times\mathbf{f}_v/||\mathbf{f}_u\times\mathbf{f}_v||$, and $\mathbf{y}_f = \mathbf{z}_f\times\mathbf{x}_f$. $\mathbf{E}^g, \mathbf{F}^g$ are defined in a similar manner.

The authors have also developed the relation in terms of world frame quaternion velocity body coordinates in the Appendix, which may be more convenient for use in haptics.

3.7 Comparing the Methods

The advantage to using Newton's iterative method is that it converges to a solution given a close initial guess. We are required to use the Newton method during the non-penetrating case so that we obtain an exact starting point for the velocity method.

The advantage to the velocity space method is that it is an exact relation at that instant in time; it is not an iterative numerical method. The integration of $\dot{\mathbf{u}}$ provides a highly accurate, strictly continuous tracing update. It is very well conditioned and does not suffer from the optimization problems of Newton's method. However, it does not converge to the true minimal distance given only an approximate starting point. It is a good algorithm for generating virtual proxy information because it is a strictly local distance update; the curves $\gamma(t)$ and $\zeta(t)$ are continuous, where they may be discontinuous in small intervals with the Newton method.

The authors have found that haptics surface tracing should be as minimally confusing as possible. When given a choice of switching to a non-local point, which would cause discontinuous force feedback, it has been our experience that the more local choice is desirable. This choice is enforced by the continuous updates provided by the velocity method.

4 ALGORITHM & IMPLEMENTATION

In sum, we have the following algorithm for the non-penetrating and penetrating case:

- Far : Global distance refinement, obtain approximate **u**
- Near : Newton iteration, obtain exact **u**
- Very Near and Penetrating : Velocity formulation, using exact **u**

During tracking (the outside, non-penetration case), we use a global "monitoring" mechanism. Newton iteration and the velocity method are run concurrently during this case. kilohertz rate updates are not critical since the haptics control is returning no force during non-penetration.

Once the Newton or velocity methods detect a penetration, the monitoring and Newton tracking are turned off. It is assumed that non-local jumps are not possible because a haptics device can hold a user to within .3 mm of the surface. Global jumps are also not desirable because of the need for virtual proxies for the finger model (see Fig.2).

Due to the use of extremal points in tracking and tracing parametric surfaces, the existing NURBS trimming implementation [Thompson, 1999] may still be used. This model is an approximation for effects such as falling off or transitioning between edges, because the "middle" of the finger is considered to be completely off as soon as the extremal point is off. A full model for tracing edges is considerably more costly in terms of computation and is a subject of future work. However, the important perception of falling off edges that is remarkably well rendered with haptic devices is not significantly diminished with this approximate model.

5 RESULTS

Our approach is efficient because two surface evaluations for for the contact points and surface partial derivatives at the points are required for the velocity and Newton formulations. Timing results [Johnson, 1998b] have shown that the point and partial evaluations are only slightly slower than evaluating only the point. Running times on an SGI R10000 Onyx 2 for two surface evaluations inside Alpha_1 are about .07 milliseconds. Other operations including the inverse of the 4×4 matrix and other restacking required in the velocity and Newton formulation are not insignificant, but run in .01 milliseconds (that is, the cost of the methods excluding the surface evaluations). Thus, a single processor system can easily perform control and surface-surface analysis at several kilohertz update rates.

Figure 4 shows the concurrent global monitoring and local Newton and velocity parametric tracing to enable surface-surface contact analysis at kilohertz rates for haptics control.

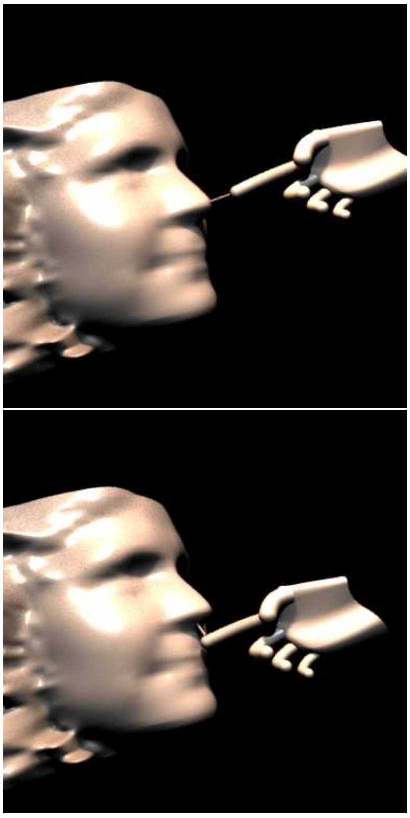

Figure 4: Combined global monitoring and local parametric tracing enable surface-surface contact analysis at kilohertz rates for haptics control.

We have avoided the introduction of unstable artifacts from a purely iterative numerical method at the time of impact that would be felt by the user. The generally accepted noticeable level of vibration are on the order of less than a micron at various frequencies from 1 Hz to 1kHz [NRC, 1995]. The artifacts caused by the numerical methods may be many orders of magnitude greater than the just noticeable error that a user may perceive.

Numerical integration of the results from the velocity formulation may be done with the simple Euler's method for short intervals of time in a haptics environment due to the very small step sizes between servo loop cycles. For other applications that use larger timesteps, as occurs in our simulation debugging code, we have used standard fourth order numerical integration techniques. A very long tracing sequence, on the order of 10^8 seconds for typical user motions, can be performed in practice with this integration technique without accumulating noticeable errors. Periodic "restarts" due to user transition to the non-penetrating condition and subsequent tracking by Newton's method occur quite often. Even higher order integration methods can be employed if some unusual circumstance or application requires it without excessive cost due to the efficiency of our tracking techniques.

The reliability of these distance methods depends partially on the underlying stability of the numerical methods. Singular regions in the case of point-to-surface computations were derived in [Johnson, 1998b]; we expect similar conditions for the surface-to-surface extremal distance computations. During nearly singular concave cases, as in Fig.5, the condition numbers during the first iteration of Newton's method is roughly 3 or 4, where it has usually been between 1 and 2 in other configurations. The velocity method maintains a condition number of roughly 1 or 1.2 for the

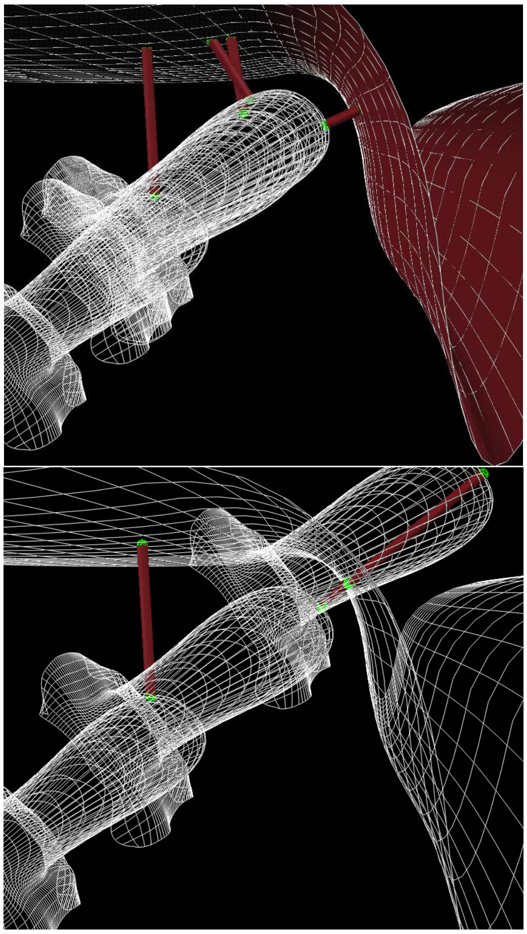

Figure 5: The reformulated Newton local distance method (top) may have problems with concave cases that the velocity (bottom) does not have.

cases tried so far, for nearly singular and other cases. Because condition numbers less than 100 are considered to be small, the loss of precision upon matrix inverse operations is shown to be small with both methods. Upon absolute singular configurations, both methods have singular matrices. However, Newton's method has additional convergence requirements [McCalla, 1967] that can produce additional instability in concave regions, even when the condition number is very small. The velocity method does not exhibit these instabilities. We are investigating more sophisticated numerical methods to improve the reliability of the Newton method approach.

6 CONCLUSION

The tracing update formulations presented here provide improved surface tracing interactions. A fast, compact method for surface-surface updates for the nearly penetrating and penetrating case have been developed and analyzed. The velocity formulation has been introduced for use in haptics penetration.

ACKNOWLEDGMENT

Thanks go to the students and staff of the *Alpha_1* project, within which this work was developed. Support for this research was provided by NSF Grant MIP-9420352, by DARPA grant F33615-96-C-5621, and by the NSF and DARPA Science and Technology Center for Computer Graphics and Scientific Visualization (ASC-89-20219).

7 APPENDIX: Surface Contact Velocity Formulation

A number of different derivations of the kinematics of contact have been developed in the last 15 years. These formulations relate the rate of change of the parametric contact coordinates to the Cartesian velocity and angular velocity of the bodies in contact. Previous works have been limited to surfaces parameterized have orthogonal surface partial derivatives. Some are also limited the in-contact case[Montana, 1986, Montana, 1988, Murray, 1990, Cai, 1987]. While any surface may be reparameterized to be orthogonal, it may be expensive and impractical to find such a parameterization. The finger may bend in our interacitve application. Finding the reparameterization with full numerical precision is also a problem[Maekawa, 1996]. We develop a new derivation for the not-in-contact, non-orthogonal surface parameterizations.

When both surfaces have partials that are everywhere orthogonal, i.e. $\mathbf{f}_u \cdot \mathbf{f}_v = 0$ and $\mathbf{g}_s \cdot \mathbf{g}_t = 0, \forall \mathbf{u}$ in the parameteric domain, an original result from [Montana, 1986], extended by [Cremer, 1996] for the not in contact case, derives the surface kinematics as

$$\dot{\mathbf{u}}_f = \mathbf{I}_f{}^{-1}\mathbf{R}_\theta(\mathbf{II}_g + \widetilde{\mathbf{II}}_f + \beta\mathbf{II}_g\widetilde{\mathbf{II}}_f)^{-1}(\begin{bmatrix} -w_y \\ w_x \end{bmatrix} + \mathbf{II}_g \begin{bmatrix} v_x \\ v_y \end{bmatrix})$$

$$\dot{\mathbf{u}}_g = \mathbf{I}_g{}^{-1}(\mathbf{II}_g + \widetilde{\mathbf{II}}_f + \beta\widetilde{\mathbf{II}}_f\mathbf{II}_g)^{-1}((\mathbf{1} + \beta\widetilde{\mathbf{II}}_f)\begin{bmatrix} -w_y \\ w_x \end{bmatrix} - \widetilde{\mathbf{II}}_f \begin{bmatrix} v_x \\ v_y \end{bmatrix})$$

where β is the (signed) distance between contacts, $\mathbf{1}$ is the 2×2 identity matrix, the relative linear and angular surface velocities of surface $\mathbf{g}$ relative to surface $\mathbf{f}$ be denoted by $\mathbf{v}, \omega$, the surface contact velocities be $\mathbf{v}^f, \omega^f$ and $\mathbf{v}^g, \omega^g$, $\mathbf{I}$ is the first fundamental form and $\mathbf{II}$ is the surface curvature or second fundamental form, with subscripts for surfaces $\mathbf{f}$ and $\mathbf{g}$. θ represents the angle between parametric axes $\mathbf{g}_s$ and $\mathbf{f}_u$, let $\mathbf{R}_\theta = \begin{bmatrix} cos\theta & -sin\theta \\ -sin\theta & -cos\theta \end{bmatrix}$ and $\widetilde{\mathbf{II}} = \mathbf{R}_\theta\mathbf{II}\mathbf{R}_\theta$. The relation may be written in the following matrix form,

$$\dot{\mathbf{u}} = \mathbf{A}\begin{bmatrix} \mathbf{v}_{x,y} \\ \omega_{x,y} \end{bmatrix}. \quad (23)$$

7.1 Non-Orthogonal Parameterizations

We provide a new derivation for $\mathbf{A}$ for regular parametric surfaces whose partials are not orthogonal so the result in Eqn. 23 can be used for typical models constructed by CAD systems.

The parametric contact frames in Fig. 6 for the extremal distance context are defined with an orthonormal set of vectors. $\mathbf{R}_f = [\mathbf{x}_f \ \ \mathbf{y}_f \ \ \mathbf{z}_f]$ is the rotation matrix from the local contact frame to the world frame, where $\mathbf{x}_f = \mathbf{f}_u/||\mathbf{f}_u||$, $\mathbf{z}_f = \mathbf{f}_u \times \mathbf{f}_v/||\mathbf{f}_u \times \mathbf{f}_v||$, $\mathbf{y}_f = \mathbf{z}_f \times \mathbf{x}_f$. $\mathbf{R}_g$ is similarly defined. $\mathbf{z}_f$ and $\mathbf{z}_g$ are parallel free vectors (Fig. 6).

Comparison of the relative surface velocities[1] can be used to re-

[1] A subscript with x or y such as $\mathbf{a}_x$ will denote the first component of the vector. Several subscripts, such as $\mathbf{a}_{-x,y}$ will represent a two-vector containing the negative of the first component and the second component of $\mathbf{a}$. A superscript T as in $\mathbf{a}^T$ will denote a vector or matrix transpose. Partitions of matrices may be selected with (row,column) indexing, with "a:b" for a range or ":" for all rows or columns, as in the $Matlab^{TM}$ notation. The operator $extract\_skew\_symmetric$ retrieves 3 independent components from 9 elements of a skew symmetric matrix.

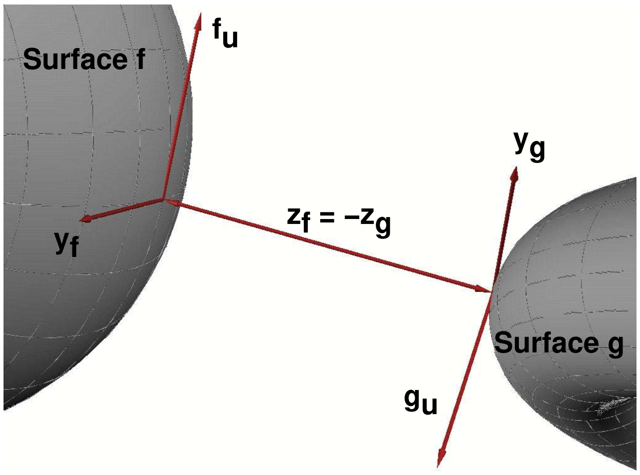

Figure 6: Closest point surface contact frames. Velocity relations allow the contact coordinate velocities to be found.

late the parametric contact coordinates $\mathbf{u}_f$ and $\mathbf{u}_g$ with the linear and angular surface velocities. Let the surface velocities $\mathbf{v}^f$ and $\mathbf{v}^g$ and relative surface velocity $\mathbf{v}$ be in the frame $\left[\begin{array}{ccc} \mathbf{x}_g & \mathbf{y}_g & \mathbf{z}_g \end{array}\right]$. In the extremal distance context (see Fig. 6), we have

$$\mathbf{v}^g_{x,y} + \mathbf{v}_{x,y} = \mathbf{R}_\theta \mathbf{v}^f_{x,y} + \beta \mathbf{R}_\theta \omega^f_{y,-x}, \tag{24}$$

$$\omega^g_{y,-x} + \omega_{y,-x} = -\mathbf{R}_\theta \omega^f_{y,-x}. \tag{25}$$

The terms $\mathbf{v}^f$ and $\mathbf{v}^g$ will contain $\dot{\mathbf{u}}_f$ and $\dot{\mathbf{u}}_g$ in the following equations through due to the chain rule of differentiation. We derive matrices $\mathbf{E}^f_{x,y}$ and $\mathbf{F}^f_{x,y}$ for surface f (and analogously for surface g) from the relations of linear and angular velocity to separate $\dot{\mathbf{u}}_f$ and $\dot{\mathbf{u}}_g$ from other terms,

$$\mathbf{v}^f_{x,y} = \mathbf{R}^T_{f x,y} \dot{\mathbf{f}}_{x,y} = \left[\left[\begin{array}{cc} \mathbf{x}_f & \mathbf{y}_f \end{array}\right]^T \mathbf{f_u} \right]_{2\times 2} \dot{\mathbf{u}}_f = \mathbf{E}^f_{x,y} \dot{\mathbf{u}}_f, \tag{26}$$

$$\omega^f_{y,-x} = extract\_skew\_symmetric(\mathbf{R}_f{}^T \dot{\mathbf{R}}_f)_{y,-x}$$

$$= \left[\begin{array}{c} \mathbf{x}_f{}^T \mathbf{z_u} \\ \mathbf{y}_f{}^T \mathbf{z_u} \end{array}\right]_{2\times 2} \dot{\mathbf{u}}_f = \mathbf{F}^f_{y,-x} \dot{\mathbf{u}}_f. \tag{27}$$

Using Eqns. 24,25,26,27, the general non-orthogonal case is reduced to the 4×4 system,

$$\left[\begin{array}{cc} \mathbf{R}_\theta(\mathbf{E}^f_{x,y} - \beta \mathbf{F}^f_{-y,x}) & -\mathbf{E}^g_{x,y} \\ (-\mathbf{R}_\theta \mathbf{F}^f_{y,-x})_{x,y} & -\mathbf{F}^g_{x,y} \end{array}\right] \left[\begin{array}{c} \dot{\mathbf{u}}_f \\ \dot{\mathbf{u}}_g \end{array}\right] = \left[\begin{array}{c} \mathbf{v}_{x,y} \\ \omega_{x,y} \end{array}\right]. \tag{28}$$

This system can be solved quickly for $\dot{\mathbf{u}}$ using the following pseudocode fragment for arbitrary surface parameterizations. We rewrite the inverse of the 4×4 coefficient matrix in Eq. 28,

$$\mathbf{A} = \left[\begin{array}{cc} \mathbf{R}_\theta(\mathbf{E}^f_{x,y} - \beta \mathbf{F}^f_{-y,x}) & -\mathbf{E}^g_{x,y} \\ -(\mathbf{R}_\theta \mathbf{F}^f_{y,-x})_{x,y} & -\mathbf{F}^g_{x,y} \end{array}\right]^{-1} \tag{29}$$

to be solved even more efficiently as a series of 2×2 matrix inverses and multiplications,

```
iEg_xy=inv(Eg_xy);
RphiFf_xy=Rphi*Ff_xy;
dRphiFf_xy=d*RphiFf_xy;
Fg_xyiEg_xy=Fg_xy*iEg_xy;
RphiEf_xy=Rphi*Ef_xy;

H=Fg_xyiEg_xy*(RphiEf_xy+dRphiFf_xy)-
RphiFf_xy;
iH=inv(H);
J=iH*[Fg_xyiEg_xy -eye(2)];

A=[J; iEg_xy*(RphiEf_xy*J+
          dRphiFf_xy*J-[eye(2) zeros(2)])];
```

Proof: From Eqs.24,26, we may write

$$\dot{\mathbf{u}}_g = \mathbf{E}^g_{x,y}{}^{-1} [\mathbf{R}_\theta \mathbf{E}^f_{x,y} \dot{\mathbf{u}}_f + \beta \mathbf{F}^f_{-y,x} \dot{\mathbf{u}}_f - \mathbf{v}]. \tag{30}$$

Substituting into Eq.25, we have

$$\mathbf{F}^g \mathbf{E}^g_{x,y}{}^{-1} [\mathbf{R}_\theta \mathbf{E}^f_{x,y} \dot{\mathbf{u}}_f + \beta \mathbf{F}^f_{-y,x} \dot{\mathbf{u}}_f - \mathbf{v}] + \omega = -\mathbf{R}_\theta \mathbf{F}^f \dot{\mathbf{u}}_f. \tag{31}$$

Gathering $\dot{\mathbf{u}}_f$,

$$\mathbf{F}^g \mathbf{E}^g_{x,y}{}^{-1} [\mathbf{R}_\theta \mathbf{E}^f_{x,y} + \beta \mathbf{F}^f_{-y,x} + \mathbf{R}_\theta \mathbf{F}^f] \dot{\mathbf{u}}_f = \mathbf{F}^g \mathbf{E}^g_{x,y}{}^{-1} \mathbf{v} - \omega. \tag{32}$$

Expressing this equation as a linear system, we have

$$\mathbf{H} \dot{\mathbf{u}}_f = \mathbf{F}^g \mathbf{E}^g_{x,y}{}^{-1} \mathbf{v} - \omega, \tag{33}$$

for which we can solve for the contact coordinates for surface $\mathbf{f}$,

$$\dot{\mathbf{u}}_f = \mathbf{H}^{-1} [\mathbf{F}^g \mathbf{E}^g_{x,y}{}^{-1} \quad -\mathbf{1}_{2x2}] \left[\begin{array}{c} \mathbf{v} \\ \omega \end{array}\right], \tag{34}$$

For convenience, let us represent $\mathbf{J}_{2x4} = \mathbf{H}^{-1} \left[\begin{array}{c} \mathbf{F}^g \mathbf{E}^g_{x,y}{}^{-1} - \mathbf{1}_{2x2} \end{array}\right]$ so that

$$\dot{\mathbf{u}}_f = \mathbf{J} \left[\begin{array}{c} \mathbf{v} \\ \omega \end{array}\right], \tag{35}$$

Now substituting this solution back into Eq.30,

$$\dot{\mathbf{u}}_g = \mathbf{E}^g_{x,y}{}^{-1} [\mathbf{R}_\theta \mathbf{E}^f_{x,y} \mathbf{J} \left[\begin{array}{c} \mathbf{v} \\ \omega \end{array}\right] + \beta \mathbf{F}^f_{-y,x} \mathbf{J} \left[\begin{array}{c} \mathbf{v} \\ \omega \end{array}\right] - \mathbf{v}]. \tag{36}$$

$$= \mathbf{E}^g_{x,y}{}^{-1} [\mathbf{R}_\theta \mathbf{E}^f_{x,y} \mathbf{J} + \beta \mathbf{F}^f_{-y,x} \mathbf{J} - [\mathbf{1}_{2x2} \;\; \mathbf{0}_{2x2}]] \left[\begin{array}{c} \mathbf{v} \\ \omega \end{array}\right] \tag{37}$$

Finally, letting $\mathbf{J}_b = \mathbf{E}^g_{x,y}{}^{-1} [\mathbf{R}_\theta \mathbf{E}^f_{x,y} \mathbf{J} + \beta \mathbf{F}^f_{-y,x} \mathbf{J} - [\mathbf{1}_{2x2} \;\; \mathbf{0}_{2x2}]]$, we may write

$$\dot{\mathbf{u}} = \left[\begin{array}{c} \mathbf{J} \\ \mathbf{J}_b \end{array}\right] \left[\begin{array}{c} \mathbf{v} \\ \omega \end{array}\right]. \tag{38}$$

The matrix $\mathbf{A}$ from Eq.29 is $\left[\begin{array}{c} \mathbf{J} \\ \mathbf{J}_b \end{array}\right]$, completing the proof. The solution is roughly as efficient as previous methods since the inverse of two 2×2 matrices and nine matrix multiplications, rather than four 2×2 matrix inversions and six matrix multiplications, is required (see pseudocode fragment). The optimized matrix inverse and multiplication implementation is a constant cost and is a small fraction of the cost associated with a surface evaluation.

7.2 Cartesian and Quaternion Generalized Velocities

Now we express the relative surface velocities in terms of world frame body coordinate velocities so that integration of orientation is possible (integration of angular velocity is meaningless). Define the local contact frame through the rotation matrix

$$\mathbf{R}_{loc} = \left[\begin{array}{ccc} \mathbf{x}_g & \mathbf{y}_g & \mathbf{z}_g \end{array} \right]^T. \tag{39}$$

Let $\mathbf{G}_f = \mathbf{G}(\mathbf{q}_{f,rot})$, $\mathbf{G}_g = \mathbf{G}(\mathbf{q}_{g,rot})$, where $\mathbf{G}(\mathbf{q}_{rot})$ is the matrix operator mapping quaternion velocities to angular velocities [Haug, 1992, Shabana, 1998], given by

$$\mathbf{G}(\mathbf{q}_{rot}) = 2 \left[\begin{array}{cccc} -q_{rot2} & q_{rot1} & q_{rot4} & -q_{rot3} \\ -q_{rot3} & -q_{rot4} & q_{rot1} & q_{rot2} \\ -q_{rot4} & q_{rot3} & -q_{rot2} & q_{rot1} \end{array} \right]. \tag{40}$$

The velocity $[\mathbf{v}^T \omega^T]^T$ is the motion of surface g relative to surface f. We write our world space velocities in terms of the local frame. We relate relative angular velocity and world frame quaternion velocity by

$$\omega = \mathbf{R}_{loc} \left[\begin{array}{cc} -\mathbf{T}_f\mathbf{G}_f & \mathbf{T}_g\mathbf{G}_g \end{array} \right] \left[\begin{array}{c} \dot{\mathbf{q}}_{f,rot} \\ \dot{\mathbf{q}}_{g,rot} \end{array} \right], \tag{41}$$

and relative linear velocity to Cartesian velocity through

$$\mathbf{v} = \mathbf{R}_{loc}(\dot{\mathbf{q}}_{g,tr} - \dot{\mathbf{q}}_{f,tr} + (\omega_g^{gl} - \omega_f^{gl}) \times (\mathbf{g}(\mathbf{u}) - \mathbf{q}_{g,tr})). \tag{42}$$

$\times$ denotes the vector cross product. $\mathbf{T}_f$ and $\mathbf{T}_g$ are rotations from the local frame to the world frame defined by quaternions $\mathbf{q}_{f,rot}$ and $\mathbf{q}_{g,rot}$. We write the relative surface velocities $[\mathbf{v}^T, \omega^T]^T$ as

$$\left[\begin{array}{c} \mathbf{v} \\ \omega \end{array} \right] = \left[\begin{array}{cc} \mathbf{R}_{loc} & -\mathbf{R}_{loc}(\mathbf{g}(\mathbf{u}) - \mathbf{q}_{g,tr})\times \\ \mathbf{0} & \mathbf{R}_{loc} \end{array} \right] *$$

$$\left[\begin{array}{cccc} -\mathbf{I}_{3x3} & \mathbf{0}_{3x4} & \mathbf{I}_{3x3} & \mathbf{0}_{3x4} \\ \mathbf{0}_{3x3} & -2\mathbf{T}_f\mathbf{G}_f & \mathbf{0}_{3x3} & 2\mathbf{T}_g\mathbf{G}_g \end{array} \right] \left[\begin{array}{c} \dot{\mathbf{q}}_f \\ \dot{\mathbf{q}}_g \end{array} \right] \tag{43}$$

where $\times$ in Eq. 43 denotes the $3x3$ skew symmetric matrix that performs the operation of a cross product (obtained from the three components of a vector, i.e. $\mathbf{a}\times = \left[\begin{array}{ccc} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{array} \right]$). From Eq. 38, the truncated part $[\mathbf{v}_{x,y}^T, \omega_{x,y}^T]^T$ is all that is required. Let $\mathbf{B}$ contain the first two rows and rows four and five of Eq. 43. Substituting $[\mathbf{v}_{x,y}^T, \omega_{x,y}^T]^T$ into Eq. 38, yields

$$\dot{\mathbf{u}} = \mathbf{AB} \left[\begin{array}{c} \dot{\mathbf{q}}_f \\ \dot{\mathbf{q}}_g \end{array} \right]. \tag{44}$$

7.3 Non-Orthogonal Surface-Curve Velocity Formulation

Similarly, it may be shown that the surface-curve extremal distance equations may be extended to for arbitrary surface parameterizations.

The time derivative of the parametric contact coordinates for surface f and curve g may be written as a function of linear and angular velocity multiplying a matrix operator,

$$\dot{\mathbf{u}}_{3x1} = \left[\begin{array}{cc} \mathbf{R}_\theta(\mathbf{E}_\mathbf{u}^f + d\mathbf{F}_{x,y}^f) & -\mathbf{g}_{u_{x,y}} \\ -\mathbf{R}_\theta\mathbf{F}_{x,y}^f & -k_g sin(\phi)\mathbf{g}_{u_{x,y}} \end{array} \right]_{3x3}^{-1} \left[\begin{array}{c} \mathbf{v}_{x,y} \\ \omega_y \end{array} \right]_{3x1}, \tag{45}$$

where

$$k_g = \frac{||\mathbf{g}_u \times \mathbf{g}_{uu}||}{||\mathbf{g}_u||^3} \tag{46}$$

and ϕ is the angle between the curve normal (which is $-\mathbf{z}^f$ for the extremal distance case) and the curve binormal $\mathbf{b}$, about the curve x axis $\mathbf{g}_u$. The binormal is

$$\mathbf{b} = \frac{\mathbf{g}_u \times \mathbf{g}_{uu}}{||\mathbf{g}_u \times \mathbf{g}_{uu}||}. \tag{47}$$

Eq. 43 is again used to establish this relation in terms of quaternion and Cartesian velocities, using rows 1,2, and 5.

References

[Bajaj, 1988] Bajaj, C., Hoffmann, C., Lynch, R., Hopcroft, J., "Tracing surface intersections," in *Computer Aided Geometric Design*, Vol 5, 1988, pp. 285-307.

[Baraff, 1990] Baraff, D., "Curved surfaces and coherence for non-penetrating rigid body simulation", in *Computer Graphics Proceedings, SIGGRAPH*, 24(4): 19-28, 1990.

[Cai, 1987] Cai, C., Roth, B., "On the spatial motion of rigid bodies with point contact," in *International Conference on Robotics and Automation,* pp. 686-695, March 1987.

[Cameron, 1997] Cameron, S., "Enhancing GFK: Computing Minimum and Penetration Distances between Convex Polyhedra," in *International Conference on Robotics and Automation,* April 1997.

[Cremer, 1996] Anitescu, M., Cremer, J., and Potra, F., "Formulating 3D Contact Dynamics Problems," *Mech. Struct. & Mach.*, vol. 24, no. 4, pp. 405-437, Nov. 1996.

[Cohen, 1980] Cohen, E., Lyche, T., and Riesenfeld, R., "Discrete B-Splines And Subdivision Techniques In Computer Aided Geometric Design And Computer Graphics," *Computer Graphics and Image Processing,* Vol 14, Number 2, October 1980.

[De, 1998] De, S., Srinivasan, M., "Rapid Rendering of "Tool-Tissue" Interactions in Surgical Simulations: Thin Walled Membrane Models", PHANToM User's Group Proceedings, PUG 1998, http://www.sensable.com/ community/ PUG98_papers.htm.

[Gregory, 1998] Gregory, A., Lin, M., Gottschalk, S., Taylor, R., "H-Collide: A Framework for Fast and Accurate Collision Detection for Haptic Interaction," UNC Report, currently unpublished, 1998.

[Gilbert, 1988] Gilbert, E., Johnson, D., Keerthi, S. "A Fast Procedure for Computing the Distance between Complex Objects in Three-Dimensional Space," IEEE Journal of Robotics and Automation, pp. 193-203, April 1988.

[Haug, 1992] Haug, E., Intermediate Dynamics, Prentice Hall, 1992.

[Johnson, 1998a] Johnson, D. E., and Cohen, E., " A framework for efficient minimum distance computations," Proc. IEEE Intl. Conf. Robotics & Automation, Leuven, Belgium, May 16-21, 1998, pp. 3678-3684.

[Johnson, 1998b] Johnson, D.E., and Cohen, E., “An improved method for haptic tracing of sculptured surfaces,” Symp. on Haptic Interfaces, ASME International Mechanical Engineering Congress and Exposition, Anaheim, CA, Nov. 15-20, 1998, in press.

[Kriezis, 1992] Kriezis, G., Patrikalakis, N., Wolder, F., “Topological and differential-equation methods for surface intersections,” in *Computer-Aided Design,* vol. 24, no. 1, January 1992, pp. 41-51.

[Lin, 1994] M.C. Lin, D. Manocha, and J. Canny. Fast contact determination in dynamic environments. in *IEEE Conference on Robotics and Automation,* pp. 602-609, 1994.

[Maekawa, 1996] Maekawa, T., Wolter, F., Patrikalakis, N., “Umbilics and lines of curvature for shape interrogation,” in *Computer Aided Geometric Design,* vol. 13, no. 2, March 1996.

[McCalla, 1967] McCalla, T., Introduction to Numerical Methods and FORTRAN Programming, John Wiley & Sons, Inc., 1967.

[Montana, 1986] Montana, D., J., 1986, Tactile sensing and the kinematics of contact, Ph.D. Thesis, Division of Applied Sciences, Harvard University.

[Montana, 1988] Montana, D., “The Kinematics of Contact and Grasp”, in *Int. J. of Rob. Res.,* vol. 7, no. 3, June 1988.

[Murray, 1990] Murray, R., Robotic Control and Nonholonomic Motion Planning, PhD. Dissertation, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 1990.

[NRC, 1995] National Research Council, Virtual Reality: Scientific and Technological Challenges, Committee on Virtual Reality Research and Development, National Academy Pres, 1995, p.69.

[Quinlan, 1994] Quinlan, S. “Efficient Distance Computation between Non-Convex Objects,” IEEE Int. Conference on Robotics and Automation, pp. 3324-3329, 1994.

[Ruspini, 1997] Ruspini, D., Kolarov, K., Khatib, O., “The Haptic Display of Complex Graphical Environments,” in *Computer Graphics Proceedings, SIGGRAPH,* August, 1997.

[Ruspini, 1998] Ruspini, D., Khatib, O., “Dynamic Models for Haptic Rendering Systems.” in *Advances in Robot Kinematics: ARK’98*, June 1998, Strobl/Salzburg, Austria, pp 523-532.

[Sato, 1996] Sato, Y., Hirata M., Maruyama T., Arita Y., “Efficient Collision Detection using Fast Distance Calculation: Algorithms for Convex and Non-Convex Objects,” in *Proceedings of the 1996 IEEE International Conference on Robotics and Automation,* April, 1996, pp. 771-778.

[Shabana, 1998] Shabana, A., Dynamics of Multibody Systems, Cambridge University Press, 1998.

[Stewart, 1997] Stewart P., Chen Y., Buttolo P., “CAD Data Representations for Haptics Virtual Prototyping,” in *Proceedings of DETC ’97*, 1997 ASME Design Engineering Technical Conferences, Sept. 14-17, 1997, Sacramento, CA.

[Snyder, 1995] Snyder, J., “An Interactive Tool for Placing Curved Surfaces without Interpenetration,” in *Computer Graphics Proceedings, SIGGRAPH,* Los Angeles, August 6-11, 1995, pp. 209-216.

[Thompson, 1997] Thompson II, T.V., Nelson, D.D., Cohen, E., and Hollerbach, J.M., “Maneuverable NURBS models within a haptic virtual environment,” *6th Annual Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems,* DSC-Vol. 61, (Dallas, TX), pp. 37-44, Nov. 15-21, 1997.

[Thompson, 1999] Thompson II, T.V., Cohen, E., “DIRECT HAPTIC RENDERING OF COMPLEX TRIMMED NURBS MODELS,” in *8th Annual Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems,* (Nashville, TN), 1999.

[Zilles, 1995] Zilles, C.B., and Salisbury, J.K., “A Constraint-based God-object Method For Haptic Display,” in *Proc. IEE/RSJ International Conference on Intelligent Robots and Systems, Human Robot Interaction, and Cooperative Robots,* Vol 3, pp. 146-151, 1995.