AN IMPROVED METHOD FOR HAPTIC TRACING OF A SCULPTURED SURFACE

By David E. Johnson and Elaine Cohen
Department of Computer Science, University of Utah
Email: dejohnso@cs.utah.edu

1.0 ABSTRACT

We present an improved method for directly tracing a sculptured surface. This method extends previous work by adding second-order surface information to a closest point tracking algorithm which greatly increases the stability and also improves the accuracy of the algorithm. As part of the derivation of the new method, we examine in detail the system of equations that tracks the closest point on the surface and highlight surface features that can cause problems. We then address these potential problems.

2.0 INTRODUCTION

NURBS surfaces are an industry standard for CAD modeling. Thus, a haptic system that directly uses a NURBS surface representation avoids a cumbersome and potentially inaccurate conversion step between the modeler and the haptic renderer. In previous work, Thompson et al [12] demonstrated direct parametric tracing (DPT) for performing haptic rendering on NURBS surfaces.

Our goal in this work is to improve the original DPT method while maintaining the advantage of directly tracing a surface. We have found that inclusion of higher-order surface information greatly increases the stability of the haptic rendering and more accurately renders the surface, while not significantly slowing the method.

3.0 BACKGROUND

In haptic rendering, a person feels a sense of contact with a computer model of an object. Often, a mechanical device attached to the person's hand or arm provides the forces needed to generate this sense of contact with the virtual model.

A wall model describes mathematically how the computer model's surface reacts locally to contact. A simple wall model reacts like a spring --- the force pushing out, the "restoring force," increases linearly with the depth of penetration into the surface. The closest local point on the surface and its associated surface normal determine the magnitude and direction of the restoring force.

A basic operation in haptics is tracing along the surface of a model. So as the person's finger (which is often treated as a single point end effector) moves along the surface of the model, the penetration depth into the model and the closest local point must be computed at numerous locations.

This closest local point must be computed rapidly, at least at several hundred Hz [5], or the surface will either feel soft or be unstable. This demand for high computation rates has led to using different surface representations in the haptic renderer and to using different approaches for finding the closest local point.

Adachi [1] and Mark [5] have used a tangent plane approximation to the surface to maintain the update rate. Adachi refers to this as an intermediate representation. With an intermediate representation, the penetration depth calculation depends on the tangent plane for several time steps before a new tangent plane is computed. Intermediate representation methods maintain high update rates since the closest point on a plane computes quickly. Problems may occur in regions of high curvature; the plane

poorly approximates the surface there. Also, the haptic device can apply discontinuous forces when the tangent plane changes [5].

We can find the closest point quickly on small polygonal models. Salisbury [11] used small polygonal models in his haptic renderer and interpolated between vertex normals to approximate a smooth surface. Ruspini [10] applied efficient means of finding the minimum distance to large polygonal models [9] to allow tracing of more complex models. However, haptic rendering of polygonal models still requires a conversion process from most CAD modelers.

Luecke [4] explored NURBS sculptured surfaces haptically by constraining the user's hand to the tangent plane at a moving surface point. This changes the problem from finding the penetration depth to satisfying a constraint. The constraints were satisfied by performing surface tangent vector inversion [7] on the movement of the user's hand. While useful, the constraint method doesn't support free-form exploration of the surface.

Stewart [13] traced spline surfaces by applying globally convergent numerical methods to the system of equations describing an orthogonal projection onto a surface $\boldsymbol{S}(u, v)$, as in:

$$(S(u,v)-P)\cdot\frac{\partial S}{\partial u}=0 \quad \text{and} \quad (S(u,v)-P)\cdot\frac{\partial S}{\partial v}=0. \qquad \text{(EQ 1)}$$

Sometimes the numerical methods directly tracked the closest point on a surface, but more often an intermediate tangent plane representation maintained haptic rates.

The above methods either approximate a sculptured surface with another representation, or fall back on simpler methods to maintain haptic rates when working directly from the surface. We would like to take advantage of the exact surface normals and curvature information available from NURBS surfaces in our haptic rendering method.

In [12] we presented the direct parametric tracing (DPT) method for NURBS surfaces. The DPT method always uses points on the surface and exact surface normals to compute the restoring force and, thus, haptically renders sculptured surfaces. The work showed how first-order information for a surface could be quickly extracted at a point and derived a relationship between movement in space and movement in the parametric domain of the surface. For a NURBS curve, defined by its control polygon $\boldsymbol{P}$ and knot vector $\boldsymbol{t}$, the change of parameter Δt can be related to movement in space of the end effector $\Delta \boldsymbol{E}$ by

$$\Delta t = \left(\frac{\Delta \boldsymbol{E}\cdot(\boldsymbol{P}_{i^*+1}-\boldsymbol{P}_{i^*})}{\left\|\boldsymbol{P}_{i^*+1}-\boldsymbol{P}_{i^*}\right\|^2}\right)\left(\frac{w_{i^*}}{w_{i^*+\bar{1}}}\right)\left(\frac{\boldsymbol{t}_{i^*+k}-\boldsymbol{t}_{i^*+1}}{k-1}\right). \qquad \text{(EQ 2)}$$

The i^* denotes that the curve has been refined to be an evaluation point. Note that the change in parameter of the closest point can be found by just looking at points in the control polygon, the rational weights w of those points, and their associated knots in the knot vector.

4.0 Concerns about the DPT method

Our current haptic rendering system [3] has used the DPT method for tracing and manipulating various CAD models. Since it uses only first derivative surface information for approximating the movement along the surface we have two main concerns about the method: accuracy and stability.

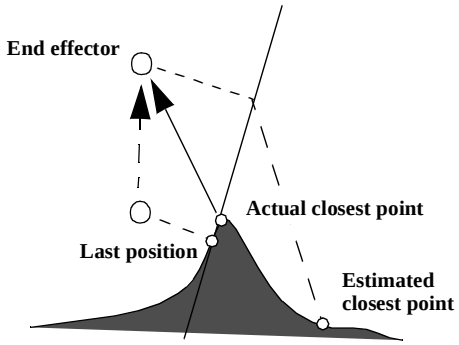

FIGURE 1. As the end effector moves away from the actual closest point, the estimated closest point overshoots.

Problems with accuracy are demonstrated in Figure 1, where the estimate of the new closest point will dramatically overshoot the correct closest point. The amount of overshoot is not bounded --- as the distance of the end effector from the actual closest point grows larger, the amount of overshoot can grow as well.

The stability concern arises from repeated iterations of the closest point tracking method. In regions of high curvature, multiple iteratons of the DPT method can produce an oscillation of the estimated closest point rather than convergence to a more accuracte solution.

Since we used the DPT method for very small penetration depths, it performed well during surface tracing. As we plan to use haptic rendering on more complex models in more complex environments, though, a more stable and accurate method becomes desirable.

5.0 AN IMPROVED DPT METHOD

The lack of curvature information led to problems in the DPT method; therefore, we add higher-order surface information to the closest point approximation. Higher-order surface information extracts from a NURBS surface in a similar manner to the first derivative information used in the DPT method.

5.1 Solving for the Closest Local Point

Eq. 1 describes an orthogonal projection of the point $\boldsymbol{P}$ onto the surface $\boldsymbol{S}(u, v)$. Given an initial point (u_i, v_i) on the surface we can find a $(\Delta u, \Delta v)$ to move towards the local closest point by using multidimensional Newton's method [8],

$$\boldsymbol{J} \cdot \Delta \boldsymbol{x} = -\boldsymbol{F}, \qquad \text{(EQ 3)}$$

where $\boldsymbol{F}$ is the system of equations described by Eq. 1, $\boldsymbol{J}$ is the Jacobian of $\boldsymbol{F}$, and $\Delta \boldsymbol{x}$ is the solved for change in uv needed to move towards the closest point to $\boldsymbol{P}$.

Expanding Eq. 3 for the case of solving for the closest point we find

$$\begin{bmatrix} \frac{\partial}{\partial u}[(\boldsymbol{S}-\boldsymbol{P})\cdot\boldsymbol{S}_u] & \frac{\partial}{\partial v}[(\boldsymbol{S}-\boldsymbol{P})\cdot\boldsymbol{S}_u] \\ \frac{\partial}{\partial u}[(\boldsymbol{S}-\boldsymbol{P})\cdot\boldsymbol{S}_v] & \frac{\partial}{\partial v}[(\boldsymbol{S}-\boldsymbol{P})\cdot\boldsymbol{S}_v] \end{bmatrix} \cdot \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = -\begin{bmatrix} (\boldsymbol{S}-\boldsymbol{P})\cdot\boldsymbol{S}_u \\ (\boldsymbol{S}-\boldsymbol{P})\cdot\boldsymbol{S}_v \end{bmatrix}. \qquad \text{(EQ 4)}$$

Note that the Jacobian contains the second partial derivatives of the surface. We solve for $\Delta \boldsymbol{uv}$ by performing LU decomposition on the system.

6.0 HIGHER-ORDER SURFACE INFORMATION

As Eq. 4 shows, we need to evaluate the second partial derivatives of the surface to solve for $\Delta \boldsymbol{uv}$. This second-order information makes the method more sensitive to the curvature of the surface.

Exact derivatives of a NURBS surface compute as a by-product of finding an evaluation point on the surface. For NURBS, differencing methods for approximating derivatives become more expensive than exact methods since differencing methods require surface evaluation of multiple points. Thus, with NURBS, we obtain more accurate derivatives at a smaller computation cost than for some other surface representations.

The second partials of a surface at an evaluation point depend on the control polygon and knot vector of the surface in a manner similar to that in Eq. 2 for the first partials [12]. See [7] for more information on deriving the second partial derivative equations. In order to gain some intuition, however, we show the result for $\boldsymbol{S}_{uu}$ for a B-spline of order m in the u direction.

$$\boldsymbol{S}_{uu} = \frac{\frac{(m-1)(\boldsymbol{P}_{j^*,i^*+2} - \boldsymbol{P}_{j^*,i^*+1})}{\boldsymbol{t}_{j^*,(i^*+k+1)} - \boldsymbol{t}_{j^*,(i^*+2)}} - \frac{(m-1)(\boldsymbol{P}_{j^*,i^*+1} - \boldsymbol{P}_{j^*,i^*})}{\boldsymbol{t}_{j^*,(i^*+k)} - \boldsymbol{t}_{j^*,(i^*+1)}}}{\frac{\boldsymbol{t}_{j^*,(i^*+k)} - \boldsymbol{t}_{j^*,(i^*+2)}}{(m-2)}}. \qquad \text{(EQ 5)}$$

The second partials at an evaluation point depend on differences of the control mesh, the knot vector, and the order of the surface. The NURBS case looks similar in structure, with a few extra terms due to the presence of rationals.

7.0 HIGH RATE VS. HIGH ACCURACY

We adopt the following approach: that performing one iteration of the Newton solver at a high sampling rate tracks the closest point about as well as performing multiple iterations of the Newton solver at a slower rate. We tested this assumption by simulating the haptic tracing of a surface using a preset path for the end effector. The simulator measured the distance from the end effector to the tracked closest point at 100 samples along the path. We either performed multiple iterations of Eq. 4 at each sample point or performed one iteration of Eq. 4 and took a number of substeps between sample points for a larger number of total steps to test the assumption.

TABLE 1. More iterations vs. more steps

#steps	#Newton iters	Avg. distance
100	2	1.520713
200	1	1.521769
100	3	1.520705
300	1	1.520998

As Table 1 shows, as the number of steps increases, the average penetration depth converges to the penetration depth found using more iterations of the solver. Thus, choosing more frequent samples over more iterations of the solver tracks the closest point well and has the advantage of running at the high rates needed for haptic rendering.

7.1 Singularities in the Jacobian

Early testing of the tracing method using the Newton solver showed that it did a good job of tracking the closest point. Unfor-

tunately, sometimes the solver produced a large $\Delta \boldsymbol{uv}$ to a point that was clearly not the closest point on the surface.

Inspection of the system of equations shows that it can tend towards singularity. To show this, we compute the partial derivatives in the Jacobian and expand Eq. 4 to

$$\begin{bmatrix} \overrightarrow{SP} \cdot \boldsymbol{S}_{uu} + \boldsymbol{S}_u^2 & \overrightarrow{SP} \cdot \boldsymbol{S}_{uv} + \boldsymbol{S}_u \cdot \boldsymbol{S}_v \\ \overrightarrow{SP} \cdot \boldsymbol{S}_{uv} + \boldsymbol{S}_u \cdot \boldsymbol{S}_v & \overrightarrow{SP} \cdot \boldsymbol{S}_{vv} + \boldsymbol{S}_v^2 \end{bmatrix} \cdot \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = - \begin{bmatrix} \overrightarrow{SP} \cdot \boldsymbol{S}_u \\ \overrightarrow{SP} \cdot \boldsymbol{S}_v \end{bmatrix} . \quad \text{(EQ 6)}$$

Notice that the Jacobian of this system of equations shares similarities with the matrices of the first and second fundamental forms [6] of a surface $\boldsymbol{G}$ and $\boldsymbol{L}$.

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{S}_u \cdot \boldsymbol{S}_u & \boldsymbol{S}_u \cdot \boldsymbol{S}_v \\ \boldsymbol{S}_u \cdot \boldsymbol{S}_v & \boldsymbol{S}_v \cdot \boldsymbol{S}_v \end{bmatrix} = \begin{bmatrix} E & F \\ F & G \end{bmatrix} . \quad \text{(EQ 7)}$$

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{S}_{uu} \cdot \boldsymbol{n} & \boldsymbol{S}_{uv} \cdot \boldsymbol{n} \\ \boldsymbol{S}_{uv} \cdot \boldsymbol{n} & \boldsymbol{S}_{vv} \cdot \boldsymbol{n} \end{bmatrix} = \begin{bmatrix} L & M \\ M & N \end{bmatrix}$$

If we decompose the vector from the surface to the end effector $\overrightarrow{SP}$ into components along the surface normal and in the surface tangent plane we have

$$\overrightarrow{SP} = \gamma \boldsymbol{n} + \beta \boldsymbol{e} , \quad \text{(EQ 8)}$$

where $\boldsymbol{e}$ is the unit projection of $\overrightarrow{SP}$ onto the tangent plane. The height above the surface in the normal direction is γ. Using Eq. 8, we can rewrite the Jacobian as

$$J = \begin{bmatrix} \gamma L + E + \beta(e \cdot \boldsymbol{S}_{uu}) & \gamma M + F + \beta(e \cdot \boldsymbol{S}_{uv}) \\ \gamma M + F + \beta(e \cdot \boldsymbol{S}_{uv}) & \gamma N + G + \beta(e \cdot \boldsymbol{S}_{vv}) \end{bmatrix} . \quad \text{(EQ 9)}$$

The Jacobian becomes singular when the determinant of J is zero. Ignoring the tangent plane component for the moment, the determinant is zero when

$$\gamma^2 \left(\frac{LN - M^2}{EG - F^2} \right) + \gamma \left(\frac{LG + EN - 2MF}{EG - F^2} \right) + 1 = 0 . \quad \text{(EQ 10)}$$

The principal curvatures of a surface κ_1 and κ_2 define the maximum and minimum curvatures at a point on the surface. Using the definitions [2] of principal curvature, the condition for singularity becomes

$$\gamma^2 \kappa_1 \kappa_2 + \gamma(\kappa_1 + \kappa_2) + 1 = 0 . \quad \text{(EQ 11)}$$

This quadratic equation has roots

$$\gamma = -\frac{1}{\kappa_1} \text{ and } \gamma = -\frac{1}{\kappa_2} . \quad \text{(EQ 12)}$$

Eq. 12 shows that the Jacobian can become singular when the end effector is at the center of radius of one of the principal curvatures of the point on the surface.

The tangent plane term that we conveniently ignored can also contribute to the singularity of the Jacobian. We are still exploring the meaning of this equation in terms of intrinsic properties of the surface. At the very least, the added term should distort the singularity away from the exact principal directions.

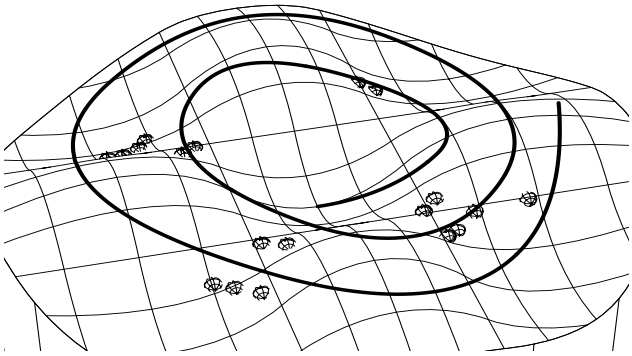

FIGURE 2. Tracked points that exhibit some Jacobian singularity problems are indicated with spheres.

We tested our analysis with a simulated tracing session as shown in Figure 2. The level dark spiral above the surface indicates the path of the end effector. The spheres show points where the Jacobian tended towards singularity. The spheres cluster in the concavities of the surface; in particular, the singularities occur in high curvature regions when the path drops close to the surface and in flatter regions when the path rises farther from the surface. These results bear out the predictions of the above analysis.

7.2 A Hybrid Method

To alleviate the poor tracking due to singularities, we have adopted the scheme of performing a tangent plane projection as done in the DPT method as well as an iteration of the multidimensional Newton's method at each time step. Since most of the computational cost is from evaluating the surface point and the associated partial derivatives, the extra tangent plane projection doesn't add much to the cost of the method. If each method computes a similar $\Delta \boldsymbol{uv}$, we choose the result from the Newton's method. If the two methods produce dissimilar results, we evalu-

ate the new locations and we choose the point closest to the end effector. We refer to this method as hybrid direct parametric tracing (HDPT).

We prefer this approach over the globally convergent methods used by [13] or other standard ways of dealing with singular Jacobians, such as Levenberg-Marquardt methods [8] because we wish to maintain high haptic rates. More complex methods often require multiple iterations which can drop the system below haptic rates.

8.0 COMPARING DPT AND HDPT

We have compared the original DPT method with HDPT method by running a simulated trace of an object using the two different methods. At each time step, the test program records the distance from the end effector to the tracked closest point on the surface. The sum of distances makes a simple comparison of how effectively the method is tracking the closest point on the surface --- the higher the summed distances, the worse the tracking. In additon, we store the highest positive and negative difference in distance between the two methods. This gives an indication of the worst-case behavior of the method.

8.1 Showing Improved Tracking

TABLE 2. Comparing DPT with the improved method

Path	Method	Steps	Rate (Hz)	Avg. Depth
Far	HDPT	500	1569	0.09159
Far	DPT	500	1860	0.09255
Near	HDPT	300	1258	0.01517
Near	DPT	300	1313	0.01520

To show improved tracking, we used two different end effector paths, one very near the surface and one farther away. Table 2 shows the results of the test. In each case, the hybrid method showed a smaller average penetration depth and thus a better tracked closest point. The improvement on the more distant path was more significant, which we expected because projection onto the tangent plane poorly approximates the closest local point at greater distances.

Perhaps more significantly than the improved average depth result, the HDPT method had better worst-case performance than the DPT method. The difference in penetration depth at the time the hydrid method improved the most over the DPT method was consistently at least an order of magnitude bigger than the difference in depth at the time the DPT performed the best over the hybrid method. This indicates better worst case behavior for the hybrid-based approach.

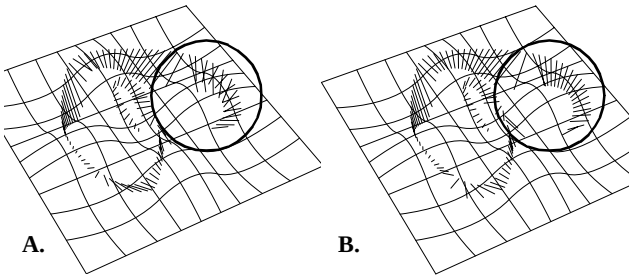

FIGURE 3. The far test path, samples, and tracked closest point on the surface are shown for the DPT method (A) and the improved HDPT method (B).

Figure 3 shows the tracked path for the far test path. While the differences are subtle, the path of the HDPT method appears smoother, especially as seen in the circled regions.

8.2 Comparing the Speed of the Two Methods

The new method runs slightly slower than the DPT method due to the inclusion of higher-order surface information and the use of LU decomposition to find a new surface point. As the “Rate” column in Table 2 shows, though, the new method performs only slightly slower and still runs at haptic rates.

8.3 Showing Improved Stability

In the introduction, we mentioned our concern over the stability of the DPT method in regions of high curvature. To compare the two methods, we used a straight line end effector path over a surface with high curvature.

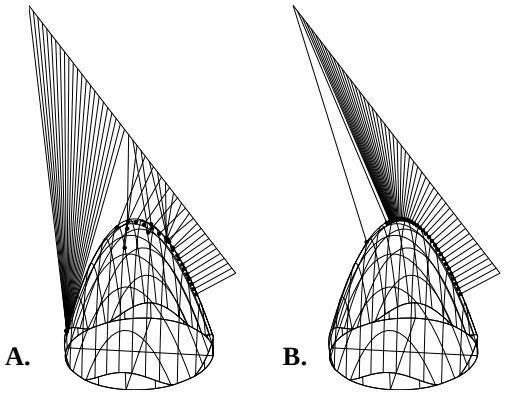

FIGURE 4. The DPT method shows instability when far from high curvature regions. The HDPT method shows greater stability. The first point is off because the seed point was poor.

As expected, the DPT method failed to track the closest point when far from the surface (Figure 4.A). As the end effector approached the surface, it tracked the closest point more effectively. The improved tracking method showed no such instability (Figure 4.B). The first two steps converged poorly because the initial start point was far away, but the HDPT method quickly found the correct closest point. The improved method showed no signs of instability due to the high curvature.

9.0 DISCUSSION

We have shown that using second-order surface information improves the stability and accuracy of closest point tracking during haptic rendering. In the future, we hope to take advantage of this increased stability by generalizing haptic rendering to surface-surface interaction, rather than the current standard of point-surface interaction. Allowing more realistic interaction of objects in the scene will create a more interesting virtual environment.

10.0 CONCLUSION

In this paper, we have presented an improved method of haptically rendering sculptured surfaces. We have extended previous work to take into account second-order surface information. As part of the extension, we discuss in detail the system of equations that tracks the closest point on a surface and highlight surface features that may cause problems. The implemented method runs at haptic rates and shows improved stability and accuracy in closest point tracking.

11.0 ACKNOWLEDGMENTS

The authors would like to thank the members of the Virtual Prototyping group for their ideas, code, and support. Susan Anderson greatly improved the readability of this paper. 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).

12.0 REFERENCES

[1]Adachi, Y. and Kumano, T. and Ogino, K., "Intermediate Representation For Stiff Virtual Objects," in Proc. Virtual Reality Annual Intl. Symp., Research Triangle Park, N.C., pp. 203-210, March 11-15, 1995.

[2]Farin, Gerald. Curves and Surfaces for CAGD: A Practical Guide. Academic Press, Inc. 1993

[3]Hollerbach, J.M., Cohen, E.C., Thompson, W.B., and Jacobsen, S.C. "Rapid Prototyping of Mechanical Assemblies," *NSF Design and Manufacturing Grantees Conference*, Albuquerque, Jan. 3-5, 1996.

[4]Luecke, G.R., Edwards, J.C., and Miller, B.E., "Virtual Cooperating Manipulator Control for Haptic Interaction with NURBS Surfaces," Proceedings of the International Conference on Robotics and Automation, p. 112-117, April, 1997.

[5]Mark, W.R., Randolph, S.C., Finch, M., Van Verth, J.M., and Taylor III, R.M., "Adding Force Feedback to Graphics Systems: Issues and Solutions," in Proc. SIGGRAPH 96, New Orleans, pp. 447-452, August 4-9, 1996.

[6]Mortenson, Michael. *Geometric Modeling*, John Wiley & Sons, New York pp. 305-317, 1985.

[7]Piegl, Les and Tiller, Wayne. *The NURBS book*. Springer-Verlag, Berlin. p. 230, 1995.

[8]Press, William et al, Numerical Recipes in C: the Art of Scientific Computing, 2nd Ed. Cambridge University Press New York, NY. 1996.

[9]Quinlan, Sean. "Efficient Distance Computation between Non-Convex Objects," *IEEE Int. Conference on Robotics and Automation*, pp. 3324-3329, 1994.

[10]Ruspini, Diego, Kolarov, Krasimir, and Khatib, Oussama, "The Haptic Display of Complex Graphical Environments", Proceedings of Computer Graphics, pp.345-352, 1997.

[11]Salisbury, K.,, Brock, D., Massie, T., Swarup, N., and Zilles, C, "Haptic Rendering: Programming Touch Interaction with Virtual Objects". In Procceedings of 1995 ACM Symposium on Interactive 3D Graphics, pp. 123-130. 1995.

[12]Thompson II, T.V., Johnson, D.E., and Cohen, E.C. "Direct Haptic Rendering Of Sculptured Models", in *Proc. 1997 Symposium on Interactive 3D Graphics*, (Providence, RI), April 1997, pp. 167-176.

[13]Stewart, Paul, et al, "CAD Data Representations For Haptic Virtual Prototyping", *Proceedings of DETC'97*, 1997 ASME Design Engineering Technical Conferences, Sept. 14-17, 1997, Sacramento, California.