Spring, 2011 Haptics Class Project Paper presented at
the University of South Florida, May 2, 2011.

Human Spatial Resolution of Phantom Omni

Jairo R. Chimento, Eric M. Tridas

***Abstract*—This project investigates the potential of varying haptic shape resolution using linear splines approximation of high order polynomial curves in the x-z plane to measure spatial resolution of Phantom Omni users. Therefore, optimization of smooth curve approximation can be performed to minimize computational requirements. Generation and rendering of the trajectory, and methodology followed during the test experimentation phase are described in this paper. A total of 16 subjects participated in experimental phase of the study. After a smooth curve was displayed, subjects were asked to indicate whether the subsequent random rendered path trajectories were the same smooth curve or a linear spline approximation of it. The collected data for each subject include their binary choice whether the rendered trajectory was a linear spline approximation or not, time spent interacting with each rendered curve, age, sex and preferred hand. Statistical analysis including multivariable analysis of variance test (ANOVA) of the data was performed using MATLAB. Accordingly, the results demonstrate that users start to perceive a smooth curve when a linear approximation of 24 splines is rendered in the Phantom Omni.**

***Index Terms*—haptic feedback, spatial resolution, linear splines**

I. INTRODUCTION

VIRTUAL reality environments enable users to truly immerse themselves in an imaginary world. The main factors in the realism of virtual environments are the unprecedented advance in computer graphics and the recent developments in force reflecting haptic interfaces that enhance the user's immersion with haptic interaction. This tactile stimulus assists users for haptic exploration where they touch and manipulate virtual objects. The haptic interface exploration in a virtual environment is equivalent to manually explore an object with a probe. In such cases, reaction forces and probe position are the sources of tactile information. The main objective of haptic rendering is to display the tactual feel of a desired virtual object's surface and stiffness. The results from the experiments performed can be used to optimize the approximation of smooth curves in virtual environments while still providing realistic haptic feedback with a minimal amount of computational effort.

II. BACKGROUND

Salisbury et all [1] describe different software techniques for haptic exploration of virtual objects. The techniques proposed in this paper are specifically developed for the Phantom haptic interface. The authors discuss the most important elements for a successful haptic exploration including free-space movement, frictionless surfaces, surface friction, surface curvature and surface texture among others. The paper states that a coarsely meshed polyhedron will be perceived as smooth if a suitable surface normal interpolation scheme is used. Additionally, the authors comment on the results of two experiments where sharp edges were rendered in accordingly to generate a smooth feeling. In [2], variation of stiffness and normal interpolation distance was investigated to analyze the impact in accuracy of surface path tracing with haptic force shading. Test participants were asked to trace continuously a v-shaped line on two flat adjacent planes with various angles between them. The authors derived a formula for the just noticeable difference (JND) of stiffness when using a single point haptic feedback simulated with a SensAble PHANToM arm. A force shading algorithm with the purpose of creating the illusion of a smoothly curved shape which is actually represented for computational purposes as a mesh of polygons was implemented and tested in [3]. The idea of the algorithm lies in the fact that a curved surface can be represented by polygons with the approximated force magnitude, but the force vector direction can be interpolated to vary smoothly across the surface. A Phantom haptic device was used as the display in the experiments where users were given one of 3 polyhedral approximations of the spherical bump or the bump itself. Users were shown the shape on a haptic display and asked to identify either which polyhedral approximation was used or if it were the real shape. Force shading was also applied and the users were asked the same questions. The results shown indicate that without force shading the polyhedral approximations are much easier to identify. Results indicate that although shapes can be easily approximated using polyhedrons, without the further application of some of type of force rendering that approximates the actual curve. Shon and McMain [4] describe several experiments which utilized a Phantom haptic device to draw on 3D surfaces. Several parameters are varied to evaluate the speed and accuracy of a haptic drawing interface. The smoothness of the curve is a measure of how many facets the object surface is approximated by. The results of the experiment conclude that the smoothness of the curve has a direct effect on the accuracy of the drawn shape. The paper also states that the "smooth" curve was made of triangular facets (maximum 2700) and that at this level the user could not discriminate between the smooth curve and one approximated by facets. However, the effect of the application of force shading on the task did not affect the accuracy. Salisbury and Tarr [5] describe a haptic rendering algorithm for implicit functions. The objective of the algorithm is to calculate, in real time, the forces that would be exerted on the user through the haptic interface. The virtual wall is rendered using successive surface points and tangent planes to the points defined by the implicit function S(p)=0. The authors state that this rendering technique can be extended from surfaces defined by implicit functions to other classes of surfaces. In this study, the rendering of the surfaces was executed in two cases. One case considers the effects of friction whereas the other assumes frictionless surfaces. The proposed algorithm was used to render different shapes including ellipses, toroids, cones and a whiffle cube in the SensAble Technologies Inc. Ghost Software Toolkit. The results obtained in the simulations demonstrated that most surfaces required less than 20 seconds per tick.

III. Haptic Display of Curves

In order to measure the spatial resolution of the PHANTOM Omni user, a program was developed in C++ using the Open Haptics Application Programming Interface (API). The program has the ability to display closed loop, high order polynomial polar curves as well as approximations of these curves using linear splines with various levels of approximation resolution in a virtual environment as walls. The level of approximation is measured by the number of splines used to make up the curve in one quadrant. The program displays the curve or approximation and the user indicates whether they believe it is an approximation or not. This data is then stored in a .txt file for later statistical analysis. Generation of the curves and their approximations were executed using a separate script written in MATLAB.

A. Curve Generation

The curves displayed by the program were generated using a MATLAB code developed by the authors. The curves generated were polar, higher order polynomials (n $\geq$ 3) over the range [0, $\frac{\pi}{2}$]. The boundary conditions of the curves are: r(0) = r_0, r($\frac{\pi}{2}$) = r_1, $\frac{dr}{d\theta}|_{\theta=0} = 0$, $\frac{dr}{d\theta}|_{\theta=\frac{\pi}{2}} = 0$ where r_0 and r_1 are randomly generated integers on the range [30,60]. By forcing the slope of the curve to be zero at θ = 0 and θ = $\frac{\pi}{2}$ the curve can be mirrored about the axis in a plane, resulting in a closed, continuous loop. All but the last four coefficients of the generated curve are chosen randomly on the range [-20,10]. These coefficients of the curve equation are then solved to ensure the boundary conditions are met and then saved to a .txt file for later use. Although any order of curve could be generated, it was found that sixth order curves produced the most usable curves.

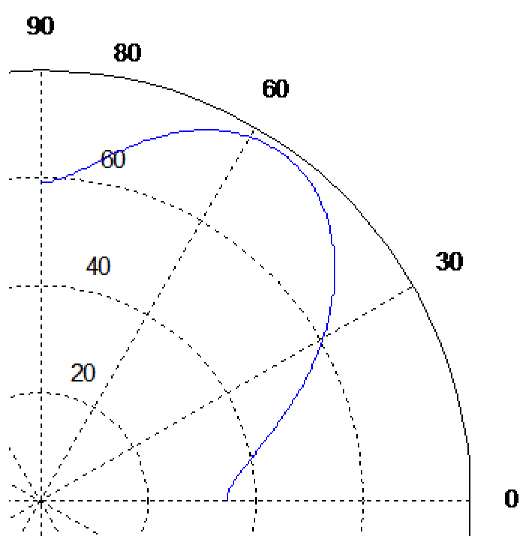


Figure 1. A sample curve generated by the MATLAB script

B. Spline Generation

To save computational time, the equations of the splines used to approximate the curves were generated beforehand again using MATLAB. The number of splines used to approximate the curves ranges from 2^1 splines to 2^{10} for each curve generated in the previous section. This number is doubled between each resolution level so that the number of splines starts at 2^1 increases to 2^2 then 2^3 and so on. The start and end points of each line are all spaced evenly in the radial direction resulting in uneven spline length. The reasoning behind this choice is that the resulting calculation to determine which spline to display is greatly simplified.

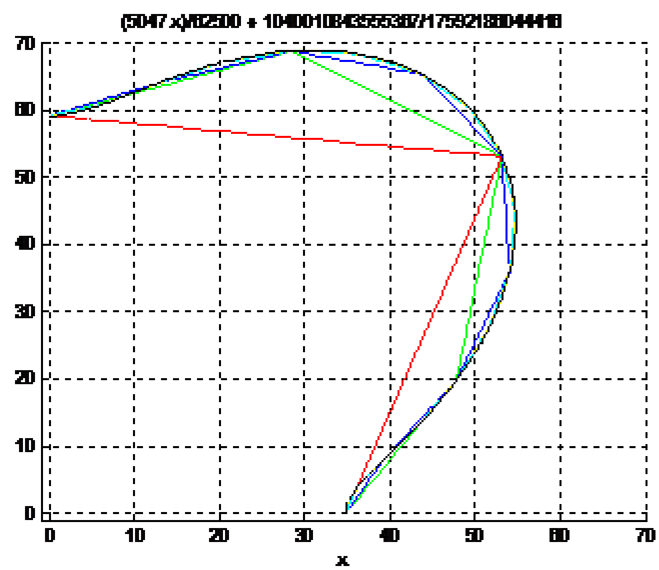


Figure 2. Linear spline approximations of sample curve

For each level of resolution there were four associated .txt files that were also generated; one each for the starting and ending x coordinate values of the individual splines, one for the slopes of the splines and one for the y-intercepts of the splines. The C++ program controlling the haptic device preloads all the curve and spline information into arrays, allowing for very short computation times.

C. Haptic Display on Omni

The Omni Haptic Device has both position sensing and force feedback in the x, y, and z-axes. Curves in this testing environment were displayed in the x-z plane of the Omni. As a result, an illusion is created where the user is exploring the inside of a tube placed with its longitudinal axis oriented vertically. To display the smooth (unapproximated) curve to the user, the position of the haptic interface point (HIP) was converted to polar coordinates. The radius of the position of the HIP is compared to the radius of the curve at the same angle. If the HIP radius is greater than the radius of the curve, force feedback is applied in the direction normal to the curve at the angle the HIP is making with the x-axis. The wall is modeled as a spring; therefore the force is proportional to the distance the user penetrates into the virtual wall. The difference of the two radii is calculated and multiplied by a constant k (equal to 0.2). This constant was chosen to be appropriately stiff to allow minimal penetration into the wall.

A similar approach was used to display the approximations of the curve. The users' position was converted to polar coordinates. Depending on the resolution of the approximation, an "index" of the users' position would also be calculated. This value is the integer part of the number of splines used for that level of resolution times the angle the HIP makes with the z-axis divided by $\frac{\pi}{2}$. This index is the number of the spline within the array of preloaded data, and in this way only one spline is being displayed at a time. A comparison of the radius of the line with the radius of the HIP is performed as in the case of the smooth curve and the resulting distance difference and applied force is calculated.

As stated previously all curves and splines were generated for one quadrant only. All radial comparisons to determine whether the HIP is inside or outside of the wall were performed using the absolute value of the position. A separate variable kept track of the quadrant the user was in and would apply the appropriate changes of sign of the force direction components as necessary.

Table I
TOTAL CURVE LENGTH OF SMOOTH CURVE AND AVERAGE SPLINE LENGTH FOR EACH LEVEL OF RESOLUTION FOR EACH OF THE CURVES. ALL OF THE VALUES ARE GIVEN IN MM.

Curve/Number of Splines	1	2	3	**Average**
Smooth	58.1341	81.4666	69.5795	**69.7267**
2	28.2675	38.6811	33.8924	**33.6137**
4	14.4225	20.0206	17.2824	**17.2418**
8	7.2520	10.1320	10.1320	**8.6891**
16	3.6315	5.0849	4.3470	**4.3545**
32	1.8165	2.5450	2.1741	**2.1785**
64	0.9083	1.2728	1.0872	**1.0894**
128	0.4542	0.6364	0.5436	**0.5447**
256	0.2271	0.3182	0.2718	**0.2724**
512	0.1136	0.1591	0.1359	**0.1362**
1024	0.0568	0.0796	0.0679	**0.0681**

The position sensing resolution of the Omni is approximately 0.055 mm. This corresponds very closely to the minimum value of spline length indicated in Table 1. By using levels of approximation through the entire range of the Omni, a minimum resolution with appropriate realism can be obtained.

IV. TESTING PROCEDURE

A total of 16 subjects participated in experimental phase of the study. The test group included 4 women, 12 men; all of them were right handed and did not have experience with the Phantom Omni. Before the start of the experiment, the test administrators explained to the users the shapes of sample smooth curves and their linear approximation using graphs. Due to abrupt vibrations when rendering coarse linear approximations, noise is generated from the haptic device motors. As a result, the subjects' hearing was isolated with the use of headphones in order to rely on the tactile feedback from the Phantom Omni. The experiment consisted on an executable file where the users had to provide demographic information. Next, the users were asked to explore and familiarize with the sensory feeling of a smooth curve that was displayed using the Phantom Omni at the beginning of each set of curves. The test consisted on 36 trials, 12 for each sample curve, which included 30 linear spline approximations and 3 smooth curves repeated twice. After the smooth curve was displayed, subjects were asked to indicate whether the subsequent 12 random rendered path trajectories were the same smooth curve or a linear spline approximation of it. The collected data for each subject include their binary choice whether the rendered trajectory was a linear spline approximation or not, time spent interacting with each rendered curve, age, sex and preferred hand.

V. RESULTS

The top chart in Figure 3 shows the percent of users that responded that they believed that the curve they were exploring in the haptic environment was an approximation made of linear splines. This is the average of the percentages of the three curves tested. There is a clear negative trend to the data with a slight rise at the higher end. The standard deviations were very large; often larger than the values themselves indicating that the values may be inaccurate. There is also a clear trend in this data. The standard deviation of the first level is about 33%. This amount increases to a plateau of about 50% at a resolution of about 16 splines. Although the trend in averages indicates that users believed the shape they were exploring was the smooth curve as they spline number increased, this seems to be inaccurate as the standard deviation of the responses rose to a level that indicated a large variation in the responses.

One of the reasons for the inaccurate responses of the user may be due to the vibrations of the motors of the Omni. Users may have learned from previous trials that when sweeping over a haptic surface approximated by linear splines with the HIP, a vibratory sensation is felt. A similar sensation is encountered when the user penetrates too deeply into the virtual wall while exploring the smooth curve. If an inexperienced Omni user is pressing with too much force into the curve they may mistake this for the cues of an approximated curve and respond with a false positive. The converse may also be true which could possibly explain the wide variation in data.

The time taken to complete the exploration portion of each trial was measured. This data was run through an ANOVA test and compared against the users, curves, and linear approximation level (noted as trial type). Due to limitations in the ANOVA algorithm, the time taken to respond could not be compared with the response of the user due to the binary nature of the data. Note that the time is given on the x-axis in milliseconds.

The results of the ANOVA indicate that both users and curve had statistical differences amongst themselves with regard to the time it took on each trial. With regard to the users, it was observed by the test administrators that individual users would have various techniques for exploring the virtual surface. Some

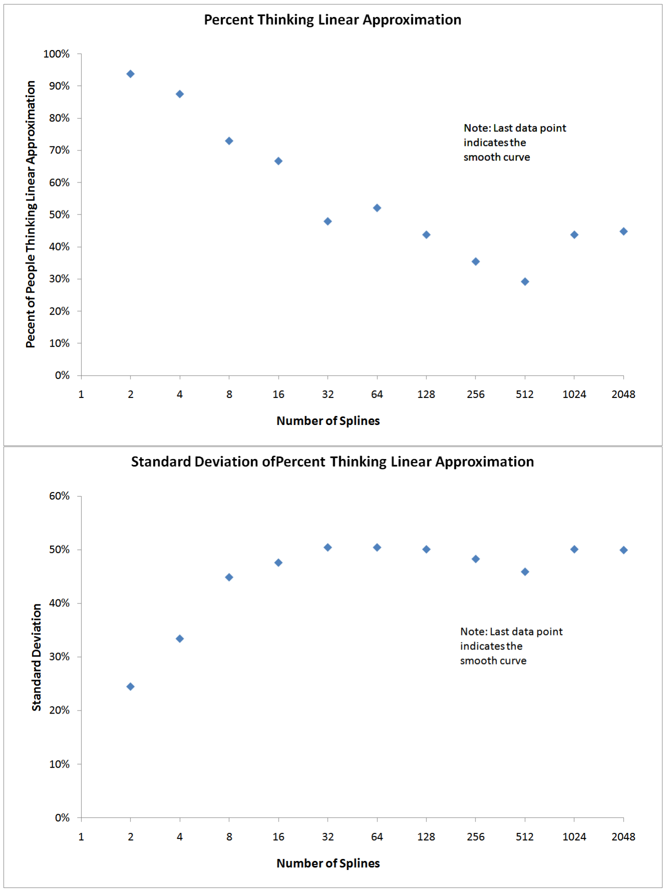


Figure 3. (top) Averages and (bottom) standard deviation of averages of percentages of those who believed the shape they were exploring in the haptic environment was made of linear splines.

Analysis of Variance

Source	Sum Sq.	d.f.	Mean Sq.	F	Prob>F
user	1.21433e+010	15	8.09551e+008	18.47	0
curve	4.69876e+009	2	2.34938e+009	53.61	0
trial type	4.50886e+008	11	4.09897e+007	0.94	0.5058
linear/not	1.91596e+007	1	1.91596e+007	0.44	0.5088
Error	2.39297e+010	546	4.38274e+007		
Total	4.14257e+010	575			

Constrained (Type III) sums of squares.

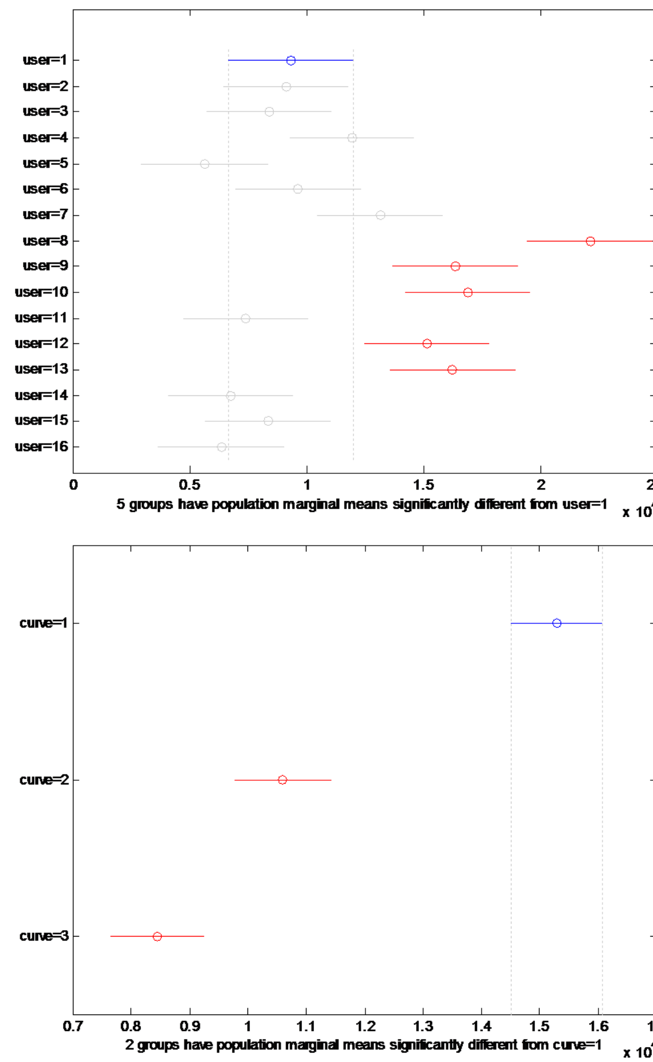


Figure 4. (top) Results of ANOVA test. (middle) Comparison of the average trial times of the users. (bottom) Comparison of the average trial times per curve.

users would move the HIP over the surfaces slowly and steadily. Others would sweep quickly back and forth in a convex area, feeling for abrupt changes in force direction. Because of this it is expected that different exploration techniques will take different lengths of time.

The difference in average times of between curves can possibly be related to the number of inflection points of the curve. Curves 1 and 2 both contain two inflection points on their surface. In the virtual environment, these inflection points feel like bumps on the surface. Without these virtual bumps, the user can quickly explore the surface. This is also qualitatively noted as users felt curve 3 was “the smoothest”.

VI. Conclusions

The results analysis concluded that users start to perceive a smooth curve when a linear approximation of 2^4 splines with the same overall shape is rendered in the Phantom Omni. This result is supported by the fact that the standard deviation of the responses of the users increases to nearly 50% after this level. The negative trend in the plot of the averages versus the number of splines is misleading due to the extremely wide range of answers given for the higher levels of approximation. Complex shaped objects in virtual environments could be rendered using a linear spline length to path length ratio of 0.0625. This approximation minimizes computational effort while maintaining realism in the haptic feedback.

VII. Future Work

For this study, due to time limitations the test did not provide a training section for the inexperienced Phantom Omni users. Moreover, the executable file can be upgraded to minimize handling error or mistyped answers during the experiment.

VIII. Works Cited

[1] Salisbury, K. (1995). “Haptic Rendering: Programming Touch Interaction with Virtual Objects", Proceedings of 1995 ACM Symposium on Interactive 3D Graphics.

[2] Shon, Y. (2006). "Haptic Force Shading Parameter Effects on Path Tracing Accuracy", In Proceedings of HAPTICS.

[3] Morgenbesser, H.B. (1996). “Force Shading for Haptic Shape Perception”, Proceedings of the ASME Dynamic Systems and Control Division, **58**, 407-412.

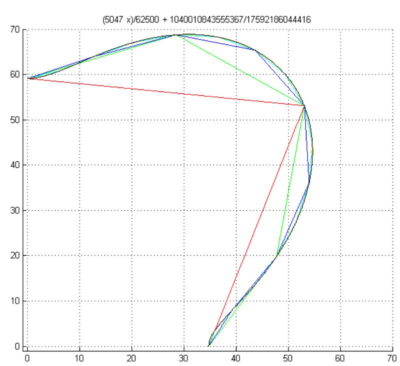
[4] Shon, Y. (2004). “Evaluation of drawing on 3d surfaces with haptics”, IEEE Computer Graphics and Applications, **24**, 40-50.

[5] Salisbury, J. (1997) “Haptic Rendering of Surfaces Defined by Implicit Functions”, Proceedings of the ASME 6th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems.

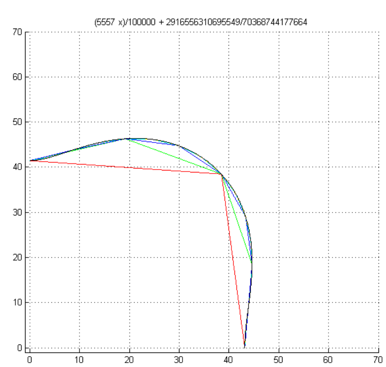
IX. Acknowledgement

The authors wish to express their gratitude to the test subjects for their valuable time and effort to complete the experiments and Dr. Kyle Reed for his guidance and patience during this learning process.

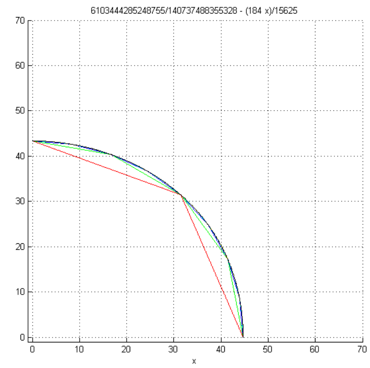
X. Appendix 1: Curve Images



Curve 1 shown both smooth and approximated



Curve 2 shown both smooth and approximated



Curve 3 shown both smooth and approximated

XI. Appendix 2: MATLAB and C++ Code

Please note that the MATLAB and C++ code would not format properly in LyX. These codes are attached as a separate PDF file.