

Force and Contact Location Shading Thresholds for Smoothly Rendering Polygonal Models

Andrew J. Doxon
University of Utah

David E. Johnson
University of Utah

Hong Z. Tan
Purdue University

William R. Provancher
University of Utah

ABSTRACT

Providing tactile feedback in concert with kinesthetic information can dramatically improve one's ability to interact with and explore virtual environments. However, the addition of tactile feedback also enhances the saliency of the vertices and edges of polygonal objects, which are commonly used as model representations in haptic environments. When the polygonal model is an approximate representation of a smooth surface, the heightened response to discontinuities reduces the effectiveness of tactile feedback. This paper addresses this issue by developing a smoothing algorithm, which can be used to provide both tactile and force shading. To investigate the increased sensitivity and effectiveness of our algorithm we conducted four perceptual experiments. Each experiment differed by the type of haptic feedback and whether haptic shading was rendered. Participants were asked to discriminate between an ideal smooth cylinder and its polygonal approximation. The number of polygons used to render the comparison stimulus was increased until the two objects were indistinguishable. Our results are reported as the maximum recommended angular change between adjacent polygons for rendering smooth objects. It was found that the addition of contact location feedback significantly increased the sensitivity of the user to the discontinuities present in polygonal models. Use of shading algorithms was able to significantly reduce the sensitivity to edges in the environment, which allows an increase in the angle between adjacent polygonal surfaces. Our results can be used as a guideline for constructing polygon models that are meant to feel smooth.

KEYWORDS: haptic rendering, perception and psychophysics, tactile feedback devices.

INDEX TERMS: tactile rendering, force rendering, haptic perception and psychophysics, tactile feedback devices.

1 INTRODUCTION

Everyone has experienced the loss of dexterity that occurs when fingers become numb in the cold. Most current haptic interfaces, which provide a purely kinesthetic interaction with virtual environments, result in a similar loss in dexterity, as reported by Frisoli et al. [5]. Providing tactile feedback in combination with kinesthetic information should dramatically improve one's ability to dexterously interact and explore virtual environments.

This paper explores the use of a contact location display, which transmits the contact location between a finger and surface to the user. Figure 1 shows an idealized contact location display (at left)

and a representation of our current 2D display capabilities (on the right). However, when the virtual shapes are faceted approximations to smooth surfaces, as is common in computer graphics [16], the contact location feedback transmits surface discontinuities to the user. Force shading, as developed by Morganbesser and Srinivasan [14], smoothes the kinesthetic feel of faceted models by continuously changing the haptic response direction. However, contact location displays provide a distinct feedback modality which is both sensitive to polygonal edges and not smoothed by force shading.



Figure 1. Concept for contact location feedback. The (left) two-dimensional or (right) one-dimensional center of contact is represented with a single tactile element. A device capable of displaying one-dimensional contacts along the length of the finger (see Figs. 3 and 4) was used in the current studies.

Figure 2 shows the contact location for a finger travelling over a sharp facet in comparison to a smooth curve. Note that the contact location stays at the same location on the finger while on a flat facet (right), moves rapidly along the finger while crossing a vertex (middle), or changes smoothly and continuously while moving along a curved surface (left). In response to the tactile artifacts generated by the polygonal representation, this paper develops an algorithm based on Bezier curve construction that provides a local, smoothed approximation of triangle mesh models for haptic and contact location display. This "tactile shading" algorithm smoothes out the edges of adjacent polygons and allows polygonal objects to feel smooth – both kinesthetically (using force shading) and tactilely (using tactile shading). In our case, tactile shading means smoothing out the motions of the computed contact location. Our presented shading algorithm can also be used to provide force shading as an alternative to Morganbesser and Srinivasan's algorithm.

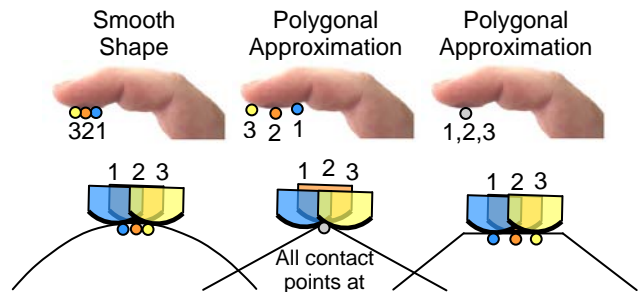


Figure 2. Contact location movement over a smooth curved surface (left), a vertex (middle), and a facet (right). The top shows a view of the fingerpad with a series of displayed contact locations, corresponding by color and number, to the virtual finger positions below.

adoxon@gmail.com, dejohnso@cs.utah.edu,
hongtan@purdue.edu, wil@mech.utah.edu

This work builds on several recent studies to provide combined tactile and kinesthetic feedback [6, 7] including ones using contact location tactile feedback in combination with a Phantom force feedback device [15, 12]. While prior studies using contact location feedback were successful in showing the potential of this type of tactile feedback, these studies were conducted using highly specialized virtual environments composed of a collection of constant curvature segments [12]. In contrast, the present work extends the usability of contact location displays to general polygonal environments.

The contact location display combined with the new tactile shading algorithm provides an opportunity to explore human perceptual capabilities in the context of combined kinesthetic and tactile feedback, and to investigate guidelines for constructing faceted representations of smooth surfaces based on these perceptual capabilities. We specifically focus on the model parameter requirements to ensure that smooth models are perceived as smooth when rendered as a collection of polygons. To make this evaluation we have conducted a series of human subject experiments by varying the number of polygons and the type of “haptic shading” that is used to represent a virtual cylinder. The tests were conducted both with and without tactile feedback. These tests are conducted in a 2D haptic environment modeled with line segments, but the results of these experiments also extend to 3D environments. Our results prescribe guidelines for the maximum angle change between adjacent polygons for a surface to feel smooth under each rendering (shading and feedback) condition. The angle between adjacent polygons was chosen as our primary metric as it is a major source for the discontinuity in force and tactile cues that are sensed as one slides a finger over a polygonal model.

The remainder of the paper provides a brief background of devices with combined kinesthetic and tactile feedback and common haptic rendering environments. We then provide an overview of our tactile display and the algorithms utilized to render the haptic interactions with the device. This is followed by a description of our experimental methods for evaluating our algorithms, along with experimental results and discussion.

2 BACKGROUND

This section provides a brief background concerning literature that is most relevant to our study.

Combined Tactile and Kinesthetic Feedback

A number of researchers have conducted studies with combined tactile and kinesthetic feedback, including the authors of the present study. Salada et al. conducted several studies that investigated the use of slip or sliding feedback in combination with kinesthetic motions [17]. Since then, others have also developed slip displays and integrated them with kinesthetic force feedback devices [7, 18]. One of these studies has also investigated providing tactile feedback from a pin array in combination with kinesthetic feedback [7]. Other interesting approaches to tactile/kinesthetic display include research on displaying the local object surface tangent [4, 6]. Dostmohamed and Hayward present a device that utilizes a gimballed plate to represent the local surface tangent plane of virtual objects. The motion of the gimballed plate is coordinated with the user’s kinesthetic motions to display curved objects [4]. As a relatively sophisticated adaptation of the work by Dostmohamed and Hayward, Frisoli et al. present a miniaturized finger-based tilting plate tactile display that can be attached to a kinesthetic display [6]. Finally, the author’s prior studies have shown the potential of contact location feedback for enhancing object curvature and motion cues [15].

Haptic Algorithms

The algorithms that are utilized in this paper build directly on prior work from the haptic rendering literature. The general formulation of our polygonal models and proxy method for rendering contact forces follow directly from methods outlined by Ruspini and Khatib [16]. A slight alteration was made to their presented algorithm to compensate for the angle change between polygons. For our implementation, the proxy is shifted slightly to allow interaction on vertices. Testing showed this addition did not affect the final results of our experiments as they pertain to the proxy method.

The force shading algorithm used in our experiment was developed by Morganbesser and Srinivasan [14]. This algorithm interpolates the surface normals from one surface to the next to guarantee a continuously changing force vector across the surface of the object. This method weakens the “popping” effect felt when passing over a vertex or edge of polygonal objects. Morganbesser et al. found that their force shading algorithm helped give the sensation of a smoother object, but they never determined the conditions under which a smooth and a shaded polygonal object were indistinguishable. Our experiments were designed to also provide this information.

3 DEVICE DESCRIPTION

The concept for contact location feedback is presented in Fig. 1, where rather than providing all possible tactile information to a user, only the center of contact is rendered. The hardware utilized in the following experiments consists of a SensAble Phantom Premium 1.5 and a contact location display. The Phantom is used to render environment forces. The contact location display is used to render the current contact position on the finger. The device utilizes a 1 cm diameter delrin roller as a tactile contact element. This ensures that only the contact position is provided to the user and no skin stretch is experienced. The position of the roller on the finger is actuated via sheathed push-pull wires attached to a linear actuator mounted on the user’s forearm. The display’s contact roller is attached to the Phantom via a one dimensional gimble with sensed tilt angle. The roller is suspended beneath the fingerpad by the drive wires so that it does not touch the user’s finger until contact is made with a virtual object. Contact forces, provided by the Phantom, push the roller into contact with the user’s fingerpad. An open-bottom thimble is used to attach the device securely to users’ finger and also provides a mounting point to anchor the sheaths of the spring steel drive wires. Several interchangeable thimbles, which allow a wider range of finger sizes, were created using fused deposition modeling (FDM) rapid prototyping.

The linear actuator is located on the user’s forearm to prevent any possible device vibrations from being transmitted to the user’s fingertip receptors and to reduce the device inertia located at the fingertip. The linear actuator utilizes a Faulhaber 2342CR DC motor and a 3.175 mm pitch leadscrew to provide approximately 2 cm of linear motion with approximately 0.8 μ m of resolution and a bandwidth in excess of 5 Hz. A prototype of the device can be seen in Figs. 3, 4, and 11. The fingertip portion of the device is shown in Fig. 4.

The device’s motor is driven by AMC 12A8 PWM amplifiers that are controlled using a Sensoray 626 PCI control card. The device’s PID controller was run at 1 kHz and was programmed in C++ and executed under Windows XP using Windows multimedia timers. Further details about the design and control of this device may be found in [15].

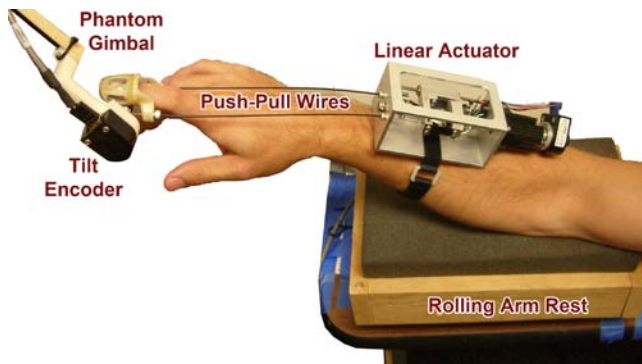


Figure 3. Contact location display prototype attached to a Phantom robot. The user's elbow is supported by a rolling armrest.

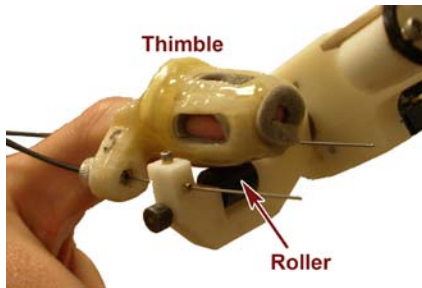


Figure 4. The user's finger is secured to the contact location display via an open-bottom thimble.

4 HAPTIC RENDERING

An overview of the methods used for kinesthetic rendering as well as the details of our shading algorithm is provided below.

4.1 Kinesthetic Rendering

Our haptic rendering system uses a basic geometric interaction model to determine where the virtual finger is located and if it is in contact with the surface of a polygonal object. The interaction of the two virtual objects tells the controller how much force to apply with the Phantom and where to place the roller, and is implemented, as typically, with the use of a proxy [16]. Additions were made by slightly shifting the proxy's location to allow interaction on vertices. This eliminates the contact location jumps commonly associated with the proxy method and allows smoother control of the device. Testing showed this addition had no effect on the experimental results.

Rather than interacting with a full 3D polygonal model, this system uses a 2D polygonal object. The virtual finger is modeled as an arc segment of a circle to allow contact positions to be computed. The arc segment for the virtual finger model utilized in our experiments was of radius 20 mm with an arc length of 16 mm (see Figs. 2, 8, 9 and 10). In all our experimental conditions, kinesthetic rendering was implemented in the same manner. However, when utilizing our shading algorithm, the user is interacting with a polygon that represents the instantaneous local tangent plane estimate rather than a polygon surface of the parent model. This does not pose an issue for proxy-based kinesthetic rendering.

4.2 Shading Algorithm Utilizing a Reparameterized Bézier Approximation

We present a method for using the polygonal model as the control polygon of a series of quadratic Bézier curves. Bézier curves have several valuable properties that allow us to link them and receive

a smooth surface [2]. In the presented experiments, our shading algorithm was used to provide tactile shading, but could also be used as a substitute for the methods presented by Morganbesser and Srinivasan [14] for force shading.

The addition of contact location feedback to kinesthetic feedback increases the user's sensitivity to vertices on the polygonal object. The motion of the displayed contact location is portrayed in macro scale in Fig. 2. Note that the contact does not move when the finger is on a facet (right), but moves rapidly when contacting a vertex (middle). When a moderate number of polygons are used, this transition between facets manifests itself as small bursts of roller motion. Because of this, a large number of line segments are required to create the perception of a continuous smooth curve. The following algorithm was developed to create a smooth contact trajectory from a polygonal object.

While it is possible to fit smooth surfaces to a polygonal model; in general, mapping parametric patches onto arbitrary geometry remains a research topic [3]. In addition, computing the contact location between two curved surfaces requires robust numerical methods that may run too slowly for haptic applications [10]. Instead, our approach computes a dynamically updated tangent line at the point of contact. This tangent is one that would be produced from an underlying smooth Bézier model, but that smooth curve is never directly computed or interacted with. This allows the continued use of the original polygonal interaction model rather than requiring the creation of a new one.

As the user slides across the model the contact point traces a new smooth surface created by the algorithm. This resulting surface is shown in Fig. 5. The black lines represent the original polygonal model and the red, thicker, curved lines represent the shape of the resulting interaction model. The local centers, and their corresponding grey regions, denote separate applications of the algorithm. The resulting curve is built from these patches.

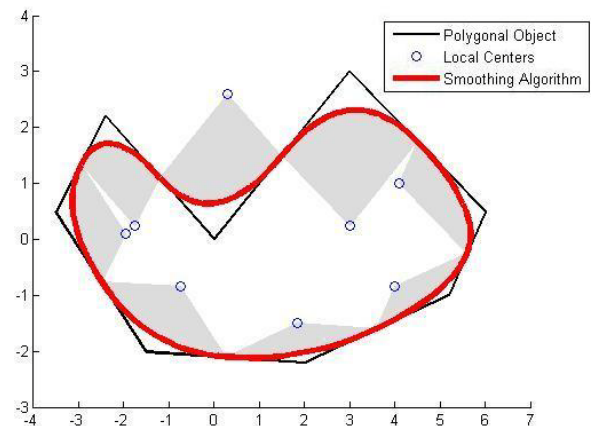


Figure 5. The original polygonal model (black) and the smooth interaction model (red).

The basis behind this smoothing algorithm is a quadratic Bézier curve. A Bézier curve is a method of producing a geometrically continuous curve out of a defining polyline. The de Casteljau algorithm is an elegant constructive algorithm to compute a point and tangent on the Bézier curve based on a parameter value, t [2]. Varying the parameter value from zero to one traces out the Bézier curve.

Using Bézier functions, three points are required to define a quadratic. For this simple case, the de Casteljau algorithm is applied twice. Each repetition of the de Casteljau algorithm defines a new set of points containing one fewer point than the previous set. In this case the first iteration defines a line segment

tangent to the curve. The second iteration defines a point, on that line segment, that lies on the curve. The equations for computing the two points that define the ends of the line segment are shown below in (1). The equation defining the point on the curve is shown as (2). As can be seen from the equations, the points are each dependent on the previous step's values. The labels used in these equations correlate to those shown in Fig. 7. The subscripts denote the location of the point.

$$\begin{aligned} P_{12} &= P_1(1-t) + P_2t \\ P_{23} &= P_2(1-t) + P_3t \end{aligned} \quad (1)$$

$$P_{1223} = P_{12}(1-t) + P_{23}t \quad (2)$$

Since our particular algorithm is required to return only the tangent line, the equations in (1) are all that are used. Thus computing the new tangent from a given parameter value is a single step. In order to retain tangent continuity over patch boundaries, our algorithm forms the Bézier patch from a vertex and the midpoints of each line segment connected to it. Figure 6 shows three tangent line segments at $t = 0.25, 0.50,$ and 0.75 at each vertex.

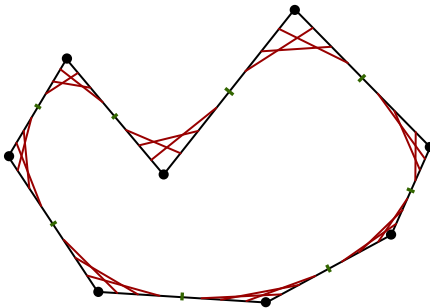


Figure 6. An arbitrary polygonal shape. Three tangent line segments are shown for each point. Only one tangent will be in existence at a single instant in time.

Unfortunately, our algorithm, given a contact location, needs a parameter value from which it can derive the tangent, while de Casteljau provides the tangent given a parameter. One method for finding this parameter value without an explicitly defined curve is nodal mapping [9]. The method relates positions on the defining polygonal model to curve parameter values. However, nodal mapping creates an uneven parameterization that leads to parametric discontinuities while inside of the object. Instead, this paper develops a radial parameterization that smoothly and uniformly changes position and contact location along the curve.

Based on these requirements, our algorithm has two main parts. The first is to find the current parameter value based on the tangent found in the prior iteration. The second is to compute the new interaction tangent from the current parameter value. Because these steps form an iterative process, the accuracy, and consequently the smoothness, depends on the update rate, as with many haptic systems. Details of these two steps are provided below.

4.2.1 Computing the Current Parameterization

The first step in our algorithm is to find the current contact point. This is done by treating the previously computed tangent line as the current surface. If the finger is not currently in contact with the surface, it is projected toward the surface and treated as though it is in contact for this time step. When moving, this position represents a small differential distance along the tangent

line and thus is a reasonable first approximation for determining the user's current contact position on the surface. Regardless of the interaction model used, only the contact position matters and no forces need to be computed or applied in this step.

The explanation of the second step in our algorithm, computing the current parameterization, is best facilitated by first defining a few geometric labels (see Fig. 7).

The two line segments that are adjacent to the current vertex of interest are labelled L_1 and L_2 . The three control points ($P_1, P_2,$ and P_3) are the vertex and the midpoints of L_1 and L_2 . The arrows denote the direction that P_{12} and P_{23} will travel for increasing values of t .

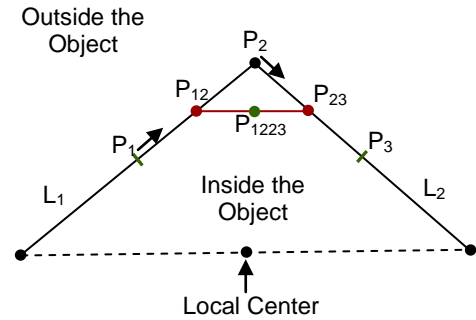


Figure 7. Basic labeling scheme used in our shading algorithm.

The local center is an integral part of the radial parameterization. As such the local center cannot be located on $L_1, L_2,$ or the resulting curve. While the local center may be placed almost anywhere, ideally it should be placed at the center of curvature of L_1 and L_2 . The center of curvature can be found by computing the intersection of lines normal to L_1 and L_2 placed at their respective midpoints. Placing the local center at the center of curvature ensures the highest numerical precision. Another convenient location for the local center is at the midpoint of the side opposite the vertex, as shown in Fig. 7.

Once the current contact position has been established, the current L_1 and L_2 lines need to be determined. These lines are likely the same ones as those from the last iteration. There are two conditions that will cause new lines to be selected. The first of these conditions is when multiple contact points exist on non-adjacent line segments. The second condition that causes a new line to be selected occurs quite frequently. This condition occurs just as the user passes over the midpoint of L_1 or L_2 . At this point the next vertex is now closer to the contact point and its corresponding line segments become the new L_1 and L_2 . A new local center is defined for this new set of L_1 and L_2 and the conditions are checked again. If constructed carefully, L_1 and L_2 will change at most once in a given iteration.

To better demonstrate these condition checks, the current contact point must lie within the shaded region shown in Fig. 8. If the current contact point is outside of this region, one of the two conditions has been met and needs to be resolved before continuing.

Once L_1 and L_2 have been verified, all that is left is to compute the corresponding parameter value. This is done by computing the angular fraction ($\lambda = \alpha/\beta$) between the current contact point and the closer line segment with respect to the local center. In Fig. 9 the closer line segment is L_2 and the angular fraction is approximately 0.2. Equations (3) and (4) show how to calculate the angular fraction for L_1 and L_2 respectively. P_0 and P_4 , as shown in Fig. 9, have been added for reference reasons and are only used in this step. Note that the fractions found while the contact point projects to P_2 will always be either 0 or 1. The fractions defined at P_1 and P_3 will be dependent on the chosen position of the local center.

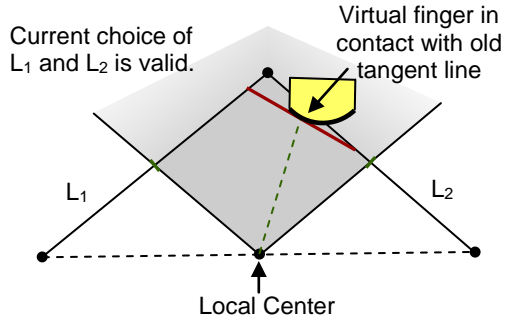


Figure 8. Shaded region bounded by the local center and the midpoints of L_1 and L_2 that must contain the computed contact point for L_1 and L_2 to be valid choices.

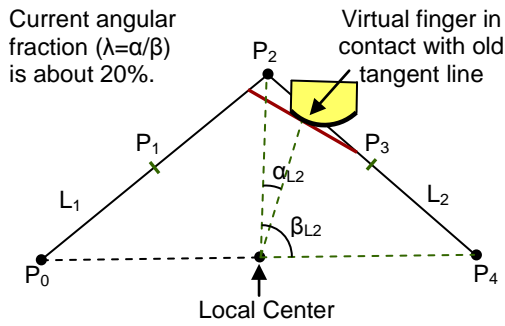


Figure 9. Computing the angular fraction.

$$\lambda_{L1} = \frac{\alpha_{L1}}{\beta_{L1}} = \frac{\theta_{contact} - \theta_{P0}}{\theta_{P2} - \theta_{P0}} \quad (3)$$

$$\lambda_{L2} = \frac{\alpha_{L2}}{\beta_{L2}} = \frac{\theta_{contact} - \theta_{P2}}{\theta_{P4} - \theta_{P2}} \quad (4)$$

The angular fraction is then placed into an equation to compute the parameter value t . Because the angular fraction differs when the current contact point projects to L_1 vs. L_2 , two separate equations may be used. These equations must meet three specific criteria for the system to be continuously smooth, or G^1 continuous.

The first criterion is that for a projection onto P_2 , both equations should evaluate to the same value of t . This guarantees continuity between the two functions and a smooth resulting curve.

The second criterion requires that $t = 0$ when the current contact point projects to P_1 . The third criterion is similar to the second in that it requires $t = 1$ when projected to P_3 . These last two criteria force the resulting curve to end at P_1 and P_2 and force the curve to be tangent to L_1 and L_2 at its ends. This allows the resulting curve to attach to the adjacent curve patches with G^1 continuity.

4.2.2 Computing the New Contact Surface

The second part, updating the tangent line segment, is to apply equation (1), shown earlier and again below, to the particular value of t in this iteration. Unfortunately, simply applying the equations in (1) will result in a line segment ending at P_1 or P_3 . Finding a point on the line segment past P_1 or P_3 in this case is impossible. Thus the system cannot transition to the next curve patch. This problem can be bypassed by instead using the infinite

line defined by the line segment. From here, the new tangent is created and the proxy based haptic interaction model takes over to compute interaction forces in a manner similar to those outlined by Ruspini and Khatib [16].

$$\begin{aligned} P_{12} &= P_1(1-t) + P_2t \\ P_{23} &= P_2(1-t) + P_3t \end{aligned} \quad (1)$$

5 DISCRIMINATION THRESHOLDS FOR POLYGONAL SMOOTH SURFACES

Four experimental conditions were evaluated in order to better understand the requirements for rendering smooth objects when using polygonal models. An adaptive procedure was utilized to assess when participants could no longer distinguish between the polygonal model and the parametrically smooth reference surface. These tests were conducted with force feedback alone and with combined tactile and force feedback. Force and tactile shading were also specifically investigated. Forces were rendered using a Phantom, and tactile feedback was rendered using our contact location display.

The first two conditions parallel the work by Morganbesser and Srinivasan [14] and utilize solely force feedback. In these conditions, the contact roller of the contact location display was simply held in its middle position. Condition 1 (C1) utilized a set of polygons (line segments) to approximate a smooth surface. This was done to establish a baseline for the number of segments required for a polygonal model to “feel smooth.”

Condition 2 (C2) was identical to Condition 1, but also included the addition of force shading, as described by Morganbesser and Srinivasan [14]. One slight difference from [14] is that we utilized a curved finger model as opposed to a single contact point. This extends the work described by Morganbesser and Srinivasan [14] to a more complete state that can more readily be used by hapticians when constructing virtual models of smooth surfaces.

The remaining two conditions utilize the contact location display. Condition 3 (C3) has participants evaluate polygonal models with force and tactile feedback (and no shading/smoothing) and the results can be compared to those of Condition 1 to examine the effect of added contact location feedback.

Condition 4 (C4) has participants utilize force and tactile feedback to evaluate polygonal models with tactile shading, but without force shading. This condition was designed to evaluate the influence of tactile feedback and can be compared to all 3 other conditions.

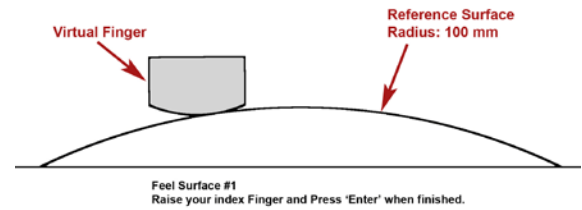


Figure 10. Screen capture of the smooth reference object used during training that preceded each test condition.

The experiment utilized a paired-comparison (two interval), forced-choice paradigm, with a 1-up, 2-down adaptive procedure [13]. On each trial, the participant was presented with two objects, the smooth reference object and the comparison object with a polygonal representation, in a random order. The participant’s task was to indicate which of the two shapes was the smooth object. The number of line segments was decreased after one incorrect response (making the difference between the reference and comparison objects larger, and therefore the task easier) and

increased after two consecutive correct responses (making the task more difficult). The threshold so obtained corresponds to the 70.7% point on the psychometric function [13]. The reference stimulus was a mathematically correct arc segment of a circle (see Fig. 10), while the comparison stimulus was a polygonal approximation of the same arc segment. Only the top portion of the circle was haptically rendered. The rendered arc section was 0.902 radians of a 100 mm radius circle, giving approximately 90 mm of travel space. Contact location on the virtual finger was calculated over a 16 mm arc length of the 20 mm radius finger model and linearly mapped to be displayed over 16 mm of travel along the length of the participant's finger.

Each condition was conducted as follows. The participant would first feel stimulus #1. Once they were finished exploring they would then raise their index finger off the surface and press the 'Enter' key to indicate they were ready for stimulus #2. After feeling the second stimulus they would again raise their index finger and press '1' or '2' and then 'Enter' to indicate which of the two stimuli was the smooth object. Then a new set of comparisons was presented. The order of the reference stimulus presentation was randomized.

The experiment continued until the participant had finished 11 reversals (a reversal occurred when the number of segments was increased after a decrease, or vice versa). A large step size was used for the first three reversals. A reduced step size was used for the remaining eight reversals for better accuracy in determining the discrimination threshold. The step sizes chosen for each condition were calibrated during pilot testing and fixed for all participants in the study. The discrimination threshold was computed from the last 6 reversals.

A Latin Squares reduction of the system was utilized to reduce the number of permutations for balancing testing order in which participants completed the four experimental conditions. The testing apparatus, as shown in Fig. 11, was obscured by a cloth cover so that the user would not be able to see either the haptic or tactile device. Instructions were posted on the screen to remind the user where within each comparison they were and how to proceed, but no other visual feedback was provided. White noise was played over headphones to block all auditory feedback, except for audio cues that were provided to indicate the transition between stimuli. Participants were given as much time as they desired to explore each stimulus, but were not permitted to go back to the first stimulus once they had proceeded to the second.

Twelve right-handed individuals (3 females) between the ages of 19 and 41 participated in the experiments. They took an average of about 10 minutes and 42 trials to complete each condition. Additionally, no learning effects due to the ordering of the conditions were observed.



Figure 11. Experiment test setup (cover pulled back for clarity)

While our experiment evaluated the number of polygons needed for a polygonal surface to be indistinguishable from a reference smooth surface, the results are also reported below in terms of the more general metric of the angle between adjacent polygons.

6 RESULTS

Two representative data sets for one participant are shown in Fig. 12. Note that this participant had some difficulty in condition 2 (force feedback with force shading). However, both of these plots still fall within the range of expected participant performance. In all cases, each participant managed to stabilize their performance before completing the 11 reversals.

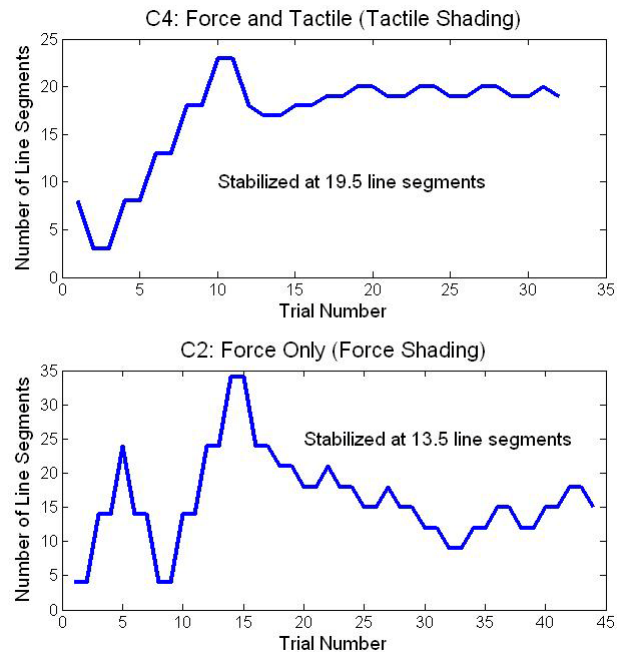


Figure 12. Two collected data plots showing (top) nearly ideal data from one participant and (bottom) less ideal data from the same participant who had difficulty with C2.

The data collected from the 12 participants passed an omnibus ANOVA test ($F(44,47) = 47.76, p < 0.001$). This implies independence between all four conditions and allows the use of Tukey's test to determine if the results are significantly different. The data was subsequently analyzed for statistically significant differences using Tukey's test with $\alpha = 0.05$. The average number of line segments for each threshold was the highest for C3 (257.3), followed by that for C1 (104.1), and the lowest for C2 and C4 (16.3 and 15.6, respectively).

It was found that C3 (force and tactile rendered) was significantly different than all other conditions. C1 (force only rendered) was also significantly different than all other conditions. The two shading conditions (C2 and C4) were not significantly different from each other. Table 1 shows the mean discrimination thresholds and the corresponding 95% confidence intervals for the collected data. If the angle between polygons used is less than the lower end of the 95% confidence interval (for example less than 0.37° for C1) then 95% or more of people should sense the model as perfectly smooth. Figure 13 plots these means and confidence intervals, and highlights the significant differences among the four conditions.

Table 1. Means and 95% confidence intervals for all four test conditions, showing the number of line segments needed for a polygonal surface to be indistinguishable from the smooth reference surface and the corresponding angle between adjacent line segments in degrees (in parentheses).

	C1	C2	C3	C4
	Force Only	Force Only with Force Shading	Force and Tactile	Force and Tactile with Tactile Shading
Mean	104.1 (0.5°)	16.3 (3.4°)	257.3 (0.2°)	15.6 (3.5°)
95% Confidence	± 35.32 (+0.25°, -0.13°)	± 1.99 (+0.44°, -0.35°)	± 63.20 (+0.065°, -0.040°)	± 3.85 (+1.09°, -0.66°)

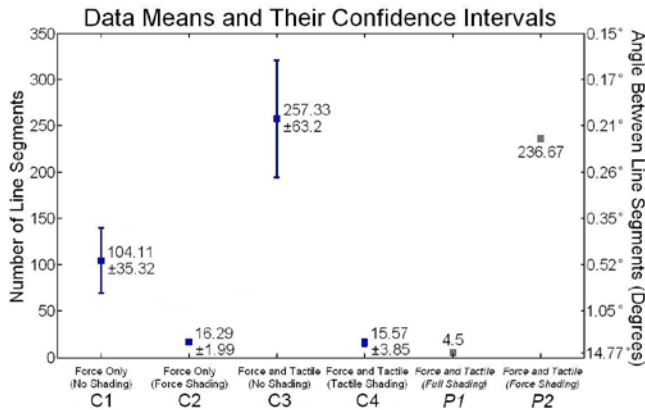


Figure 13. Plot of the mean and 95% confidence intervals for each test condition showing the number of line segments at which the polygonal model was indistinguishable from the smooth reference surface. The error bars are not linear when interpreting results based on the angle between segments.

As mentioned earlier, a more general and useful metric that can be taken from our results is the angle between adjacent polygons, as this can be applied to other generic polygon models. This measure corresponds to the way discontinuities between line segments connect. This concept is similar to that proposed by Morganbesser and Srinivasan [14] with one important distinction: The tactile feedback is felt as short rolling bursts as the user crosses the vertexes, due not only to the *instantaneous* changes in force direction but also changes in the physical shape itself. Table 1 shows the angle thresholds corresponding to the line-segment thresholds in parentheses. The same angle differences are shown in Table 2 where test conditions are organized according to rendered and shaded variables. Two additional threshold values are shown from pilot testing (P1 and P2, collected from two participants) for comparison and discussion later.

Table 2. Estimated angle difference, in degrees, between adjacent segments to create a curved surface that feels smooth.

	Rendered Condition	
	Force Only	Force and Tactile
No Shading	0.5° (C1)	0.2° (C3)
Force Shading	3.4° (C2)	0.2° (P2)
Tactile Shading	NA	3.5° (C4)
Force and Tactile Shading	NA	14.8° (P1)

7 DISCUSSION

First, we compare the measured threshold for C1 to prior work. Our results are not directly comparable to that of Morganbesser and Srinivasan, as they only tested to show improvements in perceived smoothness and explored only using up to 3 polygons. However, it is interesting to compare C1 to prior work on discriminating the angle between sequentially applied force vectors. Barbagli et al. report a discrimination threshold of 28.4° for sequentially applied force vectors, which is nearly two orders of magnitude larger than the thresholds we report for the instantaneous changes in force orientation experienced in C1 (0.5°) [1]. This is not surprising though as people have much greater sensitivity to changes presented in rapid succession [8]. Our task also utilized active rather than passive sensing in making perceptual judgements, which is also expected to provide greater perceptual sensitivity [11].

Several trends can be observed from the data presented in Tables 1 and 2. First of all, the addition of tactile feedback greatly increases one's sensitivity to edges and vertices in the system, as seen by a pair-wise comparison of the thresholds for C1 and C3 and those for C2 and P2 in Table 2. This increased sensitivity is undesirable when smooth surfaces are rendered because more line segments are needed in order to render a smooth surface with a polygonal model, causing an increase in computation time and a decrease in rendering performance.

Fortunately, force and/or tactile shading can decrease one's sensitivity to edges and vertices, as seen by the significant difference found between the thresholds for C1 and C2 and those for C3 and C4. This significant difference shows that both the force shading algorithm, developed by Morganbesser and Srinivasan, and our shading algorithm, presented in Section 4, significantly reduce the needed number of line segments to make a polygonal object feel smooth. Note that the shading algorithm developed in Section 4 also is capable of rendering force shading and can therefore reduce the number of line segments further as indicated by the threshold for P1 shown in Table 2.

Another interesting observation is that when both force and tactile feedback signals are present, people appear to rely more on tactile than force information to judge the smoothness of a surface. If the participants completely ignored the tactile sensations, then there should be no difference between the thresholds for C1 and C4 in Table 2. Instead, the participants judged polygonal surfaces in C4 to be smoother based on shaded tactile feedback, even though normal force discontinuities still existed as in C1. This indicates that the tactile sensations may carry more weight in haptic perception than the force irregularities. In fact, in the presence of unshaded tactile information (C3 and P2), there appears to be no significant benefit from applying Morganbesser and Srinivasan's force shading algorithm in P2 (see Table 2).

Utilizing both force and tactile shading can significantly reduce the required polygonal model size further, as is shown for P1 in Table 2. This condition was not evaluated in our main experiments because there were not enough stimulus levels (in terms of line segments in a polygonal model approximating a smooth surface) to implement the adaptive procedure. While the minimum model size would be 2 line segments, it took at most 5 segments for the polygonal model to feel smooth. This could be because our shading algorithm also more closely approximates the ideal shape and size of the ideal cylinder in our test conditions, hence also eliminating proprioceptive cues from being discerned.

To summarize, the use of shading algorithms can lead to a significant reduction in the size of polygonal models approximating smooth object surfaces without introducing noticeable artifacts. Referring to the angle thresholds for C1-C4 in

Table 2, it is shown that the addition of shading allows angle differences between adjacent polygons to reach $\sim 3^\circ$ before any non-smoothness is perceived as opposed to only $0.2\text{-}0.5^\circ$ when shading is not used. Furthermore, our pilot tests (P1) indicate that polygon models with as much as 15° between adjacent polygons are perceived to feel smooth while still properly representing object shape and size if our smoothing algorithm was utilized to apply both tactile and force shading. This can clearly have a huge impact on reducing the necessary size of a haptic model, without sacrificing the fidelity of the haptic interaction. Although our results were obtained with the contact location display, the angle thresholds are likely applicable to other types of tactile displays including those that render the tangent planes of a curved surface [4, 5].

8 CONCLUSIONS

In this paper we present experiments that utilize combined force and tactile feedback using a curved finger model and a polygonal object model. The addition of tactile feedback has the potential to increase immersion and dexterous capability with haptic systems. Our experiments show that the addition of tactile feedback significantly increases one's sensitivity to discontinuities in the environment and provides a new mode to gain haptic information. Because of this increased sensitivity, it may be necessary to smooth out unintended modelling artifacts that will be present when interacting with polygonal models – especially when large numbers of polygons are utilized in order to keep model size tractable. To eliminate these tactile artifacts and improve immersion, a tactile shading algorithm was developed and presented. The effect of our shading algorithm on the way polygonal models are perceived was evaluated alongside more traditional force shading. This algorithm was determined to reduce the required haptic model size significantly while still allowing greater immersion.

We have also investigated the perceptibility of surface discontinuities by comparing the use of force feedback alone to combined force and tactile feedback with and without the respective shading algorithms. Very small angle differences between adjacent polygons ($0.2\text{-}0.5^\circ$) were required when shading was not used. Thus, large numbers of polygons were needed for these models to feel smooth. The addition of force and/or tactile shading significantly reduced the required model size as can be seen in Fig. 13 and Table 2. Either form of force or tactile shading allowed a relatively large angle difference between polygons ($\sim 3^\circ$), while greater angles between polygons (15°) were possible if both force and tactile shading was simultaneously applied, thereby requiring a significantly smaller number of polygons to represent a given haptic model.

Future work will expand the shading algorithm presented here for use in three dimensional models. We are also pursuing new tactile feedback device designs that are capable of rendering positions both along and across the length of the finger in two dimensions, as shown in Fig. 1 (left). The development of new hardware and software is expected to greatly enhance our ability to render surface details in a virtual haptic environment.

9 ACKNOWLEDGEMENTS

This work was supported, in part, by the National Science Foundation under awards IIS-0746914 and IIS-0904456.

REFERENCES

[1] F. Barbagli, K. Salisbury, C. Ho, C. Spence, and H. Z. Tan. Haptic discrimination of force direction and the influence of visual information. In *ACM Transactions on Applied Perception*, Vol. 3, No. 2, pp. 125-135, 2006.

[2] E. Cohen, R. F. Riesenfeld, G. Elber. *Geometric Modeling with Splines: An introduction*. Mass: AK Peters, ch. 5-11, 2001.

[3] J. Daniels, C. T. Silva, J. Shepherd, and E. Cohen. Quadrilateral mesh simplification. In *ACM SIGGRAPH Asia 2008 Papers*, Singapore, December 10-13, 2008.

[4] H. Dostmohamed and V. Hayward. Contact location trajectory on the fingertip as a sufficient requisite for illusory perception of haptic shape and effect of multiple contacts. In F. Barbagli, D. Prattichizzo, and Salisbury, editors, *Springer Tracts in Advanced Robotics*, vol. 18, pp. 189-198, 2005.

[5] A. Frisoli, M. Bergamasco, S. Wu, and E. Ruffaldi. Evaluation of multipoint contact interfaces in haptic perception of shapes. *Multipoint interaction with real and virtual objects*. *Springer Tracts in Advanced Robotics*, 18:177-188, 2005.

[6] A. Frisoli, M. Solazzi, F. Salsedo, and M. Bergamasco. A fingertip haptic display for improving curvature discrimination. *Presence: Teleoperators and Virtual Environments*, 17(6), pp. 550-561, December, 2008.

[7] M. Fritschi, M. Ernst, and M. Buss. Integration of kinesthetic and tactile display – a modular design concept. In *2006 EuroHaptics Conference*, 2006.

[8] G. A. Gescheider. *Psychophysics: The Fundamentals*. 3rd ed: Lawrence Erlbaum Associates, 1997.

[9] John Snyder. An Interactive Tool for Placing Curved Surfaces without Interpretation. In *Proceedings of SIGGRAPH 1995*, Los Angeles, pp. 209-218, August. 6-11, 1995.

[10] Joon-Kyung Seong, David E. Johnson, Gershon Elber, and Elaine Cohen. Critical Point Analysis Using Domain Lifting for Fast Geometry Queries. Submitted to the *Journal of Computer-Aided Design*, 2008.

[11] R. L. Klatzky and S. J. Lederman. Touch. In A. F. Healy and R. W. Proctor, editors, *Handbook of Psychology*, volume 4: Experimental Psychology, chapter 6, pp. 147-176. John Wiley and Sons, 2003.

[12] K. J. Kuchenbecker, W. R. Provancher, G. Niemeyer, and M. R. Cutkosky. Haptic display of contact location. In *Proceedings of the IEEE Haptics Symposium*, pp. 40-47, 2004.

[13] H. Levitt, Transformed up-down methods in psychoacoustics. In *Journal of the Acoustical Society of America*, vol. 49, pp. 467-477, 1971.

[14] H. B. Morganbesser and M. A. Srinivasan. Force shading for shape perception in haptic virtual environments. In *Proceedings of the 5th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, ASME/IMECE, Atlanta GA, DSC:58, 1996.

[15] W. Provancher, M. Cutkosky, K. Kuchenbecker, and G. Niemeyer. Contact location display for haptic perception of curvature and object motion. *International Journal of Robotics Research*, 24(9):691-702, 2005.

[16] D. Ruspini and O. Khatib. Haptic display for human interaction with virtual dynamic environments. *Journal of Robotic Systems*, 18(12):769-783. 2001.

[17] M. Salada, J. Colgate, P. Vishton, and E. Frankel. An experiment on tracking surface features with the sensation of slip. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2005. WHC 2005. First Joint EuroHaptics Conference and Symposium on*, pp. 132-137, 2005.

[18] R. J. Webster, T. E. Murphy, L. N. Verner, and A.M. Okamura. A novel two-dimensional tactile slip display: design, kinematics and perceptual experiments. *ACM Transactions on Applied Perception*, 2(2):150-165, 2005.