DSC-Vol. 58, Proceedings of the ASME Dynamics Systems
and Control Division
ASME 1996

FORCE SHADING FOR HAPTIC SHAPE PERCEPTION

Hugh B. Morgenbesser
Mandayam A. Srinivasan

Laboratory for Human and Machine Haptics
Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge, Massachusetts

ABSTRACT

This paper proposes a novel haptic rendering algorithm and describes a perceptual experiment that demonstrates its effectiveness in causing humans to perceive shapes during manual interactions with virtual environments. The algorithm, called "Force Shading" (analogous to Phong Shading for visual displays), refers to a controlled variation in the direction of the force vector displayed by the haptic renderer for the purpose of creating the illusion of a non-flat shape on a nominally flat surface. Experiments on shape perception were done on 5 subjects using the Phantom as the haptic interface device. In these experiments, subjects manually explored the haptic display of various virtual polyhedral approximations of a circular cylindrical bump, with or without force shading. They then indicated the perceived shape by selecting one from a menu of shapes displayed on a computer monitor. Without force shading, the subjects accurately identified the polyhedral nature of the virtual bumps. With force shading, however, the subjects identified the polyhedral approximations as feeling closer to that of a smooth cylinder. The higher the number of polygons, the higher was the percentage of trials in which the shape was identified with a smooth cylinder.

INTRODUCTION

Haptic rendering can be defined as the computation of the forces to be generated by a force-reflecting haptic interface in order to convey the tactual feel of a desired environment [Salisbury et al, 1995; Srinivasan, 1995]. For haptic interfaces with a single point of interaction, such as the Phantom [Massie, 1993; SensAble Devices, 1994], haptic rendering refers to methods for computing a single force vector from the position of the input and the state of the environment. Initial methods modeled rigid surfaces as stiff springs, and whenever possible, used simple analytic expressions to represent objects in order to simplify the computation of the surface normals and the amount of indentation, which are both required to compute the resistive force. Recently, a constraint-based God-object method [Zilles and Salisbury, 1995] has been developed to enable haptic rendering of generic polygonal meshes. This allows haptic virtual objects to take on a much more general form, rather than require that they always be composed of simple analytical building blocks. This method solved the problem of determining a reasonable point of contact to be assumed for computation of the magnitude and direction of the force. A problem of this system in its pure form is that smoothly shaped objects which are represented by polygonal meshes do not feel smooth, but feel like polyhedral objects due to the slope discontinuities at the boundaries between the flat polygons.

This is a similar problem to one which has been dealt with by visual rendering algorithms. When illumination models are applied to objects to determine the color intensities, the surface normal plays a very significant role in determining the color intensity. Visual renderers also use polygonal mesh models of smooth objects, so that without an attempt at smoothing, the objects appear to be polyhedrons. In fact, humans are so sensitive to abrupt changes in intensities of adjacent polygons that the simple solution of using a finer mesh of polygons is ineffective. In visual displays, the *Mach band effect* exaggerates the intensity change at any edge where there is a discontinuity in magnitude or slope of intensity, making the dark facet look darker and the light facet look lighter [Foley et al, 1990]. To overcome this undesirable effect, various interpolation schemes have been devised to smooth out the abrupt changes across boundaries. Gouraud shading, or color interpolation shading, first uses an interpolation scheme to estimate the surface normals at each vertex in the polygonal mesh. Then, an illumination model is applied to these points to compute intensities for each vertex, and then an interpolation scheme is used to compute the illumination information for each point on the surface [Gouraud, 1971]. Another approach, called Phong shading, first uses an interpolation scheme to generate surface normals for each visible pixel, and then the illumination model is applied to each of these points using the modified surface normals [Phong, 1975]. Phong shading is computationally more expensive, but can be used to generate more photo-realistic images.

In the haptic domain, it is known that the tactile sensory system is extremely sensitive to edges [Phillips and Johnson, 1981], and more generally to object curvatures [Srinivasan and LaMotte, 1991]. Similarly, when a shaped surface is manually explored with a probe, one can sense the edges very well, but the mechanism by which this occurs is not clear. To minimize or eliminate undesirable edge effects during haptic rendering of smoothly shaped objects approximated by polygonal meshes, the force shading algorithm was developed. Force shading builds upon the constraint-based God-object haptic rendering ideas for polygonal meshes for the determination of point of contact and magnitude of force, but uses an interpolation scheme to compute the direction of the force. The effects of force shading on human perception of shapes has been evaluated in various perceptual experiments [Morgenbesser, 1995].

THEORY

When we manually explore the shape of an object with a probe, the tactual information arises from two sources: probe position and the reaction force. The variation of both of these as functions of spatial coordinates conveys the shape of the object to the user. For example, in the case of rigid, frictionless objects, the probe traces the object shape and the reaction force direction is normal to the object surface. If such a surface approximated by polygons is explored with a probe, it feels like a polyhedral object. Our working hypothesis is that the edges are felt more due to abrupt changes in the force direction rather than in the path traced by the probe. Since only the force direction and not the magnitude gives shape information, the following possibility for simplification arises: a smoothly curved surface can be represented by polygons and the force *magnitudes* can be calculated with respect to this polyhedral approximation, but the force vector *direction* can be interpolated to vary smoothly across the surface. The implementation of this idea is the force shading algorithm described below.

The Force Shading Algorithm

Force shading refers to a controlled variation in the direction of the force vector output by the haptic renderer for the purpose of creating the illusion of non-flat shapes on nominally flat surfaces. It may be used to create the illusion of a smoothly curved shape which is actually represented for computational purposes as a mesh of polygons. For Phantom-like haptic interfaces, haptic rendering is the computation of a force vector from the input position and the state of the environment. In general, the state of the world for vector-field based renderers may include the history of the input position as well. The constraint-based God-object haptic renderer [Zilles and Salisbury, 1995] uses past values of the input position to more accurately determine the initial point of contact between the user and the virtual objects.

For the sake of simplicity, the force shading algorithm presented here is for mapping any smooth height map onto a flat surface, as is the case for a single polygon. It can be extended to non-height map objects if those objects can be represented as polygonal meshes. The height map requirements are that it be a one to one function with a portion of the plane, and that the function be continuous and piecewise-differentiable. This discussion is limited to rigid, frictionless objects.

We define the *nominal geometry* of a vector-field based haptic rendering algorithm as the geometry of the region of the input space where resistive force is nonzero. The nominal geometry gives the shape of the physical region where there is force, but does not give any information about what the forces are in that region. For example, let the nominal surface be the (x, y) plane, so that reflected forces are zero only for $z > 0$. Let $U(x, y)$ be the height map defined on the xy plane, which represents the surface of a smoothly curved object whose tactual feel needs to be approximated by shading the forces calculated with respect to the nominal geometry, which in this case is the xy-plane.

In force shading for haptic display of frictionless objects, if the haptic interface input position is at (x, y, z), then the force is chosen to be in the direction of the surface normal at $(x, y, U(x, y))$.

For the height map $U(x, y)$ in three dimensions, the surface normal vector at $(x, y, U(x, y))$ is:

$$\vec{N}(x,y) = \frac{-\frac{\delta U(x,y)}{\delta x}\hat{x} - \frac{\delta U(x,y)}{\delta y}\hat{y} + \hat{z}}{\sqrt{(-\frac{\delta U(x,y)}{\delta x})^2 + (-\frac{\delta U(x,y)}{\delta y})^2 + 1}},$$

where $\hat{x}$,$\hat{y}$, and $\hat{z}$ are unit vectors along their respective coordinate axes. Force shading requires that $\vec{N}(x,y)$ be the direction of the force reflected by the haptic interface.

The force magnitude, however, is computed according to a spring law with the plane as the object surface, not the height map The nominal geometry of this simulation, therefore, is the plane, rather than the actual shape of the height map. So, for a height map $U(x,y)$ defined on the xy plane, a force shading renderer would compute the following force for an input position at (x_1, y_1, z_1):

$$F = \begin{cases} K(-z_1)\vec{N}(x_1,y_1), & \text{if } |z_1| \leq 0 \\ 0, & \text{otherwise.} \end{cases}$$

This is a computationally easy way to add height maps to polygons in a mesh for haptic rendering, because it doesn't add much complexity over the job that the renderer would do even if it were just the polygonal mesh. This algorithm uses the same exact method for computation of the force magnitude as if the shape were indeed composed of flat surfaces, but the force direction corresponds to that of the actual surface of the object that needs to be displayed.

Application to Polygonal Mesh

Force shading may be applied to polygonal meshes in the following manner. The constraint based God-object method can be used to compute the point of contact and the magnitude of the force. But the direction of the force would be computed by an interpolation scheme which would use the surface normals from adjacent polygons as follows.

For each vertex in the polygonal mesh, an associated surface normal is computed by averaging the surface normals of all of the adjacent polygons. Then, given a point of contact on a polygon, the surface normals associated with the vertices of this polygon can be interpolated in order to determine the desired surface normal at the point of contact. Figure 1 shows a point of contact on a polygon, which is distance x_i from the vertices with associated surface normal $\vec{N}_i$. For a polygon with K vertices, the direction of the surface normal at the point of contact can be computed by:

$$\frac{\Sigma_{i=1}^{K}(x_i((\Sigma_{j=1}^{K}\vec{N}_j) - \vec{N}_i))}{\Sigma_{i=1}^{K}x_i}.$$

This method only relies on an arithmetic calculation based on position of the point of contact and the normals of the adjacent polygons, so it is not very computationally intensive. The surface normals associated with each vertex may be precomputed, so that only the last interpolation step need be done in real time.

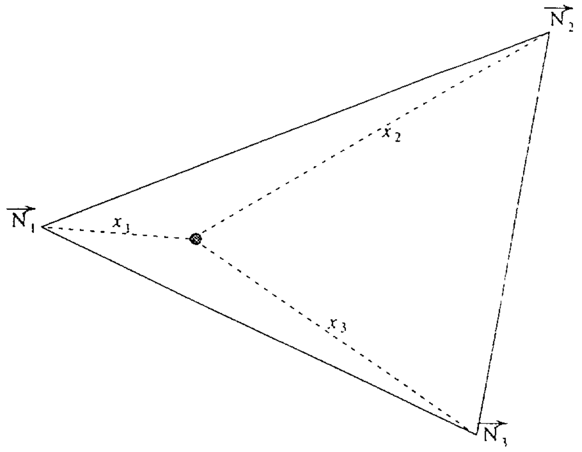

FIGURE 1: Method for applying force shading algorithm to smooth a polygonal mesh

The direction of the force may also be chosen by some other method if the desired effect is not for smoothing For example, it has been shown that perception of surface texture can been generated using horizontal forces [Minsky, 1995]. It is even possible to add perceivable textures by adding horizontal forces to a force shaded polygonal mesh to simulate textured objects that vary smoothly over a coarse length scale.

EXPERIMENT

An experiment was done in order to study the effects of force shading on human perception of haptically rendered shapes. Four different nominal geometries were chosen for use as the shapes in this experiment: a circular cylinder, and three polyhedral approximations to the circular cylinder, as shown in Figure 2. Since each of the three polyhedral geometries was rendered in both their force shaded and unshaded forms, a total of seven different virtual objects were included as stimuli.

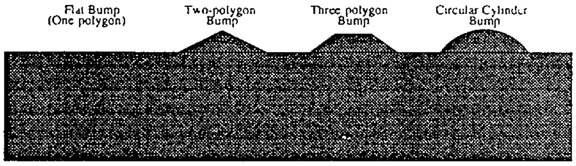

FIGURE 2: Cross sections of the four different nominal geometries used in the experiments.

The Cylindrical Bump

The cylindrical bumps were displayed using vector-field based methods and the analytical forms of a cylinder and a plane. The forces for these two objects were computed independently and then superposed. For the plane, the force was proportional to the extent of the indentation and normal to the plane. For the cylinder, the force was also proportional to the extent of the indentation, but the force was radial, away from the center.

The Polygonal Approximations

These bump display algorithms were based on a height map approach. For these, the reflected force was proportional to the extent of the indentation into the height map, and was in the direction of the surface normal of the polygon being contacted. For the force shaded bumps, the force magnitude was chosen in the same manner, but the direction was chosen to mimic the force direction for the cylindrical bump.

The Classification Experiment

The classification experiment was designed to explicitly compare the haptic feel of various shaded bumps with their unshaded counterparts. In this experiment, subjects were asked to feel a single bump and then classify it as one of four which they saw on a monitor. The four pictures on the screen are shown in Figure 2: a flat plane (equivalently, a one-polygon bump), a two-polygon bump, a three-polygon bump, and a cylindrical bump. This experiment had two different parameters to vary: the bump display algorithm, and the size of the bump. The experimental conditions are enumerated in Table 1.

The Bump Display Algorithms. All seven of the bump display algorithms listed above were used. They were presented an equal number of times but in a random order.

The Bump Sizes. The bumps were all 1000 mils (one inch) wide. Three different values for the radius of the bumps were chosen (600, 1000, and 1400 mils). This was to explore possible differences in the effectiveness of the force shading algorithm for different radii of the bumps.

The Experimental Subjects. Five MIT undergraduates participated as subjects in these experiments. All of them were right handed, and used their right hand to probe the virtual objects. During the explorations, a black fabric screen was used to prevent them from seeing their hand. After they explored each stimulus example, they entered in their choice for the shape they thought they had felt by typing either 1,2,3, or 4 on a computer keyboard. They were given no feedback indicating whether or not they were correct. At the end of the experiment, the subjects were paid for their time.

Bump Type	Subject	Reference Radius (mils)	Repeated Trials
• Cylindrical • One-Poly unshaded • One-Poly shaded • Two-Poly unshaded • Two-Poly shaded • Three-Poly unshaded • Three-Poly shaded	• 1 • 2 • 3 • 4 • 5	• 600 • 1000 • 1400	• 24

TABLE 1: Classification Experiment Conditions

A total of 2520 trials were done, using 5 subjects. These trials took only about 10 seconds each, so that this experiment took about 7 hours to complete. In addition, in other independent trials, the paths traced by the tip of the Phantom's probe when subjects explored the bumps were recorded. These independent trials were done immediately following the main data collection, and subjects were not informed of any differences between the sets. The instructions to the subjects were the same.

RESULTS

This experiment provided data on what shape the subjects thought they felt as they explored any one of the seven virtual object shapes. The results are given in Table 2. For each type of bump there were three possible radii. The numbers in each column represent the percentage of responses for which the subjects characterized the bump as the shape indicated at the top of the column. For each row in the table the highest percentage response is shown in boldface. For the unshaded bumps, subjects were fairly good at correctly assessing the shape of the bump, although there were a number of errors, especially with the two polygon unshaded bump. With the addition of shading to the polyhedral bumps, however, the cylindrical bump was chosen more often than any other bump.

DISCUSSION

This experiment demonstrated several important points about the perceptual effects of the addition of force shading to polyhedral meshes.

Force Shading enhanced the subjects' perception that the object was a smooth cylindrical bump. The classification experiment directly compared force shaded polygonal approximations of the circular cylinder with unshaded polygonal approximations. The five subjects classified the unshaded polygonal approximations appropriately as the actual geometry more often than any other geometry. When shading was added to the polygonal approximations, all the subjects classified the bumps as smooth and cylindrical in character more often than other shapes.

With force shading, as the number of polygons used to approximate the bump increased, human perception approached that of the real bump. This is shown by the higher percentage of "cylindrical bump" responses for the two and three polygon shaded bump stimuli than for the one polygon shaded bump stimuli. Subjects were more likely to think that they were feeling a cylindrical bump if the stimulus had nominal geometry closer to the cylindrical bump.

The experiment suggests that the subjects tend to perceive a shape more by feeling curvature via force direction changes than hand position changes. Even when presented with the force-shaded one-polygon bump whose nominal geometry had no curvature, the subjects perceived some curvature. The position paths that were recorded in separate trials showed that during explorations of force shaded bumps, the subjects actually traced out shapes similar to the nominal geometry. However, the subjects identified these shapes as smooth and curved in the majority of trials based on force direction changes.

Force shading seems to be more effective for lower curvatures than for higher curvatures. For the two and three polygon force shaded bumps, the subjects chose the cylindrical bump more often for the larger radii, which corresponded with smaller curvatures. Since the bumps were designed to be of constant width, the larger the radius of a bump, the smaller the height.

BumpType	Bump Radius (mils)	Flat	Two Poly	Three Poly	Cylinder
Flat Unshaded	600	**99**	1	0	0
	1000	**96**	1	1	3
	1400	**100**	0	0	0
Two-Poly Unshaded	600	0	**79**	3	19
	1000	1	**53**	3	43
	1400	8	**47**	1	44
Three-Poly Unshaded	600	0	3	**92**	6
	1000	0	3	**86**	10
	1400	0	7	**87**	6
Real	600	0	7	19	**75**
	1000	0	9	3	**88**
	1400	0	6	0	**94**
Flat Shaded	600	2	3	38	**57**
	1000	3	11	29	**57**
	1400	2	14	36	**48**
Two-Poly Shaded	600	0	35	16	**48**
	1000	0	12	2	**87**
	1400	1	6	8	**85**
Three-Poly Shaded	600	1	3	39	**57**
	1000	1	3	15	**81**
	1400	0	7	8	**86**

TABLE 2: Summary of classification results for five Subjects. The number in each cell represents the total number of times the subjects chose the shape indicated at the top of the columns, expressed as a percentage of the total number of trials for each row.

ACKNOWLEDGEMENT

This work was supported by ONR grant N61339-94-C-0087.

REFERENCES

[1] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice.* Addison Wesley, 1990.

[2] H. Gouraud. Continuous shading of curved surfaces. *IEEE Transactions on Computers*, C-20(6):623–629, June 1971.

[3] J. R. Phillips and K. O. Johnson. Tactile spatial resolution - II. Neural representation of bars, edges and gratings in monkey afferents. *Journal of Neurophysiology*, 46(6):1192–1203, December 1981.

[4] T. H. Massie. *Design of a Three Degree of Freedom Force-Reflecting Haptic Interface.* SB thesis, Massachusetts Institute of Technology, May 1993.

[5] M. D. R. Minsky. *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display.* PhD thesis, Massachusetts Institute of Technology, June 1995.

[6] H. B. Morgenbesser. *Force Shading for Shape Perception in Haptic Virtual Environments.* M.Eng. thesis, Massachusetts Institute of Technology, September 1995.

[7] B. T. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.

[8] J. K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: Programming touch interaction with virtual objects. In *Proceedings of the 1995 ACM Symposium on Interactive 3D Graphics*, April 1995.

[9] SensAble Devices, Incorporated, Cambridge, Massachusetts. *Phantom Documentation and Example Code*, 1994.

[10] M. A. Srinivasan. Haptic interfaces. In *Virtual Reality: Scientific and Technological Challenges*, N. I. Durlach and A. S. Mavor, editors, chapter 4, pages 161–187. National Academy Press, 1995.

[11] M. A. Srinivasan and R. H. LaMotte. Encoding of shape in the responses of cutaneous mechanoreceptors. In *Information Processing in the Somatosensory system*, O. Franzen and J. Westman, editors, Wenner-Gren Intl. Symposium series, pages 59–69, Macmillan Press, 1991.

[12] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, August 1995.