Haptic B-spline Surface Sculpting with a Shaped Tool of Implicit Surface

Zhan Gao[1] and Ian Gibson[2]

[1]University of Hong Kong, gaozhan@hkusua.hku.hk
[2]University of Hong Kong, igibson@hku.hk

ABSTRACT

In this paper, we propose a new implicit to B-spline surface haptic interface and present a new haptic sculpting system for B-spline surfaces with a shaped implicit surface probe/tool. In the physical world, people touch or sculpt with their fingers or tools, instead of just manipulating points. A shaped virtual haptic probe/tool helps users to relate the virtual deformation process to their real life experience. Using the PHANToM haptic device, modellers can touch and explore the model with a shaped probe/tool and feel the physically realistic presence of virtual objects. Various basic haptic sculpting tools are developed to facilitate the deformation of B-spline surfaces. The tool-model interaction is surface-object in nature instead of traditional point-object style. This leads to more intuitive virtual sculpting paradigms, which may help introduce new modeling techniques.

Keywords: Geometric design; Haptic sculpting; Haptic rendering; Physics-based modelling

1. INTRODUCTION

In CAD/CAM systems, B-spline has become the de facto standard for surface representation. Much work has been done to facilitate design using B-spline surfaces. Haptic sculpturing is a modelling technique based on the notion of sculpting a solid material while providing haptic feedback at the same time. The sense of touch, in combination with our kinaesthetic sense, adds a new modality to virtual sculpture, especially in presenting complex geometry and material properties.

In this paper, we propose a virtual haptic sculpting system which features a new surface-object haptic rendering algorithm with a shaped probe/tool and a B-spline surface sculpting technique using this shaped probe/tool. We put emphasis on the role of the shape and size of sculpting tool because the final shape of the model not only depends on the material property of the model, the moving path of the tool, but also on the shape and size of the tool. This can be compared with the commercial FreeForm haptic modelling system, which provides arrays of modelling tools of different shapes and sizes to help modellers relate the virtual sculpting with their experiences of sculpting in the physical world (Fig. 1.) [1]. The shapes of tools in FreeForm software also serve as visual hints that correlate the desired deformation on models with the shape of selected tool. In our system, we develop shaped probe/tools with that purpose as well so that modellers can anticipate a certain result before action.

Every virtual haptic sculpting system includes two main components. One is a haptic rendering subsystem; the other is a geometric engine. A haptic rendering subsystem deals with collision detection and force feedback generation. To maintain a smooth and stable haptic feedback, haptic rendering generally requires 1 kHz updating rate, which is much higher than graphic rendering update rate. Most virtual sculpting systems for B-spline surface are based on point-object interaction, i.e. the 3D cursor or the sculpting tool is a point without any spatial volume or shape. For example, in [4], the haptic interaction and sculpting technique are of point-object style. The prerequisite for a shaped probe/tool model interaction is an efficient and computationally inexpensive surface-object haptic rendering algorithm. Computational inexpensiveness is very important because the physics-based deformation of a model requires extensive computational resources. We therefore must optimize the haptic rendering's resource as much as possible. Most currently available surface-object haptic rendering methods are too expensive to be incorporated into a physics-based B-spline surface modeling system. Therefore, we developed an implicit to B-spline surface haptic rendering algorithm with that purpose. In this method, a shaped probe/tool is represented as an implicit surface, typically an ellipsoid or a sphere. Users hold the stylus of the haptic device to drive the probe/tool to touch the B-spline surface model while the system detects collisions between the implicit surface of the probe/tool and the B-spline surface to determine the output force feedback. A new physics-based spline surface sculpting method has been developed by taking advantage of the new tool-model haptic interaction technique. A B-spline surface, which is constrained to a mass-spring mesh, deforms dynamically according to the interaction between the probe/tool and the mass-spring mesh.

The remainder of the paper is organized as follows. Section 2 reviews research concerning haptic rendering techniques and haptic sculpting. Section 3 presents the new surface-object haptic rendering method. Section 4 describes the detailed components of our sculpting system and implementation issues. Section 5 concludes the paper and outlines future research directions.

2. RELATED WORK

2.1 Haptic Rendering Algorithm

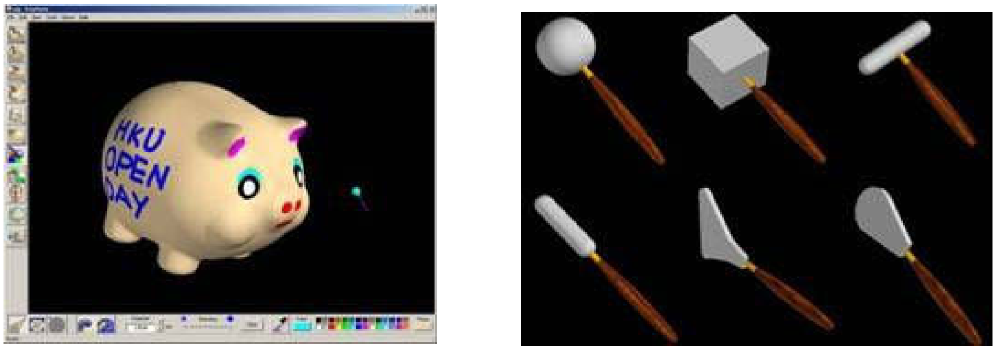
Haptic rendering is the process of applying forces in order to give the operator a sense of touch and interaction with physical objects [12]. Earlier researches mainly dealt with haptically rendering objects with a point interaction paradigm. A point-object based haptic interface generates only 3 DOFs of force feedback because a point cursor, or a Haptic Interface Point (HIP), cannot simulate torques. To provide 6 DOFs force feedback, the haptic cursor must have a shape and a size. A ray-object haptic interface uses a ray as the haptic cursor to provide both torques and forces [6]. The surface-object haptic interface uses surfaces for geometric representation of the haptic cursor. This method can introduce a much more complex haptic cursor into the haptic simulation, thus improving the degree of realism.

Gregory et al presented an algorithm for haptic display of moderately complex polygonal models with a polygonal haptic cursor by making use of incremental algorithms for contact determination between convex primitives [5]. McNeely et al put forward a simple, fast, and approximate voxel-based approach. This approach enables the manipulation of a modestly complex haptic cursor within an arbitrarily complex environment of static rigid objects [8]. Nelson derived a novel velocity formulation for use in a parametric surface-surface tracing paradigm and integrated it into a three step tracking process to compute reaction force between two NURBS surfaces [9].

Because implicit surface is convenient for collision detection, some researchers have managed to present point-surface haptic rendering techniques with implicit surface. Kim introduced a haptic algorithm which is mainly based on an implicit surface representation which represents the surface with potential values in a 3D regular grid [7]. Kim's method also allows the user to feel a smooth surface without force discontinuity.

2.2 Haptic Sculpting

Dachille et al put forward a novel haptic interface and present a physics-based geometric modelling approach that facilitates interactive sculpting of spline-based virtual material [4]. Using the PHANToM haptic device, modellers can feel the physically realistic presence of virtual spline objects and interactively deform virtual materials with force feedback throughout the design process [2].



(a) (b)

Fig. 1. FreeForm and different types of sculpting tools

FreeForm haptic modelling system, from SensAble Ltd, is the first commercially successful Computer Aided Industrial Design (CAID) tool which lets designers sculpt and form virtual clay using similar tools and techniques that are employed in the physical world [10]. To simulate a feeling of a physical tool of real life, FreeForm provides a set of virtual tools which look, function and behave similar to their counterparts in a physical world.

3. HAPTIC RENDERING OF SHAPED PROBE OF IMPLICIT SURFACES

For virtual sculpting, the requirement of complexity of probe/tool shape is not as high as for other virtual reality applications such as assembly simulation, where shape complexity is equally necessary for both probe/tool and other contacting models. Since nearly all tools in real life have quite simple shapes, computational simplicity, rather than the complexity of geometric shape, is the priority for haptic rendering of our shaped probe/tool for virtual sculpting. Using B-spline surface or polygonal meshes to represent a probe/tool is too expensive for haptic sculpting tasks. We chose implicit surface as the representation for probe/tool mainly because of its easiness for collision detection and the simplicity of shape the implicit surface can offer.

Here probe and tool is the same object. If the model to touch is a static one, we call the virtual cursor a probe; if the model deforms due to the motion of virtual cursor, we call it a tool. In this section we describe our implicit-based haptic rendering with a relatively simply shaped probe/tool.

3.1 Implicit Surface Representation

The implicit representation of the surface S is described by the following implicit equation [3]

$$S = \{(x, y, z) \in R3 \mid f(x, y, z) = 0\} \tag{3.1}$$

where f is the implicit function, and (x, y, z) is the coordinate of a point in 3D space.

Here *f(x, y, z)* could be polynomials, discreet grids of points or some black box functions. When *f(x, y, z)* is polynomial, it yields an implicit algebraic surface. When *f(x, y, z)* is linear, it describes a plane. When *f(x, y, z)* is quadratic, it describes a quadric surface, such as an ellipsoid, a sphere or a cylinder.

If the potential value of *f(x, y, z)* is 0, then the point (*x, y, z*) is on the surface. The set of points for which the potential value is 0 defines the implicit surface. If the potential value is positive, then the point (*x, y, z*) is outside the surface. If *f(x, y, z)* < 0, then the point (*x, y, z*) is inside. We use a closed quadric surface to represent our probe; more specifically, an ellipsoid. We choose an ellipsoid surface as our probe/tool because it is mathematically simple enough to evaluate thousands of points for collision detection in 1 kHz frequency.

3.2 Haptic Model

In this section we give a more detailed presentation of our haptic model including collision detection and force generation.

3.2.1 Collision Detection

The implicit surface of the probe, driven by the PHANToM device, moves as the user manipulates the PHANToM stylus. The definition function of the probe is transformed by the PHANToM transformation matrix at every time step of haptic servo loop. B-spline surfaces are discretized into a layer of uniformly distributed sampling points. Now, the inside/outside property of the implicit function makes the collision detection between the discreet sampling points on model surfaces and the implicit surface of probe trivial. Each point's coordinates are input into the implicit function to evaluate, and then we know if the point is inside or outside the probe by judging the sign of the potential value of *f(x, y, z)*. If a point is inside, a collision is detected.

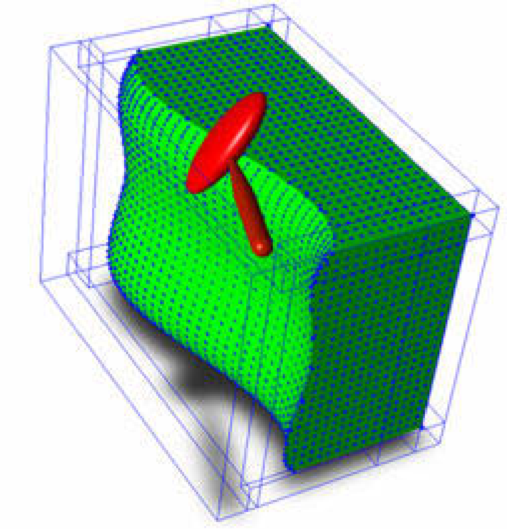


Fig. 2. Haptic rendering with an ellipsoid shaped probe

Fig. 2. shows a model being touched by a red ellipsoid probe. The surfaces of the model are green and the blue dots on surfaces are the surface sampling points. Note every surface is bounded by a boundary box which largely improves collision detection efficiency.

3.2.2 Force Magnitude

According to classical mechanics, the probe should be viewed as a rigid body instead of one single mass point, therefore forces applied on the surface of the probe not only forms a force vector but also a torque. However, the PHANToM device of our setup provides only 3 DOFs force feedback and no torque output. To simplify the problem, we only consider force at this stage. However, note that our probe is a rigid body in nature, so it would be quite easy to adapt the algorithm to 6 DOFs haptic rendering.

Penalty and constraint haptic rendering methods determine force magnitude by Hooke's law: $F = k \cdot s$, where *k* is stiffness of a spring and *s* is the displacement of mass point connecting to the spring [12]. To make the system more stable, a damping force is added, hence: $F = k \cdot s - d \cdot \dot{s}$, where d is the damping factor and $\dot{s}$ is the velocity of the point. In our method we need not just one spring, but an array of distributed damping springs. Fig. 3. shows a probe penetrating a planar surface of a model. Red dots denote the surface sampling points inside the probe and black ones are sampling points outside. The depths of colliding sampling points are the distances between those sampling points to the correspondingly closest points on the implicit surface.

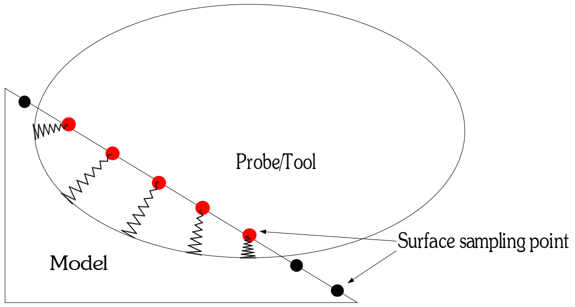


Fig. 3. Distributed springs of probe.

Note the potential value from *f(x, y, z)* is not Euclidean distance from a point to the implicit surface of probe. Although Lagrange multiplier method can be used to find the minimum distance to the algebraic surface, it requires several iterations to find the numerical roots for the equation. We find the approximate answer by shooting a normal vector from the sampling point P_0 and finding out the intersection between the vector and the implicit function of probe. The normal vector of point $P_0(X_0, Y_0, Z_0)$ can be found as $N_0(N_x, N_y, N_z)$. Since sampling point is inside the implicit surface instead of on the surface, the normal vector N_0 can be viewed as a normal vector of a smaller surface offset from the original one. If the implicit function is quadratic, in this case an ellipsoid, an analytic solution for the intersection points can be found easily. The distance between P_0 and the closer point is the depth.

Finally, we define the magnitude of force as

$$\left|F\right| = \sum_{i=0}^{n} (k \cdot s_i - d \cdot \dot{s}_i) \tag{3.2}$$

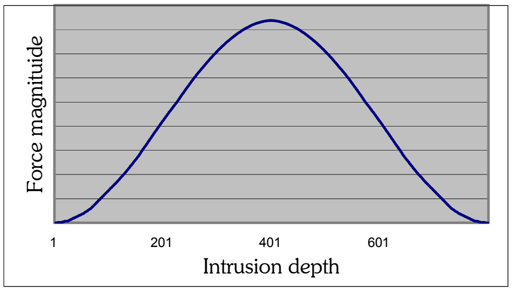


Fig. 4. Intrusion depth and force magnitude

Fig. 4 shows what happens when a spherical probe, with a diameter of 800, passes through a plane with continuous force magnitude. The output force reaches the maximum when the intrusion depth is 400. At that moment the center of probe touches the plane so that the probe has the largest contact area with a surface. After the peak the force magnitude decreases.

3.2.3 Force Direction

Penalty-based approaches and constraint-based approaches generally compute force vectors according to the normal vector of contacted surface [12]. In the following section, we put forward two schemes to compute direction of force and discuss the pros and cons of these two approaches.

One straightforward approach is to evaluate the surface to obtain normal vectors and assign them to surface sampling points during the pre-processing stage. We call this the *surface normal method*. In this case, every sampling point not only has positional information but also has a local normal vector. After collision detection, normal vectors of all contact surface sampling points, weighted with the sampling points' depth into the implicit surface, are summed up as the force direction.

The second approach is the *probe normal method*, where force vector is not derived from the surface of the model but from the implicit surface of the probe instead. If there is no friction between the probe and model surface, the repressing force and reaction force are normal to the surface of contact area. Therefore, if we can obtain the approximate normal vector of contact area on the implicit surface of the probe, we can use the sum of those normal vectors to determine the reaction force direction.

Four examples in two typical situations are used to show the difference of the two approaches. Fig. 5. (a) and (b) shows the surface normal method while Fig. 5. (c) and (d) for the probe normal method. Fig. 5. (a) and (c) show the condition when the probe passes over a sharp edge. Fig. 5. (b) and (d) show how the probe passes through the contact surface.

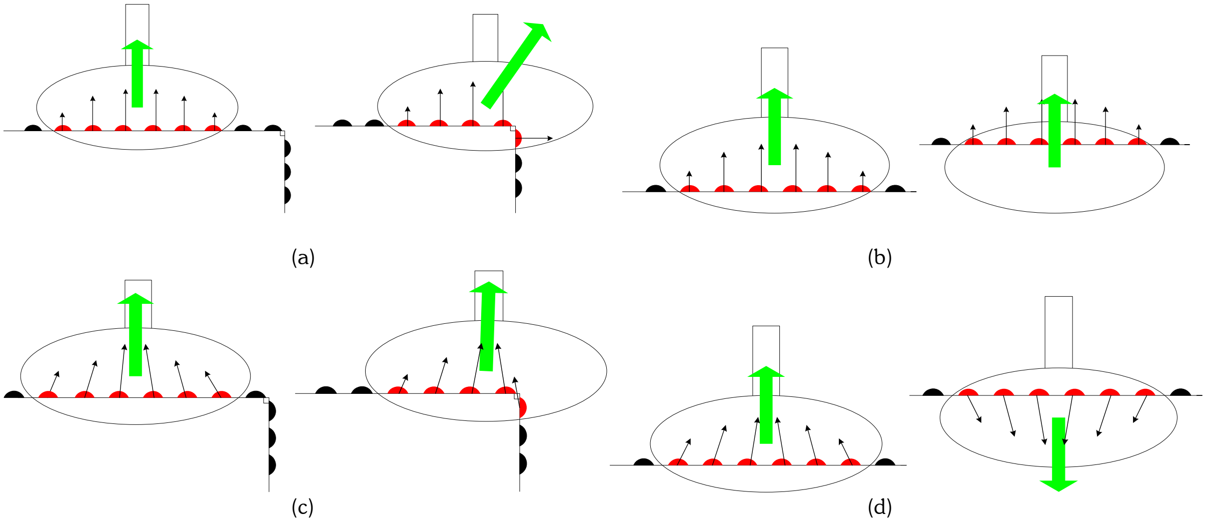


Fig. 5. Haptic rendering. The ellipse stands for the probe/tool, black dots are points not in contact with the probe while red dots are surface sampling points in collision. Green arrows show the direction of force and small black arrows show the normal vectors of surface sampling points.

The surface normal strategy seems straightforward and precise because the normal vectors are directly from the surface being touched. This method demonstrated a good performance when a model surface is continuous. In Fig. 5. (a) and (b), the left side figures represent one of the ideal situations of the surface normal haptic rendering method because all contact sampling points have the same normal vector. The sum force from the distributed spring system is always perpendicular to the planar surface. User feels the surface as a smooth frictionless plane. However, if the model has some sharp edges (Fig. 5. (a)), the surface normal method could generate force discontinuities. In Fig. 5. (a), when the probe is moving to the right so that the edge is inside the probe, the direction of force changes suddenly because the normal vectors of sampling points on the side surface is to the right direction. Therefore a user feels a sudden impulse when the probe approaches to the edge and a force discontinuity occurs. Fig. 5. (c) shows how the probe normal method prevents the force discontinuity.

In Fig. 5. (b), when the probe passes through the surface, the force direction of the surface normal method is consistent because the direction of force only depends on the direction of surface normal. In Fig. 5. (d), the probe normal method generates a force discontinuity because the force direction suddenly reverses to the other side when the centre of probe passes through the surface.

Both methods have their advantages and limitations. Since shapes of models in our haptic sculpting system are usually freeform style and smooth, the surface normal method is the prior choice. However, if loops are too tight to use the B-spline surface to generate normal vectors, the probe normal strategy would be utilized to find the local normal in a more economic manner. Furthermore, the probe is allowed to pass through the surface in our system. This is different from haptic applications such as assembly simulation, where model intrusion is generally forbidden. A haptic sculpting system requires the probe/tool not only to push the B-spline surface from outside but also to pull the surface from within. So, the probe must be allowed to pass through the wall of a model. Surface normal method certainly has better performance in this case because no force discontinuity occurs, not only in terms of force magnitude but in terms of direction.

We therefore formulate the direction of force as:

$$\vec{F} = \frac{\sum_{i=0}^{n} \vec{N}_i \cdot d_i}{n} \tag{3.3}$$

where n is the number of surface sampling points inside probe, N_i as the normal vector of surface sampling point, which could be generated by either of the two approaches, and d_i as the depth. The direction of feedback force is an average of the normal vectors weighted by the depth. This makes sense because the point having bigger depth plays a bigger role in determining the direction of force.

4. HAPTIC SCULPTING

Sculpting can be viewed as a dynamic interaction between a model and a tool. In our system, the tool is the implicit surface. The final shape of the surface not only depends on the shape of the tool, but also on the stiffness of the model, the moving path of the tool and so on. We adopt Dachille's dual representation of mass spring system (MSS) and B-spline surface as our model representation [4]. The sculpting system allows the user to touch the surface with a probe/tool and click the stylus button to deform or carve the surface.

4.1 Combine MSS With B-spline Surface

The dual representation of Dachille is a combination of Mass Spring System (MSS) and B-spline surface [4]. The term Mass Spring System means a system formed by a set of mass points and a set of spring constraints between adjacent couples of points; each point is subjected to forces due to the status of the springs connected to it and to the potential external forces. The position, velocity, and acceleration of each mass point are governed by standard Newtonian mechanics, using the following basic equation:

$$M \frac{d^2}{dt^2} \vec{X}(t) + C \frac{d}{dt} \vec{X}(t) + K\vec{X}(t) = \vec{f} \tag{4.1}$$

where *M* is the mass matrix for the mesh, *C* is the damping matrix, *K* is the overall representation of the stiffness of the mesh, $\vec{X}(t)$ is the vector of current positions of the nodes, and a *f* is the vector of applied external forces, such as gravity and pushing forces from the user, acting on the mass points. Under a specific external force, the MSS mesh deforms according to the above mechanics equation.

A continuous B-spline surface *S(u,v)* can be represented as the combination of a set of basis functions $N_{i,p}$ and $N_{j,q}$ with $(n+1)\times(m+1)$ control points.

$$S(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} N_{i,p}(u) N_{j,q}(v) P_{i,j} \tag{4.2}$$

Note that, $N_{i,p}$ and $N_{j,q}$ are piecewise polynomials of order *p* and *q*, respectively. Both *u* and *v* are parametric variables. Their parametric domain is determined by two sets of nondecreasing knot sequences, respectively.

Dachille and Qin's approach can assign the masses directly on the B-spline surface and hides the control points from the user [4]. The basic idea of their method is to set up a mapping transformation between control points of B-spline surface and the mass point system.

Assume we have an $n \times n$ MSS mesh and we require a corresponding B-spline surface which has the same number of control points. So, the control point matrix, P, must be a square matrix as well.

$$P = \begin{bmatrix} p_{0,0} & \cdots & p_{0,n} \\ \vdots & \ddots & \vdots \\ p_{n,0} & \cdots & p_{n,n} \end{bmatrix} \tag{4.3}$$

$$M = \begin{bmatrix} m_{0,0} & \cdots & m_{0,n} \\ \vdots & \ddots & \vdots \\ m_{n,0} & \cdots & m_{n,n} \end{bmatrix} \tag{4.4}$$

According to [4], a transformation matrix A is created. Entries of matrix A are basis functions evaluated at various parametric values. A is uniquely determined by the parametric values of the B-spline surfaces. Therefore matrix A is constant because all parametric values of the discretization are constant. Thus, matrix A and its inverse can be pre-computed.

$$M = A \cdot P \cdot A^T \tag{4.5}$$

where T means matrix transposition. Similarly, we have $P = A^{-1} \cdot M \cdot A^{-T}$.

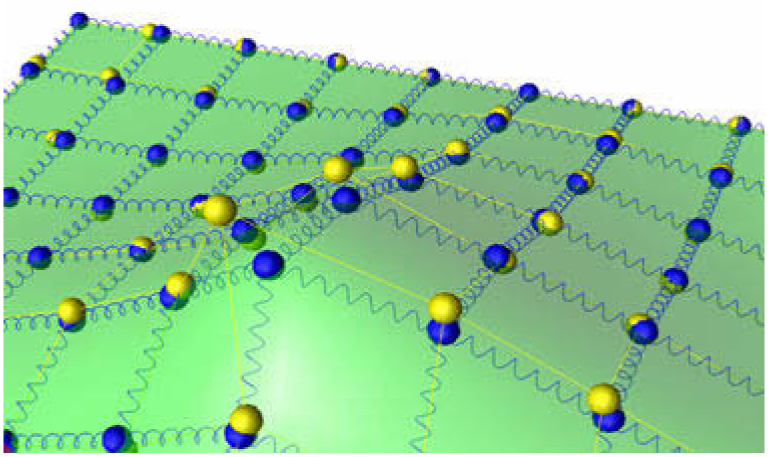


Fig. 6. Dual representation of B-spline surface and MSS. The B-spline surface is colored as transparent green. Control points of the B-spline surface are yellow balls. The blue mass points can be seen connected by springs and constrained upon the B-spline surface.

Now the B-spline surface is discretized into a set of parametrically uniform square MSS mesh, and therefore the B-spline surface and Mass Spring System are combined together.

4.2 Sculpting System and Implementation

When the user sculpts the surface with a tool, the surface reacts in a physics-based manner and deforms according to the manipulation of the user. The system runs at 1 kHz updating rate. This is mainly because of the principle of haptic rendering. Euler integration for MSS deformation also requires very small time steps to maintain stability. At each time step, the system samples the 6 DOF position of the PHANToM device and updates the position of the tool. Forces due to interaction between tool and surface are also computed in every time step. When haptic stylus clicks are detected, sculpting forces are applied to the MSS. Then, the mass points of MSS evolve to new positions according to Eqn 4.1. To visually display the deformation of a B-spline surface under sculpting, the control points of the B-spline surface must be updated from the transformation of MSS with Eqn 4.5.

The following details the major components of the current haptic sculpting system, including tool-model interaction, deformation and numerical integration, haptic rendering for deformation, and sculpting tool implementation.

4.2.1 Deformation and Numerical Integration

The position, velocity, and acceleration of each mass point in MSS are governed by standard Newtonian mechanics, using the Eqn. 4.5 [11]. The force *f* in the above equation is the sum of all external forces whilst the internal force is generated by the constraint springs connecting that mass point. During the haptic rendering loop, the implicit function of the tool evaluates the mass points to detect collision. The force applied to the mass point is proportional to the depth of mass point, which can be found in a similar way in 3.2.2.

To find out the new position of mass points, we just need to solve the ordinary differential equation with numerical method. The most straightforward approach is Euler integration. At each time-step of the integration, the velocity $\Delta\vec{X}(t)/\Delta t$ of the mass point is updated according to the applied forces and to the material quantities such as mass, damping, and stiffness. The mass points are moved to a new position $\vec{X}(t_{i+1})$

$$\frac{d}{dt}\vec{X}(t_{i+1}) = \frac{d}{dt}\vec{X}(t_i) + \frac{d^2}{dt^2}\vec{X}(t_i)\cdot\Delta t \tag{4.6}$$

$$\vec{X}(t_{i+1}) = \vec{X}(t_i) + \frac{d}{dt}\vec{X}(t_i)\cdot\Delta t \tag{4.7}$$

Since Euler integration for MSS deformation requires very small time steps to maintain stability, it can run at 1 kHz. By Eqn. 4.5 $P = A^{-1}\cdot M\cdot A^{-T}$, the new positions of MSS are transformed to new positions of control points of the B-spline surface.

4.2.2 Haptic Rendering For Deformation

During a sculpting session, forces are applied to the MSS to generate desired deformation. Newton's third law must be held throughout the sculpting: any force that the user applies to the MSS must be reacted back to the user. If the MSS grid is sparse, e.g. 10×10, there would be no problem if we update the corresponding B-spline surface at 1 kHz and we can generate surface sampling points from the updated B-spline surface for collision detection and force computation.

However, if the MSS grid is quite dense, e.g. 30×30, to update the B-spline surface control points at 1 kHz is computationally expensive. Therefore, the system cannot depend on the B-spline surface to provide surface sampling points. So, we directly use the mass points of MSS mesh as surface sampling points for haptic rendering. If MSS is not dense enough to generate smooth haptic feedback, we can increase the density of surface sampling points by linear interpolation between MSS mass points. In that way, the force feedback from MSS mesh is satisfactory. To save computation resource, we update the B-spline surface control points in another loop for visual display updating, which runs at 50 Hz updating rate. The detachment between the physical model updating and the visual display updating proves to be efficient and effective.

4.2.3 Tool-Model Interaction

Currently, the shape of probe/tool is ellipsoid or spherical. More complex implicit surface for probe shapes will be implemented in the future.

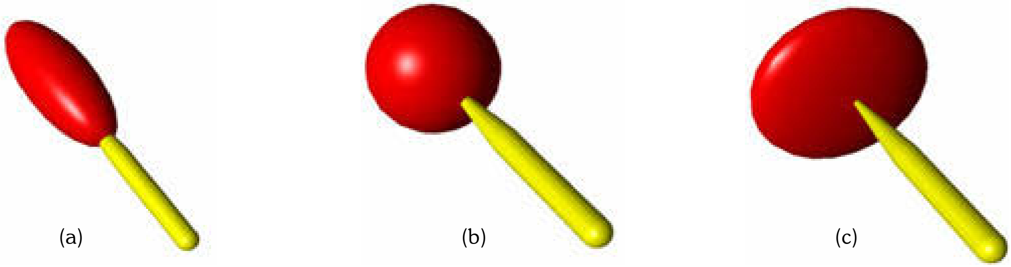


Fig. 7. Different shapes of probe/tool

Our system allows users to freely touch any position on the surface. Since the probe/tool has shape and size, it can touch a certain portion of surface, instead of just a point on the surface as with earlier point-surface haptic rendering systems.

Our system also permits users to adjust the size and shape of the ellipsoid probe/tool. By changing the size of probe/tool, users can determine themselves how big a portion they want to contact with the surface. By changing the shape of the tool, users can have different sculpting effects accordingly. For example, a tool like Fig. 7. (c) can unwrinkle and smooth the surface and the tool in Fig. 7. (a) can easily cut deep into the surface.

4.2.4 Implementing Sculpting Tools

In Dachille's system [4] users deform the surface by grabbing the surface through a rope tool. In our sculpting system, the way users sculpt the surface is more like FreeForm [10]. To sculpt, users have to touch the surface in the first place, and then click the PHANToM stylus button to push or pull the surface to form the desired shape. Our surface cannot be pulled from outside. The probe can intrude into a model through the B-spline surface so that users can pull the surface from behind or inside to form a bulge. Modelers move tools back and forth upon the surface to sculpt it. Besides the moving path of tool, the shape and size of tool also play roles in forming the final shape of surface. In Fig. 7. (b) and (c), the deformations on surfaces copy the shape of sculpting tool.

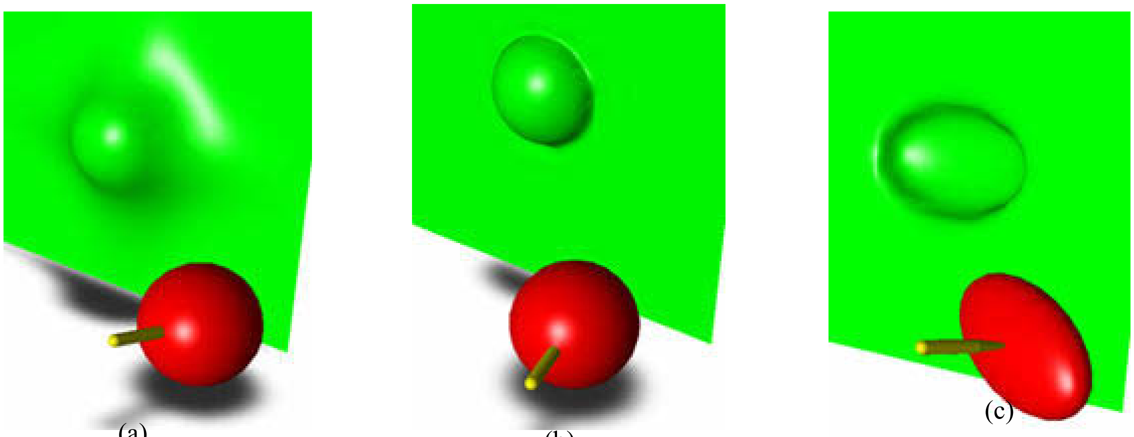

(a) (b) (c)

Fig. 8. Behaviors of sculpting under different conditions

The final shape of the surface not only depends on the shape of the tool, but also on the stiffness of the model. Changing the overall mass of the surface not only changes the extent of deformation, allowing the surface to react either wildly or mildly to sculpting forces, but also changes the stiffness and material behaviour of B-spline surface. In Fig. 8. (a) and Fig. 8. (b), two B-spline surfaces, both at 30×30 grids, but with different overall mass, are sculpted with the same tool. They demonstrated different behaviour during deformation. Fig. 8. (a) shows a lighter MSS having a bigger deformation extent and generating a feeling of softer material. In Fig. 8. (b), the mass of that MSS is 100 times heavier than the one in Fig. 8. (a). The result shows that the deformation is much more local and a sharp edge along the bulge appears.

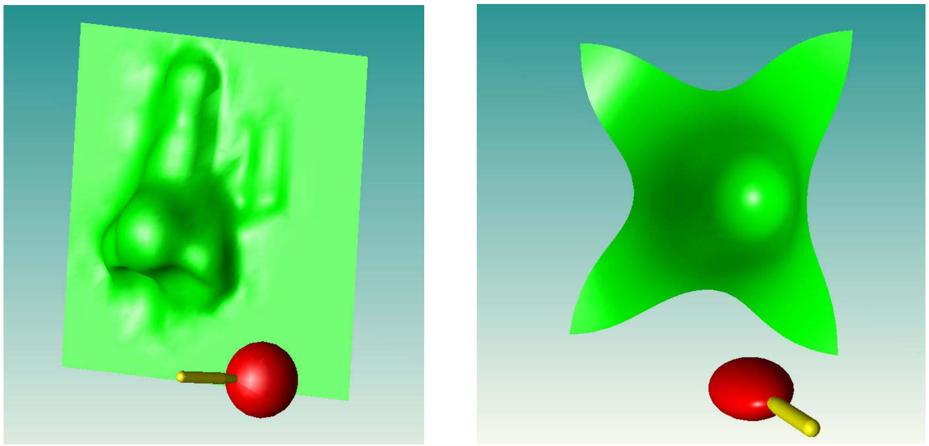


Fig. 9. Samples of a human nose and a flower.

Fig. 9. shows two more complex models. In particular, one should note the organic form of the model, which is the intended effect, as well as the continuity of the model, which is a direct consequence of using B-spline surface representation.

5. CONCLUSION

We have presented a new implicit surface to B-spline surface haptic rendering method and a new haptic sculpting system that facilitates the direct manipulation of B-spline surfaces with an implicit surface probe/tool. The haptic rendering method is efficient and computationally inexpensive. The haptic interface and sculpting method with a shaped probe/tool are intuitive and natural. Our system offers users a set of basic interaction toolkits, supporting sculpting with ellipsoid or spherical tool via haptic feedback devices, allowing the modeler to change the shape and size of tool dynamically and to modify the material property of model surface.

The reported work lays a concrete foundation for a more sophisticated haptic modeling system to be established. As can be seen in this paper, sculpting and deformation are constrained by a single surface patch. Our next research agenda is to handle multiple connected dynamic B-spline patches. When two patches are stitched together, continuity requirements must be maintained through physics based constraints such as torsion springs to cover the seam. We anticipate a more convenient modelling technique of sculpting across the stitching seam because our surface-object haptic interaction approach makes it possible to touch and deform the two connecting patches along the seam at the same time.

REFERENCES

[1] FreeForm Modeling^{TM}, SensAble Technologies, Inc., http://www.sensable.com

[2] PHANTOM^{TM}, SensAble Technologies, Inc., http://www.sensable.com

[3] Bloomenthal J. and C. Bajaj, *Introduction to implicit surfaces*. San Francisco, Calif.: Morgan KaufmannPublishers, 1997.

[4] Dachille F., H. Qin and A. Kaufman, "A novel haptics-based interface and sculpting system for physics-based geometric design," *Computer-Aided Design*, vol. Volume 33, pp. 403-420, 2001.

[5] Gregory A., A. Mascarenhas, S. Ehmann, M. Lin and D. Manocha, "Six Degree-of-Freedom Haptic Display of Polygonal Models," presented at Proceedings Visualization 2000, 2000.

[6] Ho C. H., C. Basdogam and M. A. Srinivasan, "Haptic rendering: point- and ray-based interactions," presented a the second PHANToM Users Group Workshop, 1997.

[7] Kim L., A. Kyrikou, G. S. Sukhatme and M. Desbrun, "An Implicit-based Haptic Rendering Technique," presented at IEEE IROS 2002, Switzerland, 2002.

[8] McNeely W. A., K. D. Puterbaugh and J. J. Troy, "Six Degree-of-Freedom Haptic Rendering Using VoxelSampling," presented at SIGGRAPH 99, Los Angeles, CA USA, 1999.

[9] Nelson D. D., D. E. Johnson and E. Cohen, "Haptic Rendering of Surface-to-Surface Sculpted Model Interaction,"presented at 8th Annual Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems, Nashville, TN, 1999.

[10] Sener B., P. Wormald and I. Campbell, "Evaluating a Haptic Modelling System with Industrial Designers,"presented at EuroHaptics, 2002.

[11] Physically Based Modeling, Pixar, www.pixar.com/companyinfo/research/pbm2001/

[12] Zilles C. and J. K. Salisbury, "A Constraint-based God-object Method For Haptic Display," presented at ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems, 1994.