Haptic sculpting of multi-resolution B-spline surfaces with shaped tools

Zhan Gao a, Ian Gibson b, *

a Department of Mechanical Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, China
b Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, Singapore 117576, Singapore

Received 12 October 2005; accepted 25 February 2006

Abstract

In this paper, we first propose an implicit surface to B-spline surface haptic interface, which provides both force and torque feedback. We then present a new haptic sculpting system for B-spline surfaces with shaped tools of implicit surface. In the physical world, people touch or sculpt with their fingers or tools, instead of just manipulating points. Shaped virtual sculpting tools help users to relate the virtual modeling process to physical-world experience. Various novel haptic sculpting operations are developed to make the sculpting of B-spline surfaces more intuitive. Wavelet-based multi-resolution tools are provided to let modelers adjust the resolution of sculpture surfaces and thus the scale of deformation can be easily controlled. Moreover, sweep editing and 3D texture have been implemented by taking advantage of both the wavelet technique and haptic sculpting tools.

Keywords: Geometric design; Haptic sculpting; Haptic rendering; Multi-resolution

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. Several editing operations, such as control point manipulation, direct manipulation, free form deformation (FFD) and variational modeling, can be employed to generate surface models. However, with traditional two-dimensional (2D) based human–computer interfaces, it is still difficult to create and edit complex freeform CAD surface models. The fast developing haptic technology provides new potential by allowing human operators to interact with digital models using a sense of touch [13]. Haptic sculpting is a modeling technique for sculpting virtual models while simultaneously providing haptic feedback. The sense of touch adds a new modality to virtual sculpting.

In spite of the power of haptic interfaces, modelers may still suffer similar problems to using the traditional keyboard and mouse interface. Sometimes, the modeler may need the control point mesh of the surface to be dense enough for adding further details, but for other situations, the modeler may want a sparse control point mesh, which deforms more globally. A multi-resolution framework is valuable and useful in CAD because both the higher resolution details and the lower resolution sweep are simultaneously available. Therefore, a model can be edited at different resolution levels for different purposes. Although multi-resolution B-spline surfaces can deal with global details and fine local details, traditional 2D interfaces or 3D non-haptic virtual interfaces cannot fully exploit the advantages of the multi-resolution nature of B-spline surfaces because it is difficult and time-consuming to process and modify fine details with traditional human computer interfaces.

By trying to combine the advantages of a haptic interface with a multi-resolution framework, we propose a haptic sculpting system, which features a new tool–model haptic interaction and a multi-resolution B-spline surface sculpting technique. The final shape of the model not only depends on the material property of the model and 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 modeling system, which provides arrays of modeling tools of different shapes and sizes to help modelers relate the virtual sculpting with their experiences of sculpting in the physical world [27]. The shapes of tools also serve as visual hints that correlate the desired deformation on models with the shape of selected tool. In our system, we have developed shaped probe/tools with that purpose so that modelers can anticipate some result before action.
2. Related work

2.1. B-spline surface design

B-spline or NURBS surface models are widely used in CAD applications. Various techniques have been developed to improve the modeling and editing process. Sedegerg and Parry demonstrated a deformation method called free-form deformation (FFD) for global editing [24]. Coquillart extended this method for more general deformations [6]. Celniker made of groups of points, line segments and polygons [3]. The existing techniques for haptic rendering can be categorized according to the way the probing object is modeled: (1) a point, (2) a line segment, or (3) a 3D object made of groups of points, line segments and polygons [3].

The point-object haptic rendering paradigm assumes that we interact with the virtual world with a single point probe, therefore only the three Euclidean interaction force components can be fed back. Various point-object style approaches for haptically rendering triangular mesh virtual objects can be found in [22,30,33]. Because use of implicit surface is convenient for collision detection, some researchers have managed to present point-based haptic rendering techniques with implicit surface. Salisbury and Tarr introduced an algorithm for virtual objects based on implicit surfaces with an analytical representation [23]. Kim introduced a haptic algorithm, which is for a non-analytical implicit surface [16].

Although the point-object interaction metaphor has proven to be convincingly useful and efficient, it has limitations of being unable to provide torques and hence cannot offer sufficient dexterity and control. Basdogan et al. implemented a ray-based haptic rendering method, which can provide 5-DOF interaction between a line segment probe and virtual objects [2]. However, line segments are unsuitable to represent sculpting tools in this research. The object-object haptic interface can introduce a much more complex haptic cursor into the haptic simulation, thus improving the degree of realism and hence is desirable for many applications. Although a number of object-object 6-DOF haptic rendering methods are available, their applications have been limited mainly due to computational complexity and restoring forces that must be computed at the desired force updates (typically 1000 Hz). Gregory et al. presented an algorithm for haptic display of moderately complex polygonal models with a polygonal haptic cursor [11]. Their method features applicability to dynamic environments and accurate contact determination. However, this method requires that each object must be convex or must be decomposed into convex primitives. This is undesirable in haptic sculpting systems because it entails real-time decomposition of deformable objects into convex primitives. McNeely et al. put forward a simple, fast, and approximate voxel-based approach. This approach enables the manipulation of a moderately complex haptic cursor within an arbitrarily complex environment of rigid objects [20]. However, McNeely’s method is not suitable for deformable surface or B-Rep modeling because virtual objects have to be voxelized with this method and this would be too expensive. This method would be highly efficient when dealing with static objects or volumetric sculpting. Balasubramaniam et al. presented a 5-DOF haptic machining interface by using combinations of spheres and cylinders to represent a milling tool while point clouds represent contact models [1]. This haptic interface is quite close to our approach. However, geometric shape for sculpting tools provided in this method is too simple for haptic sculpting, although they are good at simulating ball-head milling tools.

One application for a haptic interface is haptic sculpting. Haptic sculpting is a modeling technique based on the notion of sculpting a solid material with tools. Polygonal mesh is a common model representation in haptic systems. In the haptic sculpting system called inTouch [10], users can interactively edit and paint subdivision-based polygonal meshes. Volume sculpting is widely used in virtual sculpting because it is capable of modeling objects in arbitrary topology, examples include [5,15]. The FreeForm haptic modeling 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 [26]. The FreeForm system is also based on volumetric modeling techniques. The research group led by Qin at State University of New York developed several novel haptic sculpting methods based on some other geometric model representations, such as subdivision solids [19], volumetric implicit functions [14] and point set models [12].

Although geometric representations like polygonal mesh, volumetric model, subdivision solids or volumetric implicit function have certain advantages, these model representations are unsuitable for mainstream CAD applications. Models from these sculpting systems are difficult, if not impossible, to transfer to CAD applications, such as Solidworks or UG [18]. In [25], Sener investigated how industrial designers use FreeForm and emphasized the importance of file exchange. Therefore, haptic sculpting directly on NURBS surface or B-Rep model is desirable. Dachille and Qin put forward a novel haptic interface and presented a physics-based B-spline surface modeling approach [7]. Their work demonstrates the feasibility of using haptic interface to manipulate B-spline surface models and hence is considered as a breakthrough in both CAD and computer haptics field. In this research, we adapt their technique of transform mass-spring mesh into B-spline surface. Liu et al. also developed a haptic design system for CAD...
surface models [17]. Both Dachille and Liu’s systems use point–surface haptic interfaces. One shortcoming of point–surface interaction is the difficulty in anticipating results in advance. Haptic sculpting with point–surface interfaces normally works like the following: a user picks a point upon a surface with the point haptic cursor so that the surface is connected with the haptic cursor temporarily; when the user manipulates the haptic cursor, the surface deforms accordingly. However, the user often does not know exactly how the deformation is going to be—is the deformation going to be highly curved or flatter, or, is the deformation going to be global or local? Although the extent of deformation can be adjusted through physics-based or multi-resolution methods, the sculpting process is mostly trial-error style.

To sculpt physical models, a sculptor works in a way totally different from the point–surface haptic sculpting—the sculptor chooses different sculpting tools according to different purposes. To sculpt details, he/she picks a smaller sculpting tool; to cut larger lumps of material, the sculptor often uses bigger tools. This paper proposes a haptic sculpting technique by emulating how sculptors work in real life—let the user choose different sculpting tools according to a specific task instead of giving him/her several non-intuitive parameters to adjust. For example, to get a large and flatter deformation, the user can pick a tool like the one in Fig. 3(c) which is big and flatter; to work on finer or more curved region, the tool in Fig. 3(a) could help.

3. System overview

Every virtual haptic sculpting system includes two major 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 smooth and stable haptic feedback, haptic rendering generally requires 1 kHz updating rate, which is much higher than graphics rendering update rate. The prerequisite for tool–model haptic interaction is an efficient and computationally inexpensive object–object haptic rendering algorithm. Computational inexpensiveness is very important because the physics-based deformation of a model requires extensive computational resources. Current available object–object haptic rendering methods are either too expensive or unsuitable to be incorporated into a physics-based B-spline surface modeling system. Therefore, we developed an implicit surface to B-spline surface haptic rendering algorithm for 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.

In this system, the geometric engine consists of two components: (1) a physics-based sculpting simulator; (2) a wavelet-based B-spline multi-resolution surface manager. A new B-spline surface sculpting method has been developed by taking advantage of the new tool–model haptic interaction technique. According to the user’s modeling action, the physics-based sculpting simulator dynamically deforms the mass spring mesh with numeric solutions. Another task of the physics-based sculpting simulator is to find the corresponding B-spline surface of the mass-spring mesh and graphically render the B-spline surface at 50 Hz. The multi-resolution surface manager is responsible for wavelet related operations such as decomposition and reconstruction so that the resolution of the B-spline surface can be adjusted. Therefore, the B-spline surface patch can be edited at different resolutions for different purposes.

An overview of the system is shown in Fig. 1. Besides the two major components, the figure shows another component—the graphic rendering component. Both the geometric engine and the haptic rendering component run at 1 KHz while the graphic rendering component runs at 50 Hz updating rate.

---

![Fig. 1. System overview.](image-url)
A PHANToM Premium haptic device is used as haptic input/output device in this system [28]. The PHANToM Premium device provides 6 degrees of freedom (DOFs) force and torque feedback and 6 DOFs position input. It is a small robot arm equipped with computer controlled DC motors. The tip of the PHANToM device is attached to a stylus that is to be held by the user. Motors on the PHANToM device can exert a certain force or torque to the tip of the stylus, in any direction, thus the user can feel a force or a torque applied to his/her hand through the stylus.

The system runs at a dual CPU workstation. Fig. 2 shows the system in use.

4. 6-DOF haptic rendering with shaped probes

6-DOF haptic feedback includes both force feedback and torque feedback. A probe and a 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.

4.1. Implicit surface representation

The implicit representation of a surface \( S \) is described by the following equation [3]

\[
S = \{(x,y,z) \in R^3| f(x,y,z) = 0\}
\]

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. If the potential value of \( f(x,y,z) \) is 0, then the point \((x,y,z)\) is on 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. Fig. 3 shows several tools with different implicit surfaces. The shape and size of probe/tool are defined by the three axes of ellipsoid which can be adjusted independently by users.

4.2. 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 during every haptic servo loop. B-spline surfaces are discretized into a layer of uniformly distributed sampling points. 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 the probe, a collision is detected.

Fig. 4 shows a model being touched by an ellipsoid probe. The dots on surfaces are the surface sampling points. Note every surface patch is bounded by an offset boundary box which largely improves collision detection efficiency.

4.3. Force and torque feedback

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.

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 [33]. 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, $s$ is the depth of the point and $\dot{s}$ is the velocity of the point. In our method, we use not just one spring, but a distributed set of damping springs. Fig. 5 shows a probe penetrating a planar surface of a model. White dots denote the sampling points inside the probe and black ones are sampling points outside. The depths of colliding points are the distances between those inside points to the correspondingly closest points on the implicit surface of the probe/tool.

We define the magnitude of force as

$$|F| = \sum_{i=0}^{n} (k \cdot s_i - d \cdot \dot{s}_i)$$  \hspace{1cm} (2)

where $n$ is the number of surface sampling points inside probe.

Fig. 6 shows what happens when a spherical probe, with a diameter of 800 units, passes through a plane with continuous, although not linear, force magnitude. The output force reaches the maximum when the intrusion depth is 400 units. At that moment the center of the probe touches the plane so that the probe has the largest contact area with a surface. After the peak the force magnitude decreases.

Penalty-based approaches and constraint-based approaches generally compute force vectors according to the normal vector of contacted surface [33]. Our approach is called the probe normal method, where force vectors are not derived from the surface being explored but from the implicit surface of the probe instead. If there is no friction between the probe and model surface, the pressing 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.

We therefore formulate the direction of force as

$$F = \frac{\sum_{i=0}^{n} \vec{N}_i \cdot s_i}{n}$$  \hspace{1cm} (3)

where $n$ is the number of surface sampling points inside probe, $\vec{N}_i$ as the normal vector of a surface sampling point $P_i$ and $S_i$ 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.

The torque is defined as following

$$\vec{T} = \sum_{i=0}^{n} \vec{F}_i \times \vec{r}_i = \sum_{i=0}^{n} (k \cdot s_i - d \cdot \dot{s}_i) \cdot \vec{N}_i \times \vec{r}_i$$  \hspace{1cm} (4)

where $n$, $\vec{N}_i$, $k$, $d$ and $S_i$ have the same meaning as in the Eq. (3), $\vec{F}_i$ is the contact force vector applied at a point $P_i$, $\vec{r}_i$ is the radius vector from the center of probe to $P_i$.

4.4. Discussion

This haptic rendering provides both force and torque feedback thus making tool–model interaction more realistic. Although this algorithm is not highly optimized, it is relatively easy to implement and satisfy the requirements of haptic sculpting of B-spline surface models.

For haptic 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 sculpting tools in real life have relatively 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. We chose implicit surface as the representation for probe/tool mainly because of its efficiency for collision detection and the simplicity of shape that the implicit surface can offer. The experimental sculpting system runs on a dual 2.0 GHz CPU PC workstation with two gigabyte memory. The system takes 0.5 ms to haptically render a single surface patch discretized as 4000 points, which is enough to define a relatively complex surface. Boundary box technique can further accelerate the collision detection. For example, to render the model in Fig. 4 the system only needs 0.3 ms. The efficiency of this algorithm leaves enough computing resource for the more complex physics-based numerical simulation. This haptic rendering method can process models with arbitrary shape without any prerequisites such as the model having to be convex. This property makes model decomposition in every updating loop exempted. Also, the haptic sculpting unit requires finding out which mass point is inside the sculpting tool and its depth. The inside/outside property of implicit function makes this task very easy.

This 6DOF haptic rendering method provides both force and torque feedbacks, which offer sufficient dexterity and control for haptic sculpting applications. The torque feedback is desirable to simulate more general tool–object interactions so that the user can feel a torque in his/her hand as the sculpting tool passes over a sculpture surface. Although torque feedback does enhance realism, it is not an essential aspect in this haptic sculpting system. In the earlier stages of this project, the haptic device used was a PHANToM Desktop, which can only provide 3DOF force feedback. We found users can control shaped sculpting tools (particularly spherical tools) quite well without torque feedback.

5. 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 and the model is the mass spring mesh. Modelers use a haptic device to deform a mass spring mesh in a physics-based manner. A B-spline surface patch corresponding to the mass spring mesh is rendered and updated graphically in every graphic updating loop. A double-way conversion scheme can convert the mass spring mesh into a B-spline surface patch and vice versa.

5.1. Mass spring mesh

The term mass spring system (MSS) 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. In this study, mass spring system (MSS) is chosen for modeling the deformation of a surface.

Fig. 7 demonstrates the structure of part of our MSS model. Mass points are depicted as balls and are connected by two types of tension springs. Main tension springs are responsible for maintaining the distance between mass points. Diagonal tension springs are responsible for resisting shearing deformation of the MSS. In order to control surface deformation, torque springs are employed to maintain angles between two connecting main springs. In our system, corner mass points do not own any torque springs and edge mass points have only one torque spring. The remaining mass points have two torque springs. Forces applied on a mass point can be categorized into two types: external forces and internal forces. The external forces are applied by sculpting tools. The internal forces include damping forces, forces from tension springs and torque springs. Let us denote the external force on the ith point as \( f_E \), the internal force on the ith point as \( f_I \), the damping force on the ith point as \( f_D \), the tension spring force on the ith point as \( f_T \) and the torque spring force on the ith point as \( f_T \).

The total force on point \( i \) is

\[
F_i = f_E + f_I
\]

where:

\[
f_I = f_D + f_T
\]

The damping force is

\[
f_D = -K_d v_i
\]

where \( K_d \) is the damping constant and \( v_i \) is the velocity of the ith point.

The tension spring force \( f_L \) is

\[
f_L = K_m \sum_j \frac{l_{ij}}{|l_{ij}|} (|l_{ij}| - r_{ij}) + K_s \sum_k \frac{l_{ik}}{|l_{ik}|} (|l_{ik}| - r_{ik})
\]
where \( l_{ij} \) is the vector from the \( i \)th point to its \( j \)th lateral neighbour point, \( l_{ik} \) is the vector from the \( i \)th point to its \( k \)th diagonal neighbour point, \( r_{ij} \) is the rest length of main tension springs, \( r_{ik} \) the rest length of diagonal tension spring, \( K_m \) is the spring constant of main tension spring and \( K_r \) is the spring constant of diagonal tension spring.

Equation for \( f_T \) is

\[
f_T = -1 \cdot \sum_j \text{Axis} \times l_{ij} \left( \frac{K_T \cdot (\theta_r - \theta)}{|l_{ij}|} \right)
\]

where \( l_{ij} \) is the vector from the \( i \)th point to its \( j \)th lateral neighbour point, \( \text{Axis} \) is the vector of cross product of two connecting \( l_{ij} \) and \( \theta \) is the angle between two connecting \( l_{ij} \) and \( \theta_r \) is the corresponding rest angle of torque spring.

The external force on mass points is due to the contact with the sculpting tool. Force magnitudes and directions are computed in a way similar to haptic rendering in Section 4. The force magnitude is:

\[
F_e = ks
\]

The force direction is the normal vector of that mass point. Fig. 8 shows the tool applying external forces to the mass points. Therefore, we can find that the effect of external forces is to push the mass points outward from the sculpting tool.

To find out the new position of mass points, the most straightforward approach is Euler integration. Since Euler integration for MSS deformation requires small time steps to maintain stability, it runs at 1 kHz. The initial value of positions, velocities, and external forces are preset before the program runs.

In our system, users use a haptic device to deform mass spring systems in a physics-based manner. However, the mass spring mesh is not the final result we want. We need to convert the mass spring system to a visually acceptable B-spline surface. This conversion from MSS to B-spline surface has to be conducted at each cycle of the visual rendering updating loop, which runs at 50 Hz frequency in our system. Due to the large consumption of computation resource by the haptic rendering module and numeric simulation of MSS, the algorithm of the conversion from MSS to B-spline surface must be efficient, computationally inexpensive and fast.

Dachille and Qin’s method assigns the masses directly on the B-spline surface and hides the control points from the user [7]. 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. Because their method does not introduce any linear or non-linear system solving, it is not expensive computationally. Traditional global or local interpolation methods are more robust than Dachille’s method, but they are generally more demanding. Therefore, Dachille’s mapping approach is adopted by us to achieve smooth and fast graphic rendering. A bicubic B-spline surface can be obtained through Dachille’s mapping approach. Because the surface does not have any multiple knots or coincident control points, the bicubic B-spline surface is \( C^2 \) continuous.

5.2. Sculpting

Both tension and torque springs in the mass spring system have their resting value: resting lengths or resting angles. After an initialization step, the system runs in a loop and continuously updates the physical state of the mass spring system. At each time step of 1 ms, the system samples the 6 DOF position of the PHANToM device and updates the position and orientation of the tool. The haptic rendering component computes the force feedback according to the updated position of the sculpting tool. When the modeller touches the surface with the sculpting tool and presses the PHANToM stylus button, sculpting forces are applied to the mass points. The sculpting system traverses all mass points and springs to compute the internal forces and external forces. Then, the mass points of the MSS evolve to new positions through Euler numerical integration. 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.

5.2.1. Pushing and pulling operation

The simplest haptic sculpting operations are pushing and pulling. Pushing and pulling operations are much like the way users sculpt with FreeForm [9]. To sculpt, users touch the surface with a sculpting tool in the first place, then press the PHANToM stylus button to push or pull the surface and the surface deforms according to the manipulation of the user. Movement of the tool during pushing and pulling operations is often approximately perpendicular to the model surfaces.

Pushing and pulling usually creates bumps or dents which carry the shape and size of sculpting tools—a spherical tool creates a circular bump while an ellipsoidal tool can make an elliptical deformation. Since the 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 interfaces. Our system permits users to adjust the size and shape of the sculpting tool. By changing the size of tool, users can determine themselves how wide a portion they want to contact with the surface. By changing the shape of the tool, users can have different deformation shapes and corresponding effects. In Fig. 9 (a) and (b), the deformations on each surface copies the shape of sculpting tool.
5.2.2. Stiffness and mass of the MSS

The final shape of the surface also depends on the stiffness and mass of the model. Changing the overall mass of the MSS changes the extent of deformation, allowing the surface to react either wildly or mildly to sculpting forces. In Fig. 9(b and c), two B-spline surfaces, both at $30 \times 30$ grids, but with different overall mass, are sculpted with the same tool. They demonstrate different behaviors during deformation. Fig. 9(c) shows a lighter MSS having a bigger deformation extent and generating a feeling of softer material. In Fig. 9(b), the mass of that MSS is 100 times heavier than the one in Fig. 9(c). The result shows that the deformation is much more local and a sharp edge appears around the bulge.
5.2.3. Strokes

Fig. 9 shows how bumps and dents can be easily made by pushing or pulling the sculpting tools upon the surface. Although pushing and pulling operations make these primitive deformations easy to achieve, real sculpting activities require much more complex deformations. For example, none of the shapes in Figs. 10 and 11 can be created by simply pushing or pulling the MSS-B-spline surface because no sculpting tools with similar shapes are available. Another haptic sculpting operation, namely a ‘stroke’, has to be introduced. A stroke means a movement of a pen when writing or of a tool when sculpting. To make a stroke in this system, the user slides the sculpting tool over the model surface. For example, from the original cylindrical surface in Fig. 10(a), a single up–down stroke can make a deformation like that in Fig. 10(b). Fig. 10(c) shows how the sculpting tool slides from top to bottom to make the shape in Fig. 10(b). A more complex example can be seen in Fig. 11. This shape was created with a spherical tool and three strokes.

The examples show how moderately complex shapes can be created through several strokes with the simplest tool—a sphere. Although the types of sculpting tools provided in this system are relatively simple and limited, we do not consider it as a major drawback because a combination of different sculpting tools and stroke operations can create various shapes. Also, one can note that the stroke operation makes the sculpting process more intuitive. For example, to create the shape of ‘2’ in Fig. 11, with our methods, the user simply sculpts a ‘2’ on the surface instead of having to specify many constraints with point–surface haptic interfaces.

5.2.4. Plastic and elastic deformation

For mass spring systems, elastic deformation means the resting lengths or angles of each spring are not changed after deformation so that the deformed MSS is always ready to revert to its original shape. When users finish sculpting, MSS numerical simulation stops so that the MSS maintains its deformed shape. The numerical simulation restarts when users start to sculpt again. If the previous deformation is elastic, the deformed feature will fade away and the MSS reverts to its original shape. Fig. 12 (a) and (b) demonstrate the elastic deformation and self-reversing of shape. In Fig. 12(a), a big bulge is formed in the center. In Fig. 12(b), the user makes another smaller bulge at the corner; however the bigger bulge diminishes when making the smaller one.

In our system, plastic deformation changes the resting values of springs. For the numerical integration process, the plastic deformation and the elastic deformation is the same. The difference is that the resting values of springs are updated to the current parameters after the numerical integration process is finished in the plastic deformation.

Fig. 12(c) shows the application of plastic deformation. After plastic deformation of the bigger bulge, when the user sculpts the smaller one at the corner, the bigger one does not have any discernible deformation. Therefore, the previous sculpting work remains.

We believe both forms of deformation are useful in actual modeling work. Plastic deformation helps to preserve previous modeling work while elastic deformation and stroke operation altogether can produce smoother shapes. Users can choose between elastic or plastic deformation according to their preference.

![Fig. 11. Example of stroke operations.](image)

![Fig. 12. Elastic and plastic deformation.](image)
5.3. Case study: haptic sculpting of a human nose shape

In this section, the proposed haptic sculpting method is illustrated through a case study: the modeling of a human nose shape. Currently, our haptic sculpting system supports only one single B-spline patch. This leads to a limitation to the performance and expression of this modeling approach. Complex free-form shapes, such as a human hand, are difficult or impossible to model with our sculpting system at present.

We choose modeling a human nose on a single B-spline surface as the example to demonstrate and evaluate our haptic sculpting technique. Although it looks simple, to model a human nose is not a trivial task, no matter what kind of modeling technique is used. This example demonstrates how different shapes and sizes of sculpting tools, together with MSS stiffness adjustment and elastic/plastic deformation, can be combined to create such an organic form. Four steps, from Figs. 13–16, demonstrate the process of modeling. At the end of each step, the plastic deformation function was run to freeze previous deformations. The left side pictures in Figs. 13–16 are the front views of the nose sculpture. The pictures in center are side views. The diagrams at right side show how the model is sculpted.

Sculpting starts from a planar MSS-B-spline surface. The MSS has $40 \times 40$ mass points. In the first step, as Fig. 13 shows, a rough shape of a nose was created. An ellipsoid tool was used because its shape looks similar to the rough shape of a nose. Three axes of the ellipsoid were 30, 15 and 15 units,

![Fig. 13. The first step.](image1)

![Fig. 14. The second step.](image2)

![Fig. 15. The third step.](image3)
respectively. Because the task of the first step was a rough shape, sharp edges were unnecessary and mild deformation was preferred. Therefore, the stiffness of MSS was set as low. To get the shape in Fig. 13, one pulling and one stroke operations were required. The user tilted the ellipsoid sculpting tool and pulled back from behind the surface, and then had to slide the tool up along the surface to get the nose bridge. Arrows in the right side diagram of Fig. 13 show the sculpting operations.

In the second step (Fig. 14), more stroke operations were carried out to make the upper part of the nose bridge more narrow and the bottom of the nose more pronounced. Three axes of the ellipsoid tool were 30, 15 and 20 units to make it a little wider and flatter. With this kind of ellipsoid tool, it was easier to make a large, flat area as required in this step.

Fig. 15 shows the third step. Since some fine details were added to the model from here, the stiffness of MSS was increased. A spherical tool of radius 15 was used to sculpt the tip and wings of the nose. The bumps on the nose tip or nose wings are very subtle, so, the user had to be careful with these operations. Three fine pulling operations can achieve satisfying results. This step demonstrates the sensitivity and controllability of our sculpting method.

Fig. 16 demonstrates the last step, where nostrils were modeled with push operations. A spherical tool was used in this step as well. It is smaller than the previous step to fit the size of nostrils. To get such dents is particularly easy with our approach. Two simple push operations got the nostrils done.

During the modeling of the nose, four pulling operations, two pushing operations and four strokes have been applied. Besides these haptic interface operations, six keyboard and mouse operations are performed to adjust stiffness of MSS or to execute plastic deformation. The example shows how both ellipsoid and spherical tools are used. The size of sculpting tool can be adjusted according to specific situations.

A point–surface haptic sculpting interface based on the same MSS-B-spline modeling method was set-up as a contrast. The point–surface interface has all of the functions of our physics-based MSS-B-spline, such as MSS stiffness adjustment and plastic/elastic deformation. A user can manipulate the MSS-B-spline surface by pushing and pulling it with a point haptic cursor. During sculpting with this method, no relative sliding between the point haptic cursor and the surface is allowed. Thus, stroke is impossible with point–surface interface. To model a nose similar to the one in Fig. 16, 27 haptic manipulations were recorded, and numerous keyboard and mouse operations undertaken. In particular, we found that the nose wing and nostrils are difficult to model with point–surface interface because the extent of deformation is hard to control. The comparison experiment shows our tool–object haptic sculpting interface needs fewer sculpting operations than the point–surface interface to accomplish the same task. Therefore, our haptic sculpting method can be considered as more intuitive than the point–surface sculpting interface.

One can note that the geometric coverage of ellipsoid or sphere as sculpting tools is limited. For example, it would be difficult to create a plane with ellipsoid tools. However, this limitation can be solved by using more traditional modeling methods and a conventional human–computer interface. The haptic sculpting method in this paper should complement, instead of replace the traditional geometric modeling methods. Although pushing/pulling and stroke operations make this modeling approach distinct to other B-spline surface modeling methods, these operations are far from enough. To fully take advantage of a haptic interface, point–surface should be employed as well because it is good at modeling more global or wider deformation. Furthermore, position or curvature constraints would also be preferred in order to precisely control the modeling process.

Our system allows users to adjust the resolution and size of MSS-B-spline surface. The resolution of MSS is only constrained by the power of computer. On a dual 2.0 GHz CPU PC workstation, maximally around 2500 (50 × 50) mass points can be acceptable while less than 1600 (40 × 40) mass points can have very smooth performance. The adjustment of resolution is achieved through wavelet multi-resolution methods, which are to be discussed in Section 6.

6. Wavelet-based multi-resolution B-spline surface

In our system, of course haptic sculpting tools are the most important ones. Besides haptic sculpting tools, our sculpting system features multi-resolution modeling tools, which allow the user to adjust the resolution of models and help introduce sweep design and 3D texture. Forsey et al. suggested applying
6.1. Wavelets for B-spline surface

Finkelstein and Salesin put forward a wavelet-based multi-resolution B-spline curve representation [8].

Under the wavelet-based B-spline multi-resolution framework, a B-spline curve can be decomposed into a low resolution curve and a detail part. If the original curve is \( \gamma(u) = \varphi^j C^j \), where \( j \) is resolution level, \( \varphi^j \) is basis function and \( C^j \) is control points, the decomposition can be represented as:

\[
\gamma^j(u) = \varphi^j C^j = \gamma^{j-1} + \beta^{j-1}
\]

where

\[
\gamma^{j-1}(u) = \varphi^{j-1}(u) C^{j-1}
\]

is the low-resolution version of the original curve \( \gamma(u) \) and \( C^{j-1} \) is its control points

\[
\beta^{j-1} = \psi^{j-1}(u) D^{j-1}
\]

is the detail information of the original curve. The lower resolution control points \( C^{j-1} \) and its corresponding detail \( D^{j-1} \) can be found by:

\[
C^{j-1} = A^j \cdot C^j
\]

\[
D^{j-1} = B^j \cdot C^j
\]

This process of obtaining \( C^{j-1} \) and \( D^{j-1} \) is called wavelet decomposition or analysis. The original control points \( C \) can be reconstructed from the lower resolution control points matrix \( C^{j-1} \) and the detail matrix \( D^{j-1} \):

\[
C = P^j \cdot C^{j-1} + Q^j \cdot D^{j-1}
\]

The process of recovering \( C \) from \( C^{j-1} \) and \( D^{j-1} \) is called wavelet reconstruction or synthesis. With these transformations we can decompose and synthesize the endpoint interpolating B-spline curves at different resolution levels. For more detailed information about wavelet-based multi-resolution B-spline curve, please refer to [29].

A bicubic B-spline surface can be decomposed and reconstructed in a similar way to the B-spline curve wavelet method. A B-spline surface, with resolution level \( j \) at \( u \) direction and level \( k \) at \( v \) direction, can be represented as in a matrix form

\[
S^{j,k}(u,v) = \varphi^j \varphi^k [\varphi^j]^T
\]

where \( \varphi^j \) and \( \varphi^k \) are basis functions matrices at scale level \( j \) and \( k \), respectively. \( C_{\varphi^j \varphi^k} \) is the control points matrix.

The surface control point matrix \( C_{\varphi^j \varphi^k} \) can be decomposed as:

\[
C_{\varphi^{j-1} \varphi^{k-1}} = A^j C_{\varphi^j \varphi^k} A^k \quad D_{\varphi^{j-1} \varphi^{k-1}} = B^j C_{\varphi^j \varphi^k} B^k
\]

The decomposition process of the B-spline surface is shown in Fig. 17, where \( C_{\varphi^{j-1} \varphi^{k-1}} \) is the lower resolution surface and \( D_{\varphi^{j-1} \varphi^{k-1}} \) and \( D_{\varphi^{j-1} \varphi^{k-1}} \) are detail matrices.

The decomposition process can be applied recursively until low enough resolution has been reached. Similar to multi-resolution B-spline curve reconstruction, the lower resolution surface and its details can be reconstructed to form the original surface:

\[
C_{\varphi^j \varphi^k} = P^j C_{\varphi^{j-1} \varphi^{k-1}} P^k + P^j D_{\varphi^{j-1} \varphi^{k-1}} Q^k + Q^j D_{\varphi^{j-1} \varphi^{k-1}} P^k
\]

The process of reconstruction is shown in Fig. 18.

In our system, both the mass spring system and the B-spline surface representation are used. In order to take advantage of multi-resolution, the representation’s resolution that is to be adjusted in the design process has to be determined first. During the sculpting process, the user first deforms the MSS mesh, and then the system updates the B-spline surface patch according to the MSS mesh. The user does not directly work on B-spline surface patches. Therefore, it seems reasonable to firstly adjust the resolution level of mass spring system to satisfy the user’s requirement and the B-spline surface patch’s resolution is determined by that of the MSS mesh. We first used endpoint interpolating linear B-spline wavelets to decompose and synthesize the MSS mesh. Although this strategy sounds
6.2. Sweep editing

Suppose we have a B-spline surface with control points \( C_{\phi, \psi} \). After decomposition we can get all of its lower resolution control points and details such as \( C_{\phi, \psi}^{1}, C_{\phi, \psi}^{2}, \ldots, C_{\phi, \psi}^{n} \) and \( D_{\phi, \psi}^{1}, D_{\phi, \psi}^{2}, \ldots, D_{\phi, \psi}^{n} \). If we modify a lower resolution surface and keep the detail information intact, the overall sweep of the original surface will be changed.

Let \( C_{\phi, \psi}^{0} \) be the control points of the original surface \( \gamma^{JK}(u,v) \), first the surface has to be decomposed to the desired low resolution level \( m \) and \( n \). If \( \Delta C_{\phi, \psi}^{0} \) is denoted as the change to the control point mesh at the resolution level \( m \) and \( n \), then the total change at the original level \( J \) and \( K \) can be given by:

\[
\Delta C_{\phi, \psi}^{0} = p^J p^{l-1} \ldots p^{m+1} \Delta C_{\phi, \psi}^{0} p^{n+1} \ldots p^{k-1} p^K
\]

Therefore, after the sweep editing at a lower resolution level, the control point mesh of the surface \( \gamma^{JK}(u,v) \) becomes:

\[
\hat{C}_{\phi, \psi}^{0} = C_{\phi, \psi}^{0} + \Delta C_{\phi, \psi}^{0}
\]

Editing at a lower resolution level results in a change of larger portion of the higher resolution level surface.

6.3. 3D texture

In [31], Wang et al. put forward a frequency-based method for freeform feature shape (3D texture) reuse. The frequency-based operators work in the 2D frequency domain, and control the shape by manipulating its frequency components. Complex 3D texture on B-spline surface can also be decomposed into frequency domain with wavelet method. Therefore, we implement surface 3D texturisation based on the wavelet multi-resolution method and MSS-B-spline frameworks.

Suppose we have a B-spline surface with control points \( C_{\phi, \psi}^{0} \), after decomposition we can get all of its lower resolution control points and details such as \( C_{\phi, \psi}^{1}, C_{\phi, \psi}^{2}, \ldots, C_{\phi, \psi}^{n} \) and \( D_{\phi, \psi}^{1}, D_{\phi, \psi}^{2}, \ldots, D_{\phi, \psi}^{n} \). 3D texturisation is simply replacing the detail information \( D_{\phi, \psi}^{1}, D_{\phi, \psi}^{2}, \ldots, D_{\phi, \psi}^{n} \) with another set of new detail information and reconstructing. Detail information of 3D texture can be produced by the user through real time sculpting; it could also be extracted from reverse engineering data. In this way, a 3D texture library can
be implemented easily. Fig. 20(a) shows the original surface. Fig. 20 (b) and (c) are surfaces with different textures. These two textures were conveniently created with this haptic sculpting technique. Our haptic sculpting method provides a very effective and intuitive way of creating desired fine details for the multi-resolution framework. These details could be rather difficult to obtain with other modeling methods.

7. Examples

Fig. 21 shows the application of our sculpting method on the creation of an ergonomic hand grip model. In Fig. 21(a), the initial stage of model with little organic details was created in Solidworks with the function of lofting. Then the Solidworks model, in the form of B-Rep model, was transferred into our system to add more details to the lower part surface of the model. In Fig. 21(b), three pulling operations were applied to the surface. In Fig. 21(c), ‘HKU’ were sculpted with several strokes. In Fig. 21(d), 3D texturisation of hammered effect was applied to the model in Fig. 21(b). To maintain the topology of the B-Rep model and the continuity between the surfaces, mass points neighboring or close to the boundary were constrained.
8. Conclusion and future work

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 tool. In our haptic sculpting system, the MSS-B-spline surface responds to forces applied by the modeler in an intuitive way and the force and torque feedbacks significantly improve the sense of realism. The haptic rendering method can satisfy the requirements of haptic sculpting. Our system offers users haptic tools and multi-resolution tools. Haptic tools support sculpting with ellipsoid or spherical tools via haptic feedback devices. Pushing/pulling operation, stroke operation and plastic/elastic deformation can be combined flexibly to create moderately complex free-form objects. The modeler may change the shape and size of tool dynamically to satisfy specific situations. The material property of model surface can also be modified to obtain soft or rigid sculpting effects. All these aspects result in an intuitive haptic sculpting tool. Multi-resolution tools allow modelers to adjust the resolution of sculpting surfaces; thus to control the scale of deformation becomes easier. On the other side, the haptic sculpting method provides an effective way of creating desired fine details for the multi-resolution framework.

The performance of this haptic sculpting method suffers from the limitation that currently only one single patch can be modeled. 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. We anticipate a more convenient modeling technique of sculpting across the stitching seam because our haptic interaction approach makes it possible to touch and deform the two connecting patches along the seam at the same time. Another issue that needs further investigation is the shape of sculpting tools. Tools with more complex shapes will be explored.

References


Zhan Gao is currently a research associate in the Department of Mechanical Engineering, University of Hong Kong. He received his PhD from the University of Hong Kong and his MS from Zhejiang University respectively. His research interests include CAD/CAM, virtual reality and computer haptics.

Ian Gibson is an Associate Professor in the Department of Mechanical Engineering at the National University of Singapore. He has been conducting research in the areas of virtual and rapid prototyping, medical engineering and haptic interfaces for several years.