Haptic-constraint modeling based on interactive metaballs

By Hui Chen*, Hanqiu Sun and Xiaogang Jin

Adding interactive haptic-constraint sensations is important in interactive computer gaming and 3D shape design. Usually constraints are set on vertices of the object to drive the deformation. How to simulate dynamic force constraints in interactive design is still a challenging task. In this paper, we propose a novel haptic-constraint modeling method based on interactive metaballs, during which the haptic-constraint tools are attracted to the target location and then control the touch-enabled deformation within the constrained areas. The interactive force feedbacks facilitate designers to accurately deform the target regions and fine carve the details as their intention on the objects. Our work studies how to apply touch sensation in such constrained deformations using interactive metaballs, thus users can truly feel and control the soft-touch objects during the deforming interactions. Experimental results show that the dynamic sense of touch during the haptic manipulation is intuitively simulated to users, via the interacting interface we have developed. Copyright © 2010 John Wiley & Sons, Ltd.

Received: 9 July 2009; Revised: 15 November 2009; Accepted: 25 November 2009

KEY WORDS: haptics interface; constrained deformation; metaballs

Introduction

Haptics investigates the touch-based interfaces supporting the force/tactile feedbacks to users. It is an active research field to improve the immersive reality of virtual environments, especially in simulations during which it is crucial for the operators to touch, grasp and manipulate objects in the virtual worlds. Haptic simulation of deformable objects needs to estimate not only the amount of object deformation but also the interacting forces that are reflected to users via haptic device. Adding soft-touch realism in deformable objects provides new touch characteristic in addition to graphical display and covers a wide range of applications, such as geometric modeling,1 computer animation,2 and on-line game development.3

Haptic constraints have been introduced especially in guiding the user motions in virtual environments. Rosenberg4 defined “virtual fixtures” as forces super-imposed in a remote environment. During which, the fixtures were defined as virtual rigid planar surfaces to help guide a user’s motion with force perceptions. Several principles of designing 3D haptic widgets especially in sketch-oriented interfaces were given by Miller and Zeleznik.5 Further, haptic widgets have been extended for haptic task constraints including depth and altitude constraints.6 Haptic guidance cuing user motions to follow path trajectory was realized with spring-like7 method and it was evaluated usefully for the purpose of skill training.8 More predictive guidance9 with complex trajectory was realized to aid manual control of interactive tasks.

Dynamic haptic constraints in deformable environments are also necessary to be developed to control the deformations flexibly and to make the soft-touch simulation more realistically.10 Constrained deformation based on metaballs is a flexible technique for designing surfaces interactively, but short of dynamic interaction of touch sensation. The focus of our work is to apply haptics in the process of such dynamic constrained deformations. Our proposed haptic-constraint modeling has the following novel features:

- Haptic-constraint force feedback

Different constraints are set on the specialized positions of the object during dynamic interactions, but...
sometimes such positions are difficult to catch such as within the narrow valley. Dynamic attractive force is taken into effect while users passing by to help users reach to and grab the targeted positions of constraints. Interaction force is then generated during the haptic manipulation of metaball constraint. Besides, haptic frequency extension method of interpolation filter filling the gap between deformation update rate and haptic rendering frequency is applied to remove the noise and acquire stable force feedback.

- Haptic deformation using interactive metaballs

A novel soft-touch haptic-constraint deformation based on metaball constraints is established to enhance the touch feeling during constrained modeling. The haptic deformation of the object is governed by the physics-based deformation of constraint skeleton and dynamically constructed metaball. Specialized metaballs are associated with the potential function centered on the haptic constraints to determine the force distributions and local deformations within the constrained areas.

In the following, Section 2 outlines the related previous work. Section 3 describes the proposed interactive haptic constraint deformation framework based on metaballs. Section 4 presents detailed haptic-constraint force feedback and haptic-constraint deformation method. Section 5 gives the experimental results of our soft-touch modeling system. Finally, the summary goes to Section 6.

Related Work

In this section we briefly overview the previous related work in haptic rendering, constrained deformation techniques, and haptics rendering of deformable objects.

Haptic Rendering

Haptic rendering refers to the computational methods used to determine the forces resulted when we interact with virtual objects. Based on the avatar manipulated via haptic input, haptic rendering methods for geometric models can be divided into point-based, ray-based, and object-based methods. In point-based methods,\textsuperscript{11,12} only the end point of the haptic input interacted with the virtual objects, and force computed was proportional to the displacement between haptic interaction point and surface contact point. In ray-based methods,\textsuperscript{13,14} the generic probe of haptic input was modeled as a line-segment whose orientation was taken into account, and the collisions were checked between line-segment and the objects. In object-based methods,\textsuperscript{15–19} the tool probe of the haptic input was modeled as a 3D object. The position and orientation of the 3D haptic tool were traced during the haptic interaction, and collisions during interaction were checked between 3D objects. The application of object–object haptic rendering algorithms to complex models and contact scenarios remains a challenging issue, especially due to the inherent cost of collision detection that induces slow force update. Otaduy and Lin\textsuperscript{20–22} have presented a sensation-preserving simplification technique for haptic rendering of complex polygonal models by selecting contact resolution adaptively. Their work focused on the acceleration of collision detection and response, while relied on previous techniques for displaying force and torque feedbacks.

Constraint Deformation

Constraint deformation is efficient in producing controlled spatial deformations. Constraints are usually set on vertices with user-specified displacements to drive the deformation of the object, namely Simple Constrained Deformation (SCODEF), first introduced by Borrel and Rappoport.\textsuperscript{23} The displacement of any point to be deformed was the blend of the local B-spline basis functions determined by the constraint points. Further the generalized SCODEF deformation method has been extended on subdivision surface\textsuperscript{24} and NURBS surfaces\textsuperscript{10} to permit the satisfaction of geometrical constraints. Another typical constraint deformation method is radial basis functions (RBFs).\textsuperscript{25–27} RBFs assumed surface smoothness as a minimal constraint and animations produced by controlling an arbitrary sparse set of control points on or near the surface of the model. The general idea of constraint deformation\textsuperscript{28} in physically based modeling was to include not only particles and forces, but restrictions on the way the particles were permitted to move. The key issue in constraint dynamics is to model the particles obey Newton’s laws, and at the same time the geometric constraints.

Metaball modeling\textsuperscript{29} is a flexible technique for implicit surfaces modeling with advantages such as point classification, intersection computation, and unbounded geometry. It is very convenient for designing closed surfaces, and provides simple solutions for creating...
blends, ramifications and advanced human character design. Blinn\textsuperscript{30} and Nishimura et al. \textsuperscript{31} set up a basic formulation of metaballs, where a free-form surface was defined as the isosurface of a scalar field which was generated from some field generating points. The field value at any point was determined by the distance to the generating points. Later Bloomenthal and Wyvill\textsuperscript{32} and Bloomenthal and Shoemake\textsuperscript{33} extended the original metaball idea to include complex skeletons, such as lines, surfaces, and volumes. The skeleton-based model provided an intuitive way to define the desired shapes with implicit surfaces. Jin et al. \textsuperscript{34} proposed a local deformation model based on generalized metaballs of implicit modeling. Geometric constraint elements including points, lines, surfaces and volumes were developed however without any physical behavior.

**Haptic Rendering of Deformable Objects**

Haptic simulation of deformable objects needs to estimate not only the deformation of the object nodes in space, but also the interaction forces that are reflected to users via haptic device. Usually a force model is loosely coupled in geometry-based methods. Colgate et al. \textsuperscript{35} proposed a virtual coupling scheme to bring between the haptic interface and the virtual environment successfully. The extension of geometry-based techniques to haptic display of deformable objects had applications in haptic sculpting\textsuperscript{36} and CAD system.\textsuperscript{2-37} Linearized coupling force was efficiently applied in large deformations or self-collision.\textsuperscript{38} In physics-based methods, the computation of interaction forces is part of physics-based models, so we do not need a separate model for forces generation. McDonnell et al. \textsuperscript{39} developed a haptic sculpting system based on a subdivision solid and mass-spring modeling. Dachille et al. \textsuperscript{40} established a similar system through dynamic NURBS (D-NURBS) to combine NURBS and mass-spring modeling. James and Pai\textsuperscript{41} and Basdogan\textsuperscript{42} have made modeling simplifications due to the limited computational power to implement real-time FEM with haptic displays. Duriez et al. \textsuperscript{42} simulated contact space modeling using Linear Finite Elements (LEM) for realistic deformable interactions. Haptics and physically based interactions have got especially focus in surgery simulation\textsuperscript{43,44} fabrics\textsuperscript{45} and hair simulations.\textsuperscript{46} Adding force feedback in deformable modeling has advantages to increase intuition, to control deformations, and to support the development of physical constraints.

**Interactive Haptic-Constraint Modeling**

Currently, the modeling process of metaball-constraint deformations is lack of dynamic interaction via touch sensation. Our objective is to incorporate intuitive haptic interaction with constrained metaballs during deformation process. In the proposed haptic-constraint modeling, first the constraint skeleton $C$ attached with haptic input is attracted to the target location on the object to be deformed. When haptic avatar moves nearby, the attractive force $F_a$ and torque $T_a$ take into effect to guide the users place the constraint skeleton to the target position and direction. Then, the interaction force $F_i$ and torque $T_i$ are generated with the operation of haptic-constraint tool to drive the deformation of the object. Metaball $M$ is constructed dynamically in relation with the movement of constraint skeleton and the distance of force transfer area. Finally, local influenced area on the object is deformed accordingly.

A generalized metaball $M = \langle S, F(d(P, C), R) \rangle$ centered on the skeleton $C$ is defined by a boundary surface $S$ and a potential function $F(d(P, C), R)$ is defined as the composition of weight function $f(d, R)$ and distance function $d(P, C)$ as follows:

$$S = \{ P(x, y, z) \in S | d(P, C) = R \} \quad (1)$$

$$F(d(P, C), R) = f(d, R) \circ d(P, C) \quad (2)$$

where $d(P, C)$ is a distance function which defines the minimal distance from point $P(x, y, z)$ in 3D space to the individual points $Q(u, v, w)$ on the skeleton $C$; and $f(d, R)$ is a weight function drops from 1 on the skeleton $C$ to 0 beyond the effective radius $R$ of the skeleton, here Wyvill’s degree six polynomial\textsuperscript{47} is adopted as the weight function (4), because this function blends well and can avoid the calculation of square root, as below

$$d(P, C) = \inf_{Q \in C} \sqrt{(x - u)^2 + (y - v)^2 + (z - w)^2} \quad (3)$$

$$f(d, R) = \begin{cases} 
1 - \frac{d^6}{d^6} & 0 \leq d \leq R \\
0 & d > R 
\end{cases} \quad (4)$$

The deformations of the object based on metaball constraints are generated in the following: Suppose $P(x, y, z)$ is a point on the object in $\mathbb{R}^3$, then $\text{Deform}(P)$: $\mathbb{R}^3 \to \mathbb{R}^3$ is a deformation function which maps $P$ to $\text{Deform}(P)$. Let $C$ be a constraint skeleton as above, $\Delta D$ be its displacement, and $M = \langle S, F(d(P, C), R) \rangle$ be the metaball centered on the same skeleton. The defor-
Deformation function is defined as

\[ \text{Deform}(P) = P + \Delta D \cdot F(d(P, C), R) \]  

(5)

Study the potential function of metaball, we have

\[ \text{Deform}(P) = \begin{cases} 
    P + \Delta D, & d(P, C) = 0 \\
    P + \Delta D \cdot f(d, R), & 0 < d(P, C) \leq R \\
    P, & d(P, C) > R
\end{cases} \]

Therefore, the deformation based on metaball constraint yields a local result which satisfies the desired displacement precisely on the constraint skeleton \( C \), and does not affect the points outside the effective radius \( R \) of the metaball \( M \). Accordingly, deformation function with \( n \) constraints has

\[ \text{Deform}(P) = P + \sum_{i=1}^{n} \Delta D_i F(d(P, C_i), R_i) \]  

(6)

The displacement of point \( P \) is the blend of the displacements of constraint skeletons weighted by the potential functions of their corresponding metaballs.

**Haptic-Constraint Interactions**

The force feedback and haptic-constraint deformation based on interactive metaballs are detailed in this section, and the haptic-constraint tools we used are specified in Appendix.

### Haptic Force Feedback

Often the targets are located within the narrow valley or place that are difficult to catch, thus the force and torque are employed to attract the haptic tool embedded with constraint skeleton to the target region to be deformed.

In Figure 1a, haptic input is attached to the center \( O_c \) of constraint skeleton \( C_v \) and the target location \( O_t \) is marked in dashed red lines on the object. Local coordinate of haptic avatar \( \tilde{\mathbf{y}'} \tilde{\mathbf{z}'} \tilde{\mathbf{z}'} \) is defined with the origin at the end of stylus in haptic device, which is at the center \( O_c \) of constraint skeleton, and the \( y'-\)axis is consistent with the direction of stylus, which is perpendicular to the constraint skeleton. The local coordinate \( \tilde{\mathbf{y}'} \tilde{\mathbf{z}'} \tilde{\mathbf{z}'} \) is defined at the target location \( O_t \). When haptic avatar is moving into the adjacent area within the distance \( r \) of target location, the attractive force \( F_a \) and torque \( T_a \) are taken into effect to flexibly place the haptic constraint tool to the target location

\[ F_a = k_a (\vec{h} - r) \frac{\vec{h}}{|\vec{h}|} + k_v (\vec{v} \cdot \vec{h}) \frac{\vec{v}}{|\vec{v}|} |\vec{h}| - r < 0, \]  

\[ T_a = k_{\omega} \vec{\omega} + k_v (\vec{\omega} \cdot \vec{\omega}) \]

(7)

where \( \vec{h} = \overline{\text{HIP} - O_c} \), HIP is the position of haptic input point, \( k_a \) and \( k_v \) are the coefficients of the attractive force and torque respectively, and \( k_v \) and \( k_{\omega} \) are the coefficients to resist the moving velocity \( \vec{v} \) and angular velocity \( \vec{\omega} \) of haptic input.

Once the constraint skeleton is attached to the target location, users start to deform the object via the interaction force. Figure 1b outlines the interactive force feedback model during the haptic-constraint deformation. Quasi-ray-casting method along target direction is applied on the sampled discrete points on the skeleton shown in blue dotted line. Discrete points \( P_i \) of the metaball skeleton on the object are sampled and dynamic manipulating forces are generated via tracing the movement of these points. Finally the composed manipulating force replied to haptic device and the
accompanied torque are given as follows:

\[
F_{di} = k_s(|\vec{h}_i| - l_{di}) + k_d \vec{v}_i \quad F_d = \sum_{i=1}^{n} F_{di}
\]

\[
T_{di} = (|\vec{h}_i| - l_{di}) \times F_{di} \quad T_d = \sum_{i=1}^{n} T_{di}
\]

where \(l_{di}\) is the rest length of spring tied to the point \(P_i\), \(k_s\) is the spring coefficient, \(k_d\) is the damping coefficient in relation to velocity \(\vec{v}_i\) of the same point. \(F_d\) and \(T_d\) are the final composed force feedback and torque replied to the user via haptic device.

**Interpolation Force Filter**

Acute changes of force amplitude occur occasionally and discrete force rendering by haptic device tends to vibrate when force amplitude is large. To acquire stable force feedback, second-order interpolation filter for haptic frequency extension is applied to fill the gap between deformation update rate and haptic rendering frequency.

Two interpolation filters are integrated in Figure 2, the output of unit 1 is sent to unit 2 as input. Both units use the similar filters. Each interpolation filter manipulates data by two steps. First it interpolates zeros into samples sequence \(x(n)\) to get higher frequent samples sequence \(w(m)\), and then it is sent to a low-pass filter (LPF) to remove high frequent interference signals to get the output \(y(m)\):

\[
W(m) = \begin{cases} 
\frac{x(n)}{m} & m = 0, \pm L, \pm 2L, \ldots \ , \ n \in \mathbb{Z} \\
0 & \text{otherwise}
\end{cases}
\]

\[
Y(m) = \sum_{k=-\infty}^{\infty} h(k)w(m-k)
\]

where \(L\) is the up-sampling factor, and \(h(k)\) is the filter coefficient.

Butterworth LPF with fine finite impulse response can preserve original signal magnitude and keep signal shape well after filter. A second-order Butterworth LPF is designed with cut off frequency \(\pi/5\). The length of its finite impulse response \(H_B(i)\) is 44, but the response magnitude is very small when \(i\) is larger than 15 shown in Figure 3. So the filter coefficient and output have:

\[
H_{Bh}(i) = \begin{cases} 
H_B(i) & i = 1, 2, \ldots, 15 \\
0 & \text{otherwise}
\end{cases}
\]

\[
Y(m) = \sum_{i=1}^{15} H_{Bh}(i)w(m-i)
\]

To expand frequency for 40 Hz to 1 kHz, this method only need a convolution computation with fifteen multiplications and additions per milliseconds in unit 2, and a similar convolution computation per 5 milliseconds in unit 1.

**Haptic Deformation of Interactive Metaballs**

The deformation of the object during interactive haptic manipulation is governed by the physics-based deformation of constraint skeleton and dynamically constructed metaball. The movement of constraint skeleton \(C_i\) on the object is governed by the Newton’s Second Law of motion. The displacement of the \(i\)th vertex \(u_i\) within

---

**Figure 2.** Second-order interpolation filter.

**Figure 3.** The impulse response of LPF.
target area on the object to an external force $F_d$ is given as follows:

$$m_i\ddot{u}_i + d_i\dot{u}_i + \sum_j k_j\left(\frac{|\vec{u}_i\vec{u}_j|}{|\vec{u}_i|} - l_{ij}\right) \vec{u}_i\vec{u}_j = F_d$$  (11)

where $m_i$ is the mass of the vertex $i$, $d_i$ is the damping constant of the same point, $l_{ij}$ and $k_j$ are the rest length and stiffness of the spring connecting two mass points, respectively. The right-hand term $F_d$ is the sum of external forces (e.g., gravity or other user applied dragging forces, here is the manipulation force). This equation is solved numerically with the central difference method along time steps. Dynamic metaball of constraint skeleton $C_i$ with effective radius $R_i$ is then constructed

$$R_i = \Delta D_i + \Delta T_i$$

where $\Delta D_i$ is the skeleton displacement of constraint $C_i$ computed by Equation (11) and $\Delta T_i$ is the distance of force transfer area, here average distance of four layers is selected as the optimum force transfer depth.

With the dynamic construction of metaball, haptic-constraint deformation is solved through attaching a local coordinate system to each constraint $C_i$ embedded in the haptic tool. Figure 4 shows an example of haptic-constraint interacting process. In Figure 4a, the circle skeleton is attracted to the target position on the object, the interactive metaballs of $M^t$ and $M^{t+1}$ are constructed dynamically during haptic manipulation at time $t$ and $t+1$ respectively in Figure 4b and c. $N$ is the transformation matrix made up of translation matrix $T(d_x, d_y, d_z)$ in relation with displacement of skeleton computed dynamically in Equation (11) and rotation matrix $R(\theta_x, \theta_y, \theta_z)$, having $N = T(d_x, d_y, d_z)R(\theta_x, \theta_y, \theta_z)$. For any point $P$ to be deformed within dynamic constraint metaballs, we first transform it into the local coordinate system and obtain $P'$, then multiply $P'$ with transformation matrix $N'$, where $N' = T(d'_x, d'_y, d'_z)R(\theta'_x, \theta'_y, \theta'_z)$, $(d'_x, d'_y, d'_z) = F(d(P, C_i), R_i)(d_x, d_y, d_z)$ and $(\theta'_x, \theta'_y, \theta'_z) = F(d(P, C_i), R_i)(\theta_x, \theta_y, \theta_z)$.

### Experimental Results

The interactive haptic-constraint modeling based on metaballs has been developed on Dell workstation PWS420 with single Pentium III 733 MHz Processor, and the haptic feedback device is PHANToM Desktop with 6DOF input and 3DOF force feedback. The operation system is WindowsNT 4.0 Server. We use OpenGL 1.3 for graphics rendering and GHOST SDK 3.0 library for force calculation. Visual C++ 6.0 is the programming language used for the system. Haptic-constraint tools, including point/line, circle/disk and sphere/cube volume constraints (see Appendix for detail), are experimented in the following.

### Force Evaluations

Figure 5 records the force evaluation during haptic-constraint manipulation with point-picking tool when interacting with star object of 8460 triangles, where the stiffness coefficient $k_s$ was 5 and $k_d$ was 1. The force in evaluation was the force computed before reflecting to the haptic device at the sampling rate of 1 kHz. To evaluate the interpolation filter method for haptic frequency extension, a linear interpolation function was also implemented in the experiment. In Figure 5a–c, the original haptic signal was ladder-shaped, which was very unnatural; the linear interpolation output exhibited a vibration behavior; the interpolation output was smooth while original shape was preserved well. High frequency noises might occasionally occur due to arbitrary operation or complex model shape, so we also

![Figure 4. Haptic-constraint manipulation: (a) target circle constraint; (b) haptic constraint manipulation at time t; (c) haptic constraint manipulation at time t+1.](Image)
compared the effects of noise elimination by adding 1 kHz noises into original signal in Figure 5d–f. The result indicated that both methods eliminated high frequency noises. But the vibration behavior of linear interpolation was obviously, most interpolation methods faced the same problem as linear interpolation did. Interpolation filer method performed better because it was deduced based on the shape estimation of input signal.

**Haptic-Constraint Deformations**

Following describes the experiments of interactive haptic modeling based on metaball constraints. In comparison with graphics constraints often show the result of final constrained deformations, the proposed haptics constraint interactions can help users control and feel the intermediate state with touch sensations during interactive process. Figure 6 shows the original wire-frame models: star, dragon, cow and teapot objects used in our tests. Figure 7 presents the deformed star object (8640 triangles), using haptic-constraint tools (e.g., disk, line, point) shown in green in the figure. Constraint force distribution and accordingly deforming surface are shown in different color from red to blue, where red corresponds to the highest force and largest deformation while blue refers to the unchanged part. From left to right: the disk constraint with 2 mm radius was attached at the upper right horn of star; the line constraint with 5 mm line length was specified to drag the left horn; the point constraint was guided to drag and wave the upper right horn in the last two snapshots. Figure 8 shows the haptic deformation process through applying interactive sphere-volume constraint (2 mm radius) on the dragon object (10 920 triangles). During the interaction, intermediate states of the dragon head’s shaking and rising up via applying the haptic sphere-volume constraint are shown from left to right respectively.

Figure 5. Force evaluation: (a–c) without noises and (e–f) with noise.

Figure 6. Original wire-frame models (from left to right): star (8640 tris), dragon (10 920 tris), cow (5804 tris), and teapot (3751 tris).
Figure 7. Haptic-constraint modeling of star object (from left to right): disk constraint (2 mm radius), line constraint (5 mm length), and point constraints.

Figure 8. Haptic-constraint modeling on dragon head’s shaking and rising up from left to right via sphere-volume constraint with 2 mm radius.

Figure 9. Haptic-constraint modeling of cow (from left to right): close snapshot of cow head with disk constraint (3 mm radius), same disk constraint pushing head below and lifting upper left, and sphere-volume constraint (1 mm radius) to lift horns.

Figure 9 shows the haptic constraint modeling of cow object (5804 triangles), using disk-picking constraint with 3 mm disk radius to drag the head of cow downside and left. To the right, the sphere-volume constraints with 1 mm radius were attached to the forehead of the cow, to reach and lift the head. More haptic-constraint deformations were tested on teapot object (3751 triangles) shown in Figure 10. In the upper row, there are some snapshots during the interactive manipulation with haptic-constraint tool of circle (2 mm radius) applied on the lid of teapot. Haptic constraints (e.g., point, sphere-volume) were applied on the mouth and lid’s handle of the teapot in the lower row. Multiple haptic-constraint tools can do help in designing customer styled teapots such as distorted mouth, shaped lid, curved handle and twisted body. Interactive soft-touch modeling with real-time force feedbacks certainly facilitates the designers sensational positioning and fine deforming on the constrained local areas, in a more intuitive and realistic approach.

**Computational Performance**

The computational performance of our interactive haptic-constraint modeling based on metaballs is summarized in Figure 11. Four objects, teapot (2022 vertices/3751 triangles), cow (3066 vertices/5804 triangles), star (4322 vertices/8640 triangles), and dragon (5352 vertices/10 920 triangles) were tested with different data resolutions. All objects were transformed into (~10 mm, 10 mm) scale in the experiment. The control parameters of haptic-constraint tools were: line-picking constraint with the length of 2 mm; circle-picking
constraint, disk-picking constraint and sphere-volume constraint with the radius of 2 mm; cube-volume constraint with the edge length 4 mm. Figure 11 records the running time and average influenced number of vertices during the interactive haptic-constraint modeling. The effective radius with 2 and 10 mm were set and tested for comparisons, in the upper and lower rows respectively. From the figure, we observe that the time spent is mainly affected by the scale of haptic-constraint tools, effective radius of metaballs and the data resolution of the objects. High working rates (above 1 kHz) can be guarantee with lower effective radius of metaballs (2 mm) of all haptic constraint tools. Under higher effective radius (10 mm, almost bound entire
object), larger influenced areas were involved, which result relative lower working rate (around 500 Hz). Our experimental results show that the proposed haptic-constraint modeling realize the high working rate of haptic interaction, while providing the flexible control of localized deformations and intuitive soft-touch prototyping.

Discussions

The assessment of our work is to add interactive haptic constraint tools with stable force sensation in the haptic interface. Making direct comparison of haptic interface design with traditional interfaces is not so intuitively since the manipulations through mouse and haptic device have large difference. For basic operations easily expressed by regular shape/feature operations, traditional interface is suitable and sufficient. Studies have shown that as for freeform based surface operations, haptics does provide a powerful way for designing and manipulating complex surfaces. With traditional two-dimensional interfaces, interactively designing an object is often an inefficient and tedious process. Many times, designers want to touch the target and modify it by pushing, pulling or dragging as if in a real world. While with traditional interfaces, such operations usually require complex mouse manipulations and multiple view adjustments. Instead of requiring above complex adjustments, haptic interface allows designers to interact with models via their sense of touch in three-dimensional space directly, and thus provides more flexibility for the manipulation of complex object. To improve performance of haptic design interface, additional interactively haptic constraint manipulation method is integrated here. Haptic constraint feedback help designers to touch, grasp, and manipulate objects in much easier way. Stable and constant force feedbacks during the operations let users deform the target accurately; otherwise, the target can be deformed excessively without force feedback constraints. Metaball deformations of implicit surface modeling with touch sensation providing flexible solutions for creating blends, ramifications and advanced human character design more conveniently.

Summary

In this paper, we present a novel haptic-constraint modeling approach based on interactive metaballs. Our work is one of the first to simulate the dynamic touch sensation for such constrained deformations using interactive metaballs. Here, touch sensation is applied intuitively while manipulating the objects in the constrained regions, so the users can truly feel the deforming process during the haptic interaction. We develop the haptic-constraint tools interacting in the localized regions based on specified metaballs, in which interactive forces/torques are simulated to enhance the realistic manipulation of soft objects in touch-enabled virtual environments. Simulating the dynamic touch sense with interactive haptics modeling supports the deformable manipulations more intuitive and flexible to control. Our approach provides novel haptic-constraint modeling tools for 3D shape design and soft-touch interaction, and can be easily incorporated into existing modeling/animation systems. We will further work on the haptic tools constrained along the path, and extend the force model to guide the haptic tool following the path with the constraints. The other work is to address the user behaviors, such as probe’s working line or manipulation style, to make the system more stable and user friendly.

ACKNOWLEDGEMENTS

This work was supported by NSFC grants (nos. 60703120 and 60933007), RGC research grants (ref. 415806 and 416007), UGC direct grant for research (no. 2050423) and NSFC-MSRA joint funding grant (no. 60970159).

Appendix: Haptic-Constraint Tools

Interactive haptic-constraint tools are defined with different constraint skeletons. As for different constraint skeleton shape, different distance function \(d(p, C_i)\) of the metaball needs to be specified in the haptic deformable modeling.

Point/Line Constraints

Constraint skeleton \(C_i\) is the point skeleton in point constraint. The distance function \(d(P, C_i)\) of pint constraint is the distance from point \(P\) on the object to dragged point \(C_i\) having \(d(P, C_i) = ||P - C_i|| = R_i\), shown in Figure A1a.
A line segment $C_i$ with length $l$ on the object is defined in line constraint, having two end points $P_0(x_0,y_0,z_0)$ and $P_1(x_1,y_1,z_1)$. The distance function $d(P, C_i)$ is the minimal distance from point $P$ on the object to the dragged line skeleton $C_i$, having $d(P, C_i) = \min\{||P - C_i||\} = \min\{||P - P_0||, ||P - P_1||\} = R_i$. Figure A2a is the result of applying a haptic-line constraint on a triangle plane mesh.

A circle $C_i$ with center $O_c$ and radius $R_c$ on the object is defined in circle constraint. The distance function of the generalized haptic metaball is calculated from point to the circle line directly, with $d(P, C_i) = d(P, O_c) = R_c$. Figure A2b is the result of applying a haptic-circle constraint on a triangle plane mesh.

A disk $C_i$ with center $O_c$ and radius $R_c$ on the object is defined in disk constraint. The distance function of generalized metaball $d(P, C_i)$ is composed of two parts. If the perpendicular point of $P$ lies within the disk, we have $d_1(P, C_i; \text{plane}) = R_c$. Otherwise, the distance is calculated from point $P$ to the circle line of the disk $C_i$, having $d_2(P, C_i; \text{circle}) = R_c$. Finally, the distance function is the union set of above two parts, $d(P, C_i) = d_1(P, C_i; \text{plane}) \cup d_2(P, C_i; \text{circle})$. Figure A2b is the distribution of applying a haptic-disk constraint on a plane.

Sphere/Cube Volume Constraints

A sphere volume $C_i$ with center $O_c$ and radius $R_c$ on the object is defined in sphere volume constraint. The distance function of the generalized haptic metaball is calculated from point to the center of sphere volume directly, with $d(P, C_i) = R_c$. The outer surface of generalized sphere volume is displayed in Figure A3a.

For cube-volume constraint, a cube skeleton $C_i$ with center $O_c$ and edge length $2a$ on the object is defined. The distance function of the generalized haptic metaball is calculated from point to the six faces of the cube, $d_1(P, C_i; \text{face}) = R_c$; to the twelve edges, $d_2(P, C_i; \text{edge}) = R_c$; and to the eight vertices of the cube, $d_3(P, C_i; \text{vertex}) = R_c$. Finally, we have $d(P, C_i) = d_1(P, C_i; \text{face}) \cup d_2(P, C_i; \text{edge}) \cup d_3(P, C_i; \text{vertex})$. Figure A3b shows the shape of a generalized cube volume.

References


Authors’ biographies:

**Hui Chen** received the B.S. and M.S. degrees in computer science from Shandong University, China, and received the Ph.D. degree in computer science from the Chinese University of Hong Kong, Hong Kong. She is currently an assistant professor in the Centre for Human-Computer Interaction, Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Sciences/The Chinese University of Hong Kong, SIAT, China. Her research interests include haptics, virtual reality, computer-assisted surgery, and computer graphics.

**Hanqiu Sun** received her B.S. degree in electrical engineering from HUST, China, M.S. in electrical engineering from University of British Columbia, and Ph.D. in computer science from University of Alberta, Canada. She is currently an associate professor at Department of Computer Science and Engineering, The Chinese University of Hong Kong. Her research interests include virtual & augmented reality, interactive graphics/animation, hypermedia, computer-assisted surgery, mobile image/video processing, telemedicine, and realistic touch-enabled simulations.

**Xiaogang Jin** is a professor of the State Key Lab of CAD&CG, Zhejiang University, China. He received his B.Sc. degree in computer science in 1989, M.Sc. and Ph.D. degrees in applied mathematics in 1992 and 1995, all from Zhejiang University. His current research interests include implicit surface computing, special effects simulation, mesh fusion, texture synthesis, crowd animation, cloth animation, and non-photorealistic rendering.