Interactive soft-touch dynamic deformations

By Hui Chen*, Hanqiu Sun and Xiaogang Jin

It is crucial for the users to touch, grasp and manipulate the interested objects through our sense of touch in many interactive applications, such as on-line computer games, interactive cartoon design, and virtual prototyping. In this paper, we propose an interactive haptic deformation approach which incorporates the dynamic simulation of mass–spring systems and flexible control of free-form deformation in the touch-enabled soft-object deformation. Through distributing mass, spring and damping coefficients of the object to the bounded Bezier volume lattice, the deformation of the object related to the haptic avatar follows the physical laws and has high working rate. Both homogenous and inhomogenous materials are simulated. The anchor nodes of haptic input are specified to create amazing special effects during the interactive haptic deformation. Interactive haptic deformations of three-type tropic fishes, Angel, Demekin, and GuppyBlueGrass, have been experimented to simulate vivid fish swimming processes in the virtual ocean scene. Our proposed approach provides touch-enabled input and efficient performance in the flexible deforming controls, letting the objects move in a dynamic, cartoon-style deforming manner. Copyright © 2007 John Wiley & Sons, Ltd.

Received: 15 February 2007; Accepted: 19 February 2007

KEY WORDS: haptics interface; interactive modelling; free-form deformation

Introduction

Haptics1,2 investigates the touch-based interface which supports force/tactile feedbacks to the users. It is an active research topic to improve the reality of immersive virtual environments. Haptic interface enables the users to directly perform physical interaction with the computer-generated objects that other modalities cannot supply, such as stiffness, vibration, pressure, and temperature. One important use of haptics applies in the entertainment business, where haptics has been used to give the players force feedback during the events occurring in interactive computer games. Physics simulation3 with force feedback enabled the players immersive themselves in high-end games is the next frontier in the game development. In 1990s, the desktop force feedback joysticks and wheels were introduced for interactive games and digital entertainment. These devices work well in simulating the recoil of a gun in shooting games or the vibration of a steering wheel in racing vehicles. But these uses are limited in the degrees of freedom and sensitivity, making them hard to be applied for the more advanced game applications.

With the recent development of haptics techniques, implementing a realistic physics engine within interactive games is possible. Lander simulated bouncy and trouncy response of particles under Newton’s dynamics4 and also mimicked crush effect of large massive objects via matrix deformation5 in game development. However, the system did not reflect the realistic force feedback to the players besides deformation in game engine. The game development with touch-based sensation information via PHANToM6 was set up in an initial stage, handling dynamic objects only constructed from boxes, spheres and cylinders instead from complex meshes. Making the combination for simulating force feedback to the users and force-reflecting deformable objects in virtual environments is not trivial. Physically based deformation methods

*Correspondence to: H. Chen, Shenzhen Institute of Advanced Integration Technology, Chinese Academy of Sciences/The Chinese University of Hong Kong, Shenzhen, PR China. E-mail: hui.chen@siat.ac.cn
generally require intensive computation, thus not suitable for interactive haptics modeling applications in real time. Furthermore, the demand for realistic haptic feedbacks at high refresh rate leads to the additional computation complexity. On the other hand, integrating non-physical methods with haptic feedback is not natural and cannot provide realistic physical simulation of soft-object deformation.

During the interactive entertainment, the main requirements of providing force feedback to the players are the following: First is providing the real-time force feedback. The players expect to feel what they interact with in the virtual scene instantly, independent of the complexity of the objects. Second is imitating dramatic deforming effects. Rather than the precise physical simulations in the virtual surgical environment, haptics feedback in interactive gaming is mainly a method to mimic the exaggerated fanciful animations to the players enhancing attraction and impression of the system. Motivated by the realistic simulation of the dynamic mass–spring models and the effective control of free-form deformation techniques in interactive modeling, we propose an interactive haptic deformable modeling approach that incorporates the dynamic simulation of mass–spring systems and the flexible control of free-form deformation in the touch-enabled modeling of soft-object deformation. Our approach has the following features to meet the above requirements.

- Intermediate mass–spring Bezier volume space

Instead of utilizing the common mass–spring model which connects the mass points and springs on the object itself, an intermediate mass–spring Bezier volume space is constructed through attaching a basic mass–spring lattice to the bounding box of the object. Besides the springs along the edges, the springs on the face diagonals and the inside body diagonals are also added to avoid the collapse fault during the haptic interaction. Force feedback simulated via this intermediate space can always guarantee high refreshing rate, and not sensitive to the complexity of the objects manipulated.

- Interactive haptic control of deformable animations

The interactive haptic animations of the soft object keep the merits of the flexible control in the free-form deformation techniques. The touch-enabled dynamic deformation is governed via the most intuitive 3D haptic interface besides relative parameters adjustment applied in CAD/CAGD. Homogenous and inhomogenous material properties and the interactive anchor nodes are developed to mimic the special effects during the soft-touch deformation.

In the following, Related Work section outlines the previous related work. Soft-touch Dynamic Deformations section gives the overview of our interactive haptics system for the dynamic soft-object deformation we propose. Physically Based Bezier Volume Lattice section describes how to construct physical Bezier volume lattice of the object, and Dynamic Soft-object Deformation section presents the dynamic soft-object deformation process and the interactive haptic responses to pull/push. System Behavior and Performance section shows the experimental results of the system behavior and the performance in the dynamic soft-touch deformations. Finally, the summary goes to Summary section.

## Related Work

In this section we briefly overview the previous related work in haptics rendering, deformable modeling in computer graphics, and haptics rendering of deformable objects.

### Haptics rendering

Haptic rendering\(^7,8\) 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 three groups, point-based methods, ray-based methods, and object-based methods. In point-based methods, only the end of the haptic input interacts with the virtual objects. The haptic rendering methods through manipulating point avatar include penalty-based methods,\(^7\) in which the force generated is proportional to the amount of the penetration into a virtual object, and constraint-based methods,\(^9,10\) in which the force computed is proportional to the displacement between the haptic interaction point (HIP) and the surface contact point to indicate where the HIP is located on the surface if the haptic interface could not penetrate the surface.

In ray-based methods, the generic probe of the haptic input is modeled as a line-segment whose orientation is taken into account, and the collisions are checked between the finite line and the objects. Multiple objects can be touched simultaneously through manipulating the line-segment avatar; moreover, torques in addition...
Deformable modeling techniques in computer graphics have been extensively studied. Approaches for modeling object deformation include geometry-based methods and physics-based methods. In geometry-based methods, individual or groups of control points or shape parameters are manually adjusted for shape editing and design. Free-form deformation (FFD) is an efficient representation of geometry-based methods for animating soft object via a structural hyperpatch first introduced by Sederberg and Parry. The object to be deformed is embedded into an intermediate flexible space. Through deforming the shape of the intermediate space, the object within the space is following updated to match the change. Techniques based on 3D tensor-product Bezier volume, B-spline volumes, rational Bezier volumes, or NURBS volumes were also proposed. The method of parameterized hierarchical FFDs augmented with Lagrangian dynamics is proposed by Faloutsos et al. to animate and control the simulated characters efficiently.

Physics-based methods incorporate physical principles for realistic simulation of complex physical processes. Once the physical attributes of the object are specified, including material properties, external forces, and environment constraints, the approach automatically produces the deformation and the motion of the object by solving the complex differential equations. Mass-spring system can give the illusion of physical behavior that has been used widely and effectively for modeling deformable objects. Here an object is modeled as a collection of point masses connected by springs in a lattice structure. The equations of the motion for the entire system are assembled from the motions of the mass points in the lattice. Finite element method (FEM) is a really physically based technique based on the discretization of the continuum media equations. In FEM, the object is divided into a set of elements jointed at the discrete nodal points, and the interpolation functions that solve the equilibrium equations are found in each element. The continuity between the elements is achieved by the interpolation functions and by the fact that nodes are common to several elements.

Haptics Rendering of Deformable Objects

Haptic simulation of deformable objects needs to estimate not only the deformation of each object node in the space, but also the magnitude and direction of the interaction forces that are reflected to the user via a haptic device. Similar with deformable modeling methods in graphical display, the techniques rendering force-reflecting deformable objects are basically grouped into geometry-based and physics-based. A force model generating the interaction forces is loosely coupled into geometry-based methods. The extension of geometry-based techniques to haptic display of deformable objects has applications in haptic sculpting, and CAD system. In physics-based methods, the computation of the interaction forces is the part of physics-based models, so we do not need a separate model for forces generation. McDonnell et al. developed a haptic sculpting system based on a subdivision solid and mass-spring modeling, where mass-spring modeling is used to reduce the complexity of manipulating geometric elements and the force is given to the subdivision solid. Dachille et al. established a similar system through dynamic NURBS (D-NURBS) to combine NURBS and mass-spring modeling. James and Pai, and Basdogan had made modeling simplifications due to the limited computational power to implement real-time FEM with haptic displays. Adding force feedback in deformable modeling has advantages to increase intuition, to control deformations, and to support the development of physical constraints.
Soft-touch Dynamic Deformations

Figure 1 outlines the interactive system framework of the soft-touch dynamic deformation method we propose. The objects to be deformed in the virtual environment are bounded with the appropriate Bezier volume, and the properties such as mass, spring and damping coefficients are assigned to the Bezier volume to construct a mass–spring Bezier volume lattice to control the objects to be deformed or bounced accordingly. Users can touch and manipulate the soft objects via the unified graphics and haptics representation of dynamic Bezier volume lattice. Through distributing mass, spring and damping coefficients of the object to the bounded Bezier volume lattice, the deformation of the object related to the haptic input both follows the physical laws and has high working rate. Our haptics-based modeling approach can be easily integrated into the computer games to simulate force feedback of the avatars in dragging, hitting and crashing motions, letting the objects move in a dynamic, cartoon-style soft-touch manner.

Physically Based Bezier Volume Lattice

The first step in our interactive soft-touch dynamic deformation system is to construct the intermediate physically based Bezier volume space of the object. A tensor-product trivariate Bernstein polynomial, a Bezier volume, is chosen as the underlying intermediate parametric space to construct the mass–spring Bezier volume lattice for the object representation and the further haptics-based interactive modeling. The interactive haptic manipulation on the control points in the lattice determines the shape of the intermediate space and the resulted object deformation.

Bounding Bezier Volume

The generic tensor-product trivariate Bernstein polynomial functions are of the following form:

\[ Q(u, v, w) = \sum_{i=0}^{3} \sum_{j=0}^{3} \sum_{k=0}^{3} P_{ijk} B^3_i(u) B^3_j(v) B^3_k(w) \]  

where \((u, v, w)\) represents the 3D position within the Bezier volume, having \(0 \leq u, v, w \leq 1\); \(P_{ijk}\) are control points; \(B^3_i(u)\), \(B^3_j(v)\), and \(B^3_k(w)\) are all Bernstein basis functions, taking the form,

\[ B^3_i(u) = \frac{3!}{i!(3-i)!} u^i (1-u)^{3-i} \]

The initial Bezier volume structure for the object utilized in our haptics-based deformable modeling system is formed into a cubic shape, and the control points are set to form a \(4 \times 4 \times 4\) regular coordinate grid, shown in Figure 2(b). To attach a Bezier volume to an object, the resting state of Bezier volume block would

Figure 1. System framework of soft-touch dynamic deformations.
match the bounding box of the object, supposing the bounding box of the object ranges from \((x_{\text{min}}, y_{\text{min}}, z_{\text{min}})\) to \((x_{\text{max}}, y_{\text{max}}, z_{\text{max}})\) in the object space. Once the bounding Bezier volume is defined around the object, each vertex in the object must be converted from its current \((x, y, z)\) coordinate in the object to a \((u, v, w)\) lattice position, which can be calculated through normalizing all the coordinates in the object to line in the range from 0 to 1.

**Mass–spring Bezier Volume Lattice**

Physical properties of the object are assigned to the bounding Bezier volume of the object to construct the intermediate physical Bezier volume space of the object for further deformable interactions. Instead of utilizing the mass–spring model which commonly connects the mass points and springs on the object, a mass–spring system is attached to the bounding Bezier volume of the object to simulate dynamic free-form deformations.

The mass of each control point is first assigned according to the mass distribution on the object. The springs connected control points aligned with the lattice are then connected, shown as blue-springs in Figure 2(c). As the lattice connected only via blue-springs may collapse in the following deformation, it is necessary to put crossbeam supports shown in pink springs on the lattice. Figure 2(a) is a simple example of 3D mass–spring lattice for one cube unit, above is the original cube, and in lower it comprises 8 point masses connected by 28 damped linear springs: 12 run along the edges of the cube, 12 traverse face diagonals, and the remaining 4 lie along the body diagonals. Totally 27 cubes are involved and 504 springs are set for constructing mass–spring Bezier volume lattice of each object (see Figure 2(c)): 180 parallel to the edges of the cube, 216 traverse face diagonals, and 108 lie along the body diagonals within the object.

**Dynamic Soft-object Deformation**

The haptic modeling of dynamic soft-object deformation marries the free-form deformation with physical attributes and other relevant material quantities enabling extra flexibility and advantages in interactive deformation modeling. In the following we present how to apply physically based deformation through mass–spring Bezier volume lattice of the object, how to handle the interactive haptic response, and how to accelerate the deformation on the soft-objects.

**Physically Deformation of Mass–Spring Bezier Volume Lattice**

The physical Bezier volume lattice is constructed through masses and connected springs, the dynamics of control points is governed by the Newton’s Second Law of motion. The displacement of the \(i\)th control point \(u_i \in \mathbb{R}^3\) due to an external force \(F_i\) is given as follows,

\[
m_i \ddot{u}_i + d_i \dot{u}_i + \sum_j k_{ij}(|r_{ij}| - l_{ij})/|r_{ij}| r_{ij} = F_i
\]

where \(m_i\) is the mass of the point \(i\), \(d_i\) the damping constant of the same point, \(r_{ij}\) the vector distance between point \(i\) and point \(j\), \(l_{ij}\) and \(k_{ij}\) are the rest length and stiffness of the spring connecting two mass points, respectively. The right-hand term \(F_i\) is the sum of other external forces (e.g. gravity or other user applied forces).

The motion equations for the entire system are assembled from the motions of the mass points in the Bezier volume lattice. Concatenating the position vectors of the \(N\) individual masses into a single \(3N\)-dimensional position vector \(U\), then the Lagrange’s dynamics equation is satisfied,

\[
M \ddot{U} + D \dot{U} + Ku = F
\]

where \(M, D\) and \(K\) are the \(3N \times 3N\) mass, damping and stiffness matrices, respectively. \(M\) and \(D\) are diagonal matrices and \(K\) is banded because it encodes spring forces which are the functions of distances between neighboring mass points only. The vector \(F\) is a \(3N\)-dimensional vector representing the total external forces on the mass control points. To solve the Lagrange’s dynamics Eq. (4), we first replace all derivatives with their discretized approximations, central difference, and then use the iteration method to solve the resulting time-varying algebraic system.

---

**Figure 2.** Mass–spring Bezier volume lattice: hidden masses and springs are removed for the clarity of illustration.
Interactive Haptic Response

During the deformation or movement of the soft object, real-time haptic interaction needs to be modeled in response to the object accordingly. To simplify the process, the simulation of point-to-plane model is applied here in interactive haptic response.

Following is the case when a particle $U$ with a velocity vector $\vec{V}$ is moving towards plane $P$ with a normal $\vec{N}$, thus a haptic-collision may happen. Three cases have to be considered:

i. If $(U - P) \cdot \vec{N} < \varepsilon$, a collision occurs, where $\varepsilon$ is a very small threshold.

ii. If $(U - P) \cdot \vec{N} \approx 0$, a contact occurs. The relative velocity of the two bodies needs to be checked to judge whether the particle is moving towards or away from the boundary. If $\vec{N} \cdot \vec{V} < 0$, then a haptic collision takes place.

iii. If $(U - P) \cdot \vec{N} < -\varepsilon$, a particle has passed through the boundary and penetrated into it. To avoid penetrating, it is necessary to move back the simulator a bit for the further check up.

If a collision occurs in the first two cases, the haptic response needs to be computed, with two vectors representing the motion parallel and tangential to the normal of the collision.

$$\vec{V}_n = (\vec{N} \cdot \vec{V}) \vec{N}, \quad \vec{V}_t = \vec{V} - \vec{V}_n, \quad \vec{V}' = \vec{V}_t - K_r \vec{V}_n \quad (5)$$

where $\vec{V}_n$ is the amount of the normal force applied to the resulting force, $\vec{V}_t$ the parallel one, $\vec{V}'$ the velocity of the particle after collision, and $K_r$ is the coefficient of restitution. If $K_r$ is 1, an elastic response is acquired; if it is 0, the particle sticks to the plane. Multi-collisions are treated similarly through the mechanical coupling of an implicit integration progress.

Deformation of Soft objects

In order to accordingly deform the object within the intermediate mass–spring Bezier volume lattice, each vertex in the object must be passed through trivariate Bernstein polynomial function, see Eq. (1), to evaluate its deformed position. The Bernstein basis functions, taken the form in Eq. (2), serve as the role of relating the control points in the mass–spring Bezier volume lattice to the vertices in the object. The scale value between 0 and 1 of $(u, v, w)$ of each vertex in the object is calculated in advance, and plugged into the basis functions for each control point. The out pops a point weight that relates the vertex in the object to that control point. The sum of all the 64 point weights on any vertex is equal to 1 due to the nature of the Bezier basis functions, and the sum of the influences at any point along the curve is always equal to 1. Once the weight values are calculated for each vertex in the object, the object is deformed accordingly to the movement of the mass–spring Bezier volume lattice, through physically based deformation of the control points.

System Behavior and Performance

The interactive haptic deformation system has been developed on Dell workstation PWS420 with single Pentium III 733 MHz Processor, and the haptic feedback device is PHANToM Desktop with 6DOFs input and 3DOFs 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 for the system development.

Haptic Deformation

External forces must be applied from the haptic avatar to directly manipulate the object in the virtual scene. Besides gravity forces applied to particles putting down the object, drag forces are used for making the object floating around. The drag force is simulated with Hook’s law via a spring tied to the particle and haptic avatar, having the form $F = K_d d + K_v v$. The second part of the formula is viscous drag through multiplying a damping constant with the velocity to prevent too much bounce and add stability to the system. Two drag forces can be applied at the same time to simulate the two hands’ motion of the player.

First, the objects are experimented with homogeneous and isotropic materials. The homogeneous lattice is composed of the nodes with the same mass and springs of the same stiffness. Anisotropy of the model has been demonstrated by setting different stiffness to springs in the vertical and horizontal directions. Gravity is assigned additionally to help the object stick to the ground for the clarity in Figure 3(a–c). In Figure 3(a), a transversely isotropic lattice with springs along y-axis 10 times smaller than that along x- and z-axis is composed. The transversely isotropic material experiences smaller deformation and is stretched less outwards.
under the same pulling force. In Figure 3(b), the simulation of haptic deformation is performed through applying a drag force upward on the same surface node. Effects on different materials have been studied through adjusting the spring stiffness. Figure 3(c) records the scenario when forces are applied to two surface nodes at the same time. Besides the larger deformation on the node(s), the whole object starts to deform while moving in the virtual scene. Removing the gravity, with drag force only, the object begins to deform and float in the virtual scene (see Figure 3(d–f)), each with the same settings for the corresponding upper one. The results suggest that the physical model can be applied for interesting soft-touch applications through virtual hand or multi-user interactions.

Anchor nodes are set to the infinite mass in the model to create the special-effect simulation in haptic deformation shown in Figure 4. The spring and damper coefficient of the mass–spring Bezier volume lattice are 80.0 and 0.9, respectively. The mass of anchor nodes are 10 000 times larger than the other nodes, 2000 g of anchor nodes versus 0.2 g to the common nodes. In Figure 4(b), four anchor nodes are specified on the left corner of the object within red ellipse to simulate a rotational axis during the haptic deformation. When one anchor node shown in red circle is specified in Figure 4(c), more stretched deformation of the whole star is acquired in comparison with Figure 4(b). Special effects of the soft-object deformation can be interactively simulated flexibly, with the assistance of the specified anchor node(s).

Following, our interactive system has been applied to simulate the haptic deformations in interactive cartoon-style modeling. Figure 5 shows the surface model of a jiggly pixy. The spring and damper coefficient of the mass-spring Bezier volume lattice are 80.0 and 0.9, respectively. The mass of anchor nodes are 20 000 times larger than the other nodes, 4000 g of anchor nodes versus 0.2 g to the common nodes. In the upper row, eight fixed anchor nodes are set on the lower right foot within the red ellipse in the left subfigure. The haptic drag force is applied on the opposite left foot to simulate the upside kicking process from left to right. The larger force has been perceived while kicking the foot higher through the haptic input. In the lower row, the snapshots of more vividly haptic interaction are recorded with different anchor nodes set to the left foot, right ear, and middle forehead respectively. More
global deforming effects of tropical fishes are interacted haptically in our system. Figure 6 shows the original wire-frame models of three tropic fishes: Angel, Demekin, and GuppyBlueGrass. The haptic deformation processes are recorded in Figure 7. The simulations are specified on either fish head or fish tail. The vividly cartoon-style swimming motions can be interactively created in deforming sequence. Real-time force feedbacks are replied to the user during the haptic input, in relation to the interacting process.

**Computational Efficiency**

The time performance of the interactive haptic deformable modeling via intermediate mass-spring Bezier volume space is recorded in Chart 1 and Table 1. For comparison, the global mass-spring deformation method of the whole object and the further local method to solve the Newton’s dynamic system within the specified local range of the user input are tested and recorded. In global mass-spring deformation, all the vertices of the whole object are involved in dynamic Eq. (4), resulting in the very high computational cost. In local mass-spring deformation, the order of dynamic Eq. (4) is reduced largely into a low-order equation through fixing the vertices far from the acting forces. Thus, the computational rate gains great improvement. In our approach, the mass-spring Bezier lattice is applied in the experiments. The time table shows that although with the large increment of data resolution, the increment of the computational time of our approach is relatively low. It is because that the most expensive step of simulating deformations under Newton’s system during physics-based haptic interactions is restricted by control lattice in our approach, and the weights governing the vertices on the object are generated before deformation. Our control-lattice model has the significant improvement of the deformation rate, making the haptic interactive deformations of the large object applicable.

**Summary**

In this paper, we have developed the interactive soft-touch dynamic deformation approach via physical Bezier volume space. The proposed haptic deformable approach incorporates the physical realism of the
mass–spring systems and the flexible control of free-form deformation technique in interactive haptics modeling. Through distributing physical properties including mass, spring and damping coefficients of the object to the bounded Bezier volume lattice, the deformation of the object related to the haptic input follows the physical laws and acquires high deformable working rate. It can be efficiently coupled into the interactive game development to augment force feedback of the avatars when dragging, clashing or exploring

Figure 7. Haptic deformation of tropical fishes: from up to down are Angle, Demekin, and GuppyBlueGrass. The simulations are specified on either fish head or fish tail.

<table>
<thead>
<tr>
<th>Number of triangles</th>
<th>Mass-spring Bezier lattice(s)</th>
<th>Local mass-spring deformation(s)</th>
<th>Global mass-spring deformation(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>208</td>
<td>0.062</td>
<td>0.05</td>
<td>0.08</td>
</tr>
<tr>
<td>832</td>
<td>0.062</td>
<td>0.11</td>
<td>3.65</td>
</tr>
<tr>
<td>2160</td>
<td>0.063</td>
<td>0.28</td>
<td>117.76</td>
</tr>
<tr>
<td>8640</td>
<td>0.065</td>
<td>0.49</td>
<td>840.54</td>
</tr>
<tr>
<td>11660</td>
<td>0.068</td>
<td>0.78</td>
<td>1632.45</td>
</tr>
<tr>
<td>24928</td>
<td>0.076</td>
<td>2.13</td>
<td>3487.67</td>
</tr>
</tbody>
</table>

Chart 1 & Table 1. Running time of deformation interaction(s)
in the virtual gaming scene. Through distributing different physical properties of mass and spring coefficients, the special effects can be created during the touch-enabled interactions, letting the objects move in a dynamic, cartoon-style deforming manner.

With the embedded free-form deformations, our approach is somehow limited to the coarse representation of the interested object to be deformed through manipulating the bounded volume. Careful adjusting the intermediate Bezier volume space can improve the limit. For instance, similar to the anchor nodes used in the system, void nodes are needed to represent the hole or air region within the object or more closely approximated object space. Unlike the anchor nodes are set through increasing the mass of the control points, decreasing the mass of void nodes can result in frequently unstable animation and let the object very hard to calm down. The tradeoff is to remove the void nodes and the springs attached to, thus the topology of the intermediate Bezier volume space may be changed accordingly. The alternative is to utilizing the subspace technique to build a more accurate bounding shape as the intermediate control space around the original object. The variable size of such intermediate control space in response to the haptic interaction needs further investigation for the smooth deformation and flexible control of the object. We will continue our work on developing real-time haptics techniques that can be flexibly coupled in the interactive game development and digital entertainment applications.

ACKNOWLEDGEMENTS

The work was supported by UGC direct grant for research (no. 2050349), National Fundamental Research Grant of Science and Technology (973 Project: 2002CB312104), and Natural Science Foundation of Zhejiang Province (Grant No. R105431).

References

3. Hecker C. Physics, the next frontier. Game Developers Magazine, 12–20, October/November 1996.
8. Avila RS. Haptics: From Basic Principles to Advanced Applications SIGGRAPH'99; Course Notes #38.
Authors' biographies:

Hui Chen received the B.S. and M.S. degrees in Computer Science from Shandong University, P.R. 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, CAS/CUHK, SIAT, P.R. China. Her research interests include haptics, virtual reality, computer-assisted surgery, and computer graphics.

Hanqiu Sun received the Ph.D. degree in Computer Science from the University of Alberta, Canada. She is currently an Associate Professor with the Department of Computer Science and Engineering, Chinese University of Hong Kong. Her research interests include interactive behavior animation, virtual reality, hypermedia, computer-assisted surgery, internet-based visualization and navigation, telemedicine, and robotics simulation. Contact her at hanqiu@cse.cuhk.edu.hk.

Xiaogang Jin received the B.Sc. degree in Computer Science, and M.Sc. and Ph.D. degrees in Applied Mathematics, all from Zhejiang University, P.R. China. He is currently a Professor at the State Key Laboratory of CAD & CG at Zhejiang University, People's Republic of China. His research interests include implicit surface computing, cloth animation, crowd and group animation, computer animation, and digital geometry processing. Contact him at jin@cad.zju.edu.cn.