A Hybrid Physical Deformation Modeling for Laparoscopic Surgery Simulation

腹腔鏡手術模擬系統中的混合式物理變形模塑

Din-Chang Tseng (曾定章)* Jun-You Lin (林俊裕) Chih-Hung Lee (李志宏)
Institute of Computer Science and Information Engineering
National Central Universit
Chung-li, Taiwan
* E-mail: tsengdc@ip.csie.ncu.edu.tw

Abstract

A hybrid physical -based deformation modeling (HPDM) technique for our laparoscopic surgery simulation system is proposed. The proposed technique consists of three components: (i) approximate continuum free -form deformation modeling, (ii) efficient collision detection for non-rigid objects, and (iii) mass-spring modeling for force feedback implementation. Three physical -based deformation modeling techniques have been applied on th surgical simulation which are mass -spring model, approximate continuum model, and finite element model. All these models have disadvantages of less realism or time consuming. Here, we propose the HPDM to achieve the realistic deformation of organs in a computationally efficient framework. The proposed HPDM exhibits more physical accuracy and realism than the mass -spring modeling exhibits and exposes more advantages of more efficient computation, allowing topology change, and satisfying arbitrary geometric meshes than the approximate continuum model exposes.

Keywords: virtual reality, laparoscopic surgery simulation, physical modeling, force feedback, collision detection

1. Introduction

It is estimated that from 60 to 80 percent of abdominal surgeries will be performed laparoscopically in the year 2000 [2, 6]. This technology uses a small camera called a laparosocpe and customized surgical instruments for surgeries. They are inserted into the abdominal of patient through small skin incisions and performed the surgery. These instruments enable a surgeon to explore the internal abdominal on the *TV* and perform operations about surgery without large incisions. Laparoscopic surgery has many advantages over traditional abdominal surgery: (*i*) the operation is minimally invasive, (*ii*) incisions are small, and (*iii*) it is less traumatic. However, the laparoscopic surgical operations are more difficult than the traditional surgical operations, due to (*i*) the vision scope is limited by the field of the view of a laparosocpe, (*ii*) the actual hand movements and the image are reflected by a laparosocpe are not coherence, and (*iii*) a surgeon has to perform surgical operations using long surgical instruments. Therefore it is necessary to find new training approaches for the laparoscopic surgery.

Virtual reality is an immersed and interactive technology, it allows users to immerse and interact with objects in a virtual environment using virtual tools. The use of virtual reality (*VR*) technology gives users the feeling of participating in a scenario like reality. This fe eling is an essential requirement for simulation of surgery.

During the recent years, there has been growing interest in the medical and computer science fields around the minimally invasive surgery (*MIS*) simulation of *VR*. The *MIS* simulation system provides an efficient, safe, realistic, and relatively economical method for training clinicians in a variety of surgical tasks. The system not only decreases training costs, but also make s the reduction of surgical risk. However, it is difficult to develop such a surgical simulation system since not only the real-time interactive is needed but also the virtual organs should exhibit physicall correct behaviors corresponding to behaviors of the real human organs.

Many physical-based modeling have been proposed and applied on virtual human organs in a surgical simulation [1, 4-5, 9-11]. However, they have disadvantages of inaccurate deformation or complicate computation. Here, we propose a hybrid physical -based deformation modeling (*HPDM*) technique based on the approximate continuum deformation modeling proposed b Terzopoulos *et al.* [14] and the free-form geometric deformation modeling proposed by Sederberg *et al.* [12] for surgical simulation. The virtual human organs modeled by *HPDM* are interactively manipulatable, visually acceptable, topology structure changeable and real-time physicall accurate deformation. In a surgical simulation, collision detection is a time-consuming operations. Many algorithms have been proposed for detecting the collision between a moving object and a rigid model; however, those *algorithms are not applicable for deformable organs. The* virtual human organs can not be considered as a rigid model because the virtual human organs can be deformed. Thus, we also develop a collision detection that is sufficiently efficient and suitable for non-rigid models.

The remaining sections of this paper are organized as *the follows. In Section 2, we describe the proposed HPDM* technique for modeling organs. Section 3 gives an integrated environment for surgical simulation. Section 4 gives experiments of physical -based deformation. Conclusions and future works are given in Section 5.

2. Hybrid physical-based deformation modeling

In this section, we introduce the mathematical models used in the proposed *HPDM* and explain the algorithm of the *HPDM* in detail.

2.1 Hybrid physical-based deformation modeling framework

The *HPDM* is comprised of three parts: (*i*) directly manipulated free-form deformation (*FFD*), (*ii*) approximate continuum deformation modeling, and (*iii*) mass-spring model for force-feedback. The *HPDM* has the "knowledge" about the nature of an object without the influence of applied forces. When an external force is applied to an object, the object uses *FFD* approach to determine a rough shape and then uses the approximate continuu deformation technique to refine the shape. The object itsel is modeled with the mass-spring mechanism to feedback the force to the users. The virtual organ based on the *HPDM* framework is interactively manipulatable, visuall acceptable, and realistic global deformation in a computationally efficient environment.

2.2 Free-form deformation

The deformation tool used for the *FFD* modeling [3, 13] is a parallelepipedical volume called lattice. The lattice is a 3-D volume mesh defined by an array of $(l+1)\times(m+1)\times(n+1)$ control points $p_{i,j,k}$. The deformation of the *FFD* lattice is then automatically passed to a embedding object. In particular, the deformation of an object inside the lattice follows the displacement of the lattice control points.

Before deforming the lattice, each vertex of the object should be associated to the lattice. Let $x=[x_1 \quad x_2 \quad x_3]$ be the world coordinates of a vertex in the world coordinate system (o,x,y,z), where o is an origin of the world coordinate system. Let $x=[x_1\ x_2\ x_3]$ be the coordinate in the lattice coordinate system (o_l,u,v,w), where o_l is an origin of the lattice coordinate system. The transformation of x from the world coordinates to the lattice coordinates is obtained from

$$x=\frac{v\otimes w\cdot(x-o_l)}{v\otimes w\cdot u}$$
$$y=\frac{u\otimes w\cdot(x-o_l)}{u\otimes w\cdot v}$$
$$z=\frac{u\otimes v\cdot(x-o_l)}{u\otimes v\cdot w}, \qquad (1)$$

where $\otimes$ represents the cross product and $\cdot$ represents the dot product. The lattice coordinates of a point lying inside the lattice are $0\le x_1\le 1$, $0\le x_2\le 1$, and $0\le x_3\le 1$, and the world coordinates of all vertices x lying inside the lattice can be expressed as

$$x=\sum_{i=0}^{l}B_i^l(x_1)\sum_{j=0}^{m}B_j^m(x_2)\sum_{k=0}^{n}B_k^n(x_3)p_{ijk}, \qquad (2)$$

where $B_a^b(c)=\sum_{a=0}^{b}\binom{b}{a}(1-c)^{b-a}$ is the Bernstein polynomial, and p_{ijk} is a control point on the lattice. The world coordinates of control points are defined as

$$p_{ijk}=o_l+\frac{i}{l}u+\frac{j}{m}v+\frac{k}{n}w. \qquad (3)$$

Once the lattice is associated to the embedding object and the control points are moved to a new position, the world coordinates of all vertices x lying inside the lattice are re-computed by $D(x)$ based on the new control points p_{ijk}

$$D(x)=\sum_{i=0}^{l}B_i^l(x_1)\sum_{j=0}^{m}B_j^m(x_2)\sum_{k=0}^{n}B_k^n(x_3)p_{ijk}. \qquad (4)$$

The global deformation of the object is obtained based on the displacement of the control points.

2.3 Direct manipulation of free form

The formula for calculating the world coordinates of vertex x or $D(x)$ in Eq.(2) or Eq.(4) can be written in matrix form

$$x_{1\times 3}=B_{1\times((l+1)(m+1)(n+1))}P_{((l+1)(m+1)(n+1))\times 3}, \qquad (5)$$

where $B=[B_0^l(x_1)\times B_0^m(x_2)\times B_0^n(x_3),\ldots,B_l^l(x_1)\times B_m^m(x_2)\times B_n^n(x_3)]$, is a single row matrix of the Bernstein polynomials, and $P=(p_{000},p_{001},\cdots,p_{(l+1)(m+1)(n+1)})$ is an $((l+1)(m+1)(n+1))\times 3$ matrix whose rows are the world coordinates of all control points.

Assume that a vertex on the 3-D object is translated an amount of Δx to a new location $x+\Delta x$, the Eq.(5) can be rewritten as

$$(x+\Delta x)_{1\times 3}=B_{1\times((l+1)(m+1)(n+1))}(P+\Delta P)_{((l+1)(m+1)(n+1))\times 3}, \qquad (6)$$

where Δx and ΔP represent the changes in the w orld coordinates of the object vertex and the control points, respectively. Combining Eq.(5) and Eq.(6) to get

$$\Delta x_{1\times 3}=B_{1\times((l+1)(m+1)(n+1))}\Delta P_{((l+1)(m+1)(n+1))\times 3}. \qquad (7)$$

Δx is the difference between the positions of vertex before and after moving, and wish to find a value ΔP satisfying Eq.(7). This can be achieved by the least-square estimation, and the pseudoinverse solution is

$$\Delta P=(B^TB)^{-1}B^T\Delta x. \qquad (8)$$

Once the pseudoinverse of B is determined, the position change of the control points based on the movement of the vertex can be expressed as

$$\Delta P=\frac{1}{\|B\|^2}B^T\Delta x. \qquad (9)$$

Because the pseudoinverse give a least squares solution, the change in control point positions is minimized. The deformed positions of the object can be calculated from $x_{new}=B(P+\Delta P)$, where x_{new} is the deformed position of the object [8].

2.4 Approximate continuum modeling

In the approximate continuum modeling formulation [7, 15], a deformable model is formulated by using the material coordinate system in the object denoted by Ω. Let S be the material coordinates of a particle in the object. For a solid object, the material coordinate system Ω is (u_1,u_2,u_3) and S has three components $[s_1\ s_2\ s_3]$ denotes the material coordinates of a point in the material coordinate system Ω. Similarly, for a surface-represented

object the Ω is (u_1,u_2) and $s=[s_1\ s_2]$ and for a curve object the Ω is (u_1) and $s=[s_1]$. In the Euclidean coordinate system, 3-D position of points in the object are given by time -varying vector-valued function $r(s,t)=[r_1(s,t)\ r_2(s,t)\ r_3(s,t)]$.

2.4.1 Kinematics and dynamics

The equations of motion for a deformable model can be written in Lagrange's form as

$$\frac{\partial}{\partial t}(\mu\frac{\partial r}{\partial t})+\gamma\frac{\partial r}{\partial t}+\frac{\partial \varepsilon(r)}{\partial r}=f(r,t),$$

(10)

where $r(s,t)$ is the Euclidean coordinates of a vertex S at time t, $\mu(s)$ and $\gamma(s)$ are the mass and damping density of the object Ω at a vertex S, $f(r,t)$ is the externally applied force, and $\varepsilon(r)$ is a function which measures the internal energy of the elastic deformation of the object Ω. In motion theory, force is the derivative of energy with displacement. Hence $\partial\varepsilon(r)/\partial r$ represents the elastic force that is the variational derivative of th e internal energy. The external forces are balanced against the force terms on the left hand side of Eq.(10) due to the deformable model. The first term is the inertial force due to the model's distributed mass. The second term is the damping force due to dissipation. The third term is the elastic force due to the deformation of the model away from its natural shape.

2.4.2 Internal energy of deformation

The shape of a object is determined by the Euclidean distances between nearby vertices. As the object def orms, these distances change. Let S and ds denote the material coordinates of two nearby vertices in the object. The distance dl between these vertices in the deformed object in Euclidean 3-D is given b

$$dl=\sum_{ij}G_{ij}du_idu_j,$$

(11)

where G_{ij} is the (i,j)th entry of symmetric matrix G and is defined b

$$G_{ij}(r(s))=\frac{\partial r}{\partial u_i}\cdot\frac{\partial r}{\partial u_j}.$$

(12)

The symmetric matrix G is a metric tensor which is a measure of deformation. Two 3-D solid obje cts have the same shape if their 3×3 metric tensors are identical functions of $\Omega=(u_1,u_2,u_3)$. However, the changes of length between nearby vertices do not determine whether deformation happens or not when the solid object bec omes infinitesimally thin in one or more of its dimensions. Thus, the lengths between nearby vertices do not determine the shape of a surface, since curvature can be altered without affecting lengths. Thus, two surfaces have the same shape if their metric tensors G as well as their curvature tensors B are identical functions of $\Omega=(u_1,u_2)$. The curvature tensor B is defined b

$$B_{ij}=(r(s))=n\cdot\frac{\partial^2 r}{\partial u_i\partial u_j},$$

(13)

where $n=(n_1,n_2,n_3)$ is the unit surface normal.

Using the above differential quantities, we can define the internal energy of deformation for elastic curves, surface, and solid objects. These energies restore deformed objects to their natural shapes, while being natural with respect to rigid body motion. Thus, An object is in its undeformed shape if its internal energy is zero and the energy should grow larger as the object gets increasingly deformed away from its undeformed shape.

A reasonable internal energy for an elastic solid object is a norm of the difference between the fundame ntal forms of the deformed shape and the fundamental forms of the undeformed shape. This norm measures the amount of deformation away from the undeformed state. The internal energy for a solid object is defined as

$$\varepsilon(r)=\int_\Omega\left\|G-G^0\right\|_\alpha^2+\left\|B-B^0\right\|_\beta^2du_1du_2du_3,$$

(14)

where $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are the weighted matrix norms. G^0 and B^0 are the fundamental forms associated with th undeformed solid object. The internal energy is zero when the solid is in the un deformed state.

Substituting Eqs.(12) and (13) to Eq.(14) to obtain the simplified deformation internal energy for a solid object as

$$\varepsilon(r)=\int_\Omega\sum_{i,j=1}^{3}(\eta_{ij}(G_{ij}-G_{ij}^0)^2+\xi_{ij}(B_{ij}-B_{ij}^0)^2)du_1du_2du_3,\quad(15)$$

where $\eta_{ij}(s)$ and $\xi_{ij}(s)$ are weighting functions. The variational derivative $\delta\varepsilon(r)/\delta r$ can be approximated by the vector expression

$$e(r)=\sum_{i,j=1}^{3}-\frac{\partial}{\partial u_i}(\alpha_{ij}\frac{\partial r}{\partial u_j})+\frac{\partial^2}{\partial u_i\partial u_j}(\beta_{ij}\frac{\partial^2 r}{\partial u_i\partial u_j}),$$

(16)

where $\alpha_{ij}(s,r)$ and $\beta_{ij}(s,r)$ are constitutive function describing the elastic properties of the material.

$$\alpha_{ij}(s,r)=\eta_{ij}(s)(G_{ij}-G_{ij}^0).$$

(17)

When α_{ij} is positive, the solid object wants to shrink in extent and when α_{ij} is negative, it wants to grow. Thus the α_{ij} are controlling solid tensions which minimize the deviation of the solid's actual metric form its undeformed metric G_{ij}^0. As $\eta_{ij}(s_0)$ is increased, the material 's resistance to such deformation increases at material point s_0. Similarly,

$$\beta_{ij}(s,r)=\xi_{ij}(s)(B_{ij}-B_{ij}^0).$$

(18)

When β_{ij} is positive, the solid object wants to be flatter and when β_{ij} is negative, the solid wants to be more curved. Thus β_{ij} are controlling solid rigidities which act to minimize the deviation of the solid 's actual curvature from it's undeformed curvature B_{ij}^0. As $\xi_{ij}(s_0)$ is increased, the material becomes more resistant to such

deformation at material point s_0.

2.4.3 Discretization

The elastic force expressed by Eq.(16) is a continuous function. It can be discretized by applying finite element or finite difference approximation methods. Discretization transforms the partial differential Eq.(16) into a system of ordinary differential equations.

Assume a solid object is represented by a regular $L\times M\times N$ discrete volume of nodes. For a node, it 's material coordinate is represented by $s=[l\ m\ n]$, where $0\le l\le L$, $0\le m\le M$, and $0\le n\le N$. $r(l,m,n)$ is the Euclidean coordinate according to the material coordinate $s=[l\ m\ n]$. The Eq.(16) can be approximated to following equation by applying finite difference methods.

$$
\begin{aligned}
&e(l,m,n)=\\
&(2\alpha_{11}+2\alpha_{12}+2\alpha_{13}+2\alpha_{23}+2\beta_{11}+2\beta_{22}+2\beta_{33})\times r(l,m,n)-\\
&(\alpha_{11}+\alpha_{12}+\alpha_{13})\times[r(l-1,m,n)+r(l+1,m,n)]-\\
&(\alpha_{21}+\alpha_{22}+\alpha_{23})\times[r(l,m-1,n)+r(l,m+1,n)]-\\
&(\alpha_{31}+\alpha_{32}+\alpha_{33})\times[r(l,m,n-1)+r(l,m,n+1)]-\\
&2\alpha_{12}[r(l-1,m+1,n)+r(l+1,m-1,n)]-\\
&2\alpha_{13}[r(l-1,m,n+1)+r(l+1,m,n-1)]-\\
&2\alpha_{23}[r(l,m-1,n+1)+r(l,m+1,n-1)]-\\
&\beta_{11}[r(l-2,m,n)+r(l+2,m,n)]-\\
&\beta_{22}[r(l,m-2,n)+r(l,m+2,n)]-\\
&\beta_{33}[r(l,m,n-2)+r(l,m,n+2)].
\end{aligned}
\tag{19}
$$

If the nodal variables comprising the grid functions $r(l,m,n)$ and $e(l,m,n)$ are collected into the LMN dimensional vectors R and E, the Eq.(19) can be written in the vector form

$$E=K(r)R,$$

(20)

where $K(r)$ is an $LMN\times LMN$ matrix known as th stiffness matrix. K is a sparse and banded matrix because of the local nature of the finite difference discretization.

Consider the time invariant mass density $\mu(s,t)=\mu(s_1,s_2,s_3)$ and damping density $\gamma(s,t)=\gamma(s_1,s_2,s_3)$ in Eq.(10). The resulting discret densities are $\mu(l,m,n)$ and $\gamma(l,m,n)$. Let M be the mass matrix that is a diagonal $LMN\times LMN$ matrix with the $\mu(l,m,n)$ variables as diagonal components and let C be the $LMN\times LMN$ damping matrix constructed similarl from $\gamma(l,m,n)$. The dynamic motion equation Eq.(10) can be discretized as

$$M\frac{\partial^2 R}{\partial t^2}+C\frac{\partial R}{\partial t}+K(r)R=F,$$

(21)

where F is the vector representing discrete net external force.

However, Eq.(21) is the dynamic motion of second-order ordinary differential equations. In our surgical simulation system, when an external force is applied, a deformable model needs to reach the equilibrium state as soon as possible. Because the dynamic laws of motion are more difficult to handle and computationally more expensive, It is not necessary to provide the real -time interactions between a virtual organ and virtual instruments in our surgical simulation. Thus, Eq.(21) must be reduced to the static equilibrium equation by omitting the time effect as

$$K(r)R=F.$$

(22)

We are only interested in the final equilibrium state. In the equilibrium state, the $\frac{\partial^2 R}{\partial t^2}=\frac{\partial R}{\partial t}=0$. Thus, we can omit the time effect and reduce the Eq.(21) to stati equilibrium equation. The static equilibrium equation is faster to compute by Gauss -Seidel method in real -time simulation because no time integration is needed, t he stiffness matrix $K(r)=K(r^0)$ is time-invariant, and the static equilibrium equation is a sparse linear system.

2.5 Mass-spring deformation modeling

A mass-spring model is one physical -based technique that has been used widely and effectivel y for modelin deformable objects. An object is modeled as a collection of point masses connected by springs in a grid mesh structure. Each mass point is linked to its neighbors by massless springs. There are three types of springs between neighboring points. Shear springs are introduced in order t give the mass-spring model a shear rigidity and prevent the mass point from excessive and unrealistic distortion in its own plane. Flexion springs are introduced in order to give the mass-spring model a stress r igidity when flexion occurs in the surface and therefore have a three -dimensional role. These springs are used to introduce constraints int mass-spring model and provide the mass -spring model with an extra degree of angular stiffness.

The equations of mot ion for a deformable model can be written in Newton's law of dynamics as

$$E_{ext}(i,j)+E_{\text{int}}(i,j)=ma(i,j),$$

(23)

where m is the mass of the point $p(i,j)$ and $a(i,j)$ is the acceleration of that point. $E_{\text{int}}(i,j)$ and $E_{\text{int}}(i,j)$ represent the internal energies and external energies at $p_{i,j}$, respectively;

$$E_{\text{int}}(i,j)=-\sum_{(k,l)\in R}k\left(\overline{p_{i,j}\,p_{k,l}}-\left\|\overline{p_{i,j}\,p_{k,l}}\right\|_0\frac{\overline{p_{i,j}\,p_{k,l}}}{\left\|\overline{p_{i,j}\,p_{k,l}}\right\|}\right),\tag{24}$$

and

$$E_{ext}(i,j)=-cv_{i,j},$$

(25)

where k is the stiffness of the springs, R is a set of neighboring points surrounding $p_{i,j}$, $\left\|\overline{p_{i,j}\,p_{k,l}}\right\|_0$ is the undeformed length between the point $p_{i,j}$ and $p_{k,l}$, C is the damping constant, and $v_{i,j}$ is the velocity of $p_{i,j}$. The dynamic equation Eq.(23) can be explicitly integrated with respect to time by a simple Euler method.

2.6 Hybrid physical-based deformation modeling approach

In the proposed surgical simulation system, a user holds surgical instruments to touch virtual organs resulting in the deformation of organs here we propose a *HPDM* technique which combines the techniques of directly manipulation *FFD* and approximate continuum modeling to exhibit the

correctly physical deformation of virtual organs. The propose *HPDM* approach contains four stages: (*i*) re-position of control points, (*ii*) refining the *FFD* lattice, (*iii*) relocating all vertices position of the virtual organ, and (*iv*) computes the reaction force and provides a feeling of touch and force sensations for user.

A. Re-position of control points

When surgical simulation detects the collision between a virtual organ and surgical instruments, the collision vertices on the virtual organ would be moved (pressed or grabbed) by the virtual instruments. The difference between the position of collision vertices before and after moving by virtual instruments are denoted Δx and Δy, respectively. Two vertices on the 3-D object are translated to a new location $x+\Delta x$ and $y+\Delta y$. Thus, the Eq.(5) can b rewritten as

$$\begin{bmatrix} x \\ y \end{bmatrix}_{2\times 3} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}_{2\times 3} = B_{2\times((l+1)(m+1)(n+1))}(P+\Delta P)_{((l+1)(m+1)(n+1))\times 3}, \quad (26)$$

where

$$B = \begin{bmatrix} B_0^l(x_1)\times B_0^m(x_2)\times B_0^n(x_3),\ldots,B_l^l(x_1)\times B_m^m(x_2)\times B_n^n(x_3) \\ B_0^l(y_1)\times B_0^m(y_2)\times B_0^n(y_3),\ldots,B_l^l(y_1)\times B_m^m(y_2)\times B_n^n(y_3) \end{bmatrix}. \quad (27)$$

According to the directly manipulation technique, the position changes of control points based on the move ment of the vertices can be expressed as

$$\Delta P = (B^T B)^{-1}_{((l+1)(m+1)(n+1))\times((l+1)(m+1)(n+1))} B^T_{((l+1)(m+1)(n+1))\times 2} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}_{2\times 3}. \quad (28)$$

B. Refining the lattice

The *HPDM* using the metric and curvature tensor based on approximate continuum modeling to define the internal energy of deformation for the *FFD* lattice. These energies restore deformed lattice to their natural shape. When the lattice is deformed, the energy should grow larger as the lattice gets increasingly deformed away from its undeformed shape and determine the shape of the deformable lattice by moving the contr ol points to achieve a minimum energy state.

C. Relocating all vertices position of the virtual organ

After the refining the *FFD* lattice, the control points are moved to a new position and the world coordinates of all vertices on the virtual organ embedded in the lattice are relocates by Eq.(4).

D. Computing the reaction force

Once the collision between the virtual instruments and the 3-D virtual organs is detected, the reaction force will feedback to the user's hand. In our surgical simulation system, the organ models are modeled with mass-spring physical-based mechanism. Thus, the interaction between virtual organs and instruments will lead to feedback force to the user. For each iteration in our surgical simulation, the reaction force is calculated by the equation

$$F_{reaction} = k_1(p_{current} - p_{original}) + \sum_{i=0}^{N} k_2 \frac{(|l_i| - d_i)}{|l_i|} l_i, \quad (29)$$

and sent to the haptic device, where N is the number of neighboring vertices, $l_i = p_{current} - p_i$, and $d_i = |p_{original} - p_i|$. $p_{current}$ and $p_{original}$ represent the current and original coordinates of the collision vertex, p_i represents the neighboring vertex of the $p_{original}$, and k_1 and k_2 are the spring coefficients denote the pushing or pulling and shearing rigi dit

The first component of Eq.(29) represents the home force that pulls or pushes the vertex back to its original position, and the second one represents the forces due to neighboring mass vertices. The spring coefficients, k_1 and k_2 can be selected properly depending on the application to simulate the surfaces with different material properties.

3. Integrated environment for surgical simulation

In this section, we first describe how we integrate a virtual environment for surgical simulation. Then, we discuss a major requirement that are efficient collision detection in our surgical simulation.

3.1 Surgical simulation environment

The human abdomen is more complexity because there are several organs including a liv er, a gallbladder, a stomach, two pancreas, and bowel. To create a virtual abdominal model includes all 3-D virtual organs is expensive in the surgical simulation because the interaction between a virtual organ and other virtual organs do not only decrease real-time performance but also make more difficult collision detection and implementation. The surgical simulation allowing user to modify and interact with virtual organs in real-time is an essential requirement. To reduce the complexity of the virtual e nvironment, we only create two 3-D virtual organs in our virtual environment and the others composed a background image.

We used *OpenGL* ™, a standard computer graphics library, to create our laparoscopic surgical environment. There are two virtual instruments and two 3-D virtual organs with textures in our surgical simulation as shown in Fig.1. The 3-D virtual organs with textures can give users a more realistic visualization and feeling of participating in our surgical simulation. Also, the virtual organs are arranged as the positions and orientations of real organs in the virtual abdomen. In the *MIS* system, a surgeon uses a laparosocpe to explore the internal abdomen on a monitor and perform operations about surgery. Because of circular lens of the laparosocpe, the monitor display of our virtual environment is also created with a circular scene.

In the *MIS* simulation system, surgeons operate surgical procedure by insert a laparosocpe and customized surgical instruments into the abdominal model of patient through small skin incisions and explore the internal abdomen on the monitor. The accuracy position of virtual laparosocpe and instruments must be considered. We have also to define the relative position of the virtual laparosocpe and virtual surgical inst ruments according to the realistic *MIS* in our surgical simulation. In addition, we can on -line adjust the field of the view of a virtual laparosocpe, focal length of the virtual laparosocpe, and the position of virtual laparosocpes.

A surgical simulation system has several software modules: the physical-based deformation model system, the graphics rendering environment, and the control module for the device. The input device is the mechanism that produces the force feedback and tracks the position of the hands and fingers of the user at the same time. This device

directly interacts with the physical-based deformable model. To produce a user -friendly and realistic feel, the device should have low inertia and friction.

Summarily, The *VR* input device must have four basic functions

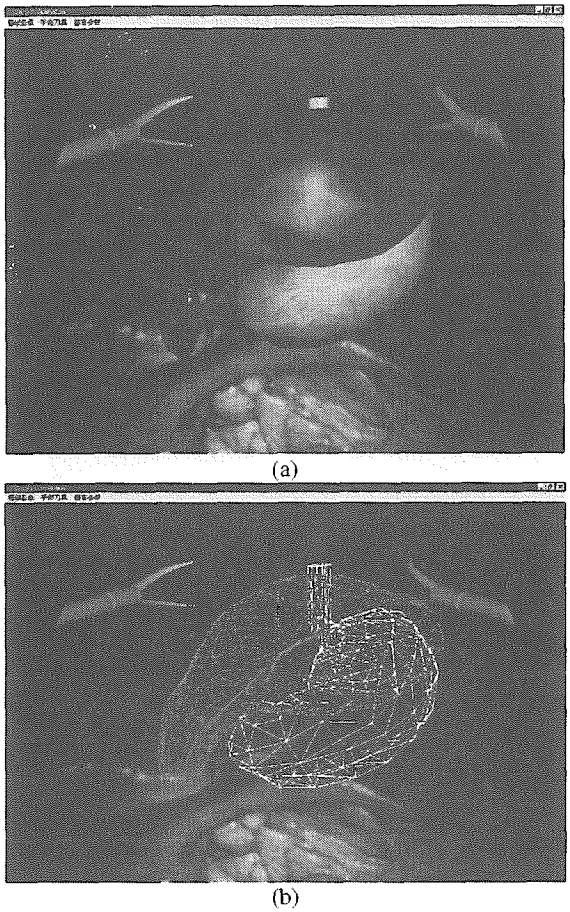
(*i*) Rapidly and accurately track the positions of the surgical instruments.

(*ii*) Realistic feelings of holding surgical instruments. The situation that users operate the input device is like they operate the actual surgical instruments.

(*iii*)Provide force feedback mechanism.

(*iv*) Low inertia and friction.

In our surgical simulation, two laparoscopic impulse engines produced by Immersion Corporation are used t interact between virtual organs and instruments. They provides the functi ons of five degree-of-freedom position tracking with and three-degree-of-freedom force feedback.



(a)

(b)

Fig.1. A virtual stomach and a virtual liver in our virtual abdominal environment. (a) Virtual organs with textures. (b) Virtual organs without textures.

3.2 Efficient collision detection

The simplest algorithm for collision detection is based on the using of bounding boxes. However, A virtual organ can not be considered as a rigid model in a surgical simulation because the virtual organ can be defor med by virtual instruments. Traditional collision detection algorithms are not suitable for non-rigid models and their performance slows down in a large number of boundin box pairs for detection. The performance of traditional collision detection is $O(n^2)$, where n is the number of polygons.

To improve the performance of collision detection and the suitability of non-rigid models, we use hierarchies of bounding box model representation using axis -aligned bounding boxe s to provide a fast way to perform exact collision detection between complex models. The hierarchies of bounding box model denoted by axis-aligned bounding boxes trees (*AAB* trees) algorithm is comprised of three parts: *i*) Building an *AABB* tree, (*ii*) intersecti testing, and (*iii*) updating a *AABB* tree after a deformation.

3.2.1 Building an AABB tree

The axis-aligned bounding boxes are aligned to the axes of the model's local coordinate system. An *AABB* tree is constructed top-down by recursive subdivision. At eac recursion step, the smallest *AABB* of the set of polygons is computed, and the set is split by ordering the polygons with respect to a well-chosen partitioning plane that is orthogonal to the longest axis of the axis-aligned bounding box. We positi on the partitioning plane denoted δ along the longest axis and the partitioning plane intersects the midpoints on the longest axis as show in Fig. 2. An *AAB* tree for n polygons has n leaves and n-1 internal nodes.

3.2.2 Intersection testing

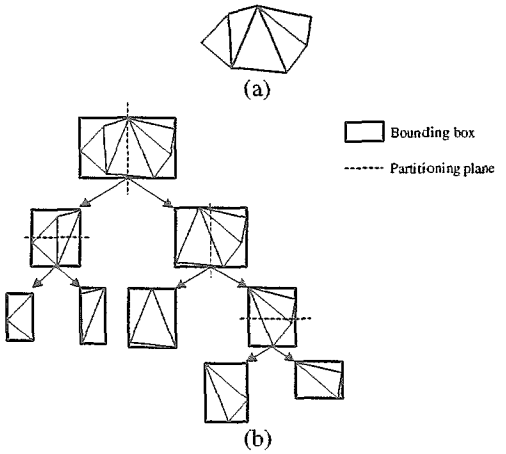
An intersection test between two models is done by recursively testing pairs of nodes. Only the nodes for which the axis-aligned bounding box overlap are further traversed. The intersection testing methods for *AABB* trees is comprised of three steps.

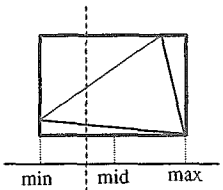
Step 1 If both nodes are leaves then the polygons are tested for intersection and retrieve the result of the test.

Step 2 If one of the nodes is a leaf and the other an internal node, then the leaf node is tested for intersection with each of the chil dren of the internal node.

Step 3 If both nodes are internal nodes and are overlap then the children nodes of both nodes are tested for intersection.

The performance of an intersection test between two *AAB* trees is $O(\log n)$, where n is the number of polygons.





(c)

Fig.2. The structure of an *AABB* tree. (a) A triangle mesh. (b) The *AAB* tree of the triangle mesh. (c) The polygon is classified as a right child node, since its midpoint on the coordinate axis is greater than δ.

3.2.3 Updating an AABB tree after a deformation

The *AABB* tree provides a straightforward method for refitting a hierarchy of *AABBs* after deformation in a bottom-up manner. At first, the bounding boxes of the leaves are recomputed, then each parent box is recomputed using the boxes of its children in a strict bottom -up order. The performance of an updating *AAB* tree operation is $O(\log n)$, where n is the number of polygons.

Summarily, the hierarchic collision detection algorith provide a fast way to perform exact collision detection between complex models and suitable for non-rigid models.

4. Experiments and discussions

Several experimental results of the proposed approach for hybrid physical -based deformation modeling are presented. The virtual abdominal environment for experiments is shown in Fig. 1.

Several comparisons are also given as follows: (*i*) the comparison between elastic deformation mode lin proposed by Terzopoulos *et al*. [14] and the proposed approach, (*ii*) the comparison of performance between two different-resolution virtual organ models, and (*iii*) the comparison of performance among different-performance computers.

A. Comparison of different approaches

Our experimental platform is an Intel-based *PC* with Pentium Pro™ 166 *CPU*, 6 *MB RAMs* and run in *Windows NT*™ operation system with *L2300* graphics accelerator. The program of surgical simulation is developed by Visual C++ 5.0 and *OpenGL*™. The 3-D virtual organs models we used are a liver model with 692 vertices and 1332 polygons and a no -deformable stomach model with 1184 vertices and 2132 polygons. The virtual organs are put in the virtual abdomen to evaluate the performances of two def ormati modeling approaches.

Two criteria of the simulation: frame rate and global deformation are given for comparing these two approaches. The results of frame-rate comparison are shown in Table 1. The proposed deformation modeling has better performance than approximate continuum modeling.

Table 1. Frame-rate Comparison between Two Approaches

Approach / Models	Approximate continuum modelin	Hybrid physical-based deformation modelin
Only a liver model	3.2 - 3.5	8.0 - 10.0
A liver model and a stomach model	-	7.0 - 8.5

Considering the second criterion, the approximate continuum modeling can not real -time simulate the volumetric deformation. Because the geometric representation of a deformable organ based on approximate continuum modeling must be a volum etric model. The proposed approach uses *FFD* and approximate continuu modeling technique to simulate the volumetric deformation of the lattice resulting in the volume -like deformation of virtual organ. In terms of volumetric deformation, the proposed deformation modeling is advantageous because it can real-time simulate the volume-like deformation for an surface models.

B. Comparison of different-resolution models

Using different resolution models in surgical simulation would affect the simulation performance including execution time and fidelity. The execution time is measured by frame rate. Two different-resolution environments are used for evaluating.

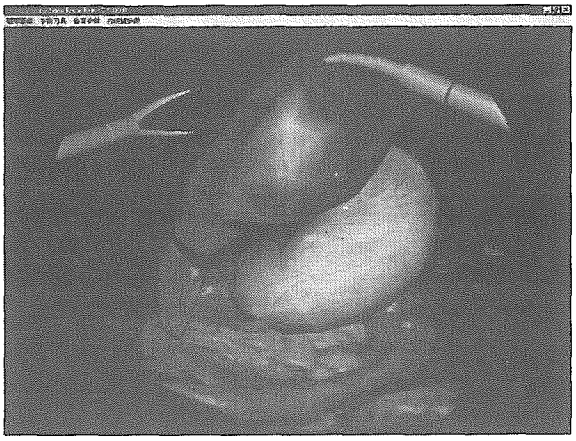
(*a*) Low-resolution environment consists of two 562-polygon virtual surgical instruments, a 274-polygon liver model, and a non-deformable stomach model with 2132 polygons.

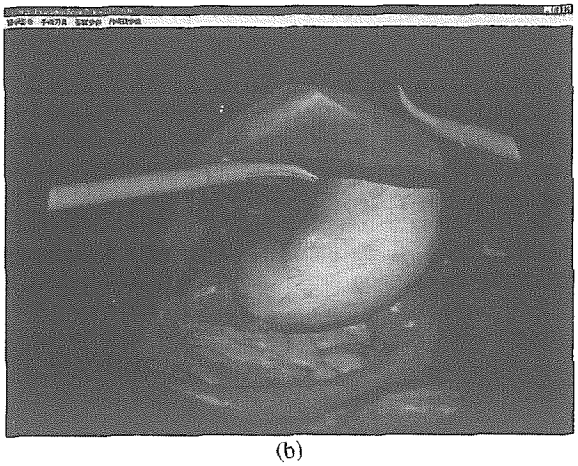
(*b*) Hig -resolution environment consists of two 562-polygon virtual surgical instruments, a 1332-polygon liver model, and a non-deformable stomach model with 2132 polygons.

The comparison is given in Table 2. For only a liver model, we find that the high-resolution model has six-fold number of vertices and polygons to the low -resolution model, but the frame rate has less than one fold. The frame rate is not linearly degraded because the *CPU* load is larger than the graphics accelerator card. Two samples of the low-resolution and high-resolution deformation models are illustrated in Fig.3.

Table 2. Comparison of Different-resolution Models

Approach / Models	Low-resolution model	Hig -resolution model
Only a liver model	11.5-17.5	8.0-10.0
A liver model and a stomach model	9.0-12.5	7.5-8.5


(a)



(b)

Fig.3. Samples of operations on different-resolution organ models. (a) Grabbing the high-resolution liver model. (b) Lifting the low-resolution liver model.

C. Comparison of different computer platforms

Using different computer platforms for the proposed surgical simulation would affect the simulation performance including execution time and fidelity. The execution time is measured by frame rate. Four different computer platforms are compared using the high-resolution model. The comparison is given in Table 3.

Table 3. Comparison of Different Computer Platforms

Platforms	Frame rate (frames/second)
Pentium Pro 166 *MHz* with *Permedia2*	7.5-8.5
InterGraph TDZ-410 *with Pentium Pro* 200 *MHz*	17.5
Pentium II 350 *MHz* with *RIVA TNT*	21.5
Pentium II 350 *MHz* with *RIVA TNT2*	26.5

Based on the experimental results, e draw several discussions as follows

(*i*) The *HPDM* approach exhibits more physical accuracy and realism than the approximate continuum modeling exhibits when the virtual organ models are pressed, pushed, or grabbed by the surgical instruments.

(*ii*) The geometric models we used in the surgical simulation are surface repre sentation models. The *HPDM* can real-time simulate the physical-based volume-like deformation for surface representation models

(*iii*)The volume preserving for surface models is more simply achieved using the *HPDM* than using mass-spring modeling and approximate continuu modeling.

(*iv*) It is very simple to change physical properties of models by adjusting parameters for *HPDM*. The *HPDM* uses two parameters α and β to adjust the material properties for stretching and bending.

5. Conclusions

In this paper, we propose a hybrid physical -based deformation modeling (*HPDM*) technique for our surgical simulation system. The *HPDM* approach achieves the realistic volume-like deformation of virtual organs in a computationally efficient framework. We also create a force feedback framework for the laparoscopic surgical simulation system. Users can interactively manipulate with virtual organs in the virtual abdominal environment and the simulation system provides force-feedback facility to users. To improve the performance, a collision detection algorithm that is sufficiently efficient and suitable for non-rigid models is developed. Since our surgical simulation system lets the user perform surgical operations intuitively, it provides a realistic environment for surgical simulation. However, there are several tasks or functions should be considered in the future:

(*i*) Tissue cutting and suturing are important for a surgical simulation system. In practice, cutting and suturing will change the topology structures of organs. The current proposed surgical simulation does not allow the cutting and suturing operation on virtual organs.

(*ii*) The force feedback system based on multi -thread technique and the virtual instruments changeable are also the future research topics.

(*iii*)The current system does not include the effect of the interaction between multiple organs. Because the computational complexity is expensive time-consuming. We need to extend the collision detection to detect collision between multiple organs for the deformation.

(*iv*) There are only a liver and a stomach virtual organs in the current proposed surgical simulation; we need to incorporate the virtual gallbladder into the virtual abdomen and build a gallbladder removing exa mple in the future study.

(*v*) A complex virtual organ based on the *HPDM* technique can not exhibit physically correct behaviors corresponding to behaviors of real human organs. Thus, a complex organ has to be divided into several components and the deformation of one component has to affect the others.

References

[1] Basdogan, C., C.-H. Ho, M. A. Srinivasan, S. D. Small, and S. L. Dawson, "Force interactions in laparoscopic simulations: haptic rendering of soft tissues," in *Proc. of the medicine meets virtu al reality conference*, San Diego, CA, Jan.19-22, 1998, pp.385-391.

[2] Bro-Nielsen, M., "Finite element modeling in surger simulation," *Proc. IEEE*, Vol.86, No.3, pp.490 -503, 1998.

[3] Coquillart, S., "Extended free-form deformation: sculpturing tool for 3D geometric modeling," in *Proc. Siggraph'90*, Vol.24, No.4, Dallas, T , Aug.6-10, 1990, pp.187-196.

[4] Cotin, S., H. Delingette, and N. Ayache, *Efficient Linear Elastic Models of Soft Tissues for Real-time Surgery Simulation*, Technique Report TR-98-3510, Institut National de Recherche en Informatique et en Automatique (INRIA), 1998.

[5] Cover, S. A., N. F. Ezquerra, J. F. O'Brien, R. Rowe, J. Gadacz, and E. Palm, "Interactively deformable models for surgery simulation," *IEEE Computer Graphics & Applications*, Vol.13, No.6, pp.68 -75, 1993.

[6] Delingette, H., "Toward realistic soft-tissue modelin in medical simulation," *Proc. IEEE*, Vol.86, No.3, pp.512-523, 1998.

[7] Gibson, S. F. F. and B. Mirtich, *A Survery of Deformable Modeling in Computer Graphics*, Technique Report TR-97-19, Mitsubishi Electric Research Laboratory, 1997.

[8] Hsu, W. M., J. F. Hughes, and H. Kaufman, "Direct

manipulation of free-form deformations ," in *Proc. Siggraph'92*, Vol.26, No.2, Chicago, I , Jul.26-31, 1992, pp.177-184.

[9] Kuhn, C., U. Kuhnapfel, H -G. Krumm, and B. Neisius, "A 'virtual realit ' based training system for minimall invasive surgery," in *Proc. Computer Assisted Radiology* (*CAR'96*), Paris, Jun., 1996, pp.764-769.

[10]Kuhnapfel, U. G. and B. Neisius, "CAD-based graphical computer simulation in endoscopic surgery," *Endoscopic Surgery and Allied Technologies* , Vol.1, No.2, pp.181-184, 1993.

[11]Meseure, P. and C. Chaillou, "Deformable body simulation with adaptive subdivision and cuttings ," in *Proc. WSCG'97*, Plzen, Feb.10-14, 1997, pp.361-370.

[12]Qin, H. and D. Terzopoulos, "Dynamic NURBS swung surfaces for physics -based shape design, " *Computer-Aided Design*, Vol.27, No.2, pp.111-127, 1995.

[13]Sederberg, T. W. and S. R. Parry, "Free-form deformation of solid geometric m odels," in *Proc. Siggraph'86*, Vol.20, No.4, Dallas, TX, Aug.18-22, 1986, pp.151-160.

[14]Terzopoulos, D. and H. Qin, "Dynamic NURBS with geometric constraints for interactive sculpting," *ACM Trans. on Graphics*, Vol.13, No.2, pp.103-136, 1994.

[15]Terzopoulos, D., J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," in *Proc. Siggraph'87*, Vol.21, No.4, Anaheim, CA, Jul.27-31, 1987, pp.205-214.