Haptic Force Feedback with an Interaction Model between Multiple Deformable Objects for Surgical Simulations

Yoshihiro Kuroda[1)], Megumi Nakao[1)], Silke Hacker[2)],
Tomohiro Kuroda[3)], Hiroshi Oyama[3)], Masaru Komori[4)],
Tetsuya Matsuda[1)], Takashi Takahashi[3)]

[1)] *Graduate School of Informatics, Kyoto University, Japan*
[2)] *IMDM University Hospital Hamburg-Eppendorf, Germany*
[3)] *Medical Informatics, Kyoto University Hospital, Japan*
[4)] *Biomedical Science, Shiga University of Medical Science, Japan*
E-mail: ykuroda@kuhp.kyoto-u.ac.jp

Abstract

This paper proposes an interaction model between multiple physically-based deformable objects. The model enables both accurate force feedback and visualization of surgical manipulations (like hold, push and move organs) while approaching the tissues of interest. Accurate force feedback improves surgical realism and enables exact simulation for diagnosis and procedural training. Interaction is represented by mutual iterative procedures of forcible displacement, calculation of deformation and conveyance of reaction force. The proposed model has been applied to a simulation where a manipulating point pushes a deformable object, which is in contact with a neighboring one, and the results assures that the advanced interaction model enables us to feel even the fine differences resulting from the physical behavior of neighboring organs.

1. Introduction

Interest in medical simulation based on Virtual Reality has been growing among medical and computer scientists in this decade. The mechanics and deformation of soft tissue, which are essential to achieve surgical simulations, have been studied intensively in both biomechanics [15] and computer graphics [1, 2, 3, 4]. In medical surgery, haptic feel is key information to identify organs and its status, especially under the limited views in endoscopic surgery. Haptic display has also been much studied in close relation to collision simulations in computer graphics [5, 6, 7]. The surgical manipulations, such as probing, piercing, suturing, cutting, puncturing and so on, are simulated with haptic display using point-based or ray-based haptic rendering techniques [8, 9, 10, 11]. Although several surgical simulators have been developed using such techniques [12, 13], most simulators display haptic force feedback with a single organ in no contact with neighboring organs. However, in real-life surgery, dynamic relations among neighboring organs affect force feedback at a surgeon's hand and change the passage and result of surgical manipulations. In order to take account of the influence of neighboring organs, organ-organ interaction must be represented, that is, the interaction between multiple deformable objects must be represented. The representation of the interaction between multiple deformable objects is a big challenge, due to the complexity of physically accurate simulation in the discrete world and the difficulty of real-time calculation. Under these constraints, the goals of this paper are determined as follows:

1. The existence of neighboring objects influences on force.
2. The physical behavior of neighboring objects is reflected to force.
3. Realistic deformation is achieved as well.

To achieve the first goal, the physical calculation of objects applies some methods and makes several assumptions for reduction of calculation cost. The second goal requires a physically-based response, although in general the physical model needs much more computation than the geometric one. High update rates for the interaction are required, because the influence of neighboring objects is fine in comparison with that of the directly pushed objects. In order to meet the requirement, the method that decreases the computational growth of the calculation, which depends on the area in contact, is applied (detail in Section 2.2). The third goal is derived from the recent report that the good combination of haptic display and graphic display is important in surgical simulation [13]. Both the force and deformation are calculated by using physically-based models. In the experiments described in Section 3, the mass spring model is applied to the object "front object", which is directly

pushed by a manipulating point and the Finite Element Method (FEM) is applied to the object "behind object", which is located behind the front object (detail in Section 2.4).

The model described below represents the interaction between multiple deformable objects. The influence of the interaction with this model is examined.

2. Methodology

2.1 Representation of interaction

The interaction is represented by a series of procedures: update of "pairs of nearest nodes", collision detection, collision procedures, calculation of deformation and conveyance of reaction force. All procedures are carried out based on "pairs of nearest nodes", which consist of two nodes belonging to different objects. For the following explanation, the pair consisting of the node "a1" of object A and the node "b1" of object B is defined as {A:a1, B:b1} and all pairs are managed in a list.

Figure 1 illustrates a series of procedures in the case of a collision between object A and B. The nodes 'a1' and 'b1', which are registered as a pair of nearest nodes before collision, belong to object A and B respectively. The procedures consist of two steps; in each step one of the pair is treated as *active* and the other as *passive* node. During the first step, where the collision between node 'a1' and object B is examined, the node 'a1' is the *active* and the node 'b1' is the *passive* node. Fig. 1-1 illustrates the initial state when no collision between the two objects is detected. If a collision is detected (see Fig. 1-2), the determined nodes including passive node 'b1' are displaced (see Fig. 1-3). The deformation of the collided object B is calculated and the reaction force is conveyed to 'a1', which is used at the next calculation of the deformation of object A (see Fig. 1-4). During the second step, where the collision between node 'b1' and object A is examined, the node 'b1' is the *active* and the node 'a1' is the *passive* node. If a collision is detected (see Fig. 1-5), the determined nodes including passive node 'a1' are displaced (see Fig. 1-6). The deformation of the collided object A is calculated and the reaction force is conveyed to 'b1', which is used at the next calculation of the deformation of object B (see Fig. 1-7). The series of procedures is carried out iteratively.

Pairs of nearest nodes have to be checked and, if necessary, updated when the relationship between them changes. If necessary, new pairs are added to a list after the collision is detected. The calculated force is conveyed to the other node of a pair as reaction force, which is used in the next calculation.

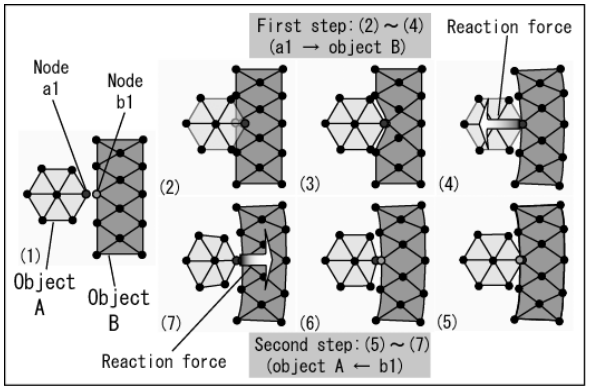

Figure 1: A series of procedures (1-7).

2.2 Improved workflow of the interaction

In Fig. 2, the workflow, which is directly derived from the model, is shown on the left side. Instead of this workflow, the improved workflow shown on the right side is applied, because, using the left workflow, the calculation of deformation is carried out every time for each pair node and therefore the calculation time increases linearly by the number of pairs of nearest nodes. The calculation cost can be reduced by improved workflow, where the calculation of deformation is carried out only once for all pairs at each step.

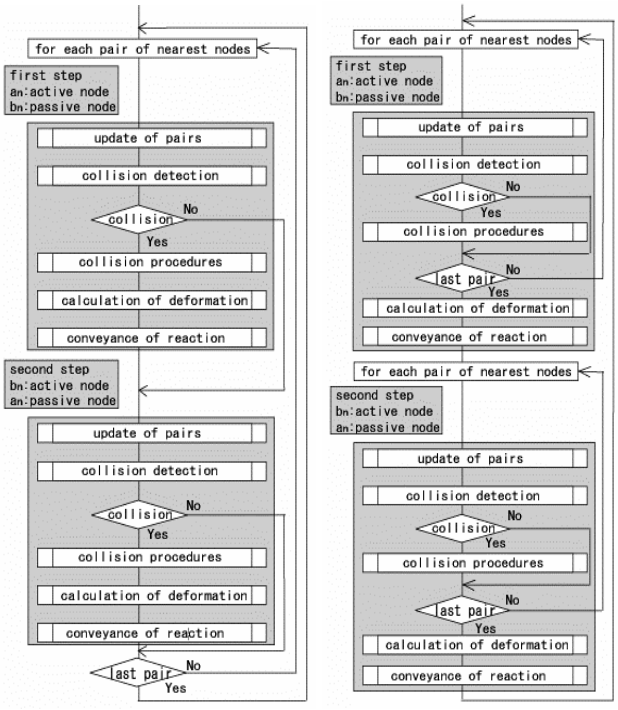

Figure 2: The workflow on the left side is directly derived from the model. The workflow on the right side is improved one.

2.3 Collision detection and collision procedures

The collision between the node of a pair and the polygon including the other node of the pair is examined. If collision is detected, the polygon is displaced forcibly. Forcible displacement is carried out with a node-polygon relation, because the node-node relation cause rough and unstable force feedback [16].

Collision detection consists of two steps. During the first step, instant collision detection is carried out based on the normal vector of the polygon, which includes the *passive* node.

$$\overrightarrow{P_{passive}P_{active}} \cdot \overrightarrow{N_i} \le 0 \tag{0}$$

where $\overrightarrow{N_i}$ is the normal vector of the polygon S_i. If the inequality (0) is satisfied for all i, the detection steps proceed to the second step, otherwise the collision is not detected. During the second step, detailed detection is carried out, checking whether the foot, F, of perpendicular of the polygon, S_i, from the active node, $P_{passive}$, is inside of S or not (see Fig. 3). If any foot is not inside, the collision is not detected, otherwise the displacement of the nodes is carried out. At first, the point, we call "*target*" point, is chosen to be the point which locally minimize the distance from the active node using Lagrange multipliers. This is in the same way as determining the position of the *God object* [5], where the polygons including a foot inside are used as constraints. Next, each node of the polygons is displaced, considering the balance of the polygon, as Fig. 3 shows.

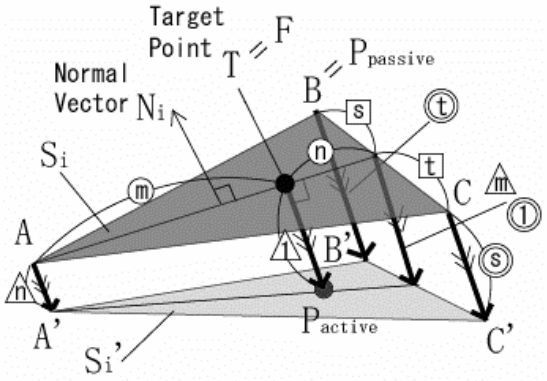

Figure 3: Displacement of a polygon

where T is the target point and the node, B, is $P_{passive}$. If there is one surface that has a foot of perpendicular F, the target point coincides with the foot. The displacement of nodes, A, B, and C is calculated as:

$$\begin{aligned}
\overrightarrow{AA'} &= \frac{\alpha}{2\alpha^2-2\alpha+1}\overrightarrow{TP_{active}} \\
\overrightarrow{BB'} &= \frac{1-\alpha}{2\alpha^2-2\alpha+1}\times\frac{\beta(\beta+\gamma)}{\beta^2+\gamma^2}\overrightarrow{TP_{active}} \\
\overrightarrow{CC'} &= \frac{1-\alpha}{2\alpha^2-2\alpha+1}\times\frac{\gamma(\beta+\gamma)}{\beta^2+\gamma^2}\overrightarrow{TP_{active}}
\end{aligned} \tag{1}$$

where $\alpha:\beta:\gamma = \Delta TBC:\Delta TCA:\Delta TAB$, $\alpha+\beta+\gamma=1$, and $\min(\alpha,\beta,\gamma)=\alpha$. The target point is displaced to the active node so that the node is on the polygon after displacement. Afterwards, the calculated forces of each node are summed up and transferred to the active node. In the case of the polygon being a square, ABDC, the new position of D is calculated as:

$$\overrightarrow{DD'} = \frac{\Delta DBC}{\Delta ABC}\overrightarrow{AA'} \tag{2}$$

where ΔABC and ΔDBC are the areas of the triangle ABC and DBC respectively.

2.4 Physically-based models and computational methods

The mass spring model is simply calculated by applying Euler's method to Newton's movement equation. On the other hand, as for FEM, some methods for fast calculation are applied and some assumptions are made, although the accurate and stable deformation based on continuum mechanics equations is achieved. A linear model, which is one of the assumptions, cannot represent realistic large deformations, although calculation cost is smaller. Since the behind object deforms smaller than the front one, the linear finite element method can be applied.

In general, the solution of FEM needs enormous computation time. This paper makes some assumptions and applies some methods to reduce the size of the stiffness matrix. Then the stiffness equation with linear elasticity is:

$$f = Ku \tag{3}$$

where f is the force, u is the displacement, K is the stiffness. For specific simulation purposes (like push, hold and so on), the surface nodes, that are *visible*, are more important than interior nodes. The interior nodes can be condensed to surface nodes. The force of internal nodes is zero in static system, in which deformation is always under equilibrium conditions.

$$f_i = 0 \tag{4}$$

where f_i is the force of interior nodes. Then a new matrix is derived which involves only the variables of

surface nodes. The force and displacement of surface nodes can be solved regardless of interior nodes. The displacement of interior nodes can be calculated, if necessary, and the volumetric behavior is managed correctly under condensation [9]. The surface nodes are classified into fixed surface nodes and free surface nodes. The displacement of surface nodes is zero in the experiment described in this paper. The force of fixed nodes is not used and therefore not necessary to be solved. The equation is reduced to:

$$f_{s_{free}} = K_{s_{free}s_{free}} u_{s_{free}} \tag{5}$$

where $f_{s_{free}}$, $K_{s_{free}s_{free}}$, $u_{s_{free}}$ are the force, stiffness, and displacement matrix of the free nodes on the surface respectively.

Hirota's method [14] enables us to compute the force of contact nodes and the displacement of other surface nodes at haptic rates. The equation (5) is classified into the contact nodes and other surface nodes.

$$\begin{pmatrix} u_o \\ u_c \end{pmatrix} = \begin{pmatrix} L_{oo} & L_{oc} \\ L_{co} & L_{cc} \end{pmatrix} \begin{pmatrix} f_o \\ f_c \end{pmatrix} \tag{6}$$

$$\begin{pmatrix} L_{oo} & L_{oc} \\ L_{co} & L_{cc} \end{pmatrix} = \begin{pmatrix} K_{oo} & K_{oc} \\ K_{co} & K_{cc} \end{pmatrix}^{-1} \tag{7}$$

where u_c, f_c are the displacement and force of contact nodes respectively, and u_o, f_o are those of other nodes. From the equations (4) and (6), the force of contact nodes f_c can be obtained only by the solution of the inverse matrix $(L_{cc})^{-1}$.

$$u_c = L_{cc} f_c \tag{8}$$

$$f_c = (L_{cc})^{-1} u_c \tag{9}$$

L_{cc} is a small part of the pre-computable matrix (8). The size of matrix L_{cc} is only $n_c \times Dimension$ and therefore it is possible to compute at haptic rates with not many contact nodes.

2.5 Addition of pairs

New pairs of nearest nodes have to be added when nodes are collided in order to represent touching surfaces. Neighboring nodes are registered in the management list as new pairs of nearest nodes. In Fig. 4-1, a pair of nearest nodes {A:a1, B:b1} is handled before the collision between two objects, A and B. In Fig. 4-2, two pairs of nearest nodes {A:a2, B:b2},{A:a3, B:b3} are added to the list.

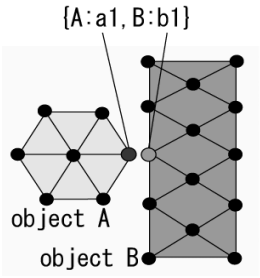

Figure 4-1: Before collision

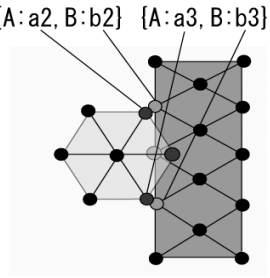

Figure 4-2: After collision

3. Results

The following experiments are carried out to examine the influence of neighboring object on force feedback. The calculation time in experiments is also described.

3.1 Components of the system and objects

The components of the system are CPU Pentium IV 1.7GHz, Main memory 1GB, Windows2000 OS, and a PHANToM Premium 1.0 haptic device (SensAble Technologies, Inc.). Objects have been made from volume datum into tetrahedral or hexahedral elements by Amira2.3 (TGS, Inc.).

Object A (size: 5.75cm*9.89cm*5.75cm) has 158 tetrahedral elements, 69 nodes, 291 lines, and 130 polygons. The elastic coefficient is 100(N/m), the damper coefficient is 0.10(Ns/m), and the mass of a node is 0.05(kg). Object B (size: 8.65cm*13.08cm*10.87cm) has 1448 hexahedral elements, 1,906 nodes, 5165 lines and 540 polygons. Young modulus is 1.0(MPa) and Poisson ratio is 0.2160(-).

3.2 Three conditions (free and fixed, contacted)

Pushing tests are performed under three conditions. In all conditions, object A is fixed on a certain plain as shown in Fig. 5. Condition 1 and 2 hold a single object A, and let all nodes on the indicated surface move freely in Condition 1, and make the nodes to be fixed in Condition 2. Condition 3 holds two objects, A and B, which are in contact with each other. As for object B, the bottom surface and the surface that is opposite side of the contacted surface are fixed.

Haptic force feedback, which is displayed when a manipulating point pushes a surface of object A towards 1.5cm depth in the speed of 0.5cm/sec, is examined.

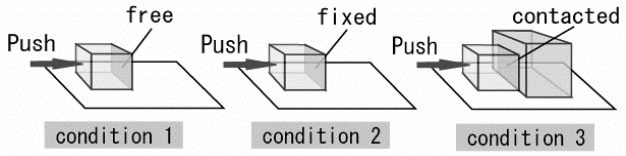

Figure 5: In Condition 1 and 2, the indicated surface of the object, A, is free and fixed respectively. In Condition 3, two objects, A and B are in contact.

Figure 6 shows the results under three conditions. The influence of the neighboring object can be found easily, such that the first goal of this paper, the existence of neighboring objects influences on force (described in Section 1), turns out to be achieved. The fact that the largest force sequence is marked in the "fixed" condition 2 implies that the fixed surface can be regarded as the surface in contact with an absolutely hard object.

The force displayed under condition 3 should change within the range of the force under condition 1 and 2, depending on the physical properties and the situation of the behind object.

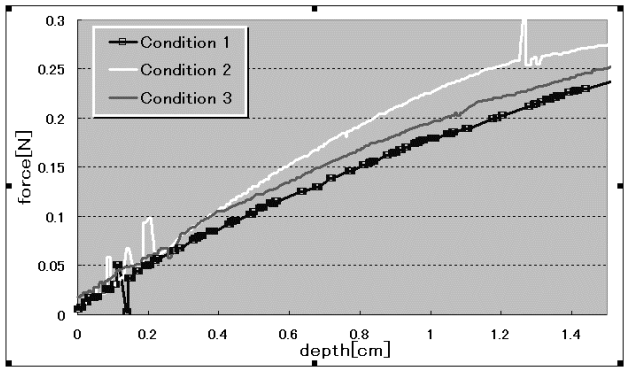

Figure 6: The force under three conditions

In Fig. 7 and 8, the forces with the several different values for physical parameters are examined. As Young modulus, 1.0, 1.5, and 2.0 (MPa) are chosen, 0.108, 0.162, and 0.216 (-) as Poisson ratio. Without the improved workflow (detail in Section 2.2), the different forces, which are caused by different physical parameters, cannot be displayed, due to low update rates. Both graphs assure the achievement of the second goal, that is, the physical behavior of neighboring objects is reflected to force (described in Section 1).

3.3 Calculation time

The average calculation time of update of pairs, instant collision detection, collision procedures, conveyance of reaction force, and addition of pairs for each object, A and B, are shown in Table 1. Table 2 and 3 show the average calculation time of the reaction force and deformation where the first column indicates the number of displaced nodes. As for object A, the calculation time of the reaction force increases linearly with the number of displaced nodes, on the other hand the calculation time of deformation is independent of it. On the other hand, as for object B, the calculation time depends on the size of the matrices corresponding to the number of displaced nodes (detail in Section 2.4). The time required for an interaction loop much depends on the calculation of deformation.

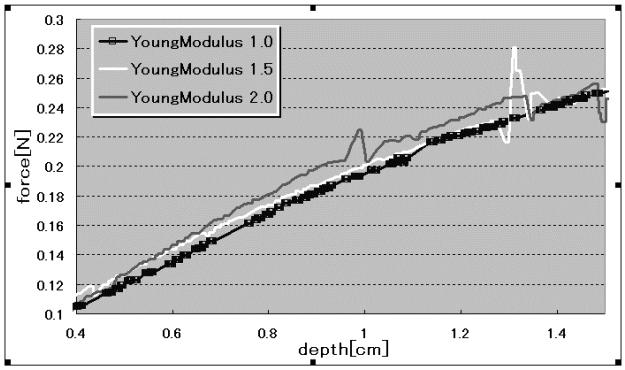

Figure 7: The force with 1.0MPa, 1.5MPa, and 2.0MPa as Young modulus.

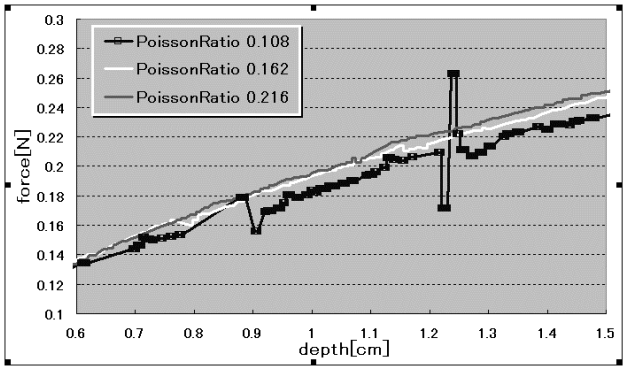

Figure 8: The force with 0.108, 0.162, and 0.216 as Poisson ratio.

Table 1: The calculation time of each procedure for each object, A and B. (msec)

	Object A	Object B
Update of pairs	0.0120	
Instant collision detection	0.0047	0.0090
Collision procedures	0.0247	0.0360
Conveyance of reaction	0.0105	
Addition of pairs	0.0026	

Table 2: The time of calculation of object A in the case of one displaced node. (msec)

	Reaction force	Deformation
1 node	0.022	1.615

Table 3: The time of calculation of object B in the case of from one to four displaced nodes. (msec)

	Reaction force	Deformation
1 node	0.076	0.157
2 nodes	0.089	0.200
3 nodes	0.110	0.480
4 nodes	0.163	1.909

4. Conclusion

This paper has described an advanced interaction model between multiple deformable objects for accurate haptic force feedback reflecting the influence of neighboring organs. Interaction has been represented by mutual iterative procedures including collision detection, forcible displacement, calculation of deformation and conveyance of reaction force and so on. The proposed model has been implemented and several experiments have been done. The results has assured that the interaction model enables us to feel the existence of neighboring objects and find the fine differences of physical behavior of neighboring objects with realistic visual display.

The objects described have relatively simple shape. In the future, the proposed model is applied to the organ objects, and applied to the surgical situation.

5. Acknowledgements

This work was partially supported by the Ministry of Education, Science, Culture of Japan, Grants-in-Aid for JSPS Fellows No.0103889.

6. References

[1] S. F. F. Gibson and B. Mitrich, "A Survey of Deformable Modeling in Computer Graphics", Technical report, http://www.merl.com, TR-97-19, MERL, Nov.1997

[2] H. Delingette, "Toward Realistic Soft Tissue Modeling in Medical Simulation", Proceedings of the IEEE, Vol.86, No.3, pp.512-523, Apr.1998

[3] R.M.Koch, M.H.Gross, F.R.Carls, D.F.von Bueren, G.Fankhauser, and Y.I.H.Parish, “Simulating facial surgery using finite element models”, ACM SIGGRAPH 96 Conference Proceedings, pp.421-428, 1996

[4] D.L.James and D.K.Pai, “ArtDefo: Accurate real time deformable objects”, ACM SIGGRAPH 99 conference proceedings, pp.65-72, 1999

[5] C.B.Zilles and J.K.Salisbury, “A constraint-based god-object method for haptic display”, Proc. IEEE/RSJ Int. Conf. On Intelligent Robotics and Systems, pp.146-151, Aug.1995

[6] D.C.Ruspini, K.Kolarov and O.Khatib, “The haptic display of complex graphical environments”, ACM SIGGRAPH 97 Conference Proceedings, pp.345-352, 1997

[7] C.Basdogan, M.A.Srinivasan, “Haptic rendering in virtual environments”, in Virtual Environments HandBook. Stanney, Ed., 2001

[8] K.D.Reinig, C.G.Rush, H.L.Pelster, V.M.Spitzer, and J.A.Heath, “Real-time visually and haptically accurate surgical simulation”, In S.H.H.Sieburg and K.Morgen, editors, Health Care in the Information Age, pp.542-546, IOS Press, 1996

[9] M. Bro-Nielsen, “Finite element modeling in surgery simulation“, Journal of the IEEE, 86(3), pp.490-503, 1998

[10] J.Brown, S.Sorkin, C.Bruyns, J.Latombe, K.Montgomery, and M.Stephanides, “Real-time simulation of deformable objects: tools and application”, Computer Animation, Seoul, Korea, Nov.2001

[11] S.P.DiMaio and S.E.Salcudean, “Simulated interactive needle insertion”, Proceedings of IEEE Haptic Interfaces for Virtual Environment and Teleoperator Systems, pp.344-351, Mar. 2002

[12] U.Kuhnapfel, H.K.Cakmak, and H.Maa, “Endoscopic surgery training using virtual reality and deformable tissue simulation”, Computers & Graphics, Volume 24, pp.671-682, 2000

[13] C. Basdogan, "Integration of Force Feedback into Virtual Reality Based Training Systems for Simulating Minimally Invasive Procedures", http://eis.jpl.nasa.gov/~basdogan/Tutorials/basdogan.html

[14] K. Hirota and T. Kaneko, “A Method of Representing Soft Object in Virtual Environment”, IPSJ (Information Processing Society of Japan) JOURNAL, Vol.39 No.12, Dec 1998

[15] W.Maurel, “Biomechanical models for soft tissue simulation”, ISBN: 3-540-63742-7, Springer, 1998

[16] Y.Kuroda, M.Nakao, S.Hacker, T.Kuroda, H.Oyama, M.Komori, T.Matsuda, and T.Takahashi, “An interaction model between multiple deformable objects for realistic haptic force feedback in surgical simulations”, Proceedings of CARS, Jun. 2002