We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800
Open access books available

122,000
International authors and editors

135M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index in Web of Science™ Core Collection (BKCI)

Interested in publishing with us? Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com

16

Manipulation of Dynamically Deformable Object using Impulse-Based Approach

Kazuyoshi Tagawa
Ritsumeikan University
Japan

Koichi Hirota
University of Tokyo
Japan

Michitaka Hirose
University of Tokyo
Japan

1. Introduction

Recent advancement of network and communication technologies has raised expectations for transmission of multi-sensory information and multi-modal communication. Transmission of haptic sensation has been a topic of research in tele-robotics for a long period. However, as commercial haptic device prevails, and as internet spreads world-wide, it became possible to exchange haptic information for more general communication in our daily life.

Although a variety of information is transmitted through haptic sensation, the feeling of a soft object is one that is difficult to transmit through other sensations. This is because the feeling of softness is represented only by integrating both the sense of deformation by somatic sensation and intensity force by haptic sensation. Feeling of softness is apt to be considered as static information that represents static relationship between deformation and force. Our previous study on implementing a static deformation model suggested that the dynamic aspect of deformation has an important effect on the reality of interactions.

A static model can not represent behavior of an object while the user is not interacting with the object. For example, it is unnatural that an object model immediately returns to its original shape just after user releases hand or finger. Also, resonant vibration of object during the interaction is often perceived through haptic sensation. These differences of dynamic model from static model are considered to become more recognizable to user as more freedom of interaction is given.

In this chapter, an outline of our approach to implement a deformable model that is capable of representing dynamic response of deformation is presented. Supplemental idea that realizes non-grounded motion of the deformable model is also stated; manipulation of deformable object becomes possible by this idea. In the next section, a survey of background research is

stated and positioning and purpose of our research is clarified. Formulation of IRDM and non-grounded object motion is discussed in section 3 and 4 respectively. Experimental results and evaluation of the proposed approach is stated in section 5. Finally, advantages and problems of the approach are discussed, and conclusion is given in section 7.

2. Related Works

2.1 Presentation of force

Presentation of the sensation of force in a virtual environment has been studied since the early stages of researches in virtual reality, and investigation has been made in both hardware and software aspects by G.Burdea (1996). Model and simulation that is used to compute force is one important part of software research, and computation of this sort is collectively called Haptic Rendering by K.Salisbury et al. (1995). Representation of deformable object has been a topic of research, because interaction with deformable objects is a quite common experience.

2.2 Motion and manipulation

The free motion of an object is computed simply by solving equations regarding the motion of the object. Computation of motion becomes difficult in cases when constraints on motion are applied by contact with other objects or user's body. A taxonomy of methodology that deals with the constraints has been presented by J.E.Colgate et al. (1995). Typically there are two approaches: one is an approach that solves equation of motion with constraint condition, and another is an approach that introduces penalty force. In computer graphics, the former approach has been presented by D.Baraff (1989), and advantage of the latter approach has been discussed by B.Mirtich & J.Canny (1995).

In haptic rendering, one of major applications of computation of motion is presentation of behavior of object while it is manipulated. Object manipulation by the user frequently causes complicated constraint conditions, and it is usually difficult to solve equations of motion with these constraints. Hence, the approach of penalty force is preferred in hatic rendering researches; Borst & Indugula (2005); K.Hirota & M.Hirose (2003); S.Hasegawa & M.Sato (2004); T.Yoshikawa et al. (1995).

2.3 Deformation model

2.3.1 Model-based approach

Visual representation of deformation has been a major topic in computer graphics. In the early stages, there was research on geometric deformation including Free Form Deformation (FFD) by T.W.Sederberg & S.R.Parry (1986). Nature of this approach that it is not based on physics-based model cause advantage and disadvantage. The nature provides more freedom in deformation including unrealistic deformation. On the other hand, notion of deforming force is not supported by the approach, and interaction force can not be defined.

Finite element method (FEM) and boundary element method (BEM) has been used in the field of computational dynamics, and there is research that introduces these methods to improve reality in computer graphics, such as Terzopoulos et al. (1987). These methods provide the means to implement precise models strictly based on dynamics of continuum. However, generally it is difficult to perform real-time simulation using models of practical complexity; although computation cost is drastically reduced by using static linear model by James & Pai (1999); K.Hirota & T.Kaneko (2001), as stated in section 1, the approximation also reduce reality of deformation. There are studies that accelerate the computation by both using advanced hardware such as GPU by Goeddeke et al. (2005) and improvement of the model structure.

Some other models such as sprig-mass network model (or, Kelvin model) and particle model are other candidates. Sprig-mass network is a model that approximates elasticity by using the network of spring. There is research that has applied this model to represent breakage in computer graphics by Norton et al. (1991), and also employed for haptic rendering. This model is preferably solved using an explicit method that apparently attains higher update rate of computation. However, it should be noted that deformation on each update cycle is not necessarily a precise solution of the model. This problem of solving method deteriorates reality of dynamic deformation. The particle model is considered to have similar problem of computation, however, the model is advantageous in that it is capable of representing plasticity and relatively large deformation of object which FEM model has difficulty of handling.

2.3.2 Record reproduction-based approach

One approach to solve the problem of computation cost is generating the response of objects based on measured or precomputed patterns of deformation rather than simulating it in real time. This idea has already been applied to presentation of high-frequency vibration of surface that is caused by collision with other object.

Wellman & Howe (1995) carried out pioneering research of this approach. In their research, the vibration of a real object that is caused by tapping was measured and approximately represented by fitting decaying sinusoidal wave, and the vibration wave was retrieved in virtual tapping operation. It was proved that this feedback of vibration is helpful to for users to discriminate materials.

Okamura et al. (1998) expanded this approach to other types of interaction including stroking textures and puncture; their approach is called *reality-based modeling*. Also, in their successive research in Okamura et al. (2000), they proposed an approach to optimizing parameters of vibration based on psychological evaluation on reality.

A similar research has been carried out by Kuchenbecker et al. (2005), where transient force at the beginning of contact is precomputed and then retrieved in interaction.

Above researches were focusing on improving realty of the sensation of contact and not dealing with macro deformation. On the other hand, in application that requires a realistic representation of deformation, approaches to measuring characteristics of deformable objects based on measurement are investigated.

Pai et al. (2001) proposed an approach to constructing virtual object model based on measurement on real object; regarding deformation model, stiffness matrix for linear elastic model is estimated based on force-deformation relationship while interacting with the real object. Also, real-time presentation of deformation is realized using an accelerated computation method for linear elastic model by James & Pai (1999).

It is generally accepted notion that the update rate of approximately 1kHz is required for usual haptic rendering, and at lowest several hundred hertz even in case of presenting a low stiffness object. One of solution for the problem is employing pre-recording or pre-computing approach.

James & Fatahalian (2003) have proposed an approach that uses precomputed trajectory of object state in state space; state transition sequences at a given initial state and force conditions are pre-computed, and there transition sequences are reproduced when these initial conditions are satisfied. In the research, however, little discussion has been made regarding increase in interaction patterns; it is not clear if this approach is applicable to realize arbitrary interaction with deformable objects.

In this chapter, as a novel approach that accommodates large DoF of interaction, impulse response deformation model (IRDM) is presented. IRDM is based on the idea of defining the relationship between input force and output deformation using impulse response; by assuming linear time-invariant model and precomputing impulse response of the system, resulting deformation is computed by convolution of input force and the impulse response.

2.4 Separate computation of deformation and motion

Use of a floating coordinate system is a common approach to define movable objects in virtual environments; scene graph is considered as a generic expansion of this approach, and it has been employed to various graphic and haptic rendering systems such as *GHOST SDK Programmer's Guide* (2002); Rohlf & Helman (1994).
In this chapter, a supplemental idea that realizes non-grounded motion of the deformable model is also presented. A floating coordinate system is introduced to our approach, and motion and deformation is simulated by motion equation and IRDM, respectively.

3. Impulse response deformation model (IRDM)

In this section, details of impulse response deformation model (IRDM) is discussed.
The idea of the IRDM is based on the premise that the model is linear, which means that the influences caused by impulse forces on different degrees of freedom or at different times are independent of each other, and the resulting deformation is computed as the sum total of the influences. The linearity regarding degree of freedom is a frequently employed assumption. For example, a linear elastic model is based on this idea. Also, the approach to compute the response of the system by the convolution of impulse response and input signals is commonly used. This approach implicitly premises temporal linearity.
Although, in a precise sense, real material is not thought to have exact linearity, in most applications, this assumption will provide more merit in reducing computational cost than the demerit of increasing inaccuracy. In a case where the assumption is not employed, the response of the object for the entire combination of the object status (i.e. position in phase space) and interaction status (i.e. boundary condition) must be defined. If these statuses are discretely described, the number of combinations of the discrete status is thought to explode even in models of relatively small complexity.

3.1 1 DoF model

Let us think of a continuous system with one force input and one displacement output. The impulse response of the system is defined as temporal sequence of deformation after the impulse force was inputted into the system. If the system is linear, then the resulting displacement $u(t)$ in response to arbitrary force input sequence $f(t)$ is obtained using the impulse response of the system $r(t)$ as follows:

$$u(t) = \int_0^\infty r(s) f(t-s) ds. \tag{1}$$

When $f(t)$ is a Dirac delta function, resulting $u(t)$ becomes identical with $r(t)$.
In the case of the discrete system, the formula is transformed as follows:

$$u^{[t]} = \sum_{s=0}^{T-1} r^{[s]} f^{[t-s]}, \tag{2}$$

where the variable inside bracket is the index of discretized time step. Also, in the formula, the length of time sequence of impulse response has been limited to finite time step T.
Generally, in case of interaction with a deformable object, the interaction point indicated by the haptic device causes boundary condition that fixes displacement on the point, and interaction force on the point unknown and left to be solved.
In the equation above, $f^{[t]}$ is unknown and $u^{[t]}$ is given, hence $f^{[t]}$ is obtained by:

$$u^{[t]} = r^{[0]} f^{[t]} + \tilde{u}^{[t]}, \tag{3}$$

where $\tilde{u}^{[t]}$ represents current (i.e. at time step t) displacement that has been caused by past sequence of force, which is defined by:

$$\tilde{u}^{[t]} = \sum_{s=1}^{T-1} r^{[s]} f^{[t-s]}. \tag{4}$$

In practical computation of interaction, all past sequence of force is known, and value of $\tilde{u}^{[t]}$ is computable. By solving Equation 3 for $f^{[t]}$, the interaction force is obtained.

3.2 Multiple DoF model

Let us suppose a system with n DoF. In the discussion below, force inputs and displacement outputs are noted using $n \times 1$ vecors $F^{[t]}$ and $U^{[t]}$. Also, impulse response of the system is represented by $n \times n$ matrix $R^{[s]}$. Similarly to 1 DoF model, the input-output relationship is formulated by:

$$\begin{aligned} U^{[t]} &= \sum_{s=0}^{T-1} R^{[s]} F^{[t-s]} \\ &= R^{[0]} F^{[t]} + \tilde{U}^{[t]}, \end{aligned} \tag{5}$$

where

$$\tilde{U}^{[t]} = \sum_{s=1}^{T-1} R^{[s]} F^{[t-s]}. \tag{6}$$

In usual haptic interaction, it is a peculiar case that fixed boundary condition is applied to all DoF of the model; in most cases, the number of haptic interaction points are limited to a small number, hence the DoF with a fixed boundary condition is also limited to similar number. Interaction forces on these fixed DoFs become unknown, and also displacements on other DoFs are unknown.
The difference of boundary conditions is more clearly represented by transforming Equation 6 as follow:

$$\begin{pmatrix} U_o^{[t]} \\ U_c^{[t]} \end{pmatrix} = \begin{pmatrix} R_{oo}^{[0]} & R_{oc}^{[0]} \\ R_{co}^{[0]} & R_{cc}^{[0]} \end{pmatrix} \begin{pmatrix} F_o^{[t]} \\ F_c^{[t]} \end{pmatrix} + \begin{pmatrix} \tilde{U}_o^{[t]} \\ \tilde{U}_c^{[t]} \end{pmatrix}, \tag{7}$$

where suffix o and c indicate values on free and fixed nodes, respectively. The equation is solved for unknown values $F_c^{[t]}$ and $U_o^{[t]}$ as follows:

$$F_c^{[t]} = (R_{cc}^{[0]})^{-1} (U_c^{[t]} - \tilde{U}_c^{[t]}), \tag{8}$$

$$U_o^{[t]} = R_{co}^{[0]} F_c^{[t]} + \tilde{U}_o^{[t]}. \tag{9}$$

3.3 Interpolation of force on triangular patch

In the implementation of the algorithm that will be discussed in section 5, the proposed computation method is adapted to models whose geometry is represented by triangular mesh. Suppose the contact point p is found on a patch that has vertices p_1, p_2, and p_3, and the interface point is causing displacement $\boldsymbol{u}_p$. In our implementation, firstly, the reacting force in the case when the displacement is caused on each of these vertex nodes. Such force is computed using equation 8; we describe these forces as $\boldsymbol{F}_{p_1}$, $\boldsymbol{F}_{p_2}$, and $\boldsymbol{F}_{p_3}$. Next, by multiplying a weighting factor to each of them, we determined the force applied to those nodes:

$$\boldsymbol{f}^{[t]}_{p_1^{[t]}} = \alpha_{p_1}\boldsymbol{F}_{p_1}, \quad \boldsymbol{f}^{[t]}_{p_2^{[t]}} = \alpha_{p_2}\boldsymbol{F}_{p_2}, \quad \boldsymbol{f}^{[t]}_{p_3^{[t]}} = \alpha_{p_3}\boldsymbol{F}_{p_3}, \tag{10}$$

where α_{p_1}, α_{p_2}, and α_{p_3} are the area coordinates (or barycentric coordinate), and has relationship as $\alpha_{p_1} + \alpha_{p_2} + \alpha_{p_3} = 1$. Using the result, the feedback force is computed as reaction of the sum of the forces applied to the nodes:

$$\boldsymbol{F}_p = -(\boldsymbol{f}^{[t]}_{p_1^{[t]}} + \boldsymbol{f}^{[t]}_{p_2^{[t]}} + \boldsymbol{f}^{[t]}_{p_3^{[t]}}). \tag{11}$$

The result of this implementation when the interface point is interacting on a node is identical with the result of equation 8. Also, the resulting feedback force is continuous on the boundary of a triangular patch, or on edges and nodes.
Finally, the displacement on entire nodes of the model is computed by:

$$\tilde{\boldsymbol{u}}^{[t]}_{k^{[t]}} = \sum_{s=0}^{T-1}\sum_{i=1}^{3} R^{[s]}_{p_i^{[t-s]}k^{[t]}} \boldsymbol{f}^{[t-s]}_{p_i^{[t-s]}}. \tag{12}$$

3.4 Complexity of computation

Generally, computation of Equation 8 becomes easy if the number of fixed DoF (i.e., DoF with fixed boundary condition) is small. In cases where DoF of a model is n and number of fixed DoF is n_c, $R^{[0]}_{cc}$ becomes a $n_c \times n_c$ matrix. If the inverse of the matrix is computed using simple Gauss elimination method, the order of the computation is $O(n_c^3)$. On the other hand, the order of computation cost of $\tilde{U}_c$ and $\tilde{U}_o$ are estimated to be $O(n_c^2 \cdot T)$ and $O(n \cdot n_c \cdot T)$ respectively, considering that all of $F^{[t]}$ other than n_c components is 0 for all past and present time t.
Amount of memory that is required to store impulse response matrix is $O(n^2 \cdot T)$, and $O(n_c \cdot T)$ to store past force boundary conditions.

4. Simulation of motion

Impulse response data of IRDM is obtained through simulation of deformation caused by impulsive force. This process of precomputation causes problems in cases when the object is not fix on the ground. Interaction with non-grounded objects causes motion of the entire body of the object that lasts for a long time, and representation of the motion of an entire body is not suited for IRDM.
Let us think a method to deal with non-grounded deformable objects using IRDM. For example, in a case where a deformable object is manipulated and pinched by the user, it becomes unclear whether the displacement on the surface is derived from motion of object as a whole or deformation of the object. It is impossible to represent the motion component that causes

permanent displacement using the IRDM model. Therefore, a computation method that separates these components apart and simulates motion and deformation is necessary.
In this section, a supplemental idea that realizes non-grounded motion of the deformable model is presented.
As stated in section 3, the IRDM is based on the premise that the model is linear, however, in a precise sense, motion and deformation of deformable object must be solved as a non-linear coupled problem. For example, a spinning object is deformed by centrifugal force, the deformation can cause change in an inertia moment, and the change affects the motion of rotation. It is impossible to represent this non-linear coupled model using a linear model.
Fortunately, this non-linearity is not considered to be significant in usual interaction using hand, hence in our approach, it is assumed that motion and deformation can be separately computed. Deformation and rigid motion of an object imposed by interaction force are computed separately, and then the resulting behavior is obtained by adding then together. The deformation and motion are simulated by using IRDM and solving equation of motion respectively.

4.1 Separate simulation of motion and deformation

Our approach to integrate motion and deformation models is illustrated in Figure 1. In the pre-computation process, as stated previously, the behavior of deformable objects in response to impulsive forces is simulated using FEM program. Since the object is non-grounded or floating in space, the impulsive force causes translational and rotational motion of the entire body as well as deformation from its original shape. Our approach deals with the components of motion and deformation separately. The component of deformation is represented by IRDM; the component of motion is approximately retrieved by solving equations of motion, hence there is no need of recording the component. In the interaction process, components of motion and deformation are computed separately based on common interaction force and then added together to obtain the resulting behavior.

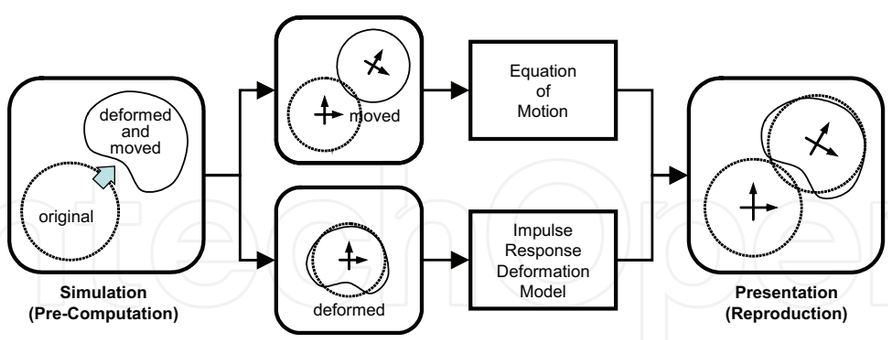

Fig. 1. Integration of motion and deformation model

4.2 Process of pre-computation

As stated in section 4.1, objects motion consists of translation and rotation. Regarding translation, the motion of the center of gravity of the object is equal to the motion of point mass that has identical mass with the object. Because of this equivalence, the translation of object is obtained by computing the center of gravity at each time step.

Regarding the rotation of the object, an estimation algorithm based on geometric matching was employed. The algorithm seeks rotation that minimizes the mean square error of node positions when the deformed object is approximately represented by a non-deformed model. The deformation component is obtained by subtracting the translational and rotational component motion from the result of the simulation. By performing the process to all combinations of DoF, the impulse response matrix $\boldsymbol{R}^{[s]}$ is determined.

4.3 Process of presentation

As stated above, the deformation component and interaction force is computed using IRDM. Then based on the interaction force, the component motion is computed by numerically solving initial-value problem of the motion equation (i.e., Newton's and Euler's equations):

$$m\frac{d\boldsymbol{V}}{dt} = \sum \boldsymbol{F}_{ext} \tag{13}$$

$$\boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}) + \boldsymbol{I}\frac{d\boldsymbol{\omega}}{dt} = \sum \boldsymbol{\tau}_{ext}. \tag{14}$$

where M is the mass of the entire body, $\boldsymbol{I}$ is inertia tensor, $\boldsymbol{V}$ and $\boldsymbol{\omega}$ are velocity and angular velocity of the rigid body respectively, and $\boldsymbol{F}_{ext}$ and $\boldsymbol{\tau}_{ext}$ are external force and torque around the center of gravity that are operated by the user. As stated above, in our approach, mutual influence between rotation and deformation of the object is ignored. The computation cost of IRDM is dominant in the total computation cost of this approach; hence the computational advantage of IRDM is also inherited to this approach.

5. Experiment

This section describes experiments that evaluate feasibility and computation cost of deformation and interaction using IRDM.

5.1 Deformation

5.1.1 Pre-computation

Pre-computation is the process that computes impulse response data though deformation simulation; impulsive force is applied to each of all degrees of freedom and deformation response on each of all degrees of freedom is recorded. Impulse response matrix R is obtained as a collective of the data. Dynamic deformation of the model is simulated by using the FEM model that consists of tetrahedral elements.

Three models of different complexity, as shown in Figure 2 were used for the evaluation: *cat*, *bunny*, and *cuboid*; complexity of these models are summarized in Table 1. Fixed boundary condition was applied to nodes on the bottom surface patches of the models; in order to fix the models to the ground. Height of the *cat* and *bunny* models is approximately 20cm, Height and width of the *cuboid* model is 20cm and 10cm respectively. Physical parameters of all of these models were defined as: Young's modulus $E = 2000\mathrm{N/m^2}$, Poisson's ratio $\nu = 0.49$, and density $\rho = 110\mathrm{kg/m^3}$.

Impulse response was recorded for one second at a sampling rate of 500 Hz, hence each impulse response wave in the impulse response matrix consists of 500 point sample values. Time step of FEM simulation was changed accordingly to the velocity of object deformation from 0.1 to 2 ms. Computation time of FEM simulation that is required to obtain the entire impulse response matrix for each model is shown in Table 1, where in house FEM routine by Pentium 4 3.0GHz processor was used.

An example of impulse response of *cuboid* model is shown in Figure 3, where an impulsive downward force has been applied on the node that is indicated by an arrow. Surface elastic wave starts to diffuse from the node and propagate to entire body within approximately 16 ms.

(a) cat (b) bunny (c) cuboid

Fig. 2. Experimental models

	cat	bunny	cuboid
free nodes (n)	359	826	1068
triangle patches	1796	3592	2178
entire nodes	690	1894	3312
tetrahedral elements	2421	8283	13310
pre-computation time (hr)	13.3	126.2	508.7
data size (GB)	4.3	22.8	38.2

Table 1. Complexity of models

5.1.2 Interaction

Experiments to evaluate interaction with models were carried out. Blockdiagram of the experimental system is shown in Figure 4. The system consists of PC1 (CPU:Itanium2 1.4GHz×4, memory:32GB, OS:Linux) that is in charge of model computation, PC2 (CPU:Pentium3 500MHz×2, OS:Windows) that serves as controller of two PHANToM devicesMassie (1996); All computation related to the IRDM model is performed by PC1. Computation of force and deformation are executed asynchronously using thread mechanism; these computations are noted as force process and deformation process respectively in the rest of this paper.

In the force process, firstly interaction point information is received from the Ethernet interface, next collision of the point with the surface of object model is detected, then interaction force on the point is computed, history of interaction force is updated, and finally the interaction force is output to the sent to PC2 through the Ethernet interface. Collision between the interaction point and the object surface is computed using an algorithm that is similar to God-Object MethodZilles & Salisbury (1995); this algorithm fits with our implementation because it eliminates ambiguity of the interaction point and provides unique displacement value. This force process is repeatedly executed every 2 ms, or at a rate of 500 Hz.

Deformation process computes deformation of an object using the history of force computed by the force process. As stated before, the impulse response matrix is a relatively large data set, and the matrix must be held on the main memory while force and deformation processes are executed. As suggested by Table 1, the data size of the *cuboid* model exceeds the size of

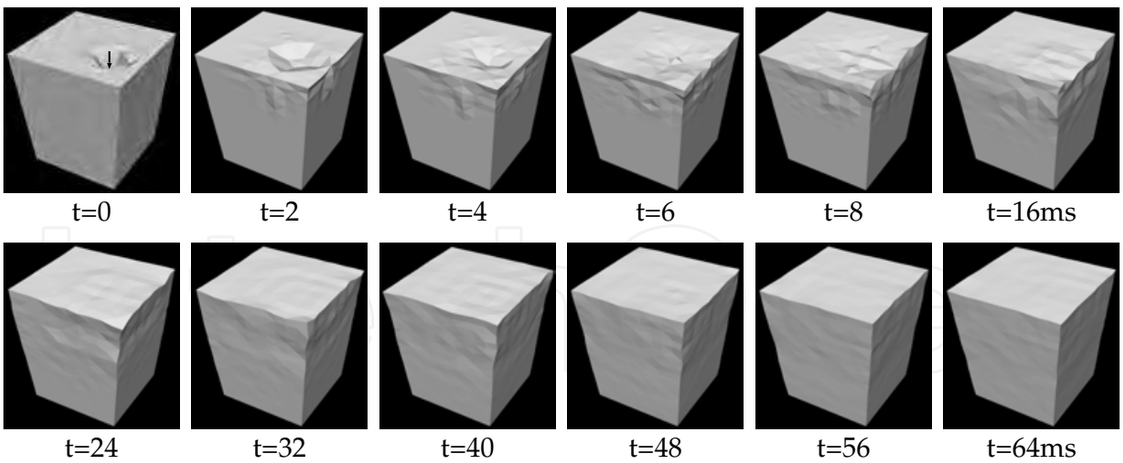

Fig. 3. Examples of impulse response

main memory of PC1, hence only half of the data where interaction force is applied to nodes on the upper half of the model were loaded on the main memory, and the area of interaction by the user was limited to these upper half nodes.
Program of force and deformation processes running on PC1 was optimized by performance using Intel Compiler and Performance Libraries. Deformation process was implemented using Math Kernel Library, parallelized by OpenMP Compiler, and three CPUs were allotted to the computation.
PC2 serves as a local controller of the PHANToM device, it simply works as bidirectional translator between the PHANToM device and Ethernet (TCP/IP) connection with PC1. Control of the device is implemented using GHOST library; control process of the library is executed at 1kHz, and in the process, the latest data that is received from the Ethernet interface is set to output force and the current position of interface point received from the device is sent back to the Ethernet interface.

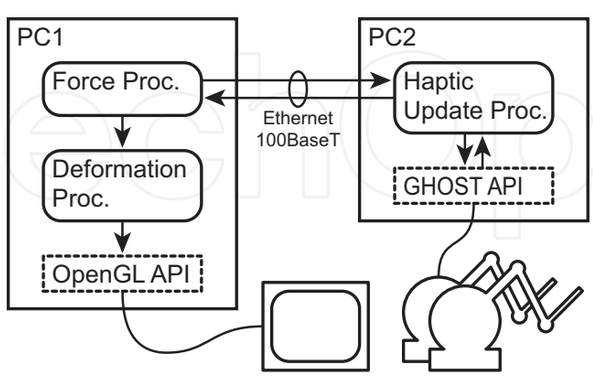

Fig. 4. System block diagram

5.1.3 Experimental Results

Figure 5 shows examples of interaction with a deformable object, where dynamic deformation is presented by a sequence of images. Since it was impossible to store images in real time, these images were generated off-line using the history of the interaction force; the arrow in the first image of each sequence indicates the point of application of force.

In figure 5(a), relatively quick motion of the *cat* model after releasing force that had been applied on a node. Figures 5(b) and (c) show the vibration of the *bunny* model that is caused by different interaction; the model was released after being pulled near and right in (b) and (c) respectively. It should be noted that different a vibration mode is presented according to different ways of interaction.

Figure 5(d) shows the deformation of *cuboid* model by step input of displacement; the force is applied to a node that is identical with the node where impulse force was being applied in Figure 3. Also, interaction force during the operation is plotted in Figure 6(a). Because of the nature of the dynamic model, interaction force gradually approaches a balance point while vibrating around the point.

Interaction using two interaction points is presented in Figure 5(e), where the user is pushing on the left and right side of the face of the *cat* model. Interaction force during the operation is plotted in Figure 6(b). As displacement on the right side increases, interaction force on the left side is also increasing.

Finally, change of interaction force while the user traced the back of the *cat* model from neck to tail is plotted in Figure 6(c). The plot suggests that interaction force is smoothly changing all through the interaction. Although invisible from the plot, subtle vibration is felt during contact with the object. The vibration is considered as an artifact that derives from sampling rate of IRDM model, which is 500Hz in our current implementation. The vibration is thought to be diminished by raising the sampling rate of the model in future implementation.

Evaluation of computation time is listed in Table 2. Computation of the interaction force comprises the evaluation of 8 for 3 to 9 times. Overhead of collision detection, communication, and graphic rendering is not included in values on the table. The computation of force is sufficiently fast for haptic presentation in that it is performed within 0.5ms per cycle even in case of using two interaction points.

Regarding deformation computation, real-time update of graphics at full video rate was not attained. For example, in the case of the *bunny* model, the update rate deteriorated to approximately 10 Hz. In spite of the low update rate, interaction was not felt greatly unreasonable subjectively, probably because the interaction is depending on information of force that is presented with less delay time.

	cat	bunny	cuboid
Computation of interaction force			
one-point	78	105	97
two-points	285	436	286
Computation of object deformation			
one-point	13040	33578	42614
two-points	26451	67339	85705

Table 2. Computation time (μs)

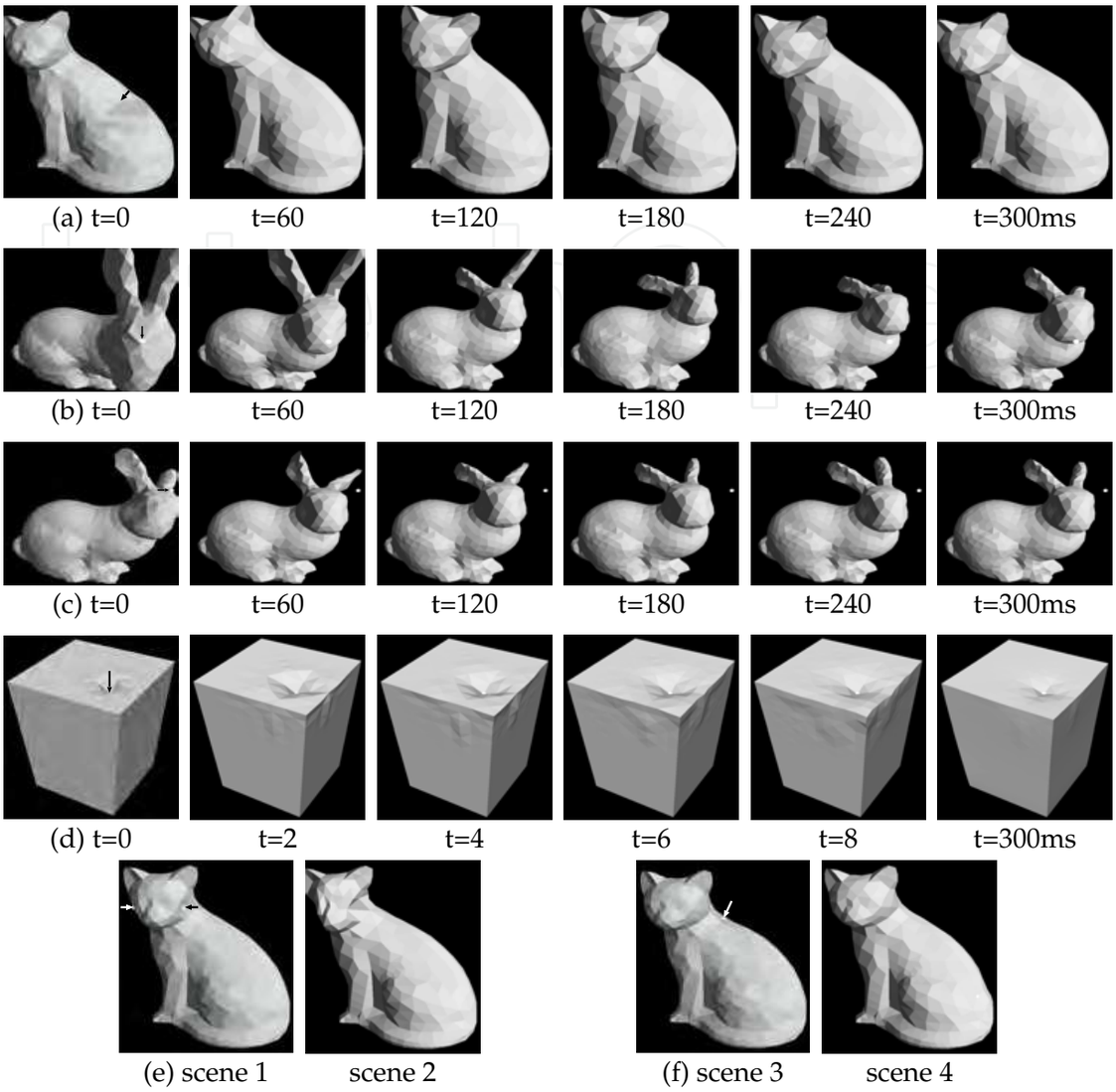

Fig. 5. Examples of dynamic deformation

5.2 Manipulation

5.2.1 Pre-computation

A *cube* model, 12cm on a side, as shown in Figure 7 was used for the evaluation; complexity of the model is summarized in Table 3. Physical parameter of the model was defined as: Young's modulus $E = 2000\text{N/m}^2$, Poisson's ratio $\nu = 0.49$, and density $\rho = 110\text{kg/m}^3$.

The computation time of FEM simulation that is shown in Table 3, where commercial FEM software (RADIOSS, Altair Engineering) with a Dual-Core Xeon 3.0GHz processor was used. Components of solid body motion and deformation were separated using the algorithm described in section 4 , and deformation component was stored as impulse response data.

Figure 8(a) shows that impulsive force is applied to the *cube* model; horizontal rightward force on the figure has been applied. Since the cube is floating, it starts moving while causing similar

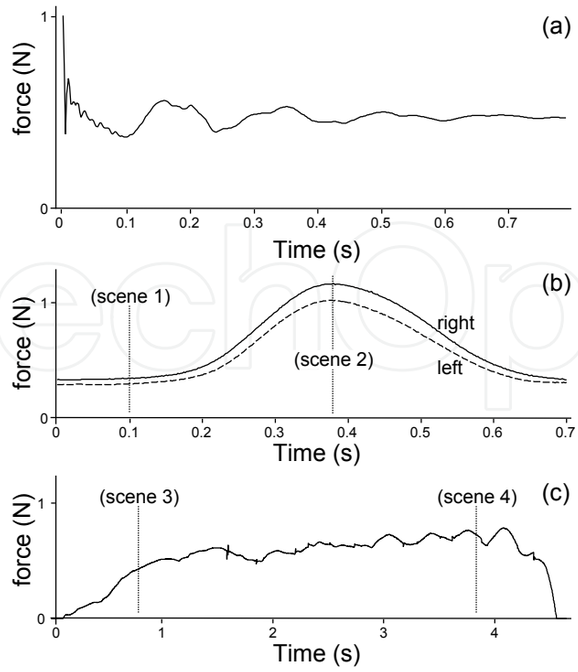

Fig. 6. Interaction force

deformation to the *cuboid* model. Figure 8(b) shows motion and deformation components separately extracted from (a).

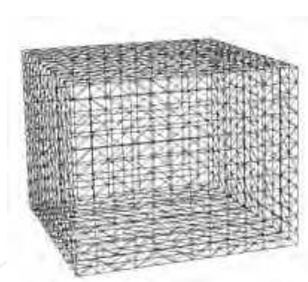

Fig. 7. Experimental model

5.2.2 Experimental Results

Figure 9 shows an example of a manipulating object; similarly to Figure 5, it presents sequences of images that were generated off-line.

In Figure 9(a), the user is picking up the top of a *cube* model and swinging right and left. Interaction force and motion of center of gravity of the object during the operation is plotted in Figure 10. The center of gravity motion is approximately sinusoidal, hence if the object is rigid, interaction force is expected to show similar sinusoidal change. However, the actual force is apparently causing oscillation at a different frequency. This fact suggests that the object is vibrating at its natural vibration frequency.

free nodes (n)	866
triangle patches	1728
entire nodes	1360
tetrahedral elements	5309
pre-computation time per d.o.f. (s)	1462
data size per d.o.f. (MB)	9.9

Table 3. Complexity of model

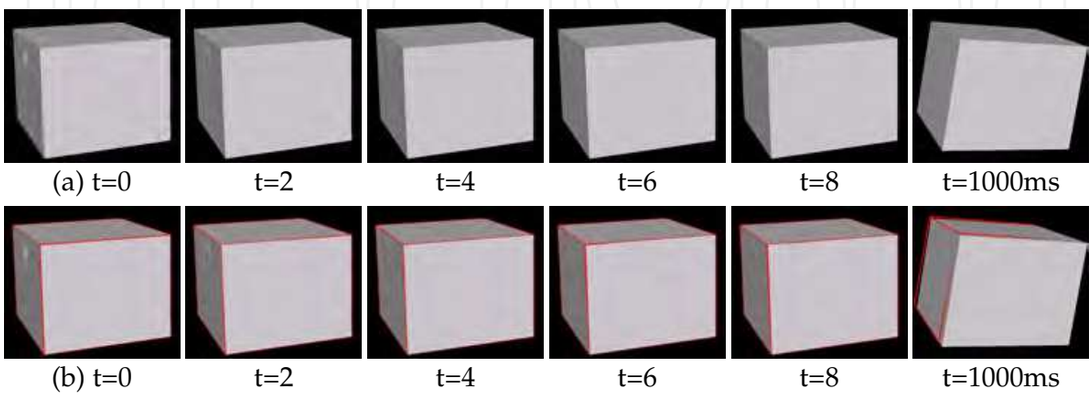

Fig. 8. Example of impulse response

Figure 9(b) shows a case where the user is tapping on a node of the *cube* model. The effect of both impact of collision and inertia of the object is reflected in the deformation and motion of the object; also, similarly to interaction with grounded models, relatively quick deformation is represented.

Figure 9(c) presents another example of interaction where the user is swirling the object along an elliptic orbit whose lengths of major and minor axes were approximately 6cm and 3cm respectively. Deformation that is caused by centrifugal force is represented naturally. Also, in the author's subjective impression, interaction force was realistic and reasonable.

6. Disucssion

6.1 Computation cost

As stated previously, computation complexity of the proposed method is independent of the DoF of the entire model n and proportional to the DoF of fixed boundary condition n_c. Experiments above have proved that, in cases when n_c is small, it was possible to compute interaction force in real time. Computation cost of deformation is $O(n^1)$ and the feature of the approach was also verified through experiments.

In cases of solving deformation by FEM, its computation cost depends on the algorithm of the solver program. The order of the computation of simple Gauss elimination method is $O(n^3)$, and even in case of using iterative algorithm such as Gauss-Seidel method, the order of computation is approximately $O(n^2)$. This fact suggests that our approach is advantageous as n becomes large.

Actually at present complexity of the model, the computation time that was required for pre-computation process suggests that it is difficult to perform the FEM simulation in real time,

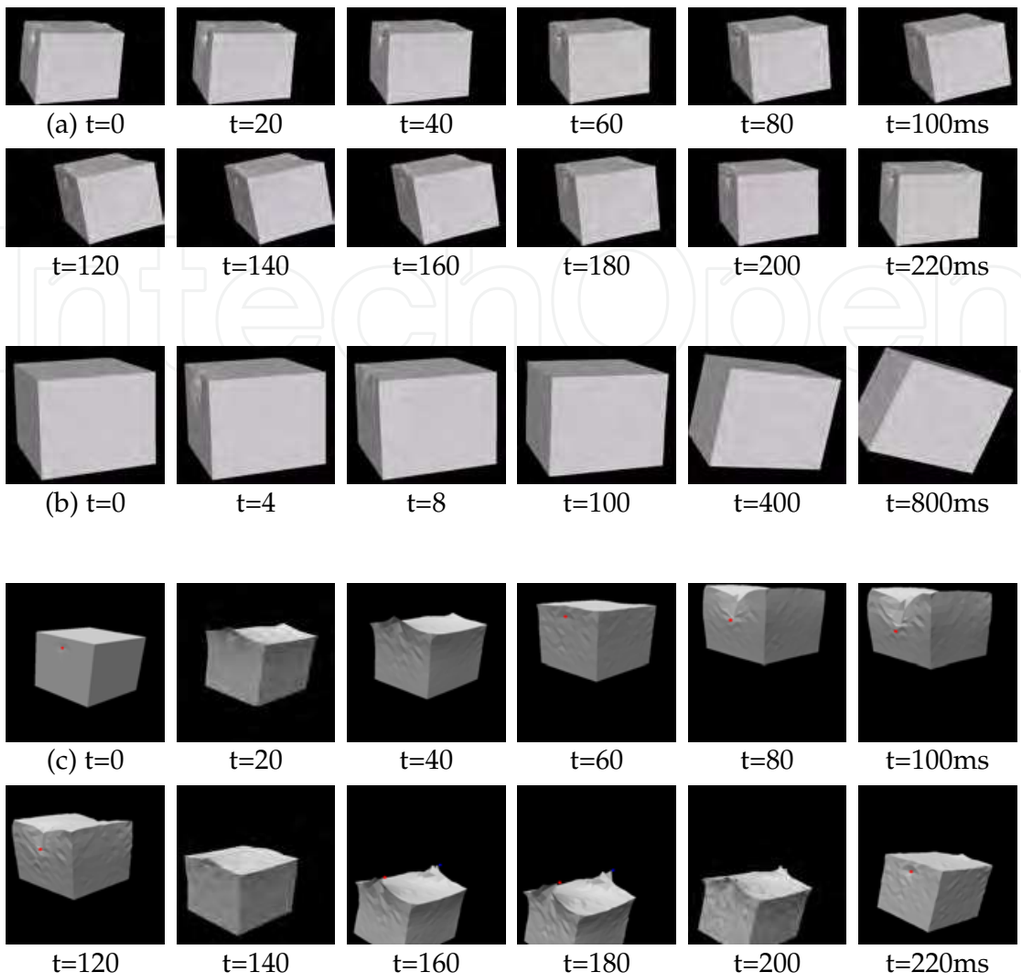

Fig. 9. Example of dynamic motion and deformation

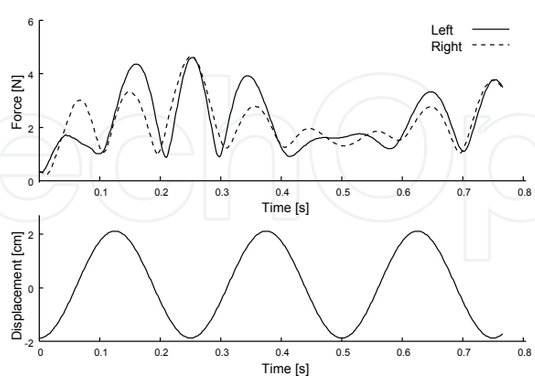

Fig. 10. Interaction force

although the FEM program that was employed for the computation was not aimed at real-time simulation.

One idea to reduce deformation computation cost is approximately generating deformed shape using reduced number of nodes; reduction of the number of nodes almost proportionally reduces computation cost, and interpolation using curved surface contributes to presentation of smooth surface. Another idea is accelerating the computation process using an advanced computing environment such as GPU. Our preliminary study is suggesting that the computation of the IRDM model is well suited to parallel computation using GPU.

6.2 Memory consumption

Regarding memory consumption, IRDM of present implementation requires a relatively large amount of memory and not applicable to practical application. Data compression method to solve this problem has been investigated, and our preliminary experiment suggests that it is possible to compress the data to approximately one-hundredth of original size by taking advantage of similarities of impulse response waves related to nodes that are geometrically close each other.

This compression method is expected to expand the area of application. For example, the size of IRDM data of the *cat* model is approximately 4GB. Since the data must be held in main memory during interaction, the computer that is available for the interaction is limited to relatively high specification machines. Also, the data size is somewhat too large to transmit over the Internet. If the data is compressed to 40MB, it is easily handled using most current computer systems and network connections.

6.3 Evaluation using subjects

Finally, evaluation of reality becomes an important topic of research, and as a basis for the research, methodology to quantify reality of dynamic interaction with deformable object must be established.

7. Conclusion

In this chapter, a novel approach to implement real-time interaction with deformable objects was presented. A core idea of the approach is modeling deformation using a set of impulse response data and computing deformation by convolution of interaction force with the model. The idea was experimentally implemented and evaluated through experiments. Also, an extension of the model to represent non-grounded object is discussed, by which manipulation of deformable object was enabled.

Finally, it should be noted that our approach is just one implementation of precomputation-based deformation model. A model of this kind has problem of trade-off between number of precomputed interaction and reality of presentation. The problem may be alleviated by introducing assumptions that effectively prevent combinational explosion of interaction patterns and by compressing precomputed data based on similarity of response. Further investigation is needed to find better representation of precomputation-based models. We hope that this paper will stimulate the discussion for such investigation.

8. References

B.Mirtich & J.Canny (1995). Impulse-based simulation of rigid bodies, *Proc. Symp. Interactive 3D Graphics* pp. 181–188.

Borst, C. & Indugula, A. (2005). Realistic virtual grasping, *Proc. IEEE VR 2005* pp. 91–98.

D.Baraff (1989). Dynamic simulation of non-penetrating rigid bodies, *Computer Graphics* **23**(3): 223–232.

G.Burdea (1996). *Force and Touch Feedback for Virtual Reality*, A Wiley-Inter-Science Publication, New York.

GHOST SDK Programmer's Guide (2002). SensAble Technologies, Inc.

Goeddeke, D., Strzodka, R. & Turek, S. (2005). Ergebnisberichte des instituts fur angewandte mathematik, *Nummer 292, FB Mathematik, Universitat Dortmund* .

James, D. L. & Fatahalian, K. (2003). Precomputing interactive dynamic deformable scenes, *Proc. ACM SIGGRAPH 2003* pp. 879–887.

James, D. & Pai, D. (1999). Artdefo, accurate real time deformable objects.

J.E.Colgate, M.C.Stanley & J.M.Brown (1995). Issues in the haptic display of tool use, *Proc. IROS95* pp. 140–145.

K.Hirota & M.Hirose (2003). Dexterous object manipulation based on collision response, *Proc. IEEE VR 2003* pp. 232–239.

K.Hirota & T.Kaneko (2001). Haptic representation of elastic object.

K.Salisbury, D.Brock, T.Massie, N.Swarup & C.Zilles (1995). Haptic rendering: Programing touch interaction with virtual objects, *Proc. Symp. Interactive 3D Graphics* pp. 123–130.

Kuchenbecker, K., Fiene, J. & Niemeyer, G. (2005). Event-based haptics and acceleration matching: Portraying and assessing the realism of contact, *Proc. WHC 2005* pp. 381–387.

Massie, T. H. (1996). *Initial Haptic Explorations with the Phantom: Virtual Touch Through Point Interaction*, Master Thiese at M.I.T.

Norton, A., Turk, G., Bacon, B., Gerth, J. & Sweeney, P. (1991). Animation of fracture by physical modeling, *Visual Computer* **7**: 210–219.

Okamura, A. M., Dennerlein, J. T. & Howe, R. D. (1998). Vibration feedback models for virtual environments, *Proc. IEEE ICRA* pp. 2485–2490 (Vol.3).

Okamura, A. M., Hage, M. W., Cutkosky, M. R. & Dennerlein, J. T. (2000). Improving reality-based models for vibration feedback, *Proc. ASME DSCD* **DSC-Vol.69-2**: 1117–1124.

Pai, D. K., van den Doel, K., James, D. L., Lang, J., Lloyd, J. E., Richmond, J. L. & Yau, S. H. (2001). Scanning physical interaction behavior of 3d objects, *Proc. ACM SIGGRAPH 2001* pp. 87–96.

Rohlf, J. & Helman, J. (1994). Iris performer: a high performance multiprocessing toolkit for real-time 3d graphics, *Proc. ACM SIGGRAPH 94* pp. 381–394.

S.Hasegawa & M.Sato (2004). Real-time rigid body simulation for haptic interactions based on contact volume of polygonal objects, *Computer Graphics Forum* **23**(3): 529–538.

Terzopoulos, D., Platt, J., Barr, A. & Fleischer, K. (1987). Elastically deformable models, *Computer Graphics* **21**(4): 205–214.

T.W.Sederberg & S.R.Parry (1986). Free-form deformation of solid geometric models, *Computer Graphics* **20**(4): 151–161.

T.Yoshikawa, Y.Yokokohji & nad X.Z.Zheng, T. (1995). Display of feel for the manipulation of dynamic virtual objects, *Trans. ASME J. DSMC* **117**(4): 554–558.

Wellman, P. & Howe, R. D. (1995). Towards realistic vibrotactile display in virtual environments, *Proc. ASME DSCD* **DSC-Vol.57-2**: 713–718.

Zilles, C. & Salisbury, K. (1995). A constraint-based god object method for haptic display, *Proc. IROS '95* pp. 145–151.

Advances in Haptics

Edited by Mehrdad Hosseini Zadeh

ISBN 978-953-307-093-3
Hard cover, 722 pages
Publisher InTech
Published online 01, April, 2010
Published in print edition April, 2010

Haptic interfaces are divided into two main categories: force feedback and tactile. Force feedback interfaces are used to explore and modify remote/virtual objects in three physical dimensions in applications including computer-aided design, computer-assisted surgery, and computer-aided assembly. Tactile interfaces deal with surface properties such as roughness, smoothness, and temperature. Haptic research is intrinsically multi-disciplinary, incorporating computer science/engineering, control, robotics, psychophysics, and human motor control. By extending the scope of research in haptics, advances can be achieved in existing applications such as computer-aided design (CAD), tele-surgery, rehabilitation, scientific visualization, robot-assisted surgery, authentication, and graphical user interfaces (GUI), to name a few. Advances in Haptics presents a number of recent contributions to the field of haptics. Authors from around the world present the results of their research on various issues in the field of haptics.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Kazuyoshi Tagawa, Koichi Hirota and Michitaka Hirose (2010). Manipulation of Dynamically Deformable Object Using Impulse-Based Approach, Advances in Haptics, Mehrdad Hosseini Zadeh (Ed.), ISBN: 978-953-307-093-3, InTech, Available from: http://www.intechopen.com/books/advances-in-haptics/manipulation-of-dynamically-deformable-object-using-impulse-based-approach

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821