Software Architecture for Audio and Haptic Rendering Based on a Physical Model

Hiroaki Yano & Hiroo Iwata

University of Tsukuba, Tsukuba 305-8573 Japan

{yano,iwata}@kz.tsukuba.ac.jp

Abstract: This paper describes a software architecture solution for rendering audio and haptic sensation. By most of existing 3D sound systems, it is difficult to generate appropriate sounds from virtual objects, depending on the material used and the location of the impact. We developed a method that we named AudioHaptics to generate audio and haptic sensation based on a physical model of the virtual objects. However, audio and haptic devices are specialised pieces of equipment, of which the software is an intrinsic part. So it is difficult to change component parts of the devices and the virtual environment.
We propose new software architecture for the synthesis of haptic and auditory senses that we named IOA (Interaction Oriented Architecture). IOA supports various types of interface devices. Any sensors and displays present are easily reconfigured by exchanging modular software. The effectiveness of this architecture is tested by users.

Keywords: virtual reality, audio rendering, haptic rendering, software, physical based modelling

1 Introduction

Interaction with virtual objects is a common event in the virtual environment. In these situations, audio and haptic feedback plays an important roll in increasing the sensation of presence in the virtual environment. For example, in the real world, if we wished to know the inner structure of a wall or the material that it is made of we would hit it to feel the reaction force and hear the sound generated. We can recognise the material and inner structure of the object by the reaction force and the sound. In the virtual environment, it is quite natural that we should want to sense the audio and haptic sensation from virtual objects.

Currently, most virtual reality systems with 3D auditory feedback can produce a good sensation of presence to the users. They can generate sounds from a car on a virtual street or from a radio in a virtual room using reverberations from pre-recorded sounds. This can also be applied to urban environmental simulators or amusement systems etc.

However, when we hit a real object, the sound depends on the location of the contact and the shape and attributes of the object. It is difficult to generate these sounds without the actual pre-recorded sounds of the objects being available in the virtual environment. Also, the sound localisation of the sound source near the user is not fine-tuned in current 3D sound systems.

Doel and others have developed the Sonic Explore system, which supports these types of sounds based on the vibration dynamics of bodies using a surface model (polygonal model)(Doel et al,1998). This system can provide real time computational sounds from objects that are being stroked externally. However it cannot render internal attributes, such as the hollow interior of objects or objects in which the inner material is different from the surface, and nor can it support haptic feedback.

As a solution to these problems, we introduce a method called AudioHaptics. AudioHaptics supports the sound of collisions based on a physical model and increases the accuracy of sound localisation near to the user.

However, each sensation requires different rendering functions and data sets. The software of those systems is tightly linked with the device control programs, so it is difficult to reconfigure the virtual environment. Many types of software tool have been developed and commercialised. SIMNET (Pope,1989) has been developed by the US Army to simulate the ground in a battlefield by networked computers. SIMNET has now evolved into NPSNET (Zyda et al, 1992), which includes thousands of tanks and missiles and infantry.

Next generation prototype systems such as Paradise, DIVE, BrikNet are also under development (Singhal et al, 1999). The commercialised products WorldToolKit^{TM} and Cyberspace Developer Kit were released in the early 90's. These systems support VR interface devices such as HMD and pointing devices.

As software tools for haptic interfaces, GHOST, made by Sensable Technologies and ArmLib, made by UNC (Randolf et al, 1997) have been developed. Also, beginning in 1991, we developed the virtual environment construction systems, VECS (Iwata et al, 1993) and LXH (Iwata et al, 1997) for haptic devices. These software tools support various types of haptic devices. By dividing some of the modules, we can easily reconfigure them for haptic interfaces and virtual environments.

Some software tools for audio rendering based on pre-recorded sounds are available, such as the Direct sound library made by Microsoft. Doel's Sonic Explore (Doel et al, 1998), was developed as a model based audio rendering software.

However, these software tools do not support haptic and audio feedback simultaneously.

In this paper, we introduce a method that can generate sound based on the physical model of virtual objects. AudioHaptics supports volumetric models. By changing their physical model and parameters, we can generate sounds originating from virtual objects, which can have arbitrary shapes, attributes and inner structures.

We propose a software architecture for audio and haptic feedback based on physical models named IOA. IOA supports various types of haptic and audio interface devices. By exchanging modular software, we can easily reconfigure the system for any sensors and displays and any virtual environments. Therefore, we believe that we have developed an improved AudioHaptics environment. The effectiveness of this system has been evaluated by the users.

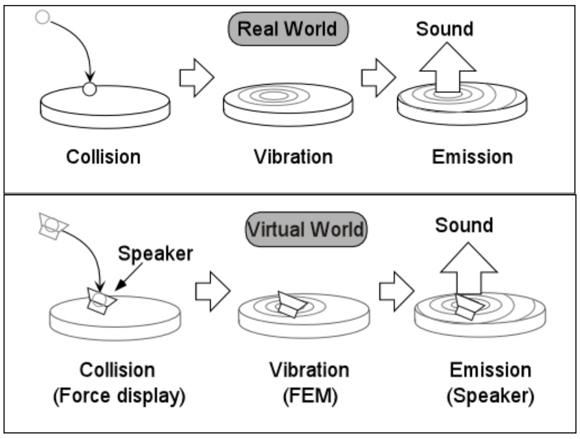

Figure 1: Basic concept of AudioHaptics

2 AudioHaptics

Sounds are transmitted by pressure fluctuations in air. These pressure fluctuations are caused by two things, "The vibration of objects", such as sounds of collisions or friction, and "aerodynamic effects" such as the sound of fumes etc. In this paper, we only deal with the vibration of objects. The occurrence of the vibration has three phases, "collision detection", "vibration calculation" and "sound emission", as shown in Figure 1.

As we described in the Introduction, generating the sounds of interactions entails the following problems.

1. The generation of sound depends on the shape, attributes, inner structure and location(s) of the impact.
2. Spatial localisation of sounds emanating from objects near to the user.

To solve problem No.1, we use the Finite Element Method (FEM) to obtain the vibration. Then, assuming that the vibration energy of the object transmits to the air without attenuation, we can calculate the sound pressure using the velocity potential. The FEM supports volumetric models, and therefore we can calculate any vibration theoretically with FEM, even if there are some hollows in the object. However, current computers do not have sufficient performance to calculate the vibration in real time. Therefore we need to calculate the vibration offline. The velocities of representative points on the object in various statuses are saved in files as digital data. When an interaction occurs between the user and the virtual objects, the sounds are calculated by the velocity data.

To solve the problem No.2, we fitted a small speaker at the end effecter of the haptic device. Figure 2 shows this speaker. If the co-ordinates of the real world and the virtual world have same scale, then the location of the speaker after the impact is near the impact location on the virtual object. We assumed that the speaker outputs the correct sound from the object virtually in the case where the size of the virtual objects are smaller than the human head and the distance between the objects and the human ears is more than 50cm (we assume that the user sits

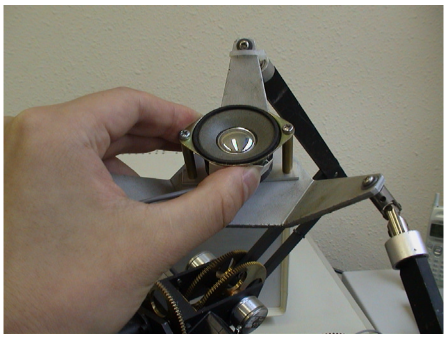
Figure 2: Speaker at the end effecter of HapticMaster

on a chair and the objects are on a table. The distance between user's ears and the objects are approximately 50 cm). If the size of the objects is larger than this, or if the relative distance between ear and objects is smaller then in that case we should use speaker arrays or combine existing 3D sound systems. Whichever is the case, the sound localisation is improved by using this configuration.

3 Basic Concept of IOA

We have developed various software tools for haptic devices(Iwata et al, 1993,1997). However, the specifications of audio and haptic rendering, such as update rate or data sets are quite different, and some data such as hand position are shared in each rendering. The rendering software should therefore work co-operatively. The software for audio and haptic rendering is tightly connected for each device, and it is difficult to reconfigure the devices and the virtual environment. It is necessary to develop the software architecture to support these sensations of rendering and for constructing virtual environments.

The requirements of the software architecture for AudioHaptics include:

- The ability to generate audio and haptic sensation simultaneously in real time.
- The ability to reconfigure the virtual world.
- The support of various types of haptic devices and audio devices.

To support these requirements, we designed an architecture for AudioHaptics called IOA. The basic strategy of IOA is:

(1) Dividing the software into various modules
(2) Implement the software using shared memory and separate the rendering processes.

Most events in the virtual environment are caused by interactions. The interactions define the behaviour of virtual objects and interface devices. We considered that the software architecture for AudioHaptics should be interaction oriented.

In this case, the sensor data are required for audio and haptic rendering. In our previous software tools, the haptic devices deal with a module which has both an input and output function. In IOA, the haptic devices are individually divided into sensor devices and display devices. We can easily configure these to share sensor data and to control many individual devices. In order to deal with these issues in AudioHaptics software, IOA (Interaction Oriented Architecture) is composed of the following seven modules: Sensor device driver, Recognition engine, Interaction server, Contents server, Model manager, Renderer and Display device driver as shown in Figure 3.

By dividing the system into these modules, the haptic devices, audio devices and the virtual environment are easily reconfigured.

The functions of these modules are:

1. Sensor Device Driver (S.D.D): the sensor device driver manages the sensor inputs. Various types of sensors can be connected by changing the S.D.D. Currently we implement position, angle and touch sensor devices.
2. Recognition Engine (R.E): the recognition engine calculates human behaviour using sensor data from the S.D.D. Currently, we implemented this function to get the position of end effecter, button input etc.
3. Model manager (M.M): the model manager manages virtual world model data, such as the shapes and attributes of virtual objects, environmental parameters etc.
4. Renderer: the Renderer generates the visual, audio and haptic sensations. It calculates the sensation value such as pixel colour, force and sound data by R.E data and M.M data.
5. Interaction Server (I.S): The interaction server manages the data path for co-operation among all the sensory feedback functions. The interaction server takes the data from the R.E and sends them to all modules that require that data. In the AudioHaptics system, audio, haptic and visual sensation should be generated simultaneously. The I.S sends hand position data from the R.E to the Contents server and to each M.M. The I.S includes a part of the R.E and the M.M to make a more complicated environment.

6. Contents server: To construct a more complex and high-level virtual environment that contains information such as physical laws or the autonomy function for virtual creatures and deformation calculations, we need support systems. For example, physical laws for the virtual world are contained in this module.
7. Display Device Driver (D.D.D): The display device driver generates sensation such as visual images, force and sounds using sensation value data from Renderer.

In addition, different update rates are required for each sensation. Haptic rendering requires approximately a 1kHz update rate. On the other hand, audio rendering needs more than an 8Hz update rate (this is not the sampling rate, Harada et.al 1998). The processes of IOA can separate according to the sensations.

4 Implementation

4.1 Hardware Configuration

The hardware configuration of our system is shown in Figure 5. For the haptic device, we use the HapticMaster (Asano et al, 1997). Our PC for rendering is equipped with a Pentium III 533EBMHz processor, 128MB memory and an SP401F YAMAHA 724 PCI sound card. The speaker at the end effecter of the HapticMaster is a full range 3.2Ω 4.5W 38mm diameter speaker. The operating system of this PC is Windows2000, and we developed all of the software using C language.

4.2 Basic Configuration of Software

The AudioHaptics environment can be divided into the audio rendering part, the haptic rendering part and "others".

For audio rendering, "collision detection", "calculation of the vibration of virtual objects" and "sound emission" functions should be implemented (Figure 6). We implemented the function of "collision detection" in the I.S. When a collision is detected, the object ID and the location of the collision are sent to the Model manager and the Contents server. The "calculator of vibration" is implemented in the Contents server. In this module the FEM module calculates the vibration of virtual objects. The shapes and attributes of the virtual objects, hand position, velocity and acceleration data are sent from the Interaction server to the Model manager. The "sound emission" function is implemented in the Sound model manager. It calculates the velocity potential at the speaker using the vibration data in the Contents server. Then the sound pressure at the speaker is calculated using the velocity potential. The sound pressure value data are sent to the Audio device driver and then the sound is output.

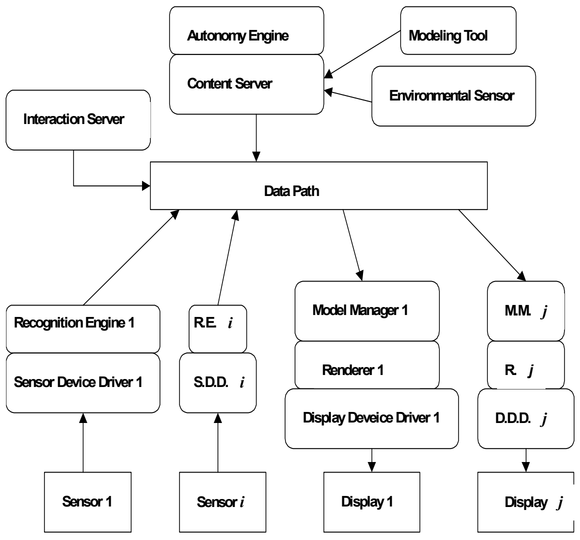

Figure 3: Basic architecture of IOA

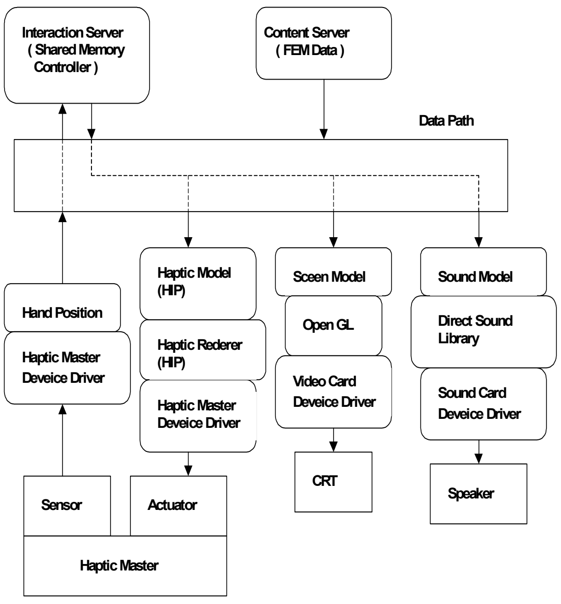
Interaction Server (Shared Memory Controller)
Content Server (FEM Data)
Data Path
Haptic Model (HIP)
Haptic Rederer (HIP)
Haptic Master Deveice Driver
Sceen Model
Open GL
Video Card Deveice Driver
CRT
Sound Model
Direct Sound Library
Sound Card Deveice Driver
Speaker
Hand Position
Haptic Master Deveice Driver
Sensor
Actuator
Haptic Master

Figure 4: Basic structure of IOA for AudioHaptics

Haptic rendering should have "collision detection", "force calculation" and "force output" functions. In our IOA architecture, these functions are implemented in the I.S, the Haptic Model manager and the Haptic Renderer, and the Haptic device driver respectively. The Collision detection module detects collisions using hand position data from the R.E. If a collision occurs, the Haptic model manager sends the data to the Haptic Renderer. The Haptic Renderer calculates the force and sends the force sensation value to the device driver. The device driver drives the haptic device using that data.

4.3 Simulation of Sound Pressure

In this system, simulation of audio rendering is divided into the Vibration module and the Emission module.

For the Vibration module we used commercially available FEM software, LS-DYNA made by Livermore Software Technology. In this module, the velocity data of each grid (Figure 7. shows sample mesh data) at each sampling time are calculated using location data of the interaction and shape and attribute data of the virtual object. The result of these calculations is saved in a file.

In the Emission module, the sound pressure values are calculated in real time. We assume that the vibration energy of the object transmits to the air without attenuation. We calculate the velocity potential by integrating the velocity of all of the minute areas on the object (Figure 8.). Then we can calculate the sound pressure value from a partial-differential-equation.

For example, we can consider a virtual aluminium disk, which has a 50mm radius, 12mm thickness, 7.03x10^{10} Pa Young's modulus and 0.33 Poisson's ratio. An impact occurs 25mm away from the centre of the disk. The FEM analysis is carried out using the mesh data shown in Figure 7. The velocity data of each grid at each sampling time are saved in a file. The file data is uploaded by the Emission module at the beginning of the program. When the impact occurs, the emission module calculates the velocity potential $\phi(t)$ at the speaker.

$$\phi(t) = \int d\phi$$
$$= \sum_{S} \frac{v(t-\frac{d}{c}, r, \theta)}{2\pi d} \Delta S$$

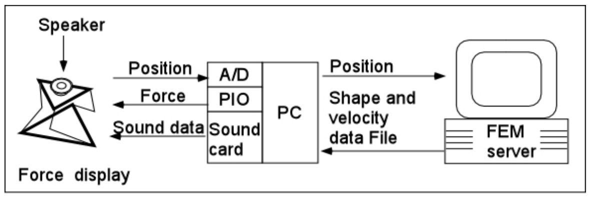

Figure 5: Hardware configuration

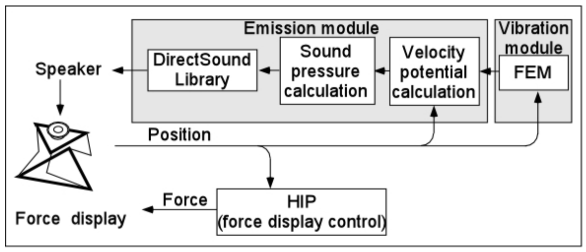

Figure 6: Software configuration

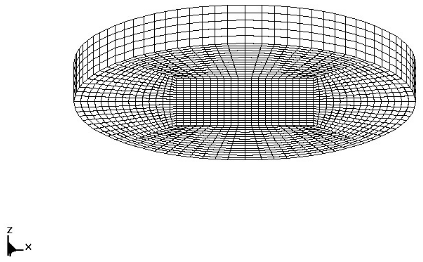

Figure 7: Backside of mesh data of the thin plate

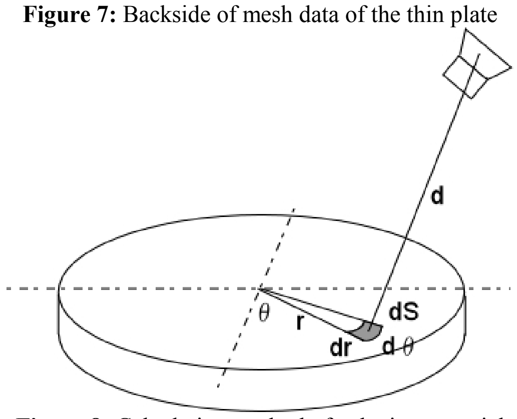

Figure 8: Calculation method of velocity potential

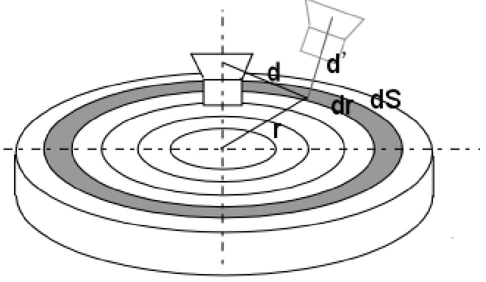

Figure 9: Calculation example of disk

$$= \sum_r \frac{v(t - \frac{d}{c}, r)\pi((r + \Delta r)^2 - r^2)}{2\pi d} \quad (1)$$

where v is the velocity of the minute area dS, d is the distance between the minute area and the speaker, r is the distance between the minute area and the centre of the disk and θ is the direction from the centre of the disk.

The sound pressure is calculated using the velocity potential $\phi(t)$.

$$p(t) = \rho \frac{\partial \phi}{\partial t} = \rho \frac{\phi(t) - \phi(t-1)}{\Delta t}$$

where ρ is the density of air (1.184kgm^{-3} at 25 °C), 0mm < r < 50mm and Δt is 1/44100 s. The magnitude of the frequency of the emission from the aluminium disk is about 1kHz, and the sampling rate is defined as 44.1kHz. However, the velocity data is simplified as shown in Figure 9. Because of the huge computational cost and memory space that are required for the calculation of the velocity potential using all grid data.

The haptic sensation is calculated using the spring model function (Iwata et al, 1997).

The haptic rendering and audio rendering modules are separated in two threads. The update rates are 1kHz and 16Hz respectively.

5 Evaluation

5.1 Comparison with Real Object

To show the effectiveness of the simulation, we conducted an experiment where we compared the results with a real object. We made an aluminium disk, as shown in Figure 10, and set up a microphone at a point 20mm from the centre of the disk. A 6mm radius steel ball was set at a height of 100mm and 25mm away from the centre of the disk, and was allowed to fall onto the surface. We recorded the sound of the impact with the system microphone. The sound data was recorded at a 44.1kHz-sampling rate with 16bit resolution, the same as in the simulation.

Figure 11 shows the time response of the real and the virtual sounds. The response of the real disk is not attenuated in the initial 40ms. The bounce of the ball dragged on compared to the simulation. However, the attenuation curve of the real disk is similar to the virtual one. Figure 12. shows the power spectrum of the sound pressures.

The results of the simulation show that the largest peak exists at 5.720kHz, and subsidiary peaks are seen at 3.668kHz, 12.484kHz, 12.944kHz and 15.168kHz. The results of the 'real' experiment show comparable peaks at 5.986kHz, 3.956kHz, 12.980kHz, 13.320kHz and 15.17kHz respectively. Each peak corresponds to its simulated counterpart within a 5% error. Subtle differences in the material attributes and the constant values used for the simulation are thought to have caused these minor differences. In addition, the 9.186kHz and 11.256kHz peaks in the real sound spectrum are not found in the simulation. The simplifications used in the simulation therefore cannot simulate whole complex sound phenomenon.

Figure 13 shows the initial 3ms time response from the impact. According to Figure 12 and Figure 13, the main ingredient of the real sound is the peak at 5.986kHz. On the other hand, the main ingredient of the simulation sound is at 5.720kHz, even though 12kHz sound exists in the initial 0.3ms of the simulation sound spectrum.

Whatever the case, the sound signatures of the real disk and the virtual disk mainly depend on sound at about 6kHz, and the IOA can support the AudioHaptics system.

5.2 Recognition of A Hollow Object

In this simulation, we use the FEM with a volumetric model. This means that the system can provide the sounds of hollow objects as well as solid objects. This experiment is conducted with a hollow and a

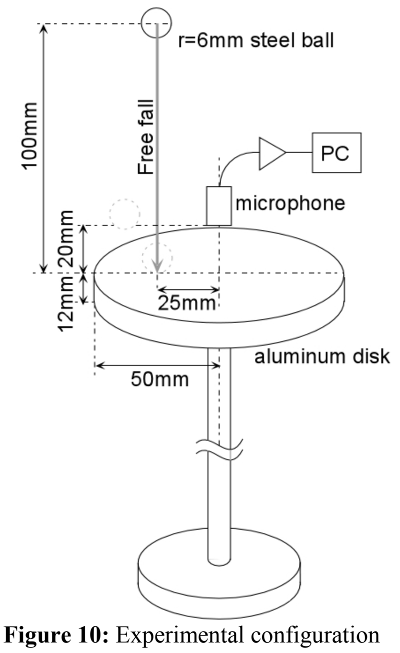

Figure 10: Experimental configuration

solid object, which both have the same appearance as far as the subjects are concerned.

We defined two 50mm radius and 25mm thickness virtual aluminium disks. One is solid (Figure 14.) and the other has an internal hollow of 80mm diameter and 13mm depth (Figure 15.). We created the sounds by hitting these disks 25mm away from their centres. The subjects identified them as solid or hollow objects. The force feedback by generated by HapticMaster is proportional to the intrusion into the virtual disk. The impact speed was not considered.

Six subjects conducted six respective trials. However, the subjects had not experienced hearing the sounds of these disks. We had prepared real aluminium disks and the subjects had heard the sounds of the real disks before the experiments. The solid disk makes a higher and clearer sound than the hollow disk.

The power spectrums of these disks in the simulation show the same tendency (Figure 16).

97% of the answers given by the subjects are correct in this experiment.

These results show that this system can provide the sound of objects even if they do not have a uniform inner structure, and that we can construct an audio and haptic environment based on physical models using the IOA.

6 Discussion

We propose a new software architecture for AudioHaptics named IOA. IOA can support other interface devices as well as audio and haptic ones. Multi-channel speakers, head trackers and any other visual display can be connected. Even though the data path should not be on the shared memory, it can be on the network data path. This means that the

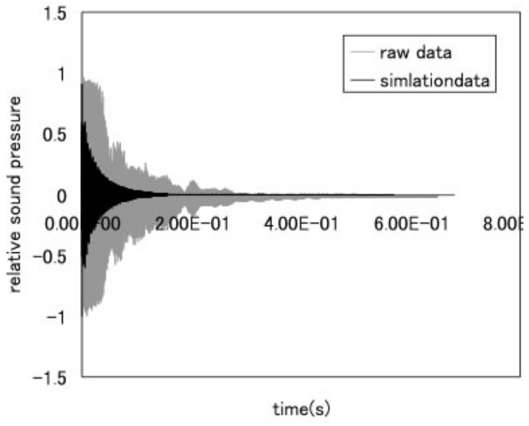

Figure 11: Time response of the sound pressures

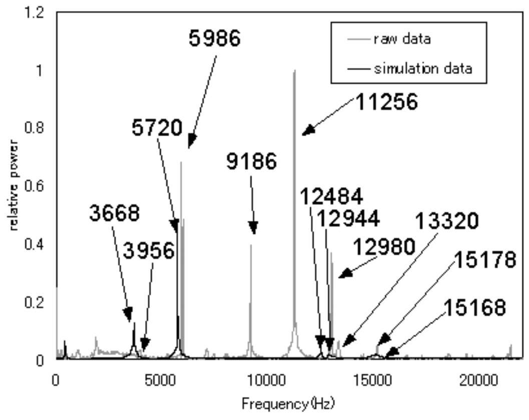

Figure 12: Power spectrum of the sound pressures

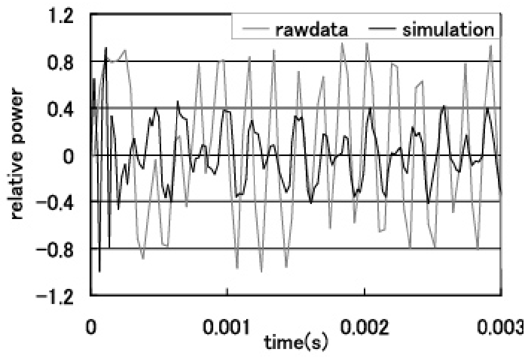

Figure 13: Initial time response of the sound pressures

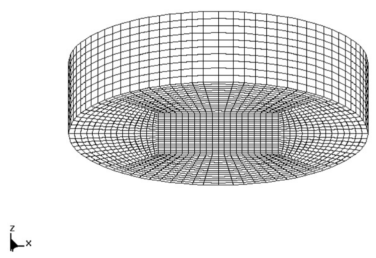

Figure 14: Mesh data of solid aluminium disk

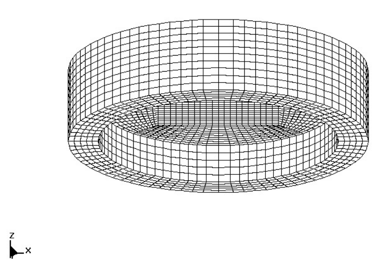

Figure 15: Mesh data of aluminium disk with hollow

IOA can provide a networked distributed virtual environment with an enormous number of interface devices.

Current computer resources have insufficient performance for full FEM analysis. However, we think that it is simply a matter of time until they do. Tsubouch and others study real time deformation analysis by FEM (Tsubouchi et al, 2000). They developed real time FEM using prediction and parallel processing techniques. Also, the data-based FEM, which can prepare many inverse matrixes and calculate the response when an impact occurs will be one solution for real time FEM. We think it will be possible to calculate the vibration of virtual objects using those methods and with the advancement of computer technologies.

7 Conclusion

In this paper, we introduce a method of generating audio and haptic feedback based on physical models and we propose a software architecture that we have named IOA. We construct an audio and haptic feedback environment using IOA, and we have evaluated the effectiveness of IOA though experiments.

For future work, we plan to construct a networked environment, and we also plan to generate the sound of other materials such as rubber or wooden objects.

8 Acknowledgement

The authors would like to thank Prof. Mizutani and Prof. Kameda at University of Tsukuba, for their many useful suggestions and co-operation.

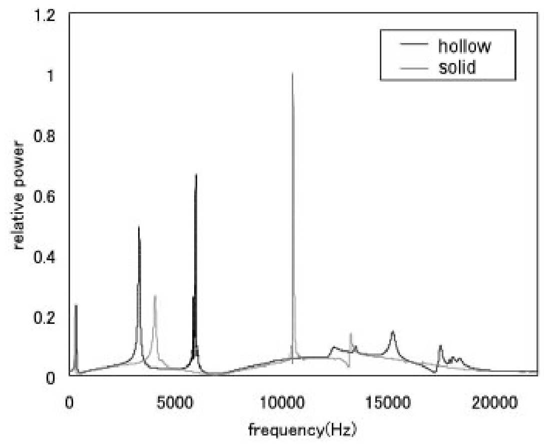

Figure 16: Power spectrum of the virtual disks

References

K.Doel, & D.K.Pai,(1998),The Sounds of Physical Shapes, *PRESENCE*,Vol 7,No.4, 382-395

Pope,A. (1989), The SIMNET Network and Protocols, BBN Report No.7102

Zyda,M. et.al.(1992), NPSNET:Constructing a 3D Virtual World, *Computer Graphics*,Special Issue on the 1992 Symposium on Interactive 3D Graphics

Singhal,S. & Zyda,M. (1999), Networked Virtual Environments - Design and Implementation, ACM Press Books

Randolf,M. et.al. (1997), Adding Force Feedback to Graphic System: issues and Solutions, Proceedings of SIGGRAPH'97

Iwata,H. & Yano,H. (1993), Artificial Life in Haptic Virtual Environment, Proceedings of ICAT'93

Iwata, H., Yano, H. & Hashimoto, W. (1997), LHX: An Integrated Software Tool For Haptic Interface, *Computers & Graphics*,Vol.21,No.4,413-420

Harada,T et.all (1998), The Influences of Multimodal Sensory Information Display on Dribbling of a Basketball in a Virtual Workspace, 4th International Conference on Virtual Systems and MultiMediaVol.2, P548-553

Asano,T. et.al (1997), Basic Technology of Simulation System for Laparoscopic Surgery in Virtual Environment with Force Display, *Medicine Meets Virtual Reality*, IOS Press,207 - 215

T.H.Massie (1996), Virtual Touch Through Point Interaction, International Conference on Artificial Reality and Tele-existence, 19-38

Tsubouchi et.al.,(2000), Real Time Deformation Analysis by Finite Element Method, Human Interface Vol.2 No.2, 3-8