TOWARDS PRACTICAL VIRTUAL TRAINING ENVIRONMENT THROUGH VR TECHNOLOGY

Yoshihiro Takamura Norihiro Abe Kazuaki Tanaka
Kyushu Institute of Technology
680-4 Kawazu, Iizuka-shi, Fukuoka, 820-8502, JAPAN
E-mail: abe@mse.kyutech.ac.jp
Hirokazu Taki ,
Wakayama University
930 Sakaedani, Wakayama-shi,
Wakayama 680-8510, Japan
Shoujie He
Eastman Kodak Company,
Plano, Texas, USA

KEYWORDS
Training system, Virtual Reality, Synchronized real-time processing,, Billiards game

ABSTRACT
Virtual reality (VR) technology has been more and more mature over the last decade. Development of a virtual environment for training purpose is considered one of the most practical applications of the VR technology. Since the VR technology involves all kinds of sensors in exchanging information between the real world and the virtual environment, it is computationally intensive in terms of data processing at an individual sensor and information integration among all the sensors. In general, the information integration has to be well synchronized in order to meet the training needs. At the same time, real-time processing capability is also considered critical. Many more practical issues could be uncovered only when a virtual training environment is actually being developed. Based on this belief, this study is experimenting on the development of a virtual environment for training billiards players. The technical difficulties encountered and the corresponding resolutions are considered beneficial to the development of other practical virtual training environments. This paper summarizes the design and implementation details about our experimental virtual training environment and reports the algorithms for the synchronization of the information from different sources.

. INTRODUCTION

Ever since the computer was invented, a lot of things that used to be impossible have become possible and many imaginations have become the reality. Constructing a 3D world inside a computer and interacting with it have been the goals that virtual reality (VR) technology is targeted at. Over the last decade, the VR technology has become more and more mature. These days, low cost and high performance computers and the advanced sensor technologies become the driving force for the VR technology to quickly find its real-world applications. Development of a virtual environment for training purpose is considered one of the most practical applications of the VR technology.

A high-quality virtual training environment is required to offer the trainee a real-world experience while interacting with virtual objects. This includes not only the high-quality 3D graphics representation of the virtual environment and virtual objects but also the high-performance interactivity. The 3D graphics representations need to be dynamically and smoothly updated based upon the user interaction. The interactivity, on the other hand, requires well synchronized real-time processing of the sensor data. These are all computationally intensive. The limited resources and processing power on a single PC become very difficult and even impossible to meet the needs.
In this study, a virtual environment for training billiards players has been developed. Behavior identification, 3D sound effects, and the force torque sensor feedback are identified as three essential components. The synchronization of the real-time processing of the sensor data is achieved through the SCRAM Net+.

SYSTEM ORGANIZATION

To construct a high-quality virtual training environment, it is a common recognition that there must be visual, auditory, and haptic sensors and the sensor data must be well synchronized. This study

Proceedings 20th European Conference on Modelling and Simulation
Wolfgang Borutzky, Alessandra Orsoni, Richard Zobel © ECMS, 2006
ISBN 0-9553018-0-7 / ISBN 0-9553018-1-5 (CD)

basically follows the same direction with the emphasis on the synchronization among the behavior generation using rigid object physics, 3D sound effects, and the haptic feedback. In order to do so, VORTEX, 3D-Sound Space and PHANToM are used in the construction of our experimental virtual environment.

Since it is impossible to have one PC to handle the processing of all the data, three PCs are used with each dedicated to a particular sensor or device. The three PCs are connected through SCRAM Net+. Figure 1 shows the system organization. The details are described as follows.

PC for vision
(1) Graphic rendering and simulation
(2) VORTEX – the rigid object physics engine

PC for auditory
It generates stereophonic sound using Roland 3D Sound Space and RSS Data Stream Management system.

PC for haptic computation

It is used as the user interface and returns haptic feedback through the GHOST and PHANToM

PARALLEL PROCESSING ISSUES

With the nature of the system organization, the following parallel processing related issues have been identified.

- The transmission of position data
- Presentation of haptic sensation
- Synchronization of sound effects

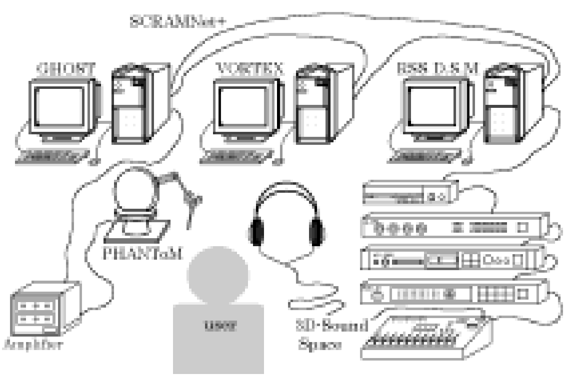

Figure.1: system organization

The transmission of position data

The three PCs are doing their jobs cooperatively by sending their data through the shared memory of the SCRAM Net+. When a user starts interacting with the VR environment, for example, the interaction is communicated to the VR environment through PHANToM. The interaction event is then simulated by VORTEX. Based on the simulated results, RSS10 will play the sound effects and the PHANToM will react with haptic effects to the user whenever it is necessary.

The position data such as shift or rotation that are generated through the interaction with PHANToM is transmitted to VORTEX through the SCRAM Net+ with a delay of less than one millisecond. The results simulated by the VORTEX must be transmitted to the PC for a haptic reaction data computation and the PC for the processing of the 3D sound effects. All the data are written into the shared memory in the form of coordinates and a transformation matrix.

As soon as a simulation result is updated, are the updated coordinates written to the shared memory. However, since the updating of the simulation result is intermittent and non-linear, the receiver has to behave in the same way. There is a possibility that a moving virtual object passes through other virtual objects as shown in Figure 2 and thus the unexpected simulation happens.

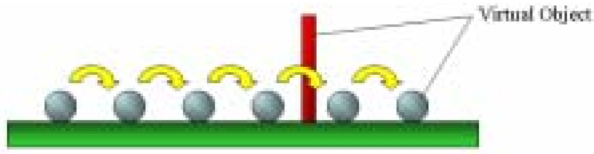

Figure.2: A penetration problem

In the rigid object physics, the impulsive force occurring at the collision is generally calculated from the inertia tensor and relative velocity based on the shape data. As velocity information is indispensable while applying the rigid object physics to objects colliding with each other, instead of updating the coordinates, updating the velocity is considered necessary.

It is, however, impossible to correctly simulate the state of a virtual object because an error slowly accumulates if the state is updated based on the velocity. Therefore, both the current coordinates and the velocity of a virtual object are considered necessary while updating the shared memory. In this case there is potentially a danger of error accumulation.

For instance, right after both the coordinates and the velocity of a virtual object have been transmitted, the virtual object suddenly stops moving. In this case, its coordinates are invariant but in the PCs receiving the coordinates and velocity of the virtual object, the virtual object keeps moving.

This gives rise to the inconsistent information between two PCs in regard to the position of the virtual object. The slower the updating is, the more significant the error becomes. Because of the error, velocity is considered inadequate for the application that concerns the accuracy of locating the virtual objects. In the regular cases, using the coordinates is recommended. It is suggested to take the velocity into account as well only at the time that collision is going to happen.

Consider the situation that a moving virtual object collides with a still virtual object. In this case, a user hits a virtual billiard ball using PHANToM. The PHANToM here is the billiard cue. The collision of the PHANToM with a virtual billiard ball is detected by GHOST. Since the coordinates of the virtual ball is not sent to the VORTEX yet at the moment when GHOST detected the collision, collision is not happening in the VORTEX. At this moment, changing the control from the coordinates to the velocity makes it possible to apply the rigid object physics in the VORTEX.

It is necessary to transmit to the VORTEX the coordinates and velocity right before the collision occurs. This is because the collision has already occurred if the coordinates at the collision are transmitted. Figure 3 illustrates how to change the control by velocity (the right) from that by coordinates (the left). Not coordinates at the time collision were detected but the ones one step before (just before collision) and velocity are transmitted to the PC for VORTEX.

It was a concern that inappropriate movement may happen due to the skipped coordinates update for one time interval. The experimental results show that haptic the movement within one time interval is so tiny and the haptic simulation in GHOST is repeated 1000 times per second and thus nothing unnatural is noticeable. This technique makes it possible to change control from that using coordinates to that using velocity at the best timing and vice versa. As a result, applying rigid object physics at collision becomes possible while keeping precise operation

The presentation of haptic sensation
When performing parallel computation in a real time simulation, it is necessary to run simulation while maintaining synchronization between two PCs. Processor speed in two PCs are so different from each other that it is difficult to synchronize them as the time taken in each process are quite different.

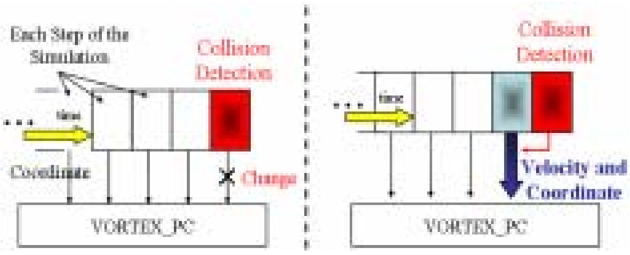

Figure.3: How to change process

The presentation of haptic sensation
When performing parallel computation in a real time simulation, it is necessary to run simulation while maintaining synchronization between two PCs. Processor speed in two PCs are so different from each other that it is difficult to synchronize them as the time taken in each process are quite different.

VORTEX needs at most 60 Hz but GHOST does 1000Hz. The smooth presentation of touching sense is generally required at least from 200 to 300 Hz. If VORTEX were synchronized with GHOST, smooth touching sense is difficult to return. To ensure that GHOST gets enough velocity to return smooth haptic feeling, asynchronous execution of two PCs is proposed.

Figure 4 illustrates how the data is exchanged through the shared memory system. It shows how VORTEX and GHOST are accessing the shared memory along the time axis. Since GHOST is running at high processing speed, it writes data into the shared memory with high frequency, but reads data out from the shared memory only after VORTEX has finishing writing. Therefore, although it is parallel computation, accessing data asynchronously makes it possible to let the individual PC perform its own job without affecting others.

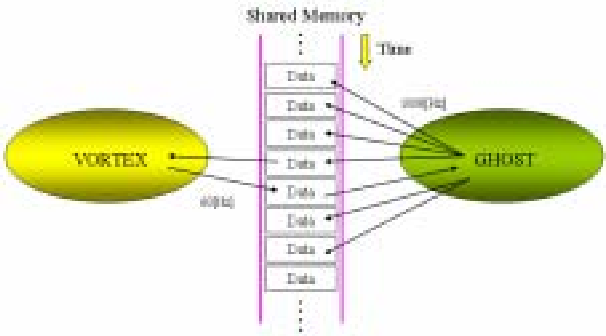

Figure.4: Operation from two PCs to a shared memory.

Synchronization between sound generation and collision detection

In a general sound revitalization instrument, when a sound is being generated, it is impossible to revitalize several sound effects simultaneously. Any new revitalization command is simply ignored until the current sound effect has been fully generated. Because of this limitation, when more than one sound effects are needed in a short period such as the simulation of the repeated collision, none of the sound effects will be generated. In addition, when multiple objects collide with each other at the same time in the real world, a loud sound will be heard. With the current limitation, instead of a loud sound, a monotonous sound will be generated.

In the system, when virtual balls collide in an extremely short period, the sounds accompanying with the collision are considered as a unified sound. When more than one virtual ball collide with each other, the sound effect is generated by recording in advance the real sound accompanied with the collision of the real balls in the real world and then replaying the recording. Different levels of sounds are also recorded in advance according to the velocities of virtual balls. Each sound must be recorded as short as possible so that the recording could be repeatedly replayed as needed.

CONSTRUCTION OF VR SPACE

In order to offer a real-world experience to the players, the virtual space must be made with high interactivity. In general, if a simulation is only for a simple scenario, it is sufficient to build only one VR space.

This system, however, is constructed with 3 PCs and each of the PCs is handling visual, auditory and haptic sensations, respectively. Since each of them is handling different type of tasks and thus the time spent is largely different. If we build only one VR space which includes everything, in the case that a particular PC spends extra long time to finish its task, the other two PCs will have to be idling during that period of time. In this study, in order to speed up the overall processing, instead of one VR space, three VR spaces are constructed with each on one PC. Since each individual VR space is handling one type of tasks (visual, auditory, or haptic), only the data related to the particular sensor or device is held in the memory. The memory usage could be reduced substantially.

Figure 5 illustrates the information needed for each of the VR spaces. The PC for vision space keeps track of the coordinates of virtual objects. The PC for sound space locates the existence of the player and also the sound sources so that the virtual sound could be generated to the best effect. The PC for haptic sensor has to keep track of all data of all the virtual objects including the cue, stand and balls for detecting the collision between the cue and virtual objects.

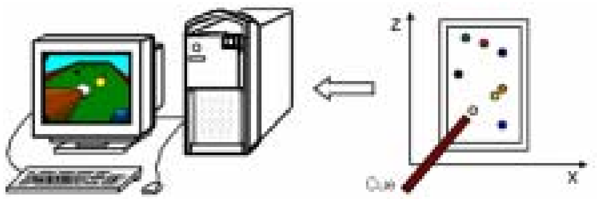

Figure.5-1: Vision space

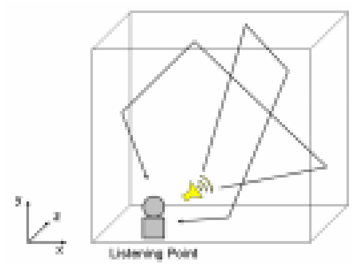

Figure.5-2: Sound space

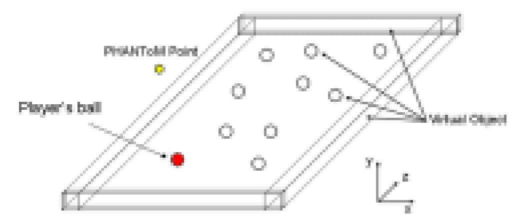

Figure.5-3: Haptic space

Although there are three separate VR spaces, the player needs to feel like interacting with a single VR space. Otherwise, the player will not have the real-world experience. This requires synchronizing the three VR spaces seamlessly. This is achieved through the SCRAM Net+. When one VR space has changes, the updates will be written into the shared memory of the SCRAM Net+. The information in the shared memory will be read by the other VR spaces. By controlling each VR space separately, the visual, auditory and haptic sensations are eventually synchronized.

FLOW OF PROCESS

Figure 6 shows the outline flow of a system,

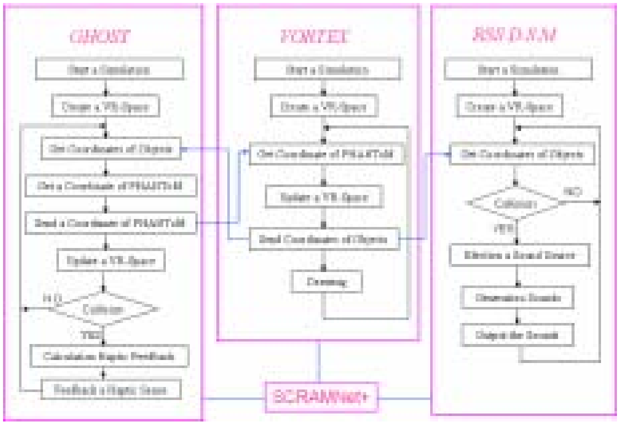

Figure.6: A flow of the system.

1. The visual, auditory and haptic simulation is started at each PC.
2. Each VR space is constructed.
3. When a user operates a cue, GHOST acquires the coordinates and posture of PHANToM through SCRAMNet+, and detects collision detection between a cue (PHANToM) and a ball which is hit with the cue. And haptic feedback is returned to the user via PHANToM and the collision information is written into SCRAMNet+.
4. VORTEX reads coordinates of the object whose position is changed and updates simulation. As a result, as soon as the collision is detected, the result will be written to the SCRAM Net+. Finally, it visualizes the virtual world on a display. The same process is repeated (Hereafter this is regarded as one cycle).
5. RSS10 acquires the name, coordinates and velocity of a set of virtual objects which collided from the SCRAM Net+ and then generates stereophonic sound and output the sound effects.

VORTEX and GHOST at the acquisition of data do not have to wait for the other output in order to access the SCRAM Net+. This makes it possible for both of them to guarantee the updating frequency of simulation. The above, however, is a basic regular flow. The other special cases such as switching from coordinates to the velocity and other information exchanging are performed only when it is needed.

A flow of a vision system

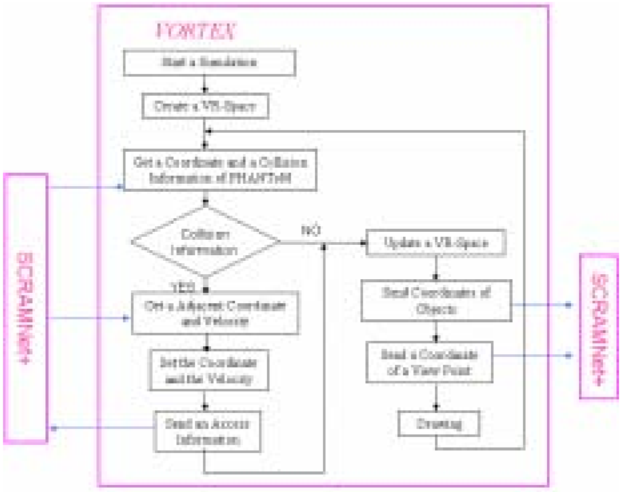

Figure 6: A flow of a vision system.

1. In addition to a transformation matrix of PHANToM, collision information is also collected. Whether there is a collision between PHANToM and a virtual object is detected by GHOST.
2. If the collision is detected with GHOST, the control is switched to be velocity-based. The SCRAM Net+ is accessed again in order to get coordinates and velocity of PHANToM right before the collision.
3. After setting the coordinates and velocity of PHANToM right before the collision with the object, the control which prevented PHANToM from updating coordinates must be canceled. It transmits information to make PHANToM write its coordinates hereafter.
4. The simulation is updated again and in order to make collision between PHANToM and an object by all means happen at this step, once again control is changed back to the coordinates-based mode from the next step.
5. Updated coordinates of all virtual objects and current cue coordinates are written to the SCRAM Net+

A flow of an auditory system.

1. Access the SCRAMNet+ and read all the updated information from VORTEX. The information includes the coordinates of all virtual objects and a position of PHANToM.
2. Check if collision among virtual balls or with the frame of the billiards board is detected with VORTEX. Also check to see if a virtual ball has been hit with the cue.

3. When collision is detected, the number of collision and information concerning pairs between objects colliding with each other is acquired in order to select the sound source for the generation of the sound effects.
4. After a sound source is selected, sound generation command in MIDI is transmitted to Roland Sound Space.

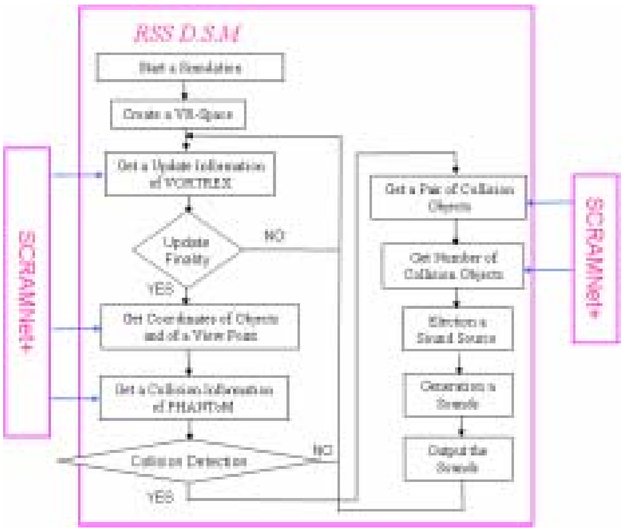

Figure 7: A flow of an auditory system

A flow of haptic system

1. Once the haptic space is constructed, the coordinates of virtual objects will be transmitted into the haptic space. Note that the coordinates of the virtual objects are all simulated by VORTEX.
2. Not only coordinates and the posture but also the velocity of PHANToM are acquired, and they are stored in memory to make it possible to use them when VORTEX switch the control to from coordinates-based mode to the velocity-based one.
3. If collision has been detected before, the transmission of the current PHANToM coordinates is restrained in order to make VORTEX shift to the velocity-based control mode. During that time, the SCRAM Net+ does not allow any data to be written until VORTEX finished reading velocity information. Here it is examined whether VORTEX has updated its state or not.
4. In the case that it has not been examined yet after collision, the transmission of new coordinates will not be done. Only PHANToM coordinates are updated. Collision analysis among virtual objects will not be completed because there is the possibility of consecutive collisions.
5. When it is not examined or when collision does not happen yet, collision detection is performed. If there is collision, collision information and the coordinates and velocity stored right before are transmitted.

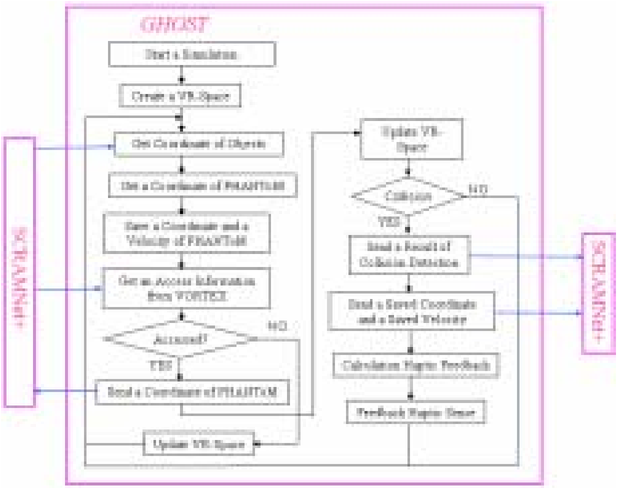

Figure 8: A flow of haptic system

VIRTUAL BILLIARDS GAME

A virtual billiards game is selected to evaluate the method proposed in this paper. The game board and a near view of a billiard ball which is going to be hit with a cue are shown in Figure 9.

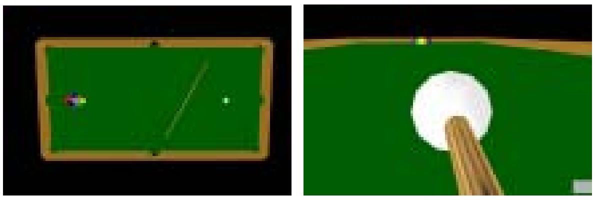

Figure.9: Virtual billiard game

Figure 10 shows a cue controlled with PHANToM. Translating or rotating the interface instrument gives a virtual cue the same movement.

Improvement of processing speed by parallel computation.

The processing speed of each PC is measured. The time taken to complete one cycle of process using each PC is measured 100 times and the maximum and the minimum of the results were recorded. Processing speed when a single PC manages 3 jobs was measured afterwards.

Experimental result

It is understood that large amount of processing time can be reduced by parallel computation. It seems that it is not impossible to perform 3 jobs with a single PC, but Table 1 shows that both haptic and auditory rendering is not fully realized. On the other hand, enough iteration necessary for giving a user high reality is achieved by each PC exploiting parallel computation.

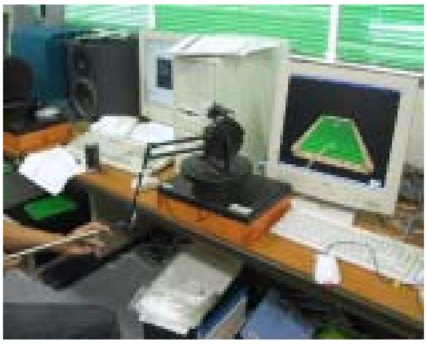

Figure.10: User interface

Table 1: Comparison between processing speed

Task	Time [ms]	Frequency[Hz]
vision	16-20	48-60
auditory	8-12	83.3-120
hap tics	3.6-5.1	196-277
All of them	25-31	32-40

Evaluation of the system

Ten subjects were asked to operate a virtual billiards game and to evaluate the validity of visual, auditory and haptic sensation and how much they are synchronized. Results are shown in Figure.11 and Figure.12. Figure 11 implies that good evaluation is obtained concerning both auditory and haptic sensation. It can be considered that parallel computation succeeded in constructing VR space in which haptic sensation synchronizes with auditory one. But evaluation on visual sensation was not so good. Concerning behavior of virtual objects, high reality is attained as simulation is performed based on rigid object physics. Rendering system provided with VORTEX does not have enough functions which GL offers and cannot draw any shadows of virtual objects. It is still necessary to evaluate how close the current behavior is to the real billiards game. In terms of improvement, at the very least, the friction factors need to be incorporated in the future.

Figure 12 shows that enough synchronization among them is established. One of the goals of this study is to make the real-time processing possible while dealing with the different type of sensor data. In that sense, the experimental results are positive.

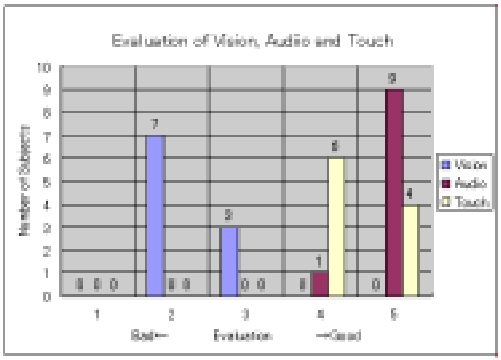

Figure11: Evaluation on visual, auditory and haptic sensation.

CONCLUSIONS

Game construction with high reality is obtained by integrating rigid object physics, haptic feedback and stereophonic sound system. For developing a system with much higher reality than that realized in this study, the operability which gives a user the feeling as if he were playing a real game is necessary. One of promising ways is to build a Mixed Reality system permitting a user to hit a virtual ball with a real cue.

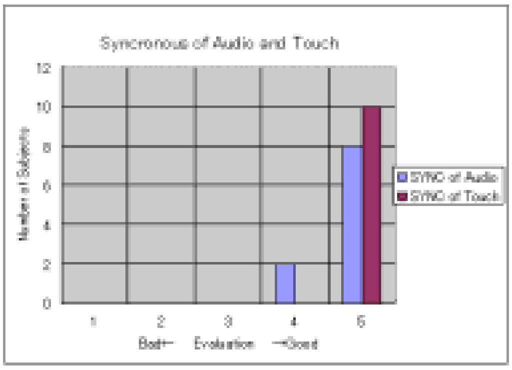

Figure.12. Evaluation of synchronization among three sensations.

The system needs not only installing both position sensors on the body and impact generation device on the tip of the real cue, but also technique realizing complete registration between real and virtual environment permitting a user to hit a virtual ball with a real cue.

REFERENCES

Dragovi´c, V., Radnovi´c. M. “Geometry of Integrable Billiards and Pencils of Quadrics”, Journal de Math´ematiques Pures et Appliqu´ees

Jebara, T., Eyster, C,. Weaver, J. Starner, T., Pentland, A. 1997. “Stochasticks: Augmenting the Billiards Experience with Probabilistic Vision and Wearable Computers”, Proc. of the Intl. Symposium on Wearable Computers,

Matsuura, H,, Abe,N., Tanaka, K., Taki, H. 2004. “Reciprocating vision and auditory sensation through network.”, 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS), pp.953-958.

Pan, Y., Abe, N., Tanaka, K., Taki, H. 2004. “The Virtual Debugging System for Developing Embedded Software Using Virtual Machinery”, Proc. of Embedded and Ubiquitous Computing, International Conference (EUC 2004), pp.85-95, Aizu Japan.

Smith, R. 1998. “Intelligent Motion Control with an Artificial Cerebellum.”, Ph.D. thesis in July 1998, department of Electrical and Electronic Engineering at the University of Auckland, New Zealand.

Tanaka, K., Kaida, M., Abe, N., Taki, H. 2002. “Training System for Crisis Avoidance using Virtual Auditory Environment.”, Proc. International conference on Machine Automation(ICMA), pp.623-629.

Tanaka, K., Kaida, M., Abe, N., Taki, H. 2002, “Synchronization of visual, haptic and auditory sense using a haptic display and a virtual sound source device” the International Society on Virtual Systems and Multimedia (VSMM), PP.673-680.

Watanabe, Y,. Abe, N., Tanaka, K., Taki, H., Yagi, T. 2005, “Multimodal communication system allowing man and avatar to use voice and beck.” 3rd International Conference on Information Technology and Applications (ICITA'2005), pp.161-166.

Watanabe, Y., Tokumochi, D., Abe, N., Tanaka, K., Taki, H., Kinoshita, Y. 2005. “Cutting Virtual Organ Model with Haptic Feedback Device.” The First International Conference on Complex Medical Engineering-CME2005, pp.255-260.

AUTHOR BIOGRAPHIES

YOSHIHIRO TAKAMURA received his Master Degree from Kyushu Institute of Technology. in 2004. He is now working at ALPINE in Japan. His hobby is billiards, and he was engaged in virtual billiards game construction.

NORIHIRO ABE received his Ph.D. from Osaka University in 1974. He is now a professor of Kyushu Institute of Technology in Japan. His research interests are Artificial Intelligence including an integration of vision and language systems, Assembly system and Virtual reality system. He has published about 200 refereed papers to various journals and international conference proceedings. Web-page can be found at. http://www.sein.mse.kyutech.ac.jp

KAZUAKI TANAKA received his Ph.D. from Kyushu Institute of Technology in 2000.and now is a lecturer of the same university. He is interested in hardware system development using FPGA..

HIROKAZU TAKI received his Ph.D. from Osaka University in 1986. He is now a professor of Wakayama University in Japan.. His research interest includes Knowledge based system, medical ontology system.

SHOUJIE HE received his Ph.D. from Osaka University in 1990. He is now a researcher of Eastman Kodak Company in USA.. His research interest is devilment of image processing system including CT images.