956 *KSME International Journal, Vol. 14, No. 9, pp. 956~967, 2000*

Cooperative Manipulation of a Virtual Object by Multiple Remote Users

Hyoukryeol Choi*, Sungmoo Ryew
School of Mechanical Engineering, Sungkyunkwan University

In this paper, we explore the issues of force display in the cooperative virtual environment shared by multiple users distributed over the network with heterogeneous hardware platforms. The proposed method is to cope with the problem of small time delay and the difference of sampling rate in the distributed configuration. In the proposed approach the interaction forces of the participants are just treated as the independent sources of acceleration. Thus the action of a participant simply changes the acceleration of the virtual object and consequently the states of the virtual object will be updated. When the updated states are reported to all the participants, the information on the time of state changes is delivered, too. Employing the discrete state information updated by the other users, each user modifies his own virtual environment and pseudo-realtime simulation can be realized. Excluding the software interface and the communication technique, it is proposed the simulation method for the operation of respective users and the way of calculating the driving input to the display device. For experimental verification we construct a cooperative virtual environment shared by two remote users and outline the results of experiments.

Key Words : Cooperative Manipulation, Force Display, Multiple Users, Virtual World

1. Introduction

Virtual reality is a powerful tool for training, simulation, entertainment and design, etc. Up to now, most researches on the virtual reality have been focused on the visualization of motions and interactions among virtual objects (Philips et al., 1990). Recently, however, the importance of the sensation of force and touch has been recognized and a number of studies have been performed in the field of haptic display. Previous researches have been made on displaying shapes (Tanie and Kotoku, 1993), forces for deforming (Yamamoto et al., 1993; Hyoukryeol Choi et al., 1998), weights (Iwata, 1990), the dynamics of the virtual objects (Yoshikawa et al., 1995), and force display devices (Massie and Kenneth, 1994), which are mostly allowed to single user. In many applications, however, it is desirable to allow multiple users to interact in a shared virtual environment. A platoon of soldiers, a plane crew, a surgical team can train together in a shared virtual environment. The shared virtual environment enables multiple and spatially dispersed users to cooperate on a common task. Here the network is used to enhance the collective performance of a group of users by allowing them to share informations with each other. The shared virtual environment extends interaction beyond the single-user virtual environment, but introduces increased programming complexity because multiple programs on different machines have to be managed, and the participants must coordinate communication among themselves. The level of interaction depends on the physical constraints of the communication media and on the available

---

* Corresponding Author,
E-mail : hrchoi@mecha.skku.ac.kr
TEL : +82-31-290-7449 ; **FAX** : +82-31-290-5849
School of Mechanical Engineering, Sungkyunkwan University, Chunchun-dong 300, Jangan-gu, Suwon, Kyungki-do, 440-746, Korea. (Manuscript **Received** February 10,2000 **Revised** June 15, 2000)

computational power. The architecture of the distributed simulation must fit the specific application.

There are many researches about virtual environment shared by multiple users (Gossweiler et al., 1994; Bricken and Coco, 1994; Macedonia et al., 1994; Calvin et al., 1993; Ishii et al., 1994; Buttolo et al., 1995). However, the force display is provided in only a few systems (Ishii et al., 1994; Buttolo et al., 1995). Employing the force display in a shared virtual environment introduces some serious constraints. If multiple users are to manipulate the same portion of the shared virtual environment we have to overcome the intrinsic problems of delay and the difference in the sampling rate among the heterogeneous hardware platforms. These problems give unnatural feel for the operator as well as the instability of the display device. As a solution for the problem, "one-user-at-a-time" approach was proposed by Buttolo et al. (Buttolo et al., 1995). In this approach, only one user at a time, the active operator, is allowed to modify and feel forces from the environment. The other users, non-active operator, just watch the on-going interaction through a visual display and move around the shared virtual environment, but not touch or modify objects. This approach, however, has disadvantageous aspects when we apply to the case that the users have direct interactions with each other. For instance, suppose that multiple users cooperatively manipulate a common virtual object. One-user-at-a-time approach allows only one user to actively manipulate and the others should be inactive until the action of the active user is ended and the results are delivered to them. The inactive users can't experience the cooperative manipulating feel in realtime. Thus, this method is not applicable though it is adequate for the virtual environment without direct interaction between the users such as the networked virtual tennis game (Buttolo et al., 1995). The force display in this situation appears not to have been tackled yet and no work could be found.

In this work, we propose a method of force display for the virtual environment shared by multiple remote users, where direct interactions are required among remote participants. Each user in the distributed configuration has a different hardware platform that brings different sampling rates for display devices, and small time delay exists though they are close together. In the proposed method we regard the interaction forces as the independent source of acceleration. Thus the action of each participant simply changes the acceleration of the virtual object and any user is allowed to affect the acceleration of the object at any instance. As the result of changes of the acceleration, the states of the objects are reported to all the participants and their database of the virtual environment will be renewed. On reporting the updated states the information of the time of state changes is delivered, too, which plays an important role in the respective virtual simulation. The other users utilize the discrete state information updated by the other users and modify their own virtual environment by applying approximation technique such as extrapolation, dead reckoning etc., and pseudo-realtime simulation can be realized. The advantage of the proposed method is that every participant is allowed to be the active user in a specific instance. He reports the results of his action to the other users and the others just need to simulate the virtual environment based on the updated information.

At first, we review two typical methods for implementing distributed virtual environment. In this work, it is focused on how to overcome the problem of small time delay and the difference of the sampling rate among distributed participants with heterogeneous hardware platforms. The software interface and the communication method are excluded, and we address the simulation method for the operation of respective users and the way of calculating the driving input to the display device. Finally, an experimental virtual environment shared by two remote users is developed and experimental results are presented to verify the effectiveness of the proposed method.

2. Centralized Model and Distributed Model

At present, there are two popular approaches

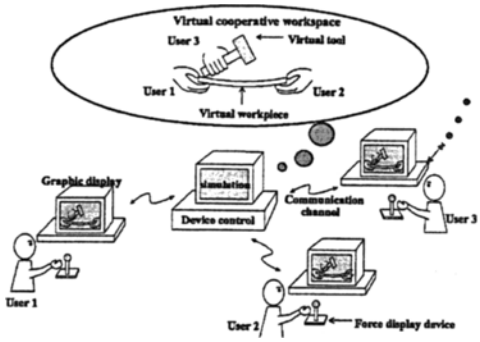


Fig. 1 Centralized model of shared virtual environment

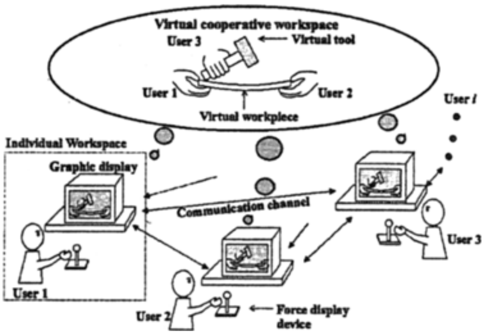


Fig. 2 Distributed model of shared virtual environment

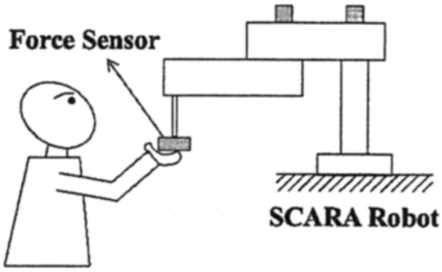


Fig. 3 Force display device with link mechanism

for implementing distributed virtual environment that is, *centralized model and distributed model* (Gossweiler et al., 1994). In a centralized model shown in Fig. 1 a central computer collects all the data from the participants, stores the changes on the centralized database, and sends the results back out to all the users. Each participant then renders the scene and handles user input according to the information delivered from the central computer. However, it is not easy to display realtime force feel to each user though the graphic simulation may be possible. Especially, under the environment with delay and sampling rate and accompanying physical interactions among the users, such as cooperative manipulation, grasping, hitting etc., it is almost impossible to apply the centralized model because the individual system does not have a right to control its force display device in realtime. The distributed model is illustrated in Fig. 2. This model is a scalable structure and thus, participation and withdrawal of the player is more flexible than the centralized model. This model lets each program maintain its own complete, local copy of the database as well as performing the rendering, computation and animation of objects. When a program makes changes to its own database, it sends the updated data out so that other programs can update their individual database by using communication techniques such as broadcast, multicast.

Fig. 4 Object manipulated by multiple users

Due to the flexibility of the distributed model it may be the most promising configuration in the future. In the future, we assume that the users with their personal resources of virtual simulation such as their own simulation software, personal haptic devices, computers, other interface devices, and willing to attend the virtual space and enjoy the mutual interactions, would build the virtual space in a peer-to-peer like fashion. Several network games popular recently, such as MUD(Multi User Dungeon) game, though it does not allow to exchange the interaction forces, are examples of the distributed model. This work mentions how to cooperate among the personal VR systems, especially personal force display devices in the distributed configuration, which is one of the most important issues on expanding the application areas of the networked VR system.

3. Basic Formulation for Virtual Cooperative Manipulation

3.1 Force display device

As shown in Fig. 3, a force display device consisting of a link mechanism with its base fixed,

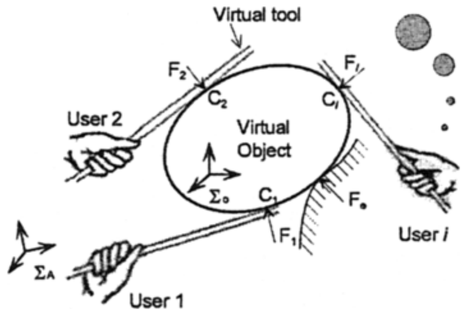


Fig. 4 Object manipulated by multiple users

is utilized to display force feel while virtually touching and manipulating objects in three dimensional space with virtual tools. This mechanism may be considered as a robot manipulator or a dedicated electro-mechanical devices with a handle at the end effector grasped by the user. Each device has sufficient degrees-of-freedom such that any motion of the user's arm can be followed. The motion information of the device is measured using internal sensors such as optical encoders at each joint of the device and the motion of the handle are then obtained using the kinematic relation of the manipulator. A force sensor is mounted at the end effector of the manipulator to measure the force applied by the user.

3.2 Basic formulation

In this section, we derive the basic equations for the virtual cooperative manipulation, which can be started from the popular technique adopted in the researches of the grasping or cooperative manipulation of multiple manipulators (Yoshikawa and Ueda, 1996; Nakamura et al., 1989). Let us now consider the situation shown in Fig. 4. Here we assume that only one virtual object exists and multiple users with virtual tools manipulate the virtual object. Though it is a little specific situation, it does not mean that the proposed algorithm just can be applied to situation. Generalization of the proposed method is always possible.

Suppose that the number of participants is denoted by k, and each user makes frictional point contact with the object. Let the position and orientation of the object represented by $x_0 \in R^3$ and $\phi_0 = [\phi \theta \psi]^T \in R^3$, respectively. Here, the orientation is assumed to be denoted by Euler angles and we choose x_0 and ϕ_0 as the generalized coordinates for the virtual object that is, $r_0 = [x_0^T \ \phi_0^T]^T \in R^6$. Then, when the user i contacts with the object, the kinematic equation can be written by

$$r_{ci} = f_{ci}(r_0) \tag{1}$$

and its time derivative will be

$$\dot{r}_{ci} = J_{oi}(r_0)\, \dot{r}_0 \tag{2}$$

where r_{ci} denotes the contact location of ith virtual tool, $J_{oi}(r_0) = \partial r_{ci} / \partial r_0^T$ is the Jacobian of r_{ci} with respect to r_0. Then, from the virtual work principle, the generalized force F_{oi} at the object coordinate $\sum_0$ which is equivalent to the force F_{ci} applied at the contact point c_i will be given by

$$F_{oi} = J_{oi}^T F_{ci} \tag{3}$$

Then the equation of motion of the virtual object becomes

$$M_o(r_o)\, \ddot{r}_o + h(r_o, \ \dot{r}_o) = \sum_{i=1}^{k} J_{oi}^T F_{ci} + J_e^T F_e \tag{4}$$

where $M_o(r_o) \in R^{6\times6}$ is the symmetric inertia matrix of the virtual object, $h(r_o, \ \dot{r}_o) \in R^6$ denotes the centrifugal, coriolis, and gravity forces. $F_e \in R^6$ represents the generalized external forces exerted on the object such as contact forces with the environment, and J_e means the Jacobian of r_e with respect to r_o.

For each user, the kinematic equation of the i th force display device is written by

$$r_{hi} = f_i(q_i) \tag{5}$$

where n_i represents the degree of freedom of the ith force display device, r_{hi} denotes the generalized coordinate of the grasp location of ith user, $q_i \in R^{ni}$ is the joint displacement vector, $J_i(q_i) = \partial r_{hi} / \partial q_i^T$ is the manipulator Jacobian matrix. Also, assuming that each user grasp the handle of the device firmly, we have the following equation between the contact force F_{ci} and displayed force F_{hi}.

$$F_{hi} = J_{hi}^T F_{ci} \tag{6}$$

where $J_{hi}(q_i) = \partial r_{ci} / \partial q_{hi}^T$ denotes the coordinate transformation Jacobian matrix between the contact point and grasped joint of the ith virtual tool.

Here, if we employ an environment model with stiffness the reaction forces are obtained such as (Tanie and Kotoku, 1993); we can take more complicated environmental model.

$$F_{ci}=K_{ci}\Delta r_{ci} \tag{7}$$

where K_{ci} is a constant expressing the stiffness of the virtual object and Δr_{ci} is the penetration depth of the object along a specified direction at the contact point of the ith user. External forces F_e also may be determined similarly. As for the consideration of the frictional effects on the contact, the approach of Yoshikawa et. al is adopted (Yoshikawa et al., 1995).

4. Simulation of Motion

By rearranging Eq. (4) we can derive a useful equation that is helpful in developing our approach. From Eq. (4), the resultant acceleration of the virtual object will be

$$\begin{aligned}\ddot{r}_o&=M_o^{-1}(r_o)\left\{\sum_{i=1}^{k}J_{oi}^TF_{ci}-h(r_o,\ \dot{r}_o)\right\}\\&\quad+M_o^{-1}(r_o)J_e^TF_e\\&=\sum_{i=1}^{k}\ddot{r}_{oi}+\ddot{r}_{oe}-M_o^{-1}(r_o)h(r_o,\ \dot{r}_o)\end{aligned} \tag{8}$$

where $\ddot{r}_{oi}=M_o^{-1}(r_o)J_{oi}^TF_{ci}$ denotes the contribution on the object acceleration by the ith user and $\ddot{r}_{oe}$ means that of the environment. Equation (8) let us know that the overall acceleration of the virtual object is the summation of the effect of each user. However, the problem is that the remote users in the network can't exert forces simultaneously, which means Eq. (8) is not available anymore. There might be latency due to time delay and the difference of sampling rates as personal systems do not have the same configuration. In our approach, when computing the acceleration of the virtual object, only one user, that is the owner of the personal system is regarded to exist in the virtual environment, and all the other personal devices are considered as simply subsystems affecting on the acceleration of the virtual object. Thus, in a specific instance the forces exerted on the object are only those that the owners do on their personal system. Each user calculates the acceleration of the virtual object caused by his contact force F_{ci} and changes the states of the object by integrating the acceleration. Finally, he reports the change of states to all the other users by using technical means such as broadcast, multicast etc. The other users similarly calculate their acceleration contribution, r_{oi}'s and updates the states of the object, respectively. The object is regarded to move as if single user manipulates the virtual object like playing a ping-pong game, but they can share the manipulating feel of the object because each player uses the newest states of the object. The individual user has contact with the object independently and only shares the information of the object motion. According to the proposed approach, the equation of motion for the ith user appears to be

$$M_o(r_{on})\ddot{r}_{oi}+h(r_{on},\ \dot{r}_{on})=J_{oi}^TF_{ci}+J_e^TF_e \tag{9}$$

where r_{on} and $\dot{r}_{on}$ represent the newest state of the object, respectively. Thus, the user i uses the newest state information of the object and the states are updated by determining the acceleration such as

$$\ddot{r}_{oi}=M_0^{-1}(r_{on})[J_{oi}^TF_{ci}+F_e-h(r_{on},\ \dot{r}_{on})] \tag{10}$$

Also, the jth user updates the states as follows.

$$\ddot{r}_{oj}=M_0^{-1}(r_{on})[J_{oj}^TF_{cj}+F_e-h(r_{on},\ \dot{r}_{on})] \tag{11}$$

The force of each user only updates the newest state of the object and this approach makes sense because in the acceleration level, the effect of contact forces on the object motion can be superimposed. In the proposed method, it is no more

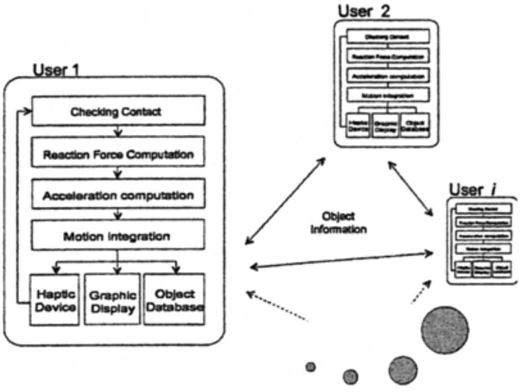


Fig. 5 Flowchart of the proposed algorithm

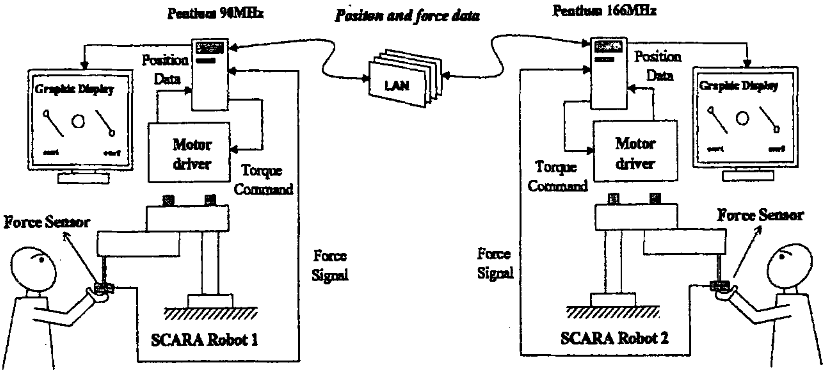


Fig. 6 Schematic diagram of force display system

than a sequential superposition is adopted.

5. Force Display Algorithm

To implement the proposed approach in the distributed model, as shown in Fig. 5 each personal system maintains its own complete, local copy of the database as well as performing the rendering, computation and animation of the object. Each user first checks the contact between his virtual tool and the virtual object. If the contact is checked, the contact force F_{ci} is computed according to Eq. (7) and updates the states of the object by using Eq. (10) or (11). When a user makes changes to its own database, he sends the update data out so that other users can update their individual database employing communication techniques such as broadcast, multicast etc. Thus, once the user has completed his interaction, each database of the personal virtual environment is updated. Until an update is received, the motion of the virtual object is dynamically simulated based on the newest states reported. This method, therefore, does not need a special mechanism such as a scheduler. Every player can update the states of the object by exerting forces and he is just required to report the results of update, that is the newest states and the time. Thus, in the proposed method, all the users can be active players simultaneously and made to feel the force sensation in realtime.

Finally, each haptic device of the ith user is simply controlled by employing the feedforward PD controller as follows; in this stage a dynamic controller for the device can be adopted.

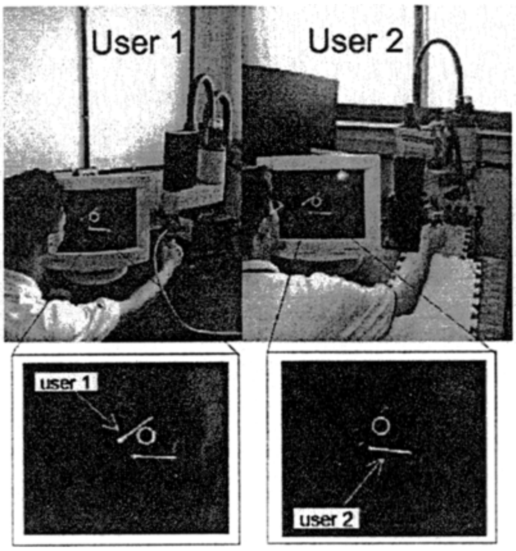


Fig. 7 Overview of force display system

$$\tau_i = J_i^T[F_{hi} + K_p(F_{hi} - F_{mi}) + K_i \int (F_{hi} - F_{mi})\,dt] \tag{12}$$

Here, F_{hi} which is calculated by utilizing Eqs. (6) and (7), becomes the desired force at the grasp location of user i. F_{mi} is the measured force. $K_p \in R^{n_i \times n_i}$ denotes the proportional gain and $K_i \in R^{n_i \times n_i}$ is the integral gain.

6. Experiments

A force display system based on the proposed method was built for confirming the validity of the proposed method. The details of the system and experimental results are described in this

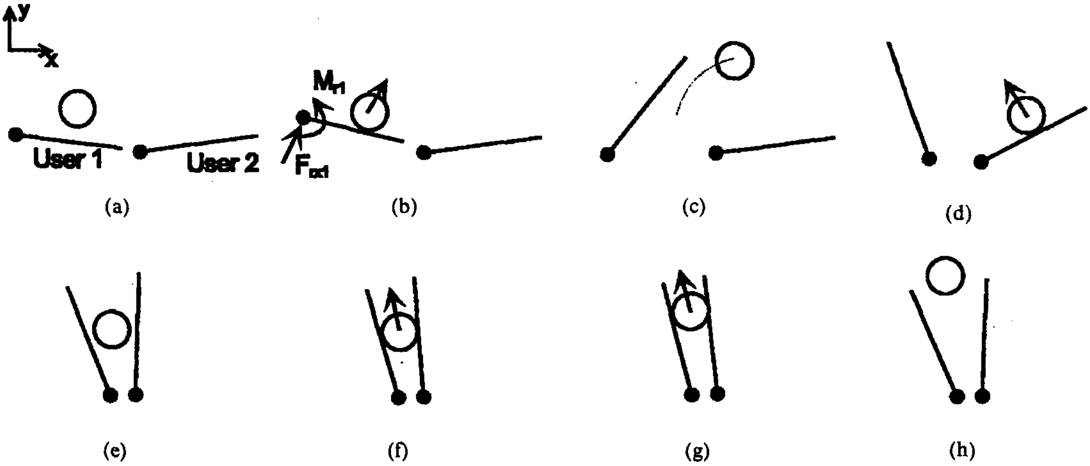


Fig. 8 Tasks in experiment

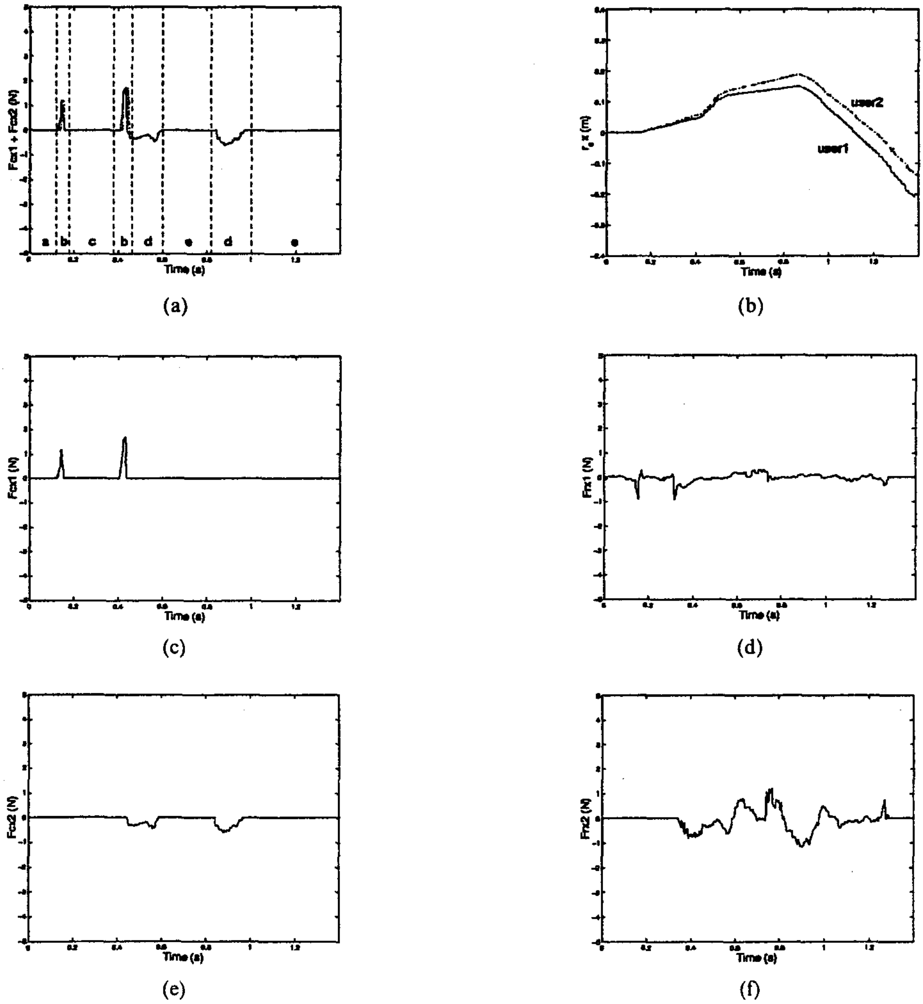
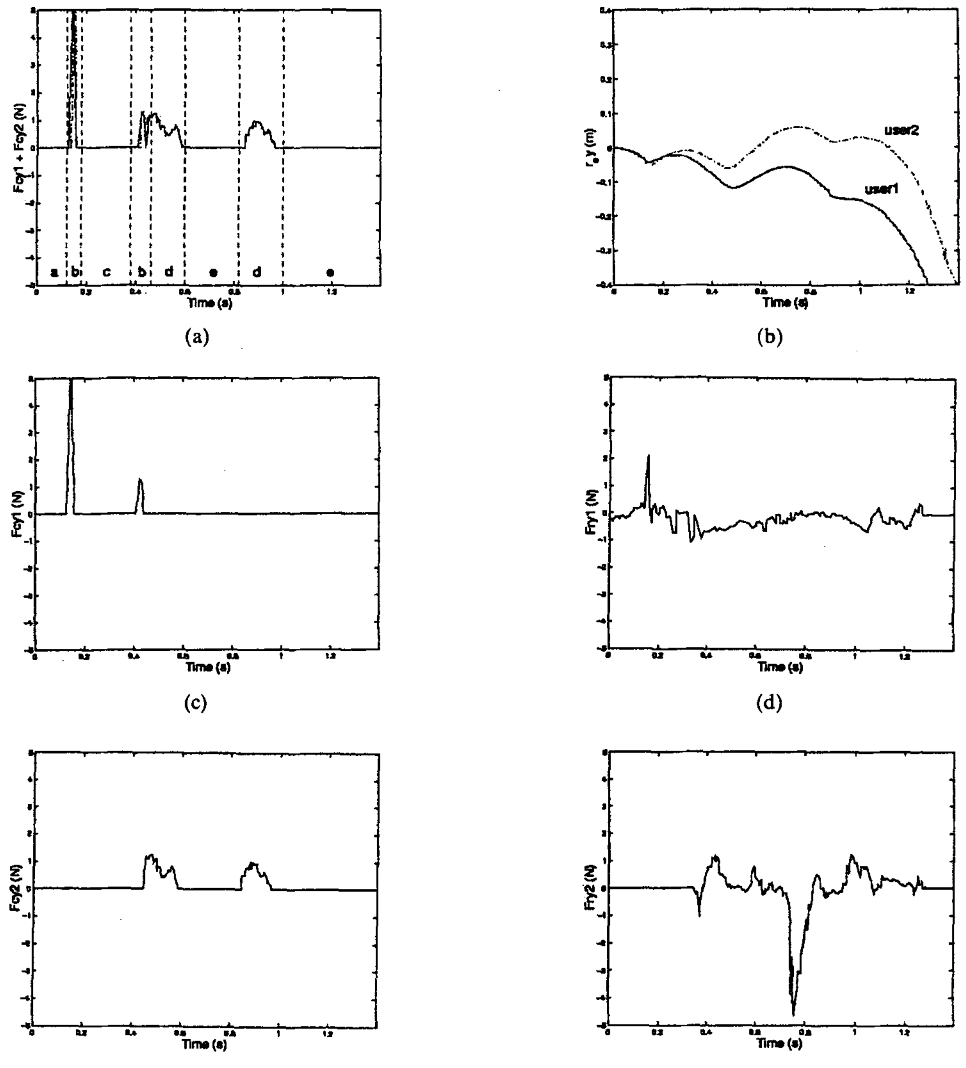


Fig. 9 Result of First Experiment(X-directional force)



(a) (b) (c) (d) (e) (f)

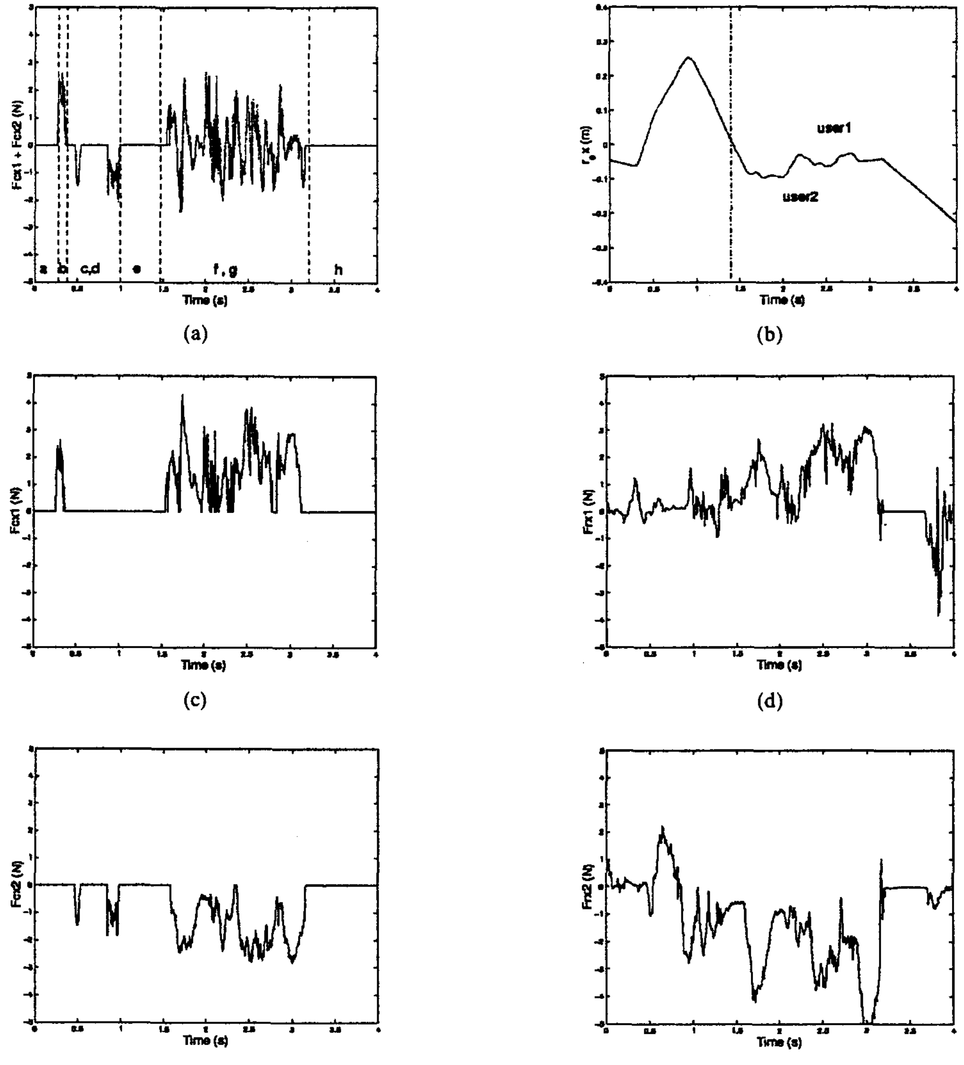
Fig. 10 Result of Second Experiment (Y-directional force)

section.

6.1 Outline of force display system

The schematic diagram of the force display system is given in Fig. 6 and its overview is displayed in Fig. 7.

In the experimental system, two remote users participate in the shared virtual environment. Each user has his personal force display device, computer, and communicates via LAN. The one user uses a personal computer with pentium 166 MHz CPU and the other pentium 90 MHz CPU. As for the force display device, both of them utilize three-DOF SCARA robot. Each joint of the robot is driven by a geared brushless DC motor with different gear ratios (80:1, 50:1, 42:1). The joint angles are measured using resolvers, and the joint velocities and accelerations are numerically differentiated from the resolver outputs, respectively. A six-axis force sensor (JR3 67M25) is mounted at the gripper attachment of the robot and senses the force given by the user. A handle for being grasped by the user is mounted on the force sensor. The control algorithm was coded with C and assembly language and their sampling rates were 600Hz and 500 Hz, respec-



(a) (b)

(c) (d)

(e) (f)

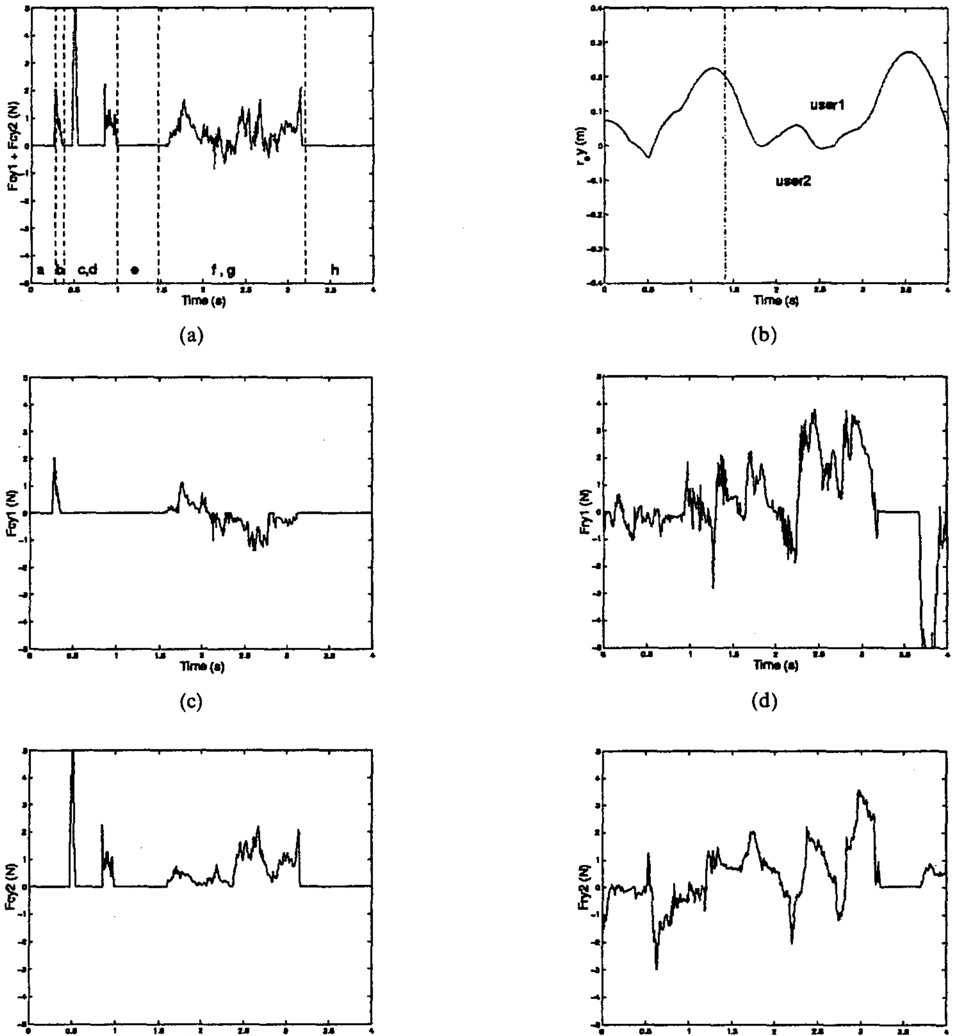
Fig. 11 Results of Second Experiment (X-directional force)

tively. Also, the object information was reported in 90Hz and 20 Hz, respectively and graphics was updated with the same rates.

6.2 Task description

In the experiment, we manipulated a virtual disk moving on the two-dimensional plane. Physical properties of the object were given such as mass M_o=0.1Kg, radius r=0.04m. The gravitational acceleration was assumed to be one third on the earth, that is 9.81/3=3.27 m/s^2, and the rotational static friction was assumed to be negligible for simplicity.

As depicted in Fig. 8, the users are enforced to do the following tasks, 1) to touch the disk as if they play ping-pong game, 2) to catch, and 3) squeeze the virtual disk. The tasks are described in detail by using Fig. 8: 1) in the stage (a~b) the user 1 pushes the disk, 2) in (c~d) the disk has contact with the virtual tool of the user 2, 3) in (e ~f) both users grasp the object cooperatively, 4) in (g~h) they squeeze the object, 5) finally the object escapes from the grasp. By the end of the state (~a) no force is expected to be detected on both users and till the head of the stage (d), user 2 will not be able to feel the manipulating forces



(a) (b)

(c) (d)

(e) (f)

Fig. 12 Results of Second Experiment (Y-directional force)

of the object. At stage (f~g) forces are displayed on both users and so on. Two experiments have been performed for comparison. In the first experiment, we simply used Eq. (8) neglecting the time delay and effect of sampling rates. In the second experiment, the proposed approach was implemented.

6.3 Results of Experiments

Figures 9 and 10 show the results of first experiment, and Figs. 11 and 12 are those of the second experiment. In the Figures, (a) is the sum of the forces acting on the disk, (b) represent the position of the disk, (c) and (e) denote desired x- and y-directional contact forces, (d) and (f) are measured x and y-directional contact forces, respectively. From Fig. 9(b) and Fig. 10(b), we can see the trajectory of the object for user1 and user2 deviate quite a lot. It means that each user has different database for the position of the virtual object and consequently. As time goes on, the difference becomes larger and in the end the cooperative manipulation is impossible as we expected. Similarly, the data of contact forces do not give any meaningful results. On the contrary, the experiment using the proposed algorithm achieved the given task successfully. Figures 11 (b) and Fig. 12(b) are the position data of the virtual object. We can perceive that both users have the same position of the object. The force data verify that each user felt the adequate force for each stage of experiments. In the stage (a) no user feels the contact force, and in state (b) only user 1 feels the contact force. In the stage (c) no force is displayed as the disk moves in the free space. In the stage (d) only user 2 feels the contact force, too and from the stage (f) to (g) contact forces are displayed on both users. In the experiments, we have used two different hardware platforms, and according to the measurements there exists maximum 0. 5 second of time delay which varies depending on the situation. The results of experiment verified that the proposed method works very well in spite of these problems. It was possible for the two remote users to manipulate the virtual object as if they were at the same time and in the same place.

7. Conclusion

A new force display algorithm was proposed, which can be adopted for the virtual cooperative world constructed by multiple remote users. The idea is based on the successive superposition of acceleration and the motion integration. The method successfully displayed the cooperative manipulating feel of the virtual object between two remote users and proved the validity of the proposed approach. Though the experiments was performed in a simple two-dimensional virtual environment, its application is not limited to. In the future work, we will be able to implement the proposed algorithm in a three-dimensional virtual environment and also, by increasing the participants such as three, four users etc., more realistic experimental proofs will be given. Though it may not be assured that to be applied for the case that large time delay are included in the network, we believe that the proposed method can be effectively used in many useful application areas.

Acknowledgement

The authors are grateful for the support (971-1001-008-2) provided by a grant from Korea Science & Engineering Foundation (KOSEF).

References

Bricken W., Coco G., 1994, "The VEOS Project," *Presence*, Vol. 3, No. 2, pp. 111~129.

Buttolo P., Hewitt J., Oboe R., and Hannaford B., 1995, "Force Feedback in Virtual and Shared Environment," Presented at the student poster/exhibit session," *IEEE Int. Conf. on System, Man and Cybernetics*, pp. 4656~4661.

Calvin J., Gaines A., Metzger B., Miller P. and Owen D., 1993, "The SIMNET Virtual World Architecture," *Proc. of IEEE VRAIS*, pp. 450~455.

Gossweiler R., Laferriere R. J., Keller M. L., and Pausch R., 1994, "An Introductory Tutorial for Developing Multiuser Virtual Environment," *Presence*, Vol. 3, No. 4, pp. 255~264.

Hyoukryeol Choi, Seungryong Lee, and Sungmoo Ryew, 1998, "Display of Operating Feel of Virtual Tool in Frictional Contact with Elastically Deforming Environment," *Transactions of the Korean Society of Mechanical Engineers*, Vol. 22, No. 4, pp. 790~800.

Ishii M., Nakata M., and Sato M., 1994, "Networked SPIDAR: A Networked Virtual Environment with Visual, Auditory, and Haptic Interactions," *Presence*, Vol. 3, No. 4, pp. 351~359.

Iwata H., 1990, "Artificial Reality with Force-feedback," *ACM Computer Graphics*, Vol. 24, No. 4, pp. 165~170.

Macedonia M. R., Zyda M. J., Pratt D. R., Barham P. T., and Zeswitz S., 1994, "NPSNET: A network software architecture for large-scale virtual environment," *Presence*, Vol. 3, No. 4, pp. 288~308.

Massie T. H. and Kenneth S. J., 1994, "The PHANToM Haptic Interface: A Device for Probing Virtual Object," *ASME DSC-C55-1*, pp. 295~301.

Nakamura Y., Nagai K., and Yoshikawa T., 1989, "Dynamics and Stability in Coordination of Multiple Robotic Mechanisms," *The International Journal of Robotics Research*, Vol. 8, No. 2, pp. 44～60.

Philips C. B., Zhao J., Badler N. I., 1990, "Interactive Real-time Articulated Figure Manipulation Using Multiple Kinematic Constraints," *Computer Graphics*, Vol. 24, No. 2, pp. 245～250.

Tanie K. and Kotoku T., 1993, "Force Display Algorithms," *Lecture Notes for IEEE Workshop on Force Display in Virtual Environments and its Application to Robotic Teleoperation*, pp. 60～78.

Yamamoto K., Ishiguro A., and Uchikawa Y., 1993, "A Development of Dynamic Deforming Algorithms for 3D Shape Modeling with Generating Interactive Force Sensation," *VRAIS* '93, pp. 504～511.

Yoshikawa T. and Ueda H., 1996, "Construction of Virtual World Using Dynamics Modules and Interaction Modules," *Int. Conf. on Robotics and Automation*, pp. 2358～2364.

Yoshikawa T., Yokokohji Y., Matsumoto T., and Xin-Zhi Zheng, 1995, "Display of Feel for the Manipulation of Dynamic Virtual Objects," *ASME J. Dynamic Systems, Measurement, and Control*, Vol. 117, No. 4, pp. 554～558.