ICAT'98

Distributed-Shared Virtual Environment Using Computer Network

Hiroshi HORII, Toru ABE, Susumu HORIGUCHI

School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Tatsunokuchi, Nomi, Ishikawa, 923-1292, JAPAN
{a-horii,beto,hori}@jaist.ac.jp

Abstract

Multi-user remote communication in the virtual environment has been constructed for training, education and entertainment and so on. User can experience touching and manipulating virtual objects,but they cannot feel their haptics such as weight, surface smoothness, compliance,or temperature. Virtual environment without force feedback may prevent users concentrating to interaction in it. So virtual reality system with haptic devices to present force feedback to users are studied. But most of them are single user environment and not assumed that communicating to other virtual environment systems. Shared haptics virtual environment system is a telecommunication method with kinesthetic informations providing force feedback to users as the operators or the receptors. It allows multiple virtual worlds connected on a network, shared haptic informations. In order to implement shared haptic virtual environment, it is important to solve some problems about communication time delay and stability of control blocks for haptic devices. In this paper, we propose a new architecture of shared haptic virtual environment. We discuss implementation of cooperative shared haptic virtual environment. We present to basic architecture of our system that allows us interact between virtual environment and multiple users simultaneously.

1 Introduction

In recent years shared virtual environment(SVE) has been investigated. Multi-user remote communication in virtual environment has been constructed for training, education and entertainment and so on. However, virtual environment system without force feedback may prevent users feeling of manipulating virtual objects. User can experience touching and manipulating virtual objects, but they cannot feel their haptics such as weight, surface smoothness, compliance,or temperature. Therefore virtual reality system with haptic devices to present force feedback for users are studied and developed[1]. However, most of them are single user environment and not assumed communications among the other systems. Shared haptic virtual environment(SHVE) system is a telecommunication method with kinesthetic informations providing force feedback to users. Virtual reality and telecommunication are closely related[2]. Especially network-based SVE can achieve multiple virtual worlds connected by the network to share information with one another. In order to implement network-based SHVE, it is important to solve some problems about communication time delay and stability of control blocks for haptic devices. In the first problem, the operation and modification to the shared virtual environment by users at different host may result in diverging representation. Coherency among the virtual environments must be kept for any interactions. In the second problem, some communications between the hosts use public network such as Internet. Throughput of Internet communication is not guaranteed generally. In virtual environment with haptics devices, communication delay for force feedback can easily bring control blocks of devices instable. Reducing delay is key to implement SHVE. In this paper, we propose a new architecture of shared haptics virtual environment. We discuss on the implementation of collaborative and cooperative shared haptic virtual environment. We present a basic architecture of our system that allows us to intarct between virtual environment and multiple users simultaneously.

2 Related work

There are some examples of shared virtual environment for multiple participant[3][4][5]. Division et al. had developed dVs for multi-user environment[6]. dVs consists of the independent parallel processes called "Actors". A set of standard processes is provided to handle input and output devices, collision detection, and maintain consistency of the environment. The dVS uses is a shared database consisting replicated data. Updates are controlled by a process called "Director". Processes can modify local data, but to change to the shared environment they must request to the Director. Each process holds only copies of the subset of the database, for example, a renderer stores only the visual properties of items in the database.

Hagsand et al. had developed DIVE for multi-user environment[7][8] at the Swedish Institute of Computer Science. DIVE is a distributed virtual reality environment that supports multiple users, worlds, and applications distributed over a heterogeneous network. And it is a coupled system based on UNIX and Internet networking protocols. It is a kind of database system sharing over network, with a set of processes making concurrent updates.

Zyda et al. had developed NSPNET for multi-user environment for military training simulation with the goal of supporting large numbers of participants[9][10].NSPNET is a networked VR system and use of the standard DIS(Distributed Interactive Simulation).

Benford et al. had developed MASSIVE for multi-user tele-conferencing applications[11]. MASSIVE currently runs on Sun and Silicon Graphics workstations. The main feature of MASSIVE is that it supports multi users, applications.

Ishii et al. had developed Spidar for haptic interaction with virtual environment[12]. Spidar was the first successful implementation of a multi-user haptic simulation.In Spidar ,two users can grasp and manipulate the same virtual object simultaneously.

Hannaford et al. had developed shared haptic virtual environment system and FFFMS(force Feedback Multi-player Squash).FFFMS is collaborative application for interaction between long distance[13][14].

3 SHVE Overview

SHVE architectures can be grouped in three major classes: static, collaborative and cooperative [13][14]. In a static SHVE, each user can recognize some outputs as virtual objects by looking and touching. However, they can not modify and touch any part of environment and objects and each other. Figure 1 shows a static SHVE in which the virtual environment on a server is duplicated and distributed to users. It is like a World Wide Web or shared databases on the network. Figure 2 shows a collaborative SHVE. In a collaborative SHVE, user can modify and edit the environment but may not operate to the virtual object at the same time. Users shared a right to edit an object according to some predefined rule. Owner of the right to edit can operate virtual objects, the other users only can get duplicate of official copy from server. This scheme can be applied for surgical or professional training, co-located CAD and entertainment. Figure 2 shows a cooperative SHVE. In a cooperative SHVE, users can modify and operate to the same object at the same time. User can recognize existence of other users directly or indirectly through a virtual object. A cooperative SHVE is ideal and closest to real environment. It is suitable for training of team operating like a surgical simulation and entertainment with force feedback and so on. However, implementation of a cooperative SHVE is much more difficult than other SHVE class. In a Static and collaborative SHVE, only one user kinesthetically interact with virtual objects. So each host receives a few fundamental informations of the virtual objects, collision detection and dynamic generation and so on. In a cooperative SHVE processes to generate virtual environment must be processed and managed on the central server to reduce communication delay.

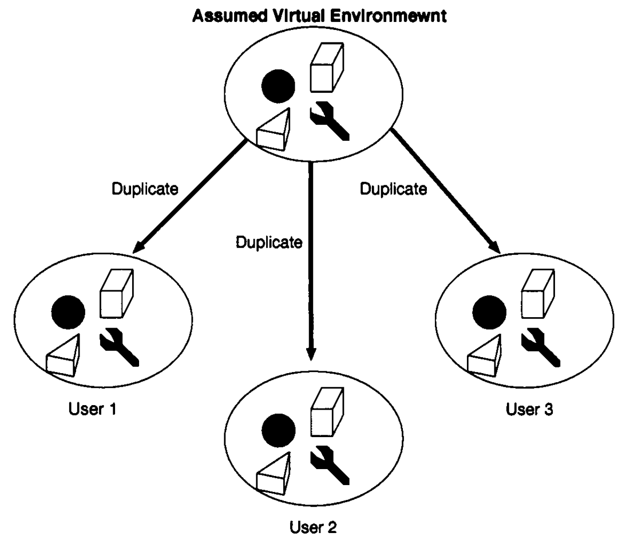

Figure 1 Static SHVE

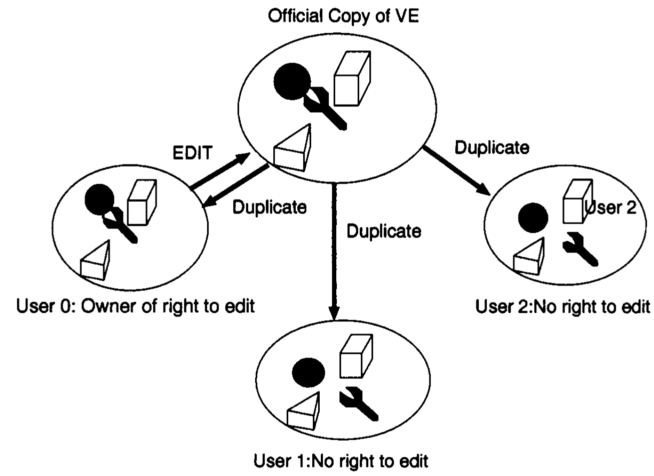

Figure 2 Collaborative SHVE

We focus on implementation of collaborative and cooperative SHVE and propose a new method for reducing communication delay for feedback to haptic devices. In the next section, we describe construction of our system.

4 System Architectures

The most different point between the virtual environment without force feedback (only graphic interface) and the systems with haptic devices is rendering

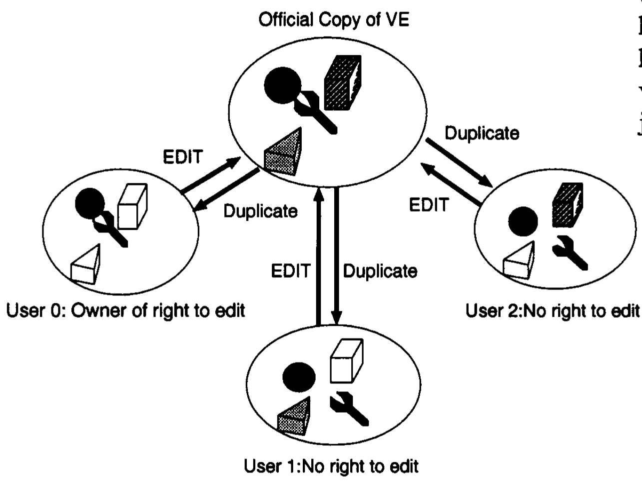

Figure 3 Cooperative SHVE

refresh rate of graphics or haptics. There are few fundamental processes to generate virtual environment, collision detection for each virtual objects, control input and output devices, and so on. On the graphics base virtual reality systems have to provide user 10-60 Hz refresh rate for continuous graphic rendering. And graphics rendering process for visual devices (such as Head Mount Display) needs no informations about feedbacks from outside such as human motions and forces. On the virtual environment system with haptics devices, it is necessary more high refresh rate for stable haptics rendering. In general, haptic devices consisted of motors and metal parts, the refresh rate comes up to nearly 1000 Hz. Delay of feedback from outside can easily make control blocks of haptics devices instable. To reduce the communication and computation delay, infomatiom processing is the key problem to implement cooperative SHVE.

To implement our system, we set goals; computation delay reducing in local architecture, and avoiding instability by distributing of system. In this section, we describe about our system architectures.

4.1 Distributed Environment

Synthetic fundamental processes to generate virtual environment is one of the feature of cooperative SHVE. It is necessary to keep coherency of the environment any interactions. As the number of users and virtual objects increase, load of the server may increase, and refresh rate for haptic rendering may decrease. Because of concentrated architecture, the worst delay among local hosts affects whole of system. So we propose the distributed of collision detect process and the dynamic generation process. By distributing of their computational process, it can avoid concentrating load on a specific server.

On visual environment such as the system only with video display, user seems that all objects have relations between others. However if any objects does not contact each other, there are no dynamic relations among them. In consideration of dynamic relation, haptic virtual environment consists of some local groups. Figure 4 shows some rules to divided virtual environment into local groups consisted of objects.

- Each tool, the object operated by users directly, has the predetection area. The form of predetection area can be detected uniquely by object size.
- Objects in predetection area belong to its tool group. They are candidates for collision detection and dynamic generation on local host.
- If an object was detected contacting with a tool, the object also becomes a part of predetection area and becomes a part of tool.

The predetection area can reduce calculation for collision detection. In general, collision detection on concentrated architecture receives to search the whole space of virtual environment in the server. On the other hand, some hosts with haptic device can concurrently search in their own virtual environment. Figure 4 shows the classification of the structure between objects.

Object-Object: This structure is the most basic and static state of virtual environment. In the environment with the mass and the inertia of objects, state becomes dynamic. An affect of computation delay in this group is not serious for stability of haptic devices. Collision detection and dynamic generation for this group is processed on the object manager.

Tool-Object: Computation performance of collision detection and dynamic generation affect the delay and the stability of haptic devices. However, some techniques reduce the computations, such as low resolution rendering or reducing stiffness on force feedback, can improve stability of the system.

Tool-Tool: users are directly and dynamically interacting with each other. The motion of each tool are complicated and not expectable for computation. Moreover, in this state, the whole haptic devices in this group must be stable to interact with each user, and delay on one host affect to all users.

The structure consisted of local group is suitable for distributed processing environment. As state of virtual environment is changed by various conditions, the structure of local group and the amount of computations are also changed. On the Tool-Object and the Tool-Tool group, high refresh is necessary to render haptics. On the other hand, on the Object-Object

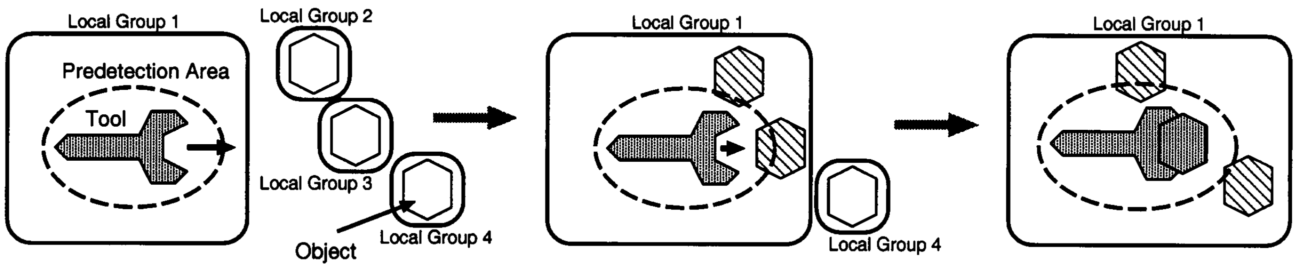

Figure 4 Generation local group with predetection area

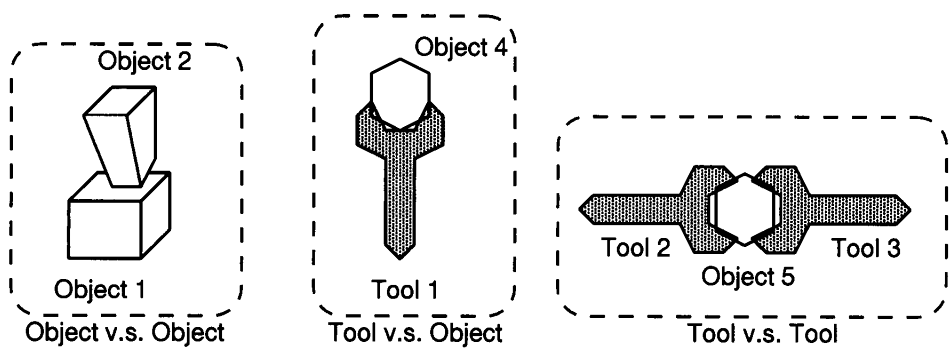

Figure 5 Classification of local group

group, delay of processing may not affect the stability of haptic devices and systems. In our system, a local host with haptic device is allocated to processes the Tool-Object and the Tool-Tool group. And the Object-Object group is processed on the object manager shown in Figure 6. We consider that the key to implement cooperative SHVE is the maintenance stability. A trouble on a haptic device makes the other part of connected system instable easily. In this structure, each Tool is managed by the tool manager. For the Tool-Tool group, each host interact with only their tool manager and processing another tool as object like the Tool-Object. This structure makes it easy to maintain the stability on whole system by tool manager, if delay makes any haptic device instable.

4.2 Object Manager

The object manager consists of three parts; communication and database management part, time sharing management part, and calculation of scene parameters part. The algorithm of object manager is shown in Figure 7. The communication and database management part is interface between tool manager and object manager. The informations about virtual objects are broadcasted to all tool managers, and result of users interactions in local groups from tool manager are obtained. Databases about the properties of virtual objects and environments are also managed by this part. The time sharing management part maintains time table, tool table and time stamp. The informations from tool manager contain the reference to time stamp and host name. Time table and tool table keep them in buffer. The calculation of scene parameters part composes complete environment on server from distributed information processed in tool manager. The collision detection and dynamic generation in Object-Object group are processed in this part to generate scene in virtual environment. The results of processing in this part added times tamp and broadcasted each tool manager.

4.3 Tool Manager

The tool manager consists of four parts; communication part, time sharing management part, calculation of distributing part and calculation of local parameters part. The algorithm of tool manager is shown in Figure 8. The communication part is interface between tool manager and object manager,and tool manager and local host. The informations about virtual objects broadcasted by object manager are obtained. In the calculation of distributing part, local groups consisted of virtual tools and objects are generated and sent to each local host. The calculation of local parameters part composes local environment from some results of dynamic generation.

4.4 Local Architecture

We consider that reducing delay in local host is important to implement the cooperative SHVE. The local host has control blocks of haptics devices and

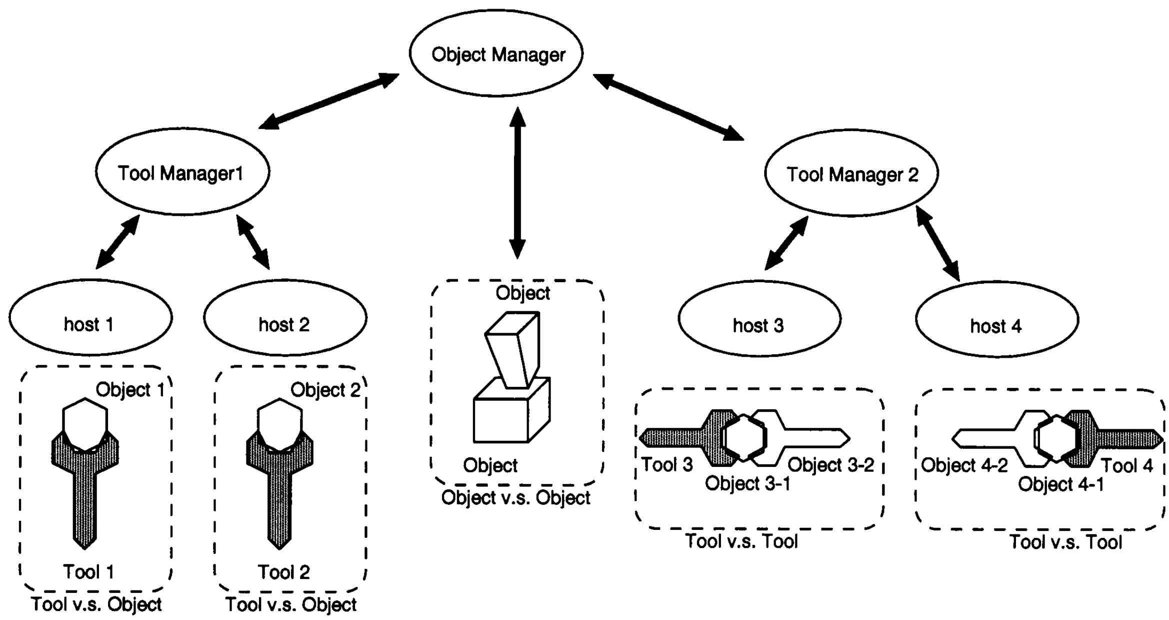

Figure 6 The structure of distributed processing environment

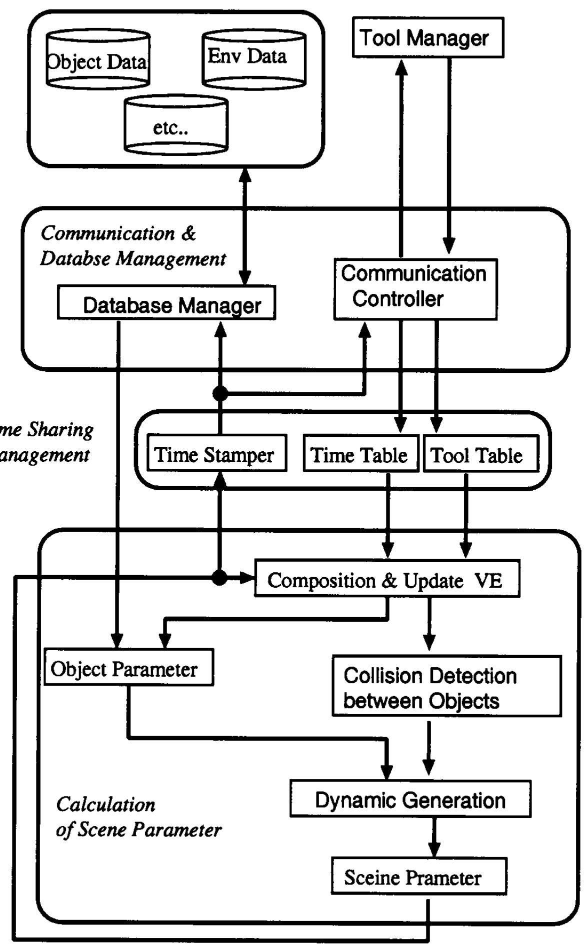

Figure 7 Proseccing Algorithm of Object Manager

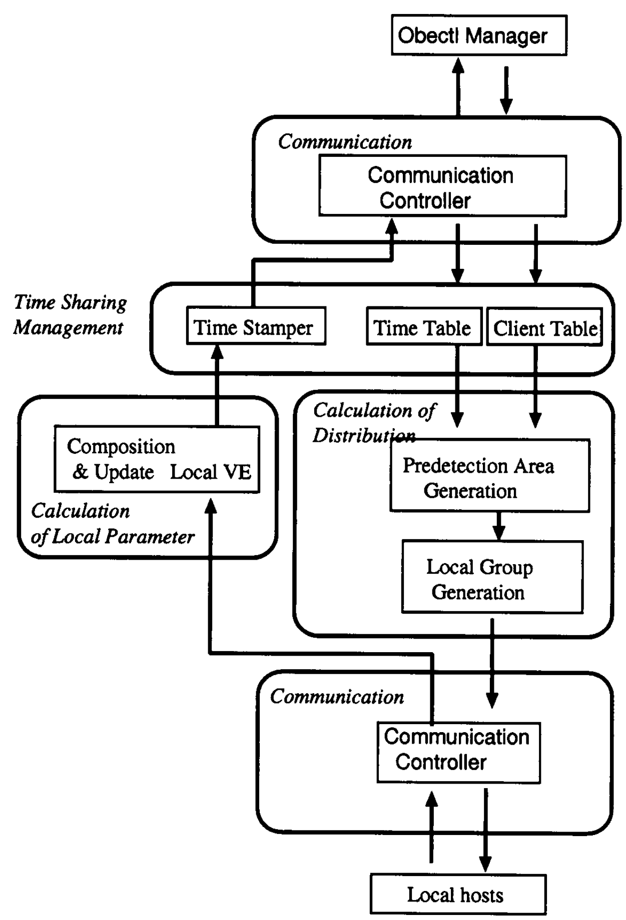

Figure 8 Proseccing Algorithm of Tool Manager