ICAT 2001
December 5-7, Tokyo, Japan

Tele-manipulation System with Synchronized Processing Method

Kazuaki Tanaka, Munehisa Kusano and Norihiro Abe

Kyushu Institute of Technology,
689-4, Kawazu, Iizuka-shi, Fukuoka-ken, 820-8502, JAPAN
kazuaki@mse.kyutech.ac.jp

Abstract

In this paper, it is shown about the tele-manipulation system of the robot hand, which is using a haptic sensation and vision. The haptic sensation is effective in the intuitive operation of the robot hand. And the vision is indispensable to grasp the work of the robot hand.
This system is composed of the master machine which a user operates, and the slave machine which controls a robot hand. When a robot hand is operated, a user uses PHANToM of the master machine. The foece of the user that PHANToM receives is transmitted to the robot hand at the slave machine. A robot hand at the slave machine is moved corresponding to the user's force. And, a slave machine captures an image from the camera which focused on the robot hand. The foece from the robot hand is acquired with the foece–torque sensor at the same time with capturing.
Delay occurs when it tries to use a robot hand through the network. Long time delay influences the intuitive operation of the robot hand greatly. Time delay causes the following problems.

- The difference between the state of a robot and that indicated in a television
- The delay caused the output of FTS to the PHANToM
- It is difficult to synchronize the TV image with the force returned with a PHANToM

The technique which synchronizes the condition of the slave with the master is shown by this paper for the tele-manipulation. Then, time delay along with the remote control is explained. And, at the end, a future subject is shown about the technique to use virtual space for the presentation of the force.

Key words: tele-manipulation, foece feedback, robot arm, vision

1. Introduction

Recently, a tele-manipulation system of a robot arm with a master/slave system is performed at many research institutes. The system is constructed that gives a user the feeling as if he were at a remote place by introducing the visual and haptic sensation. These sensations are indispensable for a user to manipulate a tele-robot correctly. As the visual sensation is necessary to a user, a television image is displayed at a master side through a network to check the behavior of the tele-robot arm. And, haptic sensation is given to the user with a force feedback device (PHANToM) to show when a robot hand comes in contact with an object. But, a tele-manipulation system using a network has some problems. Especially, "The difference between the state of a robot and that displayed in a television" and "The time difference between the output of FTS and that of the PHANToM" may destroy the environment because they cause a user erroneous operation. The cause of the delay between the state of the robot and that displayed in a television is as fellows:

(1) Time needed to send a user's command to the slave system through a network.
(2) Time taken to move a remote robot arm to the destination specified by a user.

The cause of the delay between the output of FTS and the reaction with the PHANToM is as fellows:

(1) Time needed to send the output of FTS to the PHANToM.
(2) Time taken to return the output to a user's hand with the PHANToM.

This time, we contrived two methods to solve these two delays. The first method is to synchronize the behavior of a robot arm with user's commands. We suggest a method not to have an operator fail in the operation by measuring in advance the delay time needed to move a remote robot arm to the destination. The second method is to process the FT data from the FTS to make the time taken in the reaction with the PHANToM minimum. The inter process communication generally applies a single-thread method to process received data, but we applied a multi-thread method to keep down the delay between the output of FTS and the reaction with the PHANToM. The details of the method and the experimental result are shown.

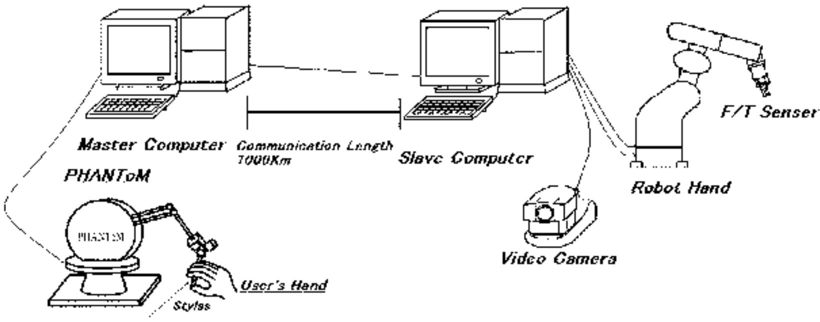

Fig. 1: Master/Slave System Configuration

2. System configuration

2.1 Master/Slave system

The tele-manipulation system configuration is shown in Fig.1.The user gives commands to a remote robot arm with the PHANToM at the master side. The Command data are delivered to the slave computer through a network and are given to the remote robot arm. The master computer can send a command to the video camera at a slave system. The slave computer sends the FT data obtained from the robot hand and the television image capture from the video camera to the master computer through the network. Then, the user can get the visual and haptic sensation at the remote location. With the second one the delay time between the output of the FTS and that of the PHANToM is kept minimum.

2.2 Force torque sensor (6 output degrees of freedom: FTS)

Fig.2 shows the FTS mounted on a robot hand in a teleoperating experiment. The change of the FT data informs the user of the occurrence of collision between the robot hand and an object. The data sent from the slave system are given to the PHANToM just as it is and are output as the reaction force with the PHANToM.

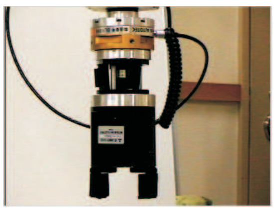

Fig. 2: Force Torque Sensor (FTS)

On this experiment, a PHANToM with the input of 6 DOF is used as a force feedback device(Fig.3). The PHANToM is usually exploited to a user a force which is generated by the interaction between invisible phantom objects. However, in this system the PHANToM is used to return the force returned from the FTS mounted on a robot arm. Though it has 6 FOD with respect with the input, it cannot return the torque output. The DC motor and 3 encoders are attached to the device. The device measures discrete values of rotational angle of the arm joint and outputs a haptic sensation. A user specifies both the position and orientation of a robot arm by operating a stylus by hand.

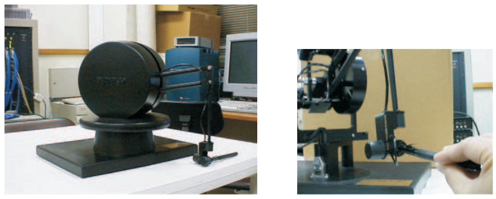

Fig. 3: Force Feedback Device (PHANToM)

2.3 Robot arm and VR simulation

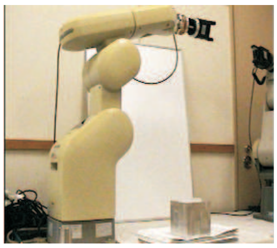

Fig. 4: Robot Arm (6 degrees of freedom)

Fig. 5: Robot Arm in VR

The task given to an operator is to have it to hold and move an object in the environment shown in Fig.4. It receives the user's commands and moves the hand to a directed position, but it takes long time for the robot to arrive at the position. A VR simulation robot arm is shown in Fig.5. The VR simulation is created in the same measurement and proportion as the real robot arm.

A robot hand is being built in the virtual space. Not only a robot hand but also a working environment itself is mounted on the virtual space. Collision with the robot hand can be checked about the objects (assembling parts and/or obstacles) put in the working environment. Therefore, a actions of real robot hand and a responses can be shown by using the virtual robot hand.

Even a virtual robot hand must present the movement that an actual robot hand is smooth. A master machine can render virtual space by using the OpenGL hardware acceleration device in real time. Then, users can use virtual space comfortably.

2.4 The implementation

The user application is implemented to realize the two methods that we proposed. The image of the robot arm is captured with a video camera sent to the application through a network and displayed in the application software. If the user operates the robot arm to use the VR simulation robot arm, they can operate both the real robot arm and VR robot by making correspondence between a stylus and the VR simulation robot hand. Each angle of the robot arm is shown in the lower left part of the window.

The user can operate a video camera (zoom up/down and pan/tilt) with pushing the application buttons in the middle lower part. When a robot is operated, the pan/tilt function of the camera is convenient. A robot hand can't be followed when an image from the fixed camera is used. We must operate a robot hand corresponding to the work in a place to vary. Therefore, the direction of the camera is changed corresponding to the place to accomplish the work.

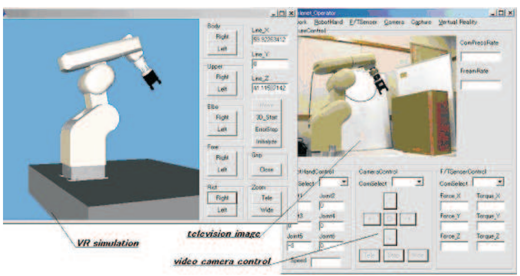

Fig. 6: Implemented Application

3. Synchronization process

The delay of the operation and the response isn't avoided in the remote control which a network was used for. The time delay of the operation influences the operation of the intuitive robot greatly. In this section, the cause of the time delay and the necessity of the synchronism are mentioned.

3.1 The delay between the robot and its camera image

When the user uses this application, it takes time 500–1800msec for the robot arm to finish the movement. This delay time is caused by both the time needed to send a user's command to the slave system through a network and that taken to move a remote robot arm to the destination specified by a user. The delay makes the user an erroneous operation because it makes difficult for the user to give its intention to the robot.

If a robot hand is only operated, it doesn't hang on a time like this. The operation of the hand is repeated by the remote control of the robot hand with verifying whether a robot being working properly. The condition of the robot hand is confirmed with a camera. Because of that, an image is captured with a camera, and that image must be shown to the user.

It lacks bandwidth, and long delay occurs when a captured raw image is transmitted to the network. Image encoding is done, and compression with a JPEG form is being used with the implemented application. But, time delay won't be avoided because image encoding takes time, too.

3.2 The delay between the FTS output and the reaction with the PHANToM

The output of FTS (FT data) is sent from the slave computer and received with the application in the master computer. Then, a FT data is output as the reaction with the PHANToM. A single-thread processing method is generally applied in an inter process communication of data. This application receives a television image and the FT data as communication data. If the application deals with the received data with a single-thread method, it takes long time to process a large amount of data such as images. As a result, almost all the CPU time is spend in a data processing other than the FT data, then the reaction with the PHANToM delays.

3.3 Synchronization of haptic sensation and visual sensation

The prescription for two delays is to synchronize the haptic sensation with the visual one. Then, we propose two prescriptions to force them to synchronize each other.

(1) The system forces the user to wait for a next instruction until a robot finishes the arm's current

motion.

(2) Applying the multi-thread processing method to process received data (a television image and FT data).

The multi-thread processing method generates multiple threads and processes data in parallel. This makes it possible to output the television image and the reaction with the PHANToM much faster than the single-thread one by allocating the adequate time slice to each thread.

multi-thread is the technique being used widely with the application which uses a network. Even the system which we mounted is using a network, and it is not special with the technique until now in that point. Furthermore, synchronism with the virtual space and the actual space was realized by using multi-thread. Because a smooth movement is demanded, the presentation of the virtual space is carried out with high priority thread. On the other hand, concurrent execution is more necessary for the communication concerned with the remote control than a speed. Then, the priority of this thread is low.

In addition to this, the acquisition of the force torque sensor and the displaying of the camera image are done in parallel, too. These concurrent executions could be solved with multi-thread.

3.4 Avoidance of an operating error owing to the delay

It was shown that time delay along with the remote control occurred in the previous section. multi-thread was used so that the influence of the time delay might not make the acquisition of the power torque sensor and the indication of the image worse. In this section, we will describe more specifically these prescriptions. Synchronizing the motion of the VR simulation robot arm with that of the robot arm needs to measure processing time, so the following parameters are defined:

$$Tvr, Ttr, Tro \approx Fro(L)$$

Tvr shows the time taken for the VR robot arm to move from the current position to a directed position. Ttr shows the time taken to send a user's command from the master computer to the slave one. Tro shows the time taken for the robot arm to move from the current position to a directed position.

A function $Fro(L)$ calculates the moving time of a robot arm. Here we show a mathematical formula to approximate the function Fro:

$$L = \sqrt{(X' - X)^2 + (Y' - Y)^2 + (Z' - Z)^2}$$

(X', Y', Z') is the next position instructed by a user, and (X, Y, Z) is the current position of the robot arm. Especially, function $Fro(L)$ is determined applying the least squares method to sample moving time of a robot arm. When $Fro(L)$ was looked for, the following function could get it by the our simple experiment.

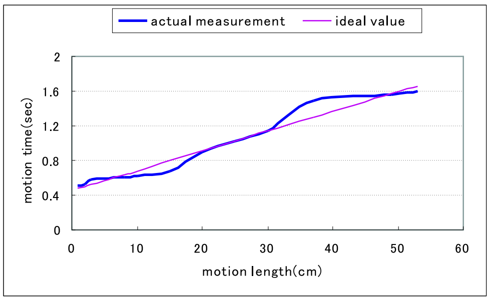

Fig. 7: Robot Arm Motion Time $Fro(L)$

$$Fro(L) = (0.022765 \pm 0.000096)L + (0.457812 \pm 0.009106)$$

The following gives a condition to synchronize a VR simulation robot arm with a real robot arm.

$$Tvr + t = Tro + Ttr$$

The left hand of previous equation is a processing time required of a master computer, and the right hand side is that of a slave computer. Really, Tvr is shorter than $Tro + Ttr$ because there are delay components caused by the transition of the real robot and a network transmission. So t must be introduced. t is a regulation time to synchronize them.

$$t = Tro + Ttr - Tvr$$

It is understood that it is possible that it synchronizes a virtual robot hand with the movement of the real robot hand from this equation. And, in the synchronism, Tro,Ttr and Tvr is necessary. Tro,Ttr depends on a communication path and the performance of the device, and Tvr depends on graphic drawing performance and the complexity of the virtual space. Graphic performance must be comparatively high from the condition of $Tro + Ttr > Tvr$.

An experimental result obtained by the synchronization process is shown later. The prescription for the delay between the output of FTS and the reaction with the PHANToM is realized with a multi-thread method. As the data processing time changes according to the amount of the data, we examined the affect of a delay by changing the compression rate and the frame transfer rate of images. Further, comparing the throughput with a single-thread method to that with a multi-threads one, we have estimated how much improvement will be expected with the multi-thread method.

4. Experimental result and consideration

4.1 Environment of experiment

An experiment about the remote control of the robot hand was done. A robot hand and a user did an experiment at the same site to verify a implemented system. However,

the communication way of going and returning 1000km was used about the communication between the master and the slave. A master and slave are being made a personal computer, and besides, it connects a robot hand, a camera and PHANToM, and it is being used.
The experiment of the tele-manipulation system is performed in the following environment:

表 1: The Environment of the Experiment

CPU	Pentium III 750MHz
OS	slave: Windows 2000 master: Windows NT
Memory	128MB
Communication	1000km (Fukuoka - Osaka - Fukuoka) Gigabit Network(JGN)
Image Encoding	JPEG

4.2 Experiment

The delay time between the image and the real state of a robot arm is estimated by comparing the case of the synchronization application to that of non-synchronization one when the robot arm is continuously operated. Also, we verified the effect of the multi-thread processing for an inter process communication. The difference of processing speed between a single-thread method and a multi-thread one is estimated by changing a compression rate and a frame rate because they affect the processing speed. The time needed for the FT data to be outputted with the PHANToM after they have been sent out from the FTS is shown in Figure 8 and 9. Figure 8 shows the relation between the delay time and the image compression rate, and Figure 9 shows that with respect to the frame rate of an image.

4.3 Experimental result

As mentioned previously, a user has to wait for 500–1800ms until a real robot arm finishes the operation given by the user, but he can attain his goal without any miss operation. The synchronization mechanism forces a user to segment a continuously trajectory into several fragments and this makes it difficult for him to operate a robot arm smoothly. Next, let us make a comparison between the efficiency of a single-thread method and that of a multi-thread one.

Followings are shown in Fig.8:

(1) There is no remarkable difference with respect to the delay time between a single-thread method and a multi-thread one.

(2) The shortest delay time is attained when the compression rate is 90%.

Fig.9 shows the followings: When the frame rate is 5–10 frames/second, the delay time gradually increases in both the single and multi-thread method. The delay time is improved by 20–30ms exploiting a multi-thread method. The delay time is the shortest at the point of 2.5frames/second. At the low frame rate from 1 2 frames/second, the single-thread method is better than the multi-thread one with regarding to the delay time. This is because the single-thread starts a process only when the data arrive, but the multi-thread always keep processing even if the data does not arrive at all whenever it is given a time slice. The shortest delay time is attained at the compression rate of 90% and at the frame rate of 2.5 frames/second in the multi-thread method. The delay time is at most 10ms.

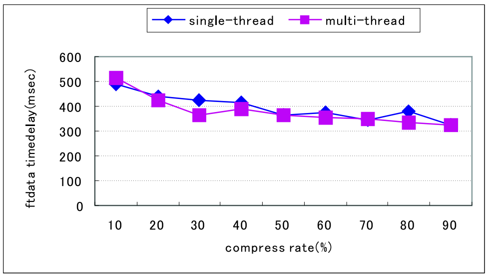

Fig. 8: FT delay in compression ratio

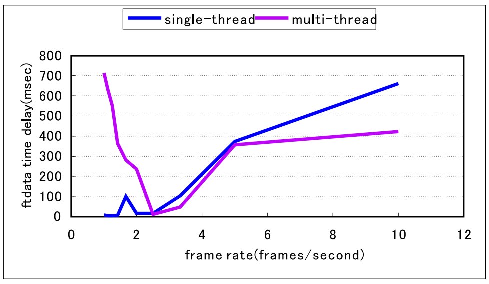

Fig. 9: FT delay in Frames ratio

4.4 Consideration

When the haptic sensation should be much more important than the visual one, a single-thread method with a low frame rate is desirable. It is turn out that a multi-thread method will arise a delay time, even if the amount of data is too much or too little. This means that the time slice allocated to each thread must be suitably adjusted to make the delay time minimum. The optimal tele-operating system will be realized by allocating the best time slice to each thread considering which of the visual sensation or the hapitc one should be important to the current purpose. From the experiment, if the visual sensation is important, then a multi-thread method is preferable and if haptic one is important, a single-thread method should be selected.

It was shown about the time delay when remote control had a robot hand by this paper. Multi-thread was adopted, and a part in problem could be solved to restrain the influence of the time delay low. But, it can't be to prevent time delay completely. Time delay is because it surely occurs as far as a network is used.

So, the environment of the intuitive remote control which virtual space was used for is proposed as a future work. Work is carried out by using the robot hand built this time in the virtual space. Even if it is virtual space, a real robot's movement is possible. So, every time a robot is operated, the force which a robot receives by checking interference in the virtual space is computed. It does not need to wait for the force from the actual robot by presenting the computed force to the user. PHANToM can be given to the force user in real time, and it can realize intuitive operation by this.

Reference

[1] Takafumi Matsumaru, Shunichi Kawabata, Tetsuo Kotoku, Nobuto Matsuhira, Kiyoshi Komoriya, Kazuo Tanie, Kunikatsu Takase: "Task-based Data Exchange for Teleoperation Through Communication Network", Journal of the Robotics Society of Japan, Vol.17 No.8, 1999

[2] Hiroyuki Shibata, Kazuaki Tanaka, Norihiro Abe, Hirokazu Taki: "Control of robot arm by Haptic Interface PHANToM", Proceedings of the Virtual Reality Society of Japan Fifth Annual Conference pp.377-380, 2000

[3] Takao Horie, Norihiro Abe, Kazuaki Tanaka, Hirokazu Taki, "Collaboration with Two Remote Robot Arms by Direct Instruction Using Stereo Vision and Haptic Master", 6th International Conference on Virtual Systems and MultiMedia (VSMM2000), pp316-322, 2000