모바일 햅틱 디스플레이를 위한 렌더링 시스템

2008

이채현

석사학위논문

모바일 햅틱 디스플레이를 위한 렌더링 시스템

이채현(李埰炫)

컴퓨터공학과

포항공과대학교 대학원

2008

모바일 햅틱 디스플레이를 위한
렌더링 시스템

Rendering System for
Mobile Haptic Display

Rendering System for Mobile Haptic Display

by

Chaehyun Lee

Department of Computer Science and Engineering

POHANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

A thesis submitted to the faculty of Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the Department of Computer Science and Engineering

Pohang, Korea
May 22, 2007
Approved by

Major Advisor: Seungmoon Choi

모바일 햅틱 디스플레이를 위한 렌더링 시스템

이채현

위 논문은 포항공대 대학원 석사 학위 논문으로 학위 논문 심사 위원회를 통과하였음을 인정합니다.

2007 년 5 월 22 일

학위논문 심사위원회 위원장 최승문 (인)

위 원 박종훈 (인)

위 원 이승용 (인)

위 원 이진수 (인)

MCSE 20052471 이채현, Chaehyun Lee, 모바일 햅틱 디스플레이를 위한 렌더링 시스템, Rendering System for Mobile Haptic Display, Department of Computer Science and Engineering, 2008, 67 p, Advisor: 최승문 (Sungmoon Choi). Text in English

Abstract

The haptic interface is an electromechanical device that can create and deliver haptic stimuli to a user. As visual displays inevitably have a fixed screen size, most of the currently available desktop haptic interfaces often suffer from their limited workspace-they cannot render real-sized objects larger than the workspace. A promising solution for this is a mobile haptic display (also known as mobile haptic interface) where a stationary haptic interface is mounted on a mobile base. This thesis presents the initial results of research toward a novel mobile haptic display, focusing on the software aspect of the whole system. When a user explores a large virtual environment, our mobile haptic display can sense the configuration of the user, move itself to an appropriate configuration, and provide adequate force feedback, thereby enabling a virtually limitless workspace. Proper software architecture for the mobile haptic display is introduced first. To avoid a collision with a user and place the mobile base at an adequate configuration, a motion planning algorithm for the mobile base with omni-directional wheels is designed and tested. We analyze the effect of the dynamics of the mobile base on the haptic stimuli perceived by the user and also provide experimental results that show the fidelity of our mobile haptic display.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

As Computer Graphics gives people information via visual sensation, Haptics does the same thing but with the sense of touch. While Computer Graphics is widely used in entertainment industries such as movie, broadcasting, animation, game, construction, advertisement, manufacturing industries, and so on, Haptics has remained in a research lab level.

The biggest reason why Haptics still remains in a research level unlike Graphics is because the haptic devices, which are used to deliver haptic sensation to users, are expensive. However, there has been much research about these haptic devices recently, and haptic devices with reasonable price are constantly being introduced to the market.(see Figure 1.1).

If the manufacturers can develop these low-priced haptic devices, and many application fields such as virtual reality games, a touchable broadcasting, and a haptic electronic commerce can be developed, desktop haptic devices can be

Fig. 1.1 SensAble PHANToM Omni.

widely installed in most desktop PCs like graphics card, in the near future. However, these desktop haptic devices usually have limited workspace like Figure 1.2 because of their design.

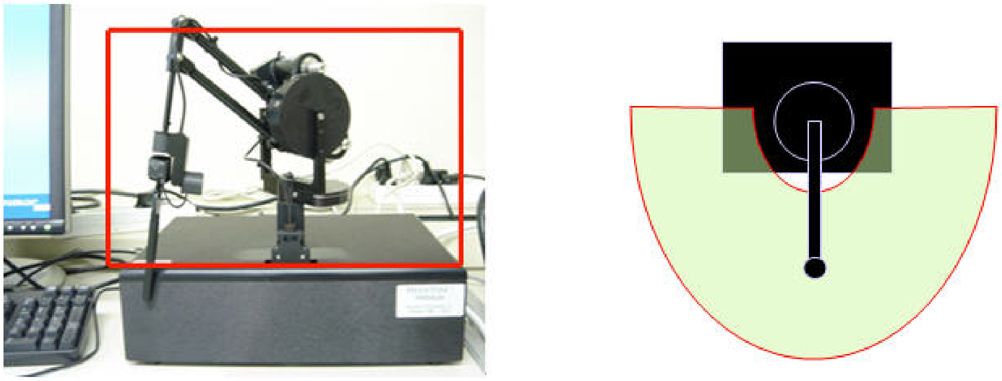


Fig. 1.2 Limited workspace of a desktop haptic device.

Therefore, users can only touch objects that are smaller than the device workspace. In order to overcome the problem of limited workspace, several haptic devices have been developed.

However, these devices have limitations in stability, detailed characteristic representation, high stiffness representation, and reusability 1.1. The concept of mobile haptic display [7] which was introduced in 2001 solves not only these problems but also problem of limited workspace.

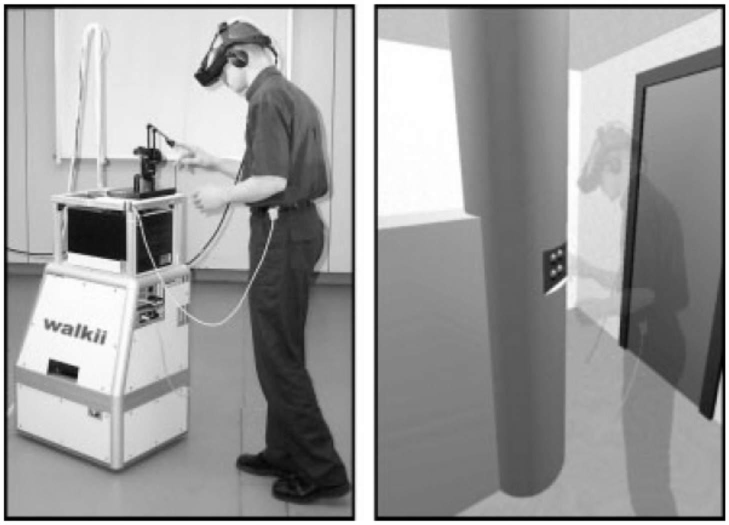


Fig. 1.3 Walkii introduced in 2001.

Mobile haptic display is the haptic device with a mobile base. Placing the desktop haptic device on a mobile base and moving the mobile base according to the user's movement enable the user to explore and feel the workspace that is virtually unlimited in size. Also users can touch virtual environments even when they are moving [7].

1.2 Related Work

Walkii, which was developed at university of Munchen in 2001, consists of a mobile robot with omni-directional wheels [11] and a desktop haptic device, PHNAToM Premium 1.0. To track the position of the mobile robot, optical, magnetical, and acoustic sensors are used for absolute position tracking, and odometry and robot kinematics are used for relative position calculation. Also, the position and orientation of the user are tracked by the magnetical tracker attached to the user's head. Mobile robot is designed to follow the user's hand. The program moves the mobile robot using a simple optimization algorithm to maximize the usability of the haptic device. In Walkii, to simplify the collision detection, only simple primitive models such as plane, cylinder, and sphere are used and the force given to the user is calculated based on the spring damper model.

Walkii is the first model which proposes the concept of mobile haptic display; however, it is a primitive prototype which just provides force feedback when the user moves. In addition, the haptic models that the user can touch are limited to plane, cylinder, and sphere models.

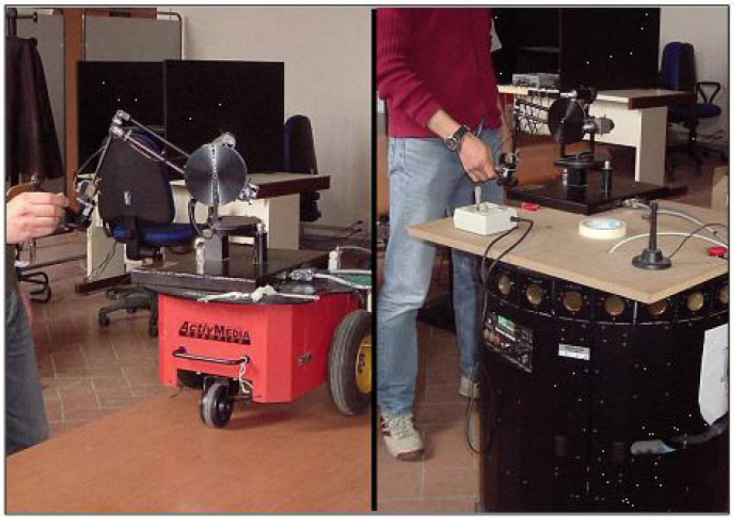


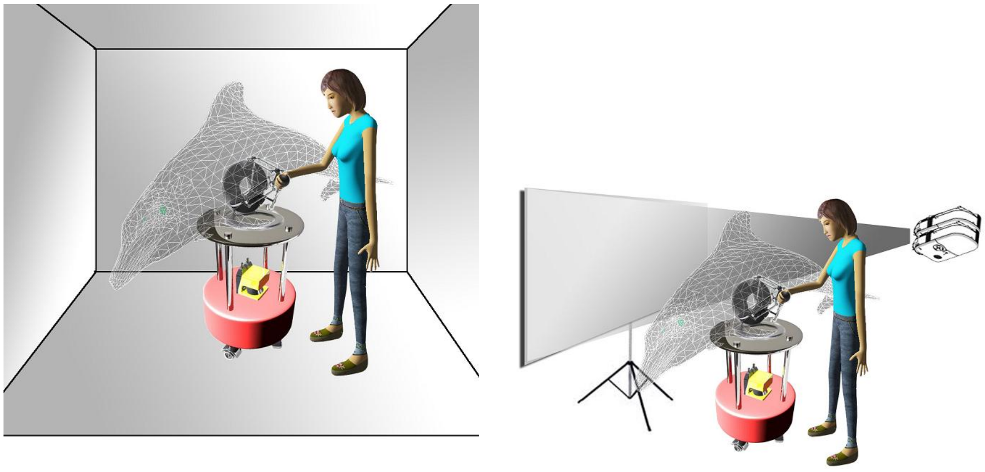
Fig. 1.4 Mobile Haptic Interface in 2004.

In 2004, Stanford University and university of Siena developed two types of experimental mobile haptic interface [10], [9]. One type uses Pioneer2DX as the mobile base, which consists of two bidirectional wheels, and the other uses Nomad XR4000, which consists of four omni-directional wheels. Both mobile robots are commercial products, and have stable performance.

Both of them use PHANToM Premium 1.5 as the haptic device. In order to provide a proper force to the user, the mobile base is moved with the path which is always perpendicular to the surface normal of the object. However, it is difficult to apply this motion planning algorithm when the object surface is rugged or when the object is complex.

1.3 Research Goal

The final goal of this research is to develop mobile haptic display for structured (e.g. CAVE) and unstructured large virtual environments such as Figure 1.5, and to develop the system for multi users using more than two mobile haptic displays and to evaluate the performance and usability of the system.



(a) Mobile Haptic Display for CAVE.

(b) Mobile Haptic Display for a Stereo Projector.

Fig. 1.5 Mobile Haptic Display for various environments.

In this paper, we limit the goal to developing the mobile haptic display system which can be operated with a tracker for position tracking and a head mounted display for visual display as a stepping-stone for our final goal. The proposed system in this paper is suitable for the mobile robot with three omni-directional wheels developed by the Robotics and Automation lab at POSTECH. In addition, we developed the motion planning algorithm that moves the mobile robot to where the haptic device can deliver force feedback most effectively.

Also, we investigated the requirements for mobile haptic display, purchased or manufactured adequate hardware, and designed and implemented the effective software architecture considering the performance of the control computer.

Finally, we examined the force effect that is caused by the movement of the mobile robot both theoretically and empirically with a force sensor.

Table 1.1: Performance evaluation of haptic devices for large virtual environments.

	Desktop Haptic Device	Largescale Manipulator type	SPIDAR type	Exoskeleton type	Mobile Haptic Display
Position Resolution	Very good	Good	Very bad	Medium	Very good
Workspace Size	Bad	Good	Good	Very good	Very good
Stability	Very good	Very bad	Medium	Medium	Very good
High Stiffness Representation	Good	Very good	Medium	Bad	Good
Detailed Characteristic Representation	Very good	Bad	Very bad	Bad	Very good
Reusability	Good	Very bad	Very bad	Very good	Very good
Reference	[1], [2]	[3]	[4], [5]	[6]	[7], [8], [9], [10]

CHAPTER 2

System Configuration

2.1 System Requirements

To operate mobile haptic display, we need to consider the following five requirements.

- Position Tracking of a user and a mobile robot
- Mobile Robot
- Haptic Device
- Visual Display
- Software Configuration

Basically, to avoid collision with the user, mobile robot needs to know the exact position and orientation of the user and itself. In addition, the motion planning algorithm depends on what kind of mobile robot is used and the workspace which the user interacts with depends on the haptic device used. Furthermore, to give the user a visual sensation which is also as important as a haptic sensation, we need to consider the visual display. Finally, to control the hardware and manage data effectively, we have to design effective and reliable software.

2.2 Position Tracking of a User and a Mobile Robot

Position tracking of the mobile robot can be classified into relative localization and absolute localization. In order to compute the force for haptic rendering, the position of the haptic device's tooltip, Haptic Interface Point (HIP), has to be known accurately. If we know the position and orientation of the mobile robot under the haptic device and the transformation from the mobile robot coordinate to haptic device coordinate, we can calculate the position of HIP. The transformation between the robot and the haptic device coordinate is constant which can be extracted by the direct measurement or Kinematics calibration (4). Therefore, if we know the position and orientation of the mobile robot in the world coordinate, we can get the HIP position in the world coordinate. That means, for the haptic rendering, the absolute localization of the mobile robot is necessary. In addition, to avoid obstacles like the wall or the user, relative localization is also needed.

Firstly, we tried using Odometry that is usually used in the position tracking of a mobile robot. Odometry is the position estimating method by using the number and direction of a revolution of each wheel; however, errors accumulated in Odometry are getting larger.

Secondly, we considered using a laser sensor. The laser sensor can track the position of the mobile robot accurately; however, it cannot detect where the user is located. Also the update rate is slow.

Therefore, we used the tracker system for estimating the position of the mobile robot and the user. In our system, we chose InterSense IS-900 as a tracker system which uses inertia and ultrasonic for estimating the position and orientation of the sensor [12].

The wired version of this model has accurate resolution, with the position of $2 \sim 3$ mm, the angle of 0.25°~ 0.5°, and the update rate of 180 Hz. The latency is 4 ms [13].

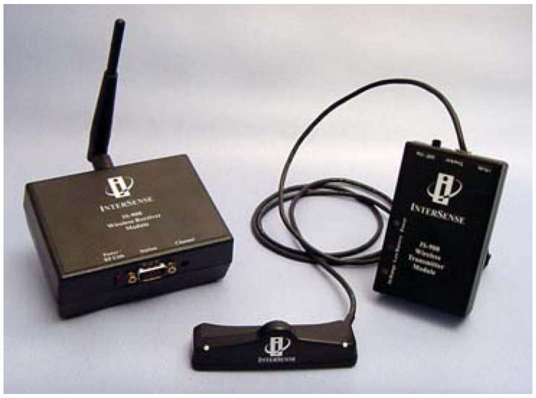


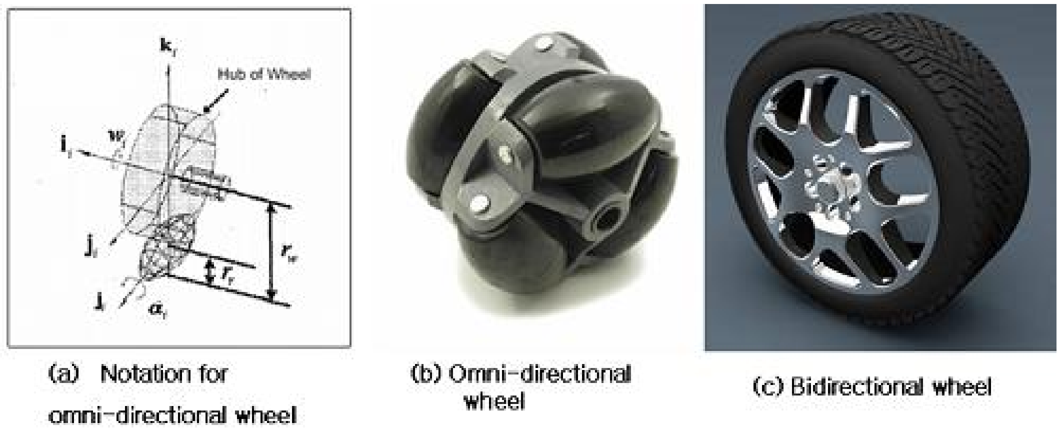
Fig. 2.1 InterSense, IS-900 tracker.

2.3 Mobile Robot

Mobile robot is developed by the Robotics and Automation laboratory at POSTECH. This robot consists of three omni-directional wheels so it can move without any directional constraints.

The mobile robot that consists of bidirectional wheels has to change the direction first, to proceed in the specific direction. During the direction change, it is possible that the HIP reaches the workspace limit of the haptic device. Considering this condition makes the motion planning algorithm complex and the mobile robot becomes unstable because of its frequent direction switches. Therefore, the mobile robot is developed with omnidirectional wheels.

In addition, to prevent slipping of wheels, three wheels which are always placed in the same plane are used. The specification and the shape of the mobile robot are shown [14].

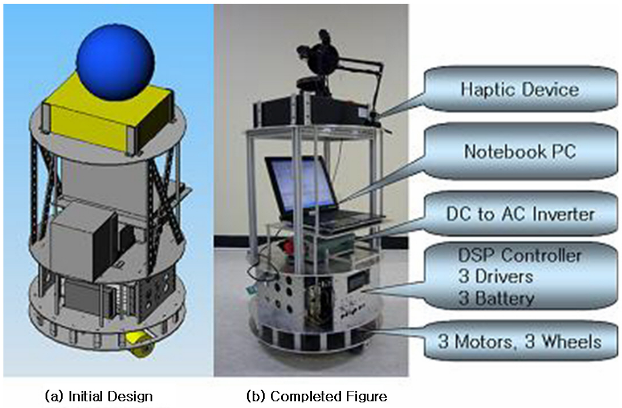


(a) Notation for omni-directional wheel

(b) Omni-directional wheel

(c) Bidirectional wheel

Fig. 2.2 Omnidirectional and bidirectional wheel.



(a) Initial Design

(b) Completed Figure

Fig. 2.3 The mobile robot developed by the Robotics and Automation Lab at POSTECH.

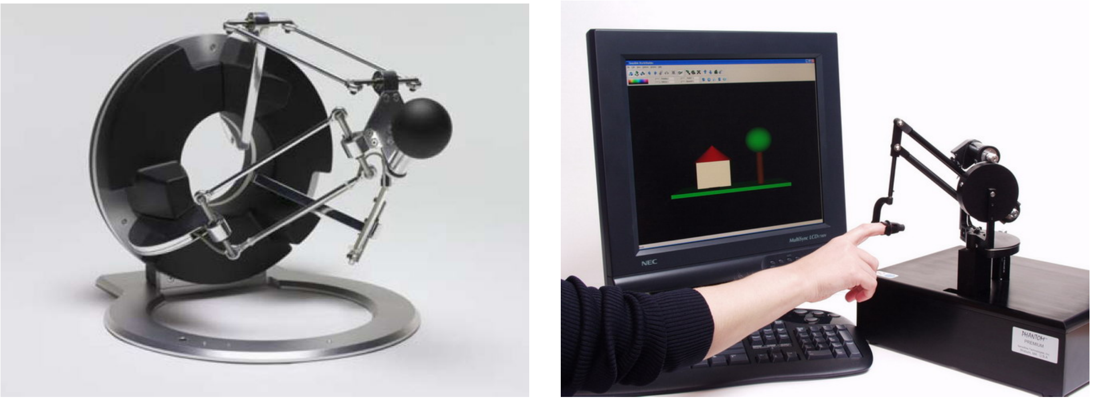
2.4 Haptic Device

Commonly used models among commercially available haptic devices are Force Dimension's Omega and SensAble's PHANToM.

Force Dimension's Omega can provide large force, 12.0N, to the user, but has relatively smaller workspace than that of other haptic devices. If the workspace

Table 2.1 Specifications of the mobile robot.

Size	50x50x70 (cm)
Weight	About 22 kg
Maximum linear velocity	0.5 m/s
Maximum angular velocity	40 °/s
Motor board	DSP Chip - TMS320F2812 clock : 100MHz control frequency : 50Hz
Wheel	omnidirectional wheel x 3 Diameter : 80 mm Roller Capacity : 45 kg



(a) Omega

(b) PHANTom Premium

Fig. 2.4 Commercially available haptic devices.

is small, HIP arrives at workspace limit too often, which results in the repeated movement of the mobile robot. Therefore, we use PHANToM Premium 1.5 for the haptic device which can provide enough force to the user and has large workspace.

Table 2.2 Comparison of haptic devices.

Model	Omega	PHANToM Premium 1.0	PHANToM Premium 1.5
Manufacture	Force Dimension	SensAble	SensAble
Workspace	140Wx140H x85D mm	254Wx178H x127D mm	381Wx267D x191H mm
Position resolution	0.01 mm	0.03 mm	0.03 mm
Maximum force	12.0 N	8.5 N	8.5 N
Stiffness	14.5 N/mm	3.5 N/mm	3.5 N/mm
Interface	USB2.0	Parallel Port	Parallel Port

2.5 Visual Display

Another important element of haptic feedback is visual sensation. Humans often feel a sense of illusion. If we can use these illusions effectively in virtual environments, we can make the user feel haptics more effectively [15].

Therefore, we provided not only the haptic sensation but also the visual sensation to the user in our mobile haptic display system. To show a real sized visual scene to the user, we considered CAVE and a head mounted display.

In the case of CAVE, the user can experience virtual environments without wearing any special equipment, but the scene obstructed by the mobile haptic display cannot be seen. Therefore, we used the head mounted display for providing visual scenes.

2.6 Software Architecture

The mobile haptic display system software consists of three parts. The tracker server program which is executed in the tracker server receives the position and orientation of each IS900 tracker and sends this information to the mobile

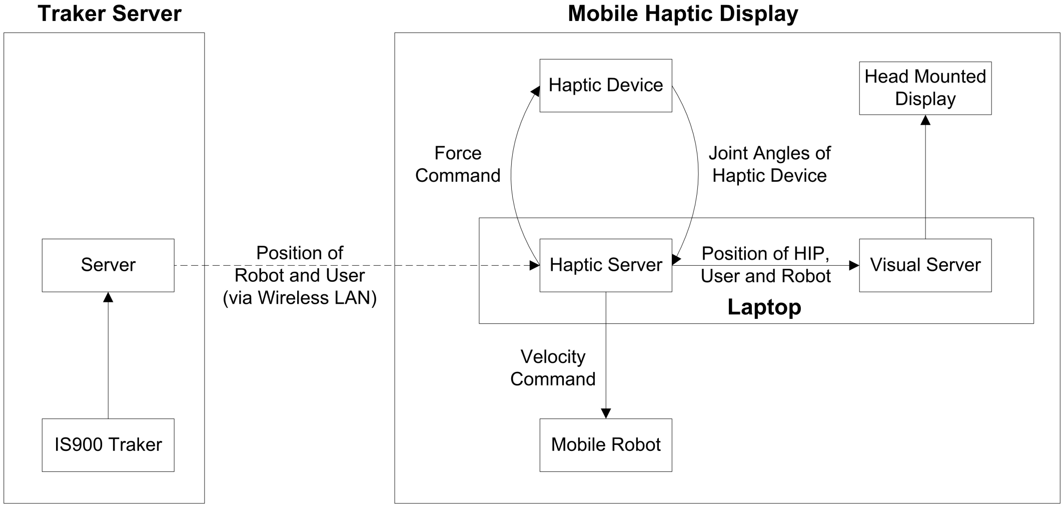


Fig. 2.5 Software Architecture.

haptic display via wireless LAN. Using this information, the software which runs on the laptop computer in the mobile haptic display controls all devices and renders visual and haptic information.

The tracker server program and the mobile haptic display communicate through wireless LAN. Due to the update rate of the IS900 tracker, the network update rate is set to 190 Hz. To maintain this speed, UDP protocol is used instead of TCP protocol. TCP, which is more reliable than UDP, is sometimes too slow and even exhibits severe jitters. Whenever a packet is missed in TCP, the protocol retransmits all packets after the missed packet and causes a long delay which is unacceptable in our application. UDP is not as complex as TCP, and is therefore faster. Even though some packets can be missed and reordered during transmission in UDP, its effect can be made transparent to the user.

The haptic server program comprises of networking, motion planning and haptic rendering parts. Firstly, the networking part receives the tracker information from the tracker server, and transforms the coordinate system from IS900 to the world coordinate frame. It updates the position and orientation of the user, the mobile robot and the HIP and sends this information to the visual server through network. Secondly, the motion planning part calculates the next proper pose of the mobile robot, and commands angular and linear velocity to the mobile robot. Finally, the haptic rendering part detects a collision between the HIP and virtual objects and calculates haptic feedback force. All of these parts run at different rates, so the haptic server is designed as a multithreaded program.

Another server program which also runs on the laptop is the visual server. The visual server receives the pose information from the haptic server and renders visual scenes. In our current system, a head mounted display is used for visual rendering, so the visual server also runs on the laptop with the haptic server. If other visual system is used (e.g. CAVETM), the visual sever can be easily moved to another machine that controls the visual display and communicate with the haptic server through network.

CHAPTER 3

Robot Motion Planning

3.1 Necessity of Motion Planning Algorithm

Basically, the mobile robot has to move to avoid collision with the user.

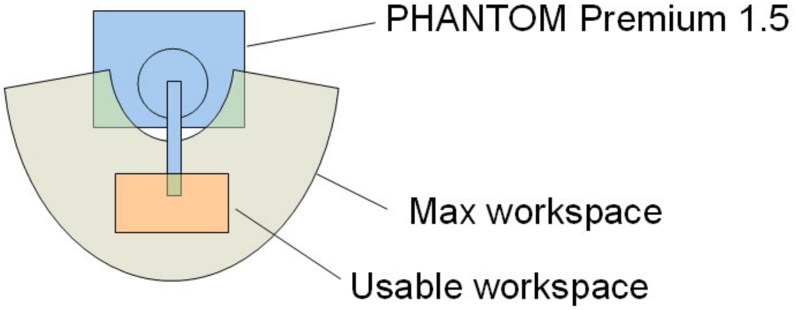


Fig. 3.1 Workspace limit and usable workspace of PHANToM (A plane figure).

According to SensAble, manufacturer of PHANToM Premium 1.5, it is guaranteed that forces can be reliably rendered within the usable workspace which is shown in figure 3.1 [16].

In other words, for the user to receive an accurate force feedback, the mobile robot needs to move to place the user's hand into the usable workspace and face the user when it moves. Considering these constraints, we designed the motion planning algorithm suitable for the mobile haptic display.

3.2 Configuration Space Design

To represent the motion of the mobile robot, we introduced a three dimensional configuration space where its x and z axes represent the 2D position of the mobile robot and its y axis the direction of the mobile robot in the workspace (see Figure 3.2).

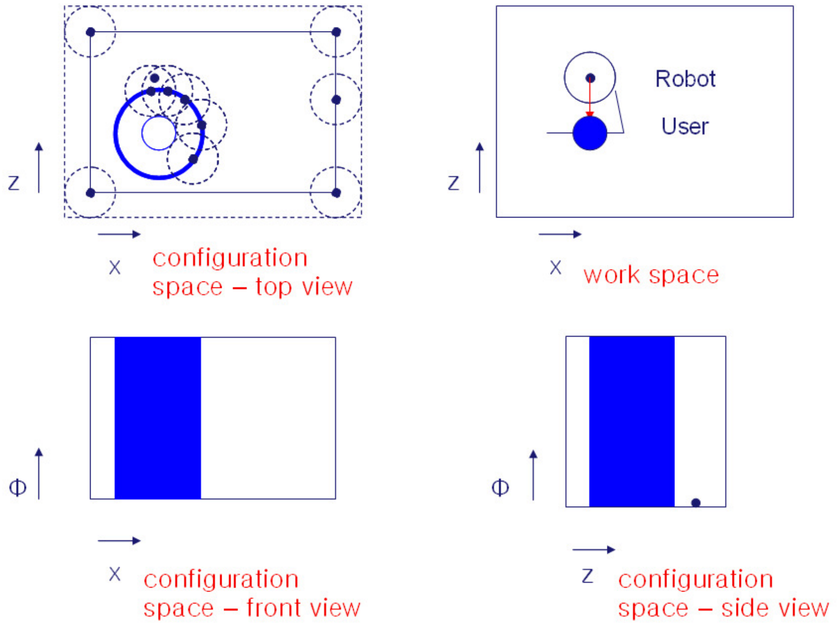

z
x
configuration space – top view
Robot
User
z
x
work space
Φ
x
configuration space – front view
Φ
z
configuration space – side view

Fig. 3.2 Configuration space.

In the configuration space, configuration of the mobile robot is represented as a point, and the direction of the mobile robot corresponds to ϕ axis. The mobile robot has to avoid the collision with the user; therefore, the user is represented as an obstacle in the configuration space. Because the user cannot collide with the mobile robot in any direction, the user is represented as the cylinder whose radius is the radius of the mobile robot plus the radius of human. In figure 3.2, (a), (c), and (d) are top, front, and side view of configuration space of the workspace shown in (b). The objective of motion planning is to find a 3 dimensional path which arrives at the target point avoiding the cylindrical obstacle (the user) in the predefined configuration space.

3.3 Target Position and Direction

To avoid collision between the mobile robot and the user, and to avoid user's hand escaping from the haptic workspace during robot movement, the target position of the mobile robot is placed in straight line with the current user position and the user's hand position (figure 3.3).

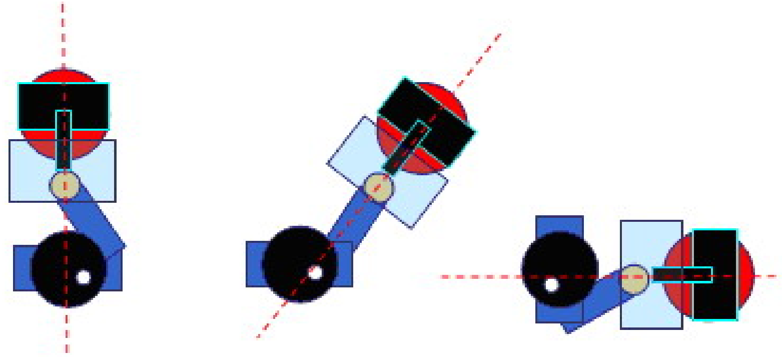
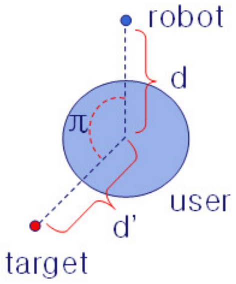


Fig. 3.3 The target position and direction of the mobile robot based on the movement of the user.

And the target direction of the mobile robot is set to always face the user.

Table 3.1 Motion planning algorithm.



```
d' = human arm length + robot radius
$\pi$' = 30
if d $<$ radius of user
    $\pi$' = 0
if $\pi < \pi$'
    SubTarget = target
else
    $\overrightarrow{User} = [User_x, 0, User_z]^T - [User_x, 1, User_z]^T$
    SubTarget = Rotate( $\overrightarrow{User}$, $\pi$' ) $P_{CurrentRobotPosition}$
change SubTarget to satisfy
    distance (user, SubTarget) == d'
go to SubTarget
```

The distance between the user and the mobile robot maintains the human arm length plus the radius of the mobile robot for placing the user's hand in the usable workspace. These target point strategies simplify the mobile robot movement caused by the user's hand movement. In addition, they move the mobile robot maintaining the specific distance between the user and the mobile robot; therefore, the collision is avoided fundamentally.

3.4 Mobile Robot Path Planning Algorithm

If the current position of the mobile robot is adjacent to the target position, the mobile robot can move to target position in a straight line. However, if the user suddenly moves a lot and the target position of the mobile robot is far from the current position, the mobile robot has to move in a smooth curve around the user not only to prevent the escape from the user but also to avoid the collision with the user. To satisfy these requirements, we propose the algorithm such as table 3.1.

In this algorithm, π represents the angle between the current and target points of the mobile robot, and π' represents the defined angle in the algorithm.

d and d' are the distance between the mobile robot and the user, and the maintained distance between the mobile robot and the user, respectively.

The motion planning algorithm proposed in this paper consists of three cases based on the current and target positions of the mobile robot.

3.4.1 When the Robot and the User are in Collision

Figures between 3.4 and 3.6 are top views of the configuration space. The blue point represents the current robot position, the red point represents the target robot position introduced in Chapter 3.3, and the big circle represents the user in the configuration space.

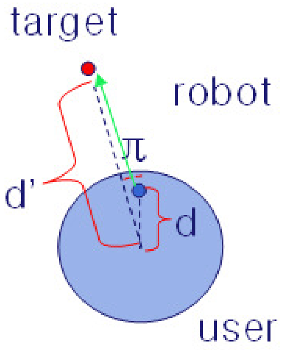


Fig. 3.4 In case of the mobile robot and the user are in collision.

First of all, the algorithm checks whether the mobile robot and the user are in collision such as figure 3.4. Although the algorithm fundamentally determines the path avoiding the collision, slipping of the wheels or an error of command process can cause the collision. As shown in figure 3.4, the blue point (current robot position) in the big circle (the user) indicates the collision between the user and the mobile robot; therefore, the mobile robot must be made distant from the user. In this case, π' is set to 0 and the temporary target position is set in the direction in which the mobile robot can escape from the user as soon as possible. In the result, the mobile robot moves in opposite direction of the user rapidly.

3.4.2 When the Robot Moves directly to the Target Position

The second case is when the mobile robot and the user are not in collision or is applied after collisions. As shown in Figure 3.5, this case is applied when the angle, π, which is between the current and the target position of the mobile robot from the user, is smaller than the angle, π', which is defined by the algorithm.

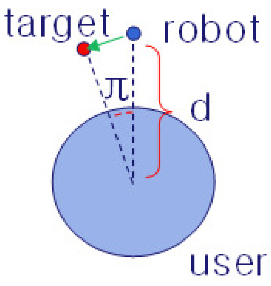


Fig. 3.5 In the case of the mobile robot moves directly to the target position.

Actually, most cases normally fall into this case. In this case, the mobile robot moves directly to the target position in a straight line without any special configurations.

3.4.3 When the Robot Moves with Temporary Target Configurations

The third case is when the angle π is bigger than π' because of the sudden movement of the user.

In this case, the mobile robot has to move along the curve which maintains the specific distance from the user for not only avoiding the collision but also for allowing the user to grab the tool tip of the haptic device constantly (figure 3.6).

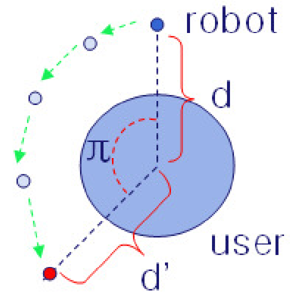


Fig. 3.6 In the case of the mobile robot cannot move directly to the target position.

To achieve these constraints, the algorithm sets temporary target positions between the current and target position and moves the mobile robot following the temporary target positions, so that the mobile robot follows the proper curve path. Figure 3.6 shows the temporary target positions used in the algorithm.

3.5 Mobile Robot Direction Setting

For motion planning, it is important to set not only the position but also the direction of the mobile robot when it moves.

As shown in figure 3.7, if the user moves in the "a" direction, the mobile robot moves following the "b" curve. At this moment, the mobile robot needs to be set facing the user to maximize the usability of the haptic device. For this constraint, the direction of the mobile robot, ϕ, in the world coordinate is calculated using the following formula each time.

$$\begin{cases} \phi = \arccos\left(\frac{\overrightarrow{V1} \cdot \overrightarrow{V2}}{|\overrightarrow{V1}||\overrightarrow{V2}|}\right) \\ sign = sign(\overrightarrow{V1} \times \overrightarrow{V2}) \end{cases} . \tag{3.1}$$

In this formula, $\overrightarrow{V1}$, also shown in figure 3.7, is the vector parallel to the z axis and $\overrightarrow{V2}$ is the vector from the user to the mobile robot. The angle ϕ between

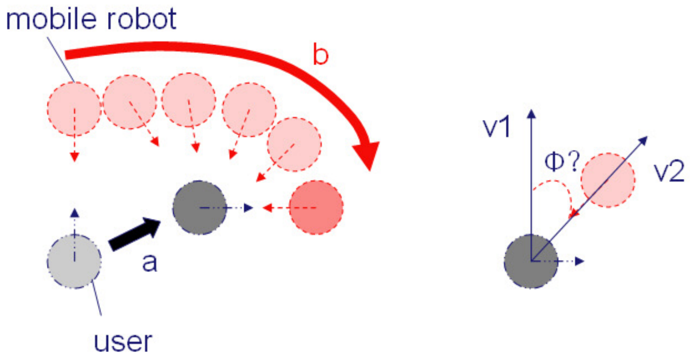


Fig. 3.7 Mobile robot direction setting.

two vectors can be calculated using the property of the inner product, and the direction of ϕ is calculated by the outer product of two vectors.

3.6 Evaluations

To evaluate the performance of the motion planning algorithm, we simulated the motion of the mobile robot and the user under limited maximum linear and angular velocity and represented results to graphs. To simplify the simulation process, we assumed that the mobile robot can always move with maximum velocity. If the user's hand is included in the sphere which is inscribed in the workspace of the haptic device, we assumed that the mobile robot follows the user's motion without problems.

3.6.1 For Varying Maximum Velocity

The blue region in figure 3.8 (a) represents the next position of the user where the mobile robot can catch up with the user in 1 second. The unit used in this graph is mm. Figure 3.8 (b) is the pictorial representation of Figure 3.8 (a). At the beginning, the user stands at the point (0, 0) facing the mobile robot and

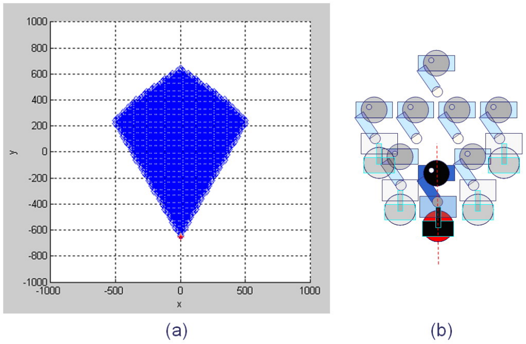
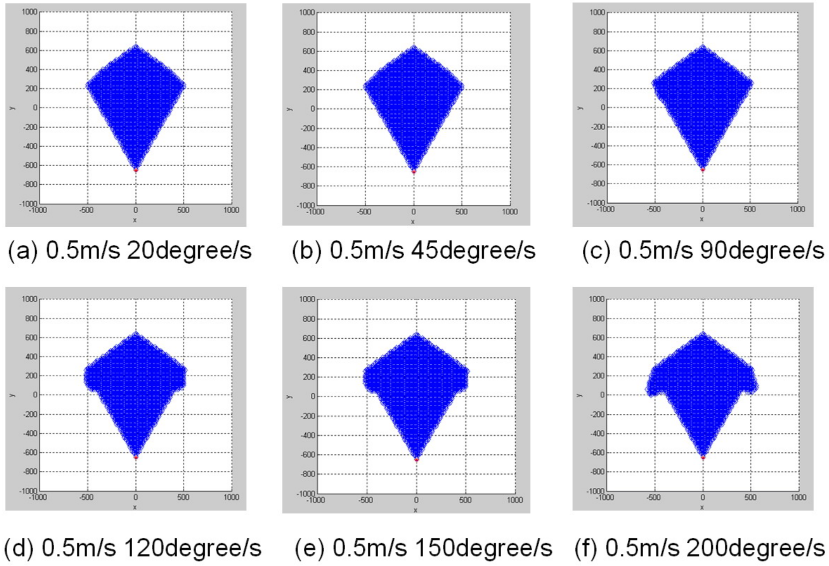


Fig. 3.8 Responses of the mobile robot depending on the user's motion.

the mobile robot is placed in the opposite direction from the user. The larger the region is, the better the performance of the algorithm .

3.6.2 For Varying Maximum Angular Velocity



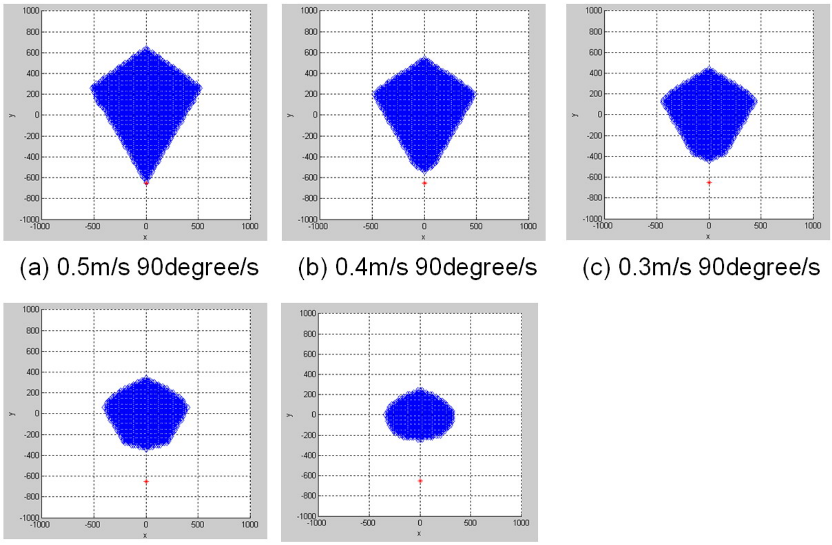
(a) 0.5m/s 20degree/s (b) 0.5m/s 45degree/s (c) 0.5m/s 90degree/s

(d) 0.5m/s 120degree/s (e) 0.5m/s 150degree/s (f) 0.5m/s 200degree/s

Fig. 3.9 Regions that the mobile robot can move in 1 second with different maximum angular velocities.

Figure 3.9 shows changes of regions according to the change in maximum angular velocity while setting the maximum linear velocity constant. The unit used in figure 3.9 and figure 3.10 is the same as the unit in figure 3.8. As shown in the results of figure 3.9, if there is no direction changes of the user, the effect of the maximum angular velocity of the mobile robot is small. Specifically, there is little change in the range of the maximum angular velocity, 45 $\sim$ 90°/s, decided considering the user's safety.

3.6.3 For Varying Maximum Linear Velocity



(a) 0.5m/s 90degree/s (b) 0.4m/s 90degree/s (c) 0.3m/s 90degree/s

(d) 0.2m/s 90degree/s (e) 0.1m/s 90degree/s

Fig. 3.10 Regions that the mobile robot can move in 1 second with different maximum linear velocities.

On the other hand, the effect of the maximum linear velocity change of the mobile robot is significant. As shown in figure 3.10, when the maximum linear velocity is 0.5 m/s, the mobile robot can follow most of user's motion. When the maximum linear velocity is 0.1 m/s, the mobile robot can catch up the user who moves less than 0.2 m. From these graphs, we can confirm that the slower the maximum linear velocity decreases, the smaller the region becomes where the user can move. Specifically, if the maximum linear velocity drops below 0.3 m/s, the regions get dramatically smaller.

3.7 Performance Evaluation using the Simulator

To verify that the motion planning algorithm moves the mobile robot in the path where the user can always interact with the haptic device without any collisions, we developed the simulator such as figure 3.11.

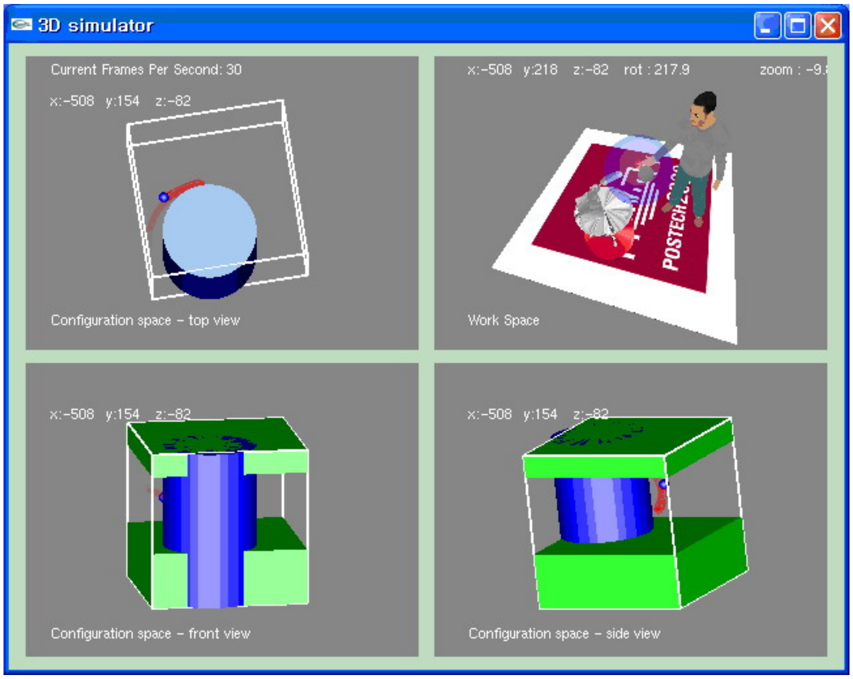

3D simulator
Current Frames Per Second: 30
x:-508 y:154 z:-82
Configuration space – top view
x:-508 y:218 z:-82 rot : 217.9
zoom : -9.
POSTECH
Work Space
x:-508 y:154 z:-82
Configuration space – front view
x:-508 y:154 z:-82
Configuration space – side view

Fig. 3.11 The simulator for the performance evaluation.

The simulator in figure 3.11 shows the view from the workspace and top front, side view of the configuration space in real-time, shown counterclockwise from the right top image. Therefore, we can comprehend changes in the configuration space due to the movement of the mobile robot and the user.

The real user's motion is captured by the tracker, and applied to the simulator. With the limitation of the linear velocity of 0.5 m/s, and the angular velocity of 90°/s, the simulator shows that in most cases the mobile robot can follow the user without collisions.

3.8 Adaptation to the Mobile Robot

After verifying the work of the motion planning algorithm by the simulator, we made some changes to the motion planning program when we applied it to the real mobile robot. At first, considering the stability of the current developed mobile robot, we set the maximum linear velocity to 0.2 m/s. After calculating the target position in the motion planning algorithm, the linear velocity is set in proportion as the distance between the target position and the current position of the mobile robot. If the calculated linear velocity is larger than the maximum linear velocity, the linear velocity is set to the maximum value. After deciding the linear velocity, the velocity is decomposed into two components, $\overrightarrow{Vx}, \overrightarrow{Vz}$, which are commanded to the mobile robot.

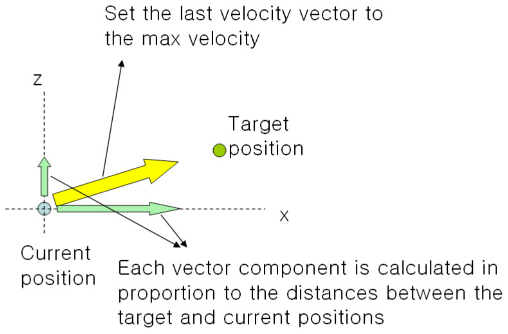

Set the last velocity vector to
the max velocity
z
Target
position
x
Current
position
Each vector component is calculated in
proportion to the distances between the
target and current positions

Fig. 3.12 Maximum linear velocity and decomposition of the velocity vector.

At this moment, each vector component is calculated in proportion to the distances between the target and current positions. The angular velocity is also set in proportion to the difference between the target and current angle. If the angular velocity is larger than the maximum angular velocity, the angular velocity is set to the maximum angular velocity. Current maximum angular velocity is 40 °/s.

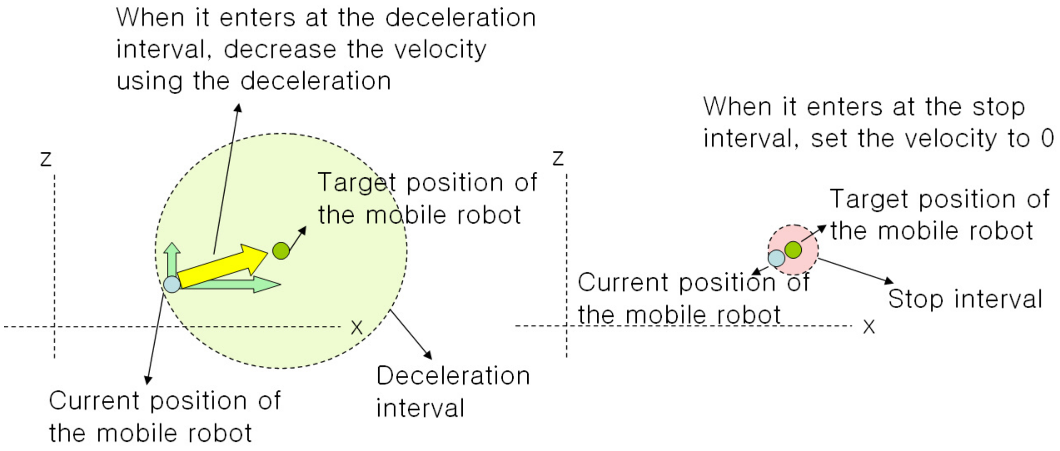

When it enters at the deceleration interval, decrease the velocity using the deceleration
z
Target position of the mobile robot
x
Current position of the mobile robot
Deceleration interval
When it enters at the stop interval, set the velocity to 0
z
Target position of the mobile robot
Current position of the mobile robot
Stop interval
x

Fig. 3.13 Deceleration and the stop intervals.

As shown in figure 3.13, in order to stop the mobile robot at the target position, we set the deceleration interval and reduced the velocity in proportion to the distance between the current and target position. In addition, because of the slipping of the wheels and the inaccuracy of the tracker, it is impossible to stop the mobile robot at the exact target position. Therefore, we set the stop interval as the threshold. If the distance between the target and current position is smaller than the stop interval, we assumed the mobile robot arrived at the target position and set the velocity to 0. Current stop interval is 50 mm.

Also, applying the motion planning algorithm at all times decreases the system stability because the mobile robot always moves to follow the user when the user moves HIP even a little. Therefore, in this system, we apply the motion planning algorithm only when the mobile robot necessarily needs to move, that is, when HIP is getting closer to the boundary of the haptic device's workspace.

As shown in figure 3.1, if HIP escapes from the usable workspace of the PHANToM device, we decided HIP is getting closer to the workspace boundary of the haptic device and applied the motion planning algorithm to move the mobile robot.

CHAPTER 4

Kinematics

4.1 Definition of Coordinate Frames

For a proper force rendering through the mobile haptic display, the position of HIP in the world coordinate is required. We defined the coordinate system such as Figure 4.2 which follows the real physical structure shown in Figure 4.1.

Firstly, the world coordinate is a right handed system like OpenGL and it is located at the bottom. The IS900 coordinate is the coordinate of the IS900 tracker system and located at the ceiling. The sensor coordinate is the coordinate of the sensor which is installed on the mobile robot, and the robot coordinate is the coordinate of the mobile robot itself. The PHANToM coordinate is the inner coordinate used in the PHANToM Premium 1.5 device and the tool coordinate is the coordinate of the tooltip grasped by the user.

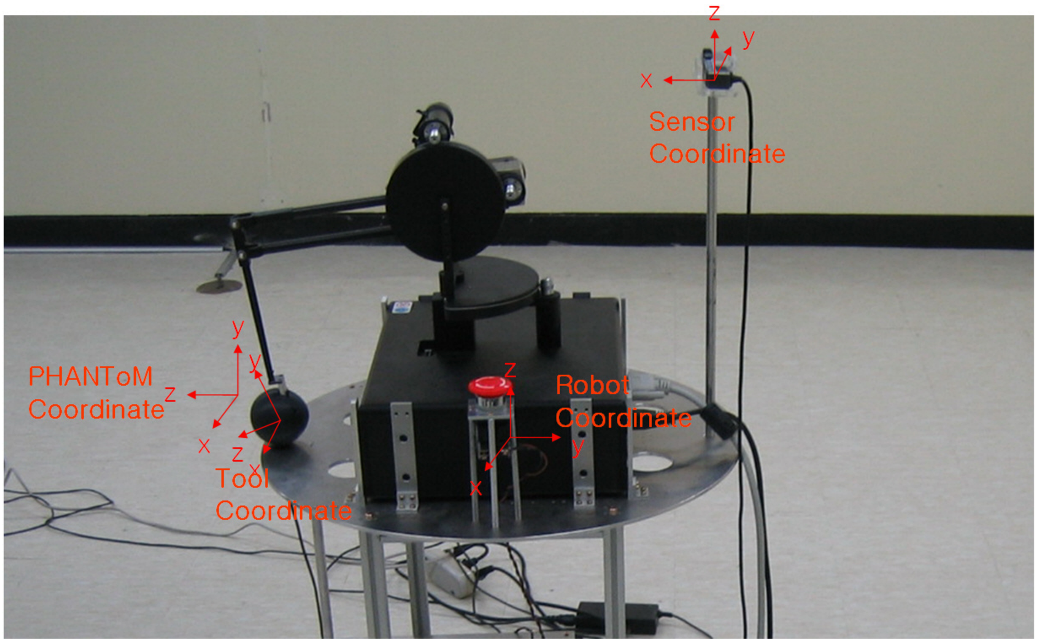


Fig. 4.1 Structure of the mobile haptic display.

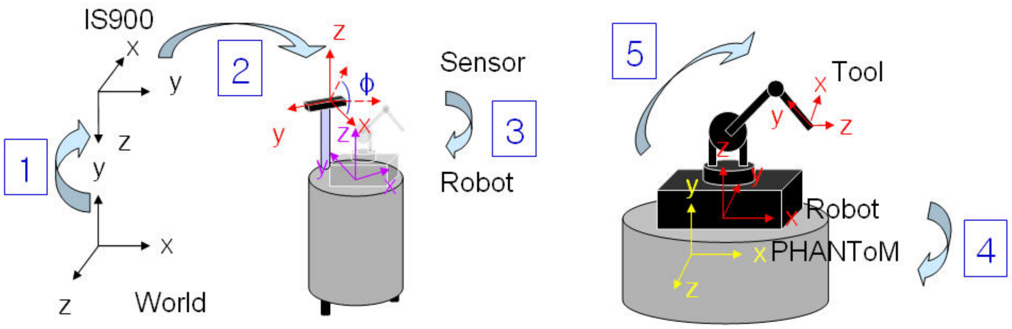


Fig. 4.2 Definition of coordinate frames.

4.2 Definition of Transformations

Transformation defines the relationship between each coordinates and transformations used in the mobile haptic display as defined in Figure 4.3.

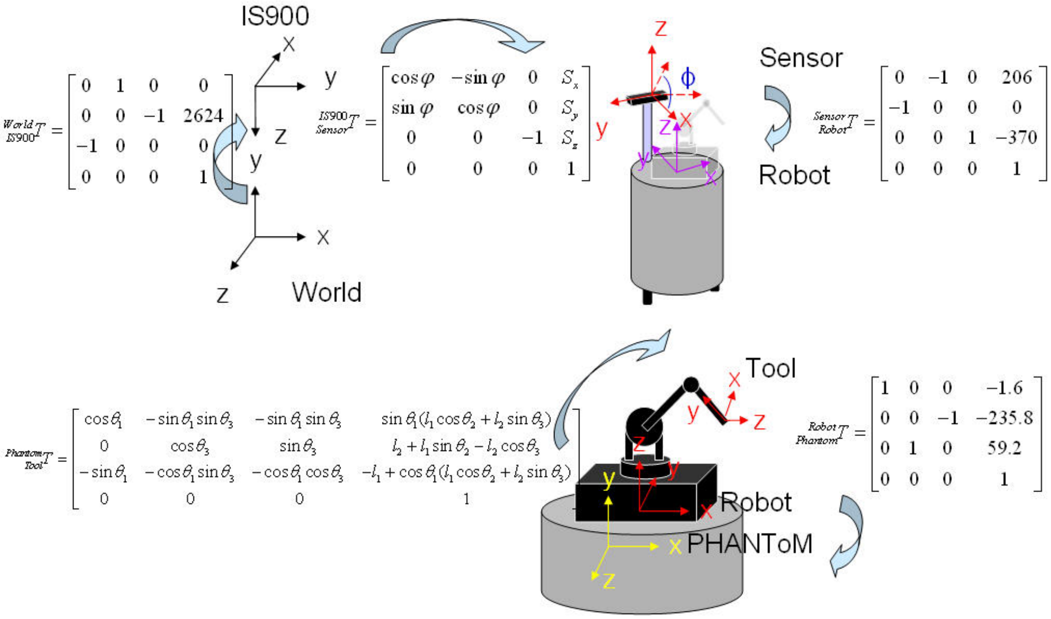

IS900
x
y
z
World
$^{World}_{IS900}T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2624 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
$^{IS900}_{Sensor}T = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & S_x \\ \sin\varphi & \cos\varphi & 0 & S_y \\ 0 & 0 & -1 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$
ϕ
Sensor
Robot
$^{Sensor}_{Robot}T = \begin{bmatrix} 0 & -1 & 0 & 206 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -370 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Tool
$^{Phantom}_{Tool}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1\sin\theta_3 & -\sin\theta_1\sin\theta_3 & \sin\theta_1(l_1\cos\theta_2 + l_2\sin\theta_3) \\ 0 & \cos\theta_3 & \sin\theta_3 & l_2 + l_1\sin\theta_2 - l_2\cos\theta_3 \\ -\sin\theta_1 & -\cos\theta_1\sin\theta_3 & -\cos\theta_1\cos\theta_3 & -l_1 + \cos\theta_1(l_1\cos\theta_2 + l_2\sin\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Robot
PHANToM
$^{Robot}_{Phantom}T = \begin{bmatrix} 1 & 0 & 0 & -1.6 \\ 0 & 0 & -1 & -235.8 \\ 0 & 1 & 0 & 59.2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Fig. 4.3 Definition of transformations.

Transformations are represented by 4x4 matrixes using homogeneous coordinate.

$$^{World}_{IS900}T = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2624 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.1}$$

$^{World}_{IS900}T$ represents the relation between the world and IS900 coordinate. It adjusts the axes and shows that the IS900 coordinate is 2624 mm higher than the world coordinate.

$$^{IS900}_{Sensor}T = \begin{pmatrix} \cos\phi & -\sin\phi & 0 & S_x \\ \sin\phi & \cos\phi & 0 & S_y \\ 0 & 0 & -1 & S_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.2}$$

$^{IS900}_{Sensor}T$ rotates the coordinate by angle ϕ about the z axis and translates it to

Sx, Sy, and Sz which are the values of the sensor position.

$$ {}^{Sensor}_{Robot}T = \begin{pmatrix} 0 & -1 & 0 & 206 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -370 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.3} $$

${}^{Sensor}_{Robot}T$ adjusts the axes between the sensor coordinate and the robot coordinate and represents the origin of the robot coordinate moved by 206 mm in x axis and -370 mm in z axis from the origin of the sensor coordinate. At this moment, the values 206 mm and 370 mm are directly measured values using a ruler.

$$ {}^{Robot}_{Phantom}T = \begin{pmatrix} 1 & 0 & 0 & -1.6 \\ 0 & 0 & -1 & -235.8 \\ 0 & 1 & 0 & 59.2 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.4} $$

${}^{Robot}_{Phantom}T$ adjusts the axes from the robot coordinate to the PAHNToM coordinate and shows that the origin of the PHANToM coordinate is moved by -1.6 mm in x axis, -234.8 mm in y axis, and 59.2 mm in z axis from the origin of the robot coordinate. The origin of the PHANToM coordinate is the point of HIP when each joint angles of the PHANToM is set to 0; therefore, it is impossible to measure the origin of the PHANToM coordinate with a ruler. Consequently, we set the offset values by locating HIP into the specific position in the robot coordinate and comparing it with the position in the the PHANToM coordinate.

$$ {}^{Phantom}_{Tool}T = \begin{pmatrix} \cos\theta_1 & -\sin\theta_1\sin\theta_3 & -\sin\theta_1\sin\theta_3 & \sin\theta_1(l_1\cos\theta_2 + l_2\sin\theta_3) \\ 0 & \cos\theta_3 & \sin\theta_3 & l_2 + l_1\sin\theta_2 - l_2\cos\theta_3 \\ \sin\theta_1 & -\cos\theta_1\sin\theta_3 & -\cos\theta_1\cos\theta_3 & -l_1 + \cos\theta_1(l_1\cos\theta_2 \\ & & & +l_2\sin\theta_3) \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.5} $$

${}^{Phantom}_{Tool}T$ represents the transformation from the PHNToM coordinate to the tool coordinate. l_1 and l_2 are link lengths of the PHANToM premium 1.5.

$\theta_1, \theta_2, and\theta_3$ are the joint angle values of the PHANToM. According to SensAble, both l_1 and l_1 are 209.55 mm. These values are obtained by an email request from the SensAble and are also used in OpenHaptics for calculating the HIP position. Detailed derivation of ${}^{Phantom}_{Tool}T$ is introduced in [17], [18].

4.3 Position Estimation

After reading the values from Sx, Sy, and Sz and the direction ϕ of the sensor that is equipped on the mobile robot, the final transformation from the world coordinate to the tool coordinate is calculated with following formulas.

$$
\begin{aligned}
{}^{World}_{Tool}T &= {}^{World}_{IS900}T \cdot {}^{IS900}_{Sensor}T \cdot {}^{Sensor}_{Robot}T \cdot {}^{Phantom}_{Tool}T = \begin{pmatrix} R_{11} & R_{12} & R_{13} & X \\ R_{21} & R_{22} & R_{23} & Y \\ R_{31} & R_{32} & R_{33} & Z \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
R_{11} &= -\cos(\phi)\cos(\theta_1) - \sin(\phi)\sin(\theta_1) \\
R_{12} &= \cos(\phi)\sin(\theta_1)\sin(\theta_3) - \sin(\phi)\cos(\theta_1)\sin(\theta_3) \\
R_{13} &= -\cos(\phi)\sin(\theta_1)\cos(\theta_3) - \sin(\phi)\cos(\theta_1)\cos(\theta_3) \\
R_{21} &= 0 \\
R_{22} &= \cos(\theta_3) \\
R_{23} &= \sin(\theta_3) \\
R_{31} &= -\sin(\phi)\cos(\theta_1) + \cos(\phi)\sin(\theta_1) \\
R_{32} &= \sin(\phi)\sin(\theta_1)\sin(\theta_3) + \cos(\phi)\cos(\theta_1)\sin(\theta_3) \\
R_{33} &= -\sin(\phi)\sin(\theta_1)\cos(\theta_3) + \cos(\phi)\cos(\theta_1)\cos(\theta_3) \\
X &= -\cos(\phi)\sin(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3)) \\
&\quad + \sin(\phi)(-4191/20 + \cos(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3))) \\
&\quad + 8/5\cos(\phi) + 2209/5\sin(\phi) + Sy \\
Y &= 10091/4 + 4191/20\sin(\theta_2) - 4191/20\cos(\theta_3) - Sz \\
Z &= -\sin(\phi)\sin(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3)) \\
&\quad - \cos(\phi)(-4191/20 + \cos(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3))) \\
&\quad + 8/5\sin(\phi) - 2209/5\cos(\phi) - Sx
\end{aligned}
$$

4.4 Kinematics Calibration

4.4.1 Motivation

To verify the formulas derived in Chapter 4.3, we installed the tracker in the HIP position shown in Figure 4.4 and compared the position of the tracker equipped in the HIP position with the position calculated through the transformation.

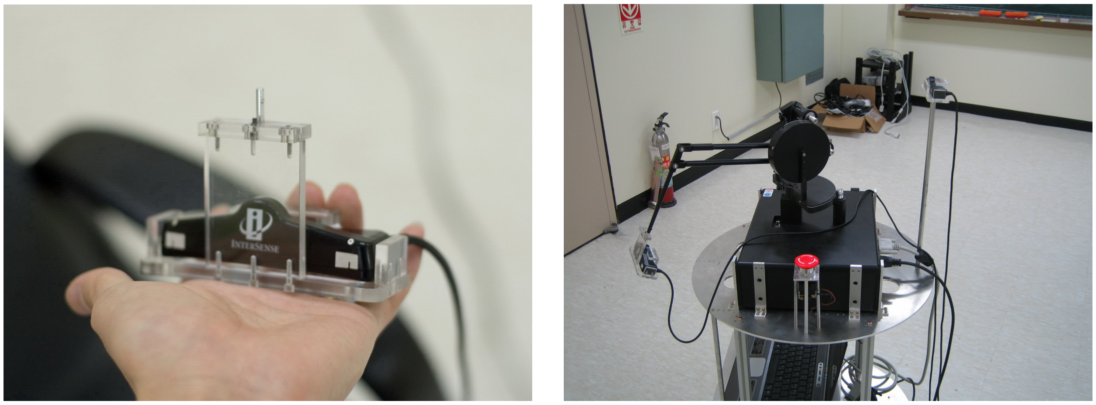


Fig. 4.4 Tracker installed in the HIP position.

Moving the mobile robot randomly, we collected the error data of each axis at 1-second intervals. Figure 4.5 shows the differences between the measured HIP position and calculated HIP position of three axes.

There are 0∼40 mm errors between the real HIP position and the position from transformation. According to the table 2.4, the position resolution of PHANToM Premium 1.5 and IS900 tracker is 0.03 mm and 2∼3 mm, respectively. Therefore, it can be seen that the HIP position estimation error occurs from other reasons. Assuming that the real position resolution of the PHANToM device and the IS900 tracker system is as accurate as the specification referred in the manuals, the errors during the HIP position estimation are expected to occur from the transformation process. During the transformation process, a factor which can cause the errors except the values from the tracker and the haptic device is actual measured values such as the offset values in the sensor to robot transformation and the robot to PHANToM transformation. Because of the structure of the mobile robot and the haptic device, we cannot measure the offset values accurately. For more accurate transformation, we

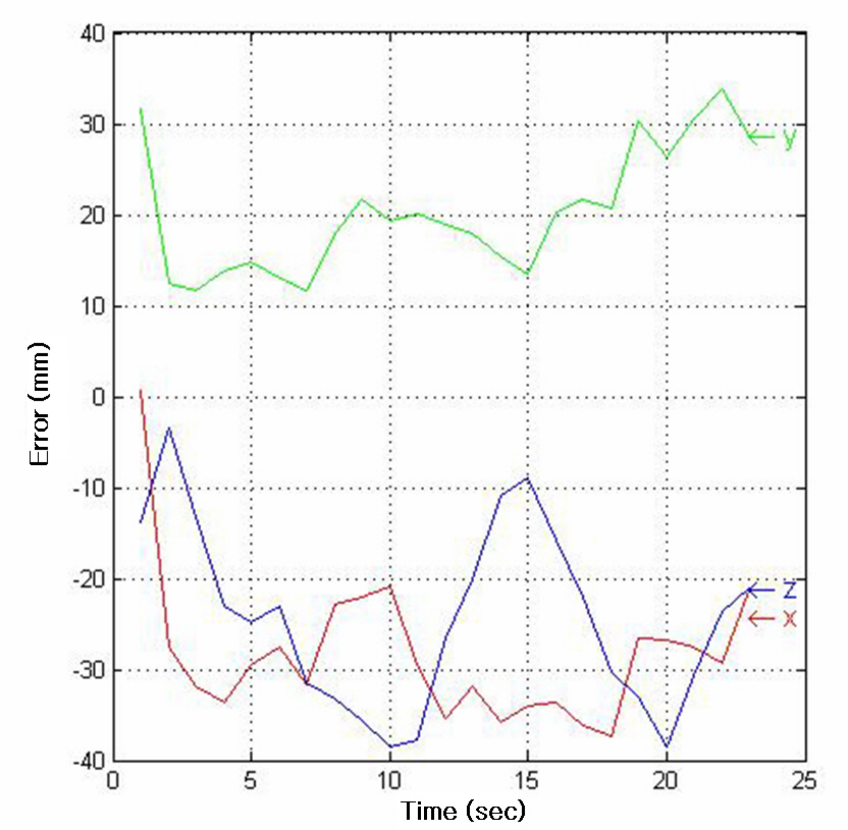
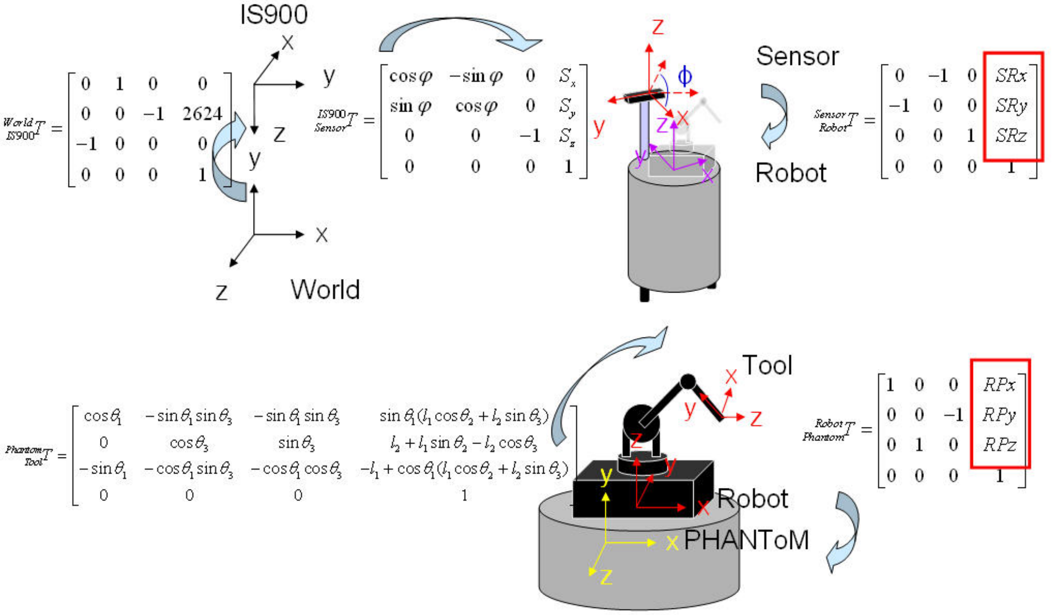


Fig. 4.5 Differences between real and calculated HIP positions.

need to compensate these measured values. As a result, we can calculate the accurate HIP position.

4.4.2 Data Collection

To compensate for the measured values, these values are changed to variables from constants in the transformation shown in Figure 4.6. The offset values from sensor coordinate to robot coordinate are defined as SRx, SRy, and SRz. And the offset values from the robot coordinate to PHANToM coordinate are defined as RPx, RPy, and RPz.



$$ {}^{World}_{IS900}T = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 2624 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{IS900}_{Sensor}T = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 & S_x \\ \sin\varphi & \cos\varphi & 0 & S_y \\ 0 & 0 & -1 & S_z \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{Sensor}_{Robot}T = \begin{bmatrix} 0 & -1 & 0 & \boxed{SRx} \\ -1 & 0 & 0 & \boxed{SRy} \\ 0 & 0 & 1 & \boxed{SRz} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{Phantom}_{Tool}T = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1\sin\theta_3 & -\sin\theta_1\sin\theta_3 & \sin\theta_1(l_1\cos\theta_2 + l_2\sin\theta_3) \\ 0 & \cos\theta_3 & \sin\theta_3 & l_2 + l_1\sin\theta_2 - l_2\cos\theta_3 \\ -\sin\theta_1 & -\cos\theta_1\sin\theta_3 & -\cos\theta_1\cos\theta_3 & -l_1 + \cos\theta_1(l_1\cos\theta_2 + l_2\sin\theta_3) \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{Robot}_{Phantom}T = \begin{bmatrix} 1 & 0 & 0 & \boxed{RPx} \\ 0 & 0 & -1 & \boxed{RPy} \\ 0 & 1 & 0 & \boxed{RPz} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

Fig. 4.6 Replacement of measure values from constants to variables.

After the replacement, the final formulas of HIP position are calculated as follows:

$$\begin{aligned}
X &= -\cos(\phi)\sin(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3)) \\
&\quad +\sin(\phi)(-4191/20+\cos(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3))) \\
&\quad -\cos(\phi)RPx-\sin(\phi)RPy+\sin(\phi)SRx+\cos(\phi)SRy+Sy \\
Y &= 56671/20+4191/20\sin(\theta_2)-4191/20\cos(\theta_3)+RPz+SRz-Sz \\
Z &= -\sin(\phi)\sin(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3)) \\
&\quad -\cos(\phi)(-4191/20+\cos(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3))) \\
&\quad -\sin(\phi)RPx+\cos(\phi)RPy-\cos(\phi)SRx+\sin(\phi)SRy-Sx
\end{aligned}$$

The formulas can be simplified as follows:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \end{pmatrix} + \begin{pmatrix} sin\phi & cos\phi & 0 & -cos\phi & -sin\phi & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ -cos\phi & sin\phi & 0 & -sin\phi & cos\phi & 0 \end{pmatrix} \begin{pmatrix} SRx \\ SRy \\ SRz \\ RPx \\ RPy \\ RPz \end{pmatrix} \quad \textbf{(4.6)}$$

$$\begin{aligned}
C_1 &= -\cos(\phi)\sin(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3)) \\
&\quad +\sin(\phi)(-4191/20+\cos(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3)))+Sy \\
C_2 &= 56671/20+4191/20\sin(\theta_2)-4191/20\cos(\theta_3)-Sz \\
C_3 &= -\sin(\phi)\sin(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3)) \\
&\quad -\cos(\phi)(-4191/20+\cos(\theta_1)(4191/20\cos(\theta_2)+4191/20\sin(\theta_3)))-Sx
\end{aligned}$$

In these formulas, $X, Y, andZ$ represent the real position of the HIP, $Sx, Sy, andSz$ the position of the sensor equipped on the mobile robot, ϕ the direction of the mobile robot, and $\theta_1, \theta_2, and\theta_3$ the joint angles of the haptic device.

For collecting the errors between the real HIP position and the calculated HIP position, we gathered 100 data samples. To minimize the effect of unwanted factors, the position of the mobile robot is randomly chosen between -1000 mm ∼ 1000 mm shown in Figure 4.7, and the direction of the mobile robot and the joint angles of the haptic device are randomly selected between 0 °∼ 360 °.

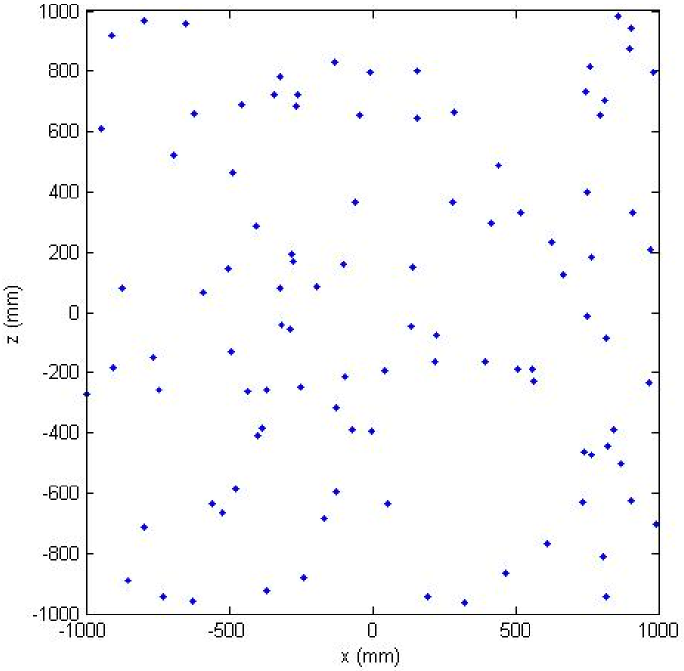


Fig. 4.7 Randomly chosen position data set of the mobile robot.

The errors are the differences between the HIP position values calculated by the transformation and measured by the tracker. Using the least square method which minimizes the squares of errors, we calculated SRx, SRy, SRz, RPx, RPy, and RPZ with 100 data sets. The initial values are given as directly measured values.

$$\begin{pmatrix} SRx \\ SRy \\ SRz \\ RPx \\ RPy \\ RPz \end{pmatrix} = \begin{pmatrix} 206 \\ 0 \\ -370 \\ -1.6 \\ -235.8 \\ 59.2 \end{pmatrix} \Longrightarrow \begin{pmatrix} 203.1 \\ 1.8 \\ -387.6 \\ -2.1 \\ -232.8 \\ 41.6 \end{pmatrix}$$

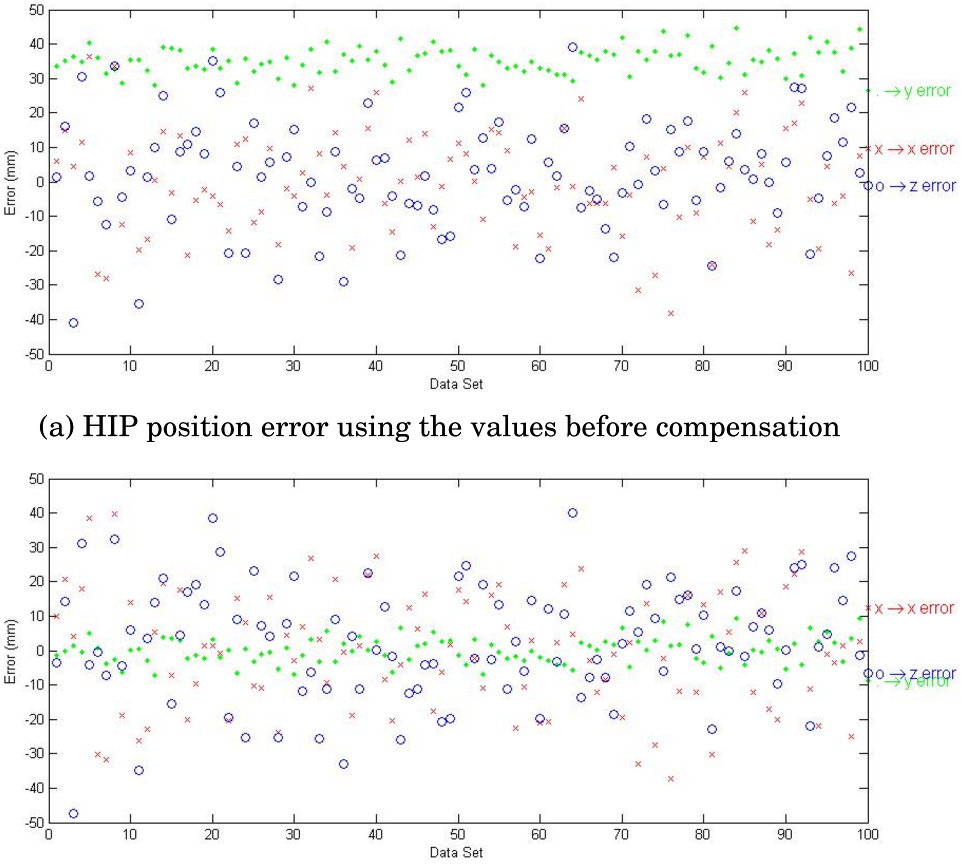
4.4.3 Error Comparison

To find out how much the errors decreased after compensating, we compared the errors of HIP position before compensation with errors after compensation.

Root mean square error decreases from 572.7 to 200.6. Also, the graph in Figure RefError shows the error of y axis decreases dramatically. However, the errors of x and z axes do not decrease as much.

Decreased error in y axis means the Kinematics calibration is effective. The error in y axis with less than 10 mm is caused by the uneven floor and the error of the IS900 tracker system.

Other remaining errors in x and z axes are caused by not the failure of the Kinematics calibration but the unconsidered terms during the calibration. We assumed that the direction of the tracker on the mobile robot and the direction of the mobile robot are the same. If this hypothesis is wrong, even small direction difference causes lots of errors to be accumulated in HIP position in x and z axes. Therefore, if we include a new term, ψ, which is the direction difference between the mobile robot and the tracker, better results can be obtained in the calibration.



(a) HIP position error using the values before compensation

(b) HIP position error using the values after compensation

Fig. 4.8 Errors before and after Kinematics calibration.

CHAPTER 5

Haptic Rendering

5.1 Definition of Jacobian

Haptic rendering refers to the computation process required for delivering realistic force to the user between virtual environments and physical devices through haptic devices [19]. In our system, the force is calculated by the virtual proxy algorithm [20] which is one of the most prevailing haptic rendering algorithms. In the virtual proxy algorithm, the virtual proxy, which follows the real HIP, is defined. After the collision between HIP and the virtual object, virtual proxy is located at the face which is nearest to HIP and the force is calculated in proportion to the distance between the virtual proxy and HIP.

From the sensor position equipped on the mobile robot, multiplying this position by the transformation gives the HIP position. Using the HIP position, the virtual proxy algorithm calculates the force felt by the user. To give torque commands to the haptic device after calculating the force, the Jacobian matrix has to be calculated.

The Jacobian matrix is the matrix of all first-order partial derivatives of a vector-valued function. In Robotics, it is usually used to find the differential motion or the velocity of a tooltip by changes of joint angles [21].

Using the transformation discussed in Chapter 4.4.3, the HIP position in the world coordinate is represented as follows:

$$\begin{aligned}
X &= -\cos(\phi)\sin(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3)) \\
&\quad + \sin(\phi)(-4191/20 + \cos(\theta_1)(4191/20\cos(\theta_2) \\
&\quad +4191/20\sin(\theta_3))) + 21/10\cos(\phi) + 1164/5\sin(\phi) + Rx \\
Y &= 5023/20 + 4191/20\sin(\theta_2) - 4191/20\cos(\theta_3) + Ry \\
Z &= -\sin(\phi)\sin(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3)) \\
&\quad - \cos(\phi)(-4191/20 + \cos(\theta_1)(4191/20\cos(\theta_2) + 4191/20\sin(\theta_3))) \\
&\quad +21/10\sin(\phi) - 1164/5\cos(\phi) + Rz
\end{aligned}$$

Rx, Ry, and Rz are the positions and ϕ is the direction of the mobile robot in the world coordinate. $\theta_1, \theta_2, and \theta_3$ are the joint angles of the haptic device in the world coordinate. The HIP position in the world coordinate is formulated by Rx, Ry, Rz, $\phi, \theta_1, \theta_2, and \theta_3$ and is represented as follows:

$$\begin{aligned}
X &= f_1(Rx, Ry, Rz, \phi, \theta_1, \theta_2, \theta_3) \\
Y &= f_2(Rx, Ry, Rz, \phi, \theta_1, \theta_2, \theta_3) \\
Z &= f_3(Rx, Ry, Rz, \phi, \theta_1, \theta_2, \theta_3)
\end{aligned}$$

Therefore, the Jacobian is defined as follows:

$$J(Rx, Ry, Rz, \phi, \theta_1, \theta_2, \theta_3) = \begin{pmatrix} \frac{\partial X}{\partial Rx} & \frac{\partial X}{\partial Ry} & \frac{\partial X}{\partial Rz} & \frac{\partial X}{\partial \phi} & \frac{\partial X}{\partial \theta_1} & \frac{\partial X}{\partial \theta_2} & \frac{\partial X}{\partial \theta_3} \\ \frac{\partial Y}{\partial Rx} & \frac{\partial Y}{\partial Ry} & \frac{\partial Y}{\partial Rz} & \frac{\partial Y}{\partial \phi} & \frac{\partial Y}{\partial \theta_1} & \frac{\partial Y}{\partial \theta_2} & \frac{\partial Y}{\partial \theta_3} \\ \frac{\partial Z}{\partial Rx} & \frac{\partial Z}{\partial Ry} & \frac{\partial Z}{\partial Rz} & \frac{\partial Z}{\partial \phi} & \frac{\partial Z}{\partial \theta_1} & \frac{\partial Z}{\partial \theta_2} & \frac{\partial Z}{\partial \theta_3} \end{pmatrix} \tag{5.1}$$

However, since the mobile robot does not move in up and down in the current system, we can remove the Ry term. Without Ry, the Jacobian is calculated as follows:

$$J(Rx, Rz, \phi, \theta_1, \theta_2, \theta_3) = \begin{pmatrix} 1 & 0 & J_{13} & J_{14} & J_{15} & J_{16} \\ 0 & 0 & 0 & 0 & J_{25} & J_{26} \\ 0 & 1 & J_{33} & J_{34} & J_{35} & J_{36} \end{pmatrix} \tag{5.2}$$

$J_{13} = 4191/40\cos(-\theta_2 - \phi + \theta_1) + 4191/40\cos(\theta_2 - \phi + \theta_1) + 4191/40\sin(\theta_3 - \phi + \theta_1) - 4191/40\sin(-\theta_3 - \phi + \theta_1) + 93/4\cos(\phi) - 21/10\sin(\phi)$

$J_{14} = -4191/40\cos(-\theta_2 - \phi + \theta_1) - 4191/40\cos(\theta_2 - \phi + \theta_1) - 4191/40\sin(\theta_3 - \phi + \theta_1) + 4191/40\sin(-\theta_3 - \phi + \theta_1)$

$J_{15} = 4191/40\cos(-\theta_2 - \phi + \theta_1) - 4191/40\cos(\theta_2 - \phi + \theta_1)$

$J_{16} = -4191/40\sin(\theta_3 - \phi + \theta_1) - 4191/40\sin(-\theta_3 - \phi + \theta_1)$

$J_{25} = 4191/20\cos(\theta_2)$

$J_{26} = 4191/20\sin(\theta_3)$

$J_{33} = -4191/40\sin(\theta_2 - \phi + \theta_1) - 4191/40\sin(-\theta_2 - \phi + \theta_1) - 4191/40\cos(-\theta_3 - \phi + \theta_1) + 4191/40\cos(\theta_3 - \phi + \theta_1) + 93/4\sin(\phi) + 21/10\cos(\phi)$

$J_{34} = 4191/40\sin(\theta_2 - \phi + \theta_1) + 4191/40\sin(-\theta_2 - \phi + \theta_1) + 4191/40\cos(-\theta_3 - \phi + \theta_1) - 4191/40\cos(\theta_3 - \phi + \theta_1)$

$J_{35} = 4191/40\sin(\theta_2 - \phi + \theta_1) - 4191/40\sin(-\theta_2 - \phi + \theta_1)$

$J_{36} = -4191/40\cos(-\theta_3 - \phi + \theta_1) - 4191/40\cos(\theta_3 - \phi + \theta_1)$

5.2 Effects of Mobile Robot Dynamics on Rendering Force

To analyze the influence of the force generated by the haptic device caused by the movement of the mobile robot, we defined the $[F_T] = [f_x, f_y, f_z]^T$ as the force applied to HIP, $[F_R] = [f_{Rx}, f_{Rz}, m_y]^T$ as the force applied to the mobile robot, and $[T_P] = [T_{\theta_1}, T_{\theta_2}, T_{\theta_3}]$ as the torque applied to the haptic device. In addition, the displacement of HIP moved by F_T is defined as $[D_T] = [d_x, d_y, d_z]^T$, the displacement of the mobile robot moved by the $[F_R]$ is defined as $[D_R] = [d_{Rx}, d_{Rz}, \delta_y]^T$, and the displacement of PHANToM's joint angles rotated by $[T_P]$ is defined as $[D_P] = [d_{\theta_1}, d_{\theta_2}, d_{\theta_3}]^T$.

According to the definition of the Jacobian, the following formula is established.

$$D_T = J \begin{bmatrix} D_R \\ D_P \end{bmatrix} \tag{5.3}$$

In the mobile haptic display system, the motion of the mobile robot is determined by the position of the user, the mobile robot and HIP. We call such situations where the motion of an object is determined by outside conditions holonomic constraint. If these conditions are given by exact position functions, the 3 by 6 Jacobian matrix can be divided into two 3 by 3 matrices J_R and J_P [22].

$$D_T = \begin{bmatrix} J_R & J_P \end{bmatrix} \begin{bmatrix} D_R \\ D_P \end{bmatrix} = J_R D_R + J_P D_P \tag{5.4}$$

If the mobile robot is tightly position-controlled and significantly heavier than a haptic device, the effect of the mobile robot motion can be ignored as follows: [22]

$$
\begin{aligned}
&\dot{D_T} = J_P\dot{D_P} + J_R\dot{D_R}\\
&\Longrightarrow [T_P] = [J_P]^T[F_T]
\end{aligned}
$$

In other words, even when the mobile robot moves, the force caused by the mobile robot can be ignored under precise position control and slow speed. Therefore, to render the force ofF_T to HIP, we commanded the torque calculated by $[J_P]^T[F_T]$ to the haptic device.

CHAPTER 6

Force Analysis

6.1 Motivation

According to Chapter 5, if the mobile robot is perfectly position controlled, the effect on the force generated by the haptic device of the mobile robot can be ignored. Besides this constraint, there are three constraints for the mobile robot to be transparent to the user [8].

- Position control of the mobile robot
- Torque control of the haptic device
- Damping between the haptic device and the mobile robot = 0

In the current system, the mobile robot is position controlled and the haptic device is torque controlled in effect. However, the mobile robot is not perfectly position controlled due to its performance limitations. In addition, we assumed the damping between the haptic device and the mobile robot to be small. However, we cannot assure that it is small enough to be ignored. Therefore, we analyzed how the motion of the mobile robot affects the force generated by the haptic device using the force sensor.

6.2 Experiment Design

The force sensor used in this experiment is Nano17 manufactured by ATI. It can measure six degrees of freedom of a force and torque. Detailed specification of the sensor is as follows.

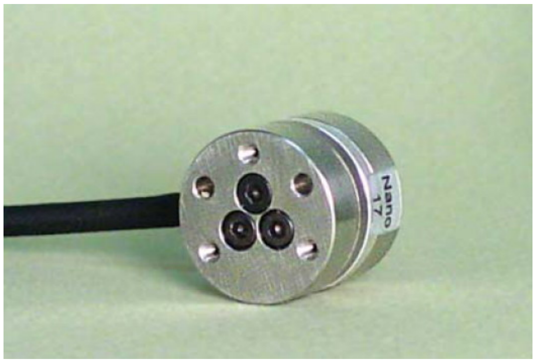


Fig. 6.1 Nano17 F/T transducer, ATI.

Table 6.1 Specification of the Nano17 sensor.

Axes	Sensing Ranges	Resolution
Fx, Fy (N)	± 12	1/1280
Fz (N)	± 17	1/1280
Tx, Ty (Nmm)	± 120	1/256
Tz (Nmm)	± 120	1/256

The force sensor is installed at HIP position with the case shown in Figure 6.2 such that the user can grab it.

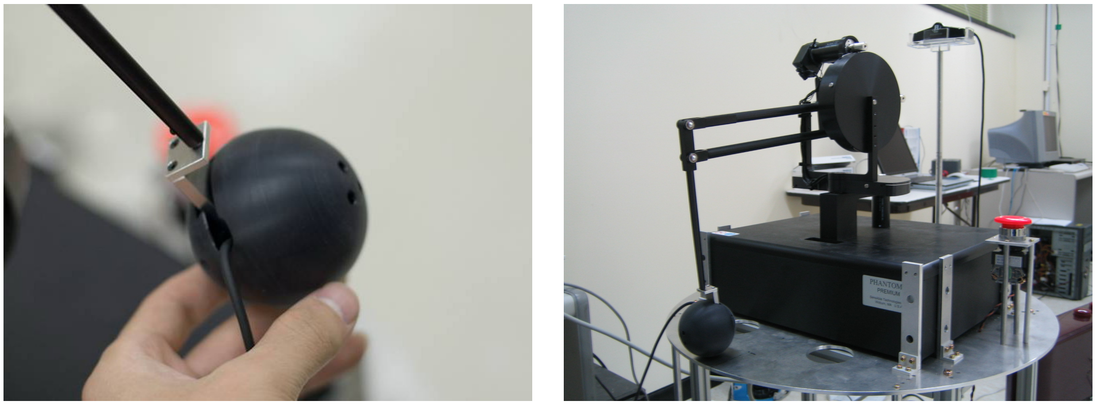


Fig. 6.2 Installation of the force sensor.

Force data is collected by the NI-DAQ of National Instruments at 10Hz. The force data is generated in the local coordinate of the force sensor; therefore, the data is transformed to the PHANToM coordinate.

6.3 Experimental Results

Two experiments have been conducted. In the first experiment, the force data of the force sensor was recorded when the force was not rendered by the haptic device and the user or the mobile robot moved. In the second experiment, the force data was recorded when the force was rendered .

6.3.1 When the Haptic Device does not render Force

The reason why we want to see the force data when the force is not rendered by the haptic device is to see how much force is delivered to HIP due to movements of the user or the mobile robot.

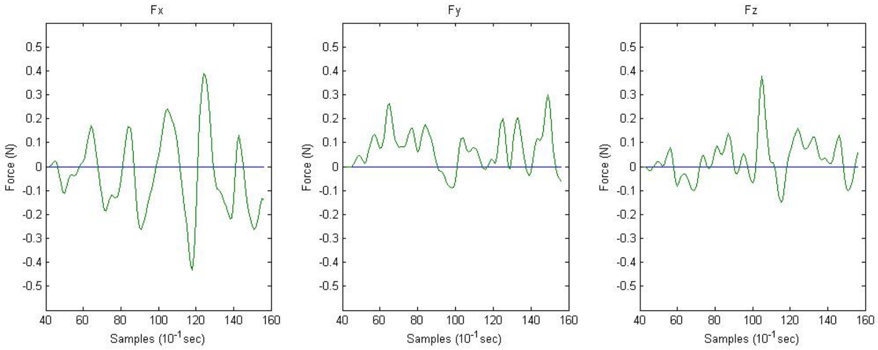


Fig. 6.3 Free motion of the user when the force is not rendered by the haptic device.

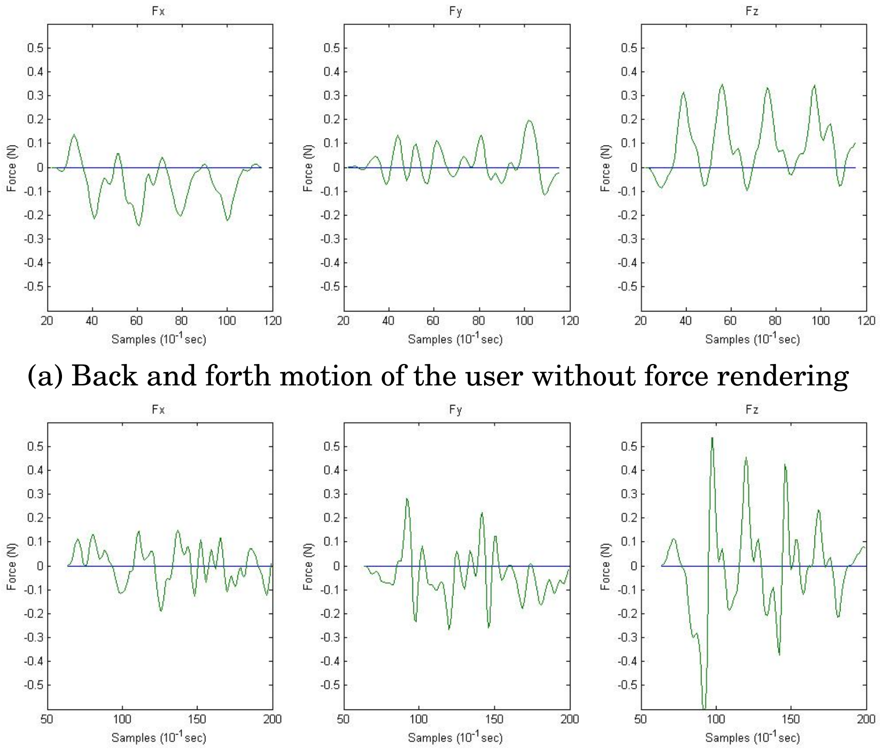
Free Motion of the User

When the user grabs HIP and moves it at a normal speed, the recorded force is 0 ∼ 0.4 N without any pattern in all axes as shown in Figure 6.3. It is expected that the recorded force is generated by acceleration or deceleration of the user's motion.

Comparisons of Motion of the User and the Mobile Robot

Next, force data is recorded when the user or the mobile robot moves back and forth along a specific axis.

When the user and the mobile robot moves along the z axis, the recorded force is less than 0.5 N as shown in Figure 6.4. Unlike the graph of the free motion of the user, there is a pattern along the z axis. It would be natural to deduce from this pattern that the recorded force was generated by the acceleration and deceleration of the motion of the user and the mobile robot. Figure 6.5 shows that most of the recorded force is on the x axis when the user and the mobile robot moved along the x axis. The pattern shown in the x axis means that the recorded force is generated by the motion of the user and the mobile robot. As a result, the force felt by the user when the user grabs HIP and moves it is



(a) Back and forth motion of the user without force rendering

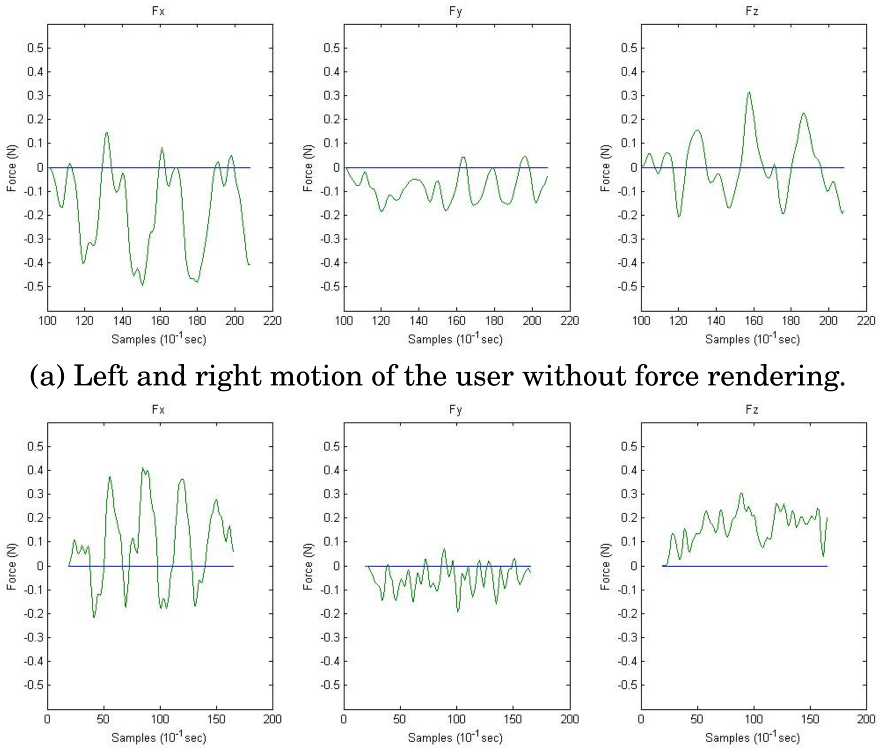
(b) Back and forth motion of the mobile robot without force rendering

Fig. 6.4 Back and forth motion of the user and the mobile robot without force rendering.

similar to the force felt by the user because of the motion of the mobile robot.

6.3.2 When the Haptic Device Renders Force

Graphs in Figure 6.6 is based on the force data when the haptic device is commanded to generate 2 N. When the user does not move the user's hand, relatively uniform force is recorded. However, when the user moves the user's hand back and forth, there is an oscillation pattern in the same axis as the user's motion. Similarly, when the mobile robot moves back and forth, there is an oscillation pattern in that direction. That is to say, when the force is rendered,



(a) Left and right motion of the user without force rendering.

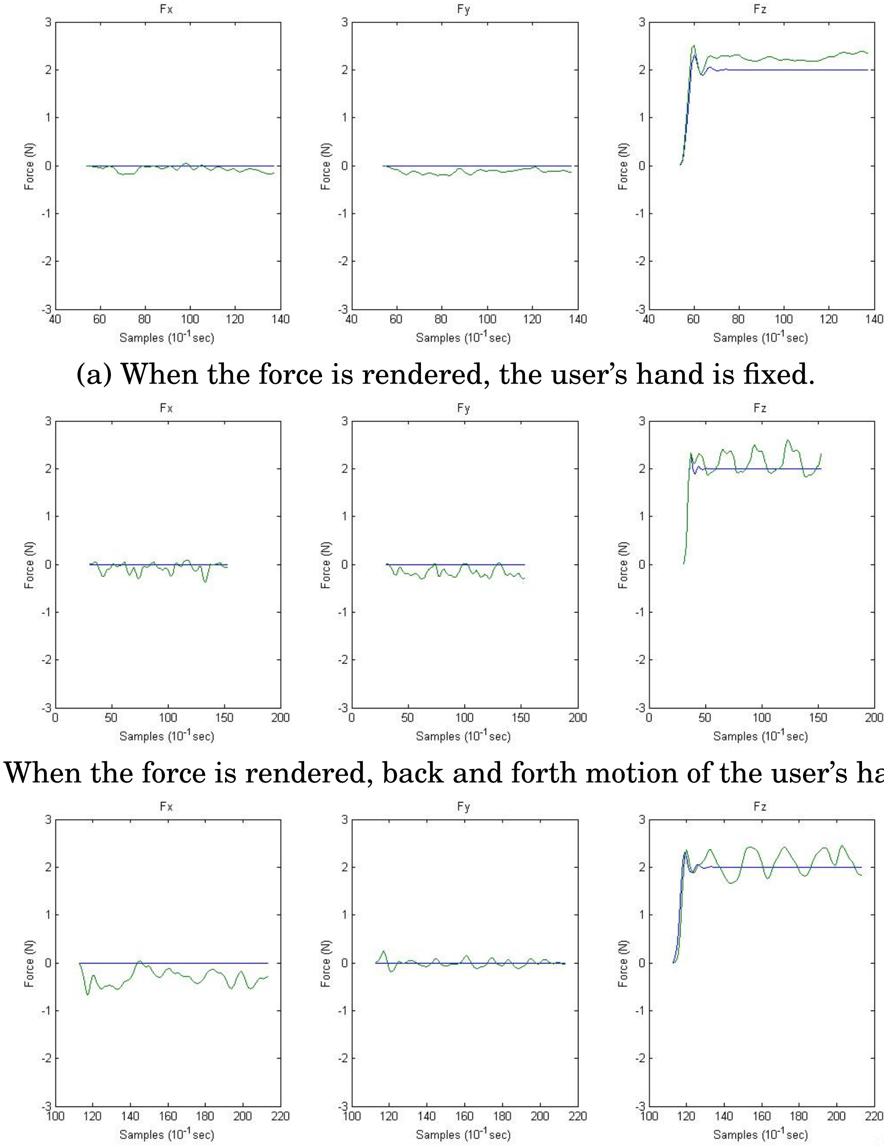
(b) Left and right motion of the mobile robot without force rendering.

Fig. 6.5 Left and right motion of the user and the mobile robot without force rendering.

the force felt by the user when the user grabs HIP and moves it is similar to the force felt by the user because of the motion of the mobile robot.

6.3.3 When the Haptic Device Renders a Wall

There is a force rendering of the cube wall which is located at the height where the user can touch. In Figure 6.7, graphs are based on the force data which is collected when the user touches the wall faces of the x and z axes. In graphs, dotted lines represent the force generated by the haptic device calculated by the virtual proxy algorithm and solid lines represent the recorded force by the force

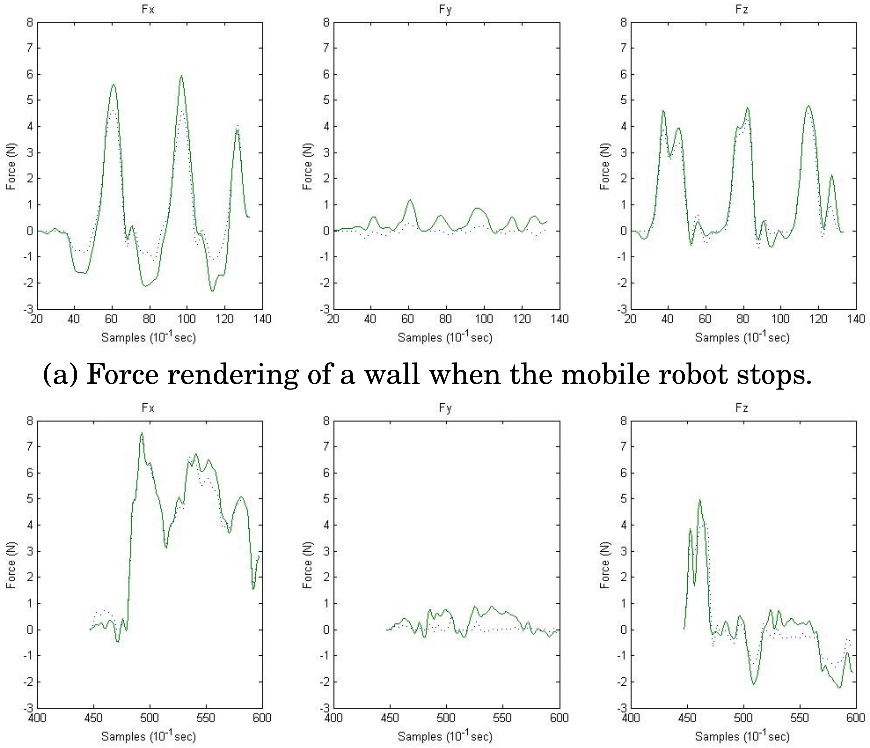


(a) When the force is rendered, the user's hand is fixed.

(b) When the force is rendered, back and forth motion of the user's hand.

(b) When the force is rendered, back and forth motion of the mobile robot.

Fig. 6.6 Data comparisons when the force is rendered.



(a) Force rendering of a wall when the mobile robot stops.

(b) Force rendering of a wall when the mobile robot moves.

Fig. 6.7 Force variation due to the movement of the mobile robot when the force rendering of a wall is given.

sensor. Even when the force calculated by the virtual proxy algorithm changes rapidly, the force generated by the haptic device is delivered to the user as it is without reference to the movement of the mobile robot.

CHAPTER 7

The Final Integrated System

7.1 System Configuration

Using two sensors, the system tracks the positions and directions of the mobile robot and the user. After reading each sensor's position and direction from IS-900 Processor at tracker server, tracker server transfers them to mobile haptic display server through wireless LAN.

A laptop computer is installed on the mobile robot, and haptic server program runs on it. At the haptic server program, haptic server transforms positions and directions to the world coordinate using the transformation ${}^{World}_{Robot}T$, ${}^{World}_{User}T$ after receiving sensor data which is set up on the user and mobile robot from tracker server. Using this position information, the motion planning algorithm is ran and the target position of the mobile robot is calculated to set up HIP of haptic device to the appropriate position without colliding the user.

After the target position of mobile robot has been set up and considering the specification of the current mobile robot and applying the region of acceleration

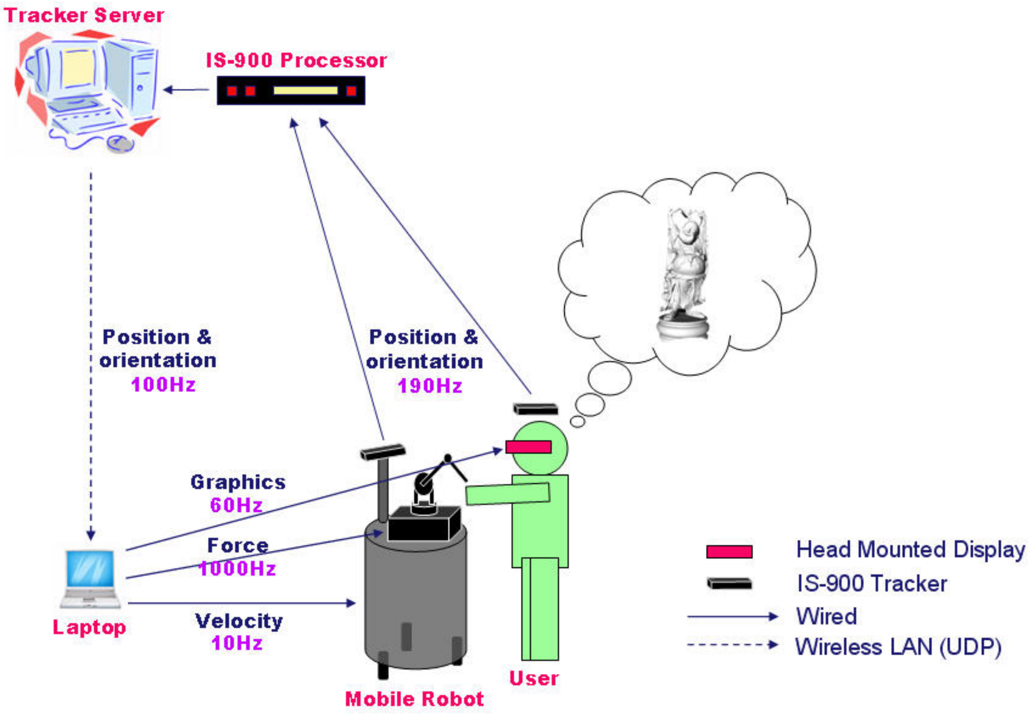


Fig. 7.1 System configuration.

and deceleration, the velocity V_x, V_z, V_θ of the mobile robot is calculated and transferred to the mobile robot through USB at 10 Hz.

When the present position and direction of the mobile robot is calculated at world coordinates, joint angles of the haptic device are read and transformation $^{Robot}_{Tool}T$ is calculated. Finally, current position of HIP is found. According to the present position of HIP, the haptic server calculates the force delivered to the user after applying the virtual proxy algorithm and this is carried out through the OpenHaptics HLAPI library.

Lastly, the haptic server transfers the information of the currently calculated position of HIP and the positions and directions of the mobile robot and the user to the visual server program at 60 Hz. The visual server program is written

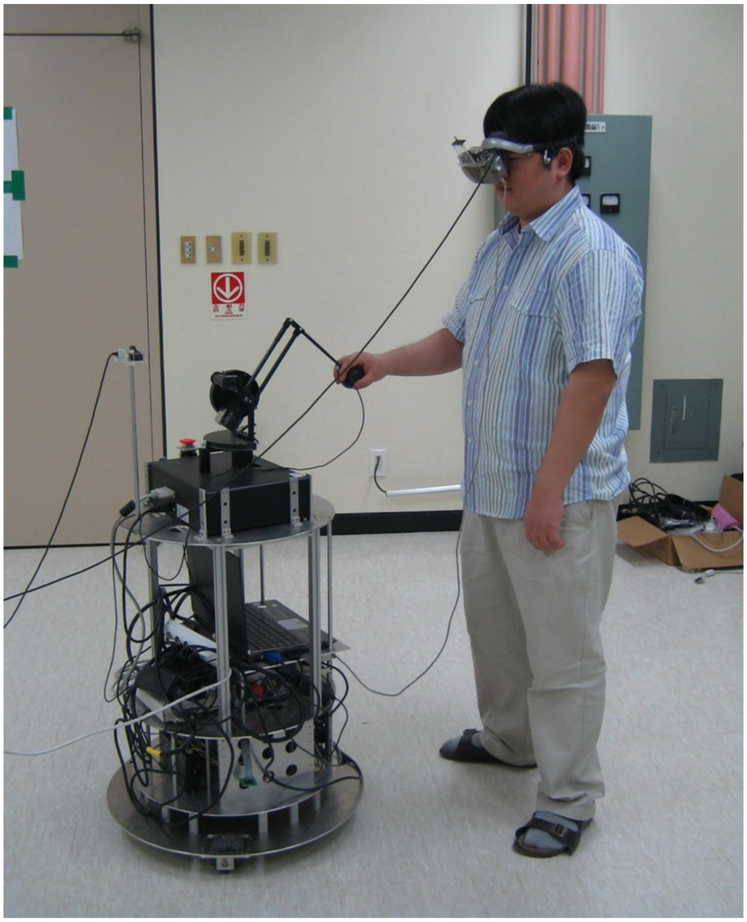


Fig. 7.2 The scene about the user using mobile haptic display.

with OpenSceneGraph library and it shows virtual environments according to the position and direction of the user's head. At this time, in order to visualize a three-dimensional scene to the user, the visual server draws two scenes for each left and right sides.

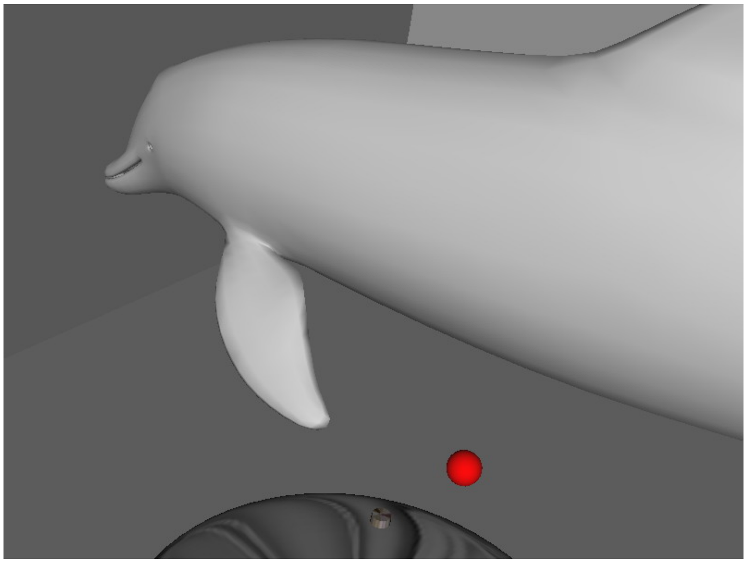


Fig. 7.3 Delivered scene to the user through head mounted display.

7.2 Limitations

As you can see in Figure 7.3, even in the case of the complicated model like a dolphin, it loads well in both haptic and visual servers. The motion planning algorithm also moves the mobile robot to the target position appropriately, avoiding the user's position. The user can feel the outline of the object using the haptic device and see the positions of the mobile robot and current HIP and the touching object in a three-dimensional vision through head mounded display.

However, we set up the maximum speed to 0.2 m/s and 40 °/s for the stability of the mobile robot. In the case when the user moves fast, HIP often reached the end of the workspace of the haptic device. In addition, because of the position - tracking error of HIP mentioned in Chapter 5.4.4.3, the position of HIP is miscalculated and leads to problems. For instance, although the real HIP is not touched with the object yet, the calculated HIP can touch the object and the unnecessary force is rendered at the calculated position of HIP.

The position information of the mobile robot is updated at 100 Hz, while the haptic rendering program runs at 1000 Hz for the stability of the haptic device. Because of this, when the position of the mobile robot is updated, HIP position is sometimes recognized to have leaped by the haptic device. Therefore, the haptic device stops force rendering due to stability reasons.

Estimation error of HIP position should be resolved by executing Kinematics calibration again with unconsidered terms included. And the problem caused by the mobile robot's late update of the position information should be solved by applying expectation of the current position of the mobile robot properly at the haptic rendering program.

CHAPTER 8

Conclusions

Mobile haptic display is the haptic rendering system in which a user can receive haptic feedback even when walking and in which he can touch an object whose size is bigger than the workspace of the haptic device with making the best use of the merits the former desktop haptic devices has (stability and detailed force rendering).

In this paper, the rendering system for mobile haptic display developed by Robotics and Automation laboratory at POSTECH is introduced. The necessary condition for the mobile haptic display operations, and proper hardware and software configuration for the device are introduced. The robot motion planning algorithm that operates the mobile robot without colliding with the user was designed and tested for the user to get haptic feedback at the appropriate positions. To modify the error between the position of the HIP calculated from sensor position and the real it of the HIP, Kinematics calibration was executed.

For a proper haptic rendering, the effect on the force generated by the haptic device of the mobile robot is theoretically analyzed. In addition, by comparing the delivered force to the user with the force generated by the haptic device using the force sensor, it is verified that the motion of the mobile robot does not affect the force generated by the haptic device.

For the future work, not only the problems obtained from the current system, such as an error of HIP position estimation and a problem caused by late update of mobile robot's position, should be modified and complemented but also the system should be further modified as an advanced system.

In the current system, the movement of the mobile robot can virtually remove the limitation of the workspace of the haptic device in the xz plane. However, the limitation of workspace of y axis (height direction) still exists. This problem can be solved by setting up a device moving up and down the haptic device on the mobile haptic display. Not satisfying with simply feeling a model, it can also be considered to render the model with the haptic texture. Lastly, it would be worth investigating to make several mobile haptic displays to operate in cooperation and analyze what the role of haptic feedback is in cooperative circumstances.

Bibliography

[1] Massie, T., Salisbury, J.: The PHANToM Haptic Interface: A Device for Probing Virtual Objects. Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (1994) 295–302

[2] Grange, S., Conti, F., Helmer, P., Rouiller, P., Baur, C.: Overview of the delta haptic device (poster). In: Eurohaptics '01, Birmingham, England (2001)

[3] Brooks, F.P., Ouh-Young, M., Batter, J.J., Kilpatrick, P.J.: Project GROPE-Haptic displays for scientific visualization. Computer Graphics **24**(4) (1990) 177–185

[4] Bouguila, L., Ishii, M., Sato, M.: Multi-modal haptic device for large-scale virtual environments. In: ACM Multimedia. (2000) 277–283

[5] Hashimoto, N., Jeong, S., Ishida, Y., Sato, M.: A novel immersive virtual environment for human-scale interaction. TENCON 2004. 2004 IEEE Region 10 Conference (2004)

[6] Hirose, M., Hirota, K., Ogi, T., Yano, H., Kakehi, N., Saito, M., Nakashige, M.: HapticGEAR: The development of a wearable force display system for immersive projection displays. In: VR. (2001) 123–130

[7] Nitzsche, N., Hanebeck, U.D., Schmidt, G.: Mobile haptic interaction with extended real or virtualenvironments. Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on (2001) 313–318

[8] Nitzsche, N., Hanebeck, U.D., Schmidt, G.: Design issues of mobile haptic interfaces. Journal of Robotic Systems **20**(9) (2003) 549–556

[9] A. Formaglio, A. Giannitrapani, M.F.D.P., Barbagli, F.: Current issues in haptic rendering using mobile haptic interfaces. In: Proceedings of the World Haptics Conference. (2005)

[10] Barbagli, F., Formaglio, A., Franzini, M., Giannitrapani, A., Prattichizzo, D.: An experimental study of the limitations of mobile haptic interfaces. In: International Symposium of Exeprimental Robotics (ISER2004). (2004)

[11] Hanebeck, U.D.; Saldic, N.S.G.: A modular wheel system for mobile robot applications. Intelligent Robots and Systems, 1999. IROS '99. Proceedings. 1999 IEEE/RSJ International Conference on **1** (1999) 17–22 vol.1

[12] InterSense, I.: Technical overview is-900 motion tracking system (2006)

[13] InterSense, I.: Intersense is-900 systems (2006)

[14] 홍민식: 전방향 이동 로봇의 속도 제어 (2007)

[15] Wu, J., Morita, S., Kawamura, S.: Human sensory fusion on visual and tactile sensing for virtualreality. Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on **3** (1996)

[16] SensAble Technologies, Inc.: Open Haptics Toolkit - API Reference. (2006)

[17] Cavusoglu, M.C., Feygin, D.: Kinematics and dynamics of phantom(TM) model 1.5 haptic interface (2001)

[18] Cavusoglu, M.C., Feygin, D., Tendick, F.: A critical study of the mechanical and electrical properties of the PHANToM haptic interface and improvements for high performance control. Presence **11**(6) (2002) 555–568

[19] Jr., J.K.S., Brock, D.L., Massie, T., Swarup, N., Zilles, C.B.: Haptic rendering: Programming touch interaction with virtual objects. In: SI3D. (1995) 123–130

[20] Ruspini, D.C., Kolarov, K., Khatib, O.: The haptic display of complex graphical environments. In: SIGGRAPH. (1997) 345–352

[21] Niku, S.: Introduction to robotics: analysis, systems, applications. Prentice Hall Upper Saddle River, NJ (2001)

[22] Haug, E.: Intermediate dynamics. Prentice Hall Englewood Cliffs, NJ (1992)

요약문

모바일 햅틱 디스플레이를 위한 렌더링 시스템

햅틱 디스플레이는 햅틱 장치를 통해 사용자에게 촉감을 제시하고, 이를 통해 가상 환경 내의 사용자의 실재감 및 몰입감을 높이는 것을 목적으로 한다. 하지만 대부분의 햅틱 장치들이 고정된 장치이고, 제한된 워크스페이스를 가지므로, 사용자는 장치의 워크스페이스보다 작은 물체나 실제보다 작은 스케일로 축소된 물체만을 촉감으로 느낄 수 있다. 본 논문에서는 이러한 햅틱 장치의 한계를 극복하고, 사용자에게 커다란 물체의 실제 크기 그대로의 촉감을 제시하기 위해서, 개발된 포스텍 이동형 햅틱 디스플레이를 소개한다. 이동형 햅틱 장치를 위한 필요한 하드웨어 구성과 적합한 소프트웨어 설계를 설명하고, 전방향성 바퀴로 제작된 이동형 로봇에 적합하도록 개발된 동작 계획 알고리즘을 제안하고, 시뮬레이터를 통해 개발된 동작 계획 알고리즘의 적합성을 보인다. 그리고 이동형 로봇이 움직임으로써 발생하는 힘이 햅틱 장치를 통해 사용자에게 어떻게 전달되는지에 대해 이론적으로 분석하고, 실제 힘 센서를 사용한 실험을 통해 그 결과를 보인다.

감 사 의 글

앗! 하는 사이에 벌써 졸업이란 걸 하게 되었습니다. 아슬아슬하게 졸업 논문을 끝내는 선배들을 보며 난 저러지 말아야지 했는데, 저도 결국 마감에 딱 맞춰서 논문을 끝내게 되는군요. 효과적인 시간관리까지는 아직 갈 길이 먼가 봅니다.

감사를 드려야 할 분들이 너무 많습니다. 우선, 부족한 제가 대학원을 무사히 마칠 수 있도록 학문적 지도는 물론, 인생의 선배로써 여러 조언을 해 주셨던 최승문 교수님께 감사의 인사를 드립니다. 날카로운 통찰력으로 논문의 문제점을 지적해 주신 이승용 교수님, 연구의 기반이 되는 하드웨어를 제작해 주신 이진수 교수님, 확신을 가지지 못한 저에게 이론적 근거를 제시해 주셨던 박종훈 박사님, 제가 연구의 길을 걷도록 도와주신 김정현 교수님 감사 드립니다.

해박한 지식과 연륜으로 제가 모르는 일에 항상 해답을 가지고 계셨던 상윤이형, 부드러운 카리스마로 일당백을 소화하셨던 남규형, 작은 일도 자기 일처럼 챙겨주신 재인이형, 뭐든지 잘 하는 성길이형, 종현이형, 석희형, 지식의 근원, 원조 백가사전 진욱이형, 모두의 걱정을 기우로 만들고 너무나 훌륭하게 랩장일을 잘 수행한 성훈이형, 동기보다 더 친하게 지낸 원조 꽃미남 재영이형, 옆 랩으로 이사 갔지만 마음 만은 언제나 우리 랩인 용진이형, 항상 투덜거리면서도 근성 있게 다 해 내는 재훈이형, 모르는 게 없는 한갑사전 갑종이, 무슨 일을 맡기더라도 안심이 되었던 인욱이, 모든 부분에 꼼꼼함이 묻어나는 인이, 투닥투닥 잘 만들어내는 건혁이까지. 이 분들 덕분에 무사히 즐겁게 랩생활을 즐길 수 있었던 것 같습니다.

같이 수업을 들으면 왠지 모르게 안심이 되었던 똑똑한 성현이형, 몇 시간 동안 묵묵히 앉아서 영어 교정을 도와준 원어민보다 영어 잘하는 후배 동연이에게도 감사의 인사를 전합니다.

6개월 동안 NASA에서 일할 때 낯선 미국 땅에서 저를 마치 아들처럼 챙겨주시

고, 학문적으로, 인간적으로 많은 영감을 주셨던 Bernard D. Adelstein 박사님께도 감사의 인사를 드립니다. (Thanks Dov!) 6개월 동안 룸메이트로 지내며, 무슨 일을 하던 항상 제 의견을 물어주고, 미국 이곳 저곳을 구경시켜 준 최고의 룸메이트 Steve 형님, 심심한 저에게 종종 인도 요리를 선사하며, 안 되는 영어로 계속 대화를 할 수 있도록 항상 말을 걸어 주었던 Raj에게도 감사의 인사를 전합니다. (Thanks Steve and Raj!)

마지막으로 기쁠 때나 슬플 때나 항상 제 옆에 있어준 든든한 여자친구 희진양에게도 고맙다는 말을 전하고 싶습니다. 사랑하는 우리 가족들. 먼저 하늘나라로 가서 저를 지켜보고 계신 엄마, 아빠. 사랑해요.

감사합니다.

Curriculum Vitae

Name	:	Chaehyun Lee

Education

2002.3. ~ 2005.8.	:	B.S. in Department of Computer Science and Engineering, POSTECH
2005.9. ~ 2008.2.	:	M.S. in Department of Computer Science and Engineering, POSTECH Thesis Title : **모바일 햅틱 디스플레이를 위한 시스템 구성 (Rendering System for Mobile Haptic Display)** Advisor: Prof. Seungmoon Choi

Publications

- **International Conference**

1. **Chaehyun Lee**, Min Sik Hong, In Lee, Oh Kyu Choi, Kyung-Lyong Han, Yoo Yeon Kim, Seungmoon Choi, and Jin S. Lee, "Mobile Haptic Interface for Large Immersive Virtual Environments: PoMHI v0.5", in *Poster and Demo Proceedings of the 2nd International Workshop on Haptic and Audio Interaction Design* , pp. 7-8, 2007.

2. **Chaehyun Lee**, Min Sik Hong, In Lee, Oh Kyu Choi, Kyung-Lyong Han, Yoo Yeon Kim, Seungmoon Choi, and Jin S. Lee, "Mobile Haptic Interface for Large Immersive Virtual Environments: PoMHI v0.5", in *Proceedings of the 4th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)* , pp. 106-111, 2007.

3. **Chaehyun Lee**, Bernard D. Adelstein, and Seungmoon Choi, "Haptic Weather", To be presented in *the 16th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems* , 2008.

- **Domestic Conference**

1. **이채현**, 이인, 최승문, "이동형 햅틱 디스플레이를 위한 동작 계획", *한국 HCI 학술대회 논문집* , pp. 578-584, 2007.

2. 정재훈, 황인욱, 이인, **이채현**, 박건혁, 황재인, 최승문, 김정현, "동작기반의 체험형 리모트 콘트롤", *한국 HCI 학술대회 논문집* , pp. 115-122, 2007.