대형 가상환경을 위한 이동형 햅틱 인터페이스: PoMHI v0.5

Mobile Haptic Interface for Large Immersive Virtual Environments: PoMHI v0.5

이 채 현[1], 홍 민 식[2], 이 인[3], 최 오 규[4], 한 경 룡[4],
김 유 연[5], 최 승 문[6], 이 진 수[7]

Chaehyun Lee[1], Min Sik Hong[2], In Lee[3], Oh Kyu Choi[4], Kyung-Lyong Han[4],
Yoo Yeon Kim[5], Seungmoon Choi[6], Jin Soo Lee[7]

Abstract We present the initial results of on-going research for building a novel Mobile Haptic Interface (MHI) that can provide an unlimited haptic workspace in large immersive virtual environments. When a user explores a large virtual environment, the MHI can sense the position and orientation of the user, place itself to an appropriate configuration, and deliver force feedback, thereby enabling a virtually limitless workspace. Our MHI (PoMHI v0.5) features with omni-directional mobility, a collision-free motion planning algorithm, and force feedback for general environment models. We also provide experimental results that show the fidelity of our mobile haptic interface.

Keywords : Mobile haptic interface, Mobile base, Omni-directional wheel, PID-control with fuzzy logic control, Haptic rendering

1. Introduction

A force-feedback haptic interface has an inherent limit in its workspace, thus in the size of virtual objects that can be rendered with it. This can be tolerable in desktop applications where virtual objects are mostly smaller than the workspace of a desktop haptic interface, but remains to be a severe limitation in large virtual environments such as the CAVE™. The traditional solution to this problem has been using a large haptic interface of a manipulator type [1] or of a string type (SPIDAR) [2][3]. However, their workspaces are still limited, and cannot be easily scaled to virtual environments of different sizes.

A promising alternative is a Mobile Haptic Interface (MHI) that refers to a force-feedback haptic interface with a mobile base. The mobile base can move the haptic interface to an adequate place to render large virtual objects, and thus can provide an unlimited workspace. Compared to other large haptic interfaces, the MHI is easier to be installed or removed due to its mobility, considerably smaller, and safer. Furthermore, the MHI can provide high-fidelity force feedback since a desktop haptic interface with greater precision (for both sensing and actuation) is used for it.

The concept of the MHI was initially proposed by Nitzsche et al [4][5]. Although their MHI, Walkii, had omni-directional mobility, a simple motion planning algorithm was used and only primitive objects could be rendered. Barbagli et al. presented two new MHIs in 2004 [6]. Both MHIs used a commercial robot as its

※ 이 논문은 2007 년도 정부(과학기술부)의 재원으로 한국과학재단의 지원을 받아 수행된 연구임(No. R01-2006-000-10808-0).

[1] LG 텔레콤 기술연구소 연구원
[2] LG 전자 디지털 스토리지 연구소/DSS 그룹 연구원
[3] 포항공과대학교 컴퓨터공학과 통합과정
[4] 포항공과대학교 전자전기공학과 박사과정
[5] 포항공과대학교 전자전기공학과 석사
[6] 포항공과대학교 컴퓨터공학과 조교수 (교신저자)
[7] 포항공과대학교 전자전기공학과 교수

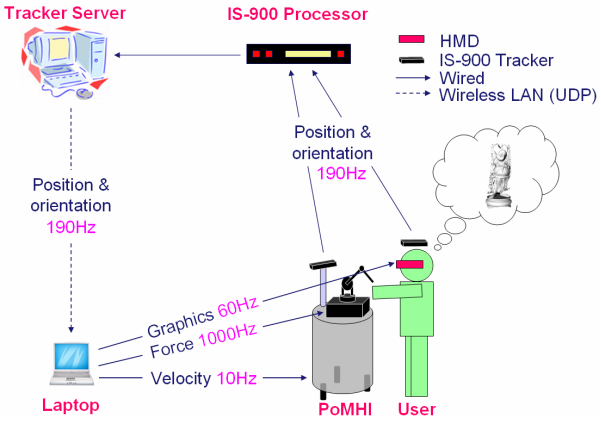

Fig. 1. Architecture of the PoMHI

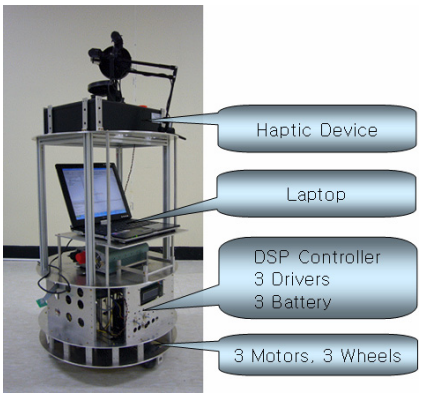

Fig. 2. Hardware structure of the PoMHI

mobile base and adopted more complex motion planning algorithm than Walkii. More recently, a MHI with two desktop haptic interfaces was proposed for two-point manipulation [7].

In this paper, we present an initial version of POSTECH Mobile Haptic Interface (PoMHI v0.5) especially designed to be used with large visual environments such as the CAVE™. The PoMHI can move in any direction using three omni-directional wheels while avoiding collisions with a user, load general virtual environment models, and provide acceptable force feedback. The mobile base with the three omni-directional wheels was controlled via a PID-control with an additional fuzzy logic control. A motion planning algorithm guiding the movement of the mobile base was also integrated. We also experimentally confirmed that the dynamics of the mobile base gives rise to little interference on the forces delivered to the user.

The remainder of this paper is organized as follows. Section 2 briefly reviews the overall architecture of the PoMHI. Section 3 describes the hardware components and the control methods used in the system. In Section 4, we describe the software structure and the motion planning strategy. The effect of the mobile base dynamics on the final rendering force is also examined. Applications of the PoMHI are discussed in Section 5. Finally, we conclude this paper in Section 6.

2. System Architecture

The overall system architecture is shown in [Fig. 1]. The IS-900 Tracking System (a processor and two trackers; InterSense Inc., USA) is responsible for tracking the current configurations (position and orientation) of both the PoMHI and the user. Tracked information is sent to a tracker server, and the server forwards it to a laptop inside the PoMHI via the UDP protocol. Based on the user's configuration, the laptop determines an appropriate configuration of the mobile base and controls the base to reach configuration. Graphic and haptic rendering modules are also managed in the laptop. The user wears a head mounted display (HMD) and sees stereoscopic scenes of a virtual environment. Relevant communication and rendering rates are also specified in the figure.

3. MHI Hardware

3.1 MHI Hardware Design

3.1.1 Overall Structure of the Hardware

The hardware structure of the PoMHI is shown in [Fig. 2]. It consists of four main parts: omni-directional wheels and geared DC motors, a DSP control board and power amplifiers, a laptop, and a desktop 3 DoF haptic interface (PHANToM Premium 1.5A; SensAble inc., USA). The mobile base is responsible for moving the whole PoMHI to face the user, and the PHANToM is for delivering appropriate force feedback to the user.

3.1.2 Omni-directional Structure

[Fig. 3] shows the mobile-base actuation design using three omni-directional wheels. Our design follows the Y-shaped structure for holonomic motion of the mobile base. This is more adequate than a bidirectional mobile base for following possibly abrupt motions of a user. Each wheel is connected to a motor by a timing belt and a pulley.

The kinematics of the mobile base is derived from the relation of parameters represented in [Fig. 4]. Here, the origin of the local coordinate is set to the center of the base and the parameters are defined as follows:

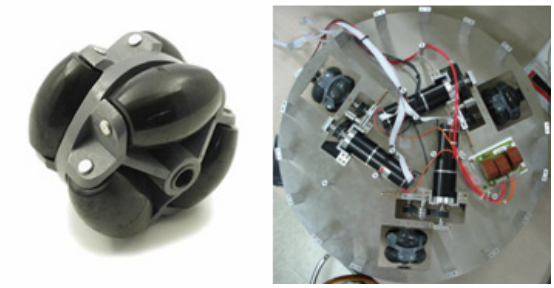

Fig. 3. An omni-directional wheel (left) and the placement of three wheels in the mobile base (right)

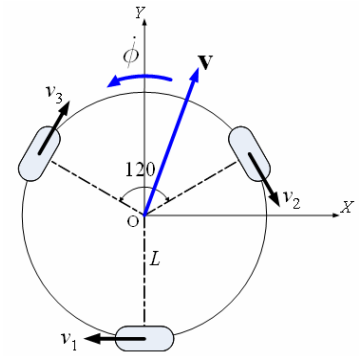

Fig. 4. Omni-directional structure on local coordinates

v_i : Linear velocity of each wheel (i = 1, 2, 3)

$\mathbf{v} = [\dot{x}, \dot{y}]^T$: Linear velocity of the mobile base

$\dot{\phi}$: Angular velocity of the mobile base

L : Distance between a wheel and the origin.

Then, the mobile base kinematics can be derived as [8][9]:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} -1 & 0 & -L \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} & -L \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & -L \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \end{pmatrix} = \mathbf{A}\dot{\mathbf{S}}, \qquad [1]$$

Where $\dot{\mathbf{S}} = \begin{bmatrix} \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^T = \begin{bmatrix} \mathbf{v}^T & \dot{\phi} \end{bmatrix}^T$.

3.2 MHI Control

3.2.1 Overview of Mobile Base Control

The motion of the mobile base is based on the PID control for its desired linear and angular velocities. As shown in [Fig. 3], each omni-directional wheel has six free-rolling sub-wheels that may cause a slip. The mobile base also contains inherent nonlinearity induced from the wheel and belt-pulley structure. Therefore, additional fuzzy logic control (FLC) and other supplementary techniques are also employed to overcome the problem. The overall control equation for motor i is:

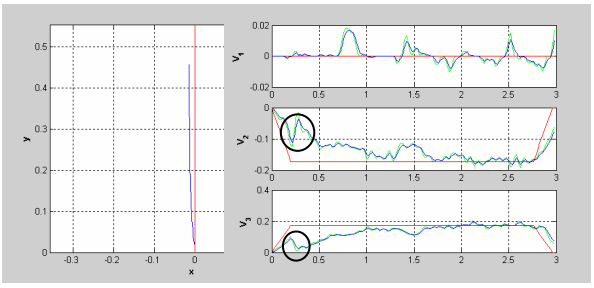

Fig. 5. Experimental results only using the PID-based velocity control ($\dot{x}$ = 0.0 m/s, $\dot{y}$ = 0.2 m/s, $\dot{\phi}$ = 0.0 rad/s)

$$U_i(k+1) = U_i(k) + \Delta U_i(k) + \Gamma_i(k), \qquad [2]$$

where $U_i(k)$ is a command to the motor amplifier and $\Gamma_i(k)$ is an output of the supplementary control. The sampling rate of the control loop is set to 50 Hz.

3.2.2 PID-based Velocity Control

Once a desired trajectory is determined the mobile base from the motion planning algorithm (will be explained later), the desired velocity of each wheel is calculated by the kinematics in [Eq. 1]. Using the velocity, each motor is controlled by the PID control. The velocity of each wheel is estimated using the exponential filter as:

$$v_{i,\,filtered}(k) = (1-\alpha)v_{i,\,filtered}(k-1) + \alpha v_i(k), \qquad [3]$$

where $0 \le \alpha \le 1$ is a filter parameter.

The PID control equation that we use to obtain the desired motor velocity is:

$$U_i(k+1) = K_p e_i(k) + K_i \sum_{i=1}^{k} e_i(i) + K_d \Delta e_i(k), \qquad [4]$$

where $e_i(k) = v_{i,desired}(k) - v_{i,\,filtered}(k)$ is the velocity error of the motor, and K_p, K_i, and K_d are the PID gains.

Using the PID control only does not result in sufficient position tracking performance in our system. This is illustrated in [Fig. 5] where the red and blue lines represent references and measured values for mobile base position, respectively. The left graph shows the position of the PoMHI, and the right three graphs show the

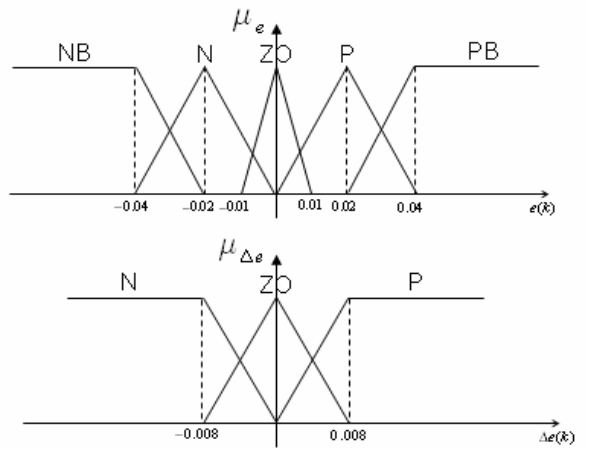

Fig. 6. Membership functions of velocity error (μ_e) and its variation ($\mu_{\Delta e}$)

Table 1. Control rule set

	$e(k)$				
$\Delta e(k)$	NB	N	ZO	P	NP
N	$\bar{y}^{(1,1)}$	$\bar{y}^{(2,1)}$	$\bar{y}^{(3,1)}$	$\bar{y}^{(4,1)}$	$\bar{y}^{(5,1)}$
ZO	$\bar{y}^{(1,2)}$	$\bar{y}^{(2,2)}$	$\bar{y}^{(3,2)}$	$\bar{y}^{(4,2)}$	$\bar{y}^{(5,2)}$
P	$\bar{y}^{(1,3)}$	$\bar{y}^{(2,3)}$	$\bar{y}^{(3,3)}$	$\bar{y}^{(4,3)}$	$\bar{y}^{(5,3)}$

velocities of the wheels. We can observe the system nonlinearity in the black circles where the measured velocity rather decreases in spite of an increase of the reference velocity. The nonlinearity seems to be caused by the increase of friction when the contact point of a wheel between a sub-wheel and the floor is changed. Because of this nonlinearity of the wheel structure, eliminating this phenomenon perfectly may not be feasible. Therefore, our approach has been minimizing the effect of the nonlinearity through additional control methods that will be described in the subsequent section.

3.2.3 Fuzzy Logic Control

We use a singleton fuzzifier, a product inference engine, and a center average defuzzifier for the FLC of the mobile base. The inputs are the velocity error $e_i(k)$ and its increment $\Delta e_i(k)$. We use triangular membership functions (μ_e and $\mu_{\Delta e}$) shown in [Fig. 6]. Membership functions consist of five sections; negative big (NB), negative (N), zero (ZO), positive (P), and positive big (PB). These sections are determined experimentally.

Table 1 shows the rule set of the FLC derived from the input membership functions. Each element of the table, $\bar{y}^{(i,j)}$, represents the output of each rule. Using the rules and input variables, the updating output of FLC is [3]:

$$\Delta U_i(k) = \frac{\sum_{m=1}^{5}\sum_{n=1}^{3} \bar{y}^{(m,n)} \mu_e^m(k) \mu_{\Delta e}^n(k)}{\sum_{m=1}^{5}\sum_{n=1}^{3} \mu_e^m(k) \mu_{\Delta e}^n(k)}. \quad [5]$$

3.2.4 Supplementary Control

We adopt additional supplementary control when the following condition is satisfied:

$$|U_i(k)| > |U_i(k-1)| \ \& \ |v_{i,\,filtered}(k)| < |v_{i,\,filtered}(k-1)|. \quad [6]$$

This condition detects the moments when the contact point of a wheel between a sub-wheel and the floor is changed. During this period, excessive errors are accumulated in the PID control, and they cause an overshoot in the position control. An algorithm to minimize this behavior is described below.

If [Eq. 6] is satisfied at time index k, the controller stores the sum of updating outputs $\Delta U_i(k)$ and the current velocity $v_e = v_{filtered}(k)$. In each control time l after k, the controller checks whether a condition, $|v_e| < |v_{filtered}(l)|$, is met. If it is, the controller compensates the excessive error by:

$$U_{i,sum} = \sum_{i=k}^{l} \Delta U_i(k) \quad \text{and} \quad [7]$$

$$U_i(l) = U(l) - \gamma \cdot U_{i,sum}, \quad [8]$$

where $0 \le \gamma \le 1$ is a proportional constant. Without the use of this control process, the system can be unstable. The final output of the supplementary control is as follows:

$$\Gamma(k) = \begin{Bmatrix} -\gamma \cdot U_{i,sum}(l), & k = l \\ 0, & k \neq l \end{Bmatrix}. \quad [9]$$

3.2.5 Experimental Results

[Fig. 7] presents the results with the adapted FLC and the supplementary control. In the velocity graph of each wheel, fine trembles due to the change of contact points still appear, but the effect of the inherent nonlinearity has been significantly decreased. Moreover, in the position graph, measured position values followed the reference values (a straight line) almost perfectly.

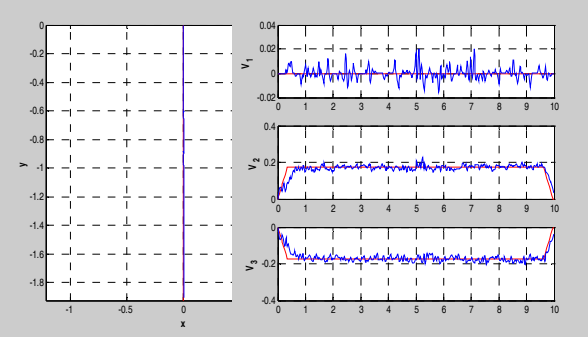

Fig. 7. Experimental results with the modified velocity control ($\dot{x}$ = 0.0 m/s, $\dot{y}$ = 0.2 m/s, $\dot{\phi}$ = 0.0 rad/s)

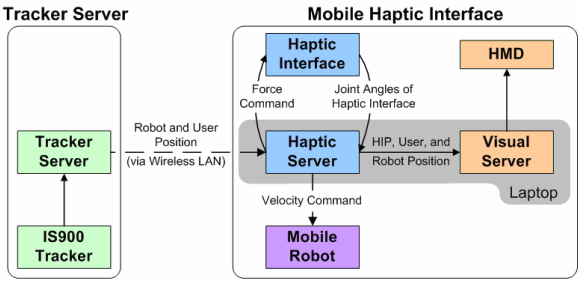

Fig. 8. Software structure of PoMHI

4. MHI Software

4.1 MHI Software Design

The software in the MHI system consists of three parts as shown in [Fig. 8]. A tracker server program that runs in the tracker server receives the configurations of the two IS-900 trackers (one for the robot base and the other for the user) and sends the information to the laptop in the MHI via wireless LAN. Another program, a haptic server, that is performed on the laptop in the MHI controls all devices in the MHI (except for the motors directly controlled by the DSP board) and send force rendering commands to the haptic device. This program communicates with a visual server responsible visual rendering with a HMD.

The tracker server program and the MHI communicate through wireless LAN. Limited by the update rate of the IS-900 tracker, the network update rate is set to 190 Hz. To maintain this speed, the UDP protocol is used instead of the TCP protocol because 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 to recover the missed packet and may result in a long delay which cannot be tolerable in our application. The UDP is not as complex as the TCP, and is therefore faster. Although some packets can be missed and reordered during transmission in UDP, its effect can be made unnoticeable to a user.

The haptic server program is comprised of networking, motion planning and haptic rendering modules. First, the networking module receives the tracker information from the tracker server and transforms the data in the coordinate system of the IS-900 tracker to those in the world coordinate frame. This updates the configuration of the user, the mobile base and the haptic interface point (HIP; a point modeling the haptic tool tip) and sends the new information to the visual server through network. Note that since the haptic and visual server programs communicate with each other via network, the two servers can operate on separate machines for purposes such as higher performance or improved convenience. For now, the two severs are executed on the same laptop. Second, the motion planning module calculates the next proper configuration of the mobile base and sends the corresponding commands of angular and linear velocities to the mobile base. Finally, the haptic rendering module detects a collision between the HIP and virtual objects and calculates appropriate haptic feedback force. Since all of these modules are performed at different rates, the haptic server is designed as a multithreaded program.

The last server program, the visual server, receives the configuration information from the haptic server and renders visual scenes. At present, a HMD is used for visual rendering, so the visual server also runs on the laptop with the haptic server. If other visual systems are used (e.g. CAVE^{TM}), the visual sever can be easily ported to another machine that controls the visual display and communicate with the haptic server through network, as was demonstrated in [11].

4.2 Motion Planning Algorithm

The role of the mobile base is to place the haptic device in a proper position where the device can give force feedback to a user most effectively while avoiding collisions with a user. Considering these constraints, we set the goal position of the MHI on a line which passes the user and the user's hand grasping the haptic tool. The distance between the user and the MHI is maintained to be within the typical arm length of an adult. Moreover, the MHI always attempt to face the user so that the haptic interface tool is positioned in front of the user.

To represent the motion of the MHI, we introduce a 3D configuration space where its x and z axes represent the 2D position of the mobile base and its y axis the orientation of the MHI, all in the world coordinate frame. In the configuration space, a configuration of the MHI is

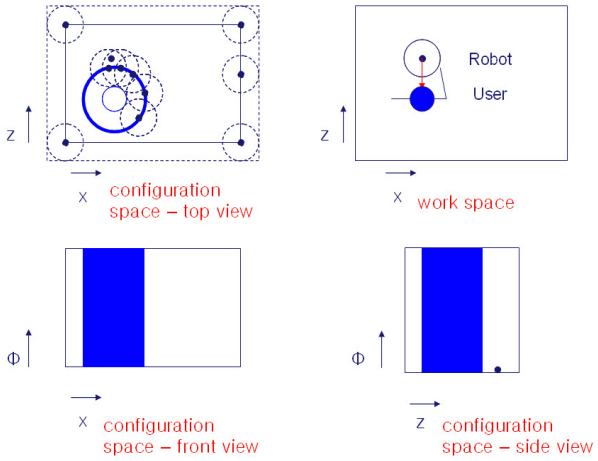

Fig. 9. Configuration space of the MHI

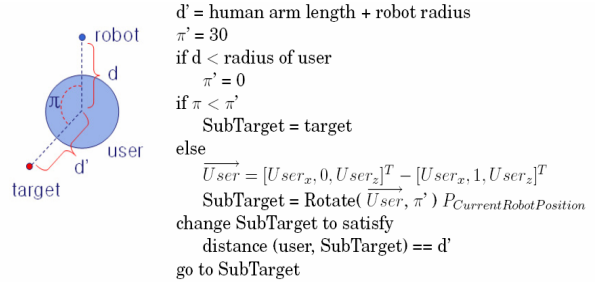

Fig. 10. Motion planning algorithm

represented as a point, and a user is represented as a cylindrical obstacle (see [Fig. 9]).

To move the MHI to a goal configuration without colliding with the user, we developed a motion planning strategy that consists of three cases. First, when the MHI is too close to the user, it moves away from the user as quickly as possible to avoid possible collisions with the user. Second, when the goal position is close to the current position, the MHI moves toward the goal. Finally, when the goal position cannot be reached directly (e.g., when the user is on the path), the MHI sets sub-goals between the goal and current position of the MHI and moves towards the nearest sub-goal. A pseudo-code for this strategy is shown in [Fig. 10].

In order to assess the performance of the motion planning algorithm, we simulated the motions of the mobile base and the user and represented the results with graphs in [Fig. 11] and [Fig. 12]. The simulation process was simplified by assuming that the mobile base could always move with its maximum velocity. In each graph, the red circle represents the current position of the MHI, and the blue region represents the next positions of the user where the mobile base can catch up with the user in 1 second. The results indicate that the maximum linear

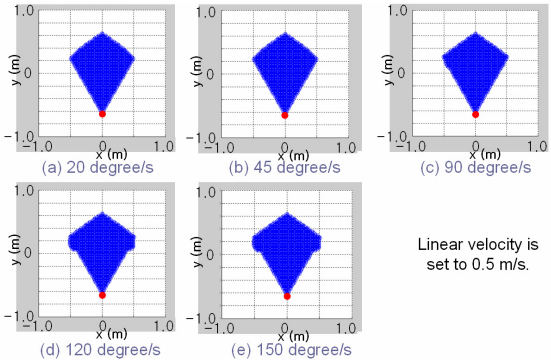

Fig. 11. Regions that the mobile base can move in 1 second with different maximum angular velocities

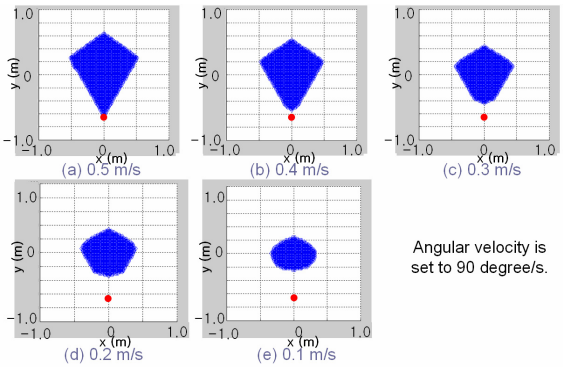

Fig. 12. Regions that the mobile base can move in 1 second with different maximum angular velocities

velocity is much more significant factor than the maximum linear velocity for the planning performance.

4.3 Force Rendering Algorithm

In order to calculate feedback force given the HIP position and a virtual environment model, we need consider the effect of the mobile base movement on the final force perceived by the user. Let $\mathbf{F}_T$ be the translational force applied to the user at the HIP, and $\mathbf{T}_P$ be the joint torque of the haptic device. In addition, we denote the displacements of the HIP, the mobile base, and the PHANToM joints by $\mathbf{D}_T$, $\mathbf{D}_R$, and $\mathbf{D}_P$. Then, using the body Jacobian,

$$\dot{\mathbf{D}}_T = \begin{bmatrix}\mathbf{J}_R & \mathbf{J}_P\end{bmatrix}\begin{bmatrix}\dot{\mathbf{D}}_R \\ \dot{\mathbf{D}}_P\end{bmatrix} = \mathbf{J}_R\dot{\mathbf{D}}_R + \mathbf{J}_P\dot{\mathbf{D}}_P . \qquad [10]$$

If the mobile base is perfectly position-controlled, $\dot{\mathbf{D}}_R$ approaches to $\mathbf{0}$, i.e., the mobile base becomes to have infinitely large apparent inertia. Then, following [12],

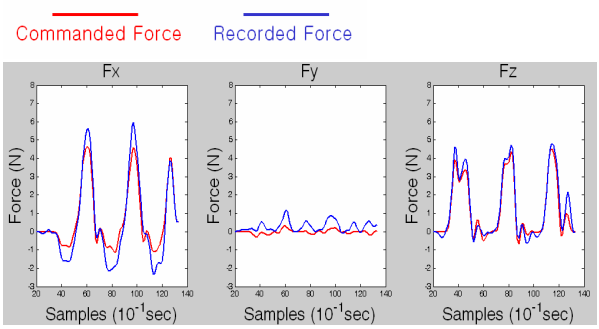

Fig. 13. Comparison between commanded and measured forces when the mobile base was stationary.

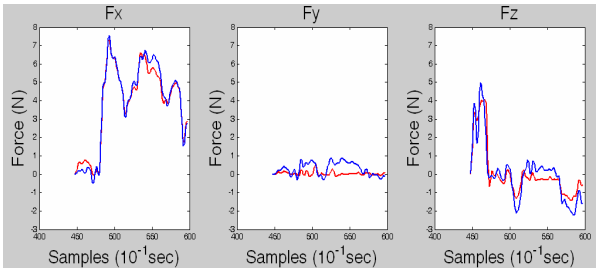

Fig. 14. Comparison between commanded and measured forces when the mobile base was moving.

$$\dot{\mathbf{D}}_T = \mathbf{J}_P \dot{\mathbf{D}}_P \Rightarrow \mathbf{T}_P = \mathbf{J}_P{}^T \mathbf{F}_T \,. \qquad [11]$$

This indicates that one may determine the joint torque to the desktop haptic device without considering the effect of mobile base movements, for simplicity. To verify this fact, we conducted an experiment with a force sensor attached between the lank link of the haptic device and the puck held by a user. The user touched a virtual wall, and commanded forces and actual forces delivered to the user were compared for two cases: when the mobile base is stationary and when it is in motion.

The results of the experiment are shown in [Fig. 13] and [Fig. 14] where the red lines represent commanded force to the haptic device and the blue lines recorded force by the force sensor. It can be observed that the differences between the commanded and recorded forces for the moving mobile base are not evidently greater than those for the stationary mobile base. This implies that the mobile base dynamics does not significantly affect the feedback force delivered to the user. It follows that in our current implementation we calculate torque commands sent to the haptic interface only considering the static torque-force relationship of the desktop haptic interface using the conventional virtual proxy algorithm that is widely used for haptic rendering with desktop haptic interfaces [12].

4.4 Virtual Environment Model

Not only primitive models such as a plane and a cylinder but also complex triangular mesh models can be loaded and rendered in the PoMHI. The visual server also shows the current position of the HIP and the wall for a warning about the physical workspace limit in a room. The model of the mobile base is also loaded and visually rendered via a HMD in order to help the user avoid colliding with the mobile base for the safety purpose.

5. Applications

The current MHI system can provide the user with visual and haptic feedback about large virtual environments. One example is shown in [Fig. 15]. A complex dolphin model consisting of 4488 faces is loaded, and the user sees and touches the real-sized dolphin model (1258 mm wide, 426 mm high, and 352 mm deep). For the stability of the mobile base, we set the maximum linear and angular velocity to 0.2 m/s and 40 degree/s, respectively. With this maximum velocity values, the MHI can follow the user in most cases, unless the user moves abruptly. Although formal user studies on the experiences of using the PoMHI is being planned, most users subjectively reported that looking at and touching large objects via the PoMHI system was fun and exciting.

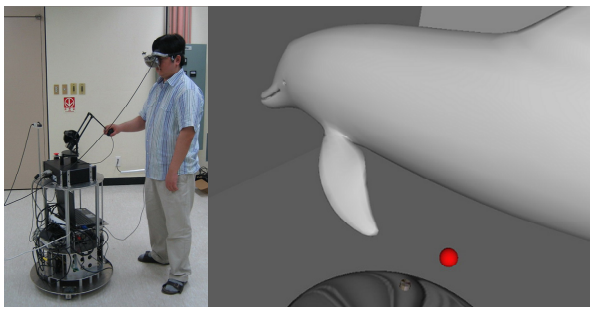

Fig. 15. A user playing with the virtual dolphin using the PoMHI (left). Visual scenes displayed to the user via the HMD (right). The red sphere represents the HIP.

6. Conclusions and Future Work

We have developed an initial version of new mobile haptic interface named PoMHI and its applications. The PoMHI locates itself to an appropriate configuration to provide boundless haptic feedback while avoiding collisions with the user, and handles general virtual environment models. We adopted several motion control methods to stably control the motion of the mobile base

with high accuracy. We also examined the fidelity of force display in the PoMHI by comparing actual force outputs with desired values.

We are currently working on a next version of the PoMHI. This version has four omni-directional wheels with advanced design and a lift for the desktop haptic interface to extend of its workspace in the height direction also. In addition, we are upgrading the software in terms of a more sophisticated motion planning algorithm, more precise kinematics calibration, and a force computation algorithm considering the effect of the mobile base dynamics. Once all of these are completed, we will integrate the PoMHI into the CAVE™ that is the most immersive large virtual environment platform among the present.

참 고 문 헌

[1] F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick, “Project GROPE - Haptic Displays for Scientific Visualization,” In Proc. SIGGRAPH 90, pp. 177-185, 1990.

[2] L. Buoguila, M. Ishii, and M. Sato, “Multi-Modal Haptic Device for Large-Scale Virtual Environment,” In Proc. 8th ACM International Conference on Multimedia, pp. 277-283, 2000.

[3] N. Hashimoto, S. Jeong, Y. Takeyama, and M. Sato, “Immersive Multi-Projector Display on Hybrid Screens with Human-Scale Haptic and Locomotion Interfaces,” In Proc. International Conference on Cyberworlds, pp. 361-368, 2004.

[4] N. Nitzsche, U. D. Hanebeck, and G. Schmidt, “Mobile haptic interaction with extended real or virtual environments,” In Proc. RO-MAN 2001, pp. 313-318, 2001.

[5] N. Nitzsche, U. D. Hanebeck, and G. Schmidt, “Design Issues of Mobile Haptic Interfaces,” Journal of Robotics Systems, vol. 20, no. 9, pp. 549-556, 2003.

[6] F. Barbagli, A. Formaglio, M. Franzini, A. Gianni-trapani, and D. Prattichizzo, “An experimental study of the limitations of mobile haptic interfaces,” In Proc. ISER 2004, 2004.

[7] Formaglio, M. D. Pascale, and D. Prattichizzo, “A mobile platform for haptic grasping in large environments,” Virtual Reality, vol. 10, no. 1, pp. 11-23, 2006.

[8] Leow Y.P., Low K.H., Loh W.K., “Kinematic modeling and analysis of mobile robots with omni-directional wheels,” In Proc. ICARCV 2002. vol. 2, pp. 820-825, 2002.

[9] Yong Liu, Xiaofei Wu, J Jim Zhu, Jae Lew, “Omni-Directional Mobile Robot Controller Design by Trajectory Linearization,” In Proc. American Control Conference, 2003.

[10] L. X. Wang, A Course in Fuzzy Systems and Control, Prentice-Hall, 1997.

[11] E. Dorjgotov, S. Choi, S. R. Dunlop, and G. R. Bertoline, “Portable Haptic Display for Large Immersive Virtual Environments,” In Proc. HAPTICS 2006, pp. 321-327, 2006.

[12] E. J. Haug, “Intermediate Dynamics,” Prentice Hall, 1992.

[13] D. Ruspini, K. Kolarov, and O. Khatib, “The Haptic Display of Complex Graphical Environments,” In Proc. ACM SIGGRAPH, pp. 345-352, 1997.

이 채 현

2005 포항공과대학교 컴퓨터공학과(공학사)
2008 포항공과대학교 전자컴퓨터공학부 컴퓨터공학전공(공학석사)

2008~현재 LG 텔레콤 기술연구소 연구원
관심분야 : 햅틱스 응용분야

홍 민 식

2005 성균관대학교 정보통신공학부(공학사)
2007 포항공과대학교 전자전기공학과(공학석사)

2007~현재 LG전자 디지털 스토리지 연구소 연구원
관심분야 : 로보틱스

이 인

2006 성균관대학교 정보통신공학부 컴퓨터공학과(공학사)

2006~현재 포항공과대학교 컴퓨터공학과 통합과정
관심분야 : 햅틱스, 가상현실, 정신물리학

최 오 규

2005 포항공과대학교 전자전기공학과(공학사)
2008 포항공과대학교 전자전기공학과(공학석사)
2008~현재 포항공과대학교 전자전기공학과 박사과정
관심분야 : 이동 로봇

한 경 룡

2002 포항공과대학교 전자전기공학과(공학사)
2004 포항공과대학교 전자전기공학과(공학석사)
2006 삼성중공업 생산기술연구소 자동화연구 선임연구원
2006~현재 포항공과대학교 전자전기공학과 박사과정
관심분야 : 이동 로봇

김 유 연

2005 북경 칭화대학 자동화과(공학사)
2008 포항공과대학교 전자전기공학과 (공학석사)
2008~현재 유학 준비
관심분야 : 로보틱스

최 승 문

1995 서울대학교 제어계측공학과(공학사)
1997 서울대학교 제어계측공학과(공학석사)
2003 Purdue University, School of Electrical and Computer Engineering (Ph.D.)
2005 Purdue University, Envision Center for Data Perceptualization, Post-Doctoral Research Associate.
2005~현재 포항공과대학교 컴퓨터공학과 조교수
관심분야 : 햅틱스, 가상현실, 로봇공학, 응용인지학, 데이터 지각화

이 진 수

1972 서울대학교 전자전기학과(공학사)
1980 Department of Electrical Engineering and Computer Science, University of California, Berkely.(공학석사)
1984 Department of System Science, University of California, LosAngeles. (공학박사)
1985 Member of Technical Staff, AT&T Bell Laboratories, Holmdel, New Jersey
1989 Senior Member of Engineering Staff, GE Advanced Technology Laboratories, Moorestown, New Jersey
1989~현재 포항공과대학교 전자전기공학과 교수
관심분야 : 로보틱스, 자동화