A Mobile Haptic Interface (MHI), a force-feedback haptic interface mounted on a mobile base, allows to render very large virtual objects in a safe and portable manner. In this paper, we present a novel MHI system, featured with: 1) extended horizontal workspace using a linear lift, 2) high-accuracy estimation of the position of the haptic interface tool, 3) efficient motion planning algorithm to move the haptic interface system, 4) closed-loop force control to compensate the undesired effect of mobile base dynamics on the final rendering, and 5) the capability of the system to handle other objects, such as the CAVE face with a mobile base, allows to render very large virtual objects in a safe and portable manner.

Index Terms:
- Haptic interfaces
- Mobile haptic interface
- Mobile robots
- Force feedback
- Motion planning
- Workspace
- Force control
- Virtual environments


details of the improvements are described in the rest of this paper.

The approach of the MHI and the human-robot interaction are described in the following sections. A next version had two desktop haptic interfaces for accomplishing two-point interaction in a large virtual environment (e.g., CAVE or large projection screen, etc.). Our first MHI system had a self-contained visual display, user tracking, motion planning, and haptic rendering software for user tracking, motion planning, haptic rendering, and visual rendering. In this paper, we present the second version of POSTECH MHI (pMHI) system that has significantly improved their performances [1]. A next version had two desktop haptic interfaces for accomplishing two-point interaction in a large virtual environment (e.g., CAVE or large projection screen, etc.). Our first MHI system had a self-contained visual display, user tracking, motion planning, and haptic rendering software for user tracking, motion planning, haptic rendering, and visual rendering. In this paper, we present the second version of POSTECH MHI (pMHI) system that has significantly improved their performances [1].

The abstract of the paper is as follows:

A Mobile Haptic Interface (MHI) refers to a force-feedback haptic interface mounted on a mobile base. In this paper, we present a novel MHI system, featured with: 1) extended horizontal workspace using a linear lift, 2) high-accuracy estimation of the position of the haptic interface tool, 3) efficient motion planning algorithm to move the haptic interface system, 4) closed-loop force control to compensate the undesired effect of mobile base dynamics on the final rendering, and 5) the capability of the system to handle other objects, such as the CAVE face with a mobile base, allows to render very large virtual objects in a safe and portable manner.

In the past five years, different MHIs have been developed by several groups. Nitzsche et al. [12] was the first to demonstrate the feasibility of MHI using an omni-directional mobile robot, magnetic tool, and soft actuators. Another group led by Prattichizzo has proposed a motion-planning algorithm that considered the manipulators of the MHI and the human arm workspace [16]. Moreover, the software of the pMHI system includes a few advanced features. First, the estimation method of HIP position is the potential field is incorporated to drive the mobile base to a proper position without colliding with the user or the environment (as Figure 1. (a) A user is touching a real-sized virtual cow (1.67 m × 0.46 m × 1.0 m; modeled with 9,593 faces) using the POSTECH MHI system. (b) The virtual environment that the user experiences. (c) An example of visual scenes displayed to the user.

The abstract of the paper is as follows:

A Mobile Haptic Interface (MHI) refers to a force-feedback haptic interface mounted on a mobile base. In this paper, we present a novel MHI system, featured with: 1) extended horizontal workspace using a linear lift, 2) high-accuracy estimation of the position of the haptic interface tool, 3) efficient motion planning algorithm to move the haptic interface system, 4) closed-loop force control to compensate the undesired effect of mobile base dynamics on the final rendering, and 5) the capability of the system to handle other objects, such as the CAVE face with a mobile base, allows to render very large virtual objects in a safe and portable manner. In this paper, we present a novel MHI system, featured with: 1) extended horizontal workspace using a linear lift, 2) high-accuracy estimation of the position of the haptic interface tool, 3) efficient motion planning algorithm to move the haptic interface system, 4) closed-loop force control to compensate the undesired effect of mobile base dynamics on the final rendering, and 5) the capability of the system to handle other objects, such as the CAVE face with a mobile base, allows to render very large virtual objects in a safe and portable manner. In this paper, we present a novel MHI system, featured with: 1) extended horizontal workspace using a linear lift, 2) high-accuracy estimation of the position of the haptic interface tool, 3) efficient motion planning algorithm to move the haptic interface system, 4) closed-loop force control to compensate the undesired effect of mobile base dynamics on the final rendering, and 5) the capability of the system to handle other objects, such as the CAVE face with a mobile base, allows to render very large virtual objects in a safe and portable manner. In this paper, we present a novel MHI system, featured with: 1) extended horizontal workspace using a linear lift, 2) high-accuracy estimation of the position of the haptic interface tool, 3) efficient motion planning algorithm to move the haptic interface system, 4) closed-loop force control to compensate the undesired effect of mobile base dynamics on the final rendering, and 5) the capability of the system to handle other objects, such as the CAVE face with a mobile base, allows to render very large virtual objects in a safe and portable manner.
2 System Overview

The hardware and software components of the pMHI system is shown in Figure 2 with data communication rates between the components. The pMHI system uses the IS-900 SimTracker tracking system (InterSense Inc., USA). The pose (position and orientation) of the mobile base is measured by a tracker mounted on the base (see Figure 4) and that of the user is by another tracker on the HMD worn by the user. The tracker server transforms the measured local coordinates to the world coordinate frame, and sends the information to the haptic server, which runs on a laptop on the pMHI, via wireless Ethernet using UDP.

The haptic server consists of three modules for haptic rendering, motion planning, and networking, respectively. The haptic rendering module updates the position of the HIP in the world coordinate frame based on the pose of the mobile base and the pose of the HIP in the local coordinate frame of the haptic interface, detects collisions between the HIP and virtual objects in a VE, and calculates appropriate feedback force. The motion planning module determines the desired pose of the mobile base considering the user’s current pose, and computes velocity commands that are sent to the mobile base. The networking module is responsible for receiving the pose information from the tracker server and sending the position of the HIP to the visual server with other information.

The visual server in charge of visual rendering is implemented in a separate program from the haptic server, so that they can run on different machines and communicate through network. This allows the pMHI system to be independent from a visual display, as in portable haptic display [5], and to use multiple visual displays simultaneously (e.g., one HMD for the user of the pMHI and a large projector screen for observers).

3 Hardware

A picture of the pMHI is given in Figure 3. Instead of using a commercial mobile robot as a mobile base [12, 6], we have developed our own, especially to integrate a linear lift that expands the workspace of a force-feedback interface in the vertical direction. We note that this feature was conceived independently from [15].

The pMHI has four main parts: force-feedback device, linear unit, driving control box, and wheel base. For a force-feedback device, we use a PHANToM (model 1.5A; SensAble Inc., USA). We replaced the default tool of the PHANToM with a ball-shaped tool (diameter = 6 cm) to provide more convenient and firm grip to the user who is usually standing to use the pMHI and a large projector screen for observers.

4 Position Estimation of Haptic Interface Point

Let $B_H$ be a homogeneous transform from coordinate frame $A$ to $B$. In the pMHI, the pose of the HIP in the world coordinate frame can be computed by

$$ W_H = W_B B_H W_F W_H, \quad (1) $$

where $B$, $P$, and $H$ are the coordinate frames of the tracker fixed on the mobile base, the PHANToM, and the HIP, respectively (see Figure 4). Thus, the accuracy of estimating $W_H$ depends on the accuracy of the three transforms on the right hand side. Among them, $W_H$ is very accurate owing to the high position sensing accuracy of PHANToM (nominal accuracy of the PHANToM 1.5A = 0.03 mm). However, $W_B$ and $W_F$ may not, due to the limited resolution of a 3D tracker and inaccurate kinematics calibration, respectively.

Such inaccuracy can cause a few problems. If the error is relatively static, the estimated position of the HIP, thus its visual and
haptic representation, has an offset from its real position. Therefore, an error exists between the true HIP position sensed by the sensorimotor system and the displayed position sensed by the vision. When the user cannot see his hand (e.g., with a video see-through HMD), the user may not notice the error unless it is fairly large. When the user can see his hand (e.g., with an optical see-through HMD or in a CAVE™), however, the error makes the visual representation of the HIP (e.g., displayed by the HMD) positioned at a different location from that of the real HIP (e.g., seen through the HMD), creating an easily-perceptible artifact. In the worst case, the user who is not touching any virtual objects by vision can receive force feedback, or vice versa. If the error is dynamic, it can make the response force of the haptic device change abruptly, even causing rendering instability. This problem can be common to any MHI systems that use 3D trackers, but has not been seriously addressed. The rest of this section explains how we reduced the position estimation error.

4.1 Tracker Performance

A key factor that determines the best achievable accuracy in estimating the position of HIP is the performance of the IS-900 SimTracker tracking system used to measure the pose of the mobile base ($W\hat{H}_B$). Thus, we carefully examined its performance. This tracker is a hybrid tracker that combines the inertial and ultrasonic tracking technologies and is widely used in large immersive VR applications. The nominal accuracies of the tracker are 2 – 3 mm in position, 0.25° RMS (Root Mean Square) for roll and pitch, and 0.5° RMS for yaw. We note that these accuracies are usually sufficient for visual applications, may not be for haptic rendering.

We first measured the jitter of tracker output. The noise was small enough to be ignored (<0.20 mm RMS in position and <0.005° RMS in orientation), implying that feedback force will not suffer from the tracker jitter. We then measured the repeatability of the tracker output. For this, the tracker was fixed on one location. We then turned its power on, read its output, turned its power off, and repeated this procedure. Repeatability was then estimated as the standard deviation of the measurements. It was about 1.0 mm for position, and 0.33°, 1.02°, and 0.11° for pitch, yaw, and roll, respectively. The relatively inferior repeatability for yaw is problematic, since 0.5° yaw error induces about 1.13 mm position error when the HIP is at the center of the workspace (yaw is the rotation angle about the y-axis of the tracker on the mobile base in Figure 4).

The latency of the tracking system was also analyzed. We replaced the haptic interface tool of the PHANToM with the tracker such that its local coordinate frame was identical to the coordinate frame of the PHANToM. Then the mobile base was positioned to make the x-axis of the PHANToM aligned with the x-axis of the world coordinate frame. In this setup, the PHANToM output can be considered as a true value, and the tracker output as a delayed estimate. The PHANToM was actuated along the x-axis so that its position followed a sinusoid (50 mm amplitude and 0.5 Hz frequency). Then, the recorded x-positions of the PHANToM and the tracker were cross-correlated to estimate a phase difference between the two signals. Note that the yaw of the HIP also follows a sinusoid when the z-position of the HIP is sinusoidal, thus time delay for the yaw was also computed in the same setup. The same procedure was performed for y- and z-axes, and the whole procedure was repeated three times.

As a result, the mean tracking latency was about 38.56 ms in position, and 17.50 ms in orientation. With this amount of delay, the user may notice that the visual and haptic representations of the HIP are delayed from its actual physical location perceived via kinesthetic information [11], when the mobile base is in motion. Therefore, this latency needs compensation, and we use a simple predictor as $\hat{x}' = x + d + v$, where $\hat{x}'$ is the latency-compensated tracker position, x is the output of the tracking system, d is the average delay, and v is the velocity. Figure 5 shows an example of the delayed (original) tracker position and its compensated trajectory. The blue dashed and green solid lines represent the reference position from the PHANToM and the tracker position, respectively. The velocity was estimated by a moving average filter using 20 samples. The mean time delay is reduced to 1.9 ms after the latency compensation.

4.2 Kinematics Calibration

The transform from the coordinate frame of the PHANToM to the tracker frame on the mobile base, $\hat{W}H_P$, was designed to be $[0.0, -85.5, 123.0]^T$ (mm) for translation and $[0.0, 0.0, 0.0]^T$ (degrees) for orientation. However, manufacturing and assembly inject inevitable error in $\hat{W}H_P$. Thus, we found $\hat{W}H_P$ experimentally as follows.

For calibration, we replaced the haptic tool with an IS-900 tracker to directly measure the position of the HIP. We regard the HIP coordinate frame obtained using the tracker, $W\hat{H}_T$, as a true value, and that computed via the forward kinematics chain in (1), $W\hat{H}_P$, as an estimate. Then, we found an optimal $\hat{W}P$ that minimizes errors between $W\hat{H}_T$ and $W\hat{H}_P$. For that, we chose 27 points in a $3\times3\times3$ grid in the PHANToM workspace. The grid size was 280 × 460 × 116 (mm), and its center position was $[0.0, 220.0, 20.0]^T$ (mm). Then, we moved the HIP to each grid point using PID position control and sampled $W\hat{H}_T$ and $W\hat{H}_P$. This procedure was repeated five times. Before each repetition, the tracking system was restarted to average out the effect of tracker repeatability.

An error function, $\varepsilon$, was defined as the sum of the squared Euclidean distance between the positions:

$$\varepsilon = \sum_{i=1}^{5} \sum_{j=1}^{27} \varepsilon^2_{i,j},$$

where $\varepsilon_{i,j} = \|W_{i,j} - W_{i,j}'\|_2,$

$$= \|W_{i,j} - WR_{i,j}^T(R_{i,j}^TP_{i,j} + R_{i,j}^T) - W_{i,j}'\|_2,$$
each sample was measured in a randomly selected position in the PHANToM workspace. The tracking system was restarted in every 20 samples. We also modified the error function, $\epsilon$, to compensate the errors caused from the repeatability of the tracker on the mobile base and to focus more on the benefit of kinematics calibration. For this, we replaced $^W R_B$ and $^W T_B$ with their means $^W \bar{R}_B$ and $^W \bar{T}_B$, respectively:

$$ e'(j) = ||^W t_{H(j)} - ^W R_B( ^B R_B^{-1} p_{H(j)} + ^B R_B^{-1} - ^W t_B) ||. $$

(4)

where $j$ represents the sample number. The mean position error due to the incorrect kinematics was 6.48 mm before calibration, but reduced to 5.63 mm after calibration.

We also computed $\epsilon$ on the same test data set to obtain the overall performance of the HIP position estimation. Note that this metric contains all error sources including the errors in tracker output, $\epsilon$ was 7.76 mm on average before calibration, and decreased to 7.25 mm after calibration. As shown in Section 4.1, the accuracy and repeatability in the position of the IS-900 tracker was 3 mm and 1 mm, respectively. The accuracy and repeatability of yaw angle were 0.5° RMS and 1.02° RMS, respectively. Since 1.02° RMS yaw error of the tracker on the mobile base causes the position error of 2.56 mm when the HIP is at the center of the grid, the worst case error due to the tracker can be as high as about 8 mm. Therefore, we can conclude that the accuracy of HIP position estimation is almost the same to the best possible performance, that is bounded by the accuracy and repeatability of the tracker used in the calibration.

5 MOTION PLANNING OF MOBILE BASE

The mobile base of the pMHI moves the PHANToM in a position where a user can hold the haptic tool in a comfortable posture. For that, the motion planning module determines two velocity commands, one for the mobile base and the other for the vertical lift.

5.1 Algorithm

We describe the planning algorithm for the mobile base first. As usual, the configuration of the mobile base is defined by its 2D position $q_B = (x_B, y_B)$ and orientation $\theta_B$ in the world coordinate frame. This is measured by the tracker mounted on the mobile base. The planner uses two more inputs, the horizontal position of the user’s head measured by the head tracker mounted on the HMD, $q_U$, and the horizontal position of the HIP, $q_H$. For planning purpose, we model both of the user and the mobile base as circles with radii $r_U$ and $r_B$, respectively. This allows to further decompose the planning problem into two subproblems, one for position and the other for orientation.

A configuration space (C-space) for position planning is simply $C = \{ q_B \}$. Then, the configurations of the user that the mobile base collides with (called configuration space obstacle or C-obstacle) forms a circle centered at $q_U$ with radius $r_U + r_B$, which enables computations associated with C-obstacle mapping and collision detection to be executed in a constant time. If desired, the boundary of the mobile base workspace can be easily included as C-obstacles for the mobile base to stay inside the boundary.

At each time $n$, we determine a goal configuration of the mobile base at next tick $n + 1$, denoted by $q_B(n + 1)$. For this, we consider distance between the user’s head and the mobile base, $d_{UB} = ||q_U(n) - q_B(n)||$ and distance between the haptic tool held by the user’s hand and the mobile base, $d_{HB} = ||q_H(n) - q_B(n)||$ (see Figure 6). If $d_{UB} \geq d_{HB}$, which is more common, we attempt to the pMHI to face the user in a comfortable distance to grasp the tool (Figure 6(a)). Therefore, we set $q_B(n + 1)$ that it is on $q_Uq_H$ (a line spanned by $q_U$ and $q_H$) and $||q_H(n + 1) - q_U(n)|| = d_1^2$. We set $d_1^2$, a target distance between the user and the mobile base, considering the typical arm length of an adult. If $d_{UB} < d_{HB}$, which sometimes occurs when the user quickly approaches to the pMHI, we push the base to the opposite direction to obtain a safe distance. Thus, $q_B(n + 1)$ is a point such that it is on $q_Uq_B$ and $||q_H(n + 1) - q_U(n)|| = d_2^2$ (see Figure 6(b)). $d_2^2$ is usually chosen to be larger than $d_1^2$ for quick response.

To move the mobile base to the goal configuration, we use a velocity commanding algorithm based on the common potential field method. The velocity command $v(q_B)$ at mobile base configuration $q_B(n)$ is computed as:

$$ v(q_B) = c_{att} v_{att}(q_B) + c_{rep} v_{rep}(q_B), $$

(5)

$$ v_{att}(q_B) = \min(||d_{goal}(q_B)||, d_{max}) \frac{d_{goal}(q_B)}{||d_{goal}(q_B)||}, $$

(6)

$$ v_{rep}(q_B) = \frac{1}{\max(||d_{obs}(q_B)||, d_{min})} \frac{d_{obs}(q_B)}{||d_{obs}(q_B)||}. $$

(7)

The attractive velocity, $v_{att}(q_B)$, is proportional to the distance vector $d_{goal} = q_B(n + 1) - q_B(n)$ (from the current MHI position to the goal position). The repulsive term, $v_{rep}(q_B)$, is inversely proportional to the distance vector $d_{obs} = q_B(n) - q_B(n)$ (from the user’s position to the current MHI position). Note that more repulsive terms can be used if we include other obstacles such as the screens in the CAVE™. $d_{max}$ and $d_{min}$ are constants representing velocity limits in motor commands. The potential field is defined by $\nabla U(q_B) = -v(q_B)$, and an example is shown in Figure 7. The pMHI position moves on the potential field to the goal being repulsed by the C-obstacle of the user.

The motion planning algorithm for the orientation of the mobile base is rather simple. Given a desired orientation (also determined in the previous step), the algorithm uses the P control; the velocity command is proportional to the error between the desired and current orientations. We also use a dead zone. When the error is smaller than a pre-defined threshold, the velocity command is set to zero to prevent possible oscillations in the pMHI orientation.
A velocity command for the vertical position is computed in the same way; it is proportional to the difference between the vertical position of the HIP and the vertical position of the center point of the PHANToM workspace. This moves the lift upward if the haptic tool is held higher by the user, and vice versa.

A previous motion planning algorithm by Formaglio et al. used information about haptic tool position only, since their MHI self-contained a visual screen and did not use an external tracker [6]. This approach, although efficient, may fail to prevent a collision between a user and the MHI. Another recent algorithm for a two-handed MHI is similar to ours in determining the goal position of the MHI [16]. However, velocity commanding in both algorithms are based on the PD control. Adding other obstacles in the environment such as the screens of a display is not as straightforward as our algorithm that uses the traditional C-space approach.

5.2 Performance

We tested the performance of the motion planning algorithm extensively and obtained satisfactory results. An example is given in Figure 8 that shows the horizontal traces of the user, the HIP, and the pMHI recorded in a 85-second long experiment. In the experiment, the user was touching the virtual cow model (shown in Figure 1) positioned in the center of the workspace with its head oriented to the z-direction of the world coordinate frame. The radius of the mobile base, $r_B$, and that of the user, $r_U$, were set to 350 and 250 mm, respectively. The maximum velocity of the pMHI was 0.2 m/s for horizontal translation, 0.1 m/s for vertical translation, and 20 degree/s for rotation. The sampling rate of the motion planning algorithm was as fast as 200 Hz to provide smooth velocity commands to the motor controller of the mobile base. The figure shows that the user approached to the cow for initial contact and walked towards $+z$, $-x$, and then $-z$ directions of the world coordinate frame to feel the shape of the cow. We can confirm that for the user’s locomotion, the pMHI was always positioned to face the user, so that the PHANToM stayed in configurations with high manipulability and high accessibility.

6 Force Rendering

The MHI has a few challenging issues for haptic rendering. One of them is that the dynamics of the mobile base may affect the final force exerted by the haptic tool to the user’s hand, especially when the base moves with large and frequent acceleration/deceleration.

The previous studies approached this problem by designing the mechanism and controller to reduce the artifact in an open-loop control [12, 6]. Although valuable, it is not clear how successful this approach can be, thus we have employed a more aggressive option using closed-loop force control.

6.1 Algorithm

We embedded a 3D force/torque sensor in the ball-shaped haptic tool to measure interaction force between the tool and the user. The force was sampled at 10 kHz, smoothed with an average filter with window size 10 (thus effective sampling rate is 1 kHz) and a low-pass filter with 100-Hz cut-off frequency, and appropriately transformed to the world coordinate frame. Then a PID controller was applied as follows:

$$f_r(n) = f_d(n) + f_{PID}(n),$$

$$f_{PID}(n) = k_p f_r(n) + k_i \sum_{i=1}^{n} f_r(i) + k_d \Delta f_r(n),$$

$$f_r(n) = f_d(n-1) - f_r(n),$$

where $f_r(n)$ is a force command to the haptic interface at time $n$, $f_d(n)$ is the desired force determined by a haptic rendering algorithm, $f_{PID}(n)$ is a force compensation term computed by the PID control, $k_p$, $k_i$, and $k_d$ are the PID gains, $f_{eu}(n)$ is the current interaction force measured by the force sensor, and $f_r(n)$ is an error term between the desired and current forces. With appropriate choices of the PID gains, this closed-loop control can adequately compensate the error between the desired force computed by a haptic simulation engine and the actual force contaminated by the mobile base dynamics.

6.2 Performance

We experimentally evaluated the performance of the closed-loop force control. In the experiment, the haptic tool was inserted into a tightly-fitted hole in a styrofoam box. The PID gains were selected to be $k_p = 0.048$, $k_i = 0.006857$, and $k_d = 0.084$ using Ziegler-Nichols method followed by fine manual tuning. The mobile base was actuated with sinusoidal velocity commands of the amplitude of 0.12 m/s and the period of 4 second, which made the base move forward and back. The readings from the force sensor were recorded for 30 seconds. In a condition for regulation performance, desired force was fixed in the local coordinate frame of the PHANToM: 0.0 N in the $x$-axis, 0.63 N in the $y$-axis (for gravity compensation) and 2.5 N in the $z$-axis. For tracking performance, the sinusoidal force command with 1.5 N offset, 1.0 N amplitude, and 1 second period was given in the $z$-axis.

The results of the experiment are shown in Figure 9. Both results of regulation (the upper panel) and tracking (the lower panel) clearly show that the mobile base movements (period = 4 s) affect the rendering force in the open-loop control. When the closed-loop control was used, the effect was significantly compensated. Detailed statistics about the errors are summarized in Table 1.

However, the closed-loop force control could not completely compensate the effect of mobile base dynamics. For example, the force trajectory for regulation under the closed-loop control shows relatively slowly-changing drift that is synchronized with the movement of the mobile base. In addition, there are errors with higher frequency that appear to be caused by the nonlinear dynamics of the mobile base, e.g., those occur when the sub-wheel of the Mecanum wheel in contact with the ground changes from one to another. We, however, note that the control errors were not perceptible because they were mostly smaller than the just noticeable difference (JND) of force perception (0.2 N for 2.5 N reference force [13]). The error can be further improved by decreasing the response time of the PID controller, but the relatively inferior stability of the PHANToM premium model did not allow it. Haptic rendering tended to become unstable if we used higher PID gains due to the high-frequency resonance present in the PHANToM [3].
7 Conclusions

A new mobile haptic interface, POSTECH MHI (pMHI), has been introduced with emphasis on its system improvements. The pMHI uses a self-developed omni-directional robot that has four Mecanum wheels and a vertical lift to expand the workspace of a haptic interface in all three dimensions in the space. In terms of software, a few aspects have been significantly improved. First, the importance of accuracy in HIP position sensing was addressed, which was often ignored in the previous research. We carefully examined a performance of the 3D tracking system and calibrated the kinematics error. As a result, we obtained estimation accuracy very close to the best possible accuracy that is determined by the tracker performance. Second, a collision-free motion planning algorithm based on the potential field framework has been integrated. The algorithm is efficient, easily extendable, and has enhanced safety by regarding the user and other objects in an environment as obstacles to avoid. Its performance was also confirmed experimentally. Third, the effect of mobile base dynamics on the force that the user perceives has been practically removed by closed-loop force control. To our knowledge, this is a unique feature to the pMHI, leading to high-fidelity force rendering. All of these improvements allow the pMHI system to be able to render complex and large virtual objects in high fidelity.

Future work includes several immediate items. At present, we are designing haptic rendering algorithms for various haptic effects (such as friction and texture) which are robust to the relatively low accuracy of HIP position sensing. We also plan to test the performance of the pMHI system when it is applied to dynamic virtual environments that would require more abrupt movements of the mobile base. We will then integrate the pMHI to a CAVE™, and tackle (possibly) various issues that may be encountered with a see-through visual display.

Acknowledgements

This work was supported in parts by a grant R01-2006-000-10808-0 and a NRI grant R0A-2008-000-20087-0 both from KOSEF and by a ITRC support program C1090-0804-0002 from IITA, all funded by the Korea government.

References


Table 1: Statistics of the errors between commanded and measured forces. All numbers for force are in N.

<table>
<thead>
<tr>
<th>Regulation</th>
<th>Open-Loop Control</th>
<th>Closed-Loop Control</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Average (Open-Loop)</td>
<td>Average (Closed-Loop)</td>
</tr>
<tr>
<td>x</td>
<td>0.00034 0.0914</td>
<td>0.0009 0.0875</td>
</tr>
<tr>
<td>y</td>
<td>0.7288 0.0857</td>
<td>0.0022 0.0619</td>
</tr>
<tr>
<td>z</td>
<td>0.5762 0.1924</td>
<td>0.0042 0.0960</td>
</tr>
</tbody>
</table>

Figure 9: Comparisons of commanded and measured forces.