Received: 20 March 2018 | Accepted: 21 March 2018

DOI: 10.1002/cav.1814

SPECIAL ISSUE PAPER

WILEY

Encountered-type haptic display for large VR environment using per-plane reachability maps

Yaesol Kim | Hyun Jung Kim | Young J. Kim

Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea

Correspondence
Young J. Kim, Department of Computer Science and Engineering, Ewha Womans University, Seoul 03760, South Korea.
Email: kimy@ewha.ac.kr

Funding information
National Research Foundation, Grant/Award Number: 2017R1A2B3012701; Ministry of Culture, Sports and Tourism (MCST)/Korea Creative Content Agency (KOCCA), Grant/Award Number: CT R&D 2018

Abstract

We show a novel encountered-type haptic system, H-Wall, to enable haptic feedback using a manipulator of seven degrees of freedom suitable for simulating indoor virtual reality environments, which are characterized and confined by a set of vertical walls and revolving doors. At runtime, our system tracks hand motion using a red–green–blue depth sensor and locates its configuration. Then, the robotic manipulator plans a trajectory for the end effector, attached to a rectangular rigid board, to make contact with the hand to deliver a sense of touch as long as the perceived hand contact force is substantial. The force feedback is generated in a passive sense for static walls that the rigid board, corresponding to a vertical wall, holds its position as long as the perceived hand contact force is substantial. For a revolving door, the force feedback is generated in an active sense based on impedance control. In order to address the issue of limited workspace, we also propose a new reachability map, called *per-plane reachability map*, that is optimized to answer whether passive haptic feedback can be generated by a manipulator when the user touches a vertical wall at a given orientation. We successfully demonstrate our system to provide an illusion to the user in a virtual environment with touch sensation to the surrounding environment.

KEYWORDS

encountered-type haptic, haptic interaction, human–robot interaction, virtual reality

1 | INTRODUCTION

Haptics is an important area by itself and in physics-based animation and robotics. In particular, due to the recent surging interest in realizing immersive virtual reality (VR) environments, realistic haptic feedback is sought after more than ever before as part of efficient human–environment interaction. In particular, *encountered-type* haptic display or *robotic graphics* utilizes a robotic arm to deliver haptic feedback to users[1] without the need for conventional handheld interfaces to feel haptic feedback. This type of approach is appealing in covering a large workspace that is required for exploring indoor VR environments consisting of many vertical walls and revolving doors. Furthermore, by attaching different objects to the end effector, one can represent a wide variety of an object's geometries, as well as its textures to deliver haptic sensation.

In this paper, we show our H-Wall system to enable encountered-haptic feedback suitable for simulating indoor virtual environments using a robotic manipulator of seven degrees of freedom (DoFs). Because indoor environments are often characterized and confined by a set of vertical walls and revolving doors, we model the walls and doors in our virtual environment using, possibly revolving, rectangular primitives and realize the encountered-haptic feedback between human hands and the rectangular primitives. At runtime, as a user explores the virtual environment, our system tracks the user's hand movement using a red–green–blue depth (RGBD) camera and locates its configuration. Then, the robotic manipulator plans a trajectory for the end effector, attached to a rectangular rigid board, to make contact with the hand to deliver a sense of touch. Force feedback is generated in a passive sense that the rigid board holds its configuration corresponding to the static object as long as the perceived hand contact force is substantial; for dynamic objects, the force is generated by altering the configuration of the objects.

Moreover, because the limited size of haptic workspace can be a serious problem for encountered-type haptics, we propose a new workspace analysis method and representation, called *per-plane reachability maps*. These maps are computed as an offline process and are used at runtime to see if the manipulator can generate haptic feedback at the user's location. These maps are generated by discretizing the workspace by sampling orientations to a proper number and finding their boundaries by performing nonlinear optimization. In our experiments, we show that this finite sampling of orientations works well, as the user is immersed into VR space and a small change in the user's orientation does not affect the believability of the H-Wall system. We implement our system using a KUKA IIWA R800 robot and the Kinect sensor and successfully demonstrated to provide an illusion to the user in a virtual environment with touch sensation to the surrounding environment.

The rest of this paper is organized as follows. In Section 2, we survey works relevant to encountered-type haptics. We give an overview of our H-Wall system in Section 3, show the details of the H-Wall system in Section 4, and explain how we compute the per-plane reachability maps in Section 5. In Section 6, experimental results on the H-Wall system are provided, and the paper is concluded in Section 7.

2 | PREVIOUS WORK

In this section, we briefly survey works relevant to realizing virtual walls using encountered-type haptics.

2.1 | Haptic interface for VR

Even though the sense of touch is a key factor for enhancing immersion in VR, methods for haptic interface are still in their early stage.[2] The most prevalent forms of force feedback interface are desktop-based ones such as pen-based interface[3,4] and haptic master,[5] which is a tool-handling-type interface. This type of interface tends to be compact and easy to install, but it can hinder the user's feeling of immersion in VR because it reduces the user's freedom of movement due to the limited workspace of the device.[6] Portable force feedback interfaces such as arm exoskeleton type[7] or hand master types[8] can be used for increasing the user's workspace to some degree.[6] Because the portable system must be light and comfortable to allow for users' free motion when virtual objects are not touched,[9] it is very difficult to design portable haptic interfaces that way.

2.2 | Encountered-type haptic

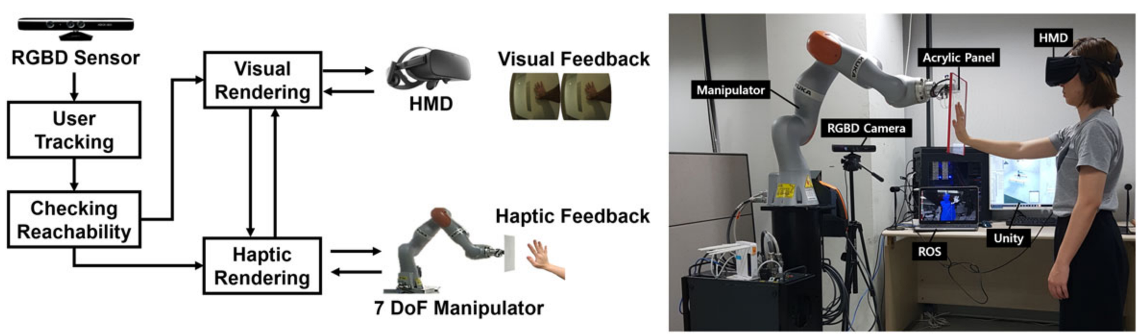
McNeely first proposed a concept of robotic graphics or encountered-type graphics to express the analogy between robots simulating haptic feedback and graphic displays simulating visual feedback.[1] Yokokohji et al. proposed a *what-you-see-is-what-you-feel* display for a visual/haptic interface to the virtual environment that registers visual and haptic feedback using vision-based tracking and PUMA 560 manipulator as an encountered-type haptic interface.[10] This system supports free-to-touch and move-and-collide haptic sensations. Colgate et al. proposed a way in coupling between the haptic display and the virtual environment using passivity techniques. They provide a criterion for the passivity of a virtual wall.[11] Araujo et al. proposed the *snake charmer* system in the context of VR, where a user with a head-mounted display (HMD) on can experience haptic feedback of various objects encompassing shape difference, surface and texture characteristics, and even temperature. They build their system using commodity hardware, such as Oculus DK2 and Robai robotic arm.[12] However, none of the previous works cover a large haptic workspace due to the limited DoF and workspace of the robotic manipulator, and more importantly, they did not provide any mechanism to determine whether haptic feedback can be generated or not for a given user's configuration. Finally, the used robotic manipulator in the prior work is not designed for human/robot coexistence (or cobot) and thus can be unsafe for encountered haptic interfaces. Yokokohji et al.[13] and Shigeta et al.[14] covered relatively small haptic workspace for small virtual objects that can be handheld and do not provide any mechanism to determine whether haptic feedback can be generated or not for a given user configuration. Compared with our system, VRRobot[15] provides only passive haptic feedback.

2.3 | Workspace analysis

The workspace of a robot manipulator can be classified into dexterous workspace, reachable space, and orientation workspace.[16] According to classifications, various workspace analysis methods were proposed. To analyze reachable space, many works were based on sampling. Zacharias et al. proposed a capability map that represents the kinematic reachability and the directional structure information of a robot arm.[17] They sampled points inside the working envelope and solved the reachability and directional structure of a point set. Using the capability map, a manipulator is able to deduce places that are easy to reach and plan an optimal path. Dong et al. proposed an orientation-based reachability map representing kinematic reachability for sampled orientations. The orientation-based reachability map needs to be computed only once for a given robot arm structure so that the end effector can be extended online.[18] Reachable space generation using the Monte Carlo method was proposed by Guan et al.[19] They presented a numerical approach using a random sampling method to avoid the conventional analytical method that is impractical for analyzing the workspace of a humanoid robot. A reachability inversion method was proposed by Vahrenkamp et al. to solve the base placement problem using a reachability map.[20] They introduced the oriented reachability map based on inverse reachability data for the target pose. Zacharias et al.[21] proposed an algorithm to position a mobile manipulator to generate 3D trajectories using a discrete representation of the reachable workspace. Existing approaches to analyze workspace using a reachability map are not very efficient or robust with respect to varying orientations to implement encountered-type haptics, where reachability should consider both the end effector's positions and orientations.

3 | SYSTEM OVERVIEW

In Figure 1, we show an overview of our encountered-haptic system consisting of visual rendering and haptic rendering. For visual rendering, an indoor virtual environment consisting of a few walls and doors is rendered to the HMD that the user wears. Using user tracking data and the HMD's orientation tracking data, the user's virtual proxy hand and virtual environments are rendered in real time. A seven-DoF manipulator, capable of human–robot coexistence, operates as a haptic device to provide touch feedback to the user. Using user tracking data, a wall panel attached to the end effector of the manipulator serves as a proxy of the virtual wall following human hands. The manipulator relies on contact force data measured from the user to hold its configuration for the user to touch a static wall. In addition, the manipulator can provide dynamic interaction for the user to open a virtual door. The states of an avatar in VR and of a user in the physical world are shared by haptic and visual rendering components. The overall system works under real-time constraints of tens of milliseconds.



(a) (b)

FIGURE 1 H-Wall system. (a) System diagram. (b) System setup. RGBD = red–green–blue depth; HMD = head-mounted display; DoF = degree of freedom; ROS = Robot Operating System

4 | HAPTIC WALL

In this section, we describe the haptic feedback mechanism of our H-Wall system using per-plane reachability maps, which will be discussed in Section 5.

4.1 | User tracking using an RGBD camera

To determine the end-effector location of the manipulator and the location of an avatar hand, the user's head and hand need to be tracked at runtime. This is achieved by using an RGBD camera to identify the precise location of the head and hands. More specifically, the user's geometry data are captured as a point cloud, and the head center and the palm of a hand are detected and tracked. Finally, the head and hand positions are transformed from the camera frame to the robot's base frame.

4.2 | Passive haptic feedback

A seven-DoF manipulator serves as a passive haptic device based on both position-based control and force-based control. The end effector of the manipulator follows the hand position by maintaining its configuration toward the virtual wall. Torque sensors embedded in the robot's joints are used to detect the user contact forces $\mathbf{f}$ in a standard way as follows:

$$\tau = J^T\mathbf{f}, \tag{1}$$

where J is the manipulator Jacobian. Then, we project $\mathbf{f}$ toward the vertical direction of the virtual wall to obtain the orthogonal component $\mathbf{f}^{\perp}$ of $\mathbf{f}$. When $\mathbf{f}^{\perp}$ is sufficiently high, the manipulator holds its configuration to provide passive haptic feedback to the user.

4.3 | Active haptic feedback

In case of an avatar trying to open the virtual door, a seven-DoF manipulator serves as an active haptic device based on impedance-based control.[22] To provide the user force feedback due to pushing the door open, the manipulator gives itself a margin $\delta\mathbf{x}$ in the normal direction of the board. Then, the user can feel the impedance forces $\mathbf{f}$ in a normal direction of the virtual door as follows:

$$\mathbf{f} = k\delta\mathbf{x}, \tag{2}$$

where k is the spring stiffness and $\delta\mathbf{x}$ is the displacement of the end effector caused by the user's push motion.

4.4 | Avatar control in VR

In order to control a virtual avatar interactively and to expand the limited workspace of our encountered-type haptic system, we employ a motion control interface using the Oculus Rift controller. Specifically, a user can control the avatar's position by the joystick and rotate an avatar by pushing the buttons in the controller. The user uses his/her left hand in operating the controller while the user can feel haptic feedback freely by his/her right hand.

5 | HAPTIC WORKSPACE

Just like any other haptic system, our encountered-type haptic system also has finite workspace. However, the limited size of workspace can be more serious for encountered-type haptics, as it may hinder the user's immersion into a large virtual space. Furthermore, without having the accurate size information of haptic workspace, the encountered-type haptic system may not deliver the user's anticipated haptic feedback to the desired location, which makes the utility of the haptic system low. In order to deal with such a limited workspace problem in our virtual environment setting where vertical walls play a dominant role in defining the space, we calculate the workspace of the haptic system using per-plane reachability maps as an offline process. In this section, we explain our optimization-based methods to compute a per-plane reachability map for our encountered-type haptic system, as well as how to use the map at runtime.

5.1 | Reachability map

The reachable space of a robot manipulator is defined as a set of points that can be reached by a reference point $\mathbf{p}$ with respect to the world frame $\mathcal{W}$. We attach the reference point $\mathbf{p}$ to an origin of the end-effector frame $\mathcal{E}$ and assume that the world frame aligns with the robot's base frame $\{\mathbf{0}\}$. The configuration of the end-effector frame can be attained by calculating the forward kinematics of the manipulator and is represented by a homogeneous matrix as follows:

$$\mathbf{T}_{\mathcal{E}} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \in SE(3), \tag{3}$$

where $\mathbf{R} \in SO(3)$ and $\mathbf{p} \in \mathbb{R}^3$. For a serial manipulator with n links/joints, the position $\mathbf{p}$, and orientation $\mathbf{r}$, represented in Euler angle, of $\mathcal{E}$ are defined as vector functions of the following:

$$\mathbf{p} = (p_x, p_y, p_z) = \mathbf{f}_{\mathbf{p}}(q_1, q_2, \dots, q_n) \tag{4}$$

$$\mathbf{r} = (\alpha, \beta, \gamma) = \mathbf{f}_{\mathbf{r}}(q_1, q_2, \dots, q_n), \tag{5}$$

where $q_1, q_2, \dots, q_n$ are the joint parameters, and $\mathbf{f}(\mathbf{q})$ is a nonlinear function of n joint parameters (i.e., the forward kinematics map). A set of possible values of $\mathbf{p}$ for all $q_1, q_2, \dots, q_n$ defines a 3D reachability map of the manipulator.

In order to calculate reachability accurately, all possible positions and orientations in robot workspace should be considered. However, practically, this will be very hard, as the forward kinematics map is highly nonlinear, and calculating inverse kinematics precisely (or algebraically) is impractical. Thus, in our case, instead of computing reachability precisely, we merely approximate it by discretizing the orientations and positions, and we extract and store its boundary points using optimization.

5.2 | Per-plane reachability maps

In our H-Wall system, we characterize the virtual environment by vertical wall surfaces that are planes only with a finite number n_θ of orientations. Thus, we sample the entire manipulator workspace by n_θ different orientations. This sampled subworkspace is sampled again into equidistant planes called per-plane reachability maps, as illustrated in Figure 2. We also compute the boundary points of each per-plane reachability map.

Formally, the configuration space of the end effector $\mathbf{T}_{\mathcal{E}}$ is sampled into a discrete set of cardinality n_θ as follows:

$$\mathbf{T}_{\mathcal{E}} \cong \left\{ \mathbf{T}_{\mathcal{E}}^1, \mathbf{T}_{\mathcal{E}}^2, \dots, \mathbf{T}_{\mathcal{E}}^i, \dots, \mathbf{T}_{\mathcal{E}}^{n_\theta} \right\}, \tag{6}$$

where n_θ is the number of sampled orientations and

$$\mathbf{T}_{\mathcal{E}}^i = \begin{bmatrix} \mathbf{R}(\alpha^i, \beta^i, \gamma^i) & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix} \tag{7}$$

with some fixed orientation $\alpha^i, \beta^i, \gamma^i$. Then, $\mathbf{T}_{\mathcal{E}}^i$ is further sampled into a discrete set of cardinality M as follows:

$$\mathbf{T}_{\mathcal{E}}^i \cong \left\{ \mathbf{T}_{\mathcal{E}}^{i_1}, \mathbf{T}_{\mathcal{E}}^{i_2}, \dots, \mathbf{T}_{\mathcal{E}}^{i_k}, \dots, \mathbf{T}_{\mathcal{E}}^{i_M} \right\}, \tag{8}$$

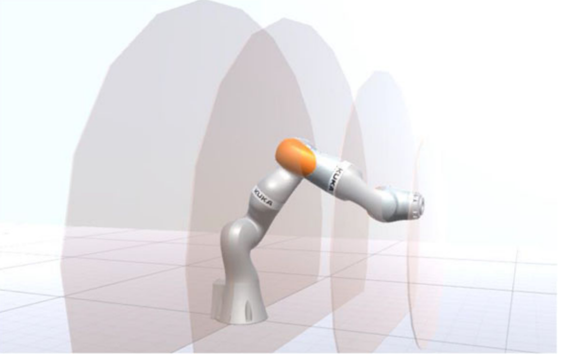
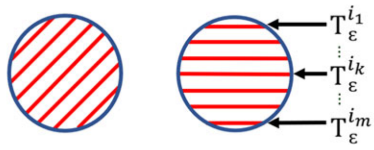


FIGURE 2 Four per-plane reachability maps of KUKA IIWA with a fixed end-effector orientation are shown in parallel gray circular planes. The boundary of these maps will be sampled to point clouds



(a) (b) (c)

FIGURE 3 The concept of workspace sampling illustrated by 2D projection. Each blue circle represents the workspace sampled by n_θ different orientations. The red lines $\mathbf{T}_{\mathcal{E}}^{i_k}$ represent a sampled plane with a fixed orientation, called per-plane reachability map. (a) $T_{\mathcal{E}}^1$. (b) $T_{\mathcal{E}}^i$. (c) $T_{\mathcal{E}}^{n_\theta}$

where M is the number of sampled planes and

$$\mathbf{T}_{\mathcal{E}}^{i_k} = \begin{bmatrix} \mathbf{R}(\alpha^i, \beta^i, \gamma^i) & \begin{bmatrix} p_x \\ p_y \\ p_z^{i_k} \end{bmatrix} \\ \mathbf{0} & 1 \end{bmatrix} \tag{9}$$

with some fixed value of $p_z^{i_k}$. We refer to each of $\mathbf{T}_{\mathcal{E}}^{i_k}$ as a per-plane reachability map. In our implementation, we find the extremal values of $(p_x, p_y, p_z^{i_k})$ in $\mathbf{T}_{\mathcal{E}}^{i_k}$ and store them at a lookup table (the blue circles in Figure 3).

5.3 | Optimization-based computation

In order to calculate per-plane reachablility map $\mathbf{T}_{\mathcal{E}}^{i_k}$, we find its constraints in terms of the position and orientation of the end effector. We formulate these constraints as a set of equality constraints that the robot manipulator must satisfy as follows:

$$C_{eq} = \begin{cases} \alpha = 0 \\ \beta = 0.5\pi \\ \gamma = \gamma^i \\ p_x - \tan(\gamma^i)p_y = d_i \\ p_z = p_z^{i_k}, \end{cases} \tag{10}$$

where α and β are constants characteristic to represent vertical walls, γ^idefines the orientation of the end effector corresponding to the orientation of the given wall, d_i defines the distance of the wall from the robot base, and $p_z^{i_k}$ is the sampled position constraint in z. In our implementation, the orientations γ of walls relative to the user are discretized to some value (e.g., seven), as a finite set of γ^i provides sufficient granularity of the user's rotational motion to generate haptic feedback when the user is immersed into virtual space.

Based on the aforementioned constraint formulation, we set up an objective function to calculate extremal y values p_y subject to given C_{eq}. p_y can be a nonlinear function of joint parameters as follows:

$$p_y = \mathbf{f}_y(q_1, q_2, \ldots, q_n), \tag{11}$$

and the extremities of p_y correspond to the boundary of the reachable plane. Given such a constraint formulation, we solve an optimization problem in terms of p_y. To solve this nonlinear optimization problem, we combine a global search algorithm with the multistart method[23] to get optimal results. The multistart method chooses several starting points for a traditional nonlinear problem solver by randomly choosing starting points defined by the upper/lower bounds of optimization variables. As a choice for the global optimization solver, we use the MATLAB global optimization toolbox. By considering the working envelope of the manipulator, we bound the workspace by an axis-aligned box, divide the box by planes in parallel to the y–z plane of $\mathcal{W}$, and find the extremities of each plane. Because this bound is conservative, some p_y contained in the box may not satisfy C_{eq}. We filter out these results from a reachability map by checking whether the optimization results satisfy the orientation constraints for each step of optimization.

5.4 | Representing and using reachability maps

Per-plane reachability maps are stored in a 3D array. The array includes the distance of the reachable plane from the robot base, the z value of the plane, and the extremities of y value. This structure of a reachability map enables the H-Wall system to look up reachability for a given input configuration of the end effector rapidly, which then generates passive or active encountered-type haptic feedback to the user.

In runtime, the precomputed per-plane reachability map is used to check if the user's hand pose is realizable by a map entry or not. Given an input pose consisting of a hand position and a user orientation, the H-Wall system checks for the orientation first and selects a corresponding reachability plane. Then, it checks for the position (x value and z value in order) and sees if the y value is between the extremities of the y value in the precomputed per-plane reachability map. If the input value is valid to provide haptic feedback, the manipulator position and avatar hand position are updated; otherwise, both the manipulator and the avatar maintain their position and wait for the next update of an input value.

Inspired by redirected walking,[24,25] we distort the user's orientation to give her/him a fake sense of a large virtual environment and generate haptic feedback with an approximated orientation. Specifically, when the user tries to touch a virtual wall, the orientation of the wall relative to the avatar $\mathbf{r}_t$ is calculated and approximated to $\mathbf{r}_a$. The difference between $\mathbf{r}_a$ and $\mathbf{r}_t$ is calculated by corresponding β values, $\beta_a - \beta_t$, and the avatar is reoriented by this difference so that the orientation of the wall relative to the reoriented avatar is mapped to a sampled orientation residing in per-plane reachability maps.

6 | RESULTS AND DISCUSSION

In this section, we describe the implementation details of our H-Wall system and show experimental results. We also briefly discuss the user study to verify the effectiveness of our system.

6.1 | System implementation details

In Figure 1b, we show our H-Wall system setup. We implemented our system using Unity3D under a Windows 10 64-bit operating system equipped with a 4.0-GHz Intel Core i7 CPU, GeForce GTX 970 GPU, and 24-GB RAM, and ROS indigo framework under an Ubuntu 14.04 LTS 64-bit operating system equipped with a 2.1-GHz Intel Core i7 CPU and an 8-GB RAM. We use Kinect for Windows as a camera with OpenNI package to generate point cloud data and track the user's hand. KUKA LBR IIWA 7 R800 is used as a haptic manipulator, which has seven DoFs, and torque sensors are integrated into all seven joints. ROS and Sunrise OS are used to perform haptic and tracking calculation and to control the robot, respectively. Each of these components is programmed in C++ and Java programming languages and communicated over Ethernet via a message-passing protocol. Unity3D and Oculus Rift CV1 are used to visually render 3D virtual scene and track the orientation of the user's head. Our rendering unit in Unity3D is programmed in C# and communicates with a ROS unit over WiFi connection. MATLAB R2017 is used to compute the end effector's pose using forward kinematics and solve nonlinear optimization with equality constraints to build per-plane reachability maps. The reaction time of the robot manipulator and the latency of stereo visual rendering are 22 ms and 11.7 ms, respectively, whereas the network latency is negligible. Note that the H-wall system is not very sensitive to the manipulator reaction time because the user does not frequently change her/his contact status with respect to the end effector.

As a haptic proxy for a virtual wall, a transparent and lightweight acrylic panel is attached to the IIWA end effector and follows the human hand in highly real time, providing a sense of illusion of touching a huge vertical wall. The transparency of the panel makes the user's hand always visible to the RGBD camera.

An example of a reachability map is shown in Figure 4. Assuming that the user always touches the virtual walls in front of him/her, we discretize the workspace of the manipulator by uniformly sampling the orientations of the walls (planes) at an angle of 15 degrees and build seven per-reachability maps. The orientation of the user's input hand pose is approximated to one of the seven orientations, and the H-Wall system searches a per-plane reachability map closest to this approximate value. Each per-plane reachability map is represented in different color and bounded by boundary points with the same color in Figure 4. The reachability map is precomputed, and at runtime, the H-wall system accesses the per-plane reachability map structure for a fast lookup of the reachability of the robot end effector's configuration. Our per-plane reachability maps take around an hour to compute per orientation, each having 38 reachability planes per orientation.

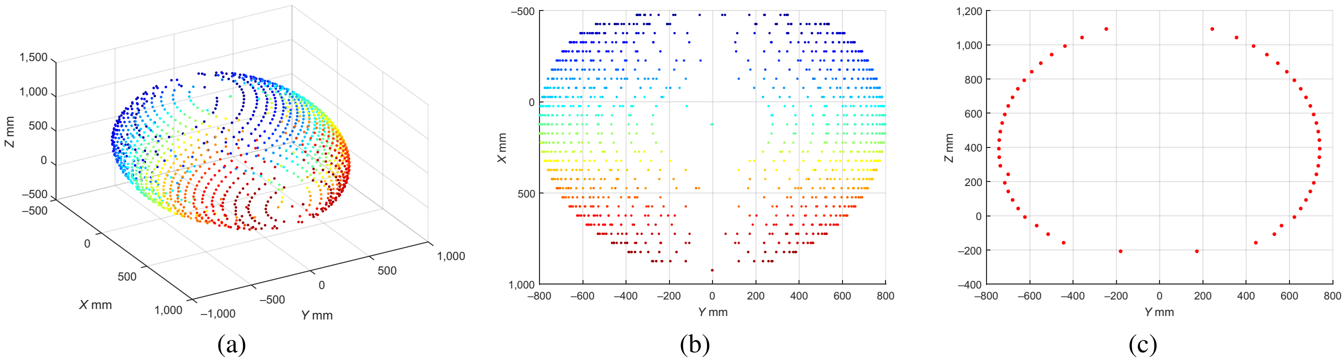


FIGURE 4 Per-plane reachability maps with $\gamma = 0$, $i = 4$. (a) $\mathbf{T}_{\mathcal{E}}^{4}$. (b) Projection of $\mathbf{T}_{\mathcal{E}}^{4}$ onto the x–y plane. (c) $\mathbf{T}_{\mathcal{E}}^{4_{10}}$

6.2 | An interaction condition for user experiment

To verify the effectiveness of our H-Wall system, we conducted a user study. The experiment was divided into two parts: (a) testing whether a user can distinguish between a real wall and H-Wall and (b) evaluating the haptic feedback of H-Wall. In both experiments, subjects wear an HMD to see the VR environment and interact with virtual objects with haptic feedback. The experiments were carried out with eight volunteers aged 23 to 28.

6.2.1 | Experiment 1: distinguish walls

The aim of Experiment 1 is to evaluate whether H-Wall is suitable for haptic feedback in comparison with a physical wall. As shown in the Figure 5, the participated subjects are asked to interact with a virtual whiteboard in the VR environment and presented with an actual whiteboard and H-Wall separately. Then, they are asked to respond to the questions shown in the Table 1.

According to this experiment, the resulting score of responses was 3.125 in average, meaning that most of the participants felt H-wall was realistic just like a real whiteboard.

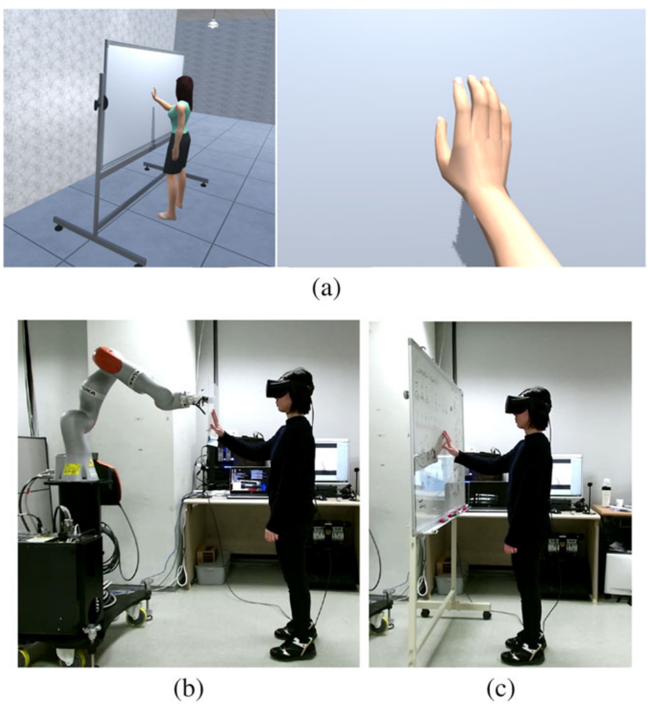


FIGURE 5 Test setup for the user study. (a) Interaction with a whiteboard in virtual reality. (b) H-Wall. (c) Physical whiteboard

TABLE 1 Experiment 1 Questionnaire

Question
Which wall felt more realistic?
(1) Whiteboard was more realistic.
(2) Whiteboard was realistic.
(3) Hard to tell.
(4) H-Wall was realistic.
(5) H-Wall was more realistic.

6.2.2 | Experiment 2: interaction

The goal of Experiment 2 is evaluating the level of immersion into VR with H-Wall. The VR environment for this experiment is modeled after an indoor gym with touchable vertical walls, mirrors, and doors. During the experiment, subjects were asked to control their avatar's position using an HMD controller by their left hand. When the avatar moves close to touchable objects in VR, after the user walks toward the objects, the user can touch or open them with their right hand, as shown in Figure 6. Then, the subjects were asked to score their subjective feelings of haptic feedback, as shown in the Table 2, from 1 (worst) to 5 (best).

In this experiment, the participants show positive feedback about the H-Wall system. The average response point to the question about the naturalness of interaction is 4.25. For the questions about the believability of static walls and dynamic doors, the scores were 4.125 and 4.375 in average, respectively. It is interesting to note that our H-wall provides even more realistic feedback for dynamic cases than static ones, which suggests a new interesting future research direction for fully dynamic encountered-type haptic feedback.


(a)

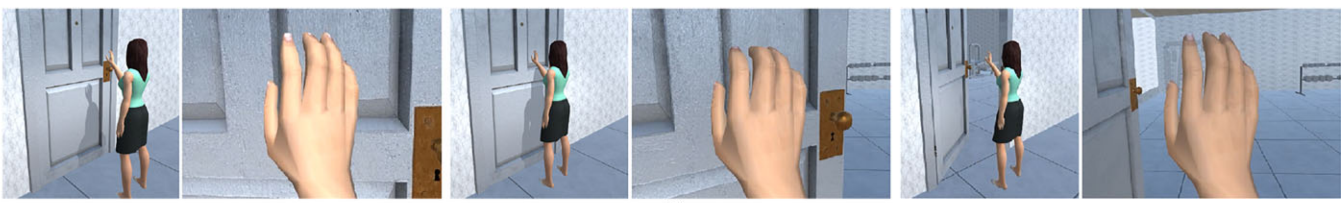

(b)

FIGURE 6 Active haptic interaction sequences in H-Wall. (a) Haptic feedback in physical space. (b) Visual feedback in virtual reality space

TABLE 2 Experiment 2 Questionnaire

Questions
How natural was the interaction with H-Wall?
How believable was the static virtual wall?
How believable was the dynamic virtual door?

7 | CONCLUSION

We have presented an effective and simple solution to encountered-haptic interfaces for VR environments using a seven-DoF manipulator. Our system relies on an RGBD camera for motion tracking of the hand and force-controlled robotic manipulation of an end effector equipped with a rigid board, acting as a stiff virtual wall in the virtual environment. There are a few limitations in H-Wall that we plan to address in the future. Currently, our hand tracking system cannot cover a large motion space. The types and textures of touchable objects need to be expanded to be used for general virtual environments.

ACKNOWLEDGEMENTS

This project was supported in part by the National Research Foundation (NRF; 2017R1A2B3012701) and the Ministry of Culture, Sports and Tourism (MCST)/Korea Creative Content Agency (KOCCA; CT R&D 2018) in South Korea.

ORCID

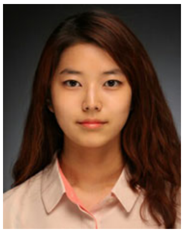
Yaesol Kim http://orcid.org/0000-0003-0374-9570
Young J. Kim http://orcid.org/0000-0003-2159-4832

REFERENCES

1. McNeely WA. Robotic graphics: a new approach to force feedback for virtual reality. Paper presented at: Proceedings of IEEE Virtual Reality Annual International Symposium; 1993 Sep 18–22; Seattle, WA. IEEE; 1993. p. 336–341.
2. Lin MC, Otaduy M. Haptic rendering: foundations, algorithms, and applications. Natick, MA: CRC Press; 2008.
3. Iwata H, Noma H. Volume haptization. Paper presented at: Proceedings of 1993 IEEE Research Properties in Virtual Reality Symposium; 1993 Oct 25–26; San Jose, CA. IEEE; 1993. p. 16–23.
4. Massie TH, Salisbury JK. The phantom haptic interface: a device for probing virtual objects. Paper presented at: Proceedings of the ASME Dynamic Systems and Control Division; 1994 Nov; Chicago, IL. American Society of Mechanical Engineers; 1994. p. 295–301.
5. Iwata H. Desktop force display. Paper presented at: SIGGRAPH 94 Visual Proceedings; 1994; Orlando, FL.
6. Burdea GC. Force and touch feedback for virtual reality. New York, NY: John Wiley & Sons, Inc; 1996.
7. Bergamasco M, Allotta B, Bosio L, et al. An arm exoskeleton system for teleoperation and virtual environments applications. Paper presented at: Proceedings of the 1994 IEEE International Conference on Robotics and Automation; 1994 May 8–13; San Diego, CA. IEEE; 1994. p. 1449–1454.
8. Iwata H, Nakagawa T, Nakashima T. Force display for presentation of rigidity of virtual objects. J Robotics. 1992;4(1):39–42.
9. Marcus B. Sensing, perception and feedback for VR. Paper presented at: Proceedings of VR System Conference; 1993 Oct; New York, NY.
10. Yokokohji Y, Hollis R, Kanade T. WYSIWYF display: A visual/haptic interface to virtual environment. Presence: Teleoperators Virtual Environ. 1999;8(4):412–434.
11. Colgate JE, Grafing PE, Stanley MC, Schenkel G. Implementation of stiff virtual walls in force-reflecting interfaces. Paper presented at: Proceedings of IEEE Virtual Reality Annual International Symposium; 1993 Sep 18–22; Seattle, WA. IEEE; 1993. p. 202–208.
12. Araujo B, Jota R, Perumal V, Yao JX, Singh K, Wigdor D. Snake charmer: Physically enabling virtual objects. Paper presented at: International Conference on Tangible, Embedded, and Embodied Interactions; 2016 Feb 14–17; Eindhoven, The Netherlands. New York, NY: ACM; 2016. p. 218–226.
13. Yokokohji Y, Kinoshita J, Yoshikawa T. Path planning for encountered-type haptic devices that render multiple objects in 3D space. Paper presented at: Proceedings IEEE Virtual Reality; 2001 Mar 13–17; Yokohama, Japan. IEEE; 2001. p. 271–278.
14. Shigeta K, Sato Y, Yokokohji Y. Motion planning of encountered-type haptic device for multiple fingertips based on minimum distance point information. Paper presented at: Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'07); 2007 Mar 22–24; Tsukuba, Japan. Washington, DC: IEEE; 2007. p. 188–193.
15. Vonach E, Gatterer C, Kaufmann H. VRRobot: Robot actuated props in an infinite virtual environment. Paper presented at: 2017 IEEE Virtual Reality (VR); 2017 Mar 18–22. p. 74–83.
16. Siciliano B, Khatib O. Springer handbook of robotics. Berlin, Germany: Springer; 2016.
17. Zacharias F, Borst C, Hirzinger G. Capturing robot workspace structure: Representing robot capabilities. Paper presented at: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2007 Oct 29–Nov 2; San Diego, CA. IEEE; 2007. p. 3229–3236.
18. Dong J, Trinkle JC. Orientation-based reachability map for robot base placement. Paper presented at: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2015 Sep 28–Oct 2; Hamburg, Germany. IEEE; 2015. p. 1488–1493.
19. Guan Y, Yokoi K. Reachable space generation of a humanoid robot using the Monte Carlo method. Paper presented at: 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems; 2006 Oct 9–15; Beijing, China. IEEE; 2006. p. 1984–1989.
20. Vahrenkamp N, Asfour T, Dillmann R. Robot placement based on reachability inversion. Paper presented at: 2013 IEEE International Conference on Robotics and Automation; 2013 May 6–10; Karlsruhe, Germany. IEEE; 2013. p. 1970–1975.

21. Zacharias F, Sepp W, Borst C, Hirzinger G. Using a model of the reachable workspace to position mobile manipulators for 3-d trajectories. Paper presented at: 2009 9th IEEE-RAS International Conference on Humanoid Robots; 2009 Dec 7–10; Paris, France. IEEE; 2009. p. 55–61.
22. Hogan N. Impedance control: An approach to manipulation. Paper presented at: 1984 American Control Conference; 1984 Jun 6-8; San Diego, CA. IEEE; 1984. p. 304–313.
23. Ugray Z, Lasdon L, Plummer J, Glover F, Kelly J, Martí R. Scatter search and local NLP solvers: A multistart framework for global optimization. Informs J Comput. 2007;19(3):328–340.
24. Razzaque S, Kohn Z, Whitton MC. Redirected walking. Paper presented at: Proceedings of EUROGRAPHICS; 2001; Manchester, UK. p. 105–106.
25. Sun Q, Wei L-Y, Kaufman A. Mapping virtual and physical reality. ACM Trans Graph. 2016;35(4):64.

Yaesol Kim is currently a PhD student of computer science and engineering at Ewha Womans University. She received her BS in computer science and engineering in 2016 from Ewha Womans University. Her research interests include virtual reality, robotics, human robot interaction and haptics.



Hyung Jung Kim is currently a MS student of computer science and engineering at Ewha Womans University. She received her BS in computer science and engineering in 2017 from Ewha Womans University. Her research interests include mixed reality, robotics, human robot interaction and robot simulation.

Young J. Kim is an Ewha fellow professor of computer science and engineering at Ewha Womans University. He received his PhD in computer science in 2000 from Purdue University. Before joining Ewha, he was a postdoctoral research fellow in the Department of Computer Science at the University of North Carolina at Chapel Hill. His research interests include interactive computer graphics, computer games, robotics, haptics, and geometric modeling. He has published more than 90 papers in leading conferences and journals in these fields. He also received the best paper awards at the ACM Solid Modeling Conference in 2003, the International CAD Conference in 2008 and the HCI Korea Conference in 2016, and the best poster award at the Geometric Modeling and Processing conference in 2006.

How to cite this article: Kim Y, Kim HJ, Kim YJ. Encountered-type haptic display for large VR environment using per-plane reachability maps. *Comput Anim Virtual Worlds*. 2018;e1814. https://doi.org/10.1002/cav.1814